International Journal of Information and Computation Technology.
ISSN 0974-2239 Volume 4, Number 5 (2014), pp. 499-506

http://www. irphouse.com /ijict.htm

Amalgamation of Six Sigma and Agile

Karan Wadhawan, Twinkle Sharma and Sujata

B.Tech (Information Technology), ITM University, Gurgaon, India.
B.Tech (Information Technology), Assistant Professor (CSE/IT Dept.), ITM University, Gurgaon, India.

Abstract

Not understanding customers' real needs is the most common reason for failure of a software project. Requirement errors are the most expensive to fix – these errors easily consume 25 percent to 40 percent of a project's budget. The Agile approach has made it possible to work closely with the customer, understand the requirements and implement the functionality progressively and in collaboration with all the stakeholders. So, Agile methodologies address the problem by using an iterative and incremental approach. At the same time, one of the primary objectives of Six Sigma is to align business goals with the requirements of customers. The DMAIC (Define, Measure, Analyze, Improve, Control) phases are focused towards the expectations of the customers and provide valuable tools to address these. Two popular agile methodologies, Scrum and Extreme Programming, have been considered in the proposed paper. This paper addresses the most common issues in agile software development.

Keywords: Six Sigma, Agile, Lean, DFSS, CTQ.

1. Introduction

This paper has the following structure; section 1 includes the basic introduction about Six Sigma and agile methodologies. In section 2 a comparison between Six Sigma-Agile and other conventional approaches has been done. Section 3 is all about Agile methodology and Section 4 is about Six Sigma. Section 5 includes the mixture of Six Sigma and Agile approaches to produce better results which is the main motive of this paper. Section 6 tells how Six Sigma tools can be used with agile software projects. Section 7 is the case study of how Agile-Six Sigma is used in educational institutions.

Agile literature provides surprises for software measurement professionals: for example, time-consuming size judgment approaches. Though, dimensions for agile software development are readily available than in traditional environments. We use Six Sigma transfer functions to map development project controls onto process responses. Agile software development methodologies treat software development as knowledge acquisition by accepting change. This facilitates measuring software processes for performance and quality.

Six Sigma orientates products and services towards quantifiable customer values. It provides tools and methods to eliminate and prevent defects whereby anything which diminishes customer values is termed as a defect. A Six Sigma approach for Software has not concerned with bugs found by engineers. Non-adherence to specification for instance does not automatically imply that customers see this as a defect. Specifications may be wrong, or may not be of any value to a customer, and therefore non-adherence to specification won't be seen as a defect. This makes defect-counting difficult, especially in software, as only the customer value counts. Weaknesses that engineers classify as minor may end up as major defects in customer's perception, and vice versa. Customers might ignore major technical weaknesses of software if it does not affect business. However, in traditional software development models, the customer viewpoint is not readily available to the development team. To make processes measurable for Six Sigma, various views of stakeholders such as engineers, customers, users, etc. have to be taken into account. In Six Sigma this is accomplished via Transfer Functions, mapping process controls onto the process response. In software development, transfer functions consist of measurable software engineering practices.

2. Six Sigma and Agile V/S Conventional Approach

To address the challenges posed by the traditional methods, a lightweight methodology called "Agile" was developed more than a decade ago. It includes XP, Scrum and other feature-driven development methods. Agile methods try to overcome the limitations of the dynamic nature of systems development projects.

One key innovation of Six Sigma involves the absolute "professionalizing" of quality management functions. Prior to Six Sigma, quality management in practice was largely delegated to the production floor and to statisticians in a separate quality department. Formal Six Sigma programs adopt a kind of elite ranking terminology (similar to some martial arts systems, like Kung-Fu and Judo) to define a hierarchy (and special career path) that kicks across all business functions and levels. One of the biggest benefits of Six Sigma is that it brings discipline to the use of facts and measures to guide significant and predictable results.

