#### #1) Scrum

Scrum is a frame work for developing complex products and systems.it is grounded in empirical process and control theory. Scrum employs on iterative and incremental approach to optimize predictability and control risk.

Scrum can easily be considered to be the most popular agile framework. The term ‘scrum’ is much considered synonymously to ‘agile’ by most practitioners. But that is a misconception. Scrum is just one of the frameworks by which you can implement agile.

The word scrum comes from the sports rugby. Where the players huddle together in an interlocked position pushing against the opponents. Each player has a defined role in their position and can play both offensive and defensive as per the demand of the situation.

Similarly, the scrum in IT believes in empowered self-managed development teams with three specific and clearly defined roles. These roles include – **Product Owner (PO), Scrum Master (SM) and the development team consisting of the programmers and testers**. They work together in iterative time boxed durations called sprints.

The first step is the creation of the product backlog by the PO. It’s a to-do list of stuff to be done by the scrum team. Then the scrum team selects the top priority items and tries to finish them within the time box called a sprint.

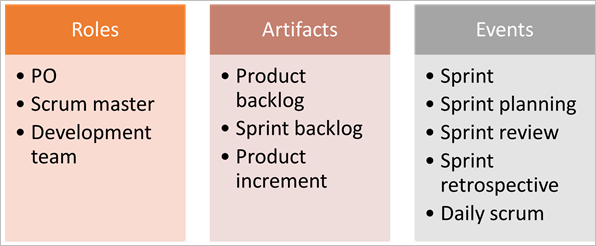
An easier way to remember all of this is to memorize the 3-3-5 framework. It means that a scrum project has 3 roles, 3 artifacts, and 5 events.

**These are** –

**Roles:**  PO, Scrum master, and development team.

**Artifacts:**  Product Backlog, Sprint Backlog and Product increment.

**Events:**  Sprint, Sprint planning, Daily Scrum, Sprint review and Sprint retrospective.



We will get to know more in detail about each of these in our subsequent tutorials.

#### #2) Kanban

Kanban is a Japanese term which means a card. These cards contain details of the work to be done on the software. The purpose is visualization. Every team member is aware of the work to be done through these visual aids.

Teams use these Kanban cards for continuous delivery. Just like Scrum, Kanban is also for helping the teams work effectively and promotes self-managed and collaborative teams.

But there are differences between these two as well – like during a scrum sprint, the items being worked upon by a team are fixed and we cannot add items to the sprint whereas, in Kanban, we can add items if there is available capacity. This is particularly useful when the requirements change frequently.

Similarly, another difference is that while the scrum has defined roles of a PO, scrum master, and development teams, there are no such pre-defined roles in Kanban.

Another difference is that while the scrum suggests prioritization of product backlogs, Kanban has no such requirement and it is totally optional. Thus Kanban requires less organization and avoids non-value adding activities and is suitable for the processes which require responsiveness towards changes.

### Important SCRUM Terminologies

**1) Scrum Team**

Scrum team is a team comprising of 7 with + or – two members. These members are a mixture of competencies and comprise of developers, testers, database people, support people etc. along with the product owner and a scrum master.

All these members work together in close collaboration for a recursive and definite interval, to develop and implement the said features. SCRUM team sitting arrangement plays a very important role in their interaction, they never sit in cubicles or cabins, but a huge table.



**2) Sprint**

Sprint is a predefined interval or time frame in which the work has to be completed and make it ready for review or ready for production deployment. This time box usually lies between 2 weeks to 1 month.

In our day to day life when we say that we follow 1-month Sprint cycle, it simply means that we work for one month on the tasks and make it ready for review by the end of that month.

**3) Product Owner**

The product owner is the key stakeholder or the lead user of the application to be developed. The product owner is the person who represents the customer side. He/she has the final authority and should always be available for the team.

He/she should be reachable when anyone has any doubts that need clarification. It is important for the product owner to understand and not to assign any new requirement in the middle of the sprint or when the sprint has already started.

**4) Scrum Master**

Scrum Master is the facilitator of the scrum team. He/she makes sure that the scrum team is productive and progressive. In case of any impediments, scrum master follows up and resolves them for the team. SCRUM Master is the mediator between the PO and the team.

He/she keeps the PO informed about the progress of the Sprint. If there are any roadblocks or concerns for the team, discusses with the PO and gets them resolved. Like the team's Daily Standup, a standup of the SCRUM Master with the PO happens every day.

**5) Business Analyst (BA)**

A Business Analyst plays a very important role in SCRUM. This person is responsible for getting the requirement finalized and drafted in the requirement docs (based on which the user stories are created).

