Agile software development was coined in 2001 began to impact software development in major ways. Software developers had begun to question whether the way they built software was the best way for them. In response, different methodologies were constructed for how to build software and eventually a framework was created. Then the Agile Manifesto was created shortly in the months following. Within that manifesto, there are values placed on individuals and interactions, working software, customer collaboration, and response to change. These principles are all important however, they may not always be ideal for software development.

The first statement within the Manifesto for Agile Software Development is that Individuals and interactions have a higher value over processes and tools (Manifesto for Agile Software Development, 2001). This is an increasingly important statement for Agile Software Development. Within Agile Software Development different teams of individuals working on different components or aspects of the same project or product need to value the others they work with as well as all interactions. Communication is essential in Agile Software Development. The members need to communicate over finances, requirements or changes to requirements, risks, performance measurements, optimization, and efficiency (Bourque, 2014). Everyone needs to communicate and be on the same page throughout the entire product or project life cycle. Otherwise, the risks will soon become reality and consume the project leading to failure and an economic loss (Bourque, 2014).

The second statement in the Manifesto for Agile Software Development is that working software has more value over comprehensive documentation (Manifesto for Agile Software Development, 2001). This, however, may not be true in all cases. The manifesto also states, “We are uncovering better ways of developing software by doing it and helping others do it.” (Manifesto for Agile Software Development, 2001). Thus, making the second statement seem contradictive. Working software is important and does have a lot of value but, in order to help other developers and other programmers, comprehensive documentation is extremely important. Others need to be able to understand the work that has already been done. Also, circumstances are unpredictable in some situations and someone may need to complete or work on someone else's started project. If they cannot understand what is going on or where to start from it causes more harm to the project timeline and even to team communication (Bourque, 2014).

The third statement in the manifesto emphasizes the importance of customer collaboration over contract negotiation (Manifesto for Agile Software Development, 2001). Customers are very important to any and all businesses and should maintain the priority and focus. Especially with software development, the customer's requirements are the end goal and the customers need to be involved in review the software to ensure that the end goal (all their requirements) are meet how they need them to be (Bourque, 2014). Therefore, customer collaboration should be a priority over contract negotiation. The last statement is the fact that the response to change is more important than following a plan (Manifesto for Agile Software Development, 2001). This is true due to the fact that responding to the customers change requests is essential, even if it throws the entire pre-set plan or schedule. There also may be other risks or challenges that are presented that may upset the schedule and response to the change will be needed (Bourque, 2014). Therefore, it is more important to respond to change rather than following a plan.

Overall the Manifesto for Agile Software Development provides an excellent basis for the focus of Agile Software Development. In some cases, comprehensive documentation may prove to be more valuable than working software, however, working software is still very important. Agile Software Development has provided for leaps within software development thanks to its flexibility and communication components. As time moves forward agile software development will continue to adapt and expand, bringing benefits to the software development community.

Work Cited

Agile Alliance. “What Is Agile Software Development?” *Agile Alliance*, 16 Jan. 2019, www.agilealliance.org/agile101/.

Beck, Kent, et al. “Manifesto for Agile Software Development.” *History: The Agile Manifesto*, 2001, agilemanifesto.org/.

“Chapter 12: Software Engineering Economics.” *Guide to the Software Engineering Body of Knowledge Version 3.0 ; SWEBOK ; a Project of the IEEE Computer Society*, by Pierre Bourque, IEEE Computer Soc., 2014, pp. 206–221.

Hales, Laury. “Pros & Cons of Agile Development Methods.” *Study.com*, Study.com, 2013, study.com/academy/lesson/pros-cons-of-agile-development-methods.html.