## Problems From Chapter Four

Ch 4, #4: Who are the likely candidates to serve as Scrum master?

* Steve is the likely candidate, you may ask why? Lemme tell you. Steve first got to know all his teammates individually and learned their problems. Steve became an advocate for the team and escalated issues for them. Steve facilitated discussions between his team when disagreements arise. When decisions were required Steve made suret to gather all information available to choose the correct path as with consulting internal and external stakeholders. His leadership allowed the team to shine by focusing on code.

Ch 4, #5: In what circumstances is it a bad idea to have the IT manager be the Scrum master?

* In most if not all circumstances they should not as they would be used to the waterfall style of management. This means they would be comfortable assigning tasks and making all the decisions. Something that would make a bad scrum master. They may not follow the principles of agile management as they are not accustomed to them and this would undermine the agile process.

Ch 4, #8: How large should a Scrum team be? Why does it matter?

* It depends on the size of the company so it is not standardized for every single company. As companies must know how much resources they can take on, along with how big projects will be. For example a start up would have 4 developers and 1 tester while a mid sized company will have 6 developers and 2 testers.

Ch 4, #9: What are some of the benefits of self-organizing teams?

Ch 4, #12: What is a project manager called in Extreme Programming?

Ch 4, #15: Provide several examples of stakeholders.

Ch 4, #16: Who is responsible for writing and prioritizing the requirements/user stories?

* The product owner is the what of the product defining the user stories of the project, but the scrum team is responsible for presenting those user stories as defined by the product owner. The scribe is the one who actually writes out all the documentation

Ch 4, #19: What are examples of impediments?

* Impediments are anything blocking the team's path, whether it be internal or external. Anything that keeps the team from getting work done or slows velocity. They could be sick members, resources lacking or even a member going on vacation.

Ch 4, #20: Within Scrum, who is responsible for testing?

* The QA testers test the code as it is being developed to ensure each sprint culminates with code that is ready to be deployed

## Problems From Chapter Five

Ch 5, #1: What is the user story format in Scrum?

* It is very precise and allows use to collect info in an easy format. The structure is:
  + As a type of user I want some goal so that some reason

Ch 5, #2: Name three of the six elements of a user story represented in the acronym INVEST.

* Independent -- the user story must be able to stand alone
* Negotiable -- user story should invite collaboration and discussion
* Valuable -- The reason why we do anything in agile is to drive business value

Ch 5, #3: What is an epic? What, if anything, should be done with it?

* User stories start as epics, it is too big to be designed, coded or tested in a single print. Requirements start as epics usually. The epic should be broken down by the team into numerous different child stories to be worked on

Ch 5, #4: What are the MoSCoW rulses in DSDM?

* The rules of MoSCoW is as follows
  + Must have -- all features that must be implemented
  + Should have -- priority features that make the system work
  + Could have -- features that enhance the experience
  + Want to have -- features that serve only a limited group

Ch 5, #5: What is Crystal software development very interested in with regard to requirements?

* It is about stakeholders. It is what the interests of our stakeholders, what are the needs of the users and what are the styles of those key users are in order to make the requirements

Ch 5, #8: What are three examples of business value?

* Increasing revenue
* Expanding toe addressable markets with new features
* Decreasing costs

Ch 5, #9: What is *release management*?

* Setting individual priorities the product owner is responsible for deciding how many features tneed to be included in the product before release. IT is critical with new products because you must launch something that resonates with consumers.

Ch 5, #15: How does the transparency afforded by Agile help the organization?

* It keeps key internal and external stakeholders well informed on the activities and their impacts. Agile teams must be transparent with customers in order to make sure they are meeting the needs of both the market and the product owners.

Ch 5, #17: Why does the Lean software development advocate for making decisions as late as possible?

* Delaying decisions is valuable because better decisions can be made when they are based on facts they know rather than early when it is all speculation.

## Problems From Chapter Six

Ch 6, #3: Should features provided by the competition influence prioritization?

* Yes as there are penalties for not offering a competitive feature set if your competition has a distinct advantage over you.

Ch 6, #7: Why is the Fibonacci sequence preferred over regular numbering for estimating?

* Because the points get higher when the degree of uncertainty increases and when used for story points it gives them a good rule of thumb for when things need to be broken down further or if something should be an epic or not.

Ch 6, #9: What does velocity tell a team?

* It tells the team the amount of work they can accomplish during a sprint. IT is a tool used to predict the amount of work they can commit to as a team and not a measure of productivity.

Ch 6, #10: What is *intentional technical debt*?

* Is incurred as trade-offs that are made in the development process. This is true for almost all products. You cannot have it all. It is deciding on the best path at a trade-off that must or may need to be fixed in the future.

Ch 6, #13: Describe at least two ways that teams can incorporate maintenance work into sprint planning.

* Some teams dedicate people to this during the sprint . So one developer will not take on new tasks and instead commit to maintenance.
* Another way is to split the team, one team will handle new features and the other will handle the maintenance. This works incredibly well for larger teams

Ch 6, #14: In Agile, what constraint is split into two distinct considerations?

* The triple constraints theory is split into two distinct considerations for all the constraints. If you shorten the schedule then you will need to add resources or reduce scope. If you remove resources, you will need to expand the time frame or reduce scope and finally if you add to the scrope you will need to extend the time frame or add resources.

Ch 6, #16: What is *relative sizing*?

* Involving identifying a task that everyone is familiar with hand assigning a point value to it, people become aware of the time required and the effort for these tasks already because they have done them before.

Ch 6, #18: With wide-band Delphi, how are the estimates gathered?

* Estimates are addressed in a more structured way. They call a meeting of experts and a feature request is discussed. Each expert completes a survey where they estimate and submit their estimation. And then the responses are redistributed then the feature is discussed again and submitted new estimates and the process is repeated again and again until there is an agreed upon value.