Authors: Jennifer Siao, Scott Nelson, Hayden Crabbs, Joseph Killeen, Erika Proctor
CMSI 543
Dr. Johnson
Due: Week 9
Homework 2

Problems From Chapter Four

- **Ch 4, #4: Who are the likely candidates to serve as Scrum master?**
  Likely candidates include those who are willing to make decisions and can remove roadblocks for the team. Those who can not take initiative are not great candidates for scrum master. These people can include a technical lead, project manager, IT Manager.

- **Ch 4, #5: In what circumstances is it a bad idea to have the IT manager be the Scrum master?**
  It is a bad idea to have the IT manager be the scrum master if the IT manager is following the waterfall system and makes all of the decisions. The team won't flourish in the agile development under a scrum master who is an IT manager following waterfall.

- **Ch 4, #8: How large should a Scrum team be? Why does it matter?**
  A scrum team should be five to nine people. This matters because a team less than five loses effect collaboration and more than nine gives room for too much collaboration.

- **Ch 4, #9: What are some of the benefits of self-organizing teams?**
  Some benefits of self organizing teams are empowerment to define and evolve in their respective roles in the team, outperform traditionally formatted teams, team members feel safe to give and receive feedback.

- **Ch 4, #12: What is a project manager called in Extreme Programming?**
  A project manager is called a tracker.

- **Ch 4, #15: Provide several examples of stakeholders.**
  Stakeholders could be: the operating team, technical help desk, customer service representatives, the finance team, internal stakeholders (CEO, users, etc), head of the sales organization.

- **Ch 4, #16: Who is responsible for writing and prioritizing the requirements/user stories?**
  The person responsible for this is the requirements gatherer.

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

Some examples of impediments are interruptions that result in restart cost, music so loud that causes one to lose focus, people in the organization continuing to ask you questions which kills your productivity.

- **Ch 4, #20: Within Scrum, who is responsible for testing?**
  While overall the development team as a whole is responsible for testing, the scrum master is the one who is mainly responsible to removing those impediments that are slowing the group down as a whole.

Problems From Chapter Five

- **Ch 5, #1: What is the user story format in Scrum?**
  The user story format 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.**
  I = independent
  N= negotiable
  V= valuable

- **Ch 5, #3: What is an epic? What, if anything, should be done with it?**
  An epic is a user story that is too large to be coded and tested within the span of a single sprint.

- **Ch 5, #4: What are the MoSCoW rulses in DSDM?**
  The MoSCoW rules are must have, should have, could have, and want to have.

- **Ch 5, #5: What is Crystal software development very intested in with regard to requirements?**
  Crystal software development has a user centric approach to requirements so they want to know the interests of key stakeholders and needs of the users.

- **Ch 5, #8: What are three examples of business value?**
  Three examples of business value are decreased cost, increased customer satisfaction, and increased revenue.

- **Ch 5, #9: What is release management?**
  Release management is the product owners job to determine how many features need to be included in a product for it to be released.

- **Ch 5, #15: How does the transparency afforded by Agile help the organization?**
  By having great communication and transparency between the stakeholders and all internal and external members, this allows for the project to develop properly without any

rushing/rough edges due to miscommunication. This all falls under the same idea of "Sharing the vision".

- **Ch 5, #17: Why does the Lean software development advocate for making decisions as late as possible?**
  The Lean software development advocates for making decisions as late as possible because better decisions can be made when they are factual and not speculated.

Problems From Chapter Six

- **Ch 6, #3: Should features provided by the competition influence prioritization?**
  Most definitely, it is highly beneficial to respond to a competitor's advantage with a similar or better feature. Not doing so can pose great risk to the overall value of the product, so often matching these features moves up in priority. However, doing so should be done carefully, as this can damage the entire development of the product. If it is to damage the "stability of the application" then it's overall prioritization must be considered thoroughly.
- **Ch 6, #7: Why is the Fibonacci sequence preferred over regular numbering for estimating?**
  The Fibonacci sequence is preferred over regular numbering for estimating because as the points get higher, the degree of uncertainty increases.

- **Ch 6, #9: What does velocity tell a team?**
  Velocity tells a team the usual amount of work they can typically deliver within the timeframe of a sprint. The velocity can be used to predict the timeframe for future sprints.

- **Ch 6, #10: What is intentional technical debt?**
  Intentional technical debt is incurred as tradeoffs that are made in the development process. These are used for tactical or strategic reasons.

- **Ch 6, #13: Describe at least two ways that teams can incorporate maintenance work into sprint planning**.
  1. Reserve time to deal with the bugs and plan out their resolvement during each sprint. Such as leaving 5 out of 40 points to be spent on maintenance.
  2. Another option would be to assign an entire team to handling the maintenance, or by splitting the team in half so that one focuses on development while the other continues to maintain the product.

- **Ch 6, #14: In Agile, what constraint is split into two distinct considerations?**
  In Agile there are actually three main constraints that determine the success and release

timing of the project. They are split into time, resources, and scope. The general consensus is that you cannot change one without affecting the other two in some manner.

- **Ch 6, #16: What is relative sizing?**
  Relative sizing is pinpointing a task that everyone is familiar with and assigning a point value to it. This is the easiest way for a team to get started with using story points.

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

Wide-Band Delphie allows for estimates to be gathered by utilizing an anonymous submission process for the facilitator. This allows for each members estimate to not be affecting by one another, and a more accurate estimate of the sizing to be achieved.