SELINUS UNIVERSITY
OF SCIENCES AND LITERATURE

Doctor of Philosophy Ph.D by research in
Project Engineering Management. Final thesis:

Agile Project Management and new Leadership styles in successful largest companies

Research by David Hernan Tardini, 2021.

UNISELINUS
SBS
BUSINESS SCHOOL
SELINUS UNIVERSITY

Declaration:

I do hereby attest that I am the sole author of this project/thesis and that its contents are only the result of the readings and research I have done.

The thesis “Agile Project Management and new Leadership styles in successful largest companies” it is completely my original work and studies of many years of research.

Table of Contents / Bibliography:

Synopsis

During the last decades the level of uncertainty in the industry and the world economy has been growing exponentially. Large companies and organizations have had to change and adapted to an increasingly challenging environment.

Due to the great global increase of the Agile paradigm, many organizations are trying to get on the Agility train without having the minimum means and/or knowledge necessary to be able to implement it correctly, thus achieving mediocre, null or worse results, negative results.

Agility, out of theory from book, to real practical application of companies, is continually confronted with the chaos and routine of bad habits. For this reason, the daily work of the leaders, managers and sponsors of this initiative is of vital importance, who must be very clear that the transformation process is done with very small steps, so small that they do not suppose a high risk to the company or a threat to people, but with a constant cadence that allows to achieve the objective without resenting the Agile transformation process.

Several historic and traditional business models are fading while new players are disrupting the market taking advantage of new emerging methodologies and technologies to create superior customer experiences and competitive advantages. The world is transforming at a very fast rhythm, creating “great opportunities”, but also “great risks”, which require constant adaptation to change.

Today, technology and agility have become the new paradigms at a global level. The needed to implement Agile project and product management methodologies has increased exponentially. Several companies have experienced the limitations of traditional software / product development methodologies (Predictive / Waterfall), such as: long delivery times, low user satisfaction, increased costs, inaccurate estimates, etc.

The Agile paradigm is an enterprise philosophy, a business identity culture, whose benefits extend to the entire organization promoting the creation of value in all areas to deliver the best products / services to our clients in an efficient and sustainable way..

Embrace change

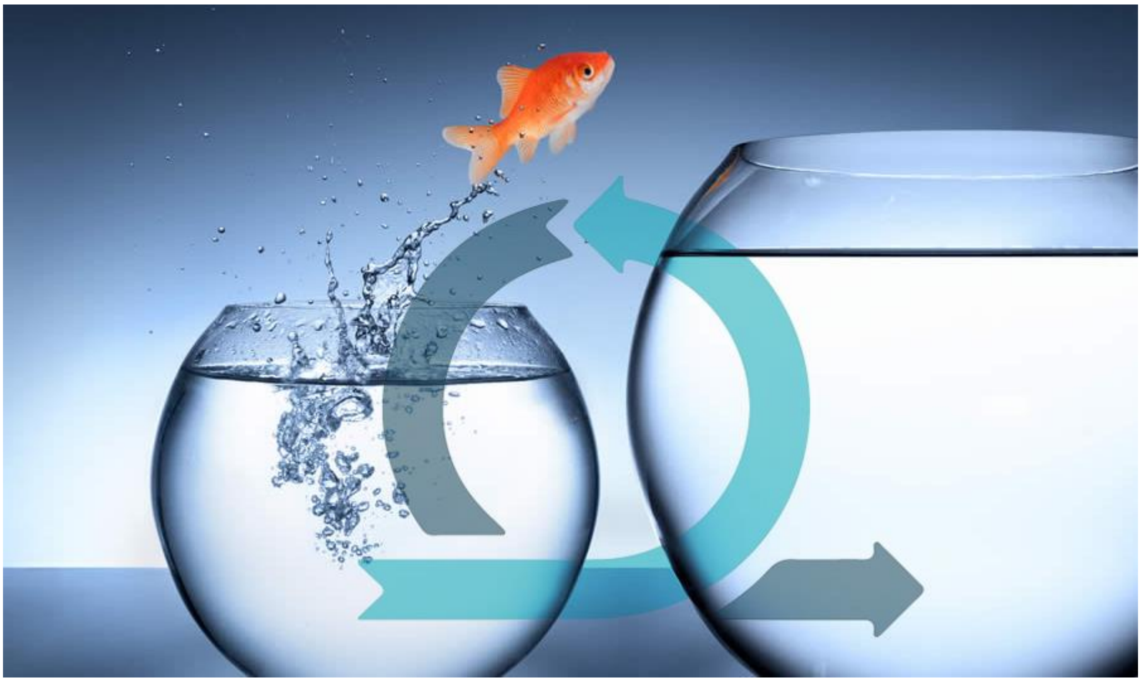


Image 1, source and credit: www.betagov.org plus my own design.

Chapter 1

The origin of Agile.

In February 2001, at the ski resort "The Lodge at Snowbird" in the Wasatch Mountains of Utah, they gathered seventeen experts in software development improvement models process-based, convened by Kent Beck, who had published previously the book: Extreme Programming Explained.

At this meeting, the term “Agile Methods” was coined to define the methods that were emerging as an alternative to traditional methodologies known so far, to

which they considered excessively "heavy" and rigid due to their normative character and strong reliance on detailed planning in pre-development phases.

The members of the meeting were able to summarize the values of the

Agility in four postulates and twelve principles, which has been named as:

Manifesto for Agile Software Development

“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.

That is, while there is value in the items on the right, we value the items on the left more.

Source: www.agilemanifesto.org

Kaizen and Lean.

Kaizen (改善) is an ancient philosophy that promotes the technique of taking small steps to achieve important goals and continue to improve the process. It is applicable both in business and staff (especially as a coaching tool).

The first approach to improving small-step work was developed in North America (USA) according to the training program within the industry known as TWI (Training Within Industry).

The primary approach is this: to introduce small improvements that can be implemented daily without much effort to achieve the objectives. This avoids having to make big radical changes in a short time, leading to high risk, work stress and uncertainty among people.

Kaizen's origin.

After the end of World War II, U.S. occupation forces in Japan assigned experts to help rebuild Japanese industry, which had been devastated by the war.

To achieve this, they developed a management training program where statistical management and control methods were taught, led by Homer Sarasohn, Charles Protzman and subsequently by Wiliam Edwards Deming, who has made one of the most significant contributions with the Deming PDCA cycle method (Plan, Do, Check, Act).

The PDCA cycle:

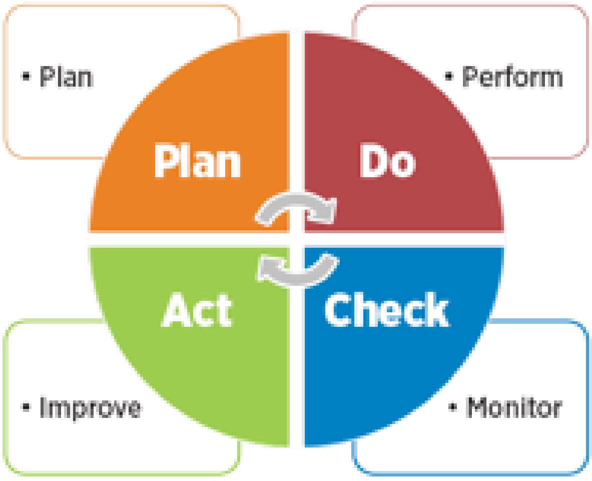


Image 2, source: www.asq.org/quality-resources/pdca-cycle

All these new concepts have been very well assimilated by Japanese managers and directors, who have immediately implemented them in large corporations, thus giving rise to what is now known as the Kaizen continuous improvement strategy, which has led Japanese industry to be among the world's major economies.

The Kaizen concept should be extended to everyone in the organization and even stakeholders, external partners and even customers (through feedback).

This would allow the company to have a wide variety of views, always be on the pursuit of excellence and achieve high quality standards in products and services.

Contributions of Kaoru Ishikawa

Picture 1, source: www.shmula.com

Kaoru Ishikawa born in July 1915 in Tokyo, Japan.

The concept of Kaizen was taken by Kaoru Ishikawa, who has managed to define the continuous improvement (kaizen)applied to industrial processes and identify all the variables involved in it.

He has developed the concept of 5S, which describes how to organize a workspace to achieve efficiency and efficiency, using decisions from employees directly involved in the process.

- Seiri (整理): classification.
- Seiton (整頓): order.
- Seiso (清掃): cleaning.
- Seiketsu (清潔):Standardize.
- Shitsuke (躾): discipline.

Contributions of Taiichi Ohno

Picture 2, source: www.history-biography.com/taiichi-ohno

Taiichi Ohno born in February 29, 1912 in Dalian, China. He aimed to eliminate waste in the part of the production process for which he was responsible.

These concepts have resulted in the Toyota Production System (TPS) which has been developed by Taiichi Ohno in collaboration with others during the 1940s and 1970s.

Many concepts and components of this system have become familiar in the West:

- Mute: waste disposal.
- Jidoka: quality self-control.
- Genba– the place where you create the product.
- Kanban: visual management method for just-in-time inventory control (JIT).

Sakichi Toyoda, Kiichiro Toyoda and Taiichi Ohno have been the people who exceled in the creation of the Toyota Production System (TPS).

Taiichi Ohno has been the driving force for the development of the system and its implementation in practice, leading by example and training all employees.

Some quotes from Taiichi Ohno:

"Progress cannot be generated when we are satisfied with existing situations."

"Not having problems is everyone's biggest problem."

"Toyota style is not creating results by working hard. It's a system that says there's no limit to people's creativity. People don't go to Toyota to work, but to think."

The term Kaizen is *usually associated* with Toyota's vision.

The importance of respect for individuals shows that the company cares about all its employees and values all the contributions they can make to the company.

Along with continuous improvement *(Kaizen),* people know that not only are they trained for their daily tasks, but are expected to continue learning and evolving, allowing all employees the ability to grow in the company and help the company use the full potential of each to become more profitable and successful.

One of the main premises of this concept is that it is always being learned. The concepts of *Kaizen (continuous* improvement) and *Shu-Ha- Ri (Learn, Detach,* Transcend) can be combined to achieve excellence and transcend beyond it.

It is necessary to adopt a beginner's mental state, to have an open mind and high motivation to be able to incorporate the new lessons that are available and at our fingertips.

Often, the main reason why people do not take greater initiatives at work is because *managers* and/or managers do not consider the opinion of the people directly and/or indirectly involved in the project, task and/or service.

One of the biggest benefits of this culture of continuous learning is that it generates greater collaboration and less rivalry among people, because a team that knows that their ideas and suggestions will be heard and valued will be more predisposed to take minor risks and boost creativity.

Encouraging and giving freedom to create

Too much control of *managers* over people and events produces a sense of widespread frustration. This is commonly referred to as "*micromanagement* ". Micromanagement nullifies the creativity of people and teams, making highly qualified and proactive people just simple workers who simply fulfill orders and tasks assigned to them.

One of the keys is to trust and build trust in people

Often, unconsciously, directors and managers can significantly weaken the relationship of trust with their employees, and it must be recognized that once it is lost, it is very difficult to recover it.

Trust is a very fragile but very valuable asset, because it can make the difference between one person who is emotionally committed to the company with another who is indifferent, and/or even become an impediment to the company.

Generally, in companies trust in people is acquired through tangible and measurable results (high employee performance, good economic results, brand loyalty, etc.).

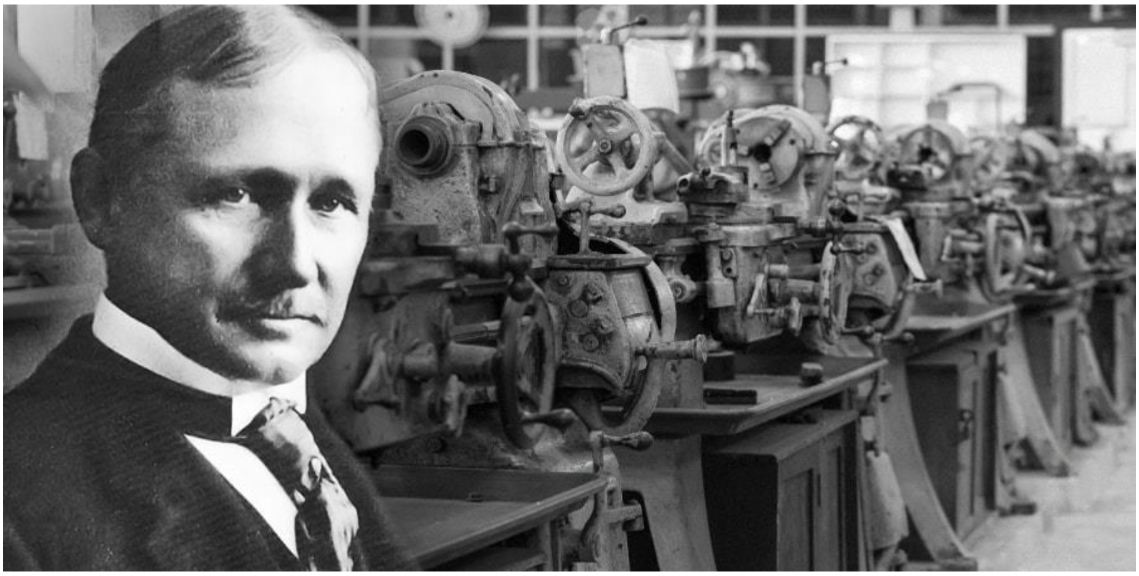
Giving people autonomy is very important to foster creativity. Low-impact decisions, which do not pose a risk to business continuity, should be decentralized and not require intervention from managers or senior managers. Pretending that everything needs authorization from a manager and/or having control of all decisions, can cause "*bottlenecks"* that penalize the time to release a product into production (*Time to Market*).

It may happen that initially this autonomy entails uncertainty among people, especially if the company's policy is to penalize failure. For this reason it is important to understand and accept failure as an opportunity to learn from experience and improve in future actions.
There is a case that some employees do not want to have autonomy or the possibility of making decisions, since they have their position as executor of

tasks. For this reason, it is very important to know how to discern which people should be given autonomy and who should not.

To achieve high-performance team, trust is an essential factor. When everyone on the team feels safe with each other, they won't be afraid to make risky decisions to achieve goals and discuss potential vulnerabilities that jeopardize them. In addition, trust fosters innovation, creativity and collaboration among all team members, thus achieving the unity of excellence.

Contributions of Frederick Taylor



Picture 3, source: www.qad.com

Frederick Winslow Taylor was a mechanical engineer and economist born in Philadelphia, USA, a pioneer in the principles of scientific management.

Frederick's efforts to apply engineering principles to factory environments become what we know today as industrial engineering. In the 1890s, he began his research on working methods and employees, resulting in proposing the first concepts of job standardization, time studies and movement studies, to achieve efficiency in methods, processes and operations.

The principles of scientific management

The principles of scientific management describe the basis of modern organization and decision theory.

In 1911, Frederick Taylor summarized all his techniques in *the book The Principles* of *Scientific Management,* which has been recognized as the most influential management book *of* the 20th century by The Academy of Management Fellows (A.O.M.)

In this innovative management method (for that time), Taylor set out four main principles:

1. Replace general standards working methods with methods based on a scientific study of specific tasks.
2. Select employees based on their skills, train them and develop them scientifically rather than passively letting them train themselves.
3. Provide detailed instructions and monitor each worker in the performance of their task.
4. Divide work almost equally between managers and workers so that managers apply scientific management principles to work planning and workers perform tasks.

Scientific management is a management theory that analyzes and synthesizes workflows to improve economic efficiency, especially labor productivity, often called Taylorism.

It can be said that some of the principles of the Lean Six Sigma methodology are related to the work of Frederick Taylor.

The Enterprise Agility in the actuality.

During the last decade, Agile is the predominant methodology for product development and project management, especially in new IT companies and startups, and also in companies with a long history that have adopted the Agile philosophy due to its high flexibility and adaptability for the products and services in in a fluctuating market with high uncertainty as has happened in actual times.

Many other organizations, with a more traditional approach and culture, are constantly struggling to implement the Agile paradigm at a slow but stable pace, because the risk of a change of this magnitude is high in such organizations.

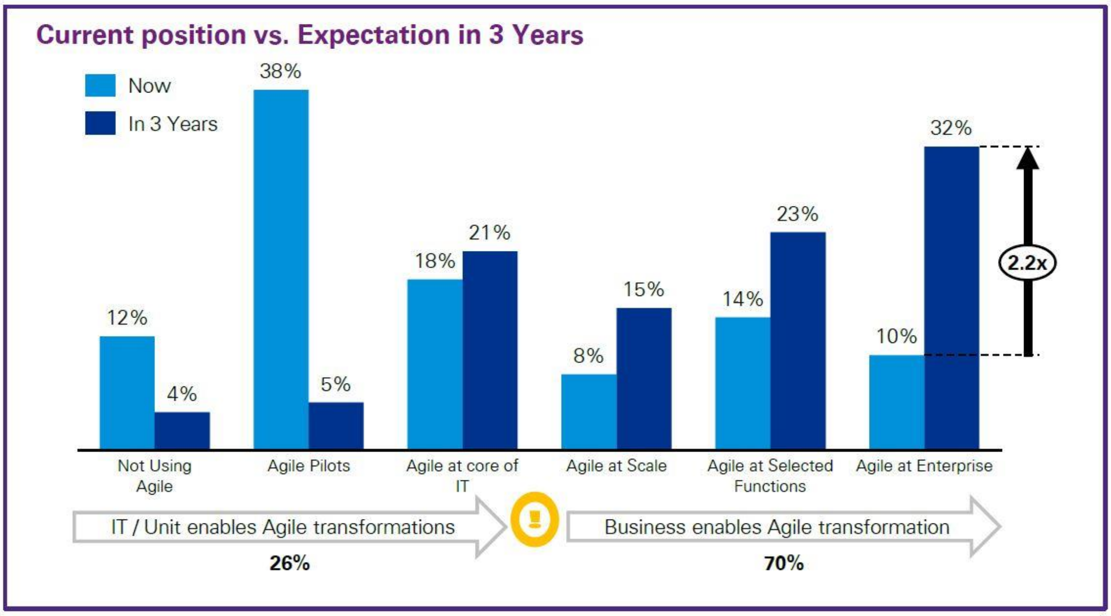
Meanwhile, other Agile native technology companies are creating constant value in their products and services by focusing on the customer. Keep in mind that the risk in such companies is much lower, so you can afford to make mistakes, learn from them, and start over from the lesson learned.

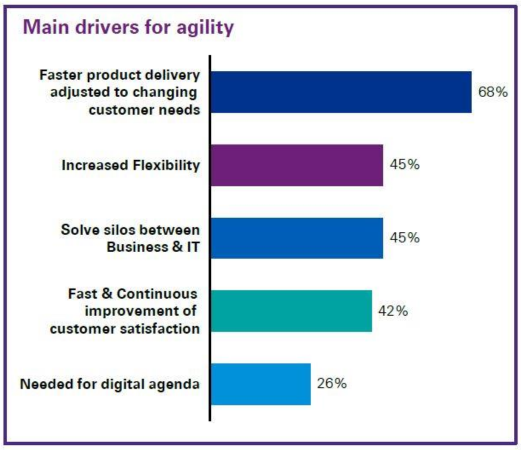
Large corporations bring with them the legacy of a process-based structure, comprehensive controls and constraints inherited from systems, where the greatest asset was the developed product. Surely, this model ten years ago was very effective and provided a high profitability, but in today's times, where the market is constantly evolving and changing, this represents one of the key points to improve to bring that value back to the customer.

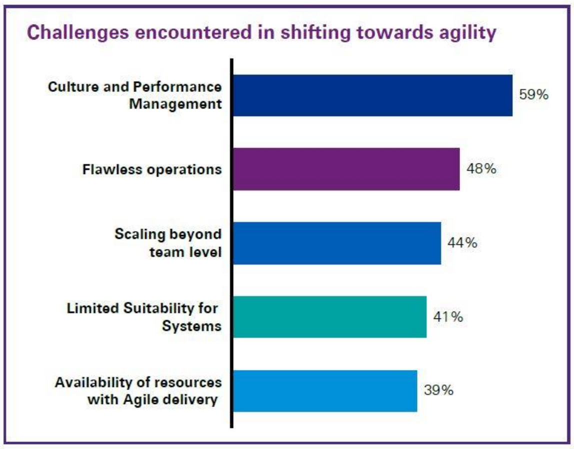
For this reason, traditional companies need to evolve, adopting the values and principles of Agile, incorporating more flexible methodologies that allow efficient management of culture, people, projects, relationships with stakeholders and clients. But you have to be very careful not to implement the Agile Paradigm abruptly and with very radical changes, because it will most likely generate an adverse effect on the organization, obtaining a great resistance to change in people with unpredictable results, which will bring a lot of frustration and probably end up abandoning the initiative.

“ We see Agile being scaled with 70% of respondents indicating an ambition to integrate both Business and IT enabled Agile transformation in the next 3 years. The most used framework among respondents for scaling is SAFe. We also see Agile quickly outgrowing front-end IT development into business units and becoming an integrated approach to deliver business value across the entire value chain.”

Source: KPMG Global Agile Survey 2019 - 2020
https://assets.kpmg/content/dam/kpmg/be/pdf/2019/11/agile-transformation.pdf

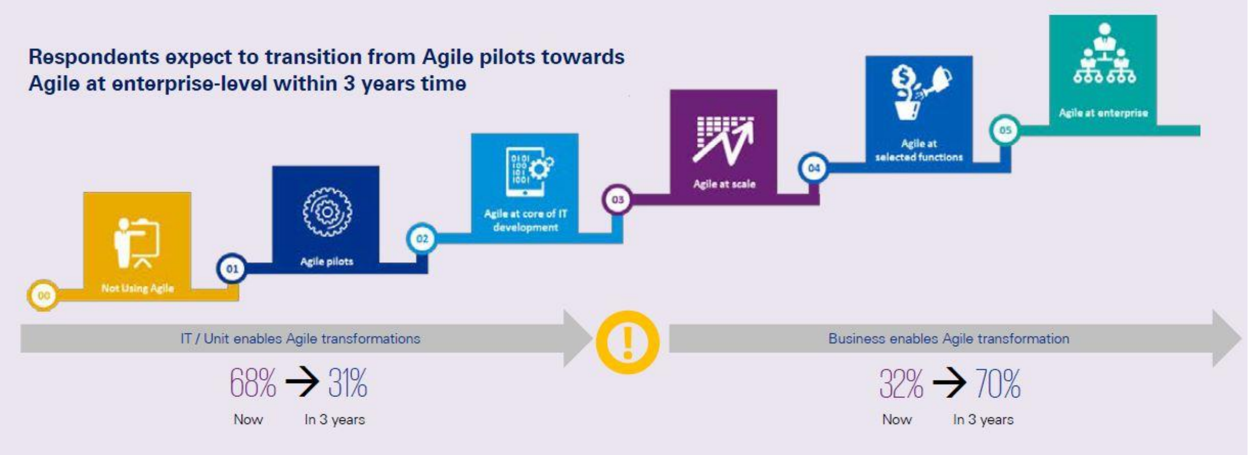






Source: KPMG Global Agile Survey 2019 - 2020
https://assets.kpmg/content/dam/kpmg/be/pdf/2019/11/agile-transformation.pdf

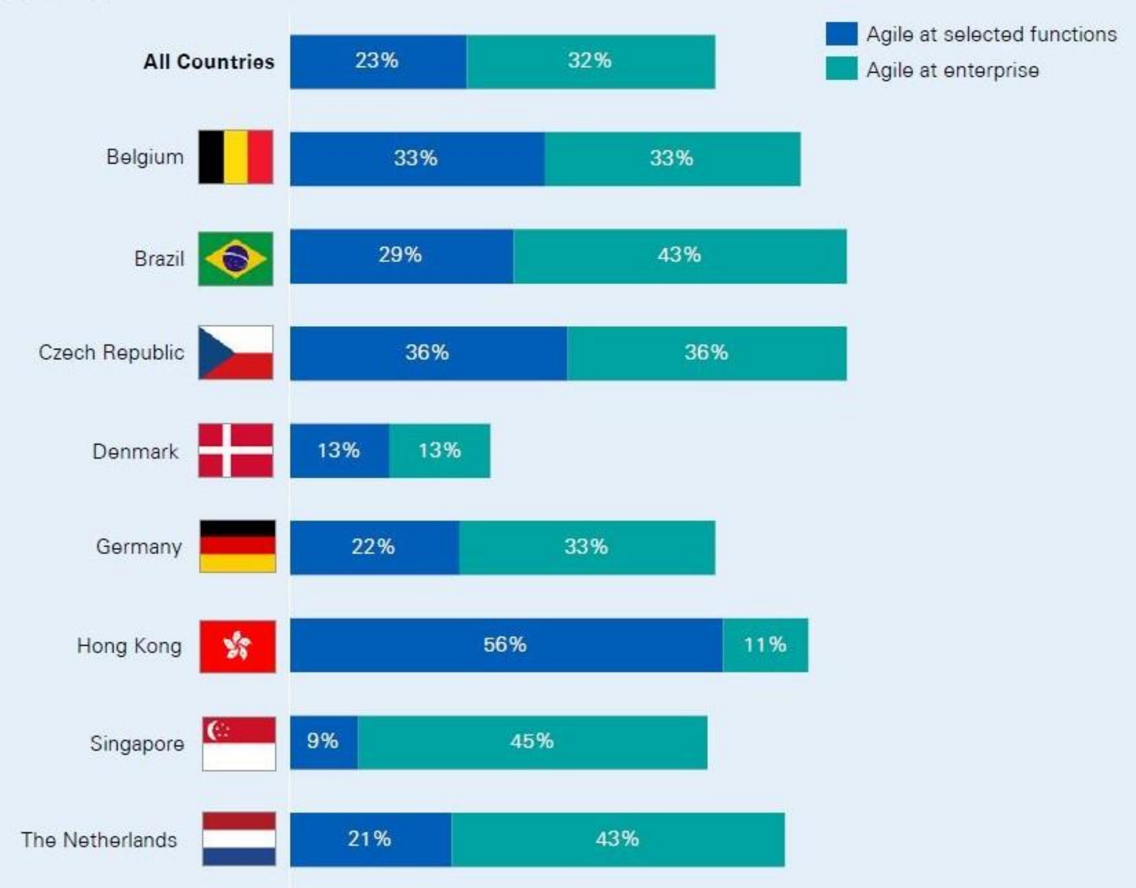
Worldwide largest companies expected to transition from Agile pilots towards Agile at enterprise-level within 3 years' time.


Respondents expect to transition from Agile pilots towards Agile at enterprise-level within 3 years time
00
Not Using Agile
01
Agile pilots
02
Agile at core of IT development
03
Agile at scale
04
Agile at selected functions
05
Agile at enterprise
IT / Unit enables Agile transformations
68% → 31%
Now
In 3 years
Business enables Agile transformation
32% → 70%
Now
In 3 years

Current Position vs. Expectation in 3 years

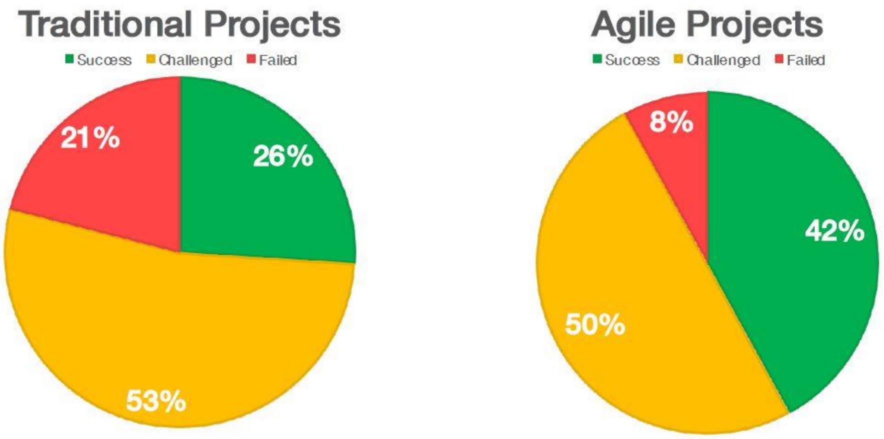
Respondents from different countries (2019/2020) reported varying ambitions to scale towards Agile at enterprise or Agile at selected functions.

"Where do you expect your organization to be in three years?"



Source: KPMG Global Agile Survey 2019 - 2020
https://assets.kpmg/content/dam/kpmg/be/pdf/2019/11/agile-transformation.pdf

Project Success Rates



On average Agile Projects have a 42% success rate compared to 26% for traditional projects during 2013-2017 .

Source: Jim Johnson, Standish Group, Chaos Report, 2018

Chapter 2

Value delivery and focus on persons.

The cultural adoption of methodological and technological developments has been a complex and difficult topic. It is not easy to adopt new paradigms at the cost of abandoning the comfort zone that we all possess.

When immersed in a highly specialized technical and methodological ecosystem, it is difficult to observe, acquire, test and implement new improvements and paradigms.

New technologies and automation provide workday reduction options in a scenario that will create new jobs while eliminating others.

The same educational structure is observing its methodological obsolescence in a world that advances much faster than it can assimilate.

Now more than ever, people need to lead and manage this new revolution with their ideas and talent. It is beginning to visualize that the next industrial revolution is on its way to becoming the age of people.

If we put people at the center, we will immediately look to the planet that supports their existence and show that something different and sustainable must be done. To seek the well-being of the community is to work for personal well-being.

