Journal Homepage: - www.journalijar.com
INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)
Article DOI: 10.21474/IJAR01/4768
DOI URL: http://dx.doi.org/10.21474/IJAR01/4768
ISSN NO. 2320-5407

RESEARCH ARTICLE

COMPARISON OF DIFFERENT AGILE METHODOLOGIES AND FIT ASSESSMENT IN AN INDUSTRIAL CONTEXT.

Andrea Margini, Gaetano Cutrona and Cesare Fantuzzi.
University of Modena and Reggio Emilia, DISMI, Italy.

Manuscript Info

Manuscript History

Received: 7 May 2017
Final Accepted: 9 June 2017
Published: July 2017

Key words:-
Agile methodologies; Agile applicability; Methodology fit: Best Agile methodology: Product Development

Abstract

The paper collects together and describes the main and most known Agile methodologies. Those methodologies are then compared in order to assess their fit in the context of highly innovative non-software companies. In particular, the focus is on a full system supplier for food processing and packaging equipment, packaging material and services. The fit between the Agile methodologies and the company is performed considering the innovation and development processes in place into the Research and Development department of the company itself. By doing so the paper aims at contributing to the theme of Agile application in non-software industries. Moreover, it will support in the choice of which method or framework suits better having clear the needs and characteristics of that specific context. Finally, the paper provides also a suggestion about how to approach this kind of choices.

Introduction:-

Agile has become one of the most used word in companies all over the world. Though everyone interprets it with his own meaning, the basic need at the bottom is the same: to keep competitiveness by being faster than today in dealing with changes during product development activities, to better satisfy customer and end-user needs and to deliver quality products.

The whole journey started in 2001 when a group of practitioners defined the Agile Manifesto, clearly stating the core values and main principles that must be followed in order to achieve agility. From that milestone several methodologies were created and used in industrial contexts in order to support companies in becoming more agile, responsive and able to satisfy customers in an efficient way with their product development.

Though they share the Agile Manifesto [1] as corner stone, those methodologies differ a lot each other. Those differences are mainly due to their fundamental and underlying aims; one may aim at the maximization of the value delivered in a fixed time unit through product increment, another one to the maximization of the ROI or even the risk management [8], [17], [21], [19], [4], [20]. Obviously, given the different focus, those methodologies are very different in the suggested team routines and practices, in the foreseen roles within the development teams and in the way activities are organized. On the other side of the coin, what they all agree is about the concept of continuous delivery of valuable product increments. They also agree in relating the value to the prioritized requirements and their fulfilment assessment performed by involving end-users and customers [17], [13], [16]. In many cases this is translated with the concept of having the customer as part of the development team.

Corresponding Author:- Andrea Margini.
Address:- University of Modena and Reggio Emilia, DISMI, Italy.

However, the difference level brings the question about which one to choose. What is the best Agile methodology? Which one is the most agile? Which one is easier and more successful to implement and apply? Researchers agree that there is no best methodology, no one size fits all. The suitability of an Agile methodology rather than another one depends on the specific context and case. Moreover, it is also often suggested to tailor methodologies case by case, hybridizing standard ones and crossing their elements [17], [21], [19], [4].

This paper aims at contributing in the choice by bringing together studies about the most known and cited Agile methodologies and by comparing them. Moreover, it shows how the choice can be supported by considering the specific context and its characteristics. Finally, this paper provides some hints about what could be the most suitable Agile methodology for each specific application into the specific context where the paper is developed, i.e. the context of a food processing and packaging company.

Another important element to take into consideration is that Agile and related methodologies were born from and for the software industry. However, considering the claimed benefits, other industries have been considering Agile and at how to apply those principles and methodologies in their product development environment. Thus, given that the context described in this paper is strongly tied to physical objects and that some suggestions and considerations are provided about the various Agile methodologies suitability into the specific environment, this paper contributes to the theme of the Agile application into non-software industries.

In the following section studies about Agile methodologies and Agile in general are compared, providing then some good reasons to shift from a classical development approach, as the waterfall, to an Agile one. Then the most famous and applied agile methodologies are described and compared based on different sets of criteria. First comparisons were developed by other researchers and have the general purposes to summarize the methodologies' characteristics and to point-out what are the elements favorable or not to their implementation. A final comparison is then provided. This last one was developed considering the previously provided knowledge. The set of criteria, or dimensions, is based on the previous descriptions and characteristics, as well as the scores per each Agile methodology. This comparison is meant to be read as which are the favorable, not favorable or neutral elements for each specific Agile methodology implementation. However, it differs from the previous comparisons because of the scoring dimensions. The set is wider and reflecting the dimensions used to describe the specific context in which the paper is developed.

