Scrum – Unit 02

- Agenda
  - Scrum Methodology
  - Characteristics of Scrum
  - Scrum Terminologies: Scrum Team (Product Owner, Scrum Master, Cross Functional Team)
  - Sprint, User Story, Epics, Product Backlog, Sprint Backlog
  - Story Points, Burn Down Charts, Velocity, Definition of Done, Time Boxing
  - Activities in Scrum: Planning Meeting, Execution Of Sprint Tasks, Daily Stand Up, Review Meeting, Retrospective Meeting

Scrum Methodology

- Scrum is Agile Development framework and has several key features that are shown the Figure given below and explained in detail.
- Scrum is a simple empirical process that enables teams to build products incrementally in iterations, to keep along with of the changing market needs and align themselves to the organization's business goals.
- *"Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.“*
-

Scrum Methodology

- Scrum advocates self-organizing teams working towards a common goal through continuous inspection and adaptation.
- A minimum viable product at the end of each iteration provides an option for the teams to quickly get feedback from end users and respond accordingly much faster.
- Scrum should not be viewed as a collection of practices but rather as a culture or a set of values

Characteristics of Scrum

- Self-organizing teams
- Product progresses in a series of month-long “sprints”
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects

Scrum Theory

- Scrum employs an iterative, incremental approach to optimize predictability and control risk upon three pillars of empirical process control:
- Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation
- **Transparency:** Significant aspects of the process must be visible to those responsible for the outcome. Transparency requires those aspects be defined by a common standard so observers share a common understanding of what is being seen.

Scrum Theory

- For example
- A common language referring to the process must be shared by all participants; and,
- Those performing the work and those inspecting the resulting increment must share a common definition of done.
- **Inspection :** Scrum users must frequently inspect Scrum artifacts and progress toward a Sprint Goal to detect undesirable variances.

Scrum Theory

Their inspection should not be so frequent that inspection gets in the way of the work.

Inspections are most beneficial when diligently performed by skilled inspectors at the point of work.

- **Adaptation:** If an inspector determines that one or more aspects of a process deviate outside acceptable limits, and that the resulting product will be unacceptable, the process or the material being processed must be adjusted.
- An adjustment must be made as soon as possible to minimize further deviation.

Scrum Terminology (Scrum Team)

Scrum Team: The Scrum team chiefly consist of three roles: The Scrum Master, Product Owner & the Development Team.

Anyone outside the core team doesn't have any direct influence over the Team.

Scrum Teams Attributes

Given below are the 2 attributes of the Scrum Team:

- Scrum Team is Self-Organizing
- Scrum Team is Cross-Functional

Scrum Terminology (Scrum Team)

Self-Organized Scrum Teams are self-reliant and self- sufficient in terms of accomplishing their work without the need for external help or guidance. The teams are competent enough to adopt the best of practices to achieve their Sprint Goals.

Cross-Functional Scrum Teams are the teams having all the necessary skills and proficiency within the team to accomplish their work. These teams do not rely on anyone outside the team for completing the work items. Thus, the Scrum Team is a very creative amalgamation of different skills that are required to complete the entire work item.

Scrum Terminology (Scrum Team)

Each team member may not necessarily have all the skills required to build the product but is competent in his/her area of expertise. Having said that, the team member need not be cross-functional but the team as a whole has to be.

The teams with high Self-Organization and Cross Functionality will result in high productivity and creativity.

Scrum Team(Product Owner)

- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed

Scrum Team(Scrum Master)

- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions

Scrum Team(Development Team)

Typically 5-9 people

- Cross-functional:

– Programmers, testers, user experience designers, etc.

- Members should be full-time

– May be exceptions (e.g., database administrator)

- Teams are self-organizing

– Ideally, no titles but rarely a possibility

- Membership should change only between sprints

Responsibilities of Product Owner

1. Manage Economics

The product owner is responsible for ensuring that good economic decisions are continuously being made at the release, sprint, and product backlog levels.

