**MIS 322, FA21 Assignment 4**

**Individual Assignment**

From Chapter 15 learning more about Agile development using Sprint/Scrum…

1. 1 pts RQ15-2 (a and b only) : Compare the following terms:
   1. **Agile task board versus burndown chart**

Both of these have very similar themes where they use categories like to do, in progress, done, and backlog; but what makes them different is that a burndown is typically used when there is a time crunch involved for the project. Agile task boards tend to have due dates for tasks, but those tasks are put up as they come into play. A burndown chart however takes into account a specific plan and list of tasks that have to be completed in the allotted time.

* 1. **Agile task board versus Kanban chart**

Again, both have the same display style as I said above but the main difference being that a Kanban chart puts more focus on the right number of tasks for the time allotted. So instead of listing all the tasks that are available and putting them up on the board, a Kanban takes all the tasks and spreads them out evenly along a timeline. Then only displays the tasks on the board when it’s the right time period.

1. 3 pts RQ15-8: Regarding velocity:
   1. **Define what it is**

This is the measure of how much work is getting done on the project. This is most likely a comparison between the projected project lifecycle time and time currently used.

* 1. **Describe the pros and cons of estimating stories and tasks and measuring velocity using ideal days versus story points**

Well, the obvious negative with projecting any projects timeline is that it’s never going to exactly how you think it is. Problems and setbacks will happen and then this will create more of a possible rush to finish things and result in errors. Now the positive of creating stories and estimating velocity is seeing where you have been slowing down in the project. This would enable you to a lot more time in another project of adjust other aspects of the project that are taking up less time.

* 1. **When measuring completed work, should we measure completed individual tasks? Why?**

This again depends on the project and what type of work is being done. If project tasks are being worked on as a group of people for a single task, then seeing how much work is being competed by each individual could be good. This would enable you to see who is pulling the most weight in the task. But a possible negative of this is that some work is not measurable enough and then worker moral could be affected.

* 1. **How many sprints should we measure before arriving at an average measure of velocity?**

Typically, three sprints are the ideal number before making any sort of measurement of velocity. This can enable a more accurate measure where out lighters aren’t affecting the final number too much.

* 1. **How can we use velocity to estimate the amount of time and cost remaining in a project?**

Simply taking the averages completed tasks in a sprint and the average money spend in the sprint can give you the numbers you want. All you have to do is calculated the total number of tasks in the project and then divide that against the average number of competed tasks per sprint to get the estimated time left. The same idea could be used for the cost remaining for the project but instead of total tasks and average completed tasks per spring you take the projected number of future sprints that we just got and multiple that by the average cost of a single sprint.

1. 3pts RQ15-20: Regarding Technical Debt:
   1. **Define what it is**

This is the estimated cost and time left after a redo of completed work needs to be done because it wasn’t completed to the right standards. For example a lot of the time in order to meet project time constraints to projects tasks are band aid fixed so that they “work” but are not up to the standards that they wanted to be. After the original project is finished you have to go back and fix these band aids to the full completed version.

* 1. **Name three typical types of issues that represent technical debt**

Deliberate, accidental, unavoidable

* 1. **Explain what teams often do that typically create technical debt**

Like I said above they usually revolve around created band aid fixes to problems that can not be fixed fully in the time allotted. Other times there are just unforeseen issues that come up when something changes.

* 1. W**hat impact does it typically have on construction velocity?**

Well as stated above technical debt is more of a long term issue so if you consider the project completed when it “works” then the velocity really wont change but if you consider the tech debt in the completed project then it will extend the project and then change the velocity as well.

* 1. **When teams work on technical debt items, what do we call that?**

Refactoring?

* 1. **When teams work to eliminate technical debt, what do we call that?**
  2. **Why Does existence of technical debt often cause teams to create even more technical debt?**

Most likely because the estimated debt of the issue is much more then it actually is. There also could be issues where when fixing debt areas, you have to fix other areas in order for it to work properly.

3 pts. PE15-3, In each of the following completed burndown charts – where, in each instance, the team is running behind – determine where the problem exists, express possible interpretations for why this is happening, and measure the team’s velocity. (Refer to the charts in the book).

15-3a.

15-3b.

15-3c.