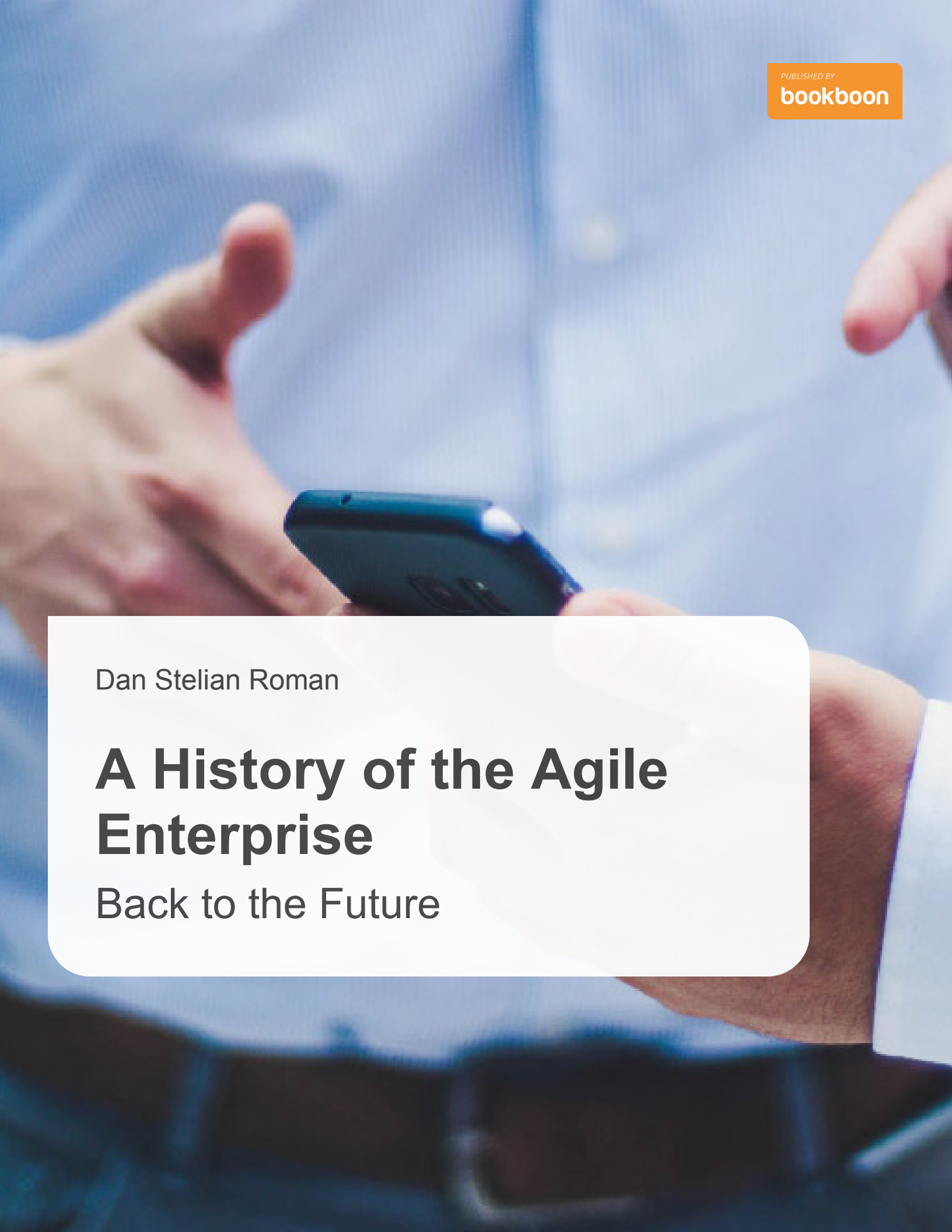

PUBLISHED BY
bookboon

Dan Stelian Roman

A History of the Agile Enterprise

Back to the Future

A History of the Agile Enterprise: Back to the Future
1st edition

ISBN 978-87-403-3928-4

CONTENTS

ABOUT THE AUTHOR

Dan is an experienced Project Manager with over 35 years commercial experience in a broad range of business areas, organization structures and cultural environments. Dan started his career as a specialist in Computer Aided Design and Manufacturing (CAD/CAM) following his involvement in research during university studies. During a successful career in research, he developed process improvement strategies, Finite Analysis Method software, forms recognition algorithms for robots and computer assistant design software for manufacturing industry. In 1990s Dan published articles and books on various topics such as process improvement CAD/CAM, manufacturing processes computer simulation, c/c++. Later he moved to software consulting working in Romania, Bahrain, France and other countries. After managing large development teams Dan moved to Project Management and is now specialized in Business Transformations Projects.

Dan has extensive hands-on experience with formal Project Methodologies (PMBOK, PRINCE2) and Agile frameworks like Scrum, Extreme Programming (XP), SAFe and Disciplined Agile and he is also a Lean Six Sigma Black Belt. He studied Risk Management at University of New South Wales, Sydney Australia.

Dan is an active volunteer for the Project Management Institute (PMI®) as a Subject Matter Expert in Agile, Traditional and Hybrid project delivery and Risk Management. He was awarded the 2021 volunteer of the year for his contribution to the development of PMP, ACP, DASSM certifications and Hybrid and Risk Management micro-credentials. He is the author of "The Agile Enterprise" series of webinars on www.projectmanagemen.com, the knowledge sharing website of the Project Management Institute, a series of over 50 webinars watched by more than 500,000 people.

https://www.linkedin.com/in/dansroman/
https://www.projectmanagement.com/profile/StelianROMAN

PREFACE

Agile, a topic that nowadays remains as exciting as 50 years ago when the Manufacturing industry realized that in their quest for perfect quality, the famous six sigma level of 4 defects per one million opportunities, enterprises lost the ability to adapt fast and efficient to market needs.

This book is an attempt to present the unknown story of the Agile Enterprise, a concept and term originated in manufacturing almost 30 years before the publication of the Manifesto for Agile Software development.

The book is the first in a series of books written for Agile practitioners that are interested in Agility at the Enterprise level, beyond the Agile frameworks used by small teams of software developers. The books are written based on an extensive literature research, especially for articles and books that were published before the Manifesto for Agile Software Development (2001), and author's 40-years of experience using Agile practices, leading Agile implementation in software companies and implementing Agile practices for project teams and organizations as part of Business Transformations and transition to a more Agile way of working.

It is a practitioner's view, from the imperfect 'shopfloor' often very different from the perfect Agile presented in conferences and training courses. The "Agile Enterprise" books are dedicated to the people that are 'doing it and help others to do it', the ones that were not afraid to take risks, make mistakes doing what was never done before, in their quest to uncover better way of developing products and delivering projects.

Progressing from product 'developer' to Product Manager and then to Project Manager was an interesting journey and sharing it may help others to make different mistakes rather than repeating the ones already done in the last century.