After the comparison, the context is introduced, indeed, together with its Research and Development process setting and characterization. The R&D department is characterized by several development processes, each one with a different aim and application. Those processes are described and compared based on dimensions that are directly mirroring those used to compare the various Agile methodologies. The comparison is meant to be read as how characterizing are those dimension is for each process.

The two comparisons are then brought together and summarized in a chart displaying what is the suitability score per each Agile methodology per each process. Those scores are then discussed, as well as improvement areas for this study.

Agile related works:-

The Manifesto for the Agile Software Development [1] was the beginning of the movement that is now changing the way innovation and product development are managed. The manifesto provides a list of values to which practitioners must adhere and clearly stating what are the priorities and most important aspects. The Agile Manifesto [1], thus, promotes individuals and their interactions, autonomy, diversity, close collaboration, customer involvement, continuous learning and change responsiveness over other aspects.

Since that time many practitioners and researchers have dealt with the Agile principles and several methodologies and framework were developed in accordance to those principles. Given that, there are general characteristics common to all the methodologies defining themselves as Agile and expressed by teams.

According to Lee and Xia [12], Agile teams are those characterized by the ability to include changing user requirements into the project outcomes and the time needed to include a particular changed requirement into their work. These two are called response extensiveness and response efficiency [12]. According to the authors these factors are the essential ingredients to achieve Agility. However, according to the Agile Manifesto [1], there are far

more factors, such as team diversity and autonomy, affecting the Agility that are not deemed so important by Lee and Xia [12].

Despite this, there are other researchers agreeing with the Manifesto about such aspects. Muller [16], for instance, states that the most important characteristics for an Agile team are self-empowerment and autonomy. These two will then positively impact on the quality of the work environment and on the overall performance. However, to reach that point, it is crucial to have a change in the role and mindset of project managers. Otherwise there will be no Agility achieved and no benefits [16].

Considering the focus given to the individuals by the Agile Manifesto [1], Lindgren and McAllister [13] agree with Muller [16] on the importance of changing mindset, but not restricting the concept to project managers only but to the whole organization. Then, considering also the importance of the iterations between people, they also identify four macro-categories to classify them. By doing so they highlight their peculiarities to consider and manage when inserted into an Agile context in order to have effective teams [13].

Another important aspect related to Agile is that the Manifesto was created for software development and so the methodologies and framework. As a consequence there are many examples of Agile implementation in software houses proving the validity of such methodologies. The benefits are so clear that Agile is appetizing also other types of industries, such as the manufacturing one. However, due to the context's characteristics the implementation is not straightforward as it was for the software industry. Moreover this particular theme is also young to the research and to practitioners.

On the other hand, Conforto et al. [5] provide some interesting insights: some typical Agile practices and ceremonies are already in place in companies considered not Agile at all. The outcome suggests that the adaptation and implementation of Agile methodologies is possible and easier than expected in those contexts as well. The same research highlights also some best practices fully adherent to Agile principles that seems to be fundamental to achieve agility: Co-location of team members, team dedication to one project at a time and customer effective involvement are the main ones [5].

Still on the topic of Agile methodologies in non-software industries, Margini et al. [14] showed that it is possible to use Agile methods in some parts of a development project only. In that case Scrum was adapted and piloted during the internal verification activities in a service-product development. The idea is to reduce the whole time required for verification and correction of problems by having concurrent testing and development and correction sprints. It results in two different versions of the same product, in the end, that the Systems Engineer (or Product Architect) integrates [14]. That adaptation was piloted to override the criticalities of verification and testing activities faced by development projects highlighted by Ghilic-Micu et al. [9] and Ivan et al. [10] studies, such as the overall length of activities, the number of effectively executable tests and the experience of testers and human factor in general.

In the end all these experiences build towards the possibility of achieving agility even in non-software companies. This is also where this paper aims at contributing. In the following section, Agile and Waterfall models are compared to show good reasons for companies to shift from a classical product development approach to an Agile one. After that, the most cited, known and debated Agile methodologies are described. Then, the context in which this paper is developed is introduced, with a specific focus on the different processes in place into the Research and Development department of a food processing and packaging company. Afterwards the already cited Agile methodologies are compared each other based on different criteria and then matched with the processes, in order to offer information about what could be the most fitting methods with the current organizational set-up.

