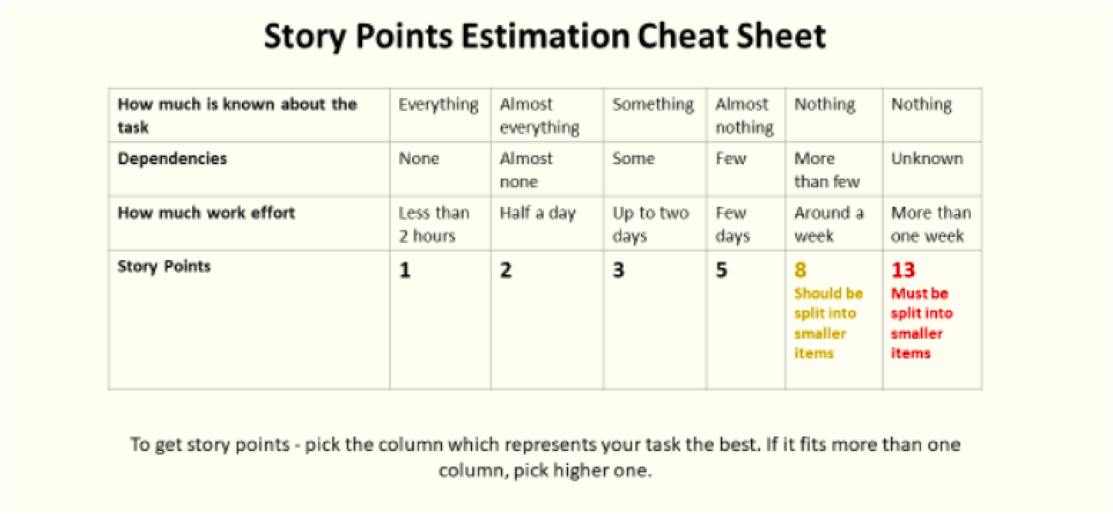
Story Point:

- A story point is a metric used in agile project management and development to estimate the difficulty of implementing a given user story.
- Story point estimation is based on three main components:
    - Risk
    - Complexity
    - Repetition
- We will use Fibonacci series to give story point on user stories
- Please refer below sheet for story point calculation:



Story Points Estimation Cheat Sheet

How much is known about the task	Everything	Almost everything	Something	Almost nothing	Nothing	Nothing
Dependencies	None	Almost none	Some	Few	More than few	Unknown
How much work effort	Less than 2 hours	Half a day	Up to two days	Few days	Around a week	More than one week
Story Points	**1**	**2**	**3**	**5**	**8** **Should be split into smaller items**	**13** **Must be split into smaller items**

To get story points - pick the column which represents your task the best. If it fits more than one column, pick higher one.

- We will split user stories if the story point is high
- Instead of giving the estimation in hours/day we will give story points to the user story in agile.

Velocity:

● The amount of work the scrum team has completed by the end of the sprint is called
velocity.

● To calculate the average of velocity we need use below formula:
Completed Story Points from sprint 1 + Completed Story Points from sprint 2+.... %
Sprint total= Avg Velocity

● Example 10+10 % 2=10

Sprint Velocity:

● By looking at the amount of work your team completed in previous sprints, you should be able to estimate how much work they can do in future sprints. In Agile development, this estimate is known as sprint velocity.

● With this knowledge in-hand, you can plan projects and predict how much work can be completed in the next sprint. You should also have a better idea of the resources you will need and the effort it will take to complete the project. In addition, your sprint velocity estimate gives senior management and other stakeholders a better idea of when to expect delivery of the product.

Smoke Testing

● **Smoke Testing** is a software testing process that determines whether the
deployed software build is stable or not.

● Smoke testing is a confirmation for QA team to proceed with further software testing.

● It consists of a minimal set of tests run on each build to test software functionalities. Smoke testing is also known as “Build Verification Testing” or “Confidence Testing.”
● In simple terms, we are verifying whether the important features are working and there are no showstoppers in the build that is under testing.
● Smoke Testing is done whenever the new functionalities of software are developed and integrated with existing build that is deployed in QA/staging environment. It ensures that all critical functionalities are working correctly or not.
● After releasing the build to QA environment, Smoke Testing is performed by QA engineers/QA lead. Whenever there is a new build, QA team determines the major functionality in the application to perform smoke testing. QA team checks for showstoppers in the application that is under testing.

Example:

The developer develops the application and handed over to the testing team, and the testing team will start the functional testing Suppose we assume that four days we are given to the functional testing. On the first day, we check one module, and on the second day, we will go for another module. And on the fourth day, we find a critical bug when it is given it to the developer; he/she says it will take another two days to fix it. Then we have to postpone the release date for these extra two days.

To overcome this problem, we perform smoke testing, let us see how it works, in the above situation, instead of the testing module by module thoroughly and come up with critical bug at the end, it is better to do smoke testing before we go for functional, integration and system testing that is, in each module we have to test for essential or critical features, and then proceed for further testing