

Brian Applegate

June 25, 2015

Communication Importance

The growth of technology over the years has exponentially increased opportunities for software development. Now, all kinds of different types of industries interact with software, whether it is construction companies needing software for logistics or even a cook needing a website. This technology growth has especially increased opportunities for multi-discipline teams. New software has allowed video game companies to have specialized artists, who know nothing about computer science, work with the software developers to create a cohesive product. With the surge of cross-discipline teams, communication is now the forefront of project success. Great communication allows teams to have extremely high project visibility. Ways of achieving project visibility has changed over the years; however, team communication has always played a large role. *Mythical Man Month*, *Rapid Development*, *Agile Software Development with Scrum*, and *Agile Project Management with Scrum* are all books relating to software development, which were written at significantly different times concerning the software industry. The books were written in 1975, 1996, 2002, and 2004, respectively. While all books agree communication is a key factor in project visibility, they each believe in different communication methods.

Mythical Man Month, written by Frederick P. Brooks, Jr, was written at a time in which development teams were composed of only project managers and computer scientists. Considering the team structures, Brooks focused more on following documentation and a hierarchical communication chain. Team members did not have many issues regarding differences in knowledge base, which meant documentation and processes were more useful since team members generally understood design in similar ways. In addition, since software

development was a new and growing industry type at the time, companies tried to follow development methods used in other industries. The other industries at the time used development methods such as waterfall and other traditional development methods, which did not allow for rapid iterations and change. Brooks talks about how having a solid design up-front with detailed specifications and writing extensive documentation is key to getting teams on the same page [1]. The theory is that if teams know exactly what to do, when, and how, then managers can easily see the project status. Highly detailed plans and design allow for high project visibility. Communication combined with organization is critical for success. Brooks relies on the documentation to enable teams to communicate better with each other and achieve project visibility.

Twenty-one years later *Rapid Development*, authored by Steve McConnell, was published and it stresses the importance of following software fundamentals as a way to communicate to team members the product plan. One software fundamental McConnell discusses is requirements management. Requirements management is the act of collecting the project requirements and prominently displaying them in many ways for the team to read them [2]. Spreading the requirements documentation is highly important to maintain the project heading. Although a team may have a perfect set of requirements, if a team member does not have easy access or an old set of requirements, then the perfect set is useless. In addition to requirements management, McConnell advocates construction fundamentals. These are a set of guidelines and practices for developers to follow which allow for consistency within the overall system [2]. Construction fundamentals are considerably important for large complex intertwining systems. Readability and debug-able code is critical for staying on track and for avoiding complete system rework. Ensuring teams follow the software development fundamentals

provides the team with improved communication and more of an understanding of the project, which leads to better project visibility.

Agile Software Development with Scrum, written by Ken Schwaber, emphasizes the importance of daily communication among the team. Schwaber wrote the book with the perspective of using scrum as the development methodology, which lends itself to have more frequent project inspection and different development stages. The book explains that the transfer of learning between team members improves team communication. The scrum methodology allows teams to share knowledge in different ways. Individuals share both explicit knowledge and tacit knowledge in scrum. Explicit is more of the systematic information that can be shared and the tacit is the experience that can be shared among team members [3]. Scrum also allows diverse team members to communicate daily, which improves rapidly as development progresses. The scrum team is self-organizing and forces communication among the team to accomplish goals. Since the team members share information among the entire team every day, there is a more clear indication of project visibility.

Two years later Ken Schwaber wrote another book, *Agile Project Management with Scrum*. Schwaber focuses more on the act of having an individual inspect the progress of the scrum team on a high level and then make a decision with the team on how to improve processes and development. This is part of the empirical process control explained by Schwaber. Visibility, inspection, and adaptation are steps needed for implementation of empirical process control [4]. Visibility is the step of the process that must be visible to those controlling the process. Teams need to identify and communicate difficulties clearly. Aspects of process need inspections frequently enough to detect unacceptable variances in the process [4]. The inspection step requires knowledge and skills from the inspector to assess what they are inspecting. The

frequency of inspection needs to be carefully chosen because the team needs time to adjust to the new process to see if it really works. After processes have changed, the empirical process relies on the team to identify if they are improving in the changed processes. Adaptation is the final step in the empirical process and it is the act of adjusting a process in an attempt to come inside the acceptable limits of the resulting product [4]. Using the empirical process control described by Schwaber advocates teams to communicate directly to help improve workflow and understand one another, which all lead to better project visibility.

All books advocate for some form of frequent examination of project status and goals. *Mythical Man Month* advocates team member to come up with a plan and design, and then frequently check their work against the plan set in place. The communication comes out of the highly detailed documents, which need to be followed by the entire team. Similar to *Mythical Man Month*, *Rapid Development* focuses on teams coming up with a plan using software fundamentals. Teams document the product requirements and construction fundamentals for the entire team to follow. These documents are shared with every team member and made highly visible for frequent review. Both *Agile Software Development with Scrum* and *Agile Project Management with Scrum* emphasize frequent evaluation through daily communication and process inspection, respectively. Although all four books were published at different times over a 25-year period, the need for frequent communication to establish project visibility has not changed.

Project visibility for software development is critically important for project success and it has not changed over the last 40 years. Communication is key for project visibility, as shown by the four books discussed. Ways of communicating have evolved with the growth in technology and communication styles will continue to change. Communication is a challenge all

teams will face because there is no perfect way for teams to communicate, whether the team is extremely diverse or even if the team has all the same communication styles. All four books offer great methods to help solve the challenges of communicating effectively. They each rely on some form of frequent communication to achieve project visibility. A key lesson for all development teams to take away from these books is communicate, communicate, and communicate to help improve project visibility.

References

- [1] Brooks, Frederick P. *The Mythical Man-month: Essays on Software Engineering*. N.p.: n.p., n.d. Print.
- [2] McConnell, Steve. *Rapid Development: Taming Wild Software Schedules*. Redmond, WA: Microsoft, 1996. Print.
- [3] Schwaber, Ken, and Mike Beedle. *Agile Software Development with Scrum*. Upper Saddle River, NJ: Prentice Hall, 2002. Print.
- [4] Schwaber, Ken. *Agile Project Management with Scrum*. Redmond, WA: Microsoft, 2004. Print.