1 THE AGILE ENTERPRISE

1.1 AGILE VS LEAN

Agile, the buzzword of the last decade. It is often presented as something new, something that started in 2001 when the "Agile Manifesto" was published. Common sense is that new product development was always Agile, especially in software development. Development is not limited to build and the actual 'development', before it is mass produced will always be Agile. Lean comes after the first version of the product was delivered. Lean is more about process rather than the product, like Agile. They can work very well together and that's where management skills and experience come very handy, in balancing capacity to adapt with delivery efficiency.

Without disputing the role that the Manifesto for Agile Software Development had for the awareness of a better way of developing products it is good to know that Agile, like Lean started in manufacturing and rather than being imposed by a person of theory, it was what the market needed at the time. The mathematical foundation n of Lean Six Sigma was triggered by the increase in commerce in the 18th century, the Industrial Revolution of 19th century created the need for mass production and mass production needed efficiency and quality, and that lead to the appearance of Lean Six Sigma. The near perfect quality of the 19th Century products, built to survive one or more generations was no longer aligned to the 21st Century market, where a product can become obsolete in a couple of years, when the customers demand what they need and have the buying power to make their voice heard.

The first Agile project

Initially known as 'mass customization' Agile started in manufacturing as a response to inability to design, develop and produce fast the product that market needed. Historically the first use of the word "Agile" in the context of adapting fast is probably the "Project Agile" initiated by the United States of America in 1963. The report was declassified in 1975. Initiated to with the declared objective of "performing the research, development, test, and engineering of more suitable and effective devices and equipment for employment under specific conditions" the project that the project had to solve was that past and present experience has demonstrated that current equipment supplied in these situations are often less effective than current technology is capable of developing and the U.S. is able to supply at an acceptable cost. The report looked at Agility from a very broad perspective, the entire product lifecycle, from market research to understand the needs, and then designing, building and delivering those products to the market.

In 70's and 80s there were many initiatives and "Agile" practices used in product development most of them empirical and done by small product development teams, like incremental and iterative delivery, and also a large scale scientific approach undertaken by the "Agile Manufacturing Enterprise Forum", at the Iacocca Institute of the Lehigh University, where 15 executives from 13 companies had the advice and counsel of over 150 people from 77 companies and 11 professional organizations developed and published in 1992 the "21st Century Manufacturing Enterprise Strategy". The forum concluded that the "industrial era dominated by mass production manufacturing is drawing to a close" and the 21st Century will be "dominated by agile manufacturing enterprises". Unlike the manifesto for Agile Software Development published 20 years later, the strategy defined by the Agile Forum were a scientific approach led by industry and involving Government and Academia. The amount of time (7500 hours) and the hundreds of people involved in producing a few hundred pages report puts in perspective the Enterprise Agile and the Team Agile frameworks developed based on 4 values and 12 principles developed by 17 people in 2 days while skying and relaxing.

1.2 TEAM VS ENTERPRISE IN AGILE

Whilst in manufacturing at the enterprise level organizations were uncovering better ways of adapt to the market, basically using Agility to improve Lean practices, the quest for perfect quality and waste elimination that led to lack of flexibility and inability to adapt fast to market changes, software teams were evolving from unstructured product development familiarly called 'cowboy development' to a more structured approach, although light. In many cases the apparition of Agile frameworks was a response to the rigidity of project management 'governance' that was introduced to solve the lack of structure. In the last decade of the 20th Century, parallel with Agile Manufacturing scaling down Agility, software development teams developed a suite of frameworks based on empiricism, their experience in a certain project or organization. When the Manifesto for Agile Software Development was published the most popular approach was Extreme programming (XP) the framework that spread Agile in software development and was later replaced by Scrum, a lighter and easier to be implemented framework. Unlike XP, Scrum was based on scientific research, a Harvard article published in 1985. In the last decade Agile become popular and teams realized that there is a need of 'scaling'. The most significant challenges for Agility are the lack of predictability and inherent waste associated with change. To address those changes, Agile Teams start 'discovering' Lean practices. Whilst combining Lean and Agile is a good solution it requires management knowledge and experience more than can be learned by leading a small team of developers. At the Enterprise level Agile transformations are disruptive and hard and the developer mindset, on which most Agile frameworks are built, can result in conflict and eventually failure.

1.3 PREDICTIVE VS ADAPTIVE DELIVERY

Most project and products follow a plan, some rigidly some less rigidly. However, all of them start with a plan. Not having a plan is the most frequent mistake done by Agile teams. The 4th value of the Manifesto for Agile Software Development ("Responding to change over following a plan") is misunderstood to the extreme of not having a plan by movements like #noestimates or #no projects. The Scrum framework, the framework generally accepted as the standard by most Agile teams, not only that doesn't exclude planning but has a dedicated ceremony for planning.

1.3.1 DEFENDING DR. ROYCE

"Waterfall" the method used, wrongly, to justify the superiority of adaptive delivery over predictive approach describe an approach that was Agile in 1970 when the paper was published and remains in our opinion more Agile than some of the 'scaled' Agile frameworks that were developed based on Lean practices. There are few important aspects that are less known:

Software specific, "waterfall" is not a project management methodology. It was developed specifically for software systems.
Size and complexity, the method was developed for large and complex software systems like spacecraft mission planning, commanding and post-flight analysis.
Prototyping, "do it twice"
Feedback, continuous customer involvement in a formal way earlier points before final delivery.

1.3.2 SCRUM IS NOT A SILVER BULLET

The authors of the Scrum Approach, Hirotaka Takeuchi and Ikujiro Nonalka recognized that there is a need for speed and flexibility in new product development because high quality and low cost are not enough to succeed and proposed new "rules of the game". They also recommended caution, warning that the proposed approach has "built-in limitations" and may not work in all situations.

2 THE HISTORY OF LEAN AND AGILE

Agile way of building products is not as new as some people may think. In fact, Agile is how the first version of a product is always built. The way products are built was always determined by the market demand, rather than the publication of a certain theory. Agile started before manufacturing and software development. None of the values listed in the Manifesto for Agile Software development are limited to software development or to the 21st Century. In fact a brief analysis of the first 3 values can easily demonstrate that they are not specific to Agile.

