James Park, SNHU ID: 2710902

CS 250 SDLC 23EW1

Module 7 Final Project

## Module 7 Final Project – Sprint Review and Retrospective

* Explain the different methodologies and stages involved in the software development life cycle
  + There are several Software Development Life Cycle (SDLC) methodologies. Historically, software was created using a construction model with phases like requirements analysis, architecture, design, construction/coding, inspection/testing, and delivery/operations. Compiling and executing code was a physical process with punch cards and heavy machinery with long turnaround times for design changes. It was advantageous to make all decisions within a phase final and fully documented before proceeding. Each phase could involve months of work followed by documentation and phase handoff milestones. An end-to-end software project could last for over a year or more. This predictive life cycle was described in literature as a ‘waterfall’ model (Royce, W. 1970) and worked for decades during a paradigm when requirements were not very dynamic and users did not have sophisticated expectations.
  + As software technology and platforms advanced so did software development as well as user requirements sophistication. The ability to deliver updated software more frequently, even in smaller feature sets, became advantageous for development organizations. The concept of delivering features for an enterprise application, like an office productivity suite, on a monthly basis (i.e. using 4 week delivery cycles) instead of yearly presented advantages for incrementally delivering customer value and iteratively responding to changing requirements (Sutherland and Schwaber, 1995). The term ‘Scrum’ was referenced in a Harvard Business Review paper (Takeuchi and Nonaka, 1986) using the rugby term to describe small autonomous self-organizing product development teams moving deliverables ‘across a field’ in a more agile manner than traditional large product development teams. This term became the model nomenclature for Sutherland and Schwaber to describe a framework of roles, management events (or ceremonies), and artifacts that enabled short increment cycles of software delivery and that has become the basis of what is known as Scrum today.
  + In 2001 a broader group of software leaders came together to form a more defined idealogy of principles around effective software delivery (Beck et.al., 2001). This Agile Manifesto outlines the values and principles of modern software development methodologies like Scrum, XP, Lean/Kanban, and others.
* Apply the stages of the software development life cycle within an agile development environment
  + As described in Takeuchi and Nonaka’s paper, smaller teams that could utilize more parallel processes for requirements gathering, inspection/adaption, and frequent deliveries could overlap phases of product development and outproduce traditional teams and processes. Within an agile development environment, this translates to a framework like Scrum by compressing the life cycle into iterations of 1-2 weeks for example with requirements analysis (backlog refinement), frequent delivery (sprint review/demo) and inspection/adaption (retrospectives) occuring multiple times a month.
* Evaluate various software development methodologies and tools
  + Historically, software development has trended towards more iterative/adaptive and shorter life cycles. This has reflected the sophistication of user requirements as well as the advancement of development platforms and management tools. A predictive model is advantageous in certain situations where total budget or timeline must be pre-determined and the scope of work is well understood and controlled. In most other situations, however, an adaptive/agile or hybrid approach provides more benefit by enabling coordinated teams to deliver higher value features faster and adapt to changing requirements more quickly.
  + Scrum has become one of the most prevalent frameworks in software development due to it’s simplistic model and low barrier of entry for most organizations. The framework is broken down into several areas (Sutherland and Schwaber, 2020):
    - Roles:
      * Dev Team – ‘the people in the Scrum Team that are committed to creating any aspect of a usable Increment each Sprint.’ ‘The team is accountable for: Creating a plan for the Sprint, the Sprint Backlog; Instilling quality by adhering to a Definition of Done; Adapting their plan each day toward the Sprint Goal; and, Holding each other accountable as professionals.’
      * Product Owner – ‘accountable for maximizing the value of the product’ and for managing the Product Backlog.
      * Scrum Master – ‘accountable for the Scrum Team’s effectiveness.’ ‘enabling the Scrum Team to improve its practices’ by coaching the team, removing impediments, and enabling the overall process.
    - Artifacts:
      * Product Backlog – A prioritized list of refined features often structured as user stories.
      * Sprint Backlog – A subset of features planned by the Dev Team to deliver within a sprint.
      * Product Increment – The completed work of the sprint often demonstrated in the Sprint Review.
    - Ceremonies:
      * Sprint Planning – A planning meeting where Dev Team works with Product Owner to determine the goals of an upcoming sprint.
      * Daily Stand Up – A short recurring team meeting to provide updates on completed work, upcoming work for the day, and any impediments. Often a standing meeting to encourage short duration.
      * Sprint Review – A review at the end of the sprint to go over completed features with stakeholders, often in a demo format. The goal is to review the deliverables.
      * Sprint Retrospective – A meeting focused on process improvement brainstorming for upcoming sprints. Often held by the Dev Team and facilitated by Scrum Master.
* Demonstrate how the various roles on your Scrum-agile Team specifically contributed to the success of the SNHU Travel project. Be sure to use specific examples from your experiences.
  + Each Scrum role contributed to the successful outcome of the SNHU Travel project. The Product Owner role collaborated with customer and other stakeholders to collect and prioritize features. The PO also was able to make changes to requirements from sprint to sprint to adapt rapidly to user or customer needs. The Scrum Master facilitated communication within the team and with the PO. At the end of the sprint, the SM coached the team to brainstorm process improvements for future sprints during the Sprint Retrospective. The Dev Team was proactive in communicating with each other (Testers and Devs) and with the PO to gather the necessary information required to get clarification on stories and efficiently respond to changing requirements.
* Describe how a Scrum-agile approach to the SDLC helped each of the user stories come to completion. Be sure to use specific examples from your experiences.
  + The Scrum-Agile approach facilitated the completion of user stories in several ways. First, the Product Backlog artifact provides a structured tool for PO and team to organize, breakdown, prioritize, and clarify the necessary work for upcoming iterations. The Definition of Ready (DoR) and Definition of Done (DoD) are effective concepts for teams to enable transparency and flow of work. Next, the Sprint Planning and Sprint Review ceremonies provide a structured process for moving stories quickly from Product Backlog to completion. Finally, information radiators such as burndown charts and scrum-kanban boards provide visual performance reporting in a quick fashion for a wide range of stakeholders such as the PO or executive management.
* Describe how a Scrum-agile approach supported project completion when the project was interrupted and changed direction. Be sure to use specific examples from your experiences.
  + Short iterations and high-fidelity communication via face-to-face or electronic platforma enabled the team to pivot to higher value work efficiently. One example of this was demonstrated when the PO presented changes to user requirements that the team was able to adapt in a short timeframe.
* Demonstrate your ability to communicate effectively with your team by providing samples of your communication. Be sure to explain why your examples were effective in their context and how they encouraged collaboration among team members.
  + Daily stand up meetings (either face-to-face or group instant messaging) enabled the team to stay apprised of dependencies, changes to tasks, and impediments in a much faster fashion than a weekly status meeting. Keeping the communication in a group context encouraged collaboration over a 1:1 method like email.
  + When clarification questions were presented in a distributed team environment, email provided the best available communication method to document the questions and direct the communication to the appropriate team members such as the PO or dev/tester. An example of this was encountered when the test team member presented questions to the PO about how to test certain features in the sprint.
* Evaluate the organizational tools and Scrum-agile principles that helped your team be successful. Be sure to reference the Scrum events in relation to the effectiveness of the tools.
  + Organizational tools such as JIRA provided a single platform to manage backlogs, scrum boards, issues/bugs, and releases. The agile principle of information radiators was enabled via JIRA burndown charts and Kanban boards. The platform also provided notifications to relevant team members for comments or issues related to any user story. Finally, the platform provided performance metrics for teams such as Control Charts which capture flow metrics of stories such as mean-time to delivery or time wasted in wait-states.
* Assess the effectiveness of the Scrum-agile approach for the SNHU Travel project. Be sure to address each of the following:
  + Describe the pros and cons that the Scrum-agile approach presented during the project.
    - Pros: Transparency of work and priorities as displayed in the Product Backlog. Enabled self-organization of the team to determine sprint goals and present deliverables in the Sprint Review. Ability to quickly adapt to changing requirements through short iteration cycles and clearly defined roles for PO and team. Encouraged collaboration through stand up and retrospective meetings.
    - Cons: Reduced focus on overall program timelines and budget. Complexity to deliver large-scale applications with multiple teams and release trains.
  + Determine whether or not a Scrum-agile approach was the best approach for the SNHU Travel development project.
    - The Scrum-agile approach is clearly the best approach for a software development project like SNHU Travel. It provides the most benefits to adapt to change and deliver the highest value features in the shortest amount of time. It encourages self-organizing teams and collaboration over a traditional waterfall approach.

