



ASQ Software Division Position Statement Agile Methods for Software Projects

Position Statement

The ASQ Software Division provides the following position statement in the form of core propositions concerning the use of agile methods for software projects within the general context of the software's asset value and use in real world applications.

Proposition 1: A preferred approach for a software project is one the stakeholders feel supports their interests. Stakeholders include any entity receiving benefits or incurring costs from a software project. There are competing interests among stakeholders, and a balance of those interests is critical in selecting software engineering practices for a software project.

Proposition 2: The Agile Manifesto¹ prefers one value over another, but it does not excuse a project from appropriately addressing both sides of the balance to support the best interests of the project and its stakeholders. As such values and principles provide guidance for balancing each stakeholder's interests, they are considered of value to the software project.

Proposition 3: In the spirit of continual improvement, all software practices should be re-evaluated periodically as to their effectiveness and efficiency. Management and engineering practices will change over time as technologies and expectations change. Agile methods have evolved to include "extreme" implementations of good practices that may fit many of today's environments. Variants of good practice, including the agile practices, suit some business environments better than others, and may be combined in ways to increase the probability of project success. Philosophies such as those stated in the Agile Manifesto, as well as those underlying other methodologies, can be useful in supporting an organization's ongoing improvement efforts.

Rationale

A project and the organizational enterprise within which the project operates need to carefully assess the benefit of any software practice/method/technique to understand the problem solved by that practice and how the associated software project properties would be of most value to stakeholders. Software project properties are those observable quality characteristics such as project cost/duration, product capability, development/support effectiveness, cost of quality, marketability, reliability, supportability, and so forth. Stakeholders may have differing views about which software project properties are key and the value of those properties. Hence, "value" is a relative term depending on the stakeholder. The particular prioritization for any software project property and implementation practices is usually subjective and qualitative. In short:

- stakeholders have needs and expectations to be satisfied by projects beyond the capabilities of the product ultimately delivered, and those needs and expectations may change over the course of the projects;
- projects have properties (qualities, characteristics, or attributes) that must satisfy those stakeholder needs and expectations;
- methods used for conducting the work of a project contribute significantly to the project's properties; and
- value of a method is assigned according to its effectiveness and efficiency in contributing to the project's properties.

¹ <http://www.agilemanifesto.org/>



ASQ Software Division Position Statement Agile Methods for Software Projects

A variety of methods for implementing software projects have been developed over the last few decades. Adaptations of some of these methods have been labeled as agile. As knowledge of and reports of success with these practices spread, a philosophical framework called the Agile Manifesto was developed and promoted. This Manifesto presents a set of value propositions and principles for a software project. For a software practice/method to follow the Manifesto, the characteristics of the practice/method should align with the Manifesto values and principles. However, application of a specific agile method (e.g., Scrum) does not necessarily mean the software project can claim to be agile, or more specifically to align with the Manifesto values and principles. Some project practices may be aligned with specific Manifesto values and principles, but other practices may be in conflict with some of the values and principles. The more important concern is under what circumstances is any specific practice appropriate for a given project.

Evidence suggests selections of software project practices are being made based on perceptions of what being agile means rather than to solve specific real world problems. Claims of following the Agile Manifesto may not provide supporting evidence as to the extent that the software project practices solve the intended real world problem. Just claiming to be agile is not enough. It may be difficult for some organizations to determine whether a particular method is agile, aligns with the Manifesto, or is even effective and efficient for their software project. This is why a balance is needed in the selection of software practices and that balance may vary considerably from project to project – depending on the stakeholders needs and expected project properties and success factors.

Since the capability to respond to change is a key value of the Agile Manifesto it is important to remember that one original purpose of software was to resolve how to make changes in a product that would be more difficult (schedule/cost) if implemented in hardware. As the functional value of software has evolved, it has also been important to include “late” system changes that are best incorporated through changes to the software component. So, not only should a software project welcome the concept of change, but the software product should reflect the capability to be changed – within well-defined and balanced controls. In effect, one value of software projects is grounded in having project practices that facilitate efficient and effective change. The capability for the software component to adapt to change will vary considerably, however, particularly when there may be competing requirements.

What appears to be difficult to understand quantitatively, and to some extent even qualitatively, is just what it means to be agile. Phrases in the Agile Manifesto such as: value over, frequently, deliver, late in development, self-organizing, working software, working daily together, face-to-face, and early and continuous can have multiple meanings and value depending on the context. For any project practice to be agile requires both a philosophical attitude and a definitive time frame. What is agile today may not be in the near future as technology and expectations change. What is clear is that the Agile Manifesto values and principles do support a disciplined approach to software.

A preferred approach is to incorporate practices based on stakeholder needs and expectations, project properties, methods for conducting the project work, and the value such methods provide to the effectiveness and efficiency of the project’s properties within the context of system project success.