> *Individuals and interactions over processes and tools*. Nothing stops a product development team, any product, to have interactions between individuals, even if there are processes in place. Tools are probably one of the things that define humans, from antiquity to the modern society.
> *Working software over comprehensive documentation*, is obviously not limited to software development. Before the first industrial revolution 'documentation' was extremely simple, if there was one. In a society where everything is made to order and/or based on the manufacturer's market knowledge, skills and experience, 'documentation' is unnecessary.
> *Customer collaboration over contract negotiation* is also an obvious practice that preceded ethe First Industrial Revolution. Even these days many customer/client relationships are based on verbal agreements and, the smaller the scale the more room for collaboration is between the end user and the service provider

The fourth value *(responding to change, over following a plan*) is what really defines Agile. Not following a plan doesn't mean that there should not be one but that having a plan doesn't mean that it is set in stone and it should be blindly followed. However, we should always keep in mind that:

"You fail to plan you plan to fail"

This chapter is a brief presentation of the path to the current 'scaled' Agile frameworks, a brief history of Lean, Six Sigma and Agile. Although chronological the focus is not on a specific year when a certain concept or practice was documented but, on the significance, and impact on what we know these days as an "Agile Enterprise"

2.1 ANCESTRY (BEFORE 20TH CENTURY)

The concepts of Agile and Lean didn't start with the publication of a couple of books in the last decade of the 20th Century or in 2001 with the Manifesto for Agile Software Development. It is hard to find precise dates and documents, for the simple fact that before the first Industrial Revolution people were not focused on information sharing using written means, it was expensive and, in many cases, impossible, when literacy was reserved to the elite.

What is really important is why Lean and Agile emerged and how they evolved to practices that are used now. Like in the case of modern computers, a lot of concepts were defined as theories on paper, before they were needed and the technology was able to implement them.

The foundation to Lean and Six Sigma are the developments in mathematics triggered by the huge increase in commerce in the 15th and 16th Century. It's not a surprise that the first event considered as a precursor to Lean Manufacturing is considered is the Venetian Arsenal's 'assembly line, capable to deliver one galley every day. Also, not a surprise that when the workforce grows from few hundreds in a large workshop to 16000 a new science will emerge: scientific management. The fact that they didn't use the term Lena is irrelevant. It was all about efficiency, a mindset change, a new way of thinking that focuses on eliminating waste and streamlining processes to save time, space, materials and money. Whatever they called the new way of working, when excess is reduced or removed from a process, the process becomes 'lean'.

More trading means more money to count, more risk to take and that's where mathematicians can help. In the late 1800's, Pareto defined the '80-20 Rule' stating that that where two related data sets or groups exist (typically cause and effect, or input and output), they often show certain behavior. The Pareto Principle is a cornerstone of Lean Manufacturing, one of the simplest and most powerful management tools available. It is a simple technique that is extremely helpful when deciding where to focus effort and resources. The last 20% improvement in quality will require 80% of the effort and resources, "80% of the downtime come from 20 percent of problems" or "80% of a company's profits come from 20% of its customers". This rule applies to any product, software products included and to any process, software development included.

Another development that is often 'forgotten' by Agile practitioners because it's association with Six Sigma, seen by some people as an Agility inhibitor, is the normal distribution, the bell-shaped curve that become very popular on TV when the COVID pandemic started. The curve shows that during any measurement values will follow a normal distribution, usually with an equal number of measurements above and below the mean value. The normal curve is perhaps the most important probability tool in all of statistics. The person who first derived the formula, Abraham DeMoivre, trying to solve a the "Problems of Chance," or

how likely it is that an "experiment" will produce a given outcome. Later work, especially by Carl Friedrich Gauss, used the normal distribution to describe the pattern of random measurement error in observational data. Neither man used the name "normal curve."

On the manufacturing side there were many innovators, although the credit of 'inventing' interchangeable parts goes usually Eli Whitney there are many others that contributed to the development of the foundation of Lean manufacturing. Eli Whitney played an important role in popularizing the concepts of "Just in Time" or JIT production.

2.2 CHILDHOOD 1900-1960

The second Industrial Revolution was the catalyst for a significant development in Lean practices. From mechanization and interchangeable parts, manufacturing evolved to mass production. In mass production any minor improvement can deliver significant benefit. This need for 'lean' processes brought statistics on to the shop floor. Although not as famous as Henry Ford's mass production of automobiles, Gilbreths' process analysis and process mapping had a significant influence in the success of mass production of affordable automobiles Frederick W. Taylor's scientific management streamlined manufacturing processes redefining management. The practices used were not called 'lean' but eliminating waste evolved from an approach practiced by a few innovators to a widespread movement Like many innovators at the time, Lillian Gilbreth was a multi-skilled scientist. one of the first female engineers to earn a Ph.D , she was a pioneer in the application of psychology to time and motion studies and the first ever industrial or organizational psychologist. Lillian studied the motivations of workers and the effects of employees' attitudes on the outcome of processes. Her husband, Frank, was also a pioneer with significant contributions on how human motion could increase the efficiency of assembly processes. Frank and Lillian developed charts that included elements such as cycle time, inventory and equipment information or maps depicting the sequences of events required to finish a product or yield an outcome, providing a visual tool to distinguish between waste and value-adding activities. They basically invented what is now known as process mapping, process charting or value stream mapping, essential tools in identifying and reducing waste, the very foundation of Lean manufacturing.

Henry Ford's model T is the nest example of what Lean production is, its benefits and shortcomings and why we need Agile. The Ford Model T is arguably the first major success of lean production. Famous for low cost and fast delivery time achieved by standardizing processes, continuous flow and advanced machining techniques it is less known for limitations like the color black, (chosen because it dried fast increasing the efficiency) or the social problems with the work force, one of Ford's biggest problem being employees retention.

When you have a workforce of tens of thousands in a single factory, empirically developed skills are not enough. Widely recognized as the 'father of Scientific management' Frederick Winslow Taylor was the first man in recorded history who believed that work deserves a systematic study. Taylor's scientific management consisted of four principles:

1. Replace rule-of-thumb work methods with methods based on a scientific study of the tasks.
2. Scientifically select, train, and develop each employee rather than passively leaving them to train themselves.
3. Provide "Detailed instruction and supervision of each worker in the performance of that worker's discrete task"
4. Divide work nearly equally between managers and workers, so that the managers apply scientific management principles to planning the work and the workers actually perform the tasks.

Although many refer to Taylor's management as an autocratic leadership style, the 'command and control' style that Agile frameworks are replacing with a collaborative style, it is important to note that his theory was developed in the 19th Century and most of principles are still valid, especially when the size of the enterprise or the particulars of the business domain makes a pay per individual performance more motivating than 'team recognition'.

The start of the 20th Century brought significant tools and concept that are still at the core of Process Improvement. Ronald Fishers, Analysis of Variance (ANOVA) and Design of Experiments (DOE), the Gantt chart and Walter Shewhart's 3 sigma, setting a target of three standard deviations to measure the quality of a product are only few of the achievements of the first couple of decades. It is interesting to note that a 3-sigma quality level, the target that was set by Shewhart for manufacturing is still a dream in software development. A 3-sigma quality is less than 10% rework, a level that in software development is equivalent with very high quality. In general software development teams are still a bit above a 2-sigma level with an average of 20-30% rework.

