ORCHESTRATE
Solutions for higher performance!

AGILE VS LEAN

THE COMPREHENSIVE FACTORS

Introduction

Introduction

Industries across the globe are burgeoning. Stiff competition has permeated every stratum among enterprises. To sustain themselves in such an environment, companies are seeking new and improved methods by which they can revamp their business and also their existing production processes.

With the emphasis firmly resting on the requirement for more robust processes, companies are transforming their project plans drastically. Now, the buzz and objective is to move on to a more adaptive process that ushers in change and provides results. Moreover, businesses need a process that offers enhanced flexibility which can alter the very nature of the process itself.

That much-sought-after process encompasses practicing Lean Startup with the Agile Methodology, in addition to the traditional project management focus on controlling activities. Quite simply: Lean is used to help build and define a marketable product. Agile is the means to achieve this in software development.

Agile and Lean principles

Agile Manifesto - 12 principles

1 Customer satisfaction by rapid delivery of useful software.

2 Welcome changing requirements, even late in development.

3 Working software is delivered frequently (weeks rather than months).

4 Close, daily cooperation between business people and developers.

5 Projects are built around motivated individuals, who should be trusted.

6 Face-to-face conversation is the best form of communication (co-location).

Agile and Lean principles

7 Working software is the principal measure of progress.

8 Sustainable development, able to maintain a constant pace.

9 Continuous attention to technical excellence and good design.

10 Simplicity—the art of maximizing the amount of work not done—is essential.

11 Self-organizing teams.

12 Regular adaptation to changing circumstance.

Agile and Lean principles

Lean principles

Lean development can be summarized by seven principles, very close in concept to lean manufacturing principles

1. Eliminate waste
2. Amplify learning
3. Decide as late as possible
4. Deliver as fast as possible
5. Empower the team
6. Build quality in
7. See the whole

Agile

Agile

In February 2001, a group of 17 software developers met at the Snowbird resort in Utah to discuss lightweight development methods. Their brainstorming led to a revolutionary set of ideas that were later published under the title: “Manifesto for Agile Software Development”.

Four guiding values of the Agile Manifesto:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- ✓ Individuals and interactions over processes and tools
- ✓ Working software over comprehensive documentation
- ✓ Customer collaboration over contract negotiation
- ✓ Responding to change over following a plan

Agile and its Importance

The Agile software development method allows for many possibilities to evaluate the direction throughout the development lifecycle.

This is achieved through regular cadences of work, known as Sprints or iterations, by the culmination of which teams must offer a shippable product increment.

In the near-obsolete waterfall method, development teams get only one opportunity to ensure every single aspect of the project has gone according to plan.

However, the opposite holds true for the agile paradigm where all aspects of development, ranging from requirements, design and so on are revisited often.

Consider this: A team gains the capacity to channelize their project in an entirely different direction if they have the benefit of re-evaluating their competencies bi-weekly.