Agile and Waterfall Models comparison:-

The Waterfall model is the classic sequential one where one activity follow the previous one without any possibility of moving backward. According to an article on the site Base36 [2], the advantages given by this model are related to the enormous importance that documentation has, such as activity plans, budgets and record keeping. This makes projects working with a waterfall set-up less impacted by people turnover [2]. However, the working approach is more than rigid when it comes to facing changes, especially in case of errors in the early activities or customers evolving needs.

On the other side of the continuum there are Agile models, where the work advances in small incremental tranches with requirement analysis, design and verification activities occurs on a regular basis. This means that projects self-adapt to the changes. However the final product can be very different to what was thought to be in the beginning [2]. As usual, there is no one size fits all solution. The benefits from the adoption of one model rather than another one are strongly related to the context. Waterfall models are best suited when the final customer don't have any possibility to change the project scope and where the clear definition of everything is more crucial than being fast. When the scope is not stable, the context is rapidly changing and faster deliveries are key, then Agile models can support activities [2].

Common Agile methodologies:-

In this section the main and most famous Agile methodologies are described. They span from Extreme Programming, that is very specific for software, to DSDM or Scrum, that are project management practices, with others like Lean, AUP and FDD sitting in between.

Extreme Programming (XP):-

According to Coffin and Lane [4] Extreme Programming (XP) is a programmer centric methodology emphasizing technical practices. This is to promote development through frequent delivery of software [17], [20], [4]. Very shot cycle time (1-4 weeks) and few artefacts (code, story cards and unit test) characterizes this set of practices. XP's core values are communication, simplicity, feedback and courage [17], [4]. From the core values the practices constituting XP are derived, such as the planning game, the organization in small releases, the use of metaphor, etc. All these practices support and reinforce each other in a positive circle [4]. According to Extreme Programming practices, the release planning is the first task a team has to accomplish. This is in conjunction and cooperation of the customer. A high level scope and plan of the project is thus defined, together with high level requirements. Then the team, focusing on requirements, prepares the development planning and establishes development cycles [17], [4]. Within XP there are few defined roles: the customer, the programmer, the coach (who teaches on techniques) and the tracker, monitoring the deliveries and the compliance with the plan ([4].

Scrum:-

Scrum is a project management framework to manage projects. Thus, it integrates well with other practices. The aim of Scrum is to deliver a valuable product increment (software in its origins) after each iteration. What provides the highest business value is established by the customer by prioritizing the requirements on a cyclic basis [17], [20], [4]. Development cycles are named Sprints, with a typical fixed duration from 2 to 6 weeks, depending on the team. During the sprint planning, based on the prioritised requirements backlog, the team commits to the satisfaction of a set of them [17]. At the end of the sprint a review and a retrospective occurs in order to assess the requirements satisfaction and to capture learnings. Together with the sprint planning, the sprint review and sprint retrospective, the daily scrum is the other one important fixed meeting point, where team members debrief their previous day work, the current day work and possible obstacles [17]. An important principle in Scrum is that teams must be self-organized. Therefore, they don't follow a prescribed detailed plan but they develop and update it. Another important principle is the customers' involvement not only in the requirements backlog prioritization but also in the periodic review [11], [4]. This cyclic involvement allows also to cope with the problem of unclear projects' scope at the beginning and to face changes in the requirements set [11]. According to the Scrum principles, there are three main defined roles in a team: the product owner, the scrum master and the team member. The product owner oversees the requirements backlog and its prioritization from the customers. The Scrum master supports the team in the scrum routines and removes obstacles from the team members work. Team members are those in charge of delivering, cycle by cycle, the prioritized value. The recommended dimension for a team is roughly around 9-10 people [11], [4].