Towards the first half of the 20th Century the focus on quality shifted from United States of America to Japan. Driven by American scientists like Deming, the inventor of statistical process control and Joseph M Juran, a Romanian born American and management consultant that defined the concept of quality management. From the Agile transformation at the Enterprise level, it is important to note that quality, like Agile should start from the top. The famous mindset change starts with a new culture, a culture of trust, collaboration and empowerment.

The Japanese lead the development in Lean for the most part of the post War manufacturing. Practices that start emerging in 21st century as 'scaled Agile' originated in this period. Shigeo Shingo's Poka Yoke (stop people to make mistakes) and non-stock approach are less known as Taiichi Ohno's Just in Time, Kanban and the Toyota Production System (TPS) that is basically the precursor of the modern Lean Six Sigma.

Other important development in this era were the Ishikawa diagram (cause and effect) or fishbone diagram a powerful tool to empirically investigate and identify the root causes of failures. Kaizen, or continuous improvement, has its roots in the Toyota Production System. The primary objectives of kaizen are to identify and eliminate "Muda," or waste. Its key elements emphasize re-engineering processes to increase efficiency of the work process, constantly improving the product quality. cognized worldwide as an important pillar of an organization's long-term competitive strategy. It is a systematic approach for business improvement involving everyone that is part of the product development process managers as well as workers.

Although not explicitly defined, the Scrum framework is basically a Kaizen implementation. It relies ion short cycles with a very important event at the end of each iteration: the Sprint Review when the entire Scrum Team reflect on how the Scrum process can be improved, what needs to be eliminated, improved or retained. It is also an opportunity to identify new practices to see what else can be done to improve the process. Very important is the fact that in Scrum, process improvement is not imposed by the management or anyone outside the Scrum Team but it is the responsibility of the entire Scrum Team.

2.3 ADOLESCENCE (1960-2000)

In manufacturing 1960s market the resurrection of the Western industry, inspired by the success of the Lean practices in the Japanese Automotive industry. It is also the time when the computer came to the enterprise, when it was not present only in universities and used mainly for scientific or military purpose. Little known by software developers, the use of computers in manufacturing proceeded the creation of the well-known FORTRAN (FORmula Translation) and COBOL(Common Business Oriented Language). The Automatically Programmed Tool (APT) is a programming language created in by Douglas T. Ross at MIT. Unlike the business (COBOL) and scientific (FORTRAN), APT does not 'compute' but generate instructions for numerically controlled machine tools. It is important to know that rather than information technology influencing manufacturing the history proves quite the opposite. The famous paper punched cards or tapes originated in manufacturing, as a mechanical control for looms. The 1804 invention is the starting point of what letter will become the most important computer manufacturer: IBM. The Hollerith's code for relating alphanumeric characters to holes in a punched card was devised by Herman Hollerith in 1888 and enabled the letters of the alphabet and the digits 0–9 to be encoded by a combination of holes in 12 rows of a card. It was the common way to write computer programs for decades, until magnetic media was invented.

Punch cards were in use when the first documented iterative and incremental software delivery was made. Gerald M. Weinberg recalls that incremental development was used as early as 1957 but the team rather than thinking that it is a revolutionary approach, assumed it as totally natural and the development of any software system can't be done in any other way. The proof is that no software system is still on the market with version 1.0. Incremental and iterative software development was used again in 1958 to develop the Project Mercury, the seed bed out of which grew the IBM Federal Systems Division. Another technique usually associated with Agile that was used in the Project Mercury was Test Driven Development.

In fact, the article that introduced the method known as "waterfall" was at the time and even today a very good example of an Agile process. "Waterfall" is neither a Project management approach nor a generic product development method. It was written specifically as a proposal for the development of large and complex software systems. It had not only the feedback loops between the development phases but also recommended continuous customer involvement and prototyping. The method was recommended for the development of software systems that are delivered to an external customer. For internal applications, the kind of systems that most Scrum teams build, the "waterfall" had 2 steps only: design and coding.

The term "Agile" as a new product development approach that can respond fast, and efficient, to changes to the market demand was very likely used for the first time in 1961 in a Department of Defense initiative: the "Project Agile". The core issue that the project tried to address was the inability of the American Manufacturing industry to produce the kind of equipment that was needed for the unconventional combat.

The 1970's years can be remembered as the time when the 'servant leader' concept was conceived by Robert Greenleaf and the Kano Model which classifies customer preferences into five categories. Important to note that according to Kano, as customer expectations change with the level of performance from competing products, product attributes can move from delighter to performance need and then to basic need. The "Servant Leader" management model resurfaced in the Scrum framework as the Scrum Master role.

The last two decades on the 20th century were marked by the definition of new concepts, concepts that were used as practices but not formally named of defined:

- **Theory of Constraints** defined by Eliyahu M. Goldratt, easy to describe as "a chain is no stronger than its weakest link" that there is always at least one constraint that should be the starting point for restructuring the entire organization around it.
- **Six Sigma** adopted by Motorola, General Electric and Honeywell
- **Spiral Model** developed by Barry Boehm
- **Tekeuchi and Nonaka's Harvard** article describing the software development inspired by the game of rugby, that later become the Scrum Framework
- **Lean** term defined by John Krafcik
- **Scrum** presented by Jeff Sutherland and Ken Schwaber
- **Extreme Programming (XP)** developed by Kent Beck
- **Crystal Clear** Developed by Alistair Cockburn

If in software development Agile emerged as a more structured approach than the 'cowboy development' done by citizen developers, in Manufacturing, Agile was a very scientific approach involving executives from large manufacturing companies, academics and government officials. The most important initiative was the Agile Forum, organized at the Iaccoca University where 15 executives from 13 companies had the advice and counsel of over 150 people from 77 companies and 11 professional organizations developed and published in 1992 the "21st Century Manufacturing Enterprise Strategy". The forum concluded that the "industrial era dominated by mass production manufacturing is drawing to a close" and the 21st Century will be "dominated by agile manufacturing enterprises".

Agile Manufacturing, based fundamentally on the concept of mass customization, included many other revolutionary concepts like the Virtual Enterprise, an enterprise where party of the functions are done by partners, and alliances will replace competitiveness in order to expand the market, and/or enter new markets. A lot of Agile literature was written in the final wo decades of the 20th Century. Due to the limited scope of this book, we will briefly mention some of the most important.

21st Century Manufacturing Enterprise Strategy Report prepared in 1992 by Roger N. Nagel for the Office of Naval Research recommending the build of a new infrastructure to support transition form mass production to agile manufacturing as a unique opportunity for U.S. Industry to regain the leadership lost in the 70s and 80s.

Agile Manufacturing published by Paul T Kidd in 1994 defined the concept Agile Enterprise, yet to be implemented as an organization capable to respond rapidly to changes in customer demand and taking advantage of any window of opportunity that appears in the market. The book identified few mandatory factors for an Enterprise that wants to be Agile:

- Highly skilled, knowledgeable and motivated people who are flexible, motivated and responsive to change
- Organizational structures that engender non-hierarchical management styles and stimulate and support employees as well as cooperation and team working
- Advanced Information technology support

Agile manufacturing from a statistical perspective, a 1995 study written by Robert G. , a member of the Agility Forum's Agile Operations Focus Group considering Agility as the next step after flexibility and defining it as the "ability to thrive on unpredictable change".