A company is just a private community of people working together for a common purpose. A person does not perform a service but collaborates on a project.

In agile companies with a culture of innovation, a person can collaborate on as many projects as they want, and their field of performance will not be tied to what we know as a company but will develop in one or more communities of interest.

One implication of this methodology is that people of great talent or knowledge will be able to collaborate with various communities, allowing to increase access to talent and knowledge.

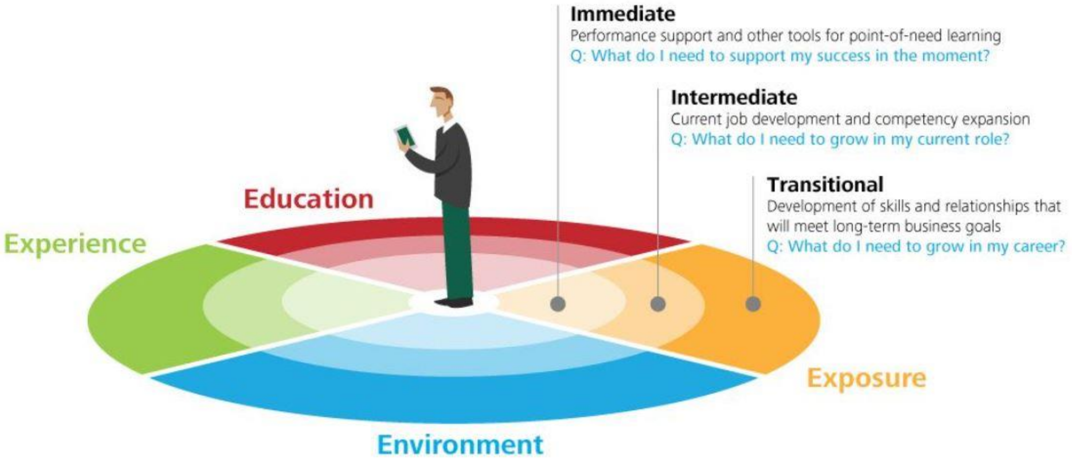

Immediate
Performance support and other tools for point-of-need learning
Q: What do I need to support my success in the moment?
Intermediate
Current job development and competency expansion
Q: What do I need to grow in my current role?
Transitional
Development of skills and relationships that will meet long-term business goals
Q: What do I need to grow in my career?
Education
Experience
Exposure
Environment

Image 3, source: "The learning architecture"

https://www2.deloitte.com/us/en/pages/finance/articles/cfo-insights-continuous-learning-environment.html

People learn because they explore, get it wrong, and learn. Fear of failure is the antithesis of learning. Also, one of the best ways to learn is to teach and share.

The chosen methodology must ensure the ability to evolve without taking big risks.

A broad and respectful vision based on information and global objective data will make us make better decisions.

Understanding the psychology of group transformation processes.

Given the different situations in which those who adopt the strategies to tackle Agile transformation projects are involved, it is noted that one of the main tasks is working with people who are in different structures and hierarchies within a given company.

The group of people who are part of that company or institution, when developing their activity, are faced with the need to work with other people and, due to this interaction, ties and connections are established according to a designed and established organization chart that is being developed or that it may be in development.

In this work with others, generally sectorized, and largely consolidated for some time, we see how a company or institution is organized, how it lasts over time and how it develops its activities and resolves its conflicts.

It is very important then, for anyone who wants to interfere or is called to undertake a change in an institution, be clear about how it works at each of the levels and how all the parts within the organization are articulated with each other.

To begin with a task of total transformation or of some of the departments, it is essential that those who are going to carry out this process have a deep understanding of the operation of each of the groups and of the people and how they are articulated with each other.

This field work, which is made up of the collection of data and information on current operations - the ways of organizing, how the different departments work, the work organization charts and who assumes these tasks - is essential when planning future interventions in the process of dynamic transformation.

One of important issues to consider in the training processes is that we must consider a few previous steps, very delicate but indispensable to achieve any approach when being able to fully develop our entrepreneurship.

In a transformation process we do not start from scratch. There are already structures created that are the ones that we will want to transform, modes of organization that pre-existing – called organization culture, paradigms, organizational systems, etc.– and on those frameworks we will focus our work. This involves two phases: first, we will have to understand how the current structure has been reached and how it works, and secondly, to lay the groundwork for a path of destructiveness.

Let us now look at what this process of deformation or deconstruction consists of based on the analysis of the psychological mechanisms involved in the formation of groups, because if we shed light on its constitution we will be able to take the reverse path and access to transformation.

We start from the premise of understanding that a group is not the same as a gathering of people.

A group consists of individuals, individuals who share certain characteristics and objectives and relate and interact with each other and with other groups or isolated persons for a certain period and for a certain purpose.

It is not tangible, but rather exists in the minds of individuals.

If we sharpen our gaze a little bit, we find that by studying the relationships that occur within the groups, we are referencing individual relationships. While individual psychology sticks to the singular human being and studies the paths by which he seeks to achieve his or her pulse satisfactions, rarely, and under certain conditions of exception, an individual can do without links with others. In all people's moods, the other counts as a model, as an object, as an auxiliary, or as an enemy. In relationships with parents, siblings, with the loved one, friends, teacher, co-workers, the boss... the individual experiences influence that are becoming hugely important to him.

We might wonder, then, what instances occur in that group entity that possesses the ability to influence the life of the individual and what is that mood disturbance.

Let's notice an important trait that manifests itself in the individuals who make up a group. The mere fact of being within one endows them with a kind of collective soul under which they feel, think and act differently than they would in isolation. There are ideas and feelings that only emerge or become acts in individuals linked to a group. So, if group individuals are tied to a unit, there must be something that unites them or encourages that union.

Another cause that fosters the union of the members of a group is the closeness and trust that occurs among its members, which causes the groups to externalize special features and mark the orientation of them.

An organized group works homogeneously when members have the same object. This makes it possible to positively strengthen the links between the members, since being identified to the same ideal, they all have the same commitment to achieve the objective.

Agility and theory of the dissemination of innovation.

In 1962 Everett Rogers, a sociologist and professor at Michigan State University (USA), published his Theory on the Dissemination of Innovations.

This theory explains the process of dissecting new ideas, concepts and products that contribute to the modernization of society.

Disseminations the communication process that aims to socialize new ideas and inventions among members of a particular social system, at a given time and through various channels of communication.

However, it is not enough to know the innovation, but you also must adapt it and get acquainted with it. To do this, there are 5 attributes that must be considered:

1. **Relative** advantages: the idea or product must show that it is beneficial against the absence of others in its kind or against existing ones, which would reduce the risk when opting for change.

2. **Possibility of observation**: the benefits must be visible and tangible to the user.

3. **Compatibility:** Innovation must be in accordance with the context and reality of the society where it is to be implemented.

4. **Complexity:** the degree of difficulty in learning, managing and maintaining innovation should be considered.

5. **Possibility of testing**: innovation must have the ability to be tested before being approved or used.

Having the presence of all or most of these elements will give a greater opportunity to adopt proposed innovations.

The approaches of this theory were considered in the development processes by several companies and entrepreneurs who realized that to promote innovation is essential the socialization of knowledge, since the experiences lived had a significant influence on people to accept or reject innovative proposals / products.

So, according to theory, human behavior has had to go through 5 stages:

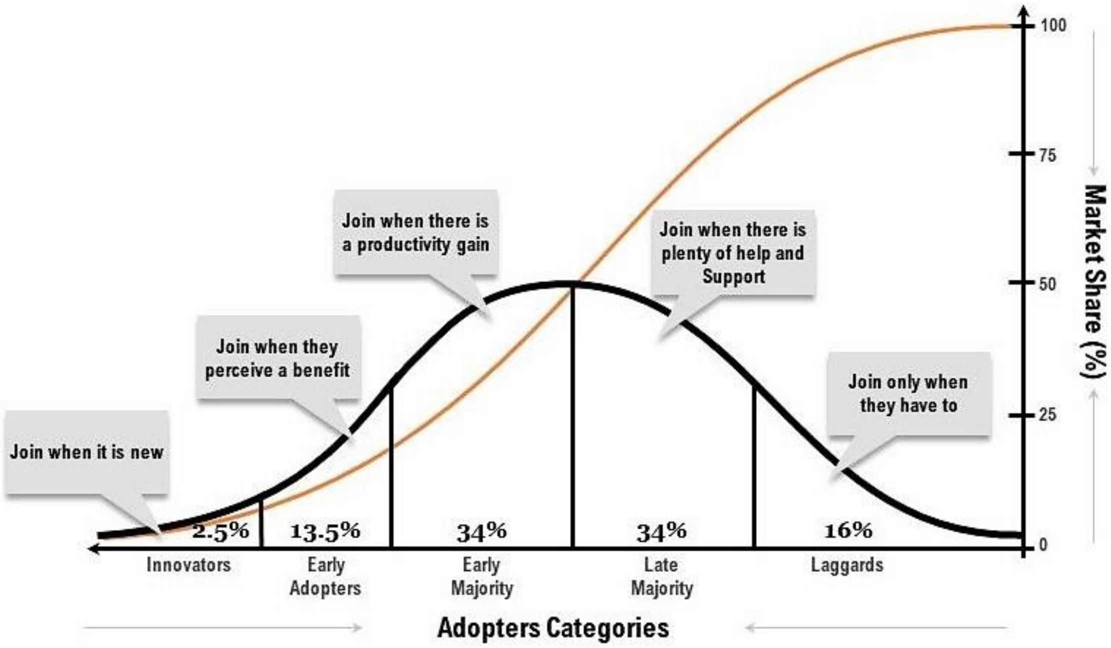
1. **Perception: it** is the first approach of the person with innovation, it is where he acquires the first knowledge and understanding of it.

2. **Interest:** a critical attitude towards innovation develops, positive and negative aspects of it are determined.

3. **Assessment:** after analyzing the pros and cons, it is agreed to accept innovation, or else it is rejected, and the process is terminated at this stage.

4. **Implementation:** This is the period in which innovation is tested before it is fully adopted.

5. **Adoption:** this is the final confirmation and acceptance stage. This stage is the most important because success lies in the commitment of people who have adopted the new idea/product to use it continuously overtime and thus allow to maintain the innovation process until its full development.



Time-based graph of the distribution of adoption capacity.

Source graphic: https://stuart-mcintyre.com/reviving-social-business-adoption

Innovators: persons ready to take new risks. They are socials and popular people and have the closest contact with scientific sources and interaction with other innovators. They generally have higher social status and financial liquidity, allowing them to have greater risk tone to adopt technologies that can ultimately fail.

Early adopters: These people have the highest degree of opinion leadership among all categories. They are a little more discreet in adoption than innovators and that allows them to maintain a central communication position with the entire community.

Early majority: innovation is doubled after a significantly longer time than *Innovators,* thus minimizing risk in the event of failure.

Late majority: these people approach innovation with a high degree of skepticism and after most society has adopted innovation.

Laggards: they are the last to adopt innovation. These people often have an aversion to agents of change. They generally tend to focus on "The Traditional" and almost all decisions are made in terms of previous generations. An innovation that is finally adopted by a straggler is already obsolete because surely during that time there are already proven innovations that are already being used by the early majority.

Conclusions:

To bring Agility to an entire organization, it is necessary to obtain the sponsorship of the executive committee, to know how to explain in a clear and concise way the benefits that this transformation will bring and to carry out an analysis of the risks that it entails. Many times, this process is also known as the digital transformation.

A very significant and high-impact change in business processes may seem like the fastest and easiest way to innovate in a company, but this also carries the greatest risk.

Before addressing drastic change at the corporate level, it is very important to start with more controlled environments, where low-impact decisions can be taken to quantify risk without affecting business continuity. It should be noted that innovation and agility are processes that are in continuous improvement and not a final solution.

Chapter 3

“The New Normality” and the Impact on companies.

For the vast majority of organizations, the challenge created by the health emergency has been to have to implement remote work in record time, solve the situation of being able to continue operational and thrive in the face of competition in order to stay in the market.

One of the most successful formulas to address the problem is the effective use of distributed Agile team. New data that has been observed globally shows that people who work properly remotely can remain connected, productive and aligned to the company's current goals, as well as working in person.

Distributed Agile Team.

It's very important to note that a distributed Agile team isn't just a collection of people working remotely. You need to create a virtual identity with a different mindset to achieve high-performance teams.

Teams must build strong communication through collaborative tools, they must learn to coordinate and provide maximum transparency through the events it proposes, for example, Agile Scrum.

Remotely distributed teams can be equally productive as teams working in person in one place. At first this requires much more discipline and probably a different set of tools that allow collaboration remotely and virtual work sessions.

Communication between distributed teams.

Video conferencing services are easy to implement and generally low cost, and some are even free (in basic version). It is very important to understand that human beings process much of the information obtained visually.

Much of our brain has evolved to read the emotions and thoughts visible on another human being's face.

Delayed tasks must be displayed transparently and accessible for patches. All Agile equipment must be able to inspect clearly to understand the true reason for the delay and visualize the impact on the Product.

Continuous collaboration between Stakeholders, Product Owner, Scrum Master and Development team is essential and must be real-time.

Product Backlog items must be clearly expressed in the collaborative tool chosen for management (Jira, Zepel, CA Rally, etc.) and it is also very important that the "Definition of Done" (DoD) and Sprint Goal are clearly available to the team at all times.

Human factors to consider

It is probably that on certain occasions the most important thing a person will have to deal with is the fear of loneliness, the feeling of isolation and the concept of "Non-Belonging". To do this it is very important to know how to detect on time and work with the affected people. The Agile Coach /Scrum Master together with other professionals in the organization can collaborate to try to improve this situation.

Agile project management in remote.

Due to paradigm shifts from the "official" workplace and co-working centers, remote work has become the current norm for many companies, which has passed suddenly and surprised many companies.

Some organizations were not prepared for this sudden change and have had to adapt quickly, both in collaborative infrastructure and tools, as well as in create a methodology to ensure maximum value delivery and with optimal quality customers.

One of the usual situations generated by remote work is that "the number of required meetings increases exponentially." On several occasions,

during the meeting, there is a lot of talk about what tasks should be done and how to do them, and after the meeting has ended there is no time to execute them because you have to connect to the next meeting, and so on.

How to deal with this situation.

As a rule, all virtual meetings should now be sessions of

productive work. That is, at the end of it, you must get something tangible that move the team towards a specific outcome.

This could be the elaboration of a presentation, a document, a working agreement, Etc. That is, if you work remotely, the vast majority of meetings should collaboratively workshop sessions, where co-creation is encouraged with all stakeholders to develop tangible solutions.

One way to help teams achieve this mindset is to establish a definition

(DoD), it presents and is shared at the beginning of the meeting with all

Members. It must be ensured that all participants understand what the DoD is and how will be achieved so that each participant focuses on the task(s) needed to achieve this.

It is also necessary to block the calendar when each of the people on the team undertake an emerging task that has come out of the previous meeting, at which it has been agreed, as a second phase, the tasks to be performed by each member alone or in small smaller virtual groups.

We may encounter the situation that some teams are frustrated by situation and/or people who are blocked working from home or in places remote away from your computer; For this reason, it is the work of the Agile Coach / Scrum Master, identify these situations and solve them as soon as possible to minimize the impact organization and value delivery.

Achieving maximum efficiency in the way you work remotely is not an easy task, requires a lot of discipline, focus and concentration, especially if the teams in question they still don't have enough maturity in Agility.

Coordinated Agile Teams in remote.

Sprint remote planning

Proper alignment and coordination of teams is a critical function in an environment Agile distributed. Being clear about the Sprint Backlog emerging from planning gives you to each team member a shared understanding of the goal to be achieved.

The Daily Scrum

It is the essential point of contact for team members working

Distributed. That's where agile Scrum team members give clarity on the

real-time status of work and impairments (if any). In addition, it is

a daily opportunity to re-inspect and adapt according to the latest comments,

information, impediments or emerging changes. In today's high times

uncertainty is a practice that brings a lot of value to teams and the organization.

Sprint Review remotely.

It's time for the team to show the work done during the Sprint to the customer,

Product Owner, Stakeholders. Doing it remotely is no easy task, it requires a

thorough pre-preparation and millimeter coordination. The team(s) will get a

Very valuable feedback that will make sense of the effort made. It's also important understand that the increase may not be accepted, but this should not diminish the morale of the should be understood as an opportunity for product improvement.

Sprint Retrospective.

It is an opportunity to identify activities to achieve continuous improvement. But you must keep in mind that in a situation of isolation, this event is vital importance, especially for teams that are not used to working remote and suddenly seen, pushed into a virtual world. The process of creation has to be thoroughly examined, iterated and improved. People must learn to work and collaborate differently. But also when things work well, they should be held and reproduced on other equipment, whenever possible.

Adversity often strengthens Agile teams and the entire organization allowing him to thrive in difficult times.

The importance of communication balance.

It is very important to understand that in many situations reducing communication and amount of information is better.

That is, not by having all the information about everything that happens in the company, the teams they're going to be more effective.

Communication must be essential and simple enough to keep the high team performance, because having a greater number of comprehensive meetings and having to generate more reports cause higher decision latency being this one of the causes of the failure of several projects.

Correctly implementing Agile Scrum will radically reduce the need for constant communication while ensuring effective communication between teams and organization.

In the case of the Product Owner who is responsible for articulating clearly the "What" and intent of the mission, if the team is able to understand and quickly execute your vision with minimal delays, you will reach a state of high Performance.

Keep in mind that, at an early stage, extensive meetings are required to train teams that work aligned to a common goal and until this does not happen, teams can't perform.

For this reason, at this early stage, proper communication is essential and will often be tedious, until all the teams are aligned. But once in operation, the communication must be minimal, essential and fluid to reduce latency time in decision-making which will enable high-performance equipment to be achieved and maintained.

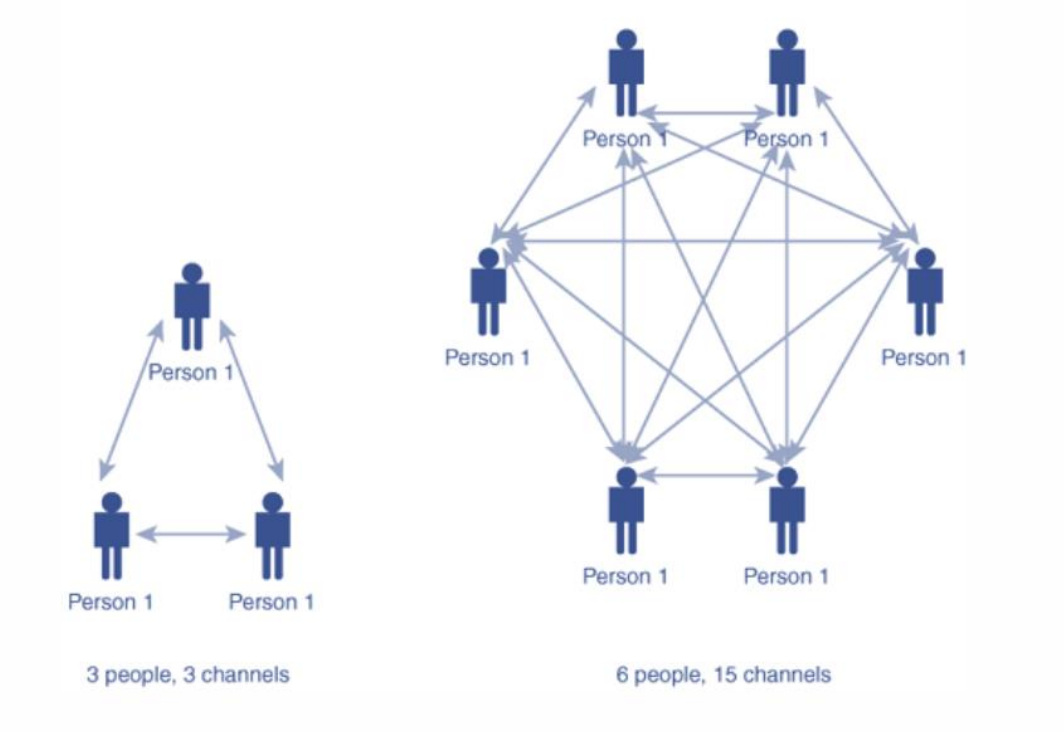
Communication Channel Formula

Communication Channel : This is the total number of potential communication channels between the Stakeholders.

The communications channels formula is N * (N-1) / 2

In the example bellow N= 6 Persons => 6 * (6-1) / 2 = **15 channels**

It is a way to numerically show the importance of proper communications management on a project.



S

Image 4, source:

https://martinsitconsulting.com/wp-content/uploads/2019/09/Communication_channel_black_white.png

Agile metrics and the importance of knowing how to measure it right.

Agile metrics contribute excellently to measuring team results and the continuous Agile transformation process. What needs to be clear is precisely the latter: that Agility is an ongoing process in search of excellence, where the whole organization must always be a part of it and be aware of it .

Metrics must be dynamic and continuously evolving toward new goals; that is, what we are measuring today over the next six months may not add the value needed for the company's future goals.

One of the most used is Velocity. The speed of delivery of the equipment is a very useful indicator, because what underlies this indicator is the empowerment of the equipment, that is, the equipment that makes its deliveries with a speed and quality superior to the rest are high performance equipment that have had to go through all stages of the Bruce Wayne Tuckman model :

- **Forming**
- **Storming**
- **Norming**
- **Performing**

Later, in 1977, in the joint work with Mary Ann Jensen, he added a fifth stage to the four stages: **Adjourning.**

Throughout this process, the agile coach's work to get the best out of each person and build high-performance teams that are sustainable over time is vital.

One of the key benefits of Agility is Time to Market, that is, the ability to market new products and/or functionalities that, although not yet fully mature, come to market with a few functionalities to receive rapid feedback from customers and see if it is necessary to incorporate new functionalities required by users in a second version, or if, on the contrary, the product has not had the expected impact and its development must be canceled.

Most important value metrics to companies

Predictable Performance

Stable velocity

Release burn-up

Cycle time

Lead time

Degree of automation

Productivity

Time to Market

Solution Quality

Ticket volume evolution

Product Health

Code quality

Defect density

Story points per work type

…

Ability to Innovate

#new ideas proposed

#ideas that made it to production

…

Business Value

End user satisfaction

Client Success Factors (i.e. collaboration intensity, common ways of working)

KPI on Business objectives (i.e. revenue, #of orders, order failure rate)

…

Motivation & People

Team morale

Business Owner morale

Employee training level

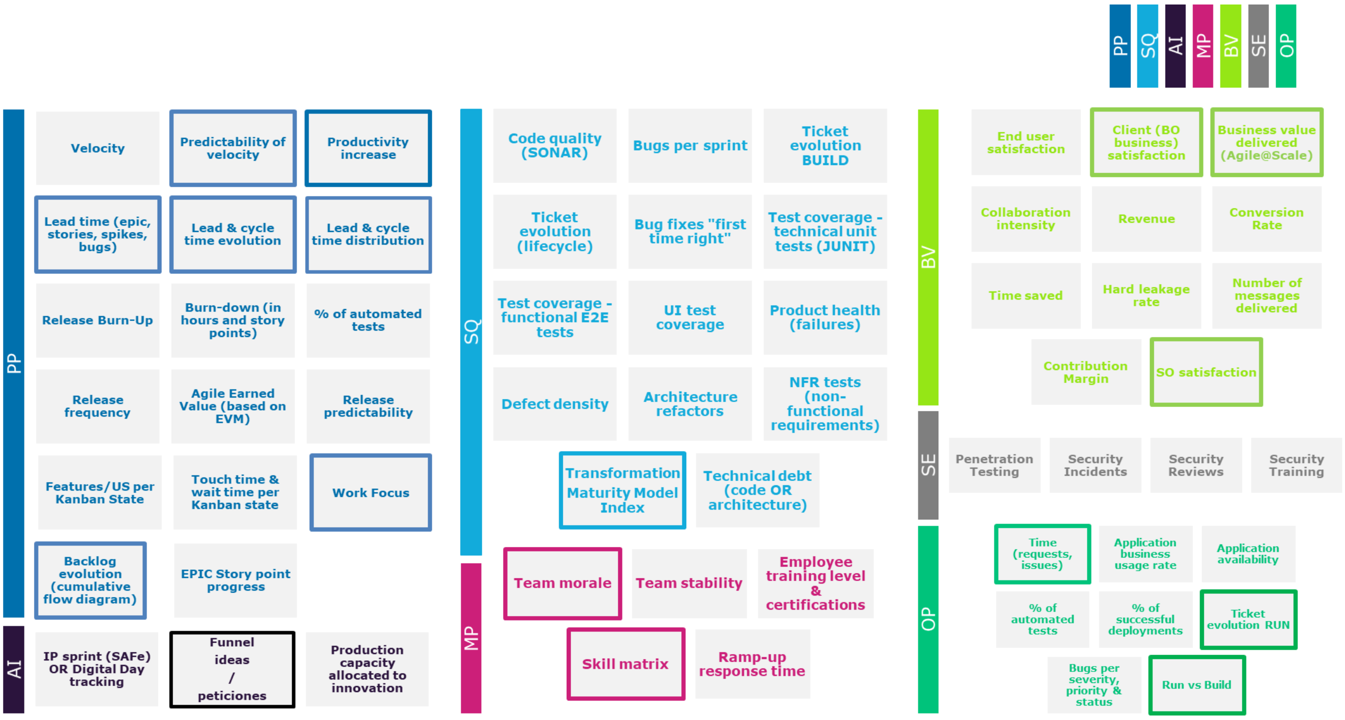
Team stability

…

Operations

Run vs Build

Ticket evolution Run



PP SQ AI MP BV SE OP

PP: Velocity; Predictability of velocity; Productivity increase; Lead time (epic, stories, spikes, bugs); Lead & cycle time evolution; Lead & cycle time distribution; Release Burn-Up; Burn-down (in hours and story points); % of automated tests; Release frequency; Agile Earned Value (based on EVM); Release predictability; Features/US per Kanban State; Touch time & wait time per Kanban state; Work Focus; Backlog evolution (cumulative flow diagram); EPIC Story point progress

AI: IP sprint (SAFe) OR Digital Day tracking; Funnel ideas / peticiones; Production capacity allocated to innovation

SQ: Code quality (SONAR); Bugs per sprint; Ticket evolution BUILD; Ticket evolution (lifecycle); Bug fixes "first time right"; Test coverage - technical unit tests (JUNIT); Test coverage - functional E2E tests; UI test coverage; Product health (failures); Defect density; Architecture refactors; NFR tests (non-functional requirements); Transformation Maturity Model Index; Technical debt (code OR architecture)

MP: Team morale; Team stability; Employee training level & certifications; Skill matrix; Ramp-up response time

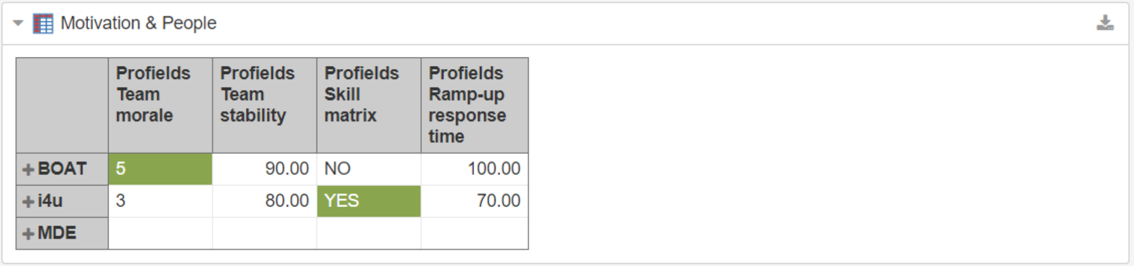
BV: End user satisfaction; Client (BO business) satisfaction; Business value delivered (Agile@Scale); Collaboration intensity; Revenue; Conversion Rate; Time saved; Hard leakage rate; Number of messages delivered; Contribution Margin; SO satisfaction

SE: Penetration Testing; Security Incidents; Security Reviews; Security Training

OP: Time (requests, issues); Application business usage rate; Application availability; % of automated tests; % of successful deployments; Ticket evolution RUN; Bugs per severity, priority & status; Run vs Build

KPIs – vision general

Team stability/Ramp-up response time



Motivation & People

	Profields Team morale	Profields Team stability	Profields Skill matrix	Profields Ramp-up response time
+ BOAT	5	90.00	NO	100.00
+ i4u	3	80.00	YES	70.00
+ MDE				

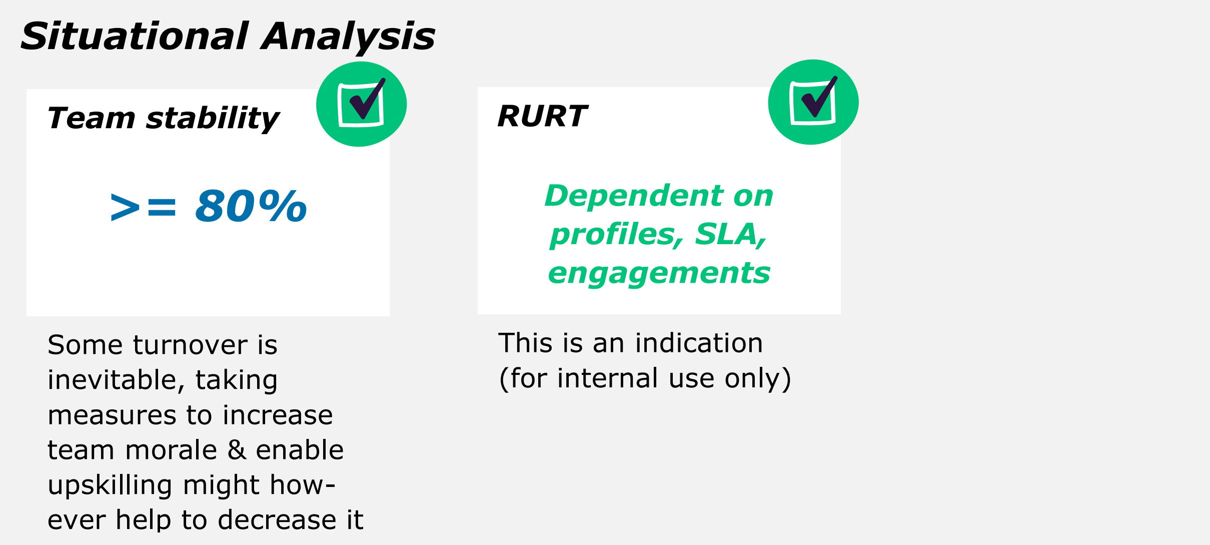
Objectives:

Measure People KPI to improve steering from HR perspective

Calculation:

Team stability: 1 – team turnover/ team size

Ramp-up response time (RURT): average days to fill new positions



Situational Analysis

Team stability ✓

>= 80%

Some turnover is inevitable, taking measures to increase team morale & enable upskilling might however help to decrease it

RURT ✓

Dependent on profiles, SLA, engagements

This is an indication (for internal use only)

Chapter 4

Introduction to Scrum framework

Scrum is an Agile Framework for developing complex products characterized by adopting an incremental and iterative development strategy.

