



The impact of agile methods on software project management

Name: Ali Hassan

Roll no: BSCS-F19-M49

Section: B

Domain Introduction:

This paper examines the impact of agile methods on the:

- people involved in a project
- the process under which a project is developed
- and on the project itself

in an attempt to allow project managers to evaluate the applicability using an agile method.

Literature review:

Year	Author	Method	Conclusion
<ul style="list-style-type: none"> • 2001 	<ul style="list-style-type: none"> • Highsmith, J. & A. Cockburn 	<ul style="list-style-type: none"> • Agile Software Development 	<p>Project managers should consider agile method usage for such projects that they have a team capable of using it and can implement the required processes. Otherwise, more traditional approaches may be more appropriate.</p>
<ul style="list-style-type: none"> • 2000 	<ul style="list-style-type: none"> • K. Beck 	<ul style="list-style-type: none"> • eXtreme Programming 	
<ul style="list-style-type: none"> • 1995 	<ul style="list-style-type: none"> • J. Stapleton 	<ul style="list-style-type: none"> • DSDM 	
<ul style="list-style-type: none"> • 2002 	<ul style="list-style-type: none"> • Boehm, B 	<ul style="list-style-type: none"> • Agile Methods, with Care 	
<ul style="list-style-type: none"> • 2002 	<ul style="list-style-type: none"> • DeMarco 	<ul style="list-style-type: none"> • Agile Methods Fray 	

Methodology:

Analysis:

As more and more software projects engage Agile Methods, there are emerging patterns of success and failure. With growing adoption of Agile Methods, project managers increasingly need to understand the applicability to their projects and factors that drive key project performance characteristics. While some organizations affirm that Agile Methods solve all their problems, few have shown consistent success over a range of typical software projects. Agile Methods have advantages, especially in accommodating change due to volatile requirements. However, they also present concomitant risks with managing the many dependent pieces of work distributed across a large project. Use of Agile Methods therefore presents a set of tradeoffs.

Software engineering, as a discipline, confronts two key challenges that separate it from other engineering disciplines. Software, a conceptual and often intangible product, changes and evolves at a much higher rate than integrated circuits or steel. While software is changeable, there is an increased cost the later in a project lifecycle the change occurs. This is true to a lesser degree in tangible products since measurable tests of the requirements and design can be more readily applied. Recognition of this fact has led to the emergence of a set of Agile Methods that embrace change and manage the related risks. Many such Agile Methods have been introduced over the last decade, including eXtreme Programming (XP), SCRUM, and Dynamic System Development Methodology (DSDM). While these methods differ in their specifics, they share a common goal of enabling teams to more rapidly respond to change. As changes are costly to accommodate later in

the project, the ability to respond rapidly to change reduces project risks and their costs. While Agile Methods are effective in some contexts, large and complex software products often require systematic discipline with the requisite process overhead to ensure success. The challenge for managers is to determine whether an Agile Method is appropriate for a given set of project activities. Concomitantly, addressing the risks involved with their use warrants attention. All methodologies have risks, and understanding those risks and finding ways to monitor, mitigate, and manage those risks is an important aspect of software project management.

Discussion:

A Brief Look at Agile Methods

Extreme Programming:

Perhaps the most recognizable Agile Method, eXtreme Programming (XP), has the overriding goal “to get the project at hand done.” No fan fare, no magic bullets – just apply a series of principles that work.

SCRUM:

SCRUM development involves several environmental and technical variables that are likely to change during the process. SCRUM concentrates on how teams can be organized to produce software in a constantly changing environment.

Dynamic System Development Method:

One key aspect that distinguishes the DSDM approach is that it fixes time and resources first and then adjusts the amount of functionality accordingly.

Agile Manifesto

The “Agile Manifesto” provides a good overview of the intent of Agile Methods. The following values express the tenor of the principles employed:

- individuals and interactions over process and tools
- working code over comprehensive documentation
- customer collaboration over contract negotiation
- responding to change over following a plan

In each of these values, the Manifesto is not saying that the second item is not important, just that it is less important than the first item.

Impact on Project Management

While the ideas and intent behind Agile Methods are by and large good, they have impacts on the people, process, and project elements of an effort. We examine some of these impacts to determine whether an Agile Method can and should be applied to a project, given its requirements, available staff, and external factors such as business and legal constraints.

People:

There can be a range of people involved in a software effort – developers, testers, project leaders to name a few. There is often a customer and an end user who wants the resulting product. There are also executive managers (business executives and directors of the development shop), who are interested in budgets and returns on investment, and human resources. Each of these has a stake in an agile project.

Developers:

Perhaps the largest impact of Agile Methods is on the Developers. Agile Methods depend on strong developers – they must be amicable, talented, skilled, and able to communicate well. Developers must be willing to work as a team, able to handle constant change, and resourceful enough to solve problems. Agile Methods are very lightweight methods, not affording strict guidelines and processes for developers to follow. Hence, they do not accommodate weaker developers well. Yet, skilled technology workers are often a rare commodity. This is a management risk as some developers may not fit in this Agile environment.