According to some practitioners, there are some conditions making Scrum ineffective even with a team mature in its application. In particular, one of them could be the novelty of a given technology [21]. As less know it is, as it becomes harder for team member to estimate efforts and establishing proper acceptance criteria for their backlog. The final consequence is a loss in team results and in people morale. Moreover, not accomplishing the plan or failing the acceptance has a negative impact when it comes to the customers and their reviews [21]. The uncertainty related to the unfamiliarity with the technology, in the end, leads to inadequate planning, making the sprint plan subject to changes on a too frequent basis [21]. However, on the other hand, it is important to remember that the learning possibilities are one of the strong points of Agile methods, making those findings worth a deeper analysis. Finally, talking about Scrum, different advantages and disadvantages were highlighted by practitioners and

researchers. Among the pros, there are the transparency and visibility on project activities due to the board, team members are more empowered and it is easier to face and embrace changes due to the detailed planning limited to the Sprint [18]. On the other side, the project scope can creep due to too many changes, team members must be skilled in order to be able to define and estimate proper tasks. Moreover the scrum master role is different from the typical project manager one, making it very delicate and critical for the success of teams [18].

Lean:-

Lean aims at delivering complex software systems that are exactly what the business needs, based on seven principles that are concerning the waste elimination, quality, knowledge creation, commitment, fast delivery, respect people and system optimization [4]. Lean shares with Scrum its being a project management framework rather than a collection of practicalities, such as XP. The business value is assessed having in consideration costs and ROI of a project, that are typical measures accepted and understood in all the management forums [4].
In Lean methods, great attention is given to the requirements gathering and definition. Like in Systems Engineering methodologies, they must be clear, complete and verifiable. Furthermore, they are measured and prioritized based on business impact. Business impact is defined by the customers, that are involved in the requirements definition [4]. Lean addresses also the matter of having the right skill set in the team. According to the principles, team members should be cross-trained, both on the technical and on the business side, so to better understand the point of view of the customer. Evaluations and choices are undertaken in terms of ROI and costs. However, this results also in less definition between roles than in other methodologies [4].

Feature Driven Development:-

As it is possible to understand from the name, in FDD the development of a product is driven by the incremental development of its features. According to the definition, features are small client functions valued by the final customer [17], [4].In FDD methodology a model representing the problem/product to be developed is built first. Based on the knowledge the team has about the model from various domains, requirements are gathered. This is done in a top down approach. With the same approach, requirements are divided into feature sets [17], [4]. With the feature sets as input, the team plans and then executes the development activities. This approach has some common points with the Systems Engineering methodologies, like the problem model that is similar to the system architecture and the requirements grouped in feature sets, like if they are functional requirements. On the other hand there are differences in the sequence of activities. In FDD the model is built before having the requirements, while in Systems Engineering the System Architecture is defined after a certain maturity level of requirements is reached [3]. Feature Driven Development relies on clear roles: project manager, chief architect, development manager, chief programmer, class owner, domain expert, tester, deployer, technical writer [17], [4].

Crystal:-

Crystal is a collection of methodologies, each of them has a specific use depending on project size and criticality. Project size is measured with the number of team members into the project, while the criticality is an estimate made considering the potential for the system to cause damage [4].All these methodologies share the same principles: frequent delivery, continual feedback from the customer with constant communication, safety in terms of product safety for the end users and working place safety and serenity for team members, focus on the top priorities, access to users and automated test and integration [4]. If Crystal's principles are compared to the Agile one, there is good overlap between the two sets [1].

Dynamic System Development Model (DSDM):-

Dynamic System Development Model methodology was developed by people working in the business side of companies rather than by technicians. It is also considered one of the heaviest agile methodology due the roles setting, and it is claimed to be in place before the Agile Manifesto was written even though less known than other methodologies [20], [4]. According to the DSDM Consortium [8], the methodology is based on eight principles, that are fundamentally shared by all the Agile methodologies and practices. Focus on business needs, on time delivery, collaboration, incrementally building of the solution and iterative development are some of them, sided by a strong quality focus, clear and continuous communication and control demonstration by using plans and reports of the appropriate detail level. This last principle about the control demonstration is one of the main differing point between the Agile principles and DSDM ones [1]. DSDM methodology consists of three main phases: pre-project phase, project lifetime and post project. In the first phase all the pre-project activities are performed, like team set-up and requirements categorization according to the MOSCOW practice [17]. During the second phase, operational and project-typical activities are performed, such as feasibility study, design and implementation. In particular, after a

first feasibility and business study a first prototype of the solution must be ready [17]. Then, functional modelling, design, built and implementation activities start and are iterated several times [8], [17], [4]. During the post-project phase, activities such as the project close-out are conducted. Projects are split into chunks with features and a time and budget estimated. Per each chunk, time and budget are mandatory fixed. This means that the only possible variable in the development activities are the features. As a consequence, everybody since the beginning of the project knows that not all the features will be developed [8], [4]. So, considering prioritized requirements, they are dropped from lowest priority on when time and budget starts running out. This also underlines the importance of the user involvement, especially in early prototypes feedback when challenging requirements [4]. According to the DSDM principles, the active user involvement is imperative, together with frequent delivery of functionalities meeting the business needs. In order to do so, integrated testing practices are a must to be able to be quick in developing quality products in shorter time [8], [4]. One peculiarity of DSDM methodology is that it is recommended for teams up to six members, and for systems that are not critical from a safety point of view [4].