References:

* Royce, W., (1970 Aug 1). *Managing the Development of Large Software Systems*. Proceedings of IEEE WESCON 26. Retrieved from https://www.praxisframework.org/files/royce1970.pdf.
* Takeuchi, H., Nonaka, I. (1986, Jan). *The New New Product Development Game.* Harvard Business Review. Retrieved from https://www.thescrummaster.co.uk/wp-content/uploads/2016/09/The-New-New-Product-Development-Game.pdf.
* Sutherland, J., Schwaber K. (1995). *The SCRUM Development Process.* Conference: Object-Oriented Programming, Systems, Languages & Applications (OOPSLA Austin, Texas). Retrieved from https://www.thescrummaster.co.uk/wp-content/uploads/2016/09/SCRUM-Development-Process-K-Schwaber.pdf.
* Beck, K. (2000). *Extreme Programming Explained: Embrace Change.* Addison-Wesley
* Beck, K. et.al. (2001 Feb 11). *The Agile Manifesto.* Agile Alliance. Retrieved from https://agilemanifesto.org/history.html.
* Leffingwell, D. (2011 Dec). *Introducing the Scaled Agile Framework.* Blog: Scaling Software Agility. Retrieved from https://scalingsoftwareagility.wordpress.com/2011/10/23/introducing-the-scaled-agile-framework%e2%84%a2/. (Updated corporate site - https://scaledagileframework.com/)
* Sutherland, J., Schwaber, K. (2020). *The 2020 Scrum Guide.* Retrived from https://scrumguides.org/scrum-guide.html.