Pathways to agility: mass customization in action, published in 1998 by John D Oleson. The book describes how Agility enables companies and individuals to respond effectively to unexpected. An Agile Enterprise grows, evolve and reinvent itself as its responds to changes in the market and fulfills the needs of its customers.

The Agile Virtual Enterprise. Cases Metrics Tools by H.T. Goranson, attempting to address the lack of management tools to manage the problem of responding to unexpected change. Part of the research was the organization of an international conference. Cosponsored by the US Government and European Strategic Program for Research in IT. This research was the business and technical motivator for two important technical developments: Java language and the Object Request Broker, a standard protocol for integrating programming units called objects to support a similar goal.

2.4 MATURITY (AFTER 2000)

From an Agile perspective the 21th century was marked by the publication, in 2001, of the "Manifesto for Agile Software Development", also known as the Agile Manifesto. Unlike the Agile Manufacturing, in Software development there was no established process. Not only that the nature of software development is hard to be limited o a certain framework

but, even in 2001, the software development was still an emerging profession. Most of the Authors of the Manifesto for Agile Software Development were consultants, not actual coders. Although the introduction mentioned 17 participants, there are only 14 names listed. Before and after the publication the most of the signatories had their own view on Agile and their own framework. All frameworks were developed by the authors in a certain context, some of them by the participants in a a project, and some as a biproduct of a book or a consulting engagement.

XP. At the time of the publication of the Agile Manifesto the Agile software development was predominantly using Extreme Programing (XP) a framework that opened the door for Agile. There were XP international conferences and training courses, although there was no XP certification. Although the first XP project was a failure, the project delivering 30% of the scope in more than twice the initial time and budget, the framework laid the foundation of the Agile software development. Most of the software teams that embraced Agile in the next couple of years used XP.

In the past couple of decades Scrum become the norm for Agile software development. The last state of Agile report indicates that 0ver 80% of the teams use Scrum (66%) or combination of Scrum (ScrumBan, Scrum hybrid). XP was too prescriptive and formal for small teams and that resulted in the tremendous success of the Scrum framework that replaced XP in most software development teams. Unlike XP Scrum is very simple and acts as a container for other practices, making it far more flexible. Scrum also has a myriad of certifications, that besides the knowledge gives a sense of achievement for individuals and organizations.

There are few 'scaled' Scrum Approaches (SAFe, Disciplined Agile, LeSS), all of them retaining the Scrum Master role, which is a clear indication of the 'scale'. Scrum Master is a managerial role defined for small teams (less than 10). Although it can be very useful for small software development teams it lacks the skills and experience required to implement Agile at the enterprise level.

However, the scientific approach used in Agile Manufacturing demonstrates that Agile mindset (responding fast and efficient to changes in market demand) is a necessity and Software Agile, in a more mature form will become the norm in the next decades.

2.5 WHAT'S NEXT

Like any new practice, methodology or framework, Agile will pass the four stages of the innovation curve. Both Manufacturing and Software Development versions are no longer a novelty or a curiosity. They were accepted by the industry and in certain areas become the

norm. Unlike Agile Manufacturing, that was a natural progression and have the infrastructure and resources to be successful, the Team Agile used in software development is yet to find it's way outside software development. The perceived success is more a marketing and wishful thinking than reality. The perceived success is a by-product of the Agile certification industry, a multibillion revenue stream from hundreds of certifications, rather than reality on the ground. Scrum, the framework that according to the State of Agile surveys, is used by more than 80% of the software development teams, sometimes in combination with other practices, is the norm because it suits the software development process. Excellent marketing and the lack of understanding among Senior Management created the false impression that Agile is the norm and the benefits are obvious.

Reading between the lines of the 15th State of Agile Report we can get a completely different picture. The report indicates that at the over 90% the organizations adopted Agile only in software development or Information Technology related activities. Taking into consideration that most of the software development in a large enterprise are outsourced, including the Software as a Service, Infrastructure as a Service it is hard to believe that Agile is used by other teams in the organization. A significant proportion of the respondents adopted Agile to 'accelerate software delivery', demonstrating that software development Agile frameworks are not used for other product development. Agile is still a way to align IT with the Business, and the 50% claims that Agile improve predictability and reduce risk demonstrate a misunderstanding of Agile values and fundamental principles. The core difference between planned and adaptive delivery, like Agile, is 'embracing change, therefore Agile can't improve predictability. Change comes with Risk and Agile fundamentally increase the risk, although risk is not always a negative aspect that needs to be reduced. Risk taking is fundamental for Agile, like standardization is fundamental for Lean.

For the moment Agile is dominated by laggards, organizations that want to be perceived as "Agile" but the only change in their culture and processes are the "Agile" label attached to traditional roles, engaging "Agile Coaches" and sending staff to become "Agile" certified.

The main emerging 'scaled' frameworks in software development and the evolution of Agile Manufacturing indicate clearly what is likely to happen in the future. Agility is bouncing back and is bringing Lean and sometimes Six Sigma in support of efficiency. Capability and capacity to respond to change can't be build chaotic or empirical. To survive, any organization must be profitable. Unlike Lean that has very good metrics, Agile 'success' is based on velocity expressed in relative 'points'. One of the most significant shortcomings of Agile Transformations is the lack of benchmarking.

"You can't improve what you can't measure"

Looking at the success of manufacturing and the 'new' Agile in software development it is clear that Lean is part of Agile's future. Rediscovery of Kanban, the development of DevOps, as well as 'scaled' Agile Frameworks like SAFe and Disciplined Agile Delivery, built on Lean concepts and practices is a clear indication that Agile reached its peak and is now trending downwards.

To be successful, Agile needs a new organization culture and once the Executive level understand that strengths and weaknesses of this new approach Agile will thrive in any organization. Agile needs knowledge and support for learning from mistakes. Agile is not failure proof but it's a way to benefit from failures. An Agile Enterprise doesn't have a pyramidal structure but a flexible structure that can adapt to the external environment. To achieve that an Agile organization needs knowledgeable workers, a supportive management and a culture of trust and collaboration.

Agile is the best option to survive in a competitive and fast changing market but organizations must get to the next stage of Agile, a stage where Agile is not a software development approach, but a product delivery way that is supported by all the teams in the enterprise. Information Technology has a pivotal role, but not the driver role. Information Technology is an enabler for Agility and a service provider to Agile Teams.