- Release-Level Economics
- Sprint-Level Economics
- Product Backlog Economics

2. Participate in Planning

The product owner is a key participant in the portfolio-, product-, release-, and sprint-planning activities.

Responsibilities of Product Owner

During portfolio planning, the product owner works with internal stakeholders (perhaps an approval committee or governance board) to position the product correctly in the portfolio backlog and to determine when to start and end product development.
During product planning, the product owner works with the stakeholders to envision the product.
During release planning, the product owner works with the stakeholders and the team to define the content of the next release.
During sprint planning, the product owner works with the development team to define a sprint goal.
He also provides valuable input that enables the development team to select a set of product backlog items that the team can realistically deliver by the end of the sprint.

Responsibilities of Product Owner

3. Groom the Product Backlog

The product owner oversees the grooming of the product backlog, which includes creating and refining, estimating, and prioritizing product backlog items.

The product owner doesn't personally perform all of the grooming work. For example, he might not write all of the product backlog items; others might contribute them.

The product owner also doesn't estimate the items but is available for questions and clarification during estimation.

The product owner is, however, ultimately responsible for making sure that the grooming activities take place in a way that promotes the smooth flow of delivered value.

Responsibilities of Product Owner

4. Define Acceptance Criteria and Verify That They Are Met

The product owner is responsible for defining the acceptance criteria for each product backlog item. These are the conditions under which the product owner would be satisfied that the functional and non-functional requirements have been met.

The product owner may also write acceptance tests corresponding to the acceptance criteria, or he could enlist the assistance of subject matter experts (SMEs) or development team members.

Responsibilities of Product Owner

5. Collaborate With Team

The product owner must closely collaborate with the development team on a frequent basis.

The product owner is an engaged, committed, everyday role.

Many organizations just starting to adopt Scrum fail to foster adequate product owner engagement with the development team, delaying essential feedback and substantially reducing the value of that feedback when it does occur.

Responsibilities of Product Owner

6. Collaborate with the Stakeholders

The product owner is the single voice of the entire stakeholder community, internal and external.

Internal stakeholders can include business systems owners, executive management, program management, marketing, and sales. **External stakeholders** can include customers, users, partners, regulatory bodies, and others.

The product owner must work closely with the entire stakeholder community to gather input and synthesize a coherent vision to guide product development.

Characteristics of Product Owner

1. Domain Skill

The product owner is a visionary who can synthesize a product vision and lead the team to achieve that vision.

To be effective at vision creation and execution, a product owner must have appropriate business and domain knowledge.

It’s difficult to be an effective product owner if you’re new to the product domain. How can you set priorities among competing features if you don’t know the subject matter?

Characteristics of Product Owner

2. People Skills

A product owner must also be the “voice of the customer,” which requires a good relationship with stakeholders.

Because there are frequently multiple stakeholders who could have conflicting needs, the product owner must also be a good negotiator and consensus builder.

A good communicator also exhibits the following qualities: is willing to speak up even if doing so goes against the status quo; confident in his ideas; knowledgeable of the subject matter; able to communicate in a simple, concise, and easily understood way; and credible.

Characteristics of Product Owner

A product owner is also a powerful motivator.

When the going gets tough, the product owner can remind people of why they are investing the effort and help people maintain an enthusiastic outlook by reinforcing the business proposition.

3. Decision making

The product owner must be empowered to make decisions.

The product owner must also be willing to make the hard decisions—usually by trading off constraints like scope, date, and budget. These decisions must be made in a timely manner and should not be reversed without good reason. In other words, the product owner should be a decisive decision maker.

Characteristics of Product Owner

4. Accountability

The product owner is held accountable for delivering good business results. This accountability doesn't absolve other Scrum team members from their own accountability to participate in generating a good return on investment.