If there are any ambiguities in the User Stories / Acceptance criteria, he/she is the one who is approached by the technical (SCRUM) team and he then takes it up to the PO or else if possible resolves on his own. In large scale projects there may be more than 1 BA but in small-scale projects, the SCRUM Master may be acting as the BA as well.

It is always a good practice to have a BA when the project kick starts.

**6) User Story**

User stories are nothing but the requirements or feature which has to be implemented.

In the scrum, we don’t have those huge requirements documents, rather the requirements are defined in a single paragraph, typically having the format as:

As a <User / type of user>  
I want to <Some achievable goal/target>  
To achieve <some result or reason for doing the thing>

**For Example:**

As an Admin I want to have a password lock in case the user enters an incorrect password for 3 consecutive times to restrict unauthorized access.

There are some characteristics of user stories which should be adhered. The user stories should be short, realistic, could be estimated, complete, negotiable and testable. A user story is never altered or changed in the middle of the Sprint.

It is the responsibility of the SCRUM Master and the BA (if applicable) to make sure that the PO has drafted the User Stories correctly with a proper set of the Acceptance Criteria”. If any changes which will impact the sprint release are to be made, then such stories are pulled out of the sprint or they are done as per the hours available.

Every user story has an acceptance criterion which should be well defined and understood by the team.

Acceptance criteria details down the user story that provides the supporting documents. It helps to further refine the user story. Anybody from the team can write down the acceptance criteria. Testing team bases their test cases/conditions on these acceptance criteria.

**7) Epics**

Epics are equivocal user stories or we can say that these are the user stories which are not defined and are kept for future sprints.

Just try to relate it with life, imagine you are going for a vacation. As you are going next week, you have everything in place like your hotel bookings, sightseeing, travelers check etc. But what about your vacation plan for next year? You only have a vague idea that you may go to XYZ place, but you have no detailed plan.

An Epic is just like you next year’s vacation plan, where you just know that you may want to go, but where, when, with whom, all these details you have no idea at this point of time.

In a similar way, there are features which are required to be implemented in future whose details are not yet known. Mostly a feature begins with an Epic and then is broken down to stories which could be implemented.

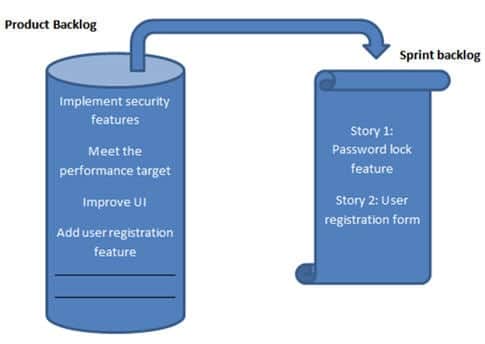
**8) Product Backlog**

The product backlog is a kind of bucket or source where all the user stories are kept. This is maintained by the Product Owner. Product backlog can be imagined as a wish list of the product owner who prioritizes it as per the business needs.

During the planning meeting (see next section), one user story is taken from the product backlog, then the team does the brainstorming, understands it and refines it and collectively decides which user stories to take, with the intervention of the product owner.

**9) Sprint Backlog**

Based on the priority, user stories are taken from the Product Backlog as one at a time. The Scrum team brainstorms on it determine the feasibility and decide on the stories to work on a particular sprint. The collective list of all the user stories which the scrum team works on a particular sprint is known as Sprint backlog.



**10) Story Points**

Story points are a quantitative indication of the complexity of a user story. Based on the story point, estimation and efforts for a story are determined.

A story point is relative and not absolute. In order to make sure that our estimate and efforts are correct, it’s important to check that the user stories are not big. The more precise and smaller is the user story, the more accurate will be the estimation.

Each and every user story is assigned to a story point based on the Fibonacci series (1, 2, 3, 5, 8, 13&21). Higher is the number, the complex is the story.

**To be precise**

* If you give 1 / 2 / 3 story point it means that the story is small and of low complexity.
* If you give points as 5 / 8, it is a medium complex and
* 13 and 21 are highly complex.

Here complexity consists of both development plus testing effort.

To decide a story point, brainstorming happens within the scrum team and the team collectively decides a story point.

It may happen that the development team gives a story point of 3 to a particular story, because for them it may be 3 lines of code change, but the testing team gives 8 story point because they feel that this code change will affect larger modules so the testing effort would be larger. Whatever story point you are giving, you have to justify it.

So in this situation, brainstorming happens and the team collectively agrees to one story point.