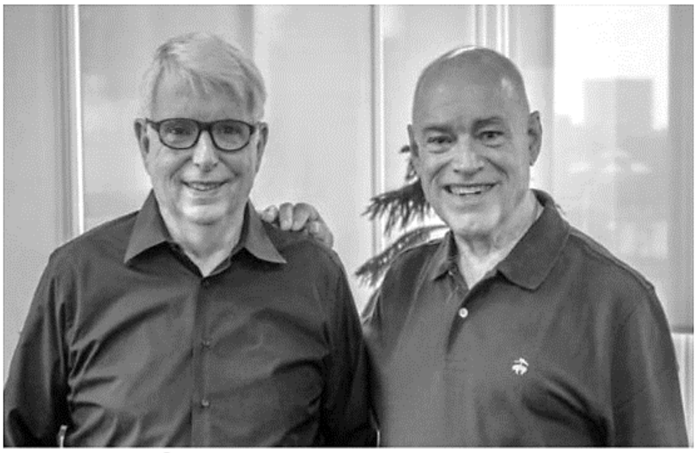
In the **1980s**, during the research study of Dr. Hirotaka Takeuchi (professor of Management Practice in the Strategy Unit at Harvard University) and Ikujiro Nonaka (professor in the International Corporate Strategy of the Hitotsubashi University), the new forms of teamwork were analyzed, with Scrum's advancement in training of Rugby players, resulting in the term "Scrum" being coined to refer to that research.

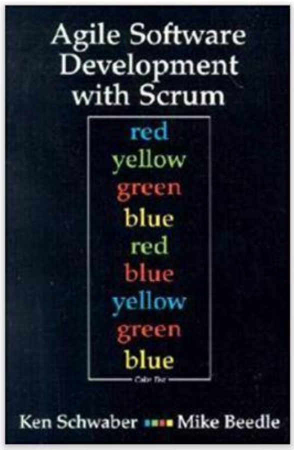
Although this form of work arose in technology products companies, it is suitable for the development of any type of Product and service.

In **1995** Dr. Jeff Sutherland and Ken Schwaber popularized Scrum after making public in an official publication "The SCRUM Development Process" at the UNRWALA conference in Texas. (Object, System, Language, and Application Oriented Programming) . The Scrum process is improved by Mike Beedle and Scrum is combined with XP

In **2001** 17 Agility Experts came together to discuss and sign the Agile Manifesto with the four values and twelve principles.

In **2001** the first Book on Scrum: "Agile Software Development with Scrum" was published.





Agile Software Development with Scrum

red
yellow
green
blue
red
blue
yellow
green
blue

Ken Schwaber ▪▪▪▪ Mike Beedle

Ken Schwaber

Jeff Sutherland Ken Schwaber

Scrum basics principles

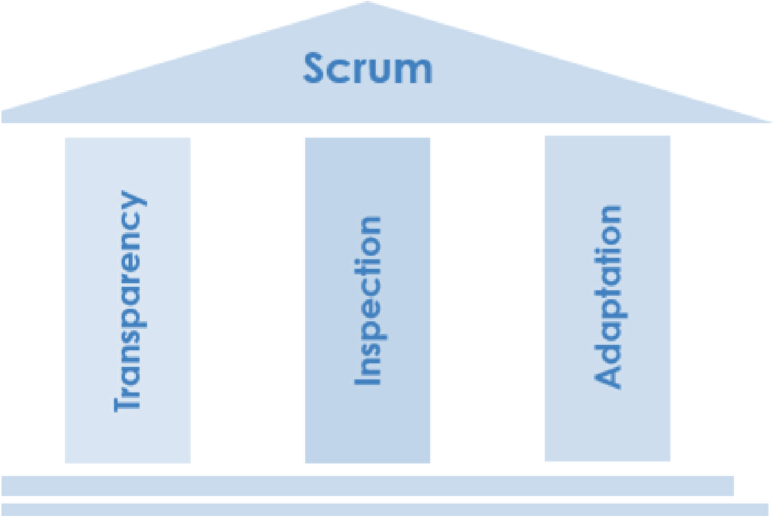
It is a framework Iterative and Incremental through which people can address complex adaptive problems, while delivering products efficiently and creatively with maximum value.

Scrum is:

- Lightweight
- Simple to understand
- Hard to master

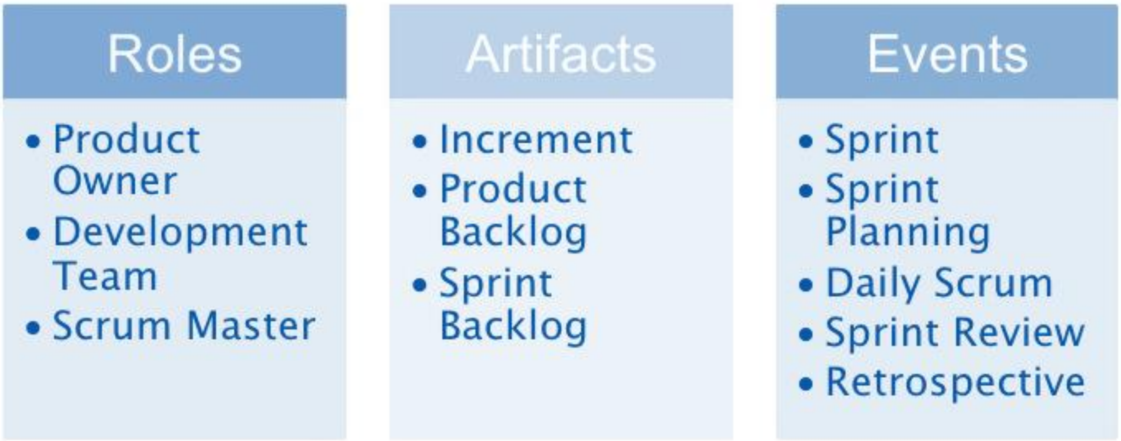
It consists of three basic pillars:

- Transparency
- Inspection
- Adaptation



It is a framework based on:

- Roles
- Artifacts
- Events



Roles	Artifacts	Events
• Product Owner • Development Team • Scrum Master	• Increment • Product Backlog • Sprint Backlog	• Sprint • Sprint Planning • Daily Scrum • Sprint Review • Retrospective

Scrum implements an **empirical process**.

Empiricism means working in a fact-based, experienced and tangible way.

Scrum Framework

Scrum combines four formal events for inspection and adaptation within a containing event, the Sprint. These events work because they implement the empirical Scrum pillars of transparency, inspection, and adaptation.

Scrum requires a Scrum Master to foster an environment where, a Product Owner orders the work for a complex problem into a Product Backlog.

The Scrum Team turns a selection of the work into an Increment of value during a Sprint producing a MVP (Minimum Viable Product)

The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.

Life cycle of Scrum from (Scrum Framework)

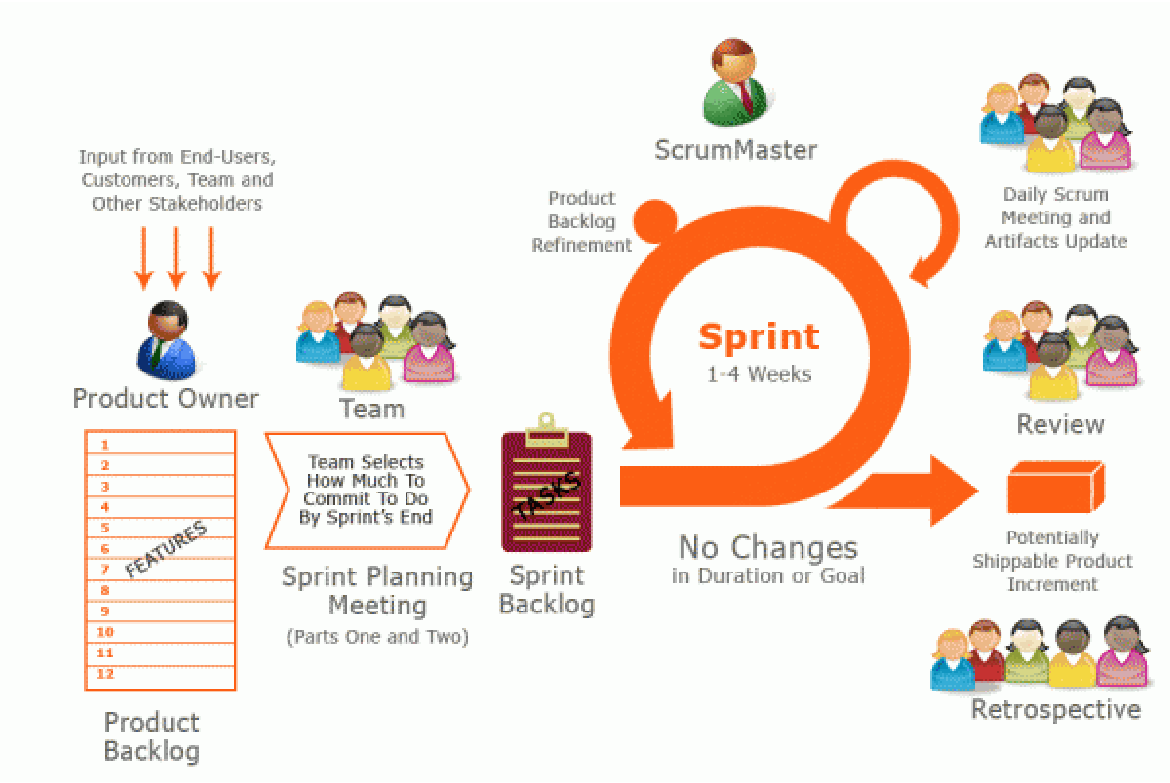

Input from End-Users, Customers, Team and Other Stakeholders
Product Owner
1
2
3
4
5
6
7
8
9
10
11
12
FEATURES
Product Backlog
Team
Team Selects How Much To Commit To Do By Sprint's End
Sprint Planning Meeting
(Parts One and Two)
TASKS
Sprint Backlog
ScrumMaster
Product Backlog Refinement
Sprint
1-4 Weeks
No Changes
in Duration or Goal
Daily Scrum Meeting and Artifacts Update
Review
Potentially Shippable Product Increment
Retrospective

Figure 1 source: http://www.agiletroop.com/product/life-cycle-of-scrum

Values of Scrum

Successful use of Scrum depends on people being more competent to live five values: ***Commitment, Focus, Openness, Respect and Courage***

Scrum's team is committed to achieving their goals and supporting each other. Its main focus is the Sprint's work to make the best possible progress towards these goals. The Scrum team and its stakeholders are open on work and challenges. Scrum team members respect each other to be capable and independent people and are respected as such by the people they work with. Scrum team members have the courage to do the right thing, to work on difficult problems.

These values give the Scrum team direction with respect to their work, actions, and behavior.

Scrum team members learn and explore values as they work with Scrum events and artifacts. When these values are assimilated by the Scrum team and the people they work with, Scrum's empirical pillars of transparency, inspection and adaptation make sense and build trust.

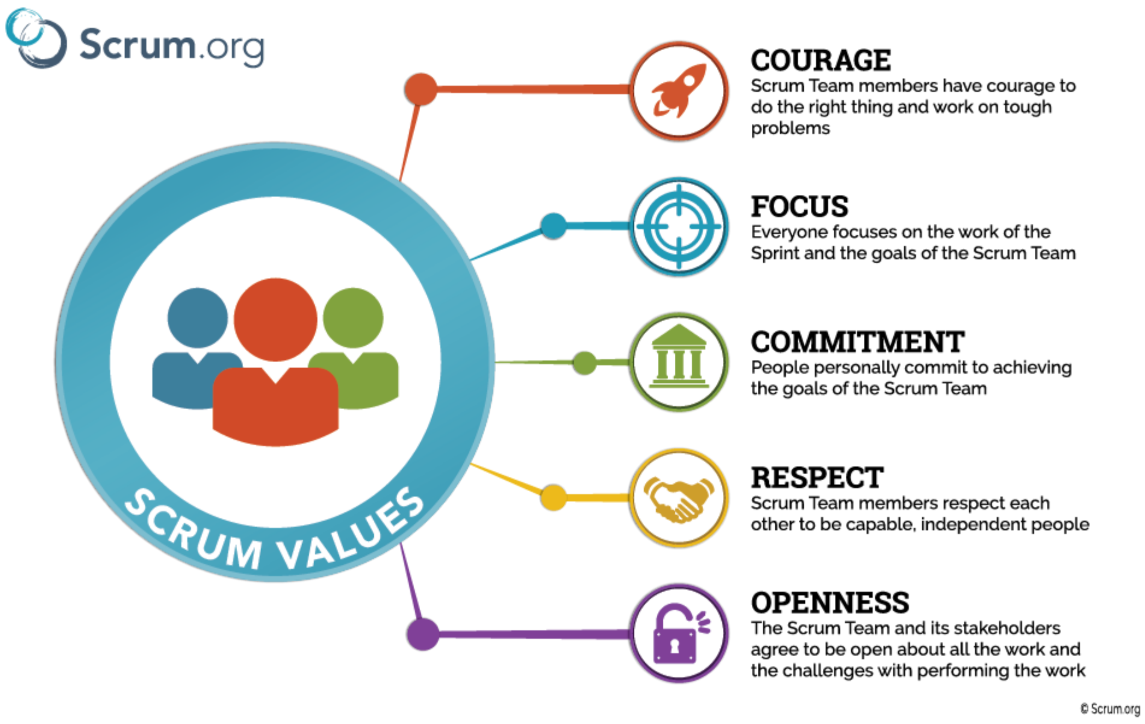

Scrum.org
SCRUM VALUES
COURAGE
Scrum Team members have courage to do the right thing and work on tough problems
FOCUS
Everyone focuses on the work of the Sprint and the goals of the Scrum Team
COMMITMENT
People personally commit to achieving the goals of the Scrum Team
RESPECT
Scrum Team members respect each other to be capable, independent people
OPENNESS
The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work
© Scrum.org

Figure 2, source: www.scrum.org/resources/blog/5-scrum-values-take-center-stage

Transparency, inspection and adaptation through empiricism

The Sprint Planning event members of the entire Scrum team agree on what to do in the next Sprint. User Stories are estimated at story points, to know how complex each one is and which of those that make sense of the Sprint Goal can be made in this Sprint.

The most important thing about this estimation process is that the team puts into common what it understands that needs to be done to properly develop the User Story, which allows to improve its understanding, update its content and, above all, adapt the acceptance criteria.

This improvement in acceptance criteria has proven to be key on many of the teams in order to see the actual scope of user story development.

Even the empirical model of inspection and constant improvement can be applied to the estimation mechanism itself, always focusing on simplifying the process to completely eliminating it and dedicating this time to what should be the goal of every team and analyzing how more value (items or improvements) could be delivered to the product at the end of the Sprint.

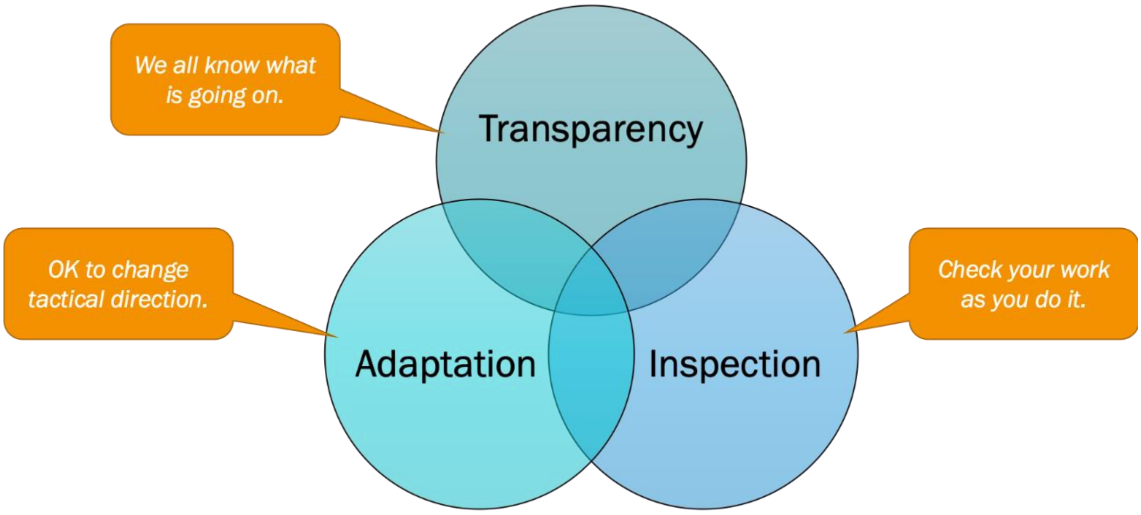

We all know what is going on.
Transparency
OK to change tactical direction.
Adaptation
Inspection
Check your work as you do it.

Image 5, Source

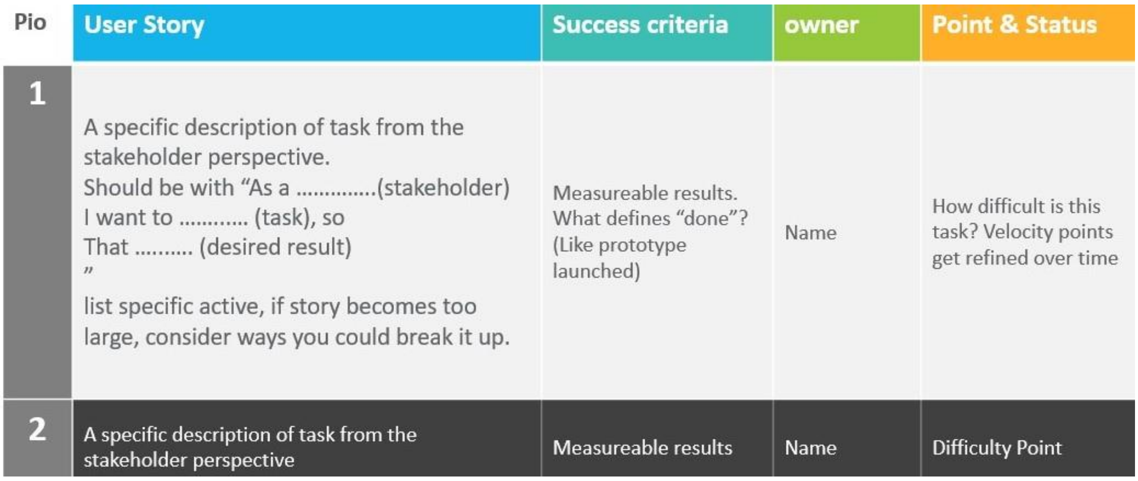
https://scrumorg-website-prod.s3.amazonaws.com/drupal/inline-images/2020-01/3pillarsofempiricism.png

User stories

A user story is a general and informal explanation of a software feature written from the end user's perspective. Its purpose is to articulate the "how" will provide a software function, value to the customer.

User stories put real end users in the center. They use non-technical language to provide context to the development team and their efforts.

They also help provide a user-centric framework for daily work, driving collaboration, creativity, and continuous improvement of the delivered product.



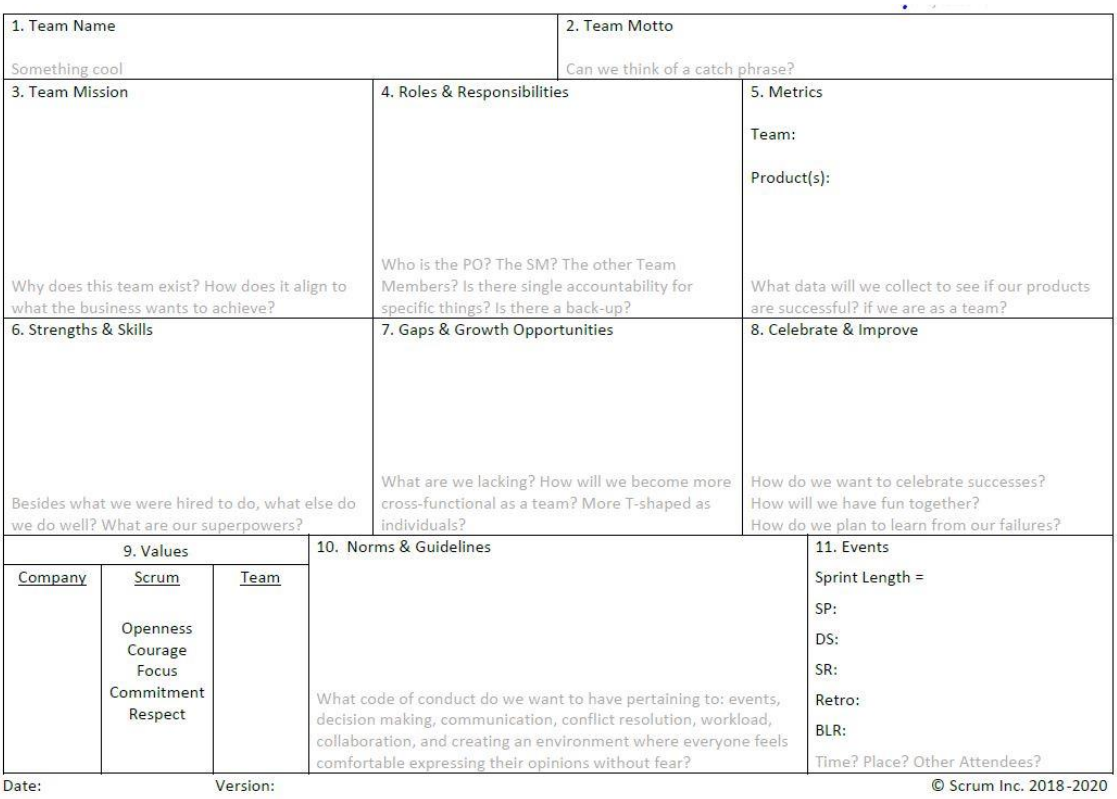
Pio	User Story	Success criteria	owner	Point & Status
1	A specific description of task from the stakeholder perspective. Should be with "As a ....(stakeholder) I want to .... (task), so That .... (desired result) " list specific active, if story becomes too large, consider ways you could break it up.	Measureable results. What defines "done"? (Like prototype launched)	Name	How difficult is this task? Velocity points get refined over time
2	A specific description of task from the stakeholder perspective	Measureable results	Name	Difficulty Point

User Story template example

Working Agreement and Lean Canvas model

The Lean Canvas is a business model visualization tool that combines elements of the Business Model Canvas and the Lean Startup method.

The Business Model Canvas was devised by Alex Osterwalder as a visual tool to analyze already established companies and propose new scenarios.



1. Team Name

Something cool

2. Team Motto

Can we think of a catch phrase?

3. Team Mission

Why does this team exist? How does it align to what the business wants to achieve?

4. Roles & Responsibilities

Who is the PO? The SM? The other Team Members? Is there single accountability for specific things? Is there a back-up?

5. Metrics

Team:

Product(s):

What data will we collect to see if our products are successful? if we are as a team?

6. Strengths & Skills

Besides what we were hired to do, what else do we do well? What are our superpowers?

7. Gaps & Growth Opportunities

What are we lacking? How will we become more cross-functional as a team? More T-shaped as individuals?

8. Celebrate & Improve

How do we want to celebrate successes?
How will we have fun together?
How do we plan to learn from our failures?

9. Values

Company	Scrum	Team
	Openness Courage Focus Commitment Respect	

10. Norms & Guidelines

What code of conduct do we want to have pertaining to: events, decision making, communication, conflict resolution, workload, collaboration, and creating an environment where everyone feels comfortable expressing their opinions without fear?

11. Events

Sprint Length =

SP:

DS:

SR:

Retro:

BLR:

Time? Place? Other Attendees?

Date: Version: © Scrum Inc. 2018-2020

Source of template:

https://34slpa7u66f159hfp1fhl9aur1-wpengine.netdna-ssl.com/wp-content/uploads/2018/10/Working-Agreement-Canvas-v1.24.pdf

Using the Lean Canvas model to create a template with the Teams work agreement.

The following are the working agreements of the teams to be reflected in the Lean Canvas Template:

* Team mission. The following are the questions that members and stakeholders must be made.

Why does this equipment exist?

How do you align with the goals the business wants to achieve?

* Roles and responsibilities

Who is the Product Owner?

Who's the Scrum Master?

Is there a unique responsibility for each specific activity?

In case of continued unavailability of some of the roles, what is the action plan to ensure the pace and delivery of sprints?

* Metrics

Equipment-specific metrics: Delivery speed, process quality, etc.

What data will we collect to see if we are successful as a team?

Product Metrics: Quality, Value contributed to the business, etc.

What data will we collect to see if our products are successful?

* Team strengths and skills

What else do we know how to do or can we develop, in addition to the functions for which we hired ?

Are we doing well as a team?

Are there hidden skills among the members we haven't detected?

* Gaps and growth opportunities

What do we need to become a high-performance team?

How can we be more multifunctional?

* Celebrate successes and continuous improvement

How do we want to celebrate success?

How are we going to have fun working together?

How do we plan to learn from our failures and register them to have an assessment of "Lessons Learned" ?

* Standards and guidelines

What code of conduct do we want to have in relation to:

Events

Decision-making

Communication

Conflict resolution

Balancing and workload

Collaboration

How can we contribute to creating a work environment where everyone feels comfortable and can express their opinions?

The importance of Scrum meetings

If **Sprint Planning** is not performed, the Scrum Team will need another method to formalize all the work where the work plan is viewed and prioritized and how it should run.

The **Daily Scrum**. If the team decides not to do the Daily meeting, another method will be needed to ensure the fluidity of communication, the aligned vision to meet the objective of the sprint and quantify the daily speed of the equipment because it is time to make visible and explain to the whole team the impediments and / or blockages. In Scrum's little scaling

(Scrum of Scrum) the Daily Scrum is analogous to the team level except that it is a virtual team composed of representatives from several individual Scrum teams who collaborate to integrate and ship one or more products. The Scrum Masters and any other necessary person communicate the necessary impediments, progress and coordination that must be made to integrate the product, following the similar format of the daily.

The **Sprint Review**. In the peculiar case that the Sprint review is not done, it is very likely that you will miss the enthusiasm of our client and the sponsor of the stakeholders. Product review is a very important ceremony to show and establish the dynamics of what to do and how to follow next, once customers and stakeholders have validated and accepted the potentially deliverable product increase.

If sprint Retrospective is not done, teams will need some way to create a shared understanding of how work is being done and what impediments and/or crashes slow down team performance. In addition, kaizen (continuous improvement) must also be produced to accelerate value delivery, and increase

quality of the process. Retrospective allows to promote in team’s integrity, honesty and self-assessment, thus allowing to create a state of well-being among people.

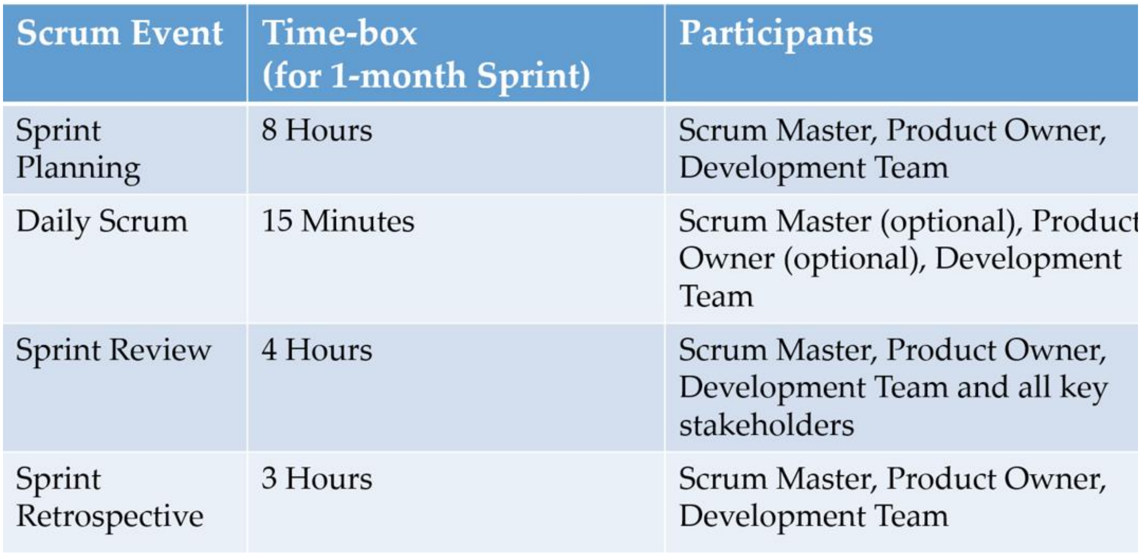
The **Refinement of Backlog** items could be said not to be a Scrum event, however, the Scrum Guide requires that the top of the Product Backlog be ready to start the Sprint.

