



Drive for Speed, Quality, and Lower Cost AGILE & LEAN Explained

1.0 INTRODUCTION

The purpose of this paper is to provide a short explanation of the most widely adopted approaches to product development and process optimization, while touching on how they relate to each other. The scope of this document includes WATERFALL, AGILE, LEAN, SIX SIGMA, and GE-WORKOUT.

As a start, it is important to understand that both Waterfall and Agile strive to improve a product, while Six Sigma and LEAN strive to improve the process that delivers the product. However, LEAN and Six Sigma have been applied to improve the Waterfall and Agile processes to reduce defects (improve quality) and remove waste (increase speed and cost). To add to the confusion, 'LEAN Six Sigma' is also a term used as both approaches can be applied simultaneously bringing together the data-driven methods of Six Sigma with the waste-cutting tools of LEAN.

Finally, the GE's Workout (GE-Workout) process introduced by General Electric (GE), can be considered as a pocket size LEAN approach looking to deliver results in 90 days. GE-Workout can also be an effective tool to use within the other approaches (e.g. WATERFALL).

2.0 'PROCESS IMPROVEMENT' APPROACHES

2.1 SIX SIGMA – 'Defect Reduction' Mindset

The Six Sigma focus is on eliminating defects, thereby improving quality to the customer and increasing process efficiency. While it was first used in manufacturing, it was quickly discovered that Six Sigma could benefit any business process, from customer support to closing the books. There are five steps to the Six Sigma approach – (DMAIC) – Define, Measure, Analyze, Improve, and Control.

Six Sigma is a very data-focused approach, requiring the performance of both the overall SIGMA capability as well as the driving process sub-measures (Y's & X's) to be baselined before analyzing and making any improvements to the process itself. Being so data demanding, Six Sigma has been perceived as perhaps too heavy and slow to achieve results, even intimidating to the average person, and with this LEAN has been seen as a friendlier and faster alternative.

2.2 LEAN – 'Remove Waste' Mindset

LEAN or LEAN MANAGEMENT is focused on continuous improvement, maximizing customer value while minimizing waste. LEAN has three key pillars. First, delivering value from the customer's perspective. Second, eliminating waste (also from the customer's perspective). And third, continuously improving the processes to better serve customers.

There are five LEAN principles, which include the "Value Stream Map", which is basically mapping out the process and resources and identifying steps that do not have value. Another principle is the "Pull-Based System" that focuses on limiting inventory and work-in-process items while ensuring that the requisite materials and information are available for a smooth flow of work. The fifth and final principal is "Pursue Perfection", making LEAN thinking and continuous process improvement a part of the organizational culture.

LEAN has frequently been associated with the concepts of CYCLE-TIME and LEAD-TIME. The first measures the average speed from when a process starts to when the process delivers a defect free item. The latter measures the average time for when a customer requests an item to when it is delivered defect free. So, Lead-Time depends on Cycle-Time, but is also linked to the organization's willingness to keep a backlog, to the customer's patience, and the customer's readiness for delivery.

The practicality of LEAN and the ability to cherry-pick the parts over the whole have made it more friendly and useful to adopt in an organization. For example, the value stream mapping can be done in isolation and deliver tremendous results. CYCLE-TIME has often been embraced to create an organization's focus on removing defects, removing waste, and seeking perfection.



2.3 GE WORKOUT – ‘Remove Unnecessary Work’ Mindset

GE’s Workout (GE-Workout) is a concentrated effort of bringing the people nearest a process together and have them focus on identifying improvements, outlining the proposed implementation, the accountability, and the measurable results. The improvements should be implementable within 90-120 days and are prioritized based on speed/ease versus pay-off. While the spotlight is on streamlining the process by removing unnecessary work, the Workout team can identify and recommend other improvements; indeed, the GE-Workout sponsor may direct the team to target on a particular problem or opportunity. GE-Workout can be considered a pocket-size LEAN approach, as it focuses similarly on process improvement, waste removal, and measurement; however, LEAN has more rigor and is not limited to the GE-Workout’s short-term focus.