**Whenever you are deciding on a story point, keep the below factors in mind:**

* The dependency of the story with other application/module.
* The skill-set of the resource.
* The complexity of the story.
* Historical learning.
* Acceptance criteria of the user story.

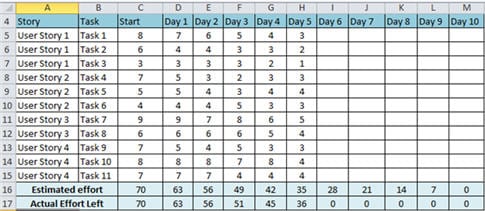
If you are not aware of a particular story, don’t size it.

Whenever a story is = or > 8 points, it is broken down into 2 or more stories.

**11) Burn down chart**

Burn down chart is a graph which shows the estimated v/s actual effort of the scrum tasks.

It is a tracking mechanism by which for a particular sprint the day to day tasks are tracked to check whether the stories are progressing towards the completion of the committed story points or not.



I have assumed:

* 2 weeks Sprint ( 10 days)
* 2 resources actual working on the sprint.

**“Story”**-> This column shows the user stories taken for the sprint.

**“Task”** -> This column shows the list of the task associated with the user story.

**“Effort”** -> This column shows the effort. Now, this measure is the total effort to complete the task. It does not depict the effort put in by any specific individual.

**“Day 1 – Day 10”** -> this column(s) shows the hours which are left to complete the story. Please see that the hour is NOT the hour which is already done BUT the hours which are still left.

**“Estimated Effort”** -> is the total of the effort. For the “Start” it is simply the sum of the entire individual task: SUM (C5: C15)

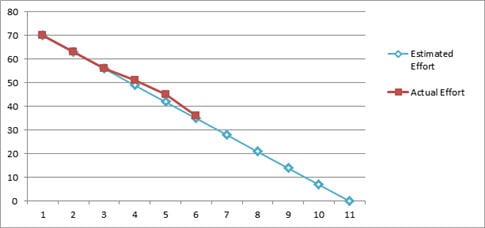
A total number of efforts that has to be completed in 1 day is 70 / 10 = 7. So at the end of day 1, the effort should reduce to 70 – 7 = 63. In a similar way, it is calculated for all the days till day 10, when the estimated effort should be 0 (Row 16)

**“Actual Effort Left”** -> As the name suggests, is the effort actually left to complete the story. It may also happen that the actual efforts increases or decreases than the estimated one.

You can use the in built functions and Chart in Excel to create this burndown chart.

**Burn down Chart steps would be:**

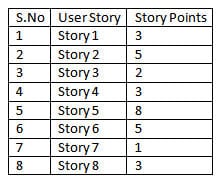
1. Enter all the stories (Column A5 – A15).
2. Enter all the Tasks (Column B5 – B15).
3. Enter the Days (Day 1 – Day 10).
4. Enter the starting efforts (Sum the tasks C5 – C15).
5. Apply the formula to calculate the “Estimated Efforts” for each day (Day 1 to Day 10). Enter the formula at D15 (C16-$C$16/10) and drag it for all the days.
6. For each day, enter the actual efforts. Enter the formula at D17 (SUM (D5:D15)) for summing up the actual efforts left, and drag it for all the other days.
7. Select it and create the chart as follows:



**12) Velocity**

The total number of story point which a scrum team archives in a sprint, is called Velocity. The Scrum team is judged or referenced by its velocity. Having said that, it should be kept in mind that the objective here is NOT achieving the maximum story points, but to have a quality deliverable, respecting the scrum team’s comfort level.

**For Example**: For a particular sprint: the total number of user stories are 8 having story points as shown below.



So here the velocity will be the sum of the story points = 30

**Definition of Done:**

A Sprint is marked as Done when all the stories are completed, all dev, research, QA tasks are marked ‘Completed', all bugs are fixed-closed else the ones that can be done later (like not completely related or are less important) are pulled out and added in the backlog, the code review and unit testing is completed, the estimated hours have met the actual hours put up in the tasks and most importantly a successful demo has been given to the PO and the stakeholders.



### Activities Done in SCRUM Methodology

#### #1) Planning Meeting

A planning meeting is the starting point of Sprint. It is the meeting where the entire scrum team gathers, the SCRUM Master selects a user story based on the priority from the product backlog and the team brainstorms on it.

Based on the discussion, the scrum team decides the complexity of the story and sizes it as per the Fibonacci series. The team identifies the tasks along with the efforts (in hours) which would be done to complete the implementation of the user story.