So, one of the goals is to add details, estimates, and orders to current Product Backlog items. This is an ongoing process in which the Product Owner and development team collaborate on the details of current Product Backlog items.

During it, items (usually user histories) are reviewed and modified.

Team Scrum decides how and when refinement is performed. Refinement should generally not consume more than 10% of the Capacity of the Development Team. However, Product Backlog items can be updated at any time by the Product Owner.

Scrum events: timeboxing and participants:



Scrum Event	Time-box (for 1-month Sprint)	Participants
Sprint Planning	8 Hours	Scrum Master, Product Owner, Development Team
Daily Scrum	15 Minutes	Scrum Master (optional), Product Owner (optional), Development Team
Sprint Review	4 Hours	Scrum Master, Product Owner, Development Team and all key stakeholders
Sprint Retrospective	3 Hours	Scrum Master, Product Owner, Development Team

Image 6, source: www.scrum.org

Sprint Planning and team capacity

Sprint Planning meeting based on the team's ability should involve at least the entire Scrum Team. The Product Owner makes the required items visible and sorts them in the Product Backlog starting with the highest priority items.

Then, the people of the Development Team select the first item(s) to take to the sprint. It will usually be the highest priority item guided by the Product Owner.

Development team members discuss the work to be done and identify and break down the necessary tasks and dependencies (if any).

The estimates resulting from this first analysis are most likely approximate (within a known tolerance range) and you should be certain that at least you will enter a Sprint, otherwise you will need to decompose more granularly.

Once the initial agreement is reached, the team (with the vote and approval of all members) should wonder if with its current capacity it can commit to performing those tasks during the Sprint and delivering the increase.

Group engagement is stronger than individual engagement, as if someone has an impairment or is delayed for some reason, anyone on the team will go to their aid to untie the problem and continue to move forward. For this reason, absolute transparency in communicating the impediment (at the same time as arises or in any case in the Daily meeting) is of vital importance.

When the Dev. Team has completed the Backlog with the items selected for the Sprint, the Scrum Master could add up the assigned story points and share and analyze the result with the team. This will allow the team to have a very valuable metric to understand their ability with respect to other Sprints and visualize the underlying risky and potential risk of technical debt.

It may happen that during the Sprint the Dev. Team identify new emerging tasks that haven't been revealed during sprint planning, or they may also discover that User Story is harder than they initially thought.

In these cases, you must immediately inform the Product Owner and re-estimate to see if it is still possible to comply with the Sprint Goal and be able to deliver the agreed increment at the end of the Sprint.

It is very important for the team to define the Sprint Goal.

The Sprint Goal is a set of goals for the Sprint that can be accomplished by implementing the Product Backlog. They are the result of the negotiation between the Product Owner and the Development Team.

Sprint planning cycle:

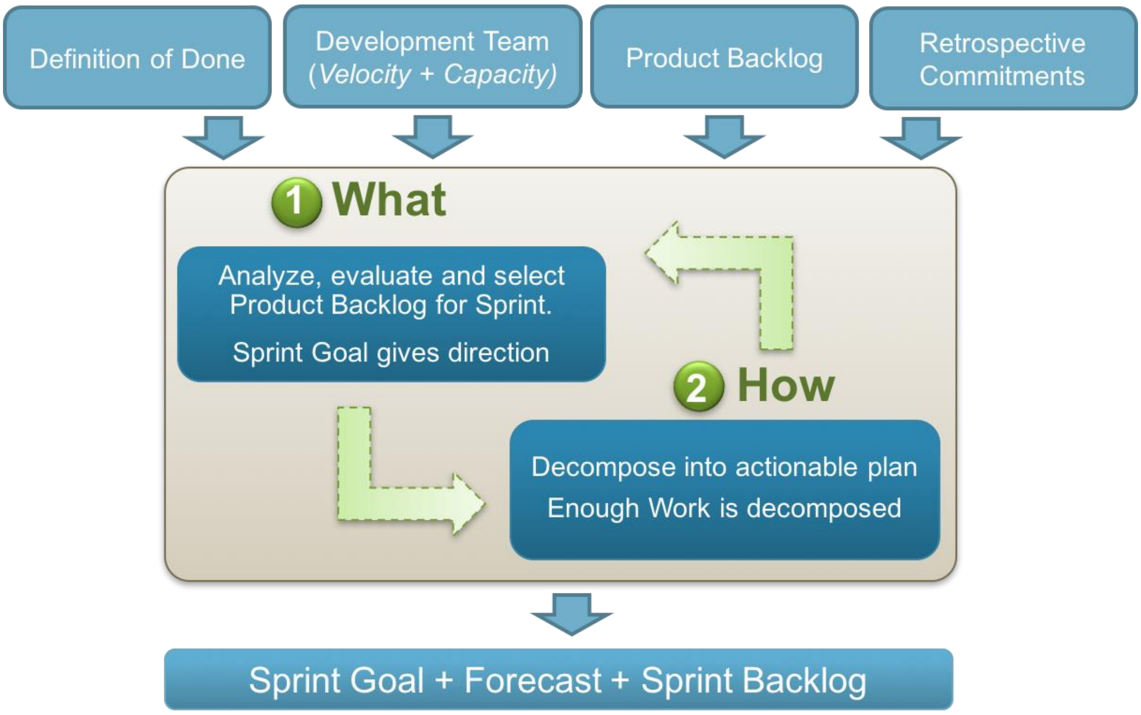

Definition of Done
Development Team (*Velocity + Capacity*)
Product Backlog
Retrospective Commitments
1 What
Analyze, evaluate and select Product Backlog for Sprint.
Sprint Goal gives direction
2 How
Decompose into actionable plan
Enough Work is decomposed
Sprint Goal + Forecast + Sprint Backlog

Image 7, source: https://www.scrum.org/resources/what-is-sprint-planning

Sprint objectives must be specific and measurable. While the work selected for the Sprint Backlog can be visualized as a forecast/estimate (which has been initially calculated), the Development Team to what it really commits to is to achieve the Sprint Goal.

What to do with work that could not be completed during the Sprint

On several occasions Agile Scrum equipment may not end all the work planned for the Sprint. Teams, without consulting, may directly move that work to the next sprint.

This is an error, and nothing recommended, as all unfinished work must return to the Backlog and the Product Owner should be queried to do. In this case she or he will determine the importance and re-prioritize the incomplete work, to decide in which Sprint is incorporated and/ or it may also be the case that the unfinished user story, no longer have the business relevance to incorporate them into the next release.

If we conjecture that the work is unfinished, it has been decided that it needs to be done in the following sprint, then we have the following question:

Should the User Story be rewritten completely (or part of it) with the tasks broken down and re-estimated?

It is very important that in the retrospective of the end of Sprint the team analyzes what have been the reasons and impediments that have caused the estimated work could not be completed, this will help the team to better understand its own capabilities and external dependencies in order to make Agility work successfully in teams and across the organization.

Agile Scrum Timeboxing

It is very important to learn how to respect the timebox that Agile Scrum sets in your meetings. In the table you can see the maximum duration (Timebox) of each ceremony depending on the size of the Sprint.

These timeboxes should not be interpreted as "recommended durations", everyone has to be clear that they are "maximums of time".

For this reason, people should be prepared for meetings with what they want to express/expose, this will allow participants to understand the appropriate level of detail to maximize the value that each ceremony brings.

If there is no prior preparation, we can find the situation that a meeting can abruptly end when time has been met, for these issues and others, planning each ceremony is very important.

Daily Scrum

Sometimes we may encounter some excuses that people often make to avoid the Daily, such as that it "reduces the productivity of the team" (in my opinion, having 15 minutes a day should not minimize anyone's productivity).

The benefits of the daily, are very important for the teams, since people will feel energized to hear what their peers do, also minimizes the possibility of making mistakes and delays in decisions. In addition, it helps foster a positive push to inform what each team member will make and show the daily progress.

In the event that you are working with distributed teams and/or fully remote teams, the rules may change, for example the Time of the Daily may be longer as participants may need a previous conversation (5 to 10 minutes) to enter the situation and then start with the Daily Scrum.

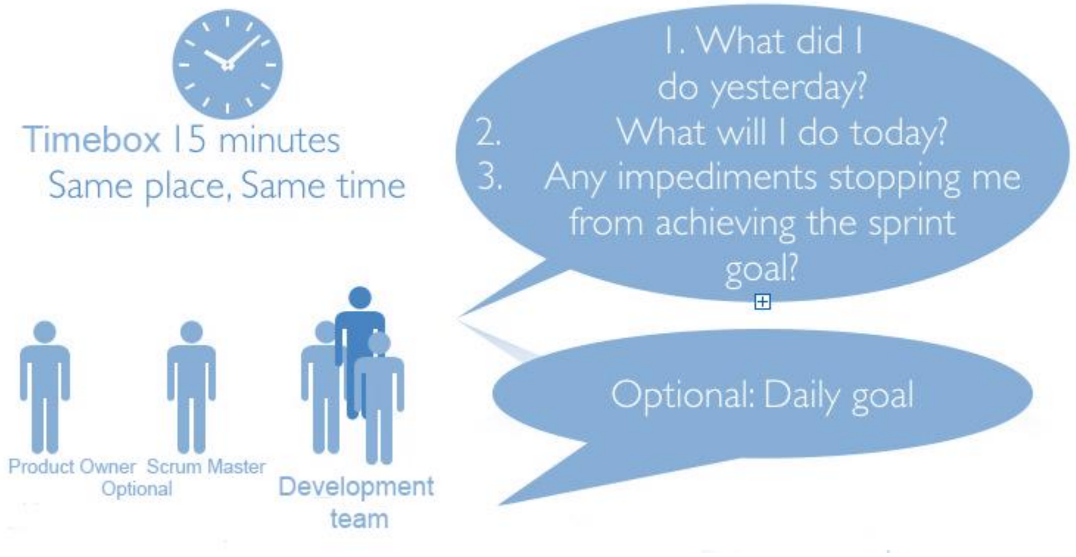


Image 8, Daily Scrum meeting

It should be noted that at first the teams will decide for themselves which timeboxes are appropriate for each meeting, but then with the maturity and excellence that high-performance teams achieve over time, those parameters will fall within the established limits proposed by scrum's guide.

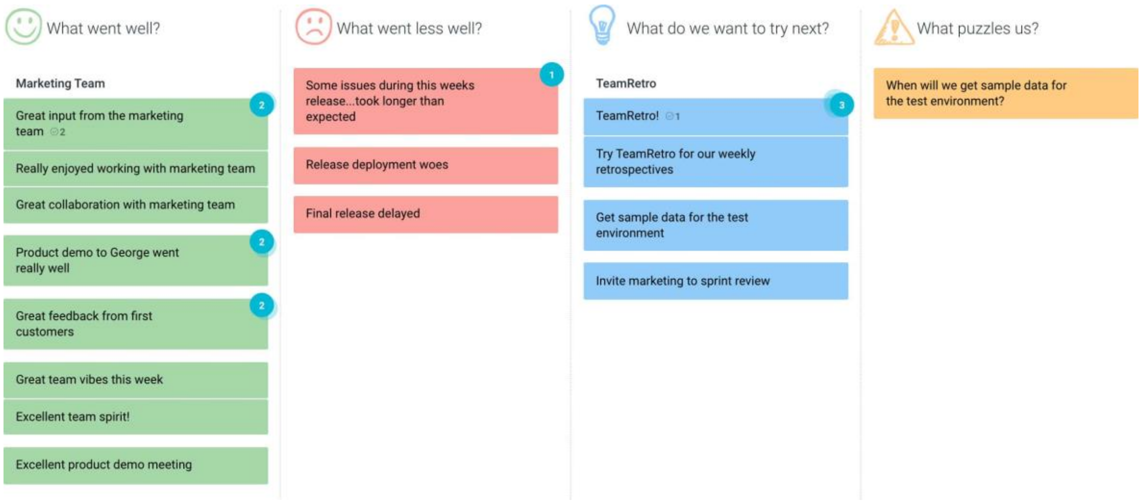
The Sprint Retrospective

The Sprint Retrospective is an essential event of Agile Scrum for the

continuous improvement of equipment, especially in the current context, where the need for working with distributed teams and fully remote teams is a global trend.

It is often observed that in practice, it is detracting from the value and/or importance that it contributes and requires. This is due to several reasons, but especially when teams and organizations feel they can no longer improve and go beyond what they are currently.

Even if the teams are working optimally, so are the people who make up it must understand the need to find ways to follow improving, to anticipate the uncertain future, to have greater adaptation and flexibility in the face of constant changes in the environment.



What went well?

Marketing Team

- Great input from the marketing team 2 — 2
- Really enjoyed working with marketing team
- Great collaboration with marketing team
- Product demo to George went really well — 2
- Great feedback from first customers — 2
- Great team vibes this week
- Excellent team spirit!
- Excellent product demo meeting

What went less well?

- Some issues during this weeks release...took longer than expected — 1
- Release deployment woes
- Final release delayed

What do we want to try next?

TeamRetro

- TeamRetro! 1 — 3
- Try TeamRetro for our weekly retrospectives
- Get sample data for the test environment
- Invite marketing to sprint review

What puzzles us?

- When will we get sample data for the test environment?

Example of team retrospective

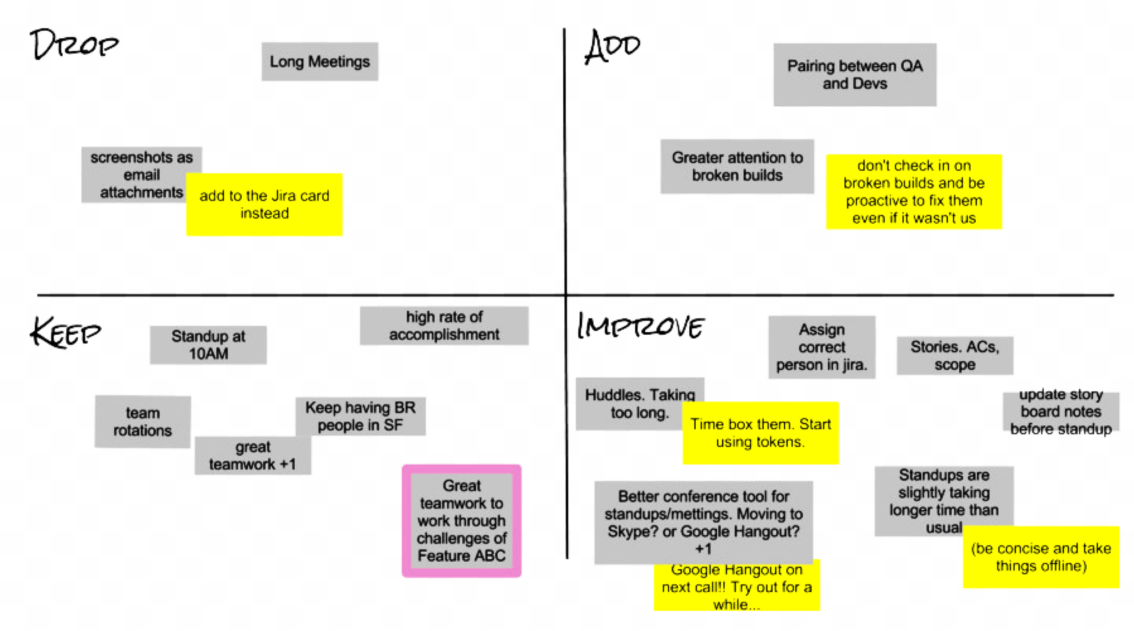
Image 9, source: www.teamretro.com/retrospectives/wrap-retrospective

There are certain gaps to consider improving retrospectives and ensure that they provide the optimum value for the purpose they have been conceived.

Honesty and transparency are an essential value for this event, all people should pose real team problems and/or admit their individual problems.

That's why it's important to create a proper environment, as people won't talk retrospective unless you are guaranteed a minimum of security and confidentiality, especially in the event of sensitive issues and/or difficulties among the interactions with other teams and stakeholders.

The Agile Coach or Scrum Master should be coaching people involved so that criticism is constructive, and whenever possible, that critical with suggestions that can be applied to obtain an improvement or solution to the situation.


DROP
Long Meetings
screenshots as email attachments
add to the Jira card instead
ADD
Pairing between QA and Devs
Greater attention to broken builds
don't check in on broken builds and be proactive to fix them even if it wasn't us
KEEP
Standup at 10AM
high rate of accomplishment
team rotations
Keep having BR people in SF
great teamwork +1
Great teamwork to work through challenges of Feature ABC
IMPROVE
Assign correct person in jira.
Stories. ACs, scope
Huddles. Taking too long.
Time box them. Start using tokens.
update story board notes before standup
Better conference tool for standups/mettings. Moving to Skype? or Google Hangout? +1
Google Hangout on next call!! Try out for a while...
Standups are slightly taking longer time than usual
(be concise and take things offline)

Example of team retrospective

Image 10, source: www.careerbuggy.com/agile-scrum-retrospective-techniques/sample-retrospective-daki-drop-add-keep-improve

Another factor to consider is knowing how to meet the agreed commitments that have been identified in previous retrospectives, as the ceremony can quickly lose the value if people perceive that changes are not met and no agreements are reached for the continuous improvement.

Collective Intelligence in Agile Scrum teams.

The concept of swarming is a way to increase the speed of

Agile teams immediately. It is a pattern that is used on a recurring basis

high-performance teams.

This concept of swarming in Agile Scrum occurs when as many people as possible teams, work simultaneously on the same Item, and what do until the item is complete.

Generally, each Sprint Backlog consists of items (User Stories) of different importance, all they must be completed at the end of the Sprint, but only one of these items should be the team's top priority. As a result, at the beginning, the team focuses on the User Most important story.

It is very important to attract all possible team members to work on that item

priority to work collaboratively, simultaneously and expand intelligence collective of the group.

When the highest priority item is completed, the team is crowded in the next User Story until completed in its entirety and thus until all the User Story of the Sprint is finished.

Another approach to addressing the Sprint is to divide items from different stories among the team members, making each team member work on

different tasks at the same time until the agreed objective to be delivered at the end of the Sprint.

The key to the effective swarming concept is whenever the people who

team members must change the focus from one thing to another, you lose a percentage considerable productivity. This concept, called "Context Change," was first presented by Gerald Marvin Weinberg (computer scientist & PhD in

Communication Sciences) in his book "Quality Software Management". Weinberg explains that if someone is working on one thing until it's finished (Done), obviously will have 100% of their focus and time spent on that work. But if you have to change context only once, the amount of time available to work drops to 80%. If you have you switch context 3 times in a day, your effective time is reduced to just 60%. Changing 5 times you will lose 75% of your time and the focus will only be on changing context on a recurring basis.

The Swarming concept reduces context change by making the whole team concentrate only one item at a time during the Sprint. Therefore, there is no mental restart and restart by context change.

Swarming also increases the productivity of Scrum team by improving one of the metrics, known as process efficiency.

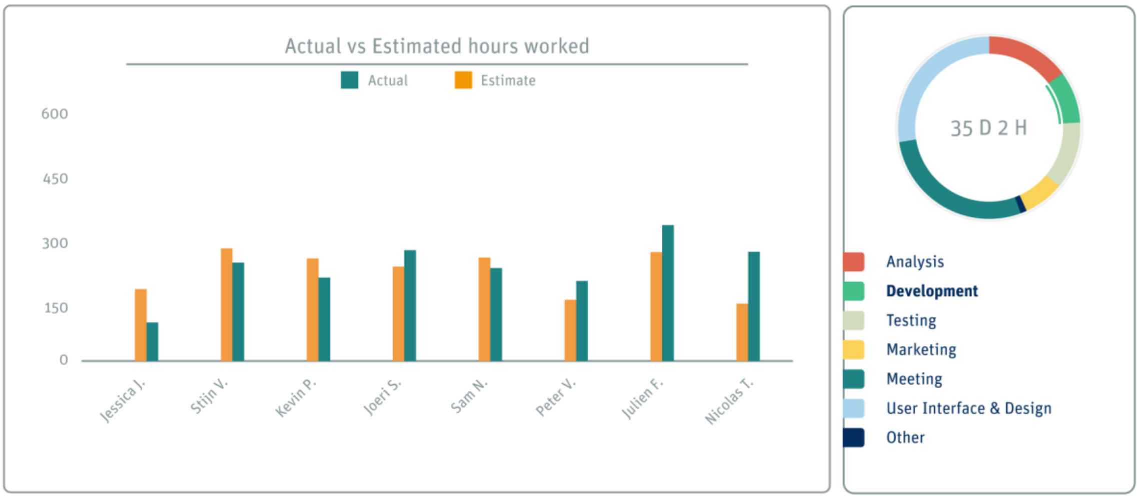
Chapter 5

Process efficiency and WIP limitation in high performance teams

The efficiency of the development process of an Agile Scrum Team is the most important to increase team performance, a team can double the Sprint speed by optimizing the efficiency of the development/construction process.

A very simple way to calculate process efficiency can be as follows:

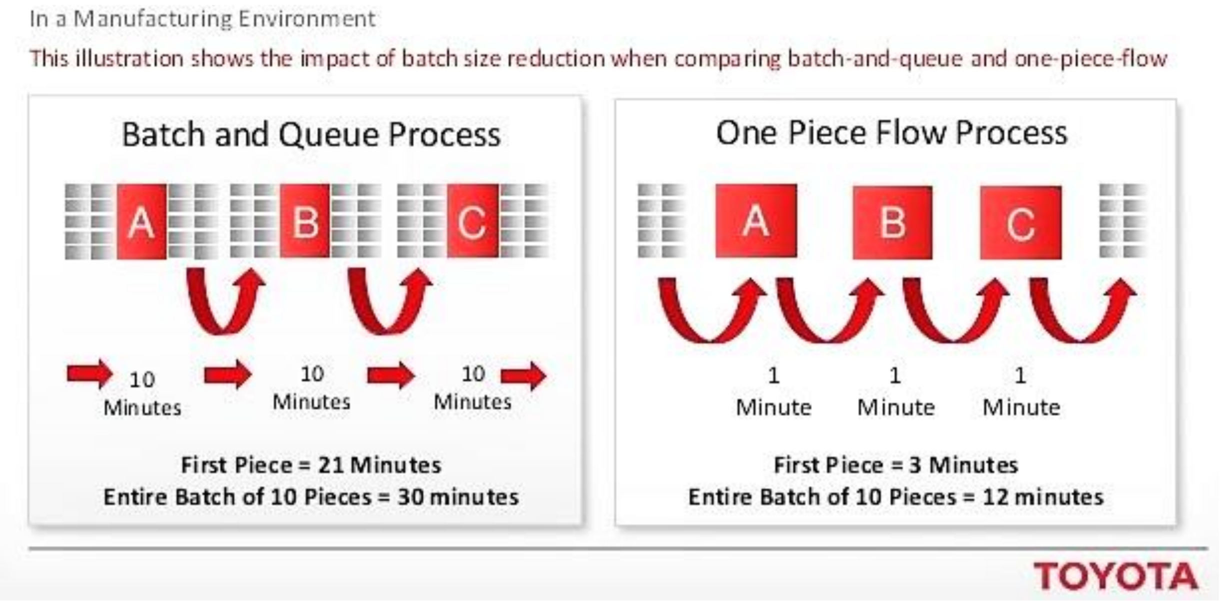
Estimated time / Actual time worked, resulting in value = 1 (ideal time)



Example of Estimated time vs Actual time worked. Source Image 11, https://timetag.it/tour

For a team to improve process efficiency, it must avoid "multitasking" the people who make it up. Instead run the Swarming pattern: One-Piece Continuous Flow (Toyota) will significantly increase the completion flow of user story Scrum's fundamentals promote the One-Piece Continuous Flow.

One-piece flow is all about reducing WIP to the point where everything is either waiting to be started, in progress, or complete. That's why it's also called "single piece flow" and "continuous flow", as everything is constantly progressing and only one item is in any given queue at a time.


In a Manufacturing Environment
This illustration shows the impact of batch size reduction when comparing batch-and-queue and one-piece-flow
Batch and Queue Process
A
B
C
10 Minutes
10 Minutes
10 Minutes
First Piece = 21 Minutes
Entire Batch of 10 Pieces = 30 minutes
One Piece Flow Process
A
B
C
1 Minute
1 Minute
1 Minute
First Piece = 3 Minutes
Entire Batch of 10 Pieces = 12 minutes
TOYOTA

Source Image 12, www.institut-lean-france.fr

On several occasions Scrum Team people end up doing more than one PBI (Product Backlog Item) at a time.

To avoid this each team member must take possession of a PBI. That's why it's very important to visually track the amount of GDP in progress on the dashboard.

This will allow you to limit the WIP (Work in Progress) and give teams more time to can understand the context and deliver value products with minimal or no value technical debt.

Reference: www.scrumatscale.com www.scruminc.com

How to handle errors detected in Agile Scrum projects

To address detected errors (bugs), there are several alternatives, one of which is to immediately communicate to developers during the ongoing Sprint.

Another possibility is to create a linked task in the same affected User Story, identifying it as a Bug tag.

It's unusual, but in some cases error or defect reports may be beneficial to write as user stories.

When an error or defect is detected, the Scrum Team should consider fixing it as soon as possible. If the error is High Priority or Blocker, it must be added to the sprint in progress. Otherwise, where the error does not affect business continuity, it can be put in the Backlog for the Product Owner and Development Team to decide when to fix them during successive Sprints, because fixing a bug of low priority and high complexity would take a long time and affect the Sprint Goal.

The Scrum Guide does not directly address how to manage errors, as it puts special focus on building and promoting from the culture of Quality. It uses one of the premises of Lean (**Built-In-Quality**), which incorporates quality to products from the beginning, as it is much more efficient and effective than repairing errors later.

Cost versus quality versus speed graphic:

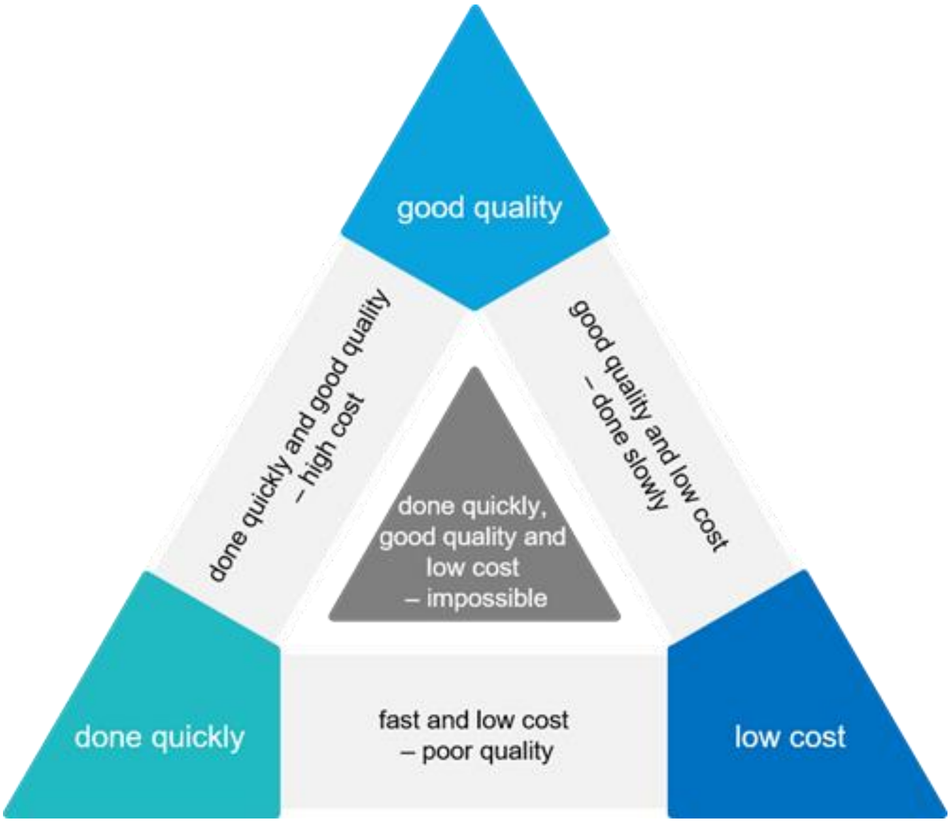

good quality
done quickly and good quality – high cost
good quality and low cost – done slowly
done quickly, good quality and low cost – impossible
done quickly
fast and low cost – poor quality
low cost

Image 13, source: https://www.innovation.co.uk/execution-risk/implementation-dilemma/

It should always be remembered that Agile Scrum evolves favorably when the Development Team self-manages and organizes its own work in the most efficient way.

It is usually the team itself, which with the guidance and help of the Scrum Master, usually finds the best solutions to the problems detected.