3.0 ‘PRODUCT DEVELOPMENT’ APPROACHES

3.1 WATERFALL – ‘Batch’ Mindset

This is a traditional approach to product development. The stages can be summarized as – requirements, design, implementation, verification, and maintenance. The development follows a linear approach, where each stage generally needs to fully finish before the next one can begin, usually with a milestone acceptance in between. Such batch like stages means for example that all the requirements are gathered within the first stage, signed off in an acceptance milestone, before next phase of design can start.

With the need to accelerate product capability quicker to market, Waterfall’s linear, batch process was proving too slow to market. Rather than taking all requirements of a product through each step of the Waterfall process supported by a large team, it was more appealing to define and deliver smaller subsets of requirements in a quick iterative process by small empowered teams. The market required it and the technology could support it. With this, the Agile approach was created.

Waterfall is still pervasive and is used mostly today for larger projects with high demands for low defects, where change is less frequent, and controlling the requirements is critical to the business. Examples include flight traffic control systems, a healthcare system, a banking system, or a military vehicle platform.

3.2 AGILE – ‘Fast Iterative’ Mindset

The ‘Agile’ term is most frequently referred to when considering the approach to software development. Ultimately, Agile is a mindset informed by the values and principles contained in the “Agile Manifesto.” These values and principles provide guidance on how to create and respond to change and uncertainty with optimal speed to customer value.

The Agile approach sits on a rapid iterative development process, where requirements and solutions evolve through the collaboration within a self-organizing, empowered and cross-functional team. The team has a ‘Product Owner’ who has full empowerment to prioritize and make decisions. The Agile process encourages frequent inspection and adaptation, encourages teamwork, self-organization and accountability. It also has a set of engineering best practices intended to allow for rapid delivery of high-quality software, and aligns development with customer value.

Agile’s set of values and principle are followed in project management processes or methodologies such as ‘Scrum’ and ‘Kanban’. When a project is so large that requires many Scrum teams to be coordinated in order to achieve a shared goal, a framework such as SAFe can be adopted. Projects are broken down into ‘Sprints’ or iterations.

At the beginning of each Sprint, the cross-functional Scrum team has a planning meeting, where the highest priority backlog items are reviewed. Next the team discusses the requirements, it makes Sprint commitments and then agrees to a delivery strategy. The team meets daily in the morning for less than fifteen minutes, in front of the task board, to review progress, discuss deliverables for the day, and raise any risks to delivery.

At the end of each Sprint, a review is performed to demonstrate the Sprint deliverables and assess the performance of the team against the pre-agreed Sprint goals. As part of Agile’s principle of continuous improvement, the team



performs a retrospective to review mistakes/inefficiencies and agrees on ways to make things better for future Sprints.

3.3 LEAN in Software Development

LEAN has also been used to work on improving the process of software development. First, by applying LEAN to the existing processes for reducing waste. The 12th principle of the Agile Manifesto, “team reflects on how to become more effective,” encourages this application of LEAN. Second, an actual creation of ‘LEAN development principles’ (see appendix). Third and last, the creation of ‘LEAN Software Development’ (LSD), which is an Agile framework based on optimizing development time and resources, eliminating waste, and ultimately delivering only what the product needs. The LSD ‘Minimum Viable Product’ strategy, in which a team releases a bare-minimum version of its product to the market, learns from users what they like, do not like and want to be added, and then iterates.

3.4 Agile in Business

As there is an ever-increasing need to be adaptable, many businesses are pushing themselves to be more agile and by doing that, they are adopting many of the principles in the software’s Agile Manifesto. According to McKinsey & Company, a US-based management consulting firm, organizations are going from hierarchical machines to flexible organisms.

To broaden adoption of Agile principles, there must be a shift to new mindsets and behaviors in the business team. Over time, as mindsets change, individuals will adopt more Agile behaviors, such as:

- Breaking down project work, or initiative-based tasks, into iterations (i.e. sprints),
- Implementing empowered, cross-functional, self-organized and collaborative teams,
- Setting and working in fast iterations with agreed goals,
- Scheduling daily 15-minute check-ins (instead of holding lengthy staff meetings),
- Seeking excellence through self-reflection and continuous improvement.

This mindset shift into empowered, cross-functional, self-organized, and collaborative teams has been proven to be successful in many fast-moving product companies such as Google, Proctor & Gamble, and 3M.