3 THE ROAD TO THE AGILE ENTERPRISE

3.1 CUSTOMER CENTRIC OR PRODUCT CENTRIC?

The customer Centric organization is not an Agile concept, nor it is a new approach. Lean Six Sigma introduced the Voice of Customer technique and many other practices like, zero Defects, Poka Yoke etc. Lean Six Sigma is customer centric, focusing on providing high quality at the lowest price. It achieved both goals almost perfectly. The Quality of a company that reached a Six Sigma Level, 4 defects in one million opportunities, is almost perfect. The cycle time, takt and efficiency are also near perfect with practices, methods and tools to eliminate or at least reduce any kind of waste, making the product affordable.

If Lean Six Sigma delivers what the customers asked, at the highest quality and at the lowest possible price, then why do you need a new approach? Why do we need Agile if we gave the Customer what they asked for? Because the Customer didn't know, and will never know what they need. Between what a customer needs and what a customer asked for is a big difference. That 's normal even when the basic 'needs' have the highest priority. The customer doesn't really know what they need, they will have an idea but not in detail. There is also the context that may and usually will change since the order was placed. The evolution of manufacturing is a very good illustration of the necessity to be Agile and move from customer centric to product centric.

Let's look at transportation. First the 'customer' asked for a mean to travel between distances. The 2-wheel cart was a better way than horse riding. Then the 'customer' wanted to take some heavy goods with them and the carte become a 4-wheel cart. If a small cart could be built by a single 'multiskilled' person, or even the 'customer' itself, when the cart become bigger and more used things changed. The 'workshop' had specialized people that could produce better carts and deliver them faster to the 'customer'. More carts required more horses, problem solved with replacing the horses with an engine. Producing an engine is no longer a 'workshop' task, the workshop become a factory. The number of 'customers' become so large that the factory couldn't cope with the demand and that's how mass production emerged. But to manufacture many and cheap versions of a product something had to be sacrificed. Ford Model T was cheap and mass produced but the choice was limited, same shape, same size, and very limited choices of color. However, the 'customer' was happy. But the society was still progressing, the 'customer' had now the means to pay for a more personalized 'cart' but not that much money to buy a cart 'made to order'. That's when Agile came to the rescue. It was called 'mass customization', the ability to produce smaller lots of products as efficient as millions of units.

Agile is no longer customer centric but product or service centric. The focus is not on what the customer is asking now but what in what they may ask when the product is ready for delivery. Some products are not (yet) needed. An Agile Enterprise must be ready to build the product that the market needs, when the marked is asking for it.

3.1.1 PROCESS, PRODUCT, PROJECT?

Product vs project is one of the Agile's most misunderstood concepts. The key to understand the difference is in the characteristics and the objectives.

> A **product** is a 'thing' that is built (manufactured) and satisfy a need of the person that built it or the person that will buy it. It can be always defined by some attributed and by the effort or cost associated to make it.
> A **project** is temporary activity effort to create value through a unique product, service or result. Unlike a product, that can be the scope of a project, it should have an clear beginning and end. A project is defined by 3 attributes: time, scope and budget. Unlike the product that always has a cost but can be left unfinished, a project must deliver the scope in the time and within the budget agreed between the one delivering it and the one paying for it.
> A **process** is a series of actions or steps taken in order to achieve a particular end.

Although some software development Agile frameworks are presented as Project Management methodologies, they were conceived for product development. The Agile Manufacturing have a process-oriented approach, inherited from Lean Six Sigma. Rather than focusing on a certain product, like software development, Manufacturing Agile is building the process that can adapt fast to market demands, inheriting he Lean efficiency.

A product will require more than one project during its life span. The first project can deliver only a proof of concept, a prototype or a small number of the products, with the goal of testing the financial viability of the product. The next project can deliver only the build at scale process, the tools, the skills, the governance required etc. That's when the release of the first version of the product kicks off. There will be more projects, either to deliver a new version of the product, enhancements to a version or enhance the build process. A real product will also need a project to decommission the infrastructure required to build and deliver the product to customer.

3.1.2 #NO AND AGILE

Agile, the software development version is usually promoted as a replacement for 'bad' practices. "No" is commonly associated with traditional practices to indicate that Agile is simpler and doesn't require any formalism to be successful.

One of the most used arguments used to promote Agile in software development is the "waterfall", something that should not be used because it is anachronic and, unlike Agile, inhibits creativity and personal initiative. That's a very big mistake that a team can make. Taking the "#nowaterfall" for granted. First of all, despite being associated with project management and with the 'no-necessary' Project Manager role in special, the method was developed for software systems. At that time it was a revolutionary approach and introduced some concepts that are seen today as the cornerstone of Agility: feedback loops, prototyping and continuous customer involvement in the build process. Unknown by many Agile enthusiasts, the method was developed for large and complex software systems that are built to be delivered to a customer. The "waterfall" method recommends only 2 steps for the kind of software that is developed internally by Scrum team: analysis and coding., far more 'agile' and less prescriptive than any Agile framework released after 2001 when the Manifesto for Agile Software development was published.

There are 2 "#no" movements that are perhaps more dangerous than the "waterfall" misconception:

3.1.3 PROJECTS OR #NOPROJECTS?

This is one of the most controversial aspects of Agile implementation. Many software development teams believe that the Project Manager role is an unnecessary overhead and can be replaced by distributing the responsibilities between the Scrum Team roles: Product Owner, Scrum Master and the Developers. That's easy on paper but catastrophic in reality:

> **Developers.** The only project managed responsibility that, in theory, can be shared with the developers is planning. Scrum delegate planning activities to the entire team, with the Product Owner responsible for setting priorities and the Developers in organizing their work. Although it is possible it requires a maturity level that needs time, knowledge and experience.
> **Product Owner.** This is the role, expected to undertake most of what a Project Manager does, especially securing finances and dealing with the tough Stakeholders. In theory, there is nothing wrong with the Product Owner 'removing impediments' like managing the budget, procurement, integration with other initiatives and projects

in the organization in addition to managing the scope and priorities and explaining clarifying the expectations for the developers. Taking into consideration that the Product Owner is also expected to validate the work done and train the end users it is obvious that giving more responsibilities will detract from its primary role: ensuring value delivery to the organization. Moreover, most of the time the PO role doesn't have the required authority level in organization to approve budgets or dal with senior stakeholders and it is in fact an administrative proxy of the person that has the decision power to help the Scrum team.

Scrum Master. This is the role that should exist only in the context of the Scrum Framework. It is a facilitator role rather than a managerial role and is not related to project management at all. Unfortunately, it is perceived in many organizations as a (cheap) replacement for the Project Manager and asked to manage finances, vendors, produce reports for the Project Management Office and completely wrong from the Scrum, and Agile, perspective: made responsible for delivery. Mindset change is not as easy as it is sold in training courses or conferences. Someone must be the image of the project and although the common sense is that that role is the Product Owner, because it should have the authority to drive the delivery, in most cases the Scrum Master will be the one paying the price for failure. Converting the Scrum Master to Project Manager is a waste when the Scrum Master was a promotion for a good developer. Not only that the organization lost a good technical resource but the person won't be happy with the 'promotion' because it is not ready for a role. Transitioning from Developer to Project Manager is a long and difficult process that require a Team Lead stake, a lot of knowledge and a very supportive management.

