**Homework Assignment #1 – SYEG 557 – Agile Project Management**

Chapter 2

1. Usually developers use self-organizational teams, what this means is a developer is responsible for their own work and gets their own achievements within a group. They become a group of people working towards a single goal. Overall, it offers team culture while developers have their own autonomy that does not exist before agile, and they get much more say in their work because it is still individual. Another benefit is learning new skills due to choosing your development tasks, you may choose something a bit over your level then have an experience developer guide you through it to better your skills.

3. Frequent releases and short development cycles are now the common place usually taking place in week long or 2-week sprints in which by the end you have to get your piece done or task done for the sprint. The issue is with limited time comes increased stress and usually crunch. Personally, I know people who develop professionally and basically have a few projects a year and get to essentially chill out the rest of the time but for those projects they go week to week crunching through them and are working crazy hours just to finish their task to not slow down development.

6. Agile development is founded on the basis that people must learn to create solutions on their own. The reason is becoming its individualistic. Team members no longer wait for managers to give them solutions but are empowered to find them on their own. If mangers give them solutions then the idea breaks down and soon the developer will be reliant on the manager instead of people empowered to find their own answers to problems and learn.

9. They must trust their employees as if they do not It conflicts with the core values of agile development. The best way according to the reading is allowing their team to learn and fail on their own and just jump right into agile development. A manager must be willing to give up some control and allow their members to make their own mistakes and decisions. They must endorse teamwork and trust.

13. They must lead by example and can control the budget and priorities. They must create a non-waterfall environment where the allocate resources towards projects that do not have fully defined development processes yet. They must shift their process to one of inspection and rapid development processes. They must keep the scope of their development initially small scale in order to keep up with constant delivery and to get executives on the side of agile development.

15. 3 issues are actual time taken vs estimated time, velocity of team A vs B and number of stories with acceptance criteria vs those without. The first causes them to spend time on unproductive efforts as they may spend time on estimates instead of great development. Two teams could have different measures of effort and velocity many be more important on one than the other. Velocity may have nothing to do with effort. Third we should not reduce the conversation about the stories, it is about business value.

19. Developers may not embrace the new accountability and transparency and therefore will leave the organization instead of embracing change.

20. Agile development is about adaptation, therefore not everything will be successful, but if we have learning anything from this chapter is that there is always room for improvement on the process. You may have to look at commitment, team composition, trust and at the end of the day skillset. There may be conflicts within a team that slow development or members that cannot adapt to an agile process. Teams must overcome this but it is the manager’s job to make sure the parts are moving correctly.

Chapter 3

1. Focus on efficiency, customer satisfaction and feedback as well as quality.  Includes frequent releases in short development cycles which can improve productivity and allow checkpoints at which new customer requirements can be introduced.

3. The five activities that need to be performed in an FDD project are:

1. Develop overall model
2. Build feature list
3. Plan by feature
4. Design by feature
5. Build by feature

4. The four requirement categories that are used for DSDM are:

1. Feasibility and business study
2. Functional model / prototype iteration
3. Design and build iteration
4. Implementation

5. The seven principles of Lean Software development are:

1. Eliminate Waste
2. Amplify Learning
3. Decide as late as possible
4. Decide as fast as possible
5. Empower the team
6. Build integrity in
7. See the whole

7. The four key principles of Kanban are:

1. Encourage everyone in the organization to act and take responsibility as a ‘Leader’
2. Working with what exists and not taking assumptions. And utilize change management to evolve from what is currently present
3. The concentration must be on an incremental change and not a sudden overhaul. Small changes that are impactful are seen as vital.
4. Respecting the current methodologies and not neglecting them just because they are ‘old’.

8. The term ‘crystal’ was used to name the crystal family because just like how no two crystals are alike, no two projects will be exactly the same. Therefore, the term crystal was used to denote the crystal family.