Rachel (Thu) Le

CSS 566: Software Management

Week 9 Café Reflection

**Agile Restrospectives – Making Good Teams Great presentation by Satine**

3 things I learned from Satine:

1) Retrospectives can be used in Agile frameworks or any other product development (even in non-software related companies). In order to really make good use of Agile retrospective, the improvement plans synthesized from retrospectives need to be included in the members’ daily work plan or else no one finds time to do the extra work

2) Mastering of retrospectives can lead to mastering of other reusable skills such as mind mapping. In addition, using analytical tools during retrospectives can result in understanding of technical problems, prioritization of stories/requirements, strategy planning and driving innovation.

3) For leader/facilitator of retrospective, it is advised that this role is rotated frequently. Facilitator should focus on process and structure of retrospective, attend to the needs of group dynamics to help reach the goals and remain neutral in discussions even if they have strong opinions.

Other questions: I don’t have any other questions. It would be interesting to know what are the problems team often faces during Spring retrospectives and how to avoid/solve them. (i.e. such as blaming, criticizing people who didn’t complete their assigned work/didn’t incorporate improvement plans into their daily work plan, etc.) Overall, the presentation delivered very clear and straightforward information about Agile retrospectives.

**3 things I learned from Sanjusha's presentation about The Role of Project Manager:**

1) Project managers have many different responsibilities that require technical project management (technical knowledge is preferred), strategic business management and leadership

2) The role requires deep understanding of the business product, its market analysis and ability to develop delivery strategy as well as get project funding.

3) In organizations implementing Agile methodologies in software development, the role of project managers have evolved from team management to team leadership:

* Positional power -> Guide, influence and collaborate using relational power
* Maintain -> Develop
* Administrative -> Innovative
* Focus on systems and structure -> Focus on relationships with people
* Rely on control -> Inspire Trust
* Focus on near-term goals -> Focus on long-term vision
* Ask how and when -> Ask what and why
* Focus on bottom line -> Focus on the horizon
* Accept the status quo -> Challenge the status quo
* Do things right -> Do the right things
* Focus on operational issues & problem solving -> Focus on vision, alignment, motivation and inspiration

Question: How do project managers interact with team members differently compared to product owners? The powerpoint slides should be parts of one final poster (which wasn't available). Overall, great presentation with more detailed explanation of the project manager role compared to the LeSS is more book we read in class.

**3 things I learned from Prianka's presentation about Agile Game Development with Scrum**

1) In Agile game development, due to the extra roles of artists and the element of artistic creativity, Agile cannot be applied to game development in a strict by-the-book way. It has to tailor to the needs of both the artists and the unique characteristics of game development.

2) Some challenges of Implementing Scrum in Game development are: developers and designers as well as animation artists speak different languages, there is a collision between studio culture and agile values, a sense of skepticism about cross-disciplinary team and the common myth that Scrum is only for programmers (not others such as artists).

3) To solve the above problems, the teams can create transparency and team spirit. The artists also have to shift their mindset into thinking that they are game developer first and artists second. Third, play-testing should be used. Lastly, some of Scrum principles can be replaced with Lean Principles and Kanban practices to accommodate the unique characteristics of game development and artists.

Question: I asked the question about specific examples of replacement of Scrum principles with other agile methodologies and Prianka gave some very good answers. Overall, this is a very interesting book and the presentation delivered well the main ideas of the book as well as proved that Scrum can be used in game development. It would be interesting to know how they coordinate the audio part of the game project.

**3 things I learned from Jason's presentation about The Toyota Way:**

1) Toyota's success is built on the Lean principles presented through "4P" Model of Toyota Way:

Base: Philosophy (long-term thinking)

Next Level 1: Process (eliminate waste)

Next level 2: People and patterns (respect, challenge and grow them)

Top level: problem solving (through continuous improvement and learning)

2) Tools and methods can help company transform (and definitely can be copied and implemented in other organizations). However, change in company's philosophy such as Lean Thinking, Lean principles and continuous improvement is difficult to copy as it requires the adopting organization to change the entire company's culture and mindset. And this takes time. This is similar to agile adoption. It is usually a long process, can take years and requires a gradual shift in mindset and values, not just tools and processes.

3) I really like the idea of putting a red light on to notify everyone that there is a problem in the production line. It also requires everyone to stop and solve the problem before moving on. It seems like a hassle from the surface from it actually saves a lot of time, effort and money in the long run and company develops higher quality product from utilizing this.

Question: I don't have any question for Jason. Overall, very good presentation with more detail about Toyota's success and explanation of its model.

**3 things I learned from Chaeng's presentation about Model-based Software Management:**

1) I learned that a simple minor and major software updates (that are regularly available such as iOs, Microsoft updates, Google updates) has to take into account service availability. Especially, any prolonged absence of service available would jeopardize the reputation of the service provider(s).

2) There are many challenges of Upgrading when service availability is taken in account:

* compatibility
* synchronizations
* errors and bugs introduced by upgrade (we've seen this a lot when a new iOs is available)
* from the software specific

3) The upgrade can be rolled out in one step or through an upgrade procedure with multiple steps: splitting up the group (to avoid service outage), take each of these subgroup out then upgrade the entities.

Question: I would like to see more specific examples in real life how these upgrades are carried out and if there are any situations when they fail (big enough to cause a huge amount of customer complaints). I also want to see how agile methodologies are used in model-based software management. (I don't know if that is discussed, we also didn't have much time at the end for questions) Overall, very interesting presentation!

**3 things I learned from Anish's presentation about The Art of Agile Development:**

1) Five Agile principles are: improve the process, rely on people, eliminate waste (lean thinking), deliver value and seek technical excellence

2) Extreme Programming has elements of test-driven development, pair programming, code simplicity, YAGNI (build only things you need to build), expect change and small frequent release. Besides the element of pair programming and YAGNI, XP shares the rest elements with other agile methodologies such as Scrum.

3) Values of XP:

* Courage: make the correct decisions when it's hard and always tell stakeholders the truth (I think it is similar and should be done in any agile approaches)
* Communication: give the right people the right information (agile value)
* Simplicity: get rid of whatever isn't really needed (Lean)
* Feedback: keep learning at every opportunity (agile)
* Respect: treat everyone with respect and dignity and acknowledge expertise and shared goals (this should be in every situation and aspect of software development, not just XP)

Questions: Are there any new/interesting things you learn from this book besides XP?

I don't see the slides being put in one final poster. Overall, great presentation and summary of agile principles and values!