After all, the product owner has many opportunities along the way to change the product backlog, readjust priorities, or even oversee canceling the development effort entirely.

The product owner must be committed and available to both the stakeholders and the rest of the Scrum team. Being a product owner is a full-time job; to try to do it part-time is a recipe for failure.

Responsibilities of Scrum Master

1. Coach

The Scrum Master is the agile coach for the Scrum team—both the development team and the product owner

Analogous to the coach of a sports team, the Scrum Master observes how the team is using Scrum and does anything possible to help it get to the next level of performance.

When problems arise that the team can and should be able to solve, the Scrum Master's attitude, like that of any good coach, is "I'm not here to solve your problems for you; instead, I'm here to help you solve your own problems." If the problem is an impediment that the team can't resolve, the Scrum Master takes ownership of getting it resolved.

Responsibilities of Scrum Master

The Scrum Master is often described as a servant leader of the Scrum team.

Even when acting as the team's coach, the Scrum Master is first and foremost a servant to the Scrum team, ensuring that its highest-priority needs are being met.

A servant leader would never ask, "So, what are you going to do for me today?" Instead, a servant leader asks, "So, what can I do today to help you and the team be more effective?"

Process Authority : The Scrum Master is the Scrum team's process authority. In this capacity, the Scrum Master is empowered to ensure that the Scrum team enacts and adheres to the values, principles, and practices along with the Scrum team's specific approaches.

Responsibilities of Scrum Master

The Scrum Master continuously helps the Scrum team improve the process, whenever possible, to maximize delivered business value.

Authority in this context is not the same type of authority that a functional manager or project manager would have. For example, the Scrum Master doesn't hire and fire and cannot dictate to the team what tasks it should do or how to do them.

The Scrum Master also is not responsible for making sure the work gets done. Instead, the Scrum Master helps the team define and adhere to its own process for making sure the work gets done.

Responsibilities of Scrum Master

Interference Shield

The Scrum Master protects the development team from outside interference so that it can remain focused on delivering business value every sprint.

Interference can come from any number of sources, from managers who want to redirect team members in the middle of a sprint, to issues originating from other teams.

No matter what the source of the interference, the Scrum Master acts as an interceptor (fielding inquiries, addressing management, and arbitrating disputes) so that the team can focus on delivering value.

Responsibilities of Scrum Master

Impediment Remover

The Scrum Master also takes responsibility for removing impediments that inhibit the team's productivity (when the team members themselves cannot reasonably remove them). For example, I observed a Scrum team that was consistently unable to meet its sprint goals. The impediment was unstable production servers that the team used during testing (as part of its definition of done).

The team itself had no control over these servers—that was the responsibility of the VP of Operations. Because the team itself could not remove the impediment, the Scrum Master took ownership of improving the server stability by working with the VP of Operations and others who could actually do something about the stability issue.

Responsibilities of Scrum Master

Change Agent

The Scrum Master must help change more than faulty servers and similar impediments. A good Scrum Master must help change minds as well.

Scrum can be very disruptive to the status quo; the change that is required to be successful with Scrum can be difficult.

The Scrum Master helps others understand the need for change, the impacts of Scrum outside of the Scrum team, and the broad-reaching benefits Scrum can help achieve.

The Scrum Master also ensures that effective change is occurring at all levels of the organization, enabling not only short-term success but, more importantly, the long-term benefits from using Scrum.

In large organizations, the Scrum Masters might band together to become a more effective force for change.

Characteristics of Scrum Master

Knowledgeable

To be an effective process coach, the Scrum Master must be very knowledgeable about Scrum.

The Scrum Master should also understand the technical issues the team needs to address and technologies the team will use to create solutions.

A Scrum Master doesn't need to have tech-lead- or dev-lead-level knowledge, but reasonable technical knowledge is an asset. The Scrum Master also doesn't need to be an expert in the business domain (the product owner does), but again, working knowledge of the business domain is very helpful.