3.1.4 ESTIMATES OR #NOESTIMATES? CAN YOU CROSS THE STREET WITHOUT LOOKING OUT FOR CARS?

The **#no-estimates movement** is a response to being held to imperfect expectations that were made with imperfect information. Inability to estimate, sometimes for very good reasons is often misinterpreted as in Agile there is no need to estimate. It is a very dangerous practice and can lead to catastrophic results in commercial engagements. It may work for internal work, when there is trust and all the resources allocated are 'free'. We estimate the speed of the incoming car and how much time we need to cross the street. That should be enough to demonstrate that estimation is not as useless as the #noestimate movement is trying to convince us. Although less estimation can be beneficial it is not something that will work in a conservative culture with string governance practices and should never be implemented because the team has poor estimation skills or because the scope can't be defined good enough.

3.2 FROM PYRAMID TO SHAPE SHIFTING

An Agile Enterprise is not the same with a new banner above the gate. It is a completely new kind of organization. It is hard to indicate what comes first: the culture or the structure but it is clear that the change is hard, it is disruptive and should always tart from the top. Agile is not something that will be easily understood by managers that are used to 'direct' the work of their reports or by employees that are used to be told in detail, what to do. In fact, Agility of an enterprise can be defined by the complete absence of micromanagement, at al levels. From the shop floor where the workers must plan their time decide, how and when a certain task will be done in the most efficient way, how to collaborate with their colleagues and with the management to the top level where power and authority must be delegated downstream.

The Agile transformation starts in the Boardroom, with a 'roundtable' where the "knights" are as powerful as the "King". Any change needs resources and without distributing funding and the authority to spend then as each business Unit or team sees fit, there is no Agility. If the pointy head of the pyramid decides everything then there is no organizational ability to respond to change. Structuring the organization in collaborative independent units is a prerequisite of a successful transition to Agile. Each person, team and department has his own appetite for Agile but to achieve the optimal level for the organization the management must identify the people that have the potential to change, the people that have the desire to change and encourage and support them.

It is irrelevant if a "Director" role is rebranded 'Head' or 'Lead' on paper only. If the behavior and responsibilities are the same then it doesn't bring any value to the organization.

3.2.1 CHAMPIONS

The Agile Champions are the ones that can and should drive Agility. to be an Agile champion one needs to understand Agile, be open to take risks, assume failures and use them to improve. An Agile Champion is neither a Trainer nor an Agile Coach, although sometimes they may perform those functions. Agile, especially Team Agile, is based on empiricism on learning from mistakes, which require the courage to do things different. Agile can't be learned in the training class although knowledge always help to make mistakes already done by others. However, something that worked in a different context may not work in another context and vice versa. The champion is the one 'selling' Agile, giving the confidence that the goals can be achieved and they are the leaders that will help the team to become Agile.

3.2.2 SUPPORTERS

Not everyone in the organization can or should be an Agile Champion. Some individuals can't be Agile, some are afraid of change, some have personal biases or interest. It is important that the Champions identify the changes that need to be done to support Agility and the people that can drive those change. There are few obvious business areas and departments that need a mindset change in order to enable and support Agile.

Information Technology (IT). Usually in charge also of communications platforms, the IT Department is perhaps the most important piece in the Agile Transformation, not as a champion but as a service provider. IT not a core business, it can't drive business change but can and should support it. Some of the easy ways in which IT can support Agility are:

- Collaboration platforms
- Integration points with partners and other service providers
- Flexible governance allowing the Business to use their own technical resources (internal and External)

Finance. That's one of the most important changes that Agility needs. Delegation of Authority to the team to decide where the money is spent. The Finance Department must provide a good governance framework and monitor the performance rather than ensure that the 'pointy' head signed off all expenses.

Procurement. Like the Finance department, Procurement can make or break the path to the Agile enterprise. Red tape, rigid procurement processes are always a strong Agility inhibitor. The Procurement Team must define the processes, flexible enough to let the Business chose what partners they want to work with, ensuring visibility and probity rather than imposing partners that are in the 'short' list. Simple things like letting the Business to run the Request for Proposals, to liaise with the respondents and providing only process supervision for the vendor engagement rather than setting up the criteria and deciding who is the preferred supplier are very powerful steps towards and Agile Enterprise

Human Resources is perhaps the most important 'cog' in the Agile mechanism. The top level is the engine, but without this team support Agile won't get to all the levels in the organization. Transforming this team into "People and Culture" is what Agile needs but is not as simple as changing the title of all the HR staff email signatures. Culture is the key ingredient in the Agile Transformation. Mandatory meaningless individual Performance Reviews 4 times a year is a good indication of an unsupportive "HR". Not all individuals can be direct associated with company's values and objectives. Some people are contributing just by doing their job. The job, skills and experience that they are paid for. Expecting them to do more just to keep their job is unfair. A performance system should include rewards not only fear of losing the job. When the manager can't or won't rate the performance as outstanding because that means that the person must be promoted or rewarded it will demotivate all the employees. People benchmark themselves against the ones that they consider the best and when 'the best' is not rewarded in any way the rest will stay on the gray area that won't be fired. Agile needs knowledge, initiative, courage to try (and fail) and that organization culture can't be achieved when the organization is fast in punishing failure and slow in rewarding success.

4 AGILE - REVOLUTION OR EVOLUTION

4.1 AGILE FOR ALL; WHAT IF THEY DON'T WANT IT?

Without any doubt Agile is a necessity for survival in the 21st Century market. But that's valid for the organization, not for each individual. Not every individual can, needs or have a motive to embrace Agile. Someone that is doing a repetitive task, in a way that works and there is no need for change, like paying vendors or chasing bad debtors. For an individual that is learning a certain trade continuous change will be confusing. Someone that has few more years until retirement won't have any strong motivation to learn new ways.

However, even where Agile is needed and/or wanted, the organization will need to provide guidance and support. The best way to help Agile Transformation is Change Management, to take the organization o the Road to Agile, some in sport cars, some in buses and some in trains, depending on their individual needs, willingness and organization's needs. There wil be indubitably casualties on this journey, people that will lose their jobs because the job become redundant or because they can't acquire in time the knowledge and the skills for the new job. Organizational Change Management and will make this journey less painful. An Agile Coach may help the executive to understand Agile but a Change Manager will help the organization to get there.