Agile

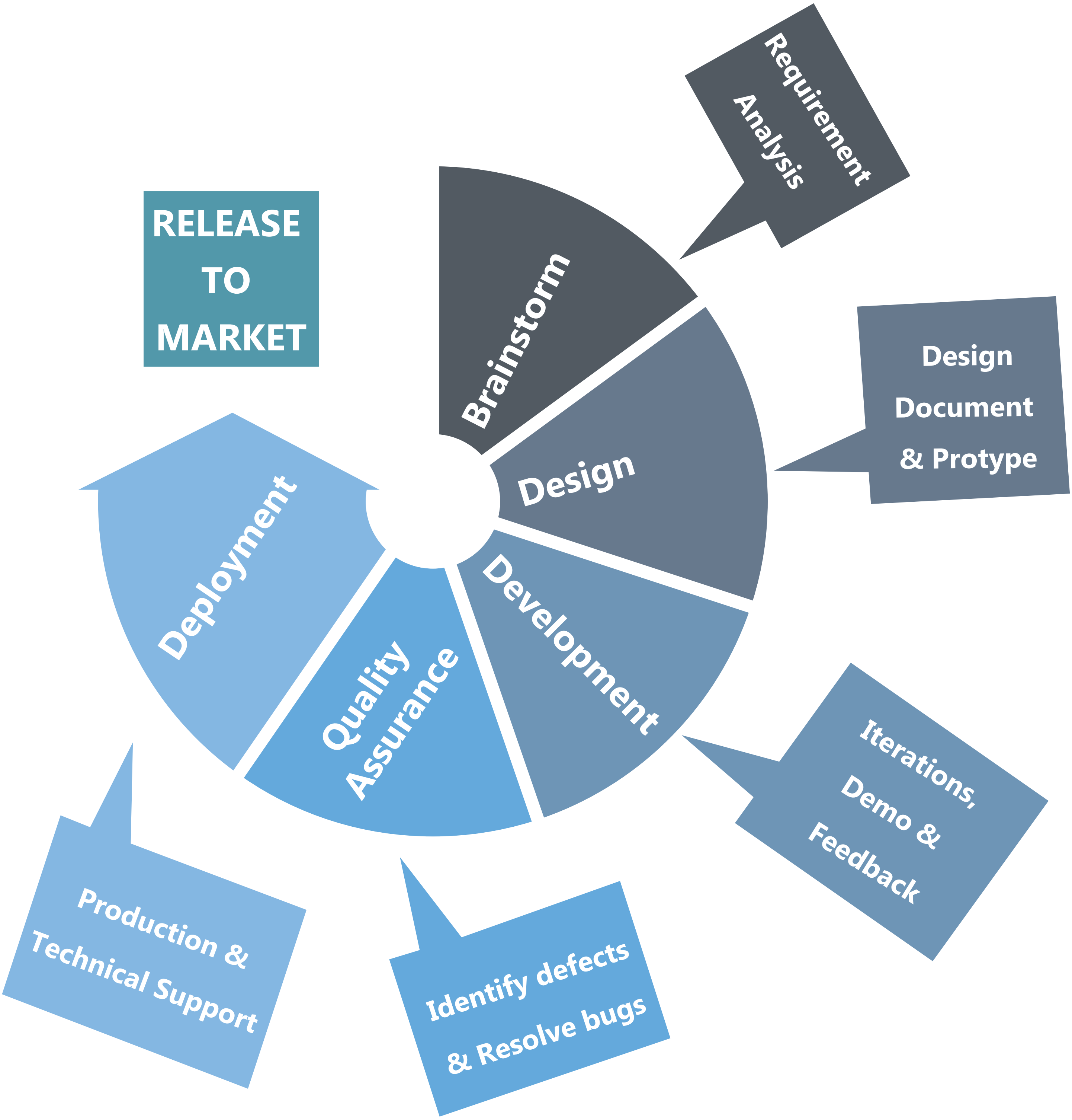
As a direct consequence of this possibility, the“ispect-and-adapt” approach to development helps to reduce cost of development and also, time to market.

Because a team’s work cycle is limited to two weeks, stakeholders have recurring opportunities to calibrate releases that can succeed in the market.

Agile development helps companies build the right product. Instead of committing to market a piece of software that hasn’t been written yet, 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 agile development method enables the preservation of a product’s crucial relevance to the market; thus, negating non-release situations.

Agile Development Methodology


RELEASE TO MARKET
Brainstorm
Requirement Analysis
Design
Design Document & Protype
Development
Iterations, Demo & Feedback
Quality Assurance
Identify defects & Resolve bugs
Deployment
Production & Technical Support

Agile Model Implementation

Agile provides a certain freedom and opportunities for far-reaching changes, which make the model highly significant. New changes can be introduced rather easily, and at minimal spend, because of the frequency of new increments that are produced.

If developers intend to implement some new features, they stand to forfeit only a few days' worth of work (at times no more than a few hours), to get back to the drawing board and originate it afresh.