Agile Scrum Team - Code Refactoring

Software code refactoring could be described as modifying the source code without changing the behavior of the source code. (Clean code), Helps improve internal consistency and quality, also improves ease of understanding of code for easy future maintenance.

An Agile Scrum team should have time to refactor the product to prevent errors from occurring and/or prevent future problems.

The problem is that refactoring can be difficult to prioritize for the Product Owner, because it does not affect in the short/medium term. To do this it is necessary to evaluate the Cost of refactorize.

The development team should submit a report (understandable to the

Product Owner) to demonstrate the justification for the effort/benefit that would entail refactoring, which could be defined as follows:

* Estimate the short-term impact it would have on users and the company considering the current situation. That is, estimate the impact of leaving the code as it currently is. It is important to do this by gathering real data about the time spent investigating, correcting, and validating defects in the code you want to refactor. An easy-to-use technique is the Pareto Chart that allows you to graphically display the Pareto principle (80-20; Few vital, many Trivial).

Pareto Analysis Diagram

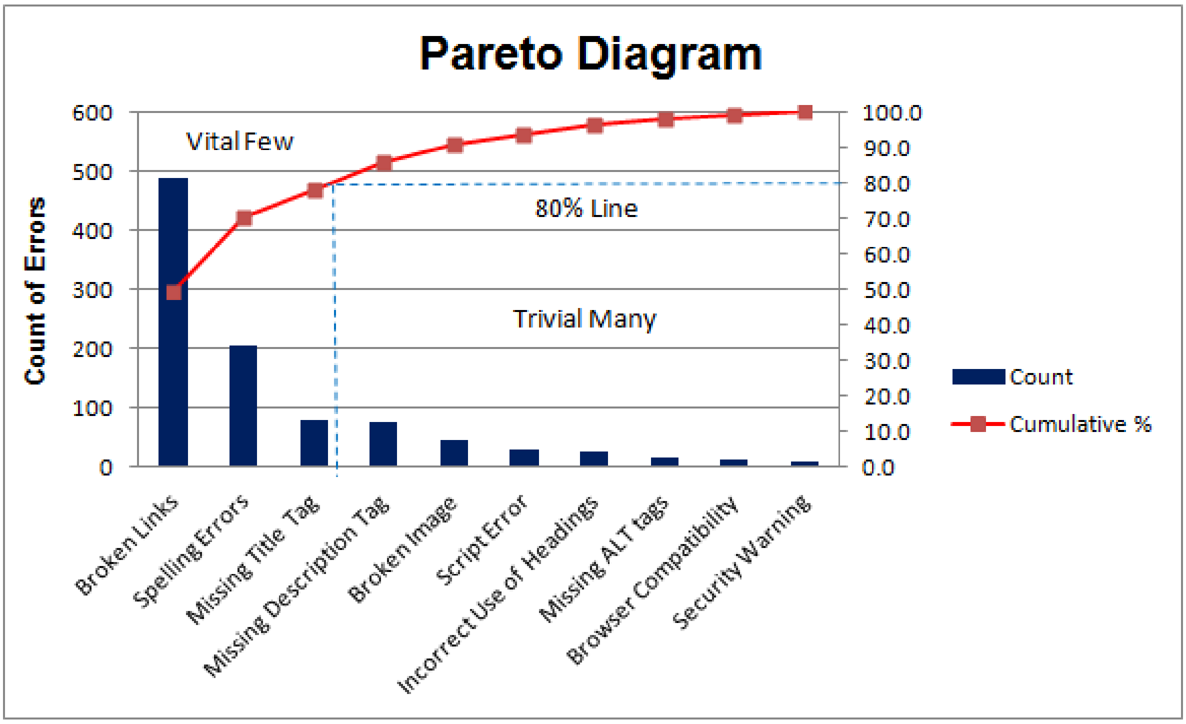


Figure 3, source: https://www.projectsmart.co.uk/pareto-analysis-step-by-step.php

* Estimate the effort, contemplating all possible variables, to perform the refactoring. The development team must estimate refactoring just as they would any other element of the Product Backlog. This means that the estimate could be in history points (if they use that unit of measure).

* Estimate time savings after refactoring. In many cases it would not be entirely realistic to assume that refactoring will eliminate all current defects and the need to review the old code in the future. For this reason the refactoring proposed by the team could reduce a little more than 50% of the current time spent fixing the defects.

* Estimate the return period of the investment. The recovery period is calculated by dividing the number of hours required to refactor by the number of hours saved per iteration:

Hours to refactor / Hours saved by iteration

Workflow of code refactoring:

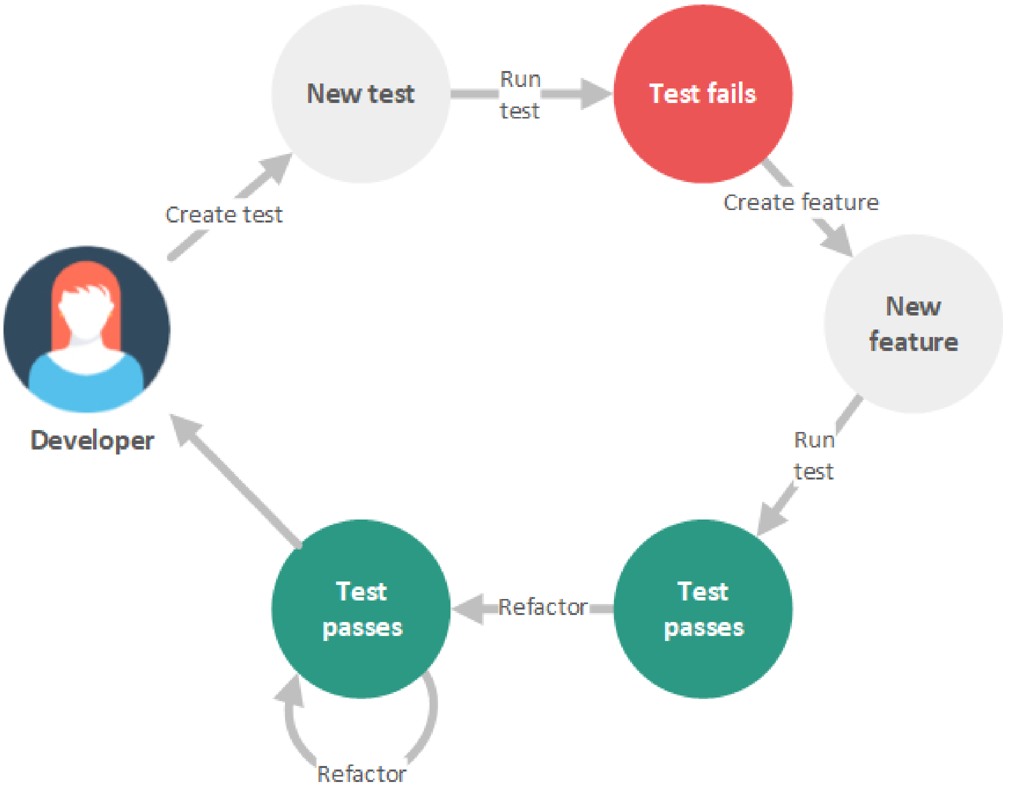


Image 14, source: https://devopedia.org/code-refactoring

The appropriate recovery period is influenced by many project factors, including the urgency of incorporating other new features and the estimated product lifecycle. For this reason, it is very important to incorporate all the variables identified with the maximum possible objective rigor. It is very difficult to justify with sustainable arguments the Product Owner and Stakeholders the refactoring of a product that will be withdrawn in the short term.

Chapter 6

Agile Coach and Scrum Master in the process of building Agility

During the process of building Agility of a company's equipment, we can find that some activities that are required to overcome this transition have complexity for the people who make up the different teams.

Part of the role of scrum Master and Agile coach is to exercise the job of facilitator, but this does not imply "passivity" but rather that it is a role that requires a lot of dynamism.

The high productivity of the equipment is a very important indicator, but so is the people's well-being, because that's necessary to be able to sustain that high-quality hyper-productivity and thus become high-quality equipment Performance.

It is important to focus on building and maintaining the commitment of people who make up the teams. Currently the demand for talented and large people capabilities are very high, which means that having to continually replace people teams have a high cost and impact on the organization, which can lead to large deviations in key projects.

In this context, it is essential that Agile Coach and Scrum Master can identify new opportunities to improve team performance and replicate that learning.

In addition, Scrum Masters should monitor and identify daily risk factors and continuously eliminate any impediments that computers have detected. This is what it's for essential that they know the business value of each of the projects involved and that able to effectively calculate and communicate the commercial impact of impediments and blockages not taken care of in a timely manner, managers and managers.

Fibonacci Sequence, Story Points and T-Shirt Sizes

Estimates are one of the biggest challenges of organizations and teams Agile methodology for product and/or service development.

Initially some teams choose to use the T-shirt size technique, (T-shirt Sizing, available in tools such as JIRA) to get started on the relative estimate.

This is a technique in which Product Backlog items are estimated in product backlog sizes shirts:

XS (extra Small), S (small), M (medium), L (large), XL (extra-large)

It is a highly advisable way to start with the relative estimate, especially in teams that are starting out in Agile methodologies, but can cause some disadvantages if over time you are not assigned numeric values.

For example, a Scrum Team cannot tell your client or stakeholder that the entire product will have a development effort of 7 sizes spread over 3 XL, 2 M and 2 Small.

This won't make any sense, especially if the customer doesn't have a reference unit relative effort. It should be remembered that T-shirt sizes are a measure

subjective effort.

Another drawback is that it does not allow to create an average of the speed of the equipment but defines a single scale that converts T-shirt sizes into a numerical model for the relative value of the speed of development can be measured.

For these reasons it is important to assign numerical values to the size of the T-shirts, a example could be:

XS (extra Small) = 1

S (small) = 2

M (medium) = 3

L (large) = 5

XL (extra-large) =8

A fragment of the Fibonacci Succession can be used, as in this case, to be able to assign numerical values to T-shirt sizes. This way teams can introduce the concept of Story Points, which over time and maturity teams provide more accurate and accurate information to analyze results and promote continuous improvement.

The concept of User History estimation is based on the US Air Defense Department in 1950 that developed the Delphi. This technique was government-classified information until the 1960s.

Differences between impediment and blocker

There are important differences between these two concepts that are repeatedly confused and applied incorrectly.

Impediment:

An impairment slows or slows down the speed of the team, but does not completely paralyze it.

The impediments generate (Waste / Muda) in the normal process of product construction, (See Lean Mute types).

Blocker:

When a blocker action occurs, the delivery of the product is stopped, i.e. if no action is taken immediately, it could happen that the team could not continue to advance development in order to deliver the product.

We mainly have 2 types of blockers: Blocked Tasks and General Blocker.

One of the most common causes of blocked tasks is an incomplete dependency that paralyzes the progress of the task being worked on, but this does not prevent the person from continuing with another task.

A general blockage is more complex, this can occur, (for example), when there is a problem in the work environment that prevents a person or more (development team member/s) from continuing to work.

Define rules and standards and leave space for innovation too.

Standardizing, creating rules and standards is an essential part of the Agile transformation process. Agile Scrum, Lean, SAFe, Scrum@Scale, SPS Nexus, Lean Six Sigma (DMAIC), LeSS, etc. have defined rules and/or standards so that the framework/methodology can be applied correctly.

Managers and people leading the Agile transformation must create daily work standards and set the primary rules teams must initially follow.

Promoting and promoting the self-organization of teams does not mean that they do NOT have to follow the established rules.

Creating standards of several general aspects of how teams should collaborate and work on large projects is extremely necessary because it allows people to move between teams quickly and more easily.

In contrast to this, there are several occasions that are the teams themselves that make the decision to standardize a process, approach and / or standard, since they themselves understand the need to put order and discipline to manage tasks and projects more efficiently.

There are situations where it is necessary to allow different teams the flexibility to do things different ways than they are already established and validated by the organization.

In some key and/or innovation projects, if teams are forced to perform all tasks/operations strictly standard, they will gradually diminish the initiative to experiment and as a result decrease the ability to explore, thus losing the benefit of continuous improvement. In addition, they will feel less belonging

throughout the product creation development process, which will impact the value delivered to customers.

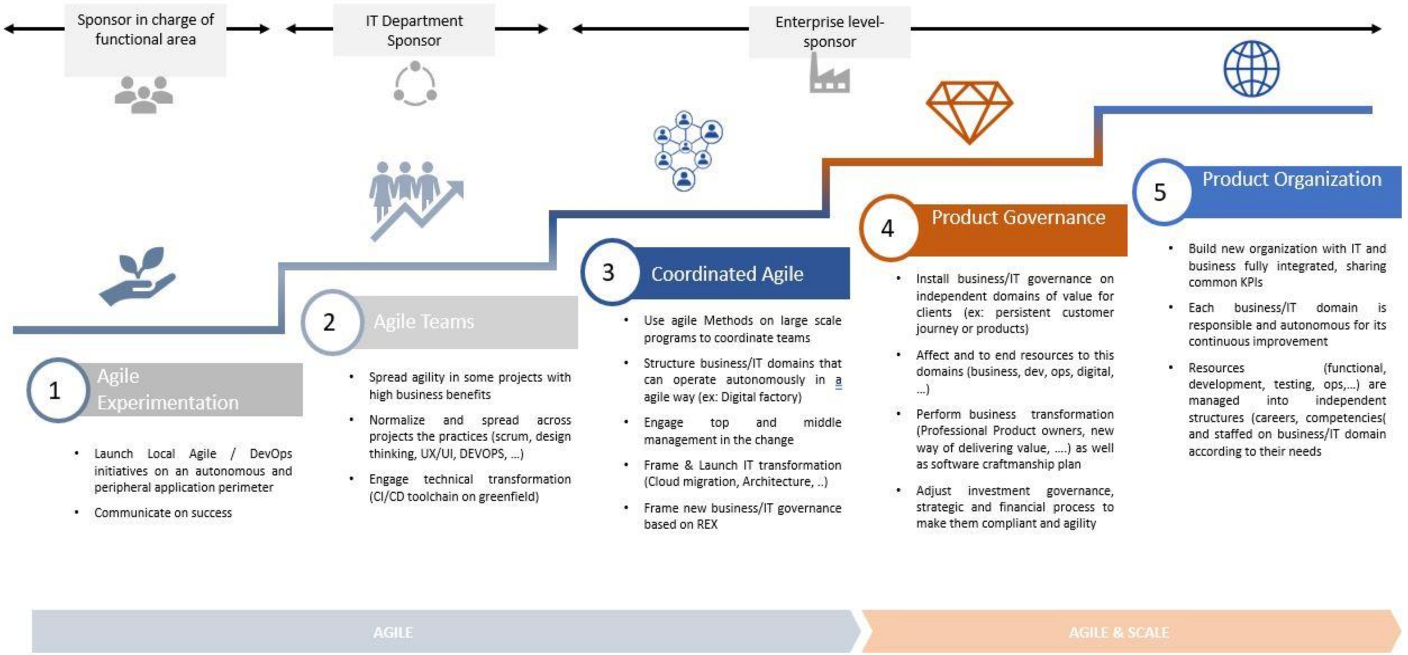
Chapter 7

How to implement Agility

Key phases, processes and tasks to implement agility:

- Implement an agile and proportional decision-making process (top-down and bottom-up).
- Clear definition of roles and responsibilities to carry out the work efficiently.
- Install a common Program Backlog to ensure coherence and priorities projects efficiently.
- Visualization of the product roadmap reacting efficiently to upcoming changes.
- Manage build and run at the same time to ensure proper prioritization and hand-over.
- Increase teams' transparency, velocity and predictability while gaining businesses trust.
- Bring together the most valuable mechanisms from the different frameworks to define the Agile way of working in the company.
- Integrate the company strategic vision in the design of a target according to its needs.
- Assess the expectations properly to ensure proper target definition and monitoring.
- Define a ready and done governance with common gates throughout the organization,
- Synchronize processes for a seamless integration between all steps.
- Gather businesses and IT capabilities to address strategic needs.
- Funding of E2E value chains reviewing periodically the Project Backlog to reallocate resources.
- Keep a healthy mix to allocate budget along the different timeframes, leaving room for decommissioning and technical debt reduction.
- Steering the outcome of the investments done to evaluate the current and potential ROI.

Agile Transformation stages



Sponsor in charge of functional area

IT Department Sponsor

Enterprise level-sponsor

1 Agile Experimentation

- Launch Local Agile / DevOps initiatives on an autonomous and peripheral application perimeter
- Communicate on success

2 Agile Teams

- Spread agility in some projects with high business benefits
- Normalize and spread across projects the practices (scrum, design thinking, UX/UI, DEVOPS, ...)
- Engage technical transformation (CI/CD toolchain on greenfield)

3 Coordinated Agile

- Use agile Methods on large scale programs to coordinate teams
- Structure business/IT domains that can operate autonomously in a agile way (ex: Digital factory)
- Engage top and middle management in the change
- Frame & Launch IT transformation (Cloud migration, Architecture, ..)
- Frame new business/IT governance based on REX

4 Product Governance

- Install business/IT governance on independent domains of value for clients (ex: persistent customer journey or products)
- Affect and to end resources to this domains (business, dev, ops, digital, ...)
- Perform business transformation (Professional Product owners, new way of delivering value, ...) as well as software craftmanship plan
- Adjust investment governance, strategic and financial process to make them compliant and agility

5 Product Organization

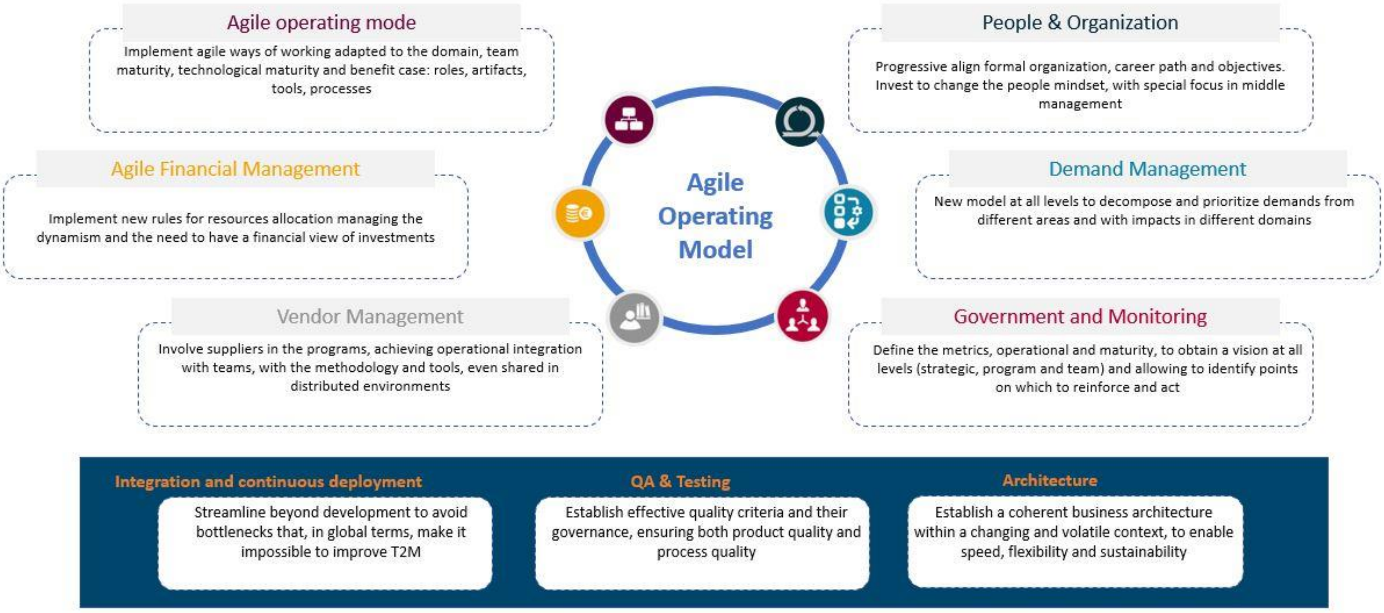
- Build new organization with IT and business fully integrated, sharing common KPIs
- Each business/IT domain is responsible and autonomous for its continuous improvement
- Resources (functional, development, testing, ops,...) are managed into independent structures (careers, competencies(and staffed on business/IT domain according to their needs

AGILE

AGILE & SCALE

Figure 15: Agile transformation stages.

Agile Transformation dimensions



Agile Operating Model

Agile operating mode

Implement agile ways of working adapted to the domain, team maturity, technological maturity and benefit case: roles, artifacts, tools, processes

People & Organization

Progressive align formal organization, career path and objectives. Invest to change the people mindset, with special focus in middle management

Agile Financial Management

Implement new rules for resources allocation managing the dynamism and the need to have a financial view of investments

Demand Management

New model at all levels to decompose and prioritize demands from different areas and with impacts in different domains

Vendor Management

Involve suppliers in the programs, achieving operational integration with teams, with the methodology and tools, even shared in distributed environments

Government and Monitoring

Define the metrics, operational and maturity, to obtain a vision at all levels (strategic, program and team) and allowing to identify points on which to reinforce and act

Integration and continuous deployment

Streamline beyond development to avoid bottlenecks that, in global terms, make it impossible to improve T2M

QA & Testing

Establish effective quality criteria and their governance, ensuring both product quality and process quality

Architecture

Establish a coherent business architecture within a changing and volatile context, to enable speed, flexibility and sustainability

Figure 16: Agile transformation dimensions in the Agile operating model

Typical mistakes when companies try implementing Agility

Many times, organizations find themselves under the pressure to become agile from the "Night to the Next Morning" and during all those rushes we encounter typical mistakes such as:

* The Manager, "who is the one really knows" assigns tasks to the team members and performs "Micromanagement" to monitor that the work is "well done", Error!

* The organization and managers believing that they are already "Agile" become very ambitious, regarding the amount of selected work that the team can address for the next Sprint, Error!

* The Scrum Master decides not to hold the meetings (Planning, Daily, Review, Retro) because the organization thinks it is "wasted a lot of time" or because the size of the team is not large enough or some members areabsent, Error!

Many Managers are focused on achieving only some of the benefits of

Agility such as:

*** Increase productivity**

*** Reduce work in progress (WIP)**

*** Increase the speed of equipment**

But the factors mentioned above are the consequence of working with Agility correctly, not the goal. That is, if we correctly apply the fundamentals of the Agile Manifesto:

- Individuals and interactions, Software running
- Collaboration with customer
- Response to Change.

we will get the benefits mentioned above and a few more, without having to "constantly force the machinery".

To expand these concepts, it is necessary to remember the principles and fundamentals of the Agile Manifesto (www.agilemanifesto.org).

The process of obtaining high-performance Agile teams requires consistency, discipline and involvement everyone in the organization. "Make up" the organization to look like "Agile" without really being, or that it is only for "Moments" is a precarious solution that will sooner or later lead to failure.

Agile adoptions often fail due to a lack of ability to change the company culture, followed by lack of Agile experience and management support.

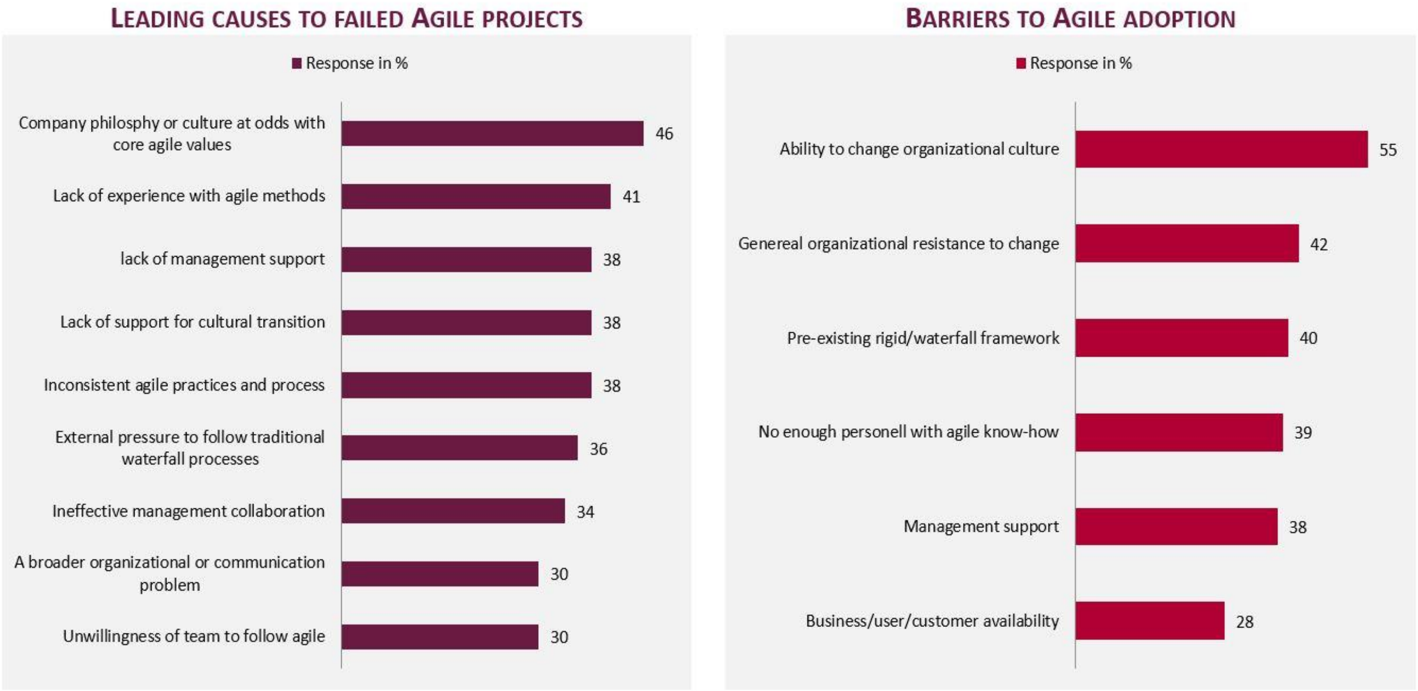


Image 27 Agile Adoption worldwide: *Source: KPMG Global Agile Survey 2019 - 2020*
https://assets.kpmg/content/dam/kpmg/be/pdf/2019/11/agile-transformation.pdf

Basic Strategy for Scaling Agility.

It is important before starting to work with Agile Scrum to define a sustainable initial strategy that allows to continuously improve organizational effectiveness, eliminating (or at least minimizing) the "Muda" (waste).

It is also necessary to identify the additional roles/functions required, and to stand up processes and engage with the teams that need the most support and coaching to achieve the high organizational performance that most companies long for.

It could basically be summarized in the following:

Prepare a Briefing for Agile Transformation Managers and Sponsors.

Perform a high-level design of current problems.

Identify the objectives and set the desired deadlines to achieve them.

Explain the deployment plan for how Agile Scrum will be implemented to achieve the goals.

Explore the current organization and identify initial agile initiatives to apply to the first teams that will become a reference model in Agile Scrum for the company.

Create an Agile leadership and transformation team that designs, prioritizes, and executes the initiatives selected by your organization's managers and sponsors.

Train managers and initiative leaders, Product Owners, Scrum Masters, development Team and keep them informed of the actions of the transformation committee.

Promote the pillars of Agile Scrum (Transparency, Inspection and Adaptation) throughout the organization. It should be remembered that Scrum is based on empiricism, which consists of making decisions based on concrete information obtained through experimentation and observation that will allow to validate hypotheses about the evolution of a product and / or the behavior of a complex system.

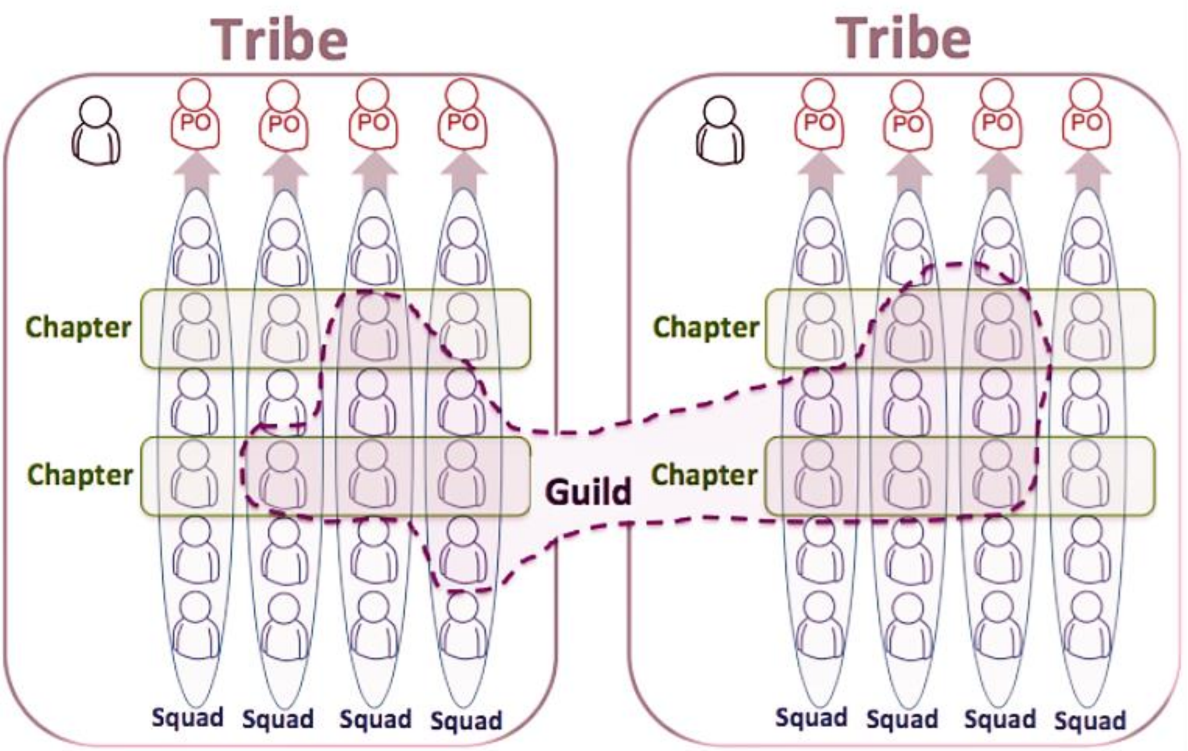
Climb Agile Scrum and launch the first pilots with teams to coordinate, observe and improve product integration.