4.0 LEAN and Agile Compared

4.1 Commonality Between Approaches

While LEAN is founded on improving a process and Agile on improving a product, the following are common themes in the LEAN and Agile approaches:

1. Working toward a vision and values with much autonomy thereafter,
2. Pushing the power down to the people nearest the process and work,
3. Requiring cross-functional teams to meet regularly face-to-face and work through the opportunity,
4. Encouraging a self-organizing spirit of collaboration, critical thinking, creativity, open to feedback and good communication,
5. Breaking the problem into smaller parts so that each can be owned, measured, and implemented,
6. Measuring performance through quality of output,
7. Seeking excellence through self-reflection and continuous improvement,
8. Boosting cycle-time performance is critical – leads to repeatability and removal of waste.

Key differences between the LEAN and Agile approaches:

- a. LEAN is not held to Agile’s fast iterations,
- b. LEAN is obsessed with waste; Agile with user experience,
- c. LEAN delivers value; Agile delivers a working product.



4.2 Commonality of People Skills

With Agile, each team member meets every day with their peers, share their work, prioritize efforts, be required to listen, be open to ideas, negotiate win-win agreements, and be ready to be accountable to meet the team's goals. Team members must handle this new fast pace of work where the people interactions are compressed. Further, because of today's pandemic, more interactions are likely to be remote. Often, an Agile team has to collaborate with other Agile teams to coordinate their efforts to achieve a larger business goal.

Agile methods (e.g. Scrum) provide the process by which Agile can be followed successfully. Such important 'process' skills must be learnt by the team members. However, the 'behavioral' skills (e.g. being open to feedback) are just as important to the team's success. Indeed, as well as high motivation the following 'behavioral' skills are critical:

Self-directing to a vision
Being open to feedback
Transcending fear

Effective collaboration
Critical thinking
Championing data

Intelligent communication
Being creative in a team
Influence and negotiate positively

For this reason, the ongoing need to invest in people is extremely important to improve both their 'behavioral' skills as well as their Agile 'process' skills. Not just skilling them up to collaborate and communicate with people in the same location, but developing them to build strong relationships with team members remotely around the world in an environment where change and speed are a priority.



A1 - APPENDIX - PRINCIPLES

A1.1 AGILE SOFTWARE DEVELOPMENT PRINCIPLES

#		PRINCIPLE
1	Satisfy the Customer	Our highest priority is to <u>satisfy the customer</u> through early and continuous delivery of valuable software.
2	Harness Change	Agile processes <u>harness change</u> for the customer's competitive advantage.
3	Deliver Frequently	<u>Deliver working software frequently</u> , from a couple of weeks to a couple of months. The sooner you deliver incremental software, the faster the feedback and the faster you can identify a wrong turn.
4	Work Together	Business people and developers (cross-functional members) <u>must work together</u> daily throughout the project.
5	Build Projects	<u>Build projects around motivated individuals</u> . Give them the environment and support they need, and trust them to get the job done.
6	Face-To-Face Conversation	The most efficient and effective method of conveying information to and within a development team is <u>face-to-face conversation</u> .
7	Progress = Working Software	<u>Working software is the primary measure of progress</u> .
8	Sustainable Development	Agile processes <u>promote sustainable development</u> . The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9	Continuous Attention to Excellence	<u>Continuous attention to technical excellence</u> and good design enhances agility.
10	Keep It Simple	Simplicity – the art of maximizing the amount of work not done – is essential.
11	Self-Organized Teams	The best architectures, requirements, and designs emerge from <u>self-organizing teams</u> .
12	Reflect for Effectiveness	At regular intervals, the <u>team reflects on how to become more effective</u> , then tunes and adjusts its behavior accordingly.

A1.2 LEAN SOFTWARE DEVELOPMENT PRINCIPLES

#		PRINCIPLE
1	Eliminate Waste	Eliminate anything that does not add value to the customer.
2	Build in Quality	Create solutions to quality aiming for no chance of defects.
3	Create Knowledge	Provide the infrastructure to properly document and retain valuable learning.
4	Defer Commitment	Keep options open and continuously collecting information, rather than making decisions quickly with less data.
5	Deliver Fast	Deliver quickly to put value into the hands of the customer as quickly as possible.
6	Respect People	Good leaders that make sure that teams have an environment that fosters engaged, thinking people focused on creating a great product.
7	Optimize the Whole	Optimizes the whole value stream.