**Jackson Tan: Scrum in Action**

Top three things I learned from Jackson’s presentation about Scrum in Action book:

1. There are two models used for identifying stakeholders and their goals as well as gathering requirements for Product Backlog: SMART (Specific, Measurable, Achievable, Realistic, Time-based) and CUTFIT (Consistent, Feasible, Independent, Traceable)

2. The book also provided information on how to “make company- wide comparable estimation” for project time, money and resources by a scoring system that calculate Points per Story (PPS). One of its components is finding points based on requirement types.

Unadjusted Points (UP) from 4 types: Interaction Type, Business Rules, Entities and Data manipulation. Simple = 1 pt, Average = 2 pt and Complex = 3 pt. This really resonates with the Cynefin framework we will have to identify if a problem/requirement is obvious, complicated, complex or complicated to come up with solutions and approaches to solve the problem appropriately.

3. The larger the value of PPS (more points per story), the longer time and resources it takes to complete that story/requirement.

One further question: In the Scrum Pentagon, we read it from Vision and travel clockwise to Analyze requirements, Design architecture, Plan Sprints and then Estimate project. Is this in chronological order? I’m wondering why it is ordered that way because I think Estimate project should come before Sprint planning.

Overall, good presentation! Maybe face the audiences more with eye contact to help check if your audiences are following and understanding your presentation (though I understand you completely).

**Andrew Nakamura: The Obstacle is The Way**

Top three things I learned from Andrew presentation:

1. We don’t (can’t) control what happens to us but we can control how we respond to what happens to us. That brings an opportunity for change that we wouldn’t have thought of if we were to accept and be passive about what happens to us. I think this concept applies to many aspects of life (entrepreneurship, personal happiness, etc.) including software management. As it turns out, the more complex a software is, the more unpredictable it becomes in terms of when something wrong happens. Focusing on what we can do to change the situation instead of what happened, takes back the control from outside external force into our hands. In addition, we assume unpredictability is inevitable and prepare our mindset and actions to always be on the lookout for changes and be ready to adapt to changes. (i.e. Monkey Army in Netflix, Cynefin framework)

Three important concepts in this book are Perception, Action and Will (how we perceive what happens to us, what actions we take to respond to the situation and the perseverance to carry through despite hardship). I think these three concepts can be related to system thinking and continuous improvement towards perfection.

2. Solving problems, especially complex ones, requires an ability to look and think in a big picture. (System thinking) In software management, it’s easy to do a quick fix, increase velocity and meet deadlines but the delayed cost of not fixing at the problem as its root can be detrimental later on.

3. Continuous improvement towards perfection is also a way of think and to perceiving the situation. If we think our product is perfect and flawless, we don’t see the need for improvement (and that gives our competitors opportunities to take our customers by making better, more improved product than ours).

One further note: Overall, I love the presentation and how Andrew relates it to LeSS principles. I’m just wondering if there are more LeSS principles that can be related to this book besides the three mentioned during the presentation: Lean Thinking, System Thinking and Continuous Improvement towards Perfection.