After several iterations, when the initial agile initiatives are successful, it is necessary to establish the Reference Model, standardize and then continue with the following initiatives prioritized by the transformation committee.

The Spotify model.

Many organizations are emulating the "Spotify Model" or part of it because of its large efficiency, which increases innovation and productivity while at the same time as promotes the autonomy, communication and responsibility of people in teams.

All of these benefits improve the quality and value delivered to customers and rapidly launch new products that promote market demand.



Source Image 15, Henrik Kniberg and Adres Ivarsson website.

The co-creator of the Spotify model is Henrik Kniberg, expert in Lean Agile, XP and Scrum among another knowledge. The model is a very good example of how to scale Agility to large-scale and has made it possible to popularize concepts among the Agile coach community and companies such as: Tribes, Squads, Chapters and Guilds.

Squads: it's the basic development unit, it resembles a Scrum team, they have all the skills and tools needed to design, develop, test and do one or more releases to production.

Tribes: is a collection of Squads working in related areas. Each tribe has a leader, who is responsible for providing the best possible conditions for Squads who are in that Tribe.

The size of the tribes is based on the theoretical concept of "Dunbar's number", it is a limit suggested cognitive to establish the number of people with which an individual can relate effectively.

The tribes hold regular meetings and an informal meeting showing the rest of the tribes (or anyone who wants it) in what they are working on, what they have delivered and what other people can learn from what they are currently consolidating.

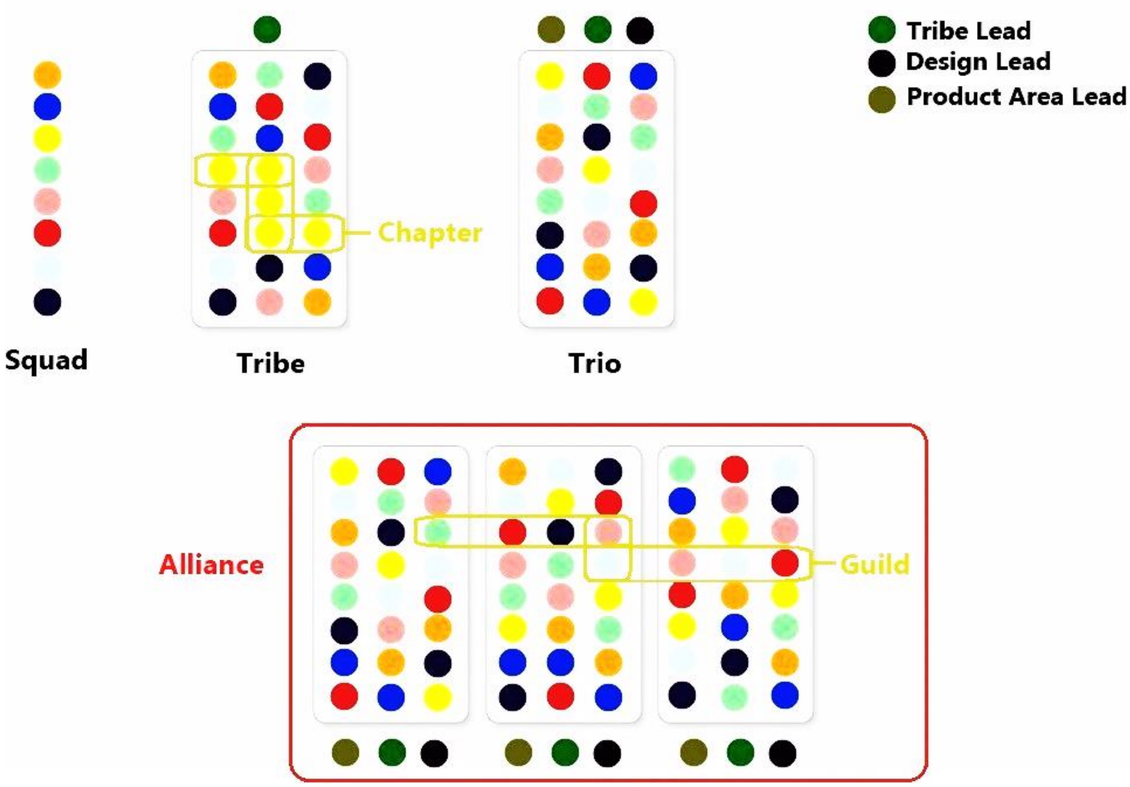


Image 16: Spotify model distribution.

Chapters: They are people who have similar skills and work within the same area competition. Each Chapter meets regularly to discuss its area of expertise and their specific challenges.

The Chapter leader is responsible for people's development and for setting wages among other things, but it's also part of a Squad and is involved in the work allowing you to be continuously in touch with reality and the day-to-day life of Projects.

Guilds: it's a "community of interest" that is formed organically with a group of people who want to share knowledge, new technologies and good practices. Each Guild has a "Guild coordinator" who is responsible for moderating the community.

An example could be "Agile Coachs Guild", as Agile Coaches are spread across organization-wide, but share knowledge and meet frequently to collaborate in the process of continuous improvement of the organization.

The Scrum at Scale Framework.

Scrum a framework for developing, delivering and sustaining complex products by a single team. Since its inception, its usage has extended to the creation of products, processes, services, and systems that require the efforts of multiple teams.

Scrum at Scale was created to efficiently coordinate this new ecosystem of teams in a way that optimizes the overall strategy of the organization. It achieves this goal through setting up a “minimum viable bureaucracy” via a scale-free architecture, which naturally extends the way a single Scrum team functions across the organization.

Dr. Jeff Sutherland developed Scrum at Scale based on the fundamental principles behind Scrum, Complex Adaptive Systems theory, game theory, and object-oriented technology.

Scrum was designed for a single team to be able to work at its optimal capacity while maintaining a sustainable pace. In the field, it was found that as the number of Scrum teams within an organization grew, the optimal output (working product) and velocity of those teams began to fall due to issues like cross-team dependencies and duplication of work. It became obvious that a framework for effectively coordinating those teams was needed in order to achieve linear scalability.

Scrum at Scale is designed to accomplish this goal via its scale-free architecture. By utilizing a scale-free architecture, the organization is not constrained to grow in a particular way determined by a set of arbitrary rules; rather it can grow organically based on its unique needs and at a sustainable pace of change that can be accepted by the groups of individuals that make up the organization.

Scrum at Scale is designed to scale across the organization as a whole: all areas, products and services. It can be applied across multiple domains in all types of organizations in industry, government, or academia.

Scrum at Scale is:

· Lightweight - the minimum viable bureaucracy
· Simple to understand - consists of only Scrum teams
· Difficult to master - requires implementing a new operating model

Scrum at Scale is a framework for scaling Scrum. It radically simplifies scaling by using Scrum to scale Scrum. It consists only of Scrum teams coordinated via Scrum of Scrums and MetaScrums.

The component-based nature of Scrum at Scale allows an organization to customize their transformational strategy and implementation. It gives them the ability to target their transformation efforts in the areas they deem most valuable or most in need of change and then progress on to others.

In Scrum, care is taken to separate accountability of the “what” from the “how”. The same care is taken in Scrum at Scale so that jurisdiction and accountability are expressly understood in order to eliminate wasteful organizational conflict that keep teams from achieving their optimal productivity.

In separating these two jurisdictions, Scrum at Scale contains two cycles:

1. The Scrum Master Cycle (the “how”)
2. The Product Owner Cycle (the “what”)

Each touching the other at two points. Taken together, these cycles produce a powerful framework for coordinating the efforts of multiple teams along a single path.

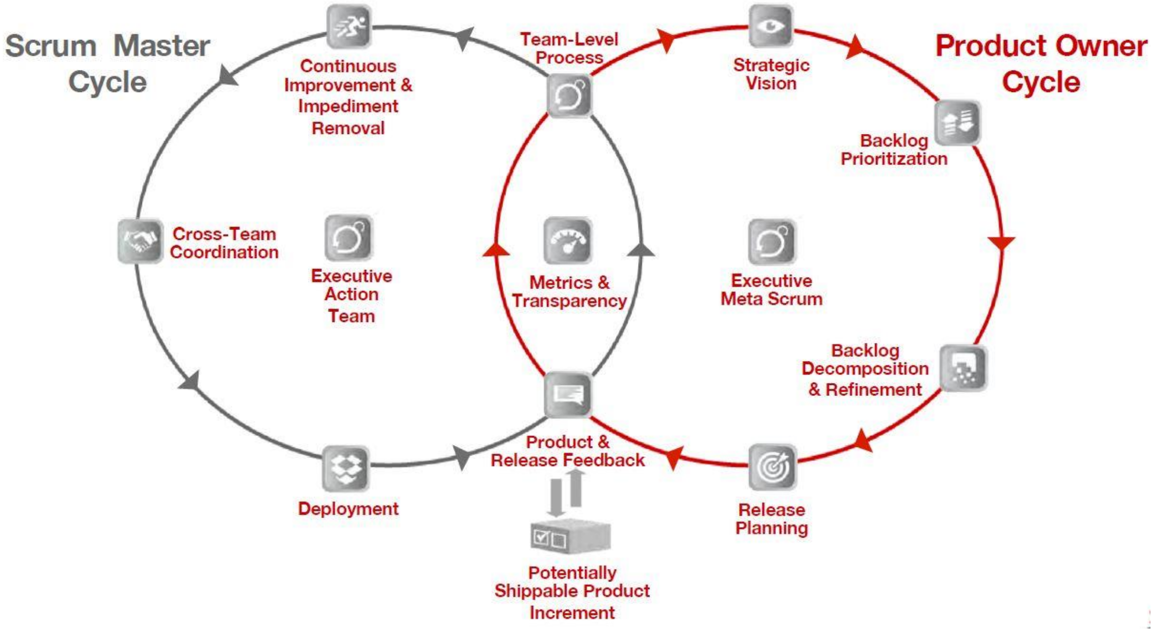


Figure 4, Components of the Scrum at Scale Framework

Values-Driven Culture

Besides separating accountability of the "what" and the "how," Scrum at Scale further aims to build healthy organizations by creating a values-driven culture in an empirical setting.

The Scrum values are:

- Openness
- Courage
- Focus
- Respect
- Commitment

These values drive empirical decision making, which depend on the three pillars of transparency, inspection, and adaptation.

Scrum at Scale helps organizations thrive by supporting both a servant-leadership and intent based leadership model, which fosters a positive environment for working at a sustainable pace and putting commitment to deliver customer-facing value at the forefront of our efforts.

When implementing large networks of teams, it is critical to develop a scalable Reference Model for a small set of teams. Any deficiencies in a Scrum implementation will be magnified when multiple teams are deployed.

Therefore, the first challenge is to create a small set of teams that implements Scrum well. This set of teams works through organizational issues that block agility and creates a Reference Model for Scrum that is known to work in the organization and can be used as a pattern for scaling Scrum across the organization.

As the Reference Model of teams accelerates, impediments and bottlenecks that delay delivery, produce waste, or impede business agility become apparent. The most effective way to eliminate these problems is to spread Scrum across the organization so that the entire value stream is optimized.

Scrum at Scale achieves linear scaling in productivity by saturating the organization with Scrum and distributing velocity and quality organically, consistent with the organization’s specific strategy, product, and services.

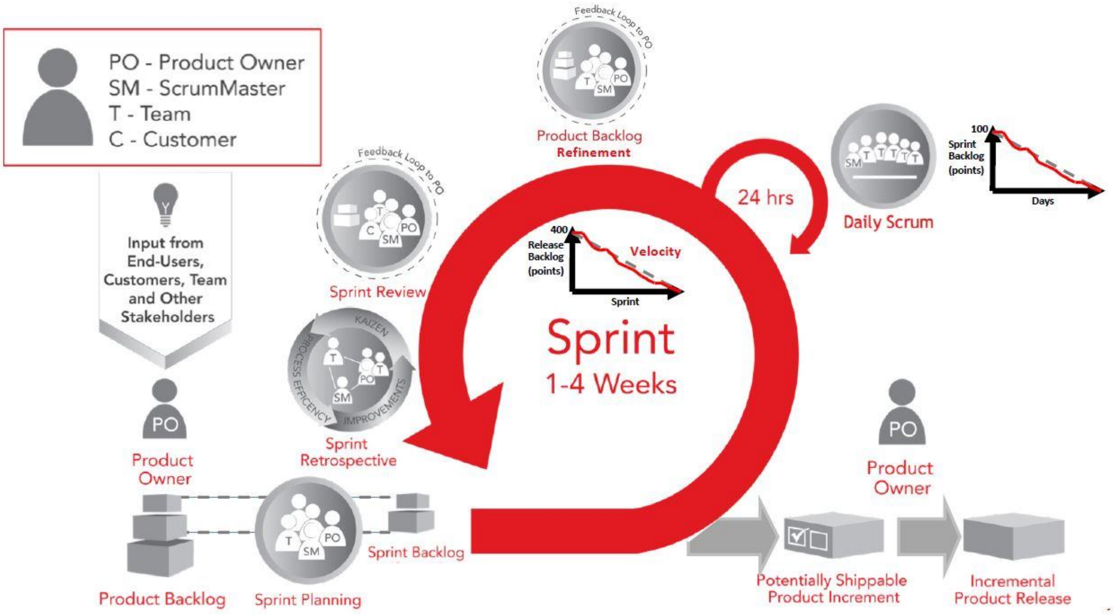


Figure5: Team-Level Process

The Team-Level Process constitutes the first touch point between the Scrum Master and Product Owner Cycles. It is composed of three artifacts, five events, and three roles.

The goals of the team level process are to:

- Maximize the flow of completed and quality tested work.
- Increase velocity a little each Sprint.
- Operate in a way that is sustainable and enriching for the team.

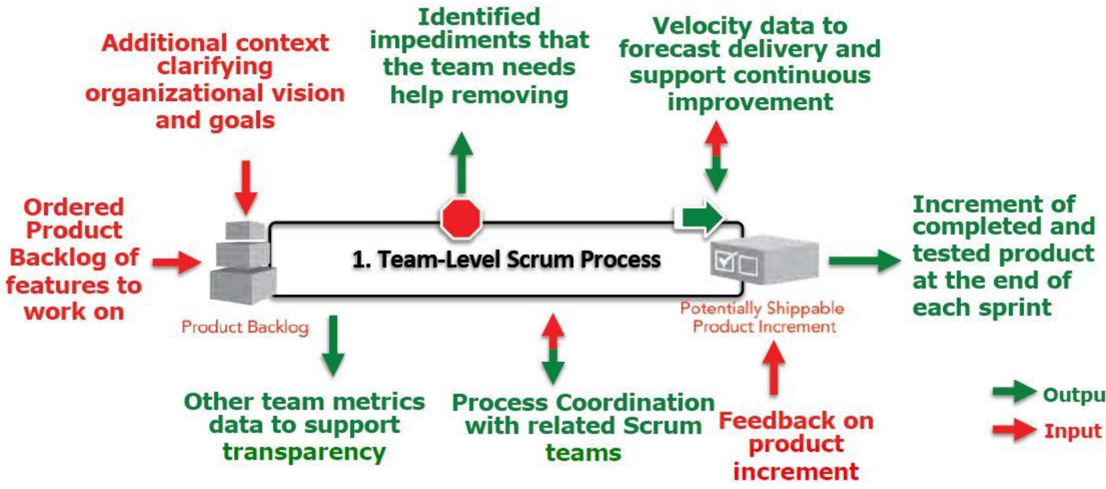

Additional context clarifying organizational vision and goals
Identified impediments that the team needs help removing
Velocity data to forecast delivery and support continuous improvement
Ordered Product Backlog of features to work on
1. Team-Level Scrum Process
Product Backlog
Potentially Shippable Product Increment
Increment of completed and tested product at the end of each sprint
Other team metrics data to support transparency
Process Coordination with related Scrum teams
Feedback on product increment
Output
Input

Figure 6: Team-Level Process workflow

Scaling the Scrum of Scrum (SoS)

Depending upon the size of the organization or implementation, more than one SoS may be needed to deliver a very complex product. In those cases, a Scrum of Scrum of Scrums (SoSoS) can be created out of multiple Scrums of Scrums. The SoSoS is an organic pattern of Scrum teams which is infinitely scalable. Each SoSoS should have SoSoSM's and scaled versions of each artifact & event.

Scaling the SoS reduces the number of communication pathways within the organization so that complexity is encapsulated. The SoSoS interfaces with a SoS in the exact same manner that a SoS interfaces with a single Scrum team which allows for linear scalability.

Harvard research determined that optimal team size of a team is 4.6 peoples. Experiments with high performing Scrum teams have repeatedly shown that 4 or 5 people doing the work is the optimal size.

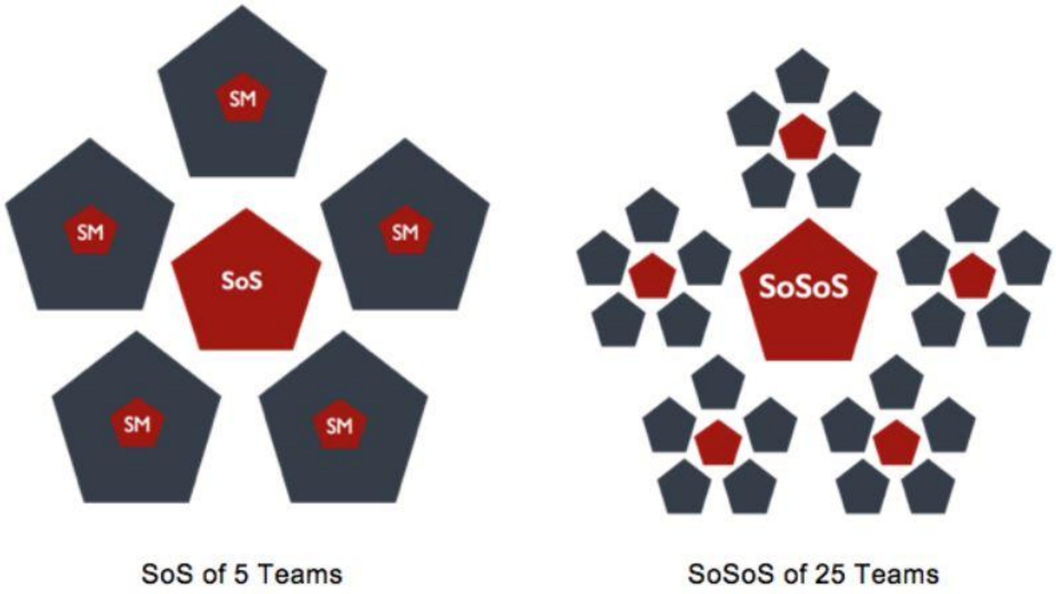


Figure 7: Scrum of Scrum teams and scalability.

It is essential to linear scalability that this pattern be the same for the number of teams in a SoS. Therefore, in the above and following diagrams, pentagons were chosen to represent a team of 5. These diagrams are meant to be examples only, your organizational diagram may differ greatly.

The Executive Action Team

The Scrum of Scrums for the entire agile organization is called the Executive Action Team (EAT). The EAT is the final stop for impediments that cannot be removed by the SoS's that feed it. Therefore, it must be comprised of individuals who are empowered, politically and financially, to remove them.

The function of the EAT is to coordinate multiple SoS's (or SoSoS's). As with any Scrum team, it needs a PO and SM. It would be best if the EAT met daily as a Scrum team. They must meet at least once per Sprint and have a transparent backlog.

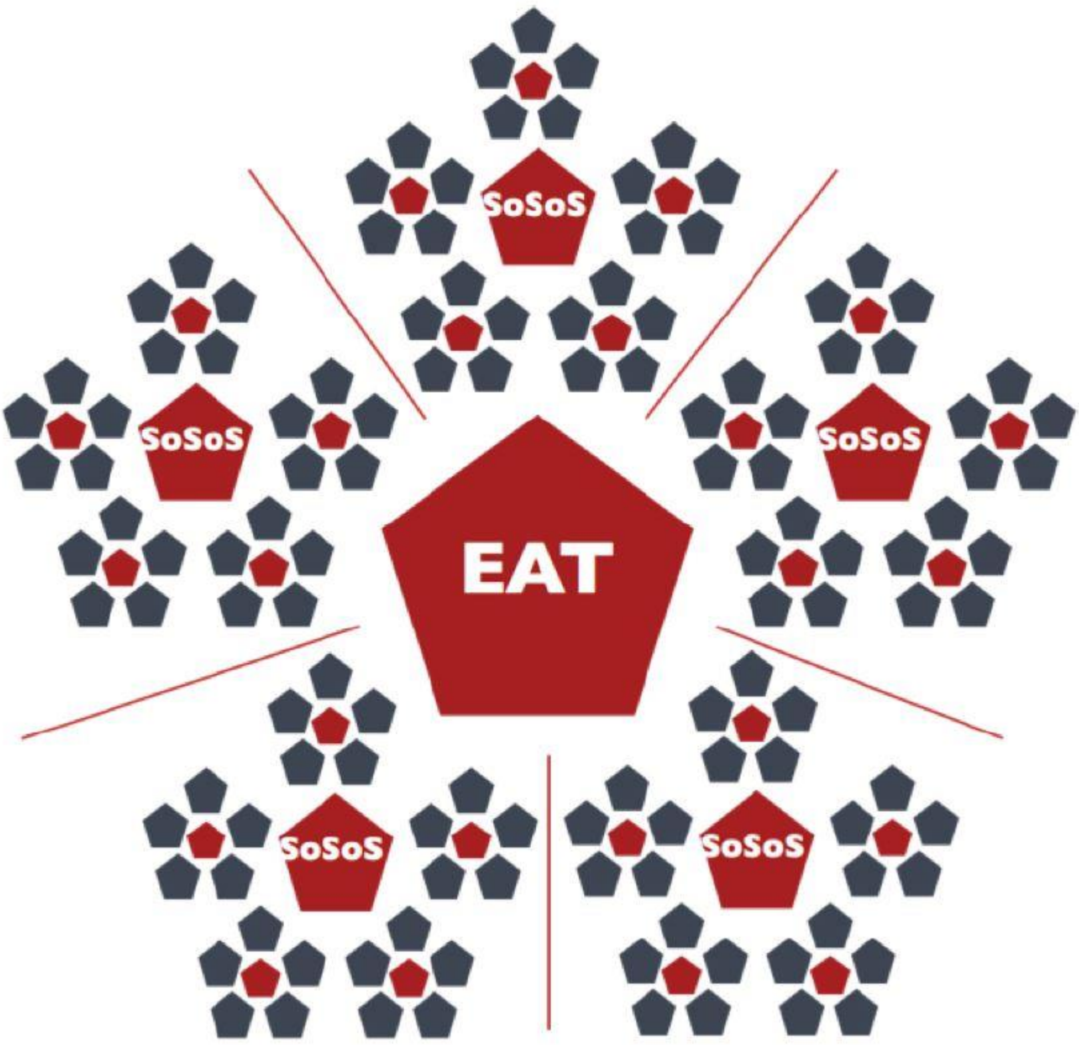


Diagram showing the coordination of the EAT with 5 groups of 25 teams.

The EAT's Backlog & Responsibilities

Scrum is an agile operating system that is different from traditional project management.
The entire SM organization reports into the EAT, which is responsible for implementing this agile operating system by establishing, maintaining, and enhancing the implementation in the organization. The EAT's role is to create an Organizational Transformation Backlog (a prioritized list of the agile initiatives that need to be accomplished) and see that it is carried out.

The EAT is accountable for the quality of Scrum within the organization. Its responsibilities include but are not limited to:

- Creating an agile operating system for the Reference Model as it scales through the organization.
- Including corporate operational rules, procedures, and guidelines to enable agility.
- Measuring and improving the quality of Scrum in the organization.
- Building capability within the organization for business agility.
- Creating a center for continuous learning for Scrum professionals.
- Supporting the exploration of new ways of working.

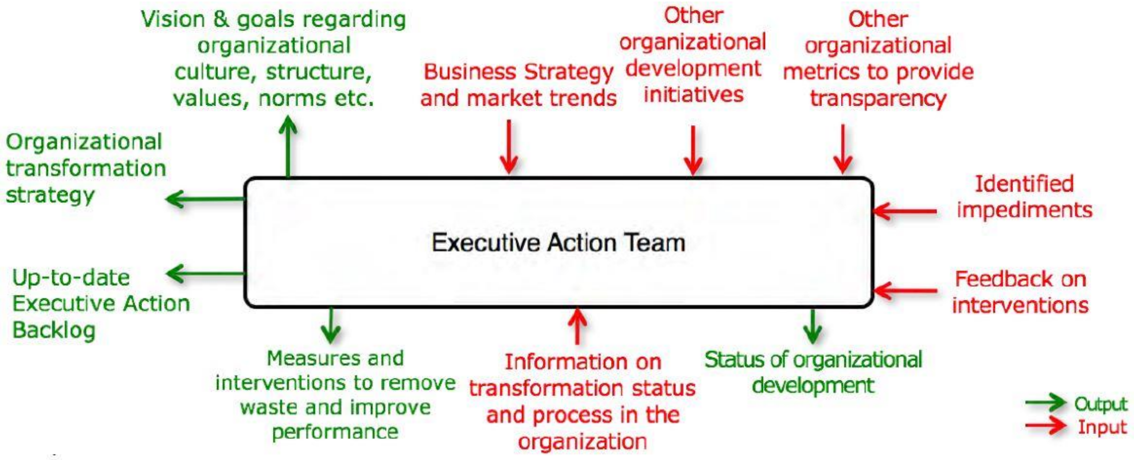

Vision & goals regarding organizational culture, structure, values, norms etc.
Business Strategy and market trends
Other organizational development initiatives
Other organizational metrics to provide transparency
Organizational transformation strategy
Identified impediments
Executive Action Team
Up-to-date Executive Action Backlog
Feedback on interventions
Measures and interventions to remove waste and improve performance
Information on transformation status and process in the organization
Status of organizational development
Output
Input

Figure 8: Executive Action Team inputs and outputs

Decision Latency Impact on Project Success.

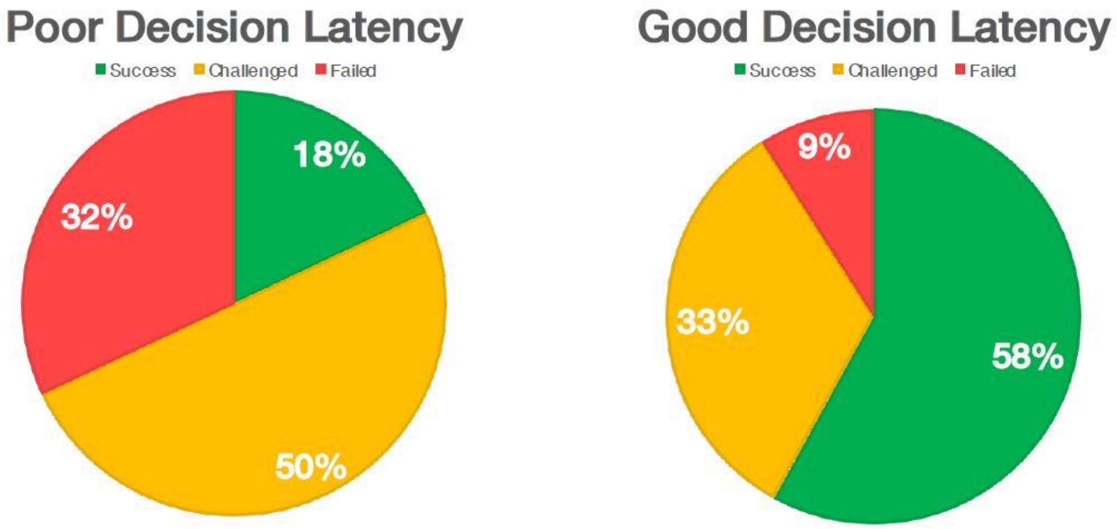


Figure 9: Decision Latency Impact on Project Success.

Decision Latency is emerging as primary driver for project success

The goals of Continuous Improvement and Impediment Removal are to:

- Identify impediments and reframe them as opportunities.
- Maintain a safe and structured environment for prioritizing and removing impediments, and then verifying the resulting improvements.
- Ensure visibility in the organization to effect change.

The goals of Cross-Team Coordination are to:

- Coordinate similar processes across multiple related teams.
- Manage cross-team dependencies to ensure they don't become impediments.
- Maintain alignment of team norms and guidelines for consistent output.

