Chapter 11: Project Management Certifications

Certification in project management is available from the Project Management Institute, PRINCE2, ITIL, Critical Chain, and others. Agile project management methodologies (Scrum, Extreme Programming, Lean Six Sigma, others) also have certifications.

11.1 Project Management Institute Overview

Five volunteers founded the Project Management Institute (PMI) in 1969. Their initial goal was to establish an organization where members could share their experiences in project management and to discuss issues. Today, PMI is a non-profit project management professional association and the most widely recognized organization in terms of promoting project management best practices. PMI was formed to serve the interests of the project management industry. The premise of PMI is that the tools and techniques of project management are common even among the widespread application of projects from the software to the construction industry. PMI first began offering the PMP certification exam in 1984. Although it took a while for people to take notice, now more than 260,000 individuals around the world hold the PMP designation.

To help keep project management terms and concepts clear and consistent, PMI introduced the book Project Management Body of Knowledge (PMBOK) Guide in 1987. It was updated it in 1996, 2000, 2004, and most recently in 2009 as the fourth edition. At present, there are more than 1 million copies of the PMBOK Guide in circulation. The highly regarded Institute of Electrical and Electronics Engineers (IEEE) has adopted it as their project management standard. In 1999 PMI was accredited as an American National Standards Institute (ANSI) standards developer and also has the distinction of being the first organization to have its certification program attain International Organization for Standardization (ISO) 9001 recognition. In 2008, the organization reported more than 260,000 members in over 171 countries. PMI has its headquarters in Pennsylvania, USA and also has offices in Washington, D.C., and Canada, Mexico, Beijing, China, as well as Regional Service Centers in Singapore, Brussels (Belgium) and New Delhi (India). Recently, an office was opened in Mumbai (India).

11.2 Scrum Development Overview

Scrum is another formal project management/product development methodology and part of agile project management. Scrum is a term from rugby (scrummage) that means a way of restarting a game. It's like restarting the project efforts every X weeks. It's based on the idea that you do not really know how to plan the whole project up front, so you start and build empirical data, and then re-plan and iterate from there.

Scrum uses sequential Sprints for development. Sprints are like small project phases (ideally 2 to 4 weeks). The idea is to take one day to plan for what can be done now, then develop what was planned for, and demonstrate it at the end of the Sprint. Scrum uses a short daily meeting of the development team to check what was done yesterday, what is planned for the next day, and what if anything is impeding the team members from accomplishing what they have committed to. At the end of the Sprint, what has been demonstrated can then be tested, and the next Sprint cycle starts.

Scrum methodology defines several major roles. They are:

- Product Owner(s): essentially the business owner of the project who knows the industry, the market, the customers and the business goals of the project. The Product Owner **must** be intimately involved with the Scrum process, especially the planning and the demonstration parts of the Sprint.

- Scrum Master: somewhat like a project manager, but not exactly. The Scrum Master's duties are essentially to: remove barriers that impede the progress of the development team, teach the Product Owner how to maximize ROI in terms of development effort, facilitate creativity and empowerment of team, improve the productivity of the team, improve engineering practices and tools, run daily standup meetings, track progress, and ensure the health of the team.

- Development Team: self organizing (light touch leadership), empowered, participate in planning and estimating for each Sprint, do the development, and demonstrate the results at the end of the Sprint. It has been shown that the ideal size for a development team is 7 +/- 2. The development team can be broken into teamlets that 'swarm' on user stories, which are created in the Sprint planning session.

- Typically, the way a product is developed is that there is a Front Burner (which has stories/tasks for the current Sprint), a Back Burner (which has stories for the next Sprint), and a Fridge (which has stories for later, as well as process changes). One can look at a Product as having been broken down like this: Product -> Features -> Stories -> Tasks

Often effort estimations are done using 'Story Points' (Tiny = 1 SP, Small = 2 SP, Medium = 4 SP, Large = 8 SP, Big = 16+ SP, Unknown = ? SP) Stories can be of various types. User stories are very common and are descriptions of what the user can do and what happens as a result of different actions from a given starting point. Other types of stories are: Analysis, Development, QA, Documentation, Installation, Localization, Training, etc.

Planning meetings for each Sprint require participation by the Product Owner, the Scrum Master, and the Development Team. In the planning meeting, they set the goals for the upcoming Sprint and select a subset of the product backlog (proposed stories) to work on. The Development team de-composes stories to tasks and estimates them, and the Development team and Product Owner do final negotiations to determine the backlog for the following Sprint.

Scrum uses metrics to help with future planning and tracking of progress. A few of them are: Burn down – The number of hours remaining in the Sprint versus the time in days. Velocity – Essentially, how much effort the team completes per Sprint. (After approximately 3 Sprints with the same team, one can get a feel for what the team can do going forward.)

Some Caveats about using Scrum methodology: 1) You need committed, mature developers, 2) You still need to do major requirements definition, some analysis, architecture definition, and definition of roles and terms up-front or early, 3) You need commitment from company and the Product Owner, and 4) It is best for products that require frequent new releases

or updates, and less good for large, totally new products that will not allow for frequent upgrades once they are released.