Characteristics of Scrum Master

Questioning

Scrum Masters use their coaching skills in conjunction with their process, technical, and business knowledge to ask great questions.

They engage in intentional inquiry, asking the kinds of questions that make people stop and say, “Hmm. I never thought about that.

Now that you ask that question, it makes me think there might be another way to go.”

Great Scrum Masters almost never directly answer a question but instead reflexively answer with their own question—not an annoying question, or a question.

Characteristics of Scrum Master

Patient

Because Scrum Masters prefer not to give out answers, they need to be patient, giving teams time to arrive at appropriate answers on their own.

At times it is hard for me to be a Scrum Master because I see the issue the team is dealing with and I “know” the answer.

Well, at least I think I know the answer! It is arrogant for me (or any Scrum Master) to believe that I am smarter than the collective intelligence of the team.

So, at times I just have to bite my tongue and be patient, letting the team work out the solution, periodically asking probing questions to help guide things along.

Characteristics of Scrum Master

Collaborative

The Scrum Master must have excellent collaboration skills to work with the product owner, development team, and all the other parties, even those who might not be directly involved with Scrum.

Also, as the process coach, the Scrum Master is always looking for opportunities to help the Scrum team members achieve an enviable level of intra-team collaboration.

A Scrum Master can assist in this effort by personally exhibiting effective collaboration skills.

Characteristics of Scrum Master

Protective

The Scrum Master should be very protective of the team. The common analogy is that the Scrum Master acts like a sheepdog, guarding the flock from wolves that might try to attack.

In our context wolves could be organizational impediments or people with differing agendas.

The Scrum Master is skilled at ensuring the protection of the team within the greater context of making economically sound business decisions.

With acute sensitivity toward both team protection and business needs, the Scrum Master helps the Scrum team achieve a healthy balance.

Characteristics of Scrum Master

The Scrum Master also helps team members who begin to wander away from the flock.
When things get difficult, it is easy for people to fall back on familiar, non-agile approaches.
In this case it is the Scrum Master’s job to help shepherd straying team members, helping them overcome their difficulties by reinforcing how to use Scrum more effectively.

Characteristics of Scrum Master

Transparent

Finally, the Scrum Master is transparent in all forms of communication.

When working with team members, there is no room for hidden agendas; what you see and hear from the Scrum Master must be what you get. People expect nothing less of a servant leader.

The Scrum Master also promotes transparent communication outside of the Scrum team. Without transparent access to information it is difficult for the organization to inspect and adapt to achieve its desired business results from using Scrum.

Without transparent access to information it is difficult for the organization to inspect and adapt to achieve its desired business results from using Scrum.

Responsibilities of Development Team

Traditional software development approaches define various job types, such as architect, programmer, tester, database administrator, UI designer, and so on.

Scrum defines the role of development team, which is simply a cross-functional collection of these types of people.

In particular, the development team is one of the three roles on every Scrum team. The development team's members, collectively, have the skills required to deliver the business value requested by the product owner.

Responsibilities of Team Member

Perform Sprint Execution

During sprint execution, development team members perform the hands-on, creative work of designing, building, integrating, and testing product backlog items into increments of potentially shippable functionality.

To do this, they self-organize and collectively decide how to plan, manage, carry out, and communicate work.

The development team spends a majority of its time performing sprint execution.

Responsibilities of Team Member

Inspect and Adapt Each Day

Each development team member is expected to participate in each daily scrum, during which the team members collectively inspect progress toward the sprint goal and adapt the plan for the current day's work.

If some team members do not participate, the team can miss pieces of the big picture and may fail to achieve its sprint goal.

Groom the Product Backlog

Part of each sprint must be spent preparing for the next. A large part of that work focuses on product backlog grooming, which includes creating and refining, estimating, and prioritizing product backlog items. The development team should allocate up to 10% of its available capacity every sprint to assist the product owner with these activities.