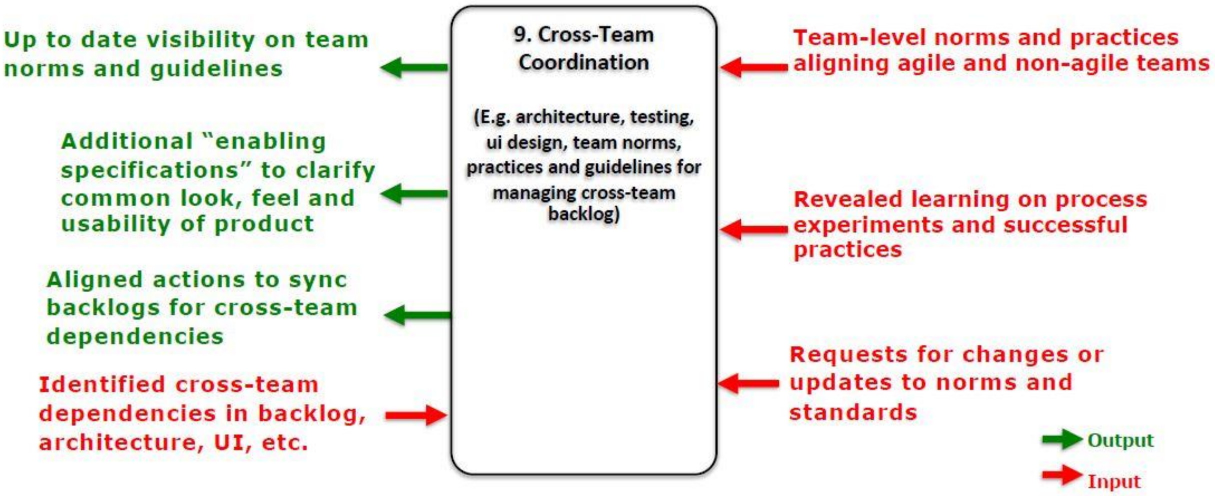

Up to date visibility on team norms and guidelines
Additional "enabling specifications" to clarify common look, feel and usability of product
Aligned actions to sync backlogs for cross-team dependencies
Identified cross-team dependencies in backlog, architecture, UI, etc.
9. Cross-Team Coordination
(E.g. architecture, testing, ui design, team norms, practices and guidelines for managing cross-team backlog)
Team-level norms and practices aligning agile and non-agile teams
Revealed learning on process experiments and successful practices
Requests for changes or updates to norms and standards
Output
Input

Figure 10: cross team coordination inputs and outputs

Since the goal of the SoS is to function as a release team, the deployment of product falls under their scope, while what is contained in any release falls under the scope of the Product Owners.

Therefore, the goals of the Deployment are to:

- Deliver a consistent flow of valuable finished product to customers.
- Integrate the work of different teams into one seamless product.
- Ensure high quality of the customer experience.

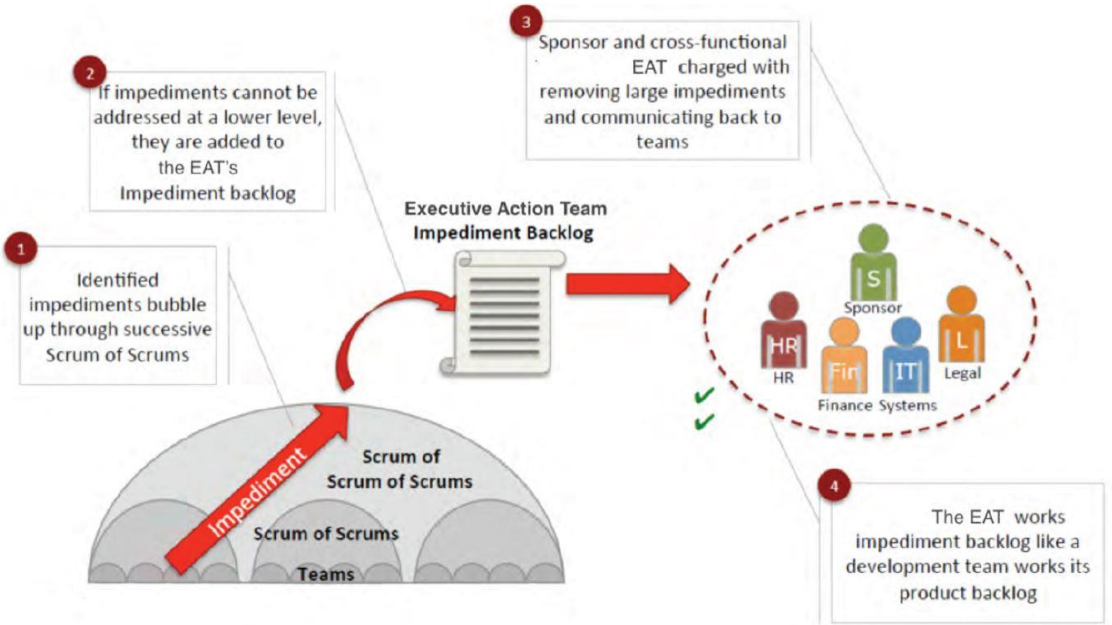

1
Identified impediments bubble up through successive Scrum of Scrums
2
If impediments cannot be addressed at a lower level, they are added to the EAT's Impediment backlog
3
Sponsor and cross-functional EAT charged with removing large impediments and communicating back to teams
Executive Action Team Impediment Backlog
S
Sponsor
HR
HR
Fin
Finance
IT
Systems
L
Legal
Impediment
Scrum of Scrum of Scrums
Scrum of Scrums
Teams
4
The EAT works impediment backlog like a development team works its product backlog

Figure 11: Executive Action Team workflow

The Executive MetaScrum (EMS)

The MetaScrums enable a network design of Product Owners which is infinitely scalable alongside their associated SoS's. The MetaScrum for the entire agile organization is the Executive MetaScrum.

The EMS owns the organizational vision and sets the strategic priorities for the whole company, aligning all the teams around common goals.

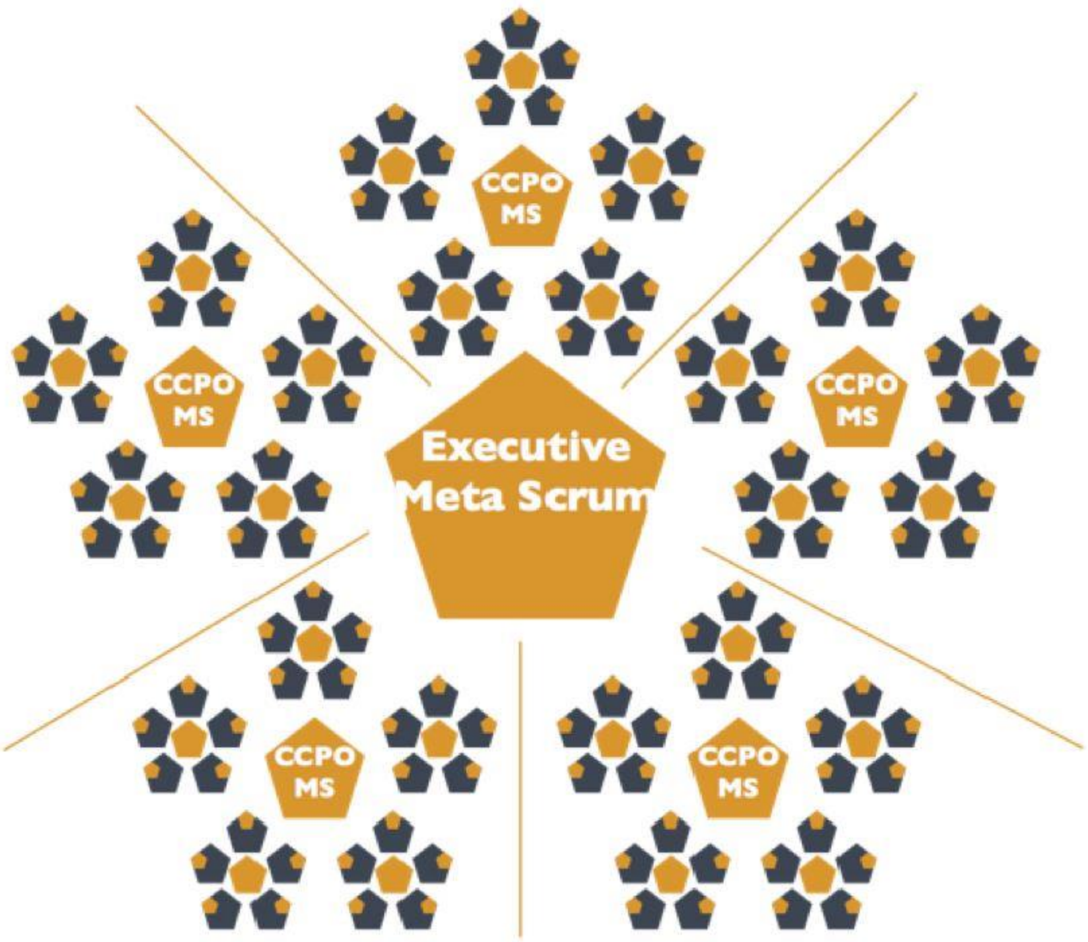


Figure 12: Sample diagram showing an EMS coordinating 5 groups of 25 teams:

Outputs/Outcomes of the Product Owner Organization

The PO organization (various MetaScrums, the CPO's, and the Executive MetaScrum) work to satisfy the components of the Product Owner Cycle: Strategic Vision, Backlog Prioritization, Backlog Decomposition & Refinement, and Release Planning.

The goals of setting a Strategic Vision are to:

- Clearly align the entire organization along a shared path forward.
- Compellingly articulate why the organization exists.
- Describe what the organization will do to leverage key assets in support of its mission.
- Update continuously to respond to rapidly changing market conditions.

The goals of Backlog Prioritization are to:

- Identify a clear ordering for products, features, and services to be delivered.
- Reflect value creation, risk mitigation and internal dependencies in ordering of the backlog.
- Prioritize the high-level initiatives across the entire agile organization prior to Backlog decomposition and refinement.

The goals of Backlog Decomposition & Refinement are to:

- Break complex projects and products into independent functional elements that can be completed by one team in one Sprint.
- Capture and distill emerging requirements and customer feedback.
- Ensure all backlog items are truly "Ready" so that they can be pulled by the individual teams.

The goals of Release Planning are to:

Forecast delivery of key features and capabilities.

Communicate delivery expectations to stakeholders.

Update prioritization, as needed.

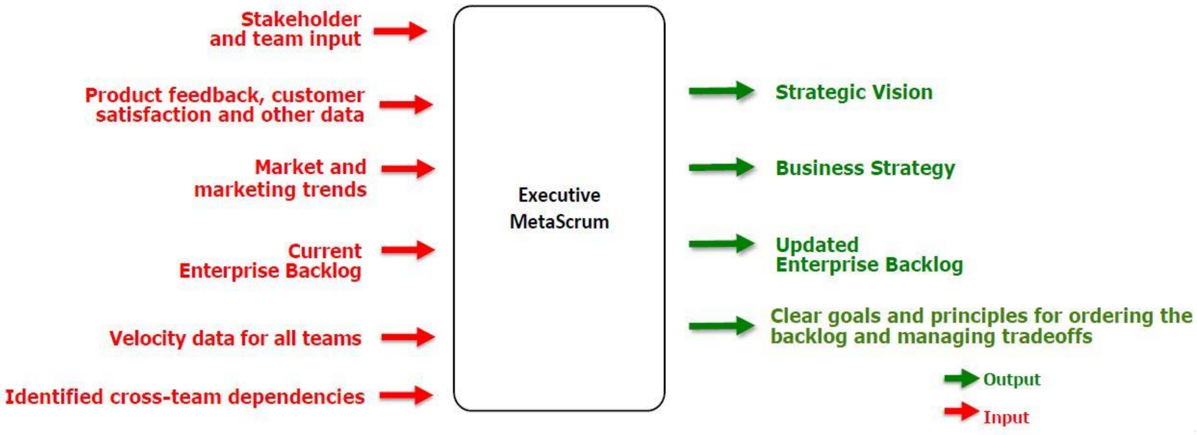

Stakeholder and team input
Product feedback, customer satisfaction and other data
Market and marketing trends
Current Enterprise Backlog
Velocity data for all teams
Identified cross-team dependencies
Executive MetaScrum
Strategic Vision
Business Strategy
Updated Enterprise Backlog
Clear goals and principles for ordering the backlog and managing tradeoffs
Output
Input

Figure 13: Executive MetaScrum, inputs and outputs

Understanding Feedback

The Feedback component is the second point where the PO & SM Cycles touch. Product feedback drives continuous improvement through adjusting the Product Backlog while release feedback drives continuous improvement through adjusting the Deployment mechanisms.

The goals of obtaining and analyzing Feedback are to:

- Validate our assumptions.
- Understand how customers use and interact with the product.
- Capture ideas for new features and functionality.
- Define improvements to existing functionality.
- Update progress towards product/project completion to refine release planning and stakeholder alignment.
- Identify improvements to deployment methods and mechanisms.

Metrics & Transparency

Scrum at Scale does not require any specific set of metrics, but it does suggest that at a bare minimum, the organization should measure:

- Productivity
- Value Delivery
- Quality
- Sustainability

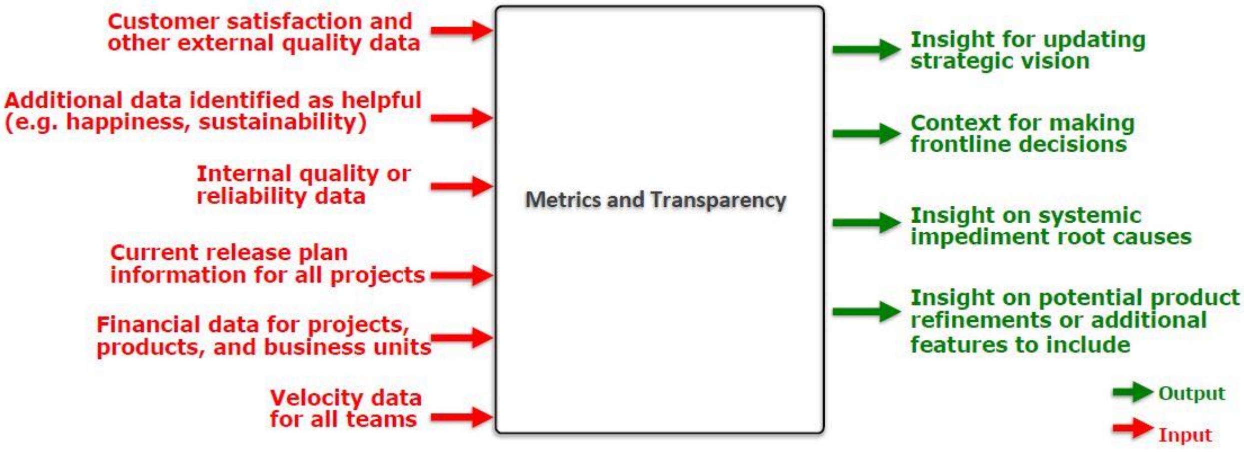

Customer satisfaction and other external quality data
Additional data identified as helpful (e.g. happiness, sustainability)
Internal quality or reliability data
Current release plan information for all projects
Financial data for projects, products, and business units
Velocity data for all teams
Metrics and Transparency
Insight for updating strategic vision
Context for making frontline decisions
Insight on systemic impediment root causes
Insight on potential product refinements or additional features to include
Output
Input

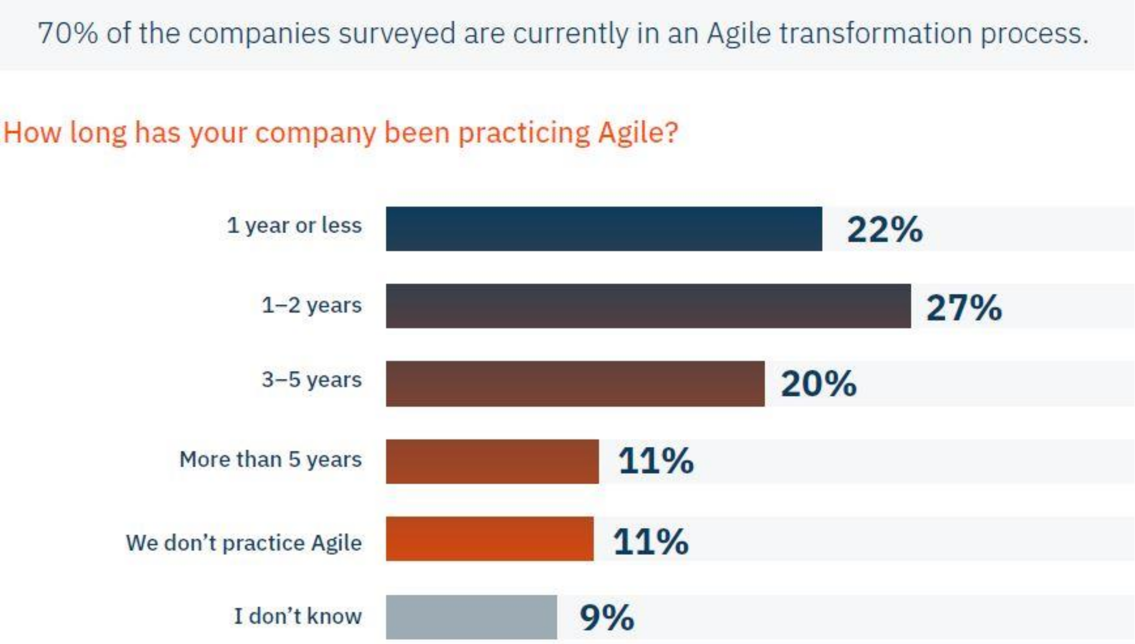
Figure 14: Metrics and transparency, inputs and outputs

Radical transparency is essential for Scrum to function optimally, but it is only possible in an organization that has embraced the Scrum values. It gives the organization the ability to honestly assess its progress and to inspect and adapt its products and processes. This is the foundation of the empirical nature of Agile Scrum. Both the Scrum Master and Product Owner cycles require metrics that will be decided upon by the separate SM and PO organizations.
Metrics may be unique to both specific organizations as well as to specific functions within those organizations.

Agile Scrum in worldwide largest companies.

According to the Project Management Institute (PMI) and the CertiProf Agile adoption report 2020, 70% of organizations worldwide are using Agile methodology in their projects, in a frequent way.

Company experience & agile adoption



70% of the companies surveyed are currently in an Agile transformation process.

How long has your company been practicing Agile?

Source Image 17: CertiProf Agile adoption report 2020

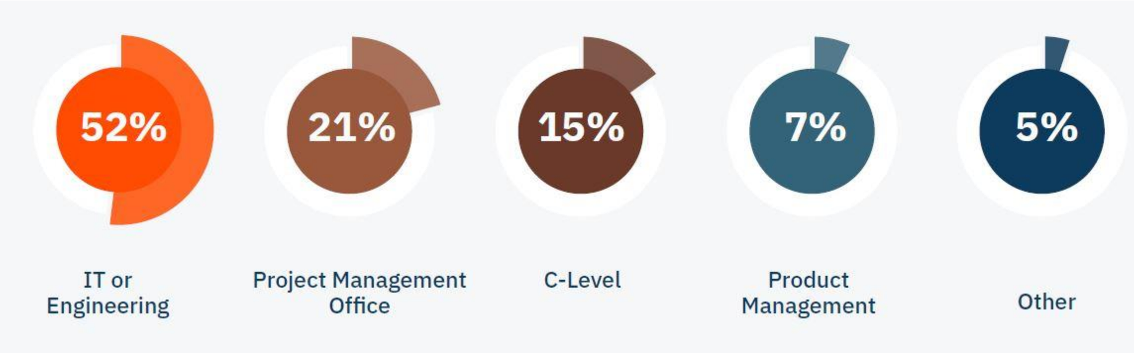
70% of organizations worldwide are using Agile methodology in their projects, often over the past 12 months. This is one of the main conclusions drawn from the annual project management institute (PMI), known as Pulse of the Profession. The report that shows the market trend in terms of Project Management and the methodology to bring them to fruition

From the research of the latest Pulse of the Profession, there is also a greater interest, on the part of many organizations, in being more agile, competitive, and customer centric.

During 2020, the study confirmed this trend in both agile transformation and PMO issues, which are driving business change. Both people, processes and corporate culture are focusing on finding greater agility and demand for greater innovation.

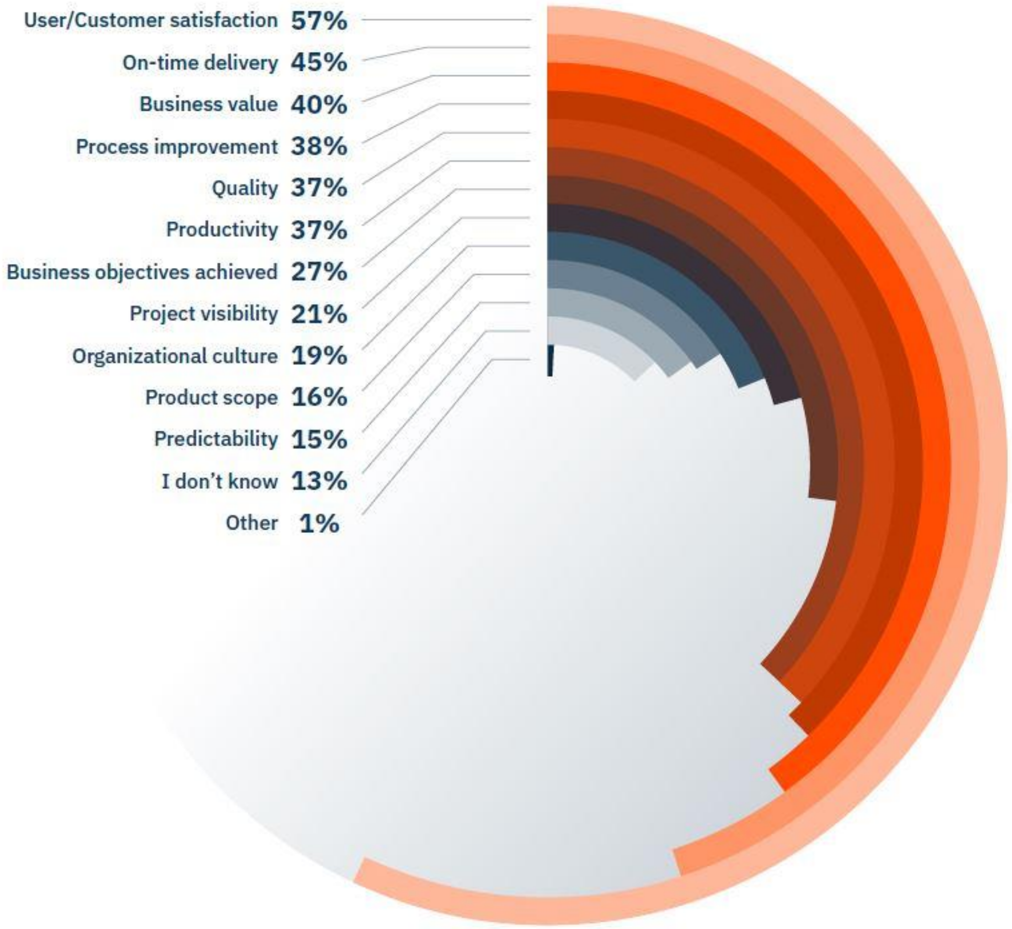
The study has shown that, in the last 12 months, one in five projects has exclusively used agile approaches, and many others have used hybrid or combined approaches. In addition, this implementation of agile methodologies has a direct impact on the return on investment of organizations. According to the report, agile companies increase revenue by 37% faster, generating 30% more profits from companies that are not agile managed.

Which area is in charge of managing Agile methodologies within your organization?



Source Image 18: CertiProf Agile adoption report 2020

How does your organization measure the success of Agile transformations?



Source Image 19: CertiProf Agile adoption report 2020

Some examples of Large companies using agile methodologies

In the following table, these cases present a verifiable application of the success of this methodology to distribute work and achieve efficiency. In the following table there are examples of large companies from various sectors that have properly implemented Agile to manage their projects.

Industry sectors	Large companies using agile methodologies
Software, Hardware	Adobe, Biko2, Spotify Central Desktop, Citrix, Gailén, IBM, Intel, Microfocus, Microsoft, Novell, OpenView Labs, Plain Concepts, Primavera, Proyectalis, Softhouse, Valtech, VersionOne, Apple.
Media y Telecomunications	BBC, BellSouth, British Telecom, DoubleYou, Motorola, Nokia, Palm, Qualcomm, Schibsted, Sony/Ericsson, Telefonica I+D, TeleAtlas, Verizon
Internet	Amazon, Google, Mozilla
ERP	SAP
Banking and Investment	Bank of America, Barclays Global Investors, Key Bank, Merrill Lynch
Health	Philips Medical
Military defense and Aerospace	Boeing, General Dynamics, Lockheed Martin, Saab
Video games	Blizzard, High Moon Studios, Crytek, Ubisoft, Electronic Arts
Multimedia	Bose
Automotive industry	Ferrari, BMW, Porsche
Industrial equipment	3M

Table 1: Some examples of Large companies using agile methodologies

It is very important to emphasize that culture is an important part of the execution of Agile and Scrum in large companies, it is not enough to divide the staff into smaller teams, collaboration between these teams must be achieved and that the participants of the same are capable of follow the agile methodology.

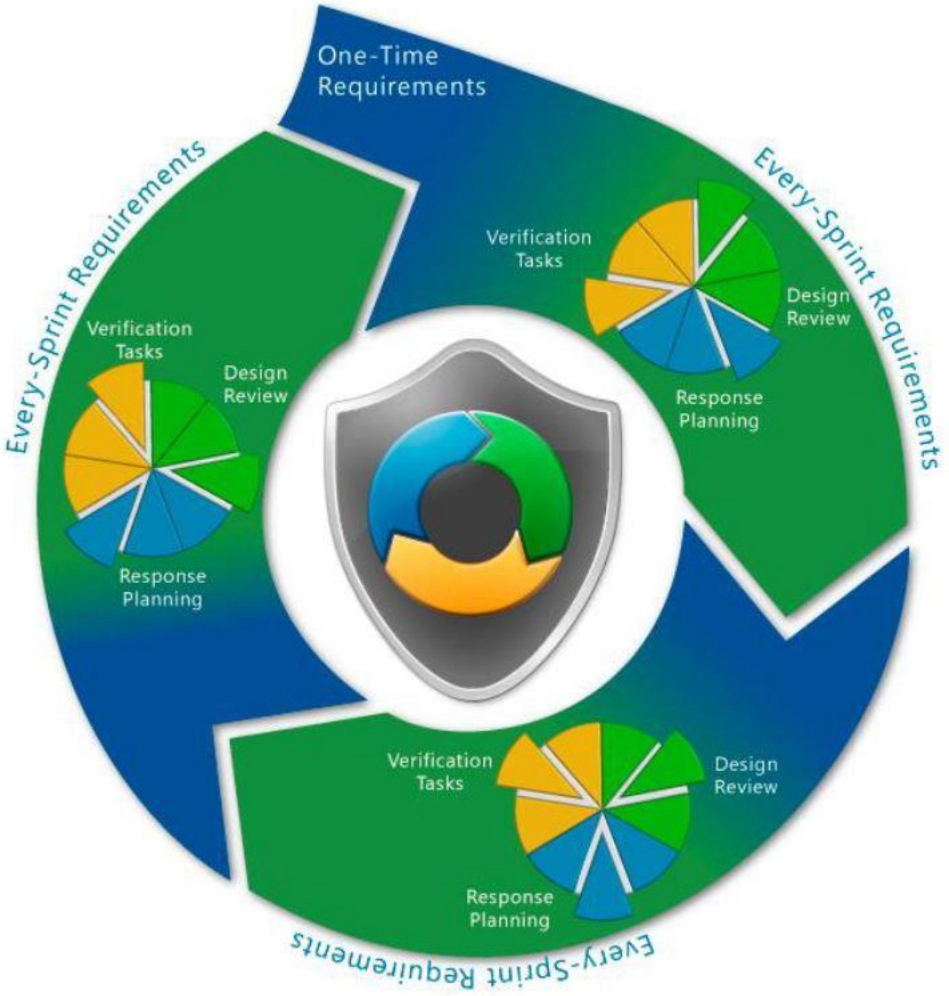
Agile methodology in Microsoft

Microsoft Headquarters, Redmond, Washington - USA. Source picture 4: news.microsoft.com/life

Software development giant Microsoft has adopted agile development practices since the launch of SQL Server 2005, according to one of its vice presidents, agile adoption was not an imposition (from the high levels), but was done by incentivize and encourage. It began with Scrum and Extreme Programming (XP) practices.

Instead of adopting all the practices of the beginning, we started by taking some ideas, for example Scrum took the one to bring the whole team together every day for half an hour, decide what they're going to do and then get the job done quickly. Extreme Programming (XP) adopted peer programming, applying the concept that two minds work best finding problems faster.

Microsoft SDL For Agile Development Source



Source Image 20: Microsoft Development Source SDL v4.1a

Microsoft learned that instead of establishing software development processes that should be followed to the letter, it is better to set quality levels, and give each team the flexibility to achieve those results in a way that is most effective for each one.

Agile adoption was not mandatory but voluntary, and the results were varied, some successful and others were not. Annex presents a link to a paper on an exploratory study of the use and perception of software development.

The fact that each Microsoft team was free to adopt agile practices according to their convenience, resulted in the development of own (internal) methodologies, an example is Customer Connected Engineering (CCE), developed by the Patterns and Practices team.

Agile Scrum methodology in Amazon

Amazon’s largest office, Seattle, USA.

They were present the Agile project management from 2005, but it was not until 2009 that a mass adoption of Agile Scrum was carried out on Amazon.