Contrary to the waterfall model, hardly any planning is required to jump-start a project based on the agile model. Given that we inhabit an ever-changing IT and business ecosystem, Agile presumes that end users' needs are also dynamic.

This affords possibilities for changes to be discussed which result in the introduction or removal of features depending on the feedback received. This caters efficiently to the unique needs of the customer.

Lean

Lean

Lean software development (LSD) is a translation of lean manufacturing and lean IT principles and practices to the software development domain. Adapted from the Toyota Production System, a pro-lean subculture is emerging from within the Agile community.

The term Lean software development originated in a book by the same name, written by Mary Poppendieck and Tom Poppendieck.

The book presents the traditional lean principles in a modified form, as well as a set of 22 tools and compares the tools to agile practices.

Waste is defined as anything which adds cost without adding any value and is categorized as:

Muda - Work which absorbs resource but adds no value.

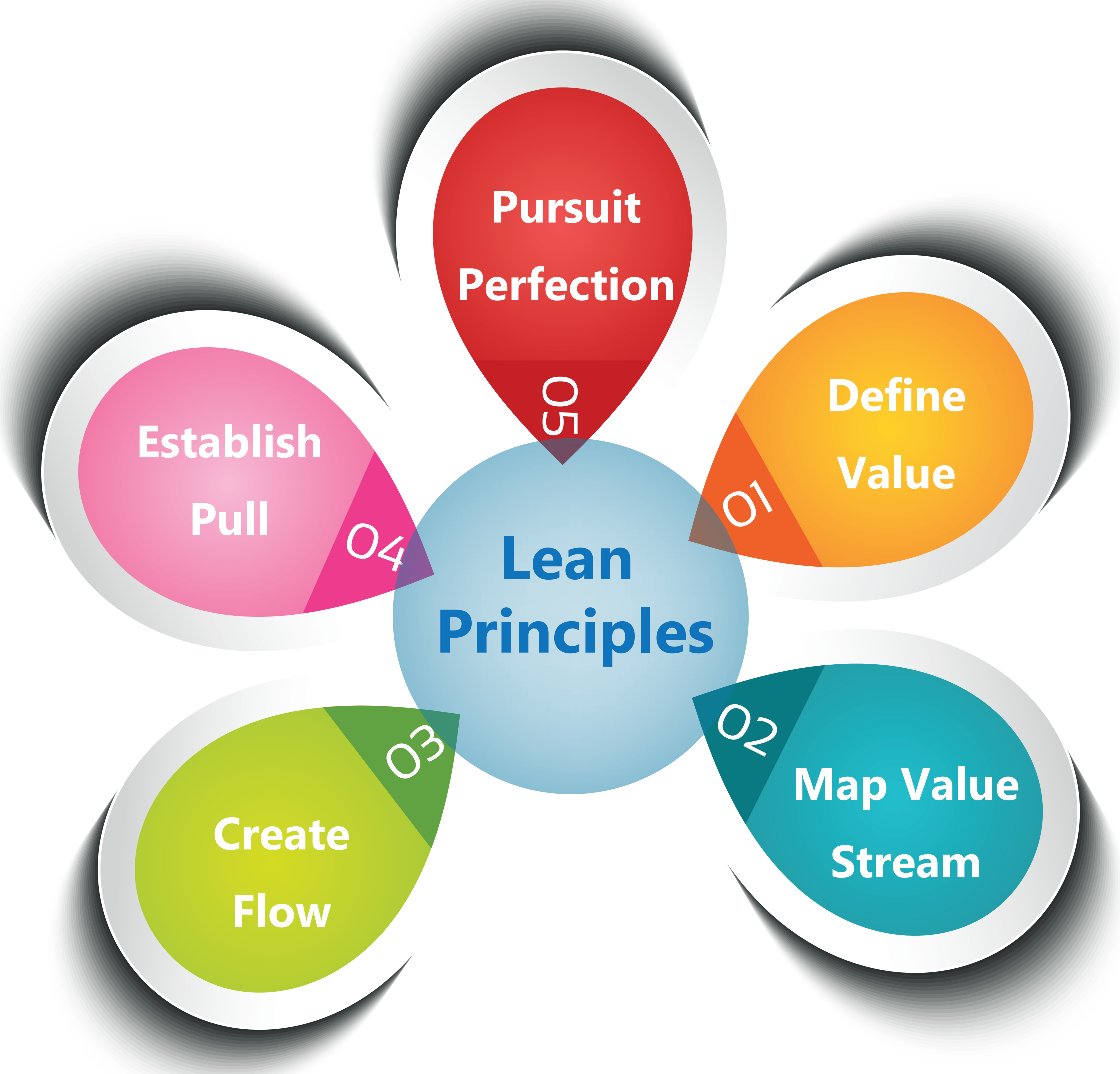
Muri - Unreasonable work imposed on workers, machines.

