The Characteristics of Agile Software Processes

Granville G. Miller
rmiller@togethersoft.com
TogetherSoft

Abstract

The competitive pressures of today’s businesses require delivery of software solutions in the shortest time possible. Software systems that once took years to deliver can now be created in months. The enabler of this transformation is the agile software process. We will look at the characteristics of an agile process. Specifically, we will look at what makes a software process agile and how different agile processes work to allow your project to be delivered more quickly.

1. Characteristics of Agile Processes

The name “agile software process”, first originated in Japan [1]. The Japanese faced competitive pressures, and many of their companies, like their American counterparts, promoted cycle-time reduction as the most important characteristic of software process improvement efforts. To reflect its importance, time would become one of the nine characteristics of an agile process [1].

Characteristic 1: Modularity

Modularity is a key element of any good process. Modularity allows a process to be broken into components called activities. A software development process prescribes a set of activities capable of transforming the vision of the software system into reality.

Activities are used in the agile software process like a good tool. They are to be wielded by software craftsman who know the proper circumstances for their use. They are not utilized to create a production-line atmosphere for manufacturing software.

Characteristic 2: Iterative

Agile software processes acknowledge that we get things wrong before we get them right. Therefore, they focus on short cycles. Within each cycle, a certain set of activities is completed. These cycles will be started and completed in a matter of weeks. However, a single cycle (called an iteration) will probably not be enough to get the element 100% correct. Therefore, the short cycle is repeated many times to refine the deliverables.
Characteristic 3: Time-Bound

Iterations become the perfect unit for planning the software development project. We can set time limits (between one and six weeks is normal) on each iteration and schedule them accordingly. Chances are, we will not (unless the process contains very few activities) schedule all of the activities of our process in a single iteration. Instead, we will only attempt those activities necessary to achieve the goals set out at the beginning of the iteration. Functionality may be reduced or activities may be rescheduled if they cannot be completed within the allotted time period.

Characteristic 4: Parsimony

Agile processes are more than just a traditional software development process with some time constraints. Attempting to create impossible deadlines under a process not suited for rapid delivery puts the onus on the software developers. This leads to burn-out and poor quality. Instead, agile software processes focus on parsimony. That is, they require a minimal number of activities necessary to mitigate risks and achieve their goals. By minimizing the number of activities, they allow developers to deliver systems against an aggressive schedule, while maintaining some semblance of a normal life (sleeping at night ...).

Characteristic 5: Adaptive

During an iteration, new risks may be exposed which require some activities that were not planned. The agile process adapts the process to attack these new found risks. If the goal cannot be achieved using the activities planned during the iteration, new activities can be added to allow the goal to be reached. Similarly, activities may be discarded if the risks turn out to be ungrounded.

Characteristic 6: Incremental

An agile process does not try to build the entire system at once. Instead, it partitions the nontrivial system into increments which may be developed in parallel, at different times, and at different rates. We unit test each increment independently. When an increment is completed and tested, it is integrated into the system.

Characteristic 7: Convergent

Convergence states that we are actively attacking all of the risks worth attacking. As a result, the system becomes closer to the reality that we seek with each iteration. As risks are being proactively attacked, the system is being delivered in increments. We are doing everything within our power to ensure success in the most rapid fashion.

Characteristic 8: People-Oriented
Agile processes favor people over process and technology. They evolve through adaptation in an organic manner. Developers that are empowered raise their productivity, quality, and performance. After all, they are the best individuals in the organization to know how to make these changes.

**Characteristic 9: Collaborative**

Agile processes foster communication among team members. Communication is a vital part of any software development project. When a project is developed in pieces, understanding how the pieces fit together is vital to creating the finished product. There is more to integration than simple communication. Quickly integrating a large project while increments are being developed in parallel, requires collaboration.

2. Conclusion

Agile processes are not a new phenomenon. They are the evolution of the best practices which have been refined over the past thirty years. They are not a silver bullet which will cure your project of all of its ills or guarantee success. Successfully delivering a project requires hard work and the understanding of the potential pitfalls.

No single process will work for every project. Yet, most projects would agree that they would like to be more agile. Developing a more agile process for your project requires an understanding of the dynamics of these projects. It also requires a certain amount of common sense.

Today, there are many new and interesting developments in the area of agile processes. Many of today’s agile processes have developed some interesting innovations to become as lean as possible while maintaining the characteristics of agility. An example of such an innovation is pair programming. A better understanding of the dynamics of these processes should foster even further innovation.

3. Bibliography