3. Agile Software Development

Dissatisfaction with the software design methods of the 1980s and 1990s led to the creation of agile methods. These methods

- Focus on the code rather than the design
- Based on iterative and incremental development (IID) approach.
- Are intended to deliver working software quickly and evolve this quickly to meet changing requirements.

The aim of agile methods is to reduce overheads in the software process (e.g. by limiting documentation) and to be able to respond quickly to changing requirements without excessive rework.

The word Agile means "marked by ready ability and to move with quick easy grace and having a quick resourceful and adaptable character". Specification, design, implementation and testing are inter-leaved and the outputs from the development process are decided through a process of negotiation during the software development process. The main principles of agile methods are:

1. Customer involvement: Customers should be closely involved throughout the development process. Their role is provide and prioritize new system requirements and to evaluate the iterations of the system.
2. Incremental delivery: The software is developed in increments with the customer specifying the requirements to be included in each increment.
3. People not process: The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes.
4. Embrace change: Expect the system requirements to change and so design the system to accommodate these changes.
5. Maintain simplicity: Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

The Scrum approach is a general agile method but its focus is on managing iterative development rather than specific agile practices. Scrum projects are split into iterations (sprints).

Extreme Programming (XP) takes an 'extreme' approach to iterative development. Incremental development is supported through small, frequent system releases.

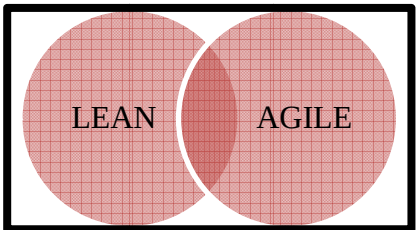

Fig. 1: Similarities and differences between Lean and Agile [13]

- Lean was conceptualized much before Agile
- Lean has a wider scope and can be applied to any industry/domain. Agile was conceptualized specifically for software development.
- Agile uses quite a few techniques/toolkits prescribed by lean

4. Six Sigma Methodology

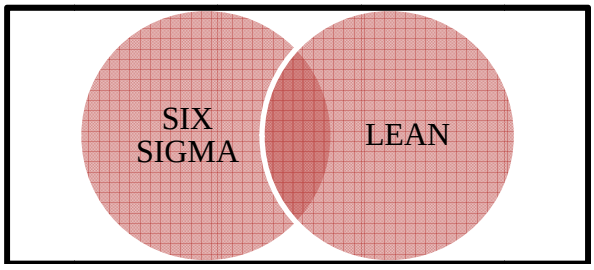

Fig. 2: Six Sigma and Lean- Natural Partners [12]

Six Sigma simply means a measure of quality that strives for near perfection. Six Sigma is a disciplined, data-driven approach and methodology for eliminating defects (driving toward six standard deviations between the mean and the nearest specification limit) in any process – from manufacturing to transactional and from product to service[4].

Six Sigma is a set of tools and techniques/strategies for process. It seeks to improve the quality of process outputs by identifying and removing the causes of defects(errors) and minimizing variability(statistical dispersion) in manufacturing and business processes. It uses a set of quality management methods, including statistical methods, and creates a special infrastructure of people within the organization ("Champions", "Black Belts", "Green Belts", "Yellow Belts", etc.) who are experts in the methods. Each Six Sigma project carried out within an organization follows a defined sequence of steps and has quantified value targets, for example; process cycle time reduction, customer satisfaction, reduction in pollution, cost reduction and/or profit increase[10].

DMAIC (Define Measure Analyze Improve Control) is used for projects aimed at improving an existing business process.

DMADV (Define Measure Analyze Design Verify) is used for projects aimed at creating new product or process designs. It is also known as DFSS(Design For Six Sigma).

Table 1: Relationship between Six Sigma and lean

Six Sigma	Lean
Costumer Focus: defects and business growth	Improved Speed and flow
Performance: thinking statistically and optimization and control	Removing waste: excess inventory, wasted time and motion and constraints
System development of infrastructure: belts, champions, leadership engagement	Reducing complexity and Increased flexibility.

