**ASSIGNMENT II**

**Name: U.R.T.D. Bandara**

**Index Number:15APC2323**

**User Stories, Velocity, and Generally Accepted Scrum Practices.**

* Stakeholders and user or customers hate unexpectedness.
* Also, your team build the valuable software for their users and stakeholders, if the working software they deliver at the end of sprint does not resemble the promise they made at the starting, the user or customer disappointed.
* Avoiding surprise and pleasant surprise can be almost harmless unpleasant, if they have the prospect that your team will always be on top it usually boils down to two things.
* the team needs to do a good job of setting expectations at the beginning of the sprint.
* as the sprint progresses, they have to keep everyone up to date on all of the changes they discover during their Daily Scrum meetings.
* The Daily Scrum is the team’s time to plan the next day’s work, not answer questions from people who aren’t on the team. Not all stakeholders will Daily Scrum may participant.

**Make your software useful**

Collective commitment means genuinely trying to make your software more useful.in this we need to understand what is that users are doing. Software means building software that really helps the users get their work done. Customers collaboration is way of building a most effective working software.

Prior to the rapid change, software that was not habitually useful to the team was developed. After a survey and study, it was discovered that the project teams did not use much of the software they had built. They have built a sophisticated, over-designed tool to solve a simple problem to get them in trouble with their proof. This is a common thing for any team to build software that is not worth the software. Also, developers want to help, and they think their software is cool and valuable. Half of these unused features. Some of the unused features are tossing it on the wall, which is why this user thinks he has limited time to work with the development team. Ag profitable agile teams rarely get this experience.

**User Stories Help Build Features Your Users Will Use**

* user story is a deceptively simple tool.
* It’s a quick and simple
* description of a specific way that a user will use the software.
* User story has a title.
* it is a lot of information packed into the user story.
* User stories also give teams an easy way to manage their backlog.
* Most teams will then break the stories down into tasks and start to estimate how long
* those tasks will take.

**Conditions of Satisfaction (Acceptance criteria)**

Conditions of satisfaction are an effective tool for helping developers to know what the software will look like when it’s complete, and to gauge how close they are to done**.**

Typically, the conditions of satisfaction can fit on the back of the same 3×5 index card as the user story. The Product Owner usually has a hand in writing the conditions of satisfaction or at least reviews them when they’re written. The conditions are valuable to developers because it helps them avoid declaring victory too early. Conditions of satisfaction help this by giving the team a concrete definition of “Done.”

**Story Points and Velocity**

Story points are a way to understand how much effort you’ll need to build a specific user story by assigning a number of points to it.

A sprint planning session using story points might go like this:

1. Start with the most valuable user stories from the product backlog.
2. Take a story in that list—ideally the smallest one, because that makes a good baseline for comparison—find a similarly sized story from a previous sprint, and assign it the same number of points.
3. Discuss with the team whether that estimate is accurate—discovering additional problems, work, or technical challenges increases the estimate; simplifying factors, reusing existing code, or scaling back what needs to be built decreases the estimate.
4. Keep going through the stories until you’ve accumulated enough points to fill the sprint.

**Why story points work**

We can identify the key points of why the story point are so effective.

* They’re simple.
* They’re not magic.
* The team is in control of them.
* They get your team talking about estimates.
* Developers aren’t scared of them.
* They help the team discover exactly what a story means.
* They help everyone on the team become genuinely committed.

**Burndown Charts**

A burndown chart is a way for anyone to see, at a glance, how the sprint is actually progressing when compared with the team’s past velocity. The Burndown Chart is a way for anyone to see what a sprint really is.

The team is making progress compared to past momentum. Many groups Burndown likes to draw charts and keep the graph on the same wall as the task board.   
It is especially satisfying for the team if each developer Once he or she completes a user story, the Burndown chart has to be updated the card is "done".

Following these steps, we can identify basic things the burndown chart.

1. X-Axis (The project/iteration timeline.)
2. Y-Axis (Starting points.)
3. Project End Point.
4. Number of workers and Efficiency Factor.
5. Ideal Work Remaining line.
6. Actual Work Remaining line.

**Planning and Running a Sprint Using Stories, Points, Tasks, and a Task Board**

One of the most common ways to design real work for Scrum teams the team is a collection of cards for individual development purposes. These tasks can be anything that the team actually does: write code, create design and architecture, build tests, install operating systems, design and build databases, deploy software to production servers, run usability tests, and do all of those other things that teams actually do every day to build and release software.

Scrum teams to plan out the work for the team is to add cards for individual development tasks.

* Write code
* Build tests
* Create design and architecture
* Install operation systems
* Run usability test

**Generally Accepted Scrum Practices.**

There are so many practices that teams use to improve how they use scrum. Many Teams find that Daily Scrum meeting are more effective when held a stand-up meeting where everyone stands until meeting is completed.