The transition was not made through centralized planning, which differs from the methods that are typically used in such transformation projects in enterprises. The following are several factors that enabled the transition to agile practices on Amazon.

Amazon teams have always been given breadth in decision-making to solve their problems. They are encouraged not to have to wait for external intervention to solve these problems.

They are not overloaded with predictive practices dictated by a centralized corporate, instead they have breadth to adopt their own practices.

The emphasis is on developing their processes, as light as possible, that allow them to create, deliver and operate the software. The teams at Amazon, have extensive freedom of choice.

Amazon's culture has always been consistent with agile practices.

The concept of "two-pizza equipment" has been present on Amazon since its inception. If you look closely, a two-pizza team is just like a Scrum team (just removing the word Scrum).A team reports to a single Direct Manager.

The two-pizza rule and the secret of Amazon's success

Jeff Bezos (CEO of Amazon), instituted a rule in the early days of his company: Each internal team must be small enough to be able to be fed two pizzas. The goal was not to reduce the catering bill. Like almost everything Amazon does, it focused on two goals: **efficiency** and **scalability**.

The thing about having lots of small teams is that they all need to be able to work together, and to be able to access the common resources of the company, in order to achieve their larger goals.

Amazon Marketplace launched in 2000, allowing third-party sellers to put up their own wares on the site. The feature has expanded over the years to become a major plank in the company’s quest to be the “everything store” – the one destination on the internet you need to go to to buy anything in existence.

Marketplace goes one better than the pizza rule, allowing Amazon to expand into new sectors without needing to employ a single extra employee.

Amazon's distribution center in Phoenix, Arizona. Photograph: Ralph Freso / Reuters/Reuters

How Scrum was adopted on Amazon

Someone inside Amazon (on one of the teams) decided to spend time driving himself and his teammates into Scrum.

Teams that showed interest in Scrum were able to make local decisions to implement it.

To the extent that these pioneering teams showed good results, the interest of other teams, who in turn began to adopt it, was aroused.

Over time, it became a scale initiative across the organization. An email-based Scrum community was formed. Scrum process and tool training was provided by volunteers who "just wanted to do it."

Over time, when enough teams adopted Scrum, the position of Scrum Coach was created. This facilitated adoption and improved the quality of Scrum deployments.

The first Scrum meeting was held in "Open Space" format in 2008.

All this is achieved without the participation of the high levels of the organization, there was never executive level support and the transition was limited at the team level. This causes organizational impediments to full adoption.

Adopting Scrum on Amazon validates an easy, frictionless method, in which: stable equipment is first established (this is a necessary starting point), then scrum information is disseminated and these teams are given permission to adopt it according to the practices and form they consider best.

The advantage of this approach is that it does not require enterprise-level planning.

Another advantage is that teams voluntarily adopt Scrum, rather than by executive-level guidelines from the organization.

Transition management under this approach is limited to reviewing and eliminating impediments to teams.

The main impediment was lack of knowledge, most of these can be solved with more information and training. Once this was resolved, the adoption progressed smoothly.

The strength of this methodology is that only the teams that were interested were the ones who tested Scrum, without an explicit mandate or default schedule, no one was forced to use Scrum. Each team made independent decisions to solve the problems. This is a pull-based approach instead of the command-control "Push" approach.

At first, the transition did not exceed the team level because the average management and senior management was not involved.

The participation of these groups is necessary to achieve maximum potential.

Training for Medium Management is crucial, but support from senior management is required for it to take place. A transition team is crucial to addressing impediments at the medium and business level.

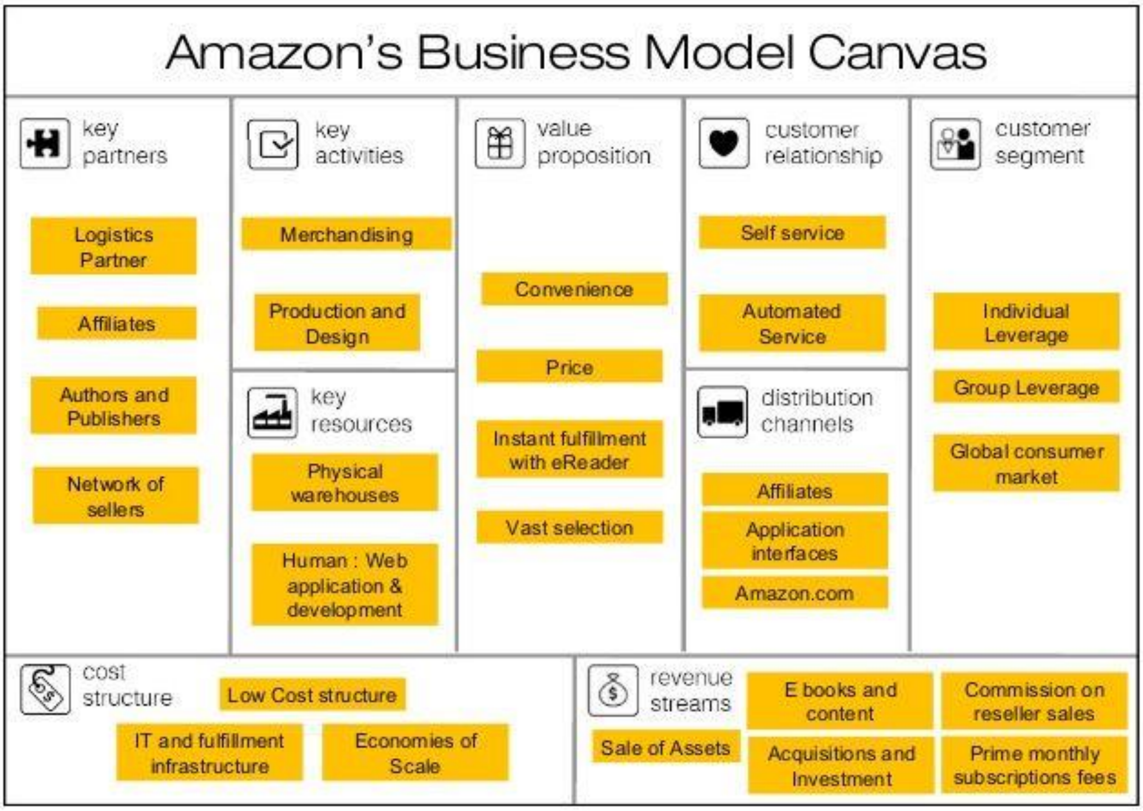
Amazon Leadership Principles

The Amazon Leadership Principles are 14 fundamental values that govern the company and its employees. The company insists that Amazon Leadership Principles are implemented daily on the job, and even in interview questions. Amazon also believes in hiring people who align with these principles and tests for these qualities during interviews.

Amazon Leadership Principles and corporate culture:

1. **Customer Obsession**
2. **Ownership**
3. **Invent and Simplify**
4. **Are Right, A Lot**
5. **Learn and Be Curious**
6. **Hire and Develop the Best**
7. **Insist on the Highest Standards**
8. **Think Big**
9. **Bias for Action**
10. **Frugality**
11. **Earn Trust**
12. **Dive Deep**
13. **Have Backbone; Disagree and Commit**
14. **Deliver Results**

Amazon Business Model Canvas



Amazon's Business Model Canvas

key partners
- Logistics Partner
- Affiliates
- Authors and Publishers
- Network of sellers

key activities
- Merchandising
- Production and Design

key resources
- Physical warehouses
- Human : Web application & development

value proposition
- Convenience
- Price
- Instant fulfillment with eReader
- Vast selection

customer relationship
- Self service
- Automated Service

distribution channels
- Affiliates
- Application interfaces
- Amazon.com

customer segment
- Individual Leverage
- Group Leverage
- Global consumer market

cost structure
- Low Cost structure
- IT and fulfillment infrastructure
- Economies of Scale

revenue streams
- Sale of Assets
- E books and content
- Commission on reseller sales
- Acquisitions and Investment
- Prime monthly subscriptions fees

Image 21, Amazon Business Canvas Model, Source: *https://www.pinterest.com/pin/20829217011453286/*

Agile and Scrum at Scale in Military development.

The high complexity of advanced military systems requires an optimal high-performance model quality for its development, due to the enormous costs in R&D, design and manufacturing.

That's why SAAB Defense has adopted an Agile model to produce the new

Fighter generation "JAS 39E SAAB Gripen" in both the construction equipment of the aircraft as in the development of the software that is implemented in it.

Figther Aircraf Saab JAS 39 Gripen. Source picture 5 : https://corporalfrisk.com/tag/jas-39-gripen/

The development of the Gripen fighter is very complex with more than 1000 engineers grouped in more than 100 teams. Scrum at Scale provides the optimal tools to manage variability and risks quickly and dynamically.

Saab introduced Agile practices about ten years ago during the upgrades from previous versions of the "Gripen fighter Jet" Military Aircraft. First, with small independent software development teams that adopted Agile Scrum and have served as Labs. Then as it spread to other teams it has evolved through continuous improvement.

In the new Gripen combat program, Agile practices are implemented at each level and in every discipline: Software, Hardware Design and Fuselage. The Saab Agile framework contains practices from Lean, Scrum, Kanban, XP and others.

Agile Scrum practices provide tools to manage Variability in real time. Transparency is inherent in the Agile Scrum Framework, which displays variations quickly and continuously.

At Saab, agile teams have a common cadence and stable pace.

All teams have three-week Sprints and start and end the same day.

Saab also discovered the need for synchronization beyond individual Sprints, and developed a method for iterations of quarterly cycles.

The increase is set to a Time boxed Quarter which leads to a steady pace of Sprint on all teams where increase targets are set each quarter.

These goals are set with a Top-Down and Bottom-Up approach where functional objectives are appropriately broken down into user stories and tasks as appropriate.

Saab Gripen building process

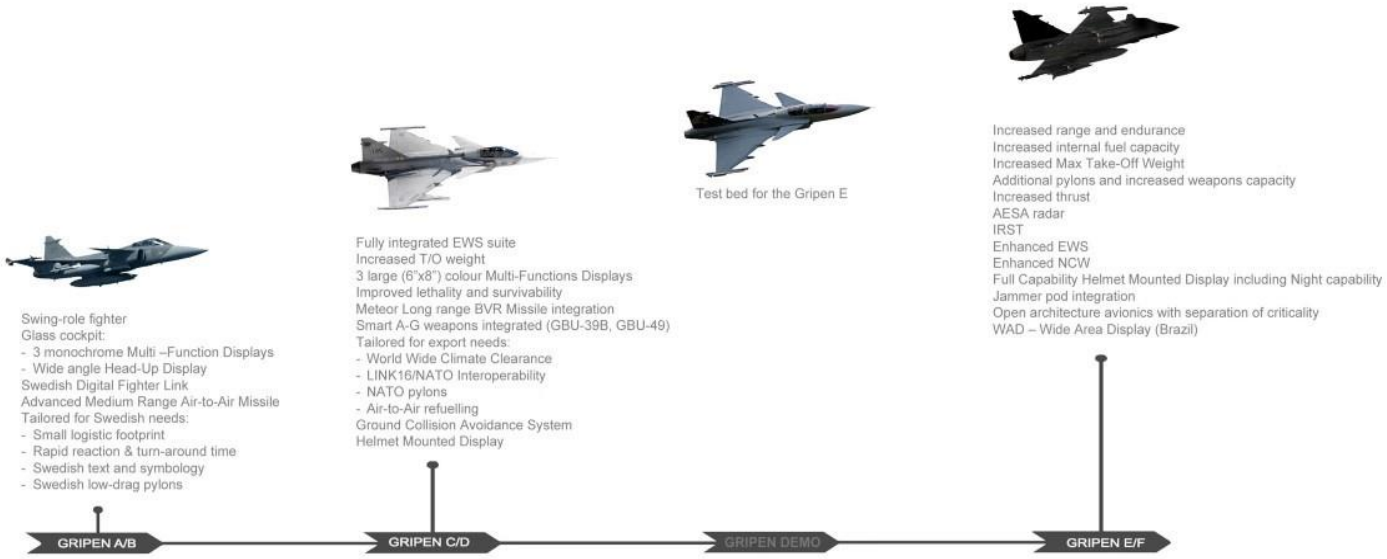

Swing-role fighter
Glass cockpit:
- 3 monochrome Multi –Function Displays
- Wide angle Head-Up Display
Swedish Digital Fighter Link
Advanced Medium Range Air-to-Air Missile
Tailored for Swedish needs:
- Small logistic footprint
- Rapid reaction & turn-around time
- Swedish text and symbology
- Swedish low-drag pylons
GRIPEN A/B
Fully integrated EWS suite
Increased T/O weight
3 large (6"x8") colour Multi-Functions Displays
Improved lethality and survivability
Meteor Long range BVR Missile integration
Smart A-G weapons integrated (GBU-39B, GBU-49)
Tailored for export needs:
- World Wide Climate Clearance
- LINK16/NATO Interoperability
- NATO pylons
- Air-to-Air refuelling
Ground Collision Avoidance System
Helmet Mounted Display
GRIPEN C/D
Test bed for the Gripen E
GRIPEN DEMO
Increased range and endurance
Increased internal fuel capacity
Increased Max Take-Off Weight
Additional pylons and increased weapons capacity
Increased thrust
AESA radar
IRST
Enhanced EWS
Enhanced NCW
Full Capability Helmet Mounted Display including Night capability
Jammer pod integration
Open architecture avionics with separation of criticality
WAD – Wide Area Display (Brazil)
GRIPEN E/F

Image 22: building process Figther Aircraf Saab JAS 39 Gripen: Saab https://www.scruminc.com/

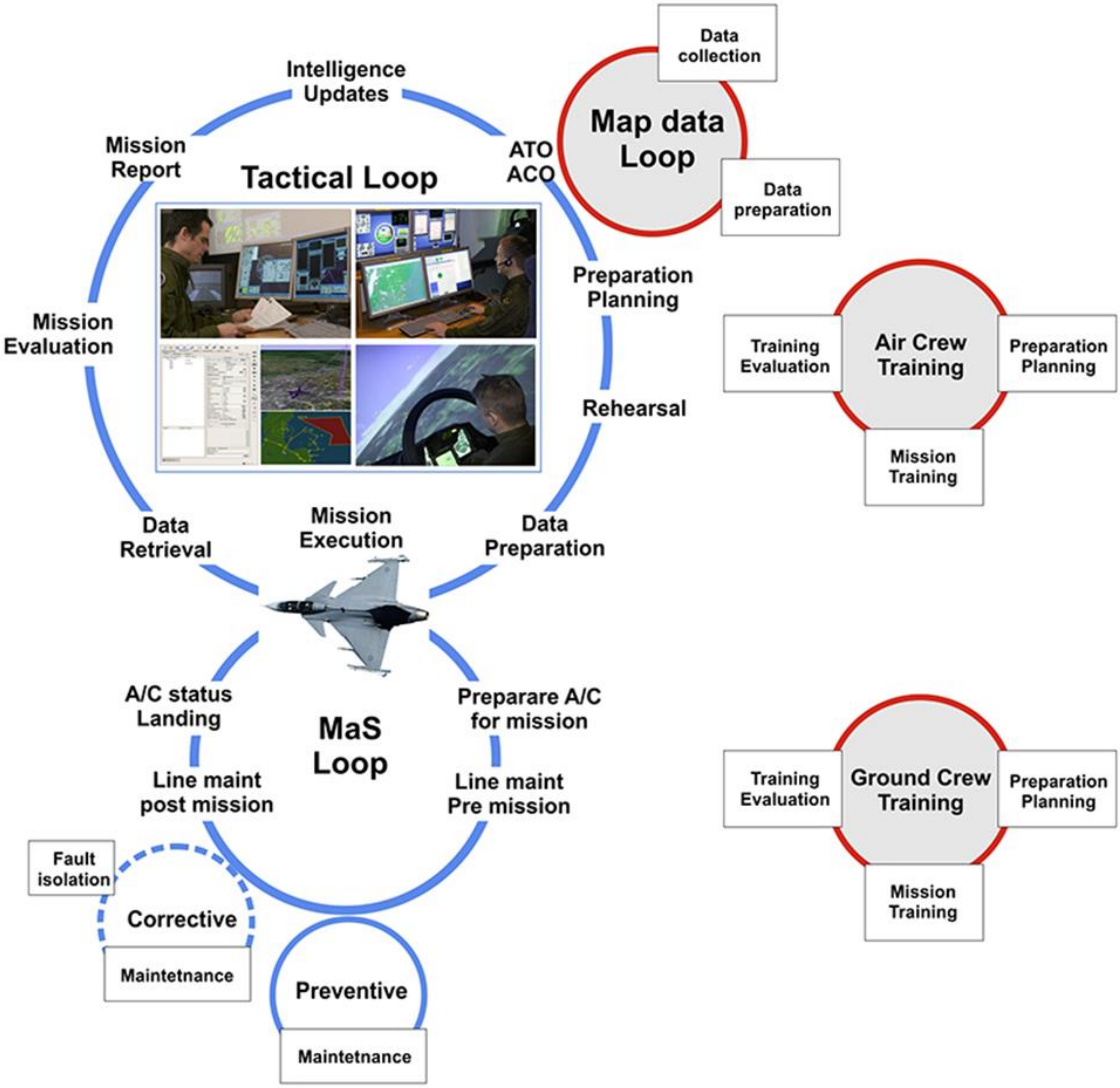


Image 23: building process Figther Aircraf Saab JAS 39 Gripen: Saab https://www.scruminc.com/

There is a structured meeting system for identifying team-level dependencies and make them visible throughout the project.

Each increase ends with a very well-defined delivery to validate and verify the

Product. In this way, if problems are detected at the integration stage, they are can make them visible early and take corrective action quickly.

Agile Scrum teams have three-week Sprints, start and end in the same day. SAAB has also developed a special model for sprint synchronization where the product is increased in quarterly cycles.

The Project Master Plan is iteratively broken down. At the top level you will find development. Below is one of the integration objectives which is well-defined, which allows to provide a larger launch of the product, is typically directed towards a specific test of a part of the aircraft.

The development step is synchronized into several main activities, such as installation fuselage, support system, etc.

Additionally, you define the goal that is important to achieve. A Step of Development is in turn divided into several increases in product deliveries which leads to a four-week Sprint pace to deliver the integrated sub-product.

Each increase ends with a very well-defined product delivery in order to validate and verify product quality.

Dependencies are quickly and more detailed identified which allows teams react quickly and take corrective action to solve them.

Agile practices at SAAB focus on empowering teams to improve continuously performance. Each engineer's daily goal is to perform the tasks of reducing the number of impediments and dependencies to a minimum Existing. This creates an environment of transparency for teams, maximizing their sense commitment to the delivered product.

Through Agile practices, SAAB can manage the variability and performance of the process with clarity and commitment. The result is a last-minute combat aircraft delivered generation with lower cost, higher speed, and optimal quality.

Bibliographic references

Books

Book: “Effective communication”, Harvard Business Review.

Author: Chris Argyris. Publication year: 1999

Book: “Why Less Communication is Better”

Author: Jeff Sutherland. Publication year: 2019

Book: “Group Psychology and the Analysis of the Ego,”

Author: Sigmund Freud, New publisher by Norton Library, 1990

Book: “The Art of Systems Thinking: Essential Skills for Creativity and Problem Solving”

Author: Joseph O'Connor, Publication year: 1997

Internet resources

Ebook: Microsoft SDL: Agile Development

Author: Nick Coblentz. Publication year: 2010

Source: https://owasp.org/www-pdf-archive/OWASP_AppSec_Research_2010_Microsoft_SDL_Agile_by_Coblentz.pdf

Reference websites and posts:

https://agileadoptionreport.com/wp-content/uploads/2020/08/CertiProf_Agile_Adoption_Report_2020_English.pdf

https://www.theguardian.com/technology/2018/apr/24/the-two-pizza-rule-and-the-secret-of-amazons-success

https://www.scruminc.com/scrum-of-scrums/

www.scrum.org/resources

www.scrumguides.org

https://www.scrum.org/resources/blog/user-story-or-stakeholder-story

www.scrum.org/resources/what-is-a-sprint-retrospective

https://www.scruminc.com/swarming-instantly-boost-scrum-teamproductivity

https://www.agilealliance.org/glossary/refactoring

https://refactoring.com/

www.scruminc.com

www.rand.org

https://agileadoptionreport.com/wp-content/uploads/2020/08/CertiProf_Agile_Adoption_Report_2020_English.pdf

https://www.theguardian.com/technology/2018/apr/24/the-two-pizza-rule-and-the-secret-of-amazons-success

www.scrumatscale.com

Images, pictures and graphics

Image 1, “Fish jumping out of fish tank”

Source: www.betagov.org plus my own design.

Image 2, The PDCA cycle.

Source: www.asq.org/quality-resources/pdca-cycle

Image 3, “The learning architecture”

Source: https://www2.deloitte.com/us/en/pages/finance/articles/cfo-insights-continuous-learning-environment.html

Image 4, Communication Channel Formula.

source: https://martinsitconsulting.com/wp-content/uploads/2019/09/Communication_channel_black_white.png

Image 5, Transparency, inspection and adaptation through empiricism.

Source: https://scrumorg-website-prod.s3.amazonaws.com/drupal/inline-images/2020-01/3pillarsofempiricism.png

Image 6, Scrum events: timeboxing and participants.

source: www.scrum.org

Image 7, Sprint planning cycle.

Source: https://www.scrum.org/resources/what-is-sprint-planning

Image 8, Daily Scrum meeting

Source: https://www.scrum.org/resources/blog/what-does-scrum-master-do-all-day

Image 9, example of team retrospective.

source: www.teamretro.com/retrospectives/wrap-retrospective

Image 10, example of team retrospective.

source: www.careerbuggy.com/agile-scrum-retrospective-techniques/sample-retrospective-daki-drop-add-keep-improve

Image 11, Example of Estimated time vs Actual time worked.

Source: https://timetag.it/tour

Image 12, One-Piece Continuous Flow (Toyota).

Source: www.institut-lean-france.fr

Image 13, Cost versus quality versus speed graphic.

source: https://www.innovation.co.uk/execution-risk/implementation-dilemma/

Image 14, workflow of code refactoring.

source: https://devopedia.org/code-refactoring

Image 15, Henrik Kniberg and Adres Ivarsson website.

Source: https://medium.com/scaled-agile-framework/exploring-key-elements-of-spotifys-agile-scaling-model-471d2a23d7ea

Image 16: Spotify model distribution.

Source: https://medium.com/scaled-agile-framework/exploring-key-elements-of-spotifys-agile-scaling-model-471d2a23d7ea

Image 17: CertiProf Agile adoption report 2020

Source CertiProf Agile adoption report 2020. pdf

Image 18: CertiProf Agile adoption report 2020

Source CertiProf Agile adoption report 2020. pdf

Image 19: CertiProf Agile adoption report 2020

Source CertiProf Agile adoption report 2020. pdf

Image 20: Microsoft Development Source SDL v4.1a

Source: https://www.microsoft.com/security/blog/2009/11/10/announcing-sdl-for-agile-development-methodologies/

Image 21, Amazon Business Canvas Model.

Source: https://www.pinterest.com/pin/20829217011453286/

Image 22: building process Figther Aircraf Saab JAS 39 Gripen.

Source: https://www.scruminc.com/

Image 23: building process Figther Aircraf Saab JAS 39 Gripen

Source: https://www.scruminc.com/

Picture 1, Kaoru Ishikawa.

Source: www.shmula.com

Picture 2, Taiichi Ohno

source: www.history-biography.com/taiichi-ohno

Picture 3, Frederick Winslow Taylor

source: www.qad.com

Thanks

Thanks to **Selinus University** and all Staff members, for giving me the opportunity to develop a serious and deep research work about the agile management of complex projects in large companies and the way to approach them. It is a very useful study for my professional career and to help other professionals and entrepreneurs who need updated research information to make accurate decisions in their companies.

Appendices: glossary of technical vocabulary

A

Agile Coach

An Agile Coach is the professional who takes care about that an organization progresses in its implementation of agile methodologies.

Agile methodology

The Agile methodology is used in software development and other projects or product. It focusses on the rapid implementation of an efficient and flexible team to plan the workflow. Agile provides the ability to make the best option in each situation without compromising the project.

B

Burndown Chart

A chart which shows the amount of work which is thought to remain in a backlog. Time is shown on the horizontal axis and work remaining on the vertical axis. As time progresses and items are drawn from the backlog and completed, a plot line showing work remaining may be expected to fall. The amount of work may be assessed in any of several ways such as user story points or task hours. Work remaining in Sprint Backlogs and Product Backlogs may be communicated by means of a burn down chart.

Burnup Chart

A chart which shows the amount of work which has been completed. Time is shown on the horizontal axis and work completed on the vertical axis. As time progresses and items are drawn from the backlog and completed, a plot line showing the work done may be expected to rise. The amount of work may be assessed in any of several ways such as user story points or task hours. The amount of work considered to be in scope may also be plotted as a line; the burn up can be expected to approach this line as work is completed.

C

Coherent/Coherence

The quality of the relationship between certain Product Backlog items which may make them worthy of consideration. See also: Sprint Goal.

D

Daily Scrum

Daily time boxed event of 15 minutes, or less, for the Development Team to re plan the next day of development work during a Sprint. Updates are reflected in the Sprint Backlog.

Definition of Done

A shared understanding of expectations that the Increment must live up to in order to be releasable into production. Managed by the Development Team.

Development Team

The role within a Scrum Team accountable for managing, organizing and doing all development work required to create a releasable Increment of product every Sprint.

E

Emergence

The process of the coming into existence or prominence of new facts or new knowledge of a fact, or knowledge of a fact becoming visible unexpectedly.

Empiricism

Process control type in which only the past is accepted as certain and in which decisions are based on observation, experience and experimentation. Empiricism has three pillars transparency, inspection and adaptation.

Engineering standards

A shared set of development and technology standards that a Development Team applies to create releasable Increments of software.

F

Forecast (of functionality)

The selection of items from the Product Backlog a Development Team deems feasible for implementation in a Sprint.

I

Increment

A piece of working software that adds to previously created Increments, where the sum of all Increments as a whole form a product.

P

Product Backlog

An ordered list of the work to be done in order to create, maintain and sustain a product. Managed by the Product Owner.

Product Backlog refinement

The activity in a Sprint through which the Product Owner and the Development Teams add granularity to the Product Backlog.

Product Owner

The role in Scrum accountable for maximizing the value of a product, primarily by incrementally managing and expressing business and functional expectations for a product to the Development Team(s).

R

Ready a shared understanding by the Product Owner and the Development a shared understanding by the Product Owner and the Development Team regarding the preferred level of description of Product Backlog items.

Refinement

Refinement: see Product Backlog Refinement.

S

Scrum

A framework to support teams in complex product development. Scrum a framework to support teams in complex product development. Scrum consists of Scrum Teams and their associated roles, events, artifacts, and rules, as defined in the Scrum Guide https://scrumguides.org/

Scrum Board

A physical board to visualize information for and by the Scrum a physical board to visualize information for and by the Scrum Team, often used to manage Sprint Backlog. Scrum boards are an optional implementation within Scrum to make information visible.

Scrum Master

The role within a Scrum Team accountable for guiding, coaching, the role within a Scrum Team accountable for guiding, coaching, teaching and assist a Scrum Team and its environments in a proper understanding and use of Scrum.

Scrum Team

A self-organizing team consisting of a Product Owner, organizing team consisting of a Product Owner, Development Team and Scrum Master.

Scrum Values

Scrum Values: a set of fundamental values and qualities underpinning the a set of fundamental values and qualities underpinning the Scrum framework; commitment, focus, openness, respect and courage.

Self-organization

The management principle that teams autonomously organize their work. Self-organization happens within boundaries and against organization happens within boundaries and against given goals. Teams choose how best to accomplish their work.

Sprint

Time boxed event of 30 days, or less, that serves as a container for the other Scrum events and activities. Sprints are done consecutively, without intermediate gaps.

Sprint Backlog

An overview of the development work to realize a Sprint’s goal, typically, a forecast of functionality and the work needed to deliver that functionality. Managed by the Development Team.

Sprint Goal

A short expression of the purpose of a Sprint, often a business problem that is addressed. Functionality might be adjusted during the Sprint in order to achieve the Sprint Goal.

Sprint Planning

Sprint Planning: Time--boxed event of 8 hours, or less, to start a Sprint. It boxed event of 8 hours, or less, to start a Sprint.

Sprint Retrospective:

Time boxed event of 3 hours, or less, to end a Sprint. It serves for the Scrum Team to inspect the past Sprint and plan for improvements to be enacted during the next Sprint.

Sprint Review

Time boxed event of 4 hours, or less, to conclude the development work of a Sprint. It serves for the Scrum Team and the stakeholders to inspect the Increment of product resulting from the Sprint, assess the impact of the work performed on overall progress and update the Product backlog in order to maximize the value of the next period.

Stakeholder

A person external to the Scrum Team with a specific interest in and knowledge of a product that is required for incremental discovery. Represented by the Product Owner and actively engaged with the Scrum Team at Sprint Review.

V

Values

When the values of Commitment, Courage, Focus, Openness and Respect are embodied and lived by the Scrum Team, the Scrum pillars of transparency, inspection, and adaptation come to life and builds trust for everyone. The Scrum Team members learn and explore those values as they work with the Scrum events, roles and artifacts.

Velocity

An optional, but often used, indication of the average amount of Product Backlog turned into an Increment of product during a Sprint by a Scrum team, tracked by the Development Team for use within the Scrum Team.