5. Six Sigma(DFSS) Used With Agile

Define - understands the target environment, users, stated and latent requirements, and important result measures. In an Agile world, DFSS would drive for the best global understanding of those things possible on the first iteration along with building on the learning of the next time around. One aspect of most Six Sigma tools is the way the data is captured, distilled and stored, making it easy for a team to revisit and build on its thinking during a subsequent cycle.

Measure -includes tools for prioritizing requirements which can help focus effort on the ones most suitable for iteration. Six Sigma guidance can help any software team pay attention to what matters most to customers and the business. That is in line with Agile principles. Here analysis of the CTQ's takes place. CTQ refers to characteristics that are **C**ritical **T**o **Q**uality, product capabilities, production process capability, and risks.

Analyze - looks at solution choices at the appropriate level(s). DFSS would underline the importance of having an overall architecture and implementation plan as a backdrop and principle for iterations. This is recommended in the Agile community as well – where DFSS tools for quickly appraising iteration-level learning about the architecture, design/feature set, or implementation could support Agile's quick but informed thinking. Predicting cost, schedule and performance, even for an iteration, is an important part of the DFSS-driven dialogue with the business. Teams that iterate get lots of practice forecasting and comparing predictions with actuals.

Design/Build -constructs the capabilities appropriate for the current iteration. What is meant by "tracking product and process performance" can be auto-sized based on the risk and opportunity at hand for iteration. At any level of scope or scale, DFSS still offers efficient approaches and tools for managing those aspects of a team's progress during this stage.

Verify -checks product and development process performance as appropriate. DFSS provides some effective tools for testing and measuring technical and business results. As an incremental delivery is made at this stage, the target environment is changed as now it includes the cumulative incremental deliveries. New learning in the next Define phase can drive the next iteration, with each DFSS stage posing and answering questions using the data and scope appropriate for that cycle [5].

Some common messages are beginning to emerge from several software-relevant areas – Six Sigma for software, Agile development and Lean thinking. The links between Design for Six Sigma (DFSS) and Agile have been explored recently, but now a broader view yet can illustrate the way that Lean thinking, evolved from just-in-time manufacturing, aligns well with both Agile and software Six Sigma.

Table 2: Relationship between Six Sigma and Scrum roles [6]

Six sigma	Scrum
Champion	Product owner
Black belt	Scrum master
Green belt	Team member

6. Using Six Sigma Tools in an Agile Software Project

It may be counter-intuitive to consider using Six Sigma tools in an agile software project. Six Sigma's genesis was in the manufacturing world where one of its primary goals has been to reduce process variation. On the other hand, Agile software development is built on the premise that complex software projects, unlike manufacturing, cannot be successful in an environment using defined process control. Instead, Agile development supports empirical process control that relies on continuous inspection and adaptation of the process used.

Six Sigma offers a groundbreaking way of reducing defects in the end product – something that any software project can definitely use. Hence, it makes a lot of sense to at least evaluate the possibilities of integrating both processes.

Agile software methodologies, based on the principles of the Agile Manifesto, are typically adaptive processes that provide a process framework. This framework guides a development team to build software using a process adapted to suit the domain of the project. This approach actually opens up possibilities of using appropriate tools, including tools of Six Sigma that can help improve the quality of the product.

There are many areas of software development that can benefit from Six Sigma. The fact is, there is a low-barrier approach to deriving benefits of Six Sigma, in a controlled way, while following an Agile software development methodology.

Before going any further, it is important to acknowledge that Six Sigma is a major initiative usually involving the entire organization. The scope of a Six Sigma project often is much wider than a single software project. Therefore, this discussion will be limited to only certain aspects of Six Sigma that can be implemented relatively easily and provide maximum return on investment [7].