Agile Unified Process (AUP):-

Agile Unified Process is an iterative and incremental development process framework. Within the framework activities and artefacts involved are specified. In AUP great importance is given to the risk identification and risk management. According to its principles, the elements with the highest risk should be prioritized and faced earlier [4].

Another important aspect in AUP framework is the product architecture. In order to have an effective incremental development, the architecture must be stable, like a backbone to fill [4]. Every increment in AUP process adds something to the backbone architecture. Increments are developed during iterations, that are divided in four main phases: inception, elaboration, construction and transition. During inception phase the team develops a shared understanding of the scope of the new increment. During the second phase, elaboration, that understanding is expanded into requirements and into a concept solution. Then, in construction phase, the solution is physically developed and finally tested and integrated into the whole product in transition phase [4]. Such approach can be very effective in cases like product extensions, where the overall product architecture is stable and the scope change level not that high. In the cases of complete new product developments, AUP can show some limits due to uncertainty, if it affects the overall architecture [6].

The Context:-

In this section the industrial context and the company where the paper is developed are introduced. Then the company's Research & Development department with its main development processes is described as well. The company this paper refers to is an international company, market leader into the food processing and packaging machines markets and strongly project oriented. Projects are initiated in order to develop new products for the product portfolio and to improve the existing ones. Furthermore there are projects aiming at improving the whole company way of working and way of developing new products.In order to have all the sites worldwide work in the same structured way, the overall culture is process based. This means that there are sets of rules, guidelines and templates scaled and used to guide the teams' work depending on the project scope. To keep the competitive edge the overall processes evolve and change depending on the company needs.

Before proceeding, it is important to bear in mind that the processes here mentioned are not the only ones in the company. There are processes related to the supply chain, to the finance and to the human resources functions, for instance, in place in other departments. However only the processes in place in the Research and Development department are considered into this paper: Agile methodologies were born to improve product development and similar processes thus, company wise, they are those mainly affecting its future competitiveness.

The processes:-

The following picture shows the high level processes composing the Research and Development department of the company.

The Technology Development process is used when projects aim at scouting, developing and delivering new knowledge and know-how to the company. The know-how is then fed into other receiving projects that, following the other processes, aim at exploiting it in finalized products. It is an explorative process, not too formalized, allowing a certain level of freedom to team members in conducting activities. From one hand, time to market is not

very crucial for this kind of project, whilst the outcome itself could be of the utmost importance for future competitiveness.