Responsibilities of Team Member

Plan the Sprint

At the beginning of each sprint, the development team participates in sprint planning.

In collaboration with the product owner and with facilitation from the Scrum Master, the development team helps to establish the goal for the next sprint.

The team then determines which high-priority subset of product backlog items to build to achieve that goal. For a two-week sprint, sprint planning typically takes about half a day.

A four-week sprint might need up to a full day for sprint planning.

Notice that planning happens iteratively. Rather than focusing on a very large,uncertain, and overly detailed plan at the start of a development effort, the team makes a series of smaller, more certain, and more detailed plans just in time at the beginning of each sprint.

Responsibilities of Team Member

Inspect and Adapt the Product and Process

At the end of each sprint, the development team participates in the two inspect-and-adapt activities: sprint review and sprint retrospective.

The sprint review is where the development team, product owner, Scrum Master, stakeholders, sponsors, customers,
and interested members of other teams review the just-completed features of the current sprint and discuss how to best move forward.

The sprint retrospective is where the Scrum team inspects and adapts its Scrum process and technical practices to improve how it uses Scrum to deliver business value.

Characteristics of Team Member

Team members self-organize to determine the best way to accomplish the sprint goal.

There is no project manager or other manager who tells the team how to do its work (and a Scrum Master should never presume to).

Self-organization is a bottom-up, emergent property of the system—there is no external dominating force applying traditional top-down, command-and-control management.

A cross-functionally diverse team has members from different backgrounds.

Each team member brings a set of cognitive tools for problem solving; these tools can involve different interpretations (of the same data), different strategies (or heuristics) for solving problems, different mental models of how things work, and different preferences for both approaches and solutions.

This kind of diversity typically leads to better outcomes in terms of faster solutions, higher-quality deliverables, and greater innovation, all of which translate into greater economic value.

Characteristics of Team Member

T-Shaped Skills

T-shaped skills mean that a team member (say, Sue) has deep skills in her preferred functional area, discipline, or specialty. For example, Sue is a great user-experience (UX) designer—that is her specialty and where she prefers to do work.

Cross-functionally diverse

Cross-functional team members Better outcomes

Diverse perspectives

Strategies/heuristics

Different backgrounds

Interpretations

Mental models

Preferences

Better solutions

Faster solutions

Greater innovation

Characteristics of Team Member

Musketeer Attitude

Members of the development team (and the Scrum team as a whole!) need to have the same attitude as the Three Musketeers —**“All for one and one for all.”**

This **Musketeer attitude reinforces the point that the team members collectively own the responsibility** of getting the job done. They win as a team or they fail as a team.

In a well-functioning Scrum team, I would never expect anyone to say, “I got my part done. You didn’t get your part done. Therefore we failed.”

This attitude misses the point that team members are all in the same boat together.

Team members must appreciate that they must work together to meet their commitments, because if they fail, it’s going to be everybody’s problem in the end.

Characteristics of Team Member

Having team members with a Musketeer attitude is critical to achieving shared success.

Having team members with T-shaped skills encourages this attitude and makes it practical because people are capable of working on more than one type of task.

On these teams I don’t expect to hear a person who is capable of doing the work say, “That’s not my job.”

Characteristics of Team Member

High-Bandwidth Communications

Development team members need to communicate with one another, as well as with the product owner and Scrum Master, in a high-bandwidth manner, where valuable information is exchanged quickly and efficiently with minimal overhead.

High-bandwidth communications increase both the frequency and quality of information sharing. As a result, the Scrum team has more frequent opportunities to inspect and adapt, leading to better and faster decision making.

Because the economic value of information is time-sensitive, accelerating the rate of information sharing allows the team to maximize its value.

By quickly exploiting emergent opportunities and recognizing wasteful situations, the team can avoid expending more resources by going down the wrong path.