7. Agile and Six Sigma in Educational Institutions

Education sector which is considered to be most one of the most dynamic yet inefficient and unsteady business today needs a tool to create, monitor and improve quality of its each deliverable and delivery processes. This necessity of the education sector can be very well fulfilled by the inception of Agile and Six-Sigma concept in education sector. Since teaching is a learning practice; it has to be updated as the new technologies arrive in market to enhance the knowledge for the development of both teachers as well as students. In today's world of globalization Quality has taken a centre stage due to the continuous competition among institutions, emergence of new technologies and the knowledge driven economy. Here institutions act as business unit while qualified students are outputs who are either consumed by society and / or industry as end customers. And in quality centric world it is of prime importance that the product i.e. the students meet high quality standards for society and industry or company jobs. Implementation of Agile and Six-Sigma can help establish this confidence and accelerate the value creation process [11].

Six sigma factors

- Timely processes of critical work like timely issuance of results
- Acquiring quality students as well as faculty members
- Improving teaching techniques to impart quality education in efficient and timely manner
- Reducing waiting time waste for activities such as form submissions, fee payments
- Benchmarking institutional performance vis-a-vis other institutions
- Establishing procedures and eco-system which will help students to acquire specific qualities of their interest while fulfilling social and industrial criterions [11].

Agile factors

- It is difficult for the inexperienced students to detect mistakes made in early stages before the development process reaches last stages. Finally, as in the planning of the scope and the project milestones some students do not take part in; it provokes a lack of commitment in the rest of them by giving changing surroundings.
- Scrum enables the students to have a better teamwork environment and a better communication that results in high-quality products.

8. Conclusion

In an agile paradigm, every aspect of development like requirements, design is continually revisited throughout the lifecycle. When a team stops and re-evaluates the direction of a project every two weeks, there's always time to steer it in another direction. The results of this "inspect-and-adapt" approach to development greatly reduce both development costs and time to market. Agile empowers teams to continuously re-plan their release to optimize its value throughout development, allowing them to be as competitive as possible in the marketplace. The central idea behind Six Sigma is that if "defects" can be measured in a process, one can systematically Fig. out how to eliminate them and get as close to "zero defects" as possible. It's a management approach that aims to achieve the apex of quality by measuring, analyzing, improving, and controlling processes to root out defects and boost bottom-line results.

This leads to the conclusion that Six Sigma tools and techniques, if used in conjunction with the Agile process, can enhance the effectiveness of gleaning the real needs out of customers.

References

[1] Dr. Thomas M. Fehlmann ,Six Sigma for Agile Teams, Euro Project Office, Zurich

[2] Álvaro Soria Marcelo R. Campo Guillermo Rodríguez, Improving Software Engineering Teaching by Introducing Agile Management, ISISTAN Research Institute, UNICEN University
[3] Yu Beng Leau , Wooi Khong Loo, Wai Yip Tham and Soo Fun Tan , Software Development Life Cycle AGILE vs Traditional Approaches, School of Engineering and Information Technology Universiti Malaysia Sabah, Malaysia
[4] http://www.isixsigma.com/new-to-six-sigma/getting-started/what-six-sigma/
[5] http://www.isixsigma.com/new-to-six-sigma/design-for-six-sigma-dfss/dfss-agile-development/
[6] http://www.scrumalliance.org/community/articles/2010/february/can-scrum-support-six-sigma
[7] http://www.isixsigma.com/industries/software-it/using-six-sigma-tools-agile-software-project/
[8] http://www.isixsigma.com/industries/software-it/integrating-lean-six-sigma-agile-and-scrum/
[9] http://www.asq.org/edu/2012/05/six-sigma
[10] http://en.wikipedia.org/wiki/Six_Sigma
[11] http://www.knewron.co.in/Writeups.asp?ID=11
[12] http://www.isixsigma.com/methodology/lean-methodology/software-development-convergence-six-sigma-lean-agile/
[13] http://www.processexcellencenetwork.com/lean/articles/using-lean-in-agile-software-development-a-compari/