Week 5 Tutorial

Goals

1. Create an initial release plan for your project, based on the priority of stories and their estimated size.
2. Create an initial sprint plan for the first development sprint of the project.

Assumptions

1. Your stories describe a complete system.
2. You have prioritised and estimated all of your stories.
3. You attended the week 4 and 5 lectures and are familiar with the process of planning a release and a sprint.

Activities

Activity 1 (40 min.)	As a team, create an initial release plan for your system. Each release should represent a coherent set of functionality that provides meaningful value to your stakeholders. Create columns on a wall, table, or online tool for releases 1, 2, … n. Place user story cards in each column to represent which stories are to be delivered in each release. Do this for all of your user stories. Make a note of each release's theme, for documentation in your release plan. At the end of this tutorial, you will create a team specific estimate of your velocity, based on calculations you need to perform at the end of the sprint planning activity. Use this velocity estimate to set planned delivery dates for each release. The starting date of the first sprint will be April 4. Assume that sprints will be two weeks long. Ignore public holidays in your schedule. Each release should have a theme, which should determine the stories that are delivered in that release. Within a release, stories should be grouped together into sets of similar functionality, which are called features. These features can be related to an epic from which a number of stories were generated. It is possible that there are other stories that conceptually belong to the same feature, but which are not included in a release. This can be thought of as planning to deliver the minimum expectations for a feature early, and then planning to add "nice to have" stories to the feature later. The important point of this activity is to ensure that you deliver value from the project as soon as possible. Releases are for general deployment. The set of features in a release need to deliver meaningful value to your stakeholders and provide a stable system for general use. A release may be made up of a few sprints to be able to deliver the goals for a release. If you have time, start to populate the release plan in the provided template for the assignment.
5 min.	**Break**
Activity 2 (40 min.)	As a team, create an initial sprint plan for the first development sprint of your project. Sprints will have a duration of two week. *Initially*, select twenty story points worth of stories from the first release. These should be stories that have the highest priority from the first release. Value the stories deliver, and risk management, should be considered in determining the stories' priorities. For each of the selected stories, break the story down into a set of ***detailed*** tasks that need to be completed to deliver the completed story. These tasks need to include ***all*** work required to take the story and deliver working software that can be given to the client. Try to think of all necessary design, implementation and testing activities. A

	task should be work a single person could complete in half a day or less. A task should represent one item of work (e.g. design a database table to store an order); not a set of related activities (e.g. design an order table, implement SQL queries for orders, and test the queries). For each task, estimate the number of hours required to complete the task. These estimates need to be realistic, based on the skills and experience you have within your team. If you have difficulty agreeing on the estimated number of hours for a task, try using planning poker, where the card numbers represent hours. Split any tasks larger than eight hours into multiple tasks.
Activity 3 (15 min.)	After estimating the hours for all the tasks, in all the selected stories, compare the total hours between stories. Check to see if all stories that were estimated to be the same story point size will take a similar number of hours to complete. If there is a significant difference in the total hours for some stories, revise their story point estimates (up or down) as appropriate. A consequence of this is that it is likely you will find that your selected stories total up to be too many (or too few) hours for your team to complete all of the tasks in one sprint. (Assume each team member is able to work forty hours per week on the project.) Add or remove stories to, or from, your selected stories so that you have enough stories for the first sprint. If you add stories, you will need to break the stories down into detailed tasks, and estimate the hours required to complete each task, to add to your sprint plan. Remember that the completed stories need to be demonstrable at the end of the sprint. Consequently, the stories need to represent some coherency in what is delivered.
Activity 4 (10 min.)	Determine the average number of hours you have estimated it will take to complete one story point. Use this to calculate an estimated velocity for your first sprint. $$V = {NT \times PH \times SL}/{SPH}$$ *V* – Velocity *NT* – Number of team members *PH* – Number of hours each person can work per week (e.g. 40) *SL* – Length of sprint, as number of weeks (e.g. 2) *SPH* – Number of hours to complete a one-point story For example, if you have estimated that one story point takes on average eighteen hours to complete, and you have four team members, that means your estimated velocity should be seventeen or eighteen story points per sprint. $$17.\dot{7} = {4 \times 40 \times 2}/{18}$$ You will need to include this calculation in the release and sprint plans, where it has places for the estimated and current velocity. Use this estimated velocity to create a delivery schedule in your release plan. The first sprint is to start on April 4. The schedule should provide a delivery date for each release. Remembering that delivery means providing a deployable product.

Required Outcome

Teams should have draft release and sprint plans for their system. They should have a clearer understanding of the amount of work involved in delivering the stories in the first release.

Further Work

Complete the release plan, and first sprint plan, based on the detailed task breakdowns identified for each story in the first sprint.