

## **PMO Community of Practice December 2018 Meeting Summary**

### **Header (Required)**

Title: Agile tools and techniques

Brief description: Agile tools and techniques - agile estimation, planning, development, testing and sprints.

Learning objectives: Understanding agile processes and how they are practiced in different organizations

PDU Amount: 1 PDU

PDU Allocation: Technical

Technical Domains: Agile

*Note: Technical PDUs must be within the specific technical domains of: portfolio management, business analysis, agile, risk, and scheduling*

### **Member Discussion Topic 12/11/18: Agile tools and techniques**

- Tools
  - Jira, Rally (Agile Central) and VersionOne
  - The tool works for you when data is good
  - Reporting capabilities and different points of view
- Agile leverages short periods of development, typically 2-4 weeks
  - Quickly adjust as you learn
  - Sustainable pace
  - Planning 2 weeks in detail is easier than planning the entire project
- Agile stories (typically the requirements or functionality)
  - Based on Who needs What and Why
  - Define done and how you know the story can be verified as completed
  - When ready to work on for the next sprint, drill down into tasks
- Agile estimation
  - Product Owner explains the story, which is discussed among the team members
  - Size of the story is order of magnitude, typically using points for small, medium, large, extra large, etc.
  - Team discusses relative size (not hours). Faster developer may take less time than slower developer, but the story size would be the same. 10 story points would be expected to take either developer twice as long as 5 story points.
    - Note: For an illustrative example, the faster developer might need 1 day for 5 points and 2 days for 10 points and the slower developer might need 2 days for 5 points and 4 days for 10 points
  - The team discusses their size estimates and assumptions to arrive at a consensus (not an average!)
- Planning the sprint
  - The Product Owner sets priorities, typically ranking the stories
  - The team uses their number of points finished in recent / representative sprints (velocity) to determine which stories can be done in the next sprint
  - Testing is normally part of the definition of done. Some teams do testing of stories in the next sprint, but this is problematic (and not recommended)



- Automation is also normally part of done to catch anything that was working before but broke in the current sprint without putting unreasonable pressure on testing
- Execution
  - Set a cadence – sustainable pace
  - Minimize work in progress – much better to have stories worth 100 points completed, including testing and acceptance, than stories worth 200 points half completed
  - Determine how many stories can be done at the same time (collaboration vs parallel play)
  - Test throughout the sprint – do not leave it to the end
- Release planning
  - Based on the team’s velocity and the sum of the sizes for the stories the number of sprints can be estimated
  - Some organizations leave the last sprint for final regression testing

**Next Meeting**

Next meeting is Tuesday, January 8, 2019

**Participants:** *Adella Rosenthal, David Morgen, Marlo Moss, Richard Loeb, T. T. Ruwan Fonseka*

Chair: Richard Loeb, PMP  
Scribe: David Morgen, PMP