4.2 AGILE TRANSFORMATION; IS THE TAIL WAGGING THE DOG?

The most often and biggest mistake for an organization when it comes to Agile is to attempt to implement an Agile framework that was specifically defined for Software Development. A Scrum Master driving Agility at the Enterprise level is less likely to succeed than emptying a bathtub using a teaspoon. The Scrum Master is a Scrum Coach, a Coach for a team of 10 software developers and for one manager performing the Product Owner role, not for an enterprise with thousands of employees that have nothing to do with software development. They don't understand how software is built, the concepts of iterative and incremental product build or self- organized teams.

The success of Agile in Manufacturing demonstrates that Agile require more than a framework developed for 10 people and for a specific industry. Agile at the Enterprise level needs Business Leadership. Leaders that want to improve the bottom line, the efficiency and profitability of the organization, not to tick a box that a certain framework was implemented or 50 people attended a 2-day training course and obtained a Certificate. For an Enterprise the success or failure is 'measured' by the market, by its clients and by its competitors.

5 SCALING AGILE

5.1 SCALING UP OR SCALING DOWN?

The real history of Agile demonstrate that there are two different paths to Agile. The Enterprise Agile that started as a scientific approach to improve Lean, to replace standardization with adaptability while keeping efficiency and the Team Agile, originated in Software Development as a way to improve unstructured processes used to develop software. Neither approach is a silver bullet and they are not exclusive. While Enterprise Agile brings science and ways to measure the impact on the bottom line, Team Agile brings energy, innovation and individual participation. There is not much new in the Agile frameworks developed following the Manifesto for Agile Development. Three of the values expressed in the MASD are not exclusive to Agile or will ensure Agility. Without processes individual interactions can generate conflict and chaos. Without tools processes can be inefficient and without clear engagement rule between providers and consumers there will always be some conflict at the end of the engagement.

The most important of the four values of the Manifesto for Agile Software Development, and really define what Agile is about, is the capability and capacity of a Team or an Enterprise to respond to unpredictable change. Readiness for Change, readiness to take Risks and learn from failures and mistakes is what can provide a market advantage, and make the Enterprise stronger in the market. An Agile Enterprise will thrive in a volatile market rather than fear change, stay defensive and miss opportunities.

Agile mindset, although completely opposed to Lean, doesn't mean that Lean and Agile are exclusive. They can complement very well and a good Project Manager will use practices from either approach, depending on the context. Scaling up from Team Agile (software development Agile frameworks) means in many situations implementing Lean practices, as demonstrated by 'scaled' Agile frameworks like SAFe, introduced as a Lean framework or Disciplined Agile where the first 'blade' DevOps is a Lean process. At the same time scaling down from practices defined for large Enterprise, the Manufacturing version of Agile, can benefit from practices defined for small teams, like Scrum ceremonies and artifacts. Scaling up or down and the practices that are the most effective can't be prescriptive, it always depends on context.

A good example where the context makes Lean a better option than Agile is DevOps. Despite being included in any Agile pitch, in Agile Training courses and Agile Certification frameworks, DevOps is a clear example of Lean practices used in software development.

- **One less process step**. DevOps basically eliminates a step in the "waterfall" process: Testing. Testing is not eliminated but rather included in Development and done by the Developers rather than a specialized Team
- **Automation.** A good DevOps pipeline is a fully automated process
- **Specialization.** DevOps is a software development practice

Whilst from the Agile Manufacturing perspective DevOps is a more Agile option for a small and specific subprocess of the product development: the deployment to production, DevOps it is a Lean implementation of manufacturing practices in software development. Analyzed from the Manifesto for Agile Software Development perspective, DevOPs fails all the core Agile values:

- **There are no individuals and no interactions.** DevOps is processes oriented with only automated processes and tools
- **There is no "working software".** DevOps is a suite of scripted 'pipelines', in most cases heavily documented
- **There is no Customer collaboration.** In fact, there is no customer involvement. It is a 100% technical activity, that is done by developers or in many cases by a specialized IT Team of 'DevOps' engineers.
- **There is no change,** therefore there is no easy response to change. DevOPs pipelines are optimized and standardized, any change to the pipelines must comply with a very rigid IT change Control governance

DevOps not being Agile doesn't mean that it is not good for Agile. Fast deployment means that changes, the real functional changes can be deployed faster and safer gives the end users the opportunity to provide feedback faster and for Developers more time to 'develop' the product and implement changes based on feedback from end users.

5.2 RISK AND CHANGE MANAGEMENT

Risk and Change Management are two core success factors. Despite good marketing and wishful thinking, Agile doesn't and should not reduce risk. Risk is introduced by change and where there is no change there is no need to be Agile. What many Agile Teams are not aware of is that Risk doesn't have a negative connotation only. Some organizations look at Agile as a way to reduce risk, by deploying in small increments, with the false assumption that (financial) risk is limited to the cost of one iteration. First of all, that's not true, a failed iteration (Sprint) can mean the cancelation of the project or product, either because the opportunity was missed or because the client is not happy with the outcome. Agile's 'customer collaboration over contract negotiation' means that the customer can easier terminate the engagement where there are no benefits delivered.

Change Management is another knowledge area that extremely important in any Transformation, is little known by Agile Teams. None of the Software Agile frameworks has recommendations for Organization Change Management. Most Agile teams believe that Change Management is limited to deployments to production, an area very well covered by the Lean practice of DevOps. Transition to Agile at the Enterprise level is hard, it is a very disruptive change and can be very painful. A good Agile transformation must include Organization Change Management and Scrum can very easily be used as a container for Change Management frameworks like ADKAR (Awareness, Desire, Knowledge, Ability and Reinforcement). If for a developer the lack of control, governance and in many situations, the relaxed commitment to delivery, is 'natural', for other teams like Finance and Procurement, a relaxed governance is not easy to understand and accept.

An Agile Transformation must start from the Top, from the Executive Level and it should be properly planned, resourced and managed.

As mentioned before,

6 CONCLUSION

Agile at scale, the Agile Enterprise is neither new nor originated in Software Development. The concept started in Manufacturing as a scientific approach to solve industry's slow response to market demands. The famous near perfect quality the Six Sigma level, had a price: standardization. Perfect products needed perfect processes, to eliminate waste, reduce the cycle time and deliver quality products at the lowest cost possible. In the last decades of the 20th Century the Manufacturing industry realized that something needs to change. They asked help from Academia and from the Government.

At the same time, the software developing emerging domain, needed some structure to improve efficiency. They tried to use processes that worked in large projects and failed. This failure led to attempts to define lighter processes that, inspired by the manufacturing literature, were called "Agile". IT is not a surprise that the most popular Agile framework used by small teams is Scrum, inspired by an article written by two Management Professors from Harvard. It is also no surprise that Agile Teams are rediscovering Lean practices, like Kanban and kaizen, and use them to make Software Agile efficient.

Those two parallel approaches to Agile are now merging into something new, something that, like Lean Six Sigma needs to start from the Executive level but unlike Lean Six Sigma, must be simpler, easy to understand at the shop floor and based on collaboration between all the tiers, teams and individuals in an organization: The Agile Enterprise.