Characteristics of Team Member

We should also reduce time spent on ceremonies where team members perform a process that adds little or no value.
For example, if team members have to go through three levels of indirection before they can speak with an actual customer or user, the ceremony of **“talking to a customer”** is probably a serious impediment to **high-bandwidth communications.**
Having to create low- or no-value documents or requiring lengthy and potentially unnecessary approval and sign-off procedures reduces bandwidth.
We need to identify and eliminate these impediments to improve overall team communication performance.

Characteristics of Team Member

Right-Sized

Scrum favors small teams. The general rule is that having five to nine people on the team is best. There is published research that backs up the claim that small teams tend to be the most efficient (Putnam 1996; Putnam and Myers 1998).

My experience over the past 25 years is that teams of five to seven are the sweet spot for rapidly delivering business value.

Mike Cohn lists a handful of reasons to keep teams small, which include the following (Cohn 2009):

There is less social loafing—people exerting less effort because they believe that others will pick up the slack.

Constructive interaction is more likely to occur on a small team.

Less time is spent coordinating efforts.

No one can fade into the background.

Small teams are more satisfying to their members.

Harmful overspecialization is less likely to occur.

Characteristics of Team Member

Focused and Committed

Team members need to be focused and committed to the team's goal. Focused means that each team member is engaged, concentrating on and devoting her attention to the team's goal. Committed means that during both good times and bad, each team member is dedicated to meeting the team's collective goal.

If a person is working on only one product, it is far easier for that person to be focused and committed.

When asked to work on multiple concurrent product development efforts, a person must split her time across those products, reducing her focus and commitment on all products.

Ask any person who works on multiple products about her focus and commitment and you will likely be told something like "I have so much to do that I just try to do the best job that I can on each product and then hop to the next product.

Characteristics of Team Member

Based on this data, working on three or more concurrent projects is a bad economic choice because more time is spent on coordinating, remembering, and tracking down information and less time is spent doing value-adding work.

So, how many projects/products (and probably different teams) should a person be on simultaneously?

Probably not more than two. I have a strong preference for one, because in today's highly connected, information-rich world with email, instant messaging, Twitter, Facebook, and other technologies, being a good corporate citizen is probably the equivalent of being on one project!

Characteristics of Team Member

Working at a Sustainable Pace

One of Scrum's guiding principles is that team members must work at a sustainable pace. In doing so, they deliver world-class products and maintain a healthy and fun environment.

Using sequential development, we defer important activities like integration and testing until near the end, when there typically is a crushing workload of issues to deal with as we approach the delivery date. The result is an extensive increase in intensity in the latter phases.

Characteristics of Team Member

Long-Lived

Effective use of Scrum requires teams, not groups. A **team is made up of a diverse,**

cross-functional collection of collaborating people who are aligned to a common vision and work together to achieve that vision.

A **group is a collection of people with** a common label. Other than sharing the group name, group members don't share much else and won't effectively fulfill the responsibilities I described for the development team role.

As a rule, teams should be long-lived. I keep my teams together as long as it is economically sensible to do so. And the economics are very favorable for long-lived teams. Research by Katz has shown that long-lived teams are more productive than newly formed groups (Katz 1982).

Characteristics of Team Member

Furthermore, research by Staats demonstrates that team familiarity (team members' prior shared work experience) can positively impact the efficiency and quality of team output (Staats 2011).

Improved productivity, efficiency, and quality lead to improved business results. Far too often I see organizations failing to appreciate the asset value of teams.

Most organizations have developed skills and processes for moving people around to dynamically form "teams" (really groups). In my opinion such practices miss a critical aspect of Scrum—the value is in the team.

The *currency of agile is the team. In* fact, one of the core values of the Agile Manifesto is "Individuals and Interactions."

In other words, the team is the valuable asset.

Sprint

Scrum organizes work in iterations or cycles of up to a calendar month called sprints.

key characteristics of sprints: They are time-boxed, have a short and consistent duration, have a goal that shouldn't be altered once started, and must reach the end state specified by the team's definition of done.