Many a time, the planning meeting is preceded by a “Pre-Planning meeting”. It’s just like a homework which the scrum team does before they sit for the formal planning meet. The team tries to write down the dependencies or other factors which they would like to discuss in the planning meeting.

#### #2) Execution of Sprint Tasks

As the name suggests, these are the actual work done by the scrum team to accomplish their task and take the user story into the “Done” state.

#### #3) Daily Standup

During the sprint cycle, every day the scrum team meets for, not more than 15 minutes (could be a stand-up call, recommended to have during the beginning of the day) and state 3 points:

1. What did the team member do yesterday?
2. What did the team member plan to do today?
3. Any impediments (roadblocks)?

It is the Scrum master who facilitates this meeting. In case, any team member is facing any kind of difficulties, the scrum master follows up to get it resolved. In Stand ups, the board is also reviewed and in itself shows the progress of the team.

#### #4) Review Meeting

At the end of every sprint cycle, the SCRUM team meets again and demonstrates the implemented user stories to the product owner. The product owner may cross verify the stories as per its acceptance criteria. It’s again the responsibility of the Scrum master to preside over this meeting.

Also in the SCRUM tool, the Sprint is closed and the tasks are marked done.

#### #5) Retrospective Meeting

The retrospective meeting happens after the review meeting.

The SCRUM team meets, discusses & document the following points:

* What went well during the Sprint (Best practices)?
* What did not go well in the Sprint?
* Lessons learned
* Action Items.

The Scrum team should continue to follow the best practice, ignore the “not best practices” and implement the lessons learned during the consequent sprints. The retrospective meeting helps to implement the continuous improvement of the SCRUM process.

### How the Process is done? An Example!

Having read about the technical jargons of SCRUM. let me try to demonstrate the whole process with an example.

**Example:**

**Step #1**: Let’s have a SCRUM team of 9 people comprising of 1 product owner, 1 Scrum master, 2 testers, 4 developers and 1 DBA.

**Step #2**: The Sprint is decided to follow a 4 weeks cycle. So we have 1-month Sprint starting 5th June to 4th of July.

**Step #3**: The Product owner has the prioritized list of user stories in the product backlog.

**Step #4:** The team decides to meet on 4th June for the “Pre Planning” meeting.

* The product owner takes 1 story from the product backlog, describes it and leaves it to the team to brainstorm on it.
* The entire team discusses and communicates directly to the product owner to have clearly understood the user story.
* In a similar way, various other user stories are taken. If possible, the team can go ahead and size the stories as well.

After all the discussion, Individual team members go back to their workstations and

* Identify their individual tasks for each story.
* Calculate the exact number of hours on which they will be working. Let's check how the member concludes these hours.

Total number of working hours = 9

Minus 1 hour for a break, minus 1 hour for meetings, minus 1 hour for emails, discussions, troubleshooting etc.

So the actual working hours = 6.

A total number of working days during the Sprint = 21 days.

Total number of hours available = 21\*6 = 126.

The member is on leave for 2 days = 12 hours (This varies for each member, some may take leave and some may not.)

Number of actual hours = 126 – 12 = 114 hours.

This means that the member will actually be available for 114 hours for this sprint. So he will break down his individual sprint task in such a way that a total of 114 hours is reached.

**Step #5**: On the 5th of June the entire Scrum team meets for the “Planning Meeting”.

* The final verdict of the user story from the product backlog is done and the story is moved to the Sprint Backlog.
* For each story, each team member declares their identified tasks, if required they can have a discussion on those tasks, can size or resize it (remember the Fibonacci series!!).
* The Scrum master or the team enter their individual tasks along with their hours for each story in a tool.
* After all the stories are completed, Scrum master notes the initial Velocity and formally starts the Sprint.

**Step #6**: Once the Sprint has started, based on the tasks assigned, each team member starts working on those tasks.

**Step #7**: The team meets daily for 15 minutes and discusses 3 things:

* What did they do yesterday?
* What they plan to do today?
* Any impediments (roadblocks)?

**Step #8**: The scrum master tracks the progress on a daily basis with the help of “Burn down chart”.

**Step #9**: In case of any impediments, the Scrum master follows up to resolve those.

**Step #10**: On 4th July, the team meets again for the review meeting. A member demonstrates the implemented user story to the product owner.

**Step #11**: On 5th July, the Team meets again for the Retrospective, where they discuss

* What went well?
* What did not go well?
* Action Items.

**Step #12**: On 6th July, the Team again meets for pre-planning meeting for the next sprint and the cycle continues.

