Comparing Scrum vs Plan Driven Approach

Simon Codrington III

Grand Canyon University

SWE-520: Advanced Software Engineering Fundamentals

Dr. Wibbenmeyer

April 3, 2024

In project management, there are several key factors that are important for a successful outcome or product. These factors include the allocation of people to projects, estimating the cost for the project, strategies for maintaining team cohesion, and managing changes to the actual development team/stakeholders. There are two frameworks for project management that dominate the industry which are Scrum vs Plan Driven development. Each has their strengths and weaknesses and each has a different impact on the aforementioned factors.

*Planning for Allocation of People*

Who will work on what is an import process and scrum vs plan driven approaches handle them very differently. As it pertains to plan driven approaches, how engineers interact with the code is thoroughly planned out. Time is taken to identify what people are good at and comfortable with. This process tends to happen during the design phase of what the technology stack of the system will be. Highly specialized people are placed according to their strengths to work on their tasks. This works well for complex systems that need to have other subsystems to function. It is also great for larger organizations with a lot of people to allocate. For instance, backend developers focus on server-side programming, while front-end developers handle user interface design. In scrum this is not the case. In scrum, while we do need to allocate people to specific tasks, scrum takes a more team-oriented approach. Everyone is responsible for a small sprint and a deliverable. We need to allocate a given amount of time in person-work hours to this sprint. Each sprint being no more than 2-4 weeks. (Sommerville, 2015, p.73) At the end of each sprint, allocation becomes more about managing the back log due to not extending sprint lengths. (Sommerville, 2015, p.73) This is great with small organizations with talented engineers.

*Estimating Project Cost*

As it pertains to trying to estimate the cost of a project, Scrum and plan driven development take vastly different approaches. While things change, in the plan driven approach we tend to measure cost based on the requirements. The client states a detailed plan as to what the requirement of the system is, and a detailed plan is crafted with estimates of time needed to bring to production. After planning out the architecture, technology stack, and times estimated, a contract can be drawn up to reflect said cost. This is good for companies that normally operate in this manner, like most large-scale companies do. This does not adapt well to change, however. Change is something that is almost entirely inevitable. Scrum’s approach is short and iterative. Scrum does not want to focus on requirements and final product. Therefore, the only way to determine cost will be the number of contracted hours of people time you will need to dedicate to each sprint. After each sprint the client gives a lot of feedback for the next sprint. Cost in scrum environments tend to be difficult to scope out due to not planning too far ahead. Their argument is *well, this was going to cost more even if we planned anyway because change is inevitable.*

*Managing Changes to Development Team/Stakeholders*

This is another area where plan driven development vs scrum is diametrically opposed. Plan driven development takes the stance that *people will leave and not everyone is Tony Stark*. As in agile requires high skill level from their engineers more so than the scrum counterpart.(Sommerville, 2015, p.79) The anticipation of team turnover and lack of high skill is planned for and is a large reason for such high levels of planning and documentation. Plan driven development handles this well because it is easier to on-board new and different talent and talent of differing skill levels. Scrum, on the other hand, does not handle this well. Due to less formal documentation, a lot of knowledge is lost when developers leave. This causes systems that are difficult to maintain and scale.

*Maintaining Team Cohesion*

Team cohesion takes two different forms in Scrum vs Plan Driven. As it pertains to PDD, cohesion comes in the form of coordination. On larger scale projects, often times there are different teams working on different things at completely different times in different locations. Coordinating time and task dependencies are crucial. Certain tasks cannot start until others are completed. This is referred to as the “critical path.” (Sommerville, 2015, p.664) These plans are not suited to change well and can be susceptible when unforeseen problems occur. Cohesion in scrum comes in the form of “people, not process.”(Sommerville, 2015, p.77) Scrum is all about collaboration, and in this environment, cohesion is everything. This comes in the form of daily standups where developers update progress and look for solutions to problems. Also redirect attention to certain needs. Scrum is meant for co-located teams and this is well-supported. PDD can be difficult in large organizations that are more spread out.

*Citations*

OpenAI. (2024, March 26). GPT-3.5 (or ChatGPT). https://www.openai.com/

Sommerville, I (2015). *Software engineering* (10th ed.). Pearson.