Testers:

The impact of using an Agile Method on the testing (or quality assurance) organization hinges on the shorter development cycles where testing occurs throughout the development process. Testers must work closely with the developers as code is being written. Testers may need to be more capable as programmers to automate their system and functional tests and incorporate them into the automated testing framework. This may represent a different skill set.

Project Leaders:

Since Agile teams involve experienced staff with sizeable responsibility, a mentor or coach leadership approach is most effective. Team leads must be willing to enable members to take initiative. Leadership is done via collaboration rather than command and control type leadership. This can represent a cultural shift for some as they must be willing to share decision making authority. The job of a team lead is to facilitate the team into making decisions. In contrast, project managers in agile processes are responsible for tracking progress and making business decisions.

Customers:

The impact of Agile Methods is to have customers much more involved than usual Methods. Customers in more traditional methodologies may be involved at the inception of the project – helping define requirements and contractual obligations – and at the end of the project with alpha, beta, and acceptance testing. Customers in Agile Methods are instead involved much more frequently and with more influence. Many Agile Methods assume, or at least highly recommend, a full-time customer presence on site working directly with the development organization. Finding a customer willing to be this involved can be difficult.

The Team:

Since Agile Methods rely substantially on collaboration and communication, the team is key for success. A single strong-willed developer, developers who do not work well together, a customer who doesn't engage with the team, each could destroy the collaborative nature of a group. The team chemistry is of represents a significant risk for the Agile project.

Process:

Since Agile Methods represent a new principles, processes activities, and sub-goals, they have an impact on many of an organization's processes. Old processes (e.g., planning, development, delivery, operations) must be replaced by agile ones. Cultural shifts in the organization towards Agile Methods turn old ways of thinking on their end, inducing resistance.

Planning:

Agile processes are characterized by placing less emphasis on formal planning. This is not to say that planning does not occur. Agile planning is a relatively informal process. For example, deciding what will go into each time-box is accomplished through the daily SCRUM meeting by discussing pending problems, prioritizing work, and assigning resources to the problems. In other Agile methods, even this level of planning may not be considered. It is important to factor in informality.

Documentation:

In Agile Methods, documentation is sparse – often limited to source code. With Agile processes, information is communicated informally and is simply kept as part of the collective knowledge of the organization. While reducing the amount of documentation can increase productivity, it does come at some risk and cost. From a business perspective, documents form the basis for audits assuring proper quality procedures are followed. Documentation serves as a domain knowledge repository. If the organization changes dramatically, this knowledge can be lost.

Development Processes:

Agile processes often encourage principles that dramatically change the process. While many of these are not limited to Agile Methods, Agile development encourages if not require their usage. Key development processes of interest are refactoring, minimalist development, code reviews, and continuous integration.

Refactoring is the process of taking code and improving it without losing any functionality.

Minimalist development within the Agile Methods community it is known as the YAGNI precept – an acronym for “You Aren’t Going to Need It.” Under YAGNI, features not needed for the current functional product are stripped out to keep the implementation simple.

Code reviews are the process whereby one or more developers examines the code written by another.

Continuous integration is the process whereby the system is tested often, usually nightly if not even more frequently. Developers integrate their code into a baseline and run a set of regression tests on it.

Many development groups already practice these principles. However, many developers are prickly about the notions of peer programming and may chafe at having to write a significant number of tests. It may be necessary for the project manager to incorporate these processes slowly and with incentives to increase the chances of their acceptance.

Project:

Different types of projects are more suited for Agile Methods than others. Business factors may prevent the utilization of Agile Methods. And there are several project characteristics that reduce the effectiveness and applicability of using Agile methods.

Project Types:

Agile Methods are most applicable to projects where requirements are ill-defined and fluid since they seek to accommodate change easily.

Business Factors:

A key business factor affecting the appropriateness of Agile Methods is contractual obligation. For many contracting companies, what is to be performed by the contractors is determined by a statement of work defining key requirements and tasking. If the requirements for work to be performed are part of a legal contract, an Agile Method may be inappropriate since requirements are malleable.

Conclusion

Agile Methods offer a reasonable approach for the high degree of change and uncertainty in today's software development. There are proven principles employed in Agile Methods that, when applied singularly under the right circumstances, result in lower risk projects and ultimately better productivity and quality (e.g., smaller teams result in lower risks due to the better communications). Additionally, when these are combined with other agile principles, there can be a synergy that provides even more traction on the project goals (e.g., small teams and pair programming result in fewer errors and less rework).

However, Agile Methods are not appropriate for all projects. A project manager must consider the characteristics of the project to ensure that an Agile Method is appropriate. The impact on the people, the process, and the project must all be considered.

Agile Methods offer software project managers an alternative development and management methodology that provides good support for projects with ill-defined or rapidly changing requirements. Even on project that are questionable for the application of the entire Agile Method, underlying agile principles may still be effective. Project managers should consider its usage for such projects assuming that they have a team capable of using it and can implement the required processes. Otherwise, more traditional approaches may be more appropriate.