Sprints should be consistent in length, though exceptions are permitted under certain circumstances.

As a rule, no goal-altering changes in scope or personnel are permitted during a sprint.

Finally, during each sprint, a potentially shippable product increment is completed in conformance with the Scrum team's agreed-upon definition of done.

Time Boxed

Sprints are rooted in the concept of **time-boxing, a time-management technique that** helps organize the performance of work and manage scope.

Each sprint takes place in a time frame with specific start and end dates, called a time-box. Inside this time-box, the team is expected to work at a sustainable pace to complete a chosen set of work that aligns with a sprint goal.

Epics

An epic is a large user story that cannot be delivered as defined within a single iteration or is large enough that it can be split into smaller user stories.

There is no standard form to represent epics. Some teams use the familiar user story formats (As A, I want, So That or In Order To, As A, I want) while other teams represent the epics with a short phrase.

Product backlog items included the following epic-level user stories:

As a Certified Scrum Trainer I want to be able to post my public Scrum class on the Scrum Alliance website so that the community will know the details surrounding where and when I am offering the class.

As a prospective student I want to be able to see details of all publicly available Scrum classes so that I can find one that meets my criteria for attendance.

EPIC Example

Epic 01: Provide ordering and priority options to user to manage backlog easily

User Story 01: As a release manager, I want to have my releases mapped to different sprints. I also want to see the priority of each item on same board.

User Story 02: As a system admin, I should be able to provide rights who can prioritize product backlog

User Story 03: As a user I should be able to re-order my backlog using numbered items, star priority or simple drag n drop

Product Backlog

The product owner, with input from the rest of the Scrum team and stakeholders, is ultimately responsible for determining and managing the sequence of this work and communicating it in the form of a prioritized (or ordered) list known as the **product backlog**. On new-product development the product backlog items initially are features required to meet the product owner's vision. For ongoing product development, the product backlog might also contain new features, changes to existing features, defects needing repair, technical improvements, and so on.

The product owner collaborates with internal and external stakeholders to gather and define the product backlog items.

He then ensures that product backlog items Scrum Activities and Artifacts are placed in the correct sequence (using factors such as value, cost, knowledge, and risk) so that the high-value items appear at the top of the product backlog and the lower-value items appear toward the bottom.

Product Backlog

The product backlog is a constantly evolving artifact.
Items can be added, deleted, and revised by the product owner as business conditions change, or as the Scrum team's understanding of the product grows (through feedback on the software produced during each sprint).

Sprint Backlog

To acquire confidence in what it can get done, many development teams break down each targeted feature into a set of tasks. The collection of these tasks, along with their associated product backlog items, forms a second backlog called the **sprint backlog.**

Sprint backlog is a list of the tasks and requirements to be completed within the sprint.

The sprint backlog includes:

- The list of user stories within the sprint in order of priority.
- The relative effort estimate for each user story.
- The tasks necessary to develop each user story.
- The effort, in hours, to complete each task.
- A burn down chart that shows the status of the work the development team has completed.

Story Points

Story point is used for effort estimation for a user story, it’s a number without any unit, and sp value represses the comparative complexity from one story to another.
Classic example is reading a book or travelling from point A to C. Usually this could be value such as t-shirt size or number of cookies.

Velocity and Burn down Chart

Velocity is a measure in the Agile methodology. Simply put, velocity measures the rate at which the team gets work done in a single iteration. It deals with what actually was achieved as opposed to what was planned in that iteration.

A burn down chart is a graphical representation of work left to do versus time. The outstanding work is often on the vertical axis, with time along the horizontal.

It is useful for predicting when all of the work will be completed.

Burn up Chart

A burn-up chart tracks progress towards a project's completion. In the simplest form of burn up chart, there are two lines on the chart:

A total work line (the project scope line)

A work completed line