Mura - Work coming in dribs and drabs with sudden periods of rush rather than a constant or regular flow, unevenness.

Lean Software Practices

Lean software development practices, or what the Poppendiecks call "tools" are expressed slightly differently from their equivalents in Agile software development, but there are parallels. Examples of such practices include:

- ✓ Seeing waste
- ✓ Value stream mapping
- ✓ Set-based development
- ✓ Pull systems
- ✓ Queuing theory
- ✓ Motivation
- ✓ Measurements



Eliminate Waste from Processes

Lean philosophy regards everything not adding value to the customer as waste (muda). Such waste may include:

- ✓ Unnecessary code and functionality
- ✓ Delay in the software development process
- ✓ Unclear requirements
- ✓ Avoidable process repetition (often caused by
- ✓ insufficient testing)
- ✓ Bureaucracy
- ✓ Slow internal communication

In order to eliminate waste, one should be able to recognize it. If some activity could be bypassed or the result could be achieved without it, it is waste. Partially done coding eventually abandoned during the development process is waste.

Extra processes and features not often used by customers are waste. Waiting for other activities, teams, processes is waste. Defects and lower quality are waste. Managerial overhead not producing real value is waste.

Avoid the Unnecessary

Lean thinking changes the focus of management from optimizing separate technologies, assets, and vertical departments to optimizing the flow of products and services through entire value streams that flow horizontally across technologies, assets, and departments to customers.

Lean thinking forces teams to eliminate anything that isn’t adding value to the product or service development process. Lean organizations try to avoid unnecessary meetings, tasks and documentation.

They also try to eliminate building things that are not needed, which can be challenging because nowadays, market needs change more rapidly than ever.

Some of the tools map quite easily to Agile methods. Lean Workcells, for example are expressed in Agile methods as cross-functional teams.

Conclusion

Conclusion

Both the Agile and Lean methodologies require a Project Manager to work closely with the development team and customer; driving the pace of the project. As a result, the Project Manager plays an integral role in the success of the overall project.

There is a distinct lean software community, as in a mailing list calling itself lean and people who label themselves as lean thinkers. But this is no different to the fact that there are also strong XP, Scrum, and other communities.

Most people in these communities consider themselves part of the broader agile movement and many people are active in more than one of these agile communities.

The whole point of coining the word 'agile' comes from recognition that we share a core set of values and principles and this common core means what we have in common is greater than our differences.

About Orchestrate

Orchestrate is a US based business process management organization with Headquarters in Dallas, Texas. Orchestrate offers services to diverse outsourcing requirements of clients in an extensive range of businesses including IT, finance, mortgage, and contact center. We provide a comprehensive suite of technology and services to our clients that help accelerate sales and boost their profit. Our solutions and services help SMEs and enterprises to implement technologies and processes that boost their profitability across the organization.

ORCHESTRATE
Solutions for higher performance!

1330 Capital Parkway, Carrollton TX 75006

Toll Free: 800-232-5130 | success@orchestrate.com

www.orchestrate.com