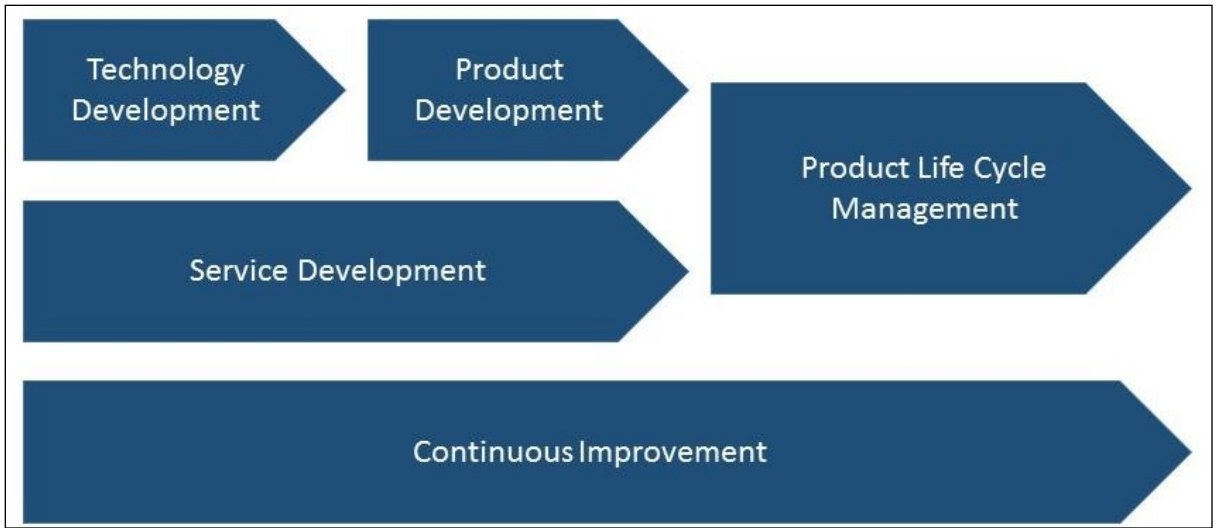

Figure 1:- The R&D processes into the company

The Product Development process is the main receiver of new know-how developed during Technology Development projects. It supports projects aiming at developing new products based on stakeholder needs and having as reference, when possible, existing product families. Projects' outcomes have a direct business impact, and timing in releasing products is critical. This makes the process very structured and formal. Moreover there is a considerable amount of stakeholders involved into projects, mainly through requirements and, in many cases, end-users are involved as well. That may happen in form of requirements or in form of end-user testing. The Service Development process is used to develop service-products, commonly known as services. They end-up in the product portfolio as physical products, and they can be integrated with the existing equipment or can be sold separately. It is a less formalized and structured process than the Product Development one because of its lesser criticality, making it more similar to the Technology Development. That is also due to the fact that services are not core products. Moreover, Service Development process is the latest introduced and thus the less mature one. Considering also the type of outcome, services are potentially less critical than physical products. Nonetheless the set of stakeholders of Service Development projects is bigger than those having a voice into Technology Development, but fewer than Product Development projects. Product Life Cycle Management is a process used for projects solving issues and claims from the market and to extend the current products lifetime with feature additions or performance improvements. Similar to the Product Development process, it is driven by stakeholder needs, time is a relevant dimension to consider and important technical decision are made. However, since the product is already developed and known, the process requires less guidance and structure than Product Development, making it less formalized and easier to scale. On the other hand, the scope of projects using this process is limited compared to those in Technology Development or Product Development, allowing thus team to face smaller level of scope change and allowing also project teams of a smaller dimension. Continuous Improvement process is the process used to define and implement new ways of working. The outcome of projects using this process is not a technology nor a product nor an improvement to existing products. What is delivered are improvements to the previously mentioned processes. Given this fact, it is a process that is different from the previous ones also in terms of required competences. It is possible to state that the Continuous Improvement process is not an engineering process, while others are. On the other hand, considering that the outcome of Continuous Improvement projects directly affect the way that know-how and solutions are invented and industrialized, its potential impacts are as high as the projects' scope is wide. This make this process strictly ruled by governance, but less described in terms of activities. However, the vast majority of the projects following this project has a very limited impact, making then the outcome far less critical than the potential, and allowing small project teams.

Key Process elements:-

The previously introduced processes have similar structure and underlying principles. They are requirements driven, where requirements represent stakeholder needs. Then design approach is based on the Systems Engineering

principles and the V-model [3]. Those needs are translated into technical language, representing what the solution has to achieve, and then used to drive the solution design. Then requirements are cascaded to sub-systems, i.e. sub-partitions of the solution architecture [3]. This requirements translation and design cycle is repeated until the desired detail level, e.g. sub-system, module, component, part, etc. [3]. Then, from the most detailed level up to the whole solution, integration and verification activities are undertaken, until the final validation with stakeholders [3]. There are some differences in terms of what is a sub-system (or a module, part, component) and thus how design and verification activities are undertaken in the case of Service Development projects, but the overall approach is still the same [15].

All the engineering processes are gate based, meaning that they are divided into phases. Every phase has a determined goal and a set of criteria to assess and evaluate the project advancement. At the end of each phase there are gate reviews where the projects are evaluated against those criteria and decision about their continuation are taken. Depending on the process and the project size, the governance at the gate reviews changes [7]. The various processes and their phases are described through flowcharts representing the workflow and which activities have to be completed per each phase [7]. Moreover, the workflow clearly shows the competence areas required into each process, highlighting so their contribution and responsibilities. Those competence areas are called swimlanes and they represent the minimum competence set required to accomplish projects. Spread in the workflow there are Design Review Meetings used to assess the technical maturity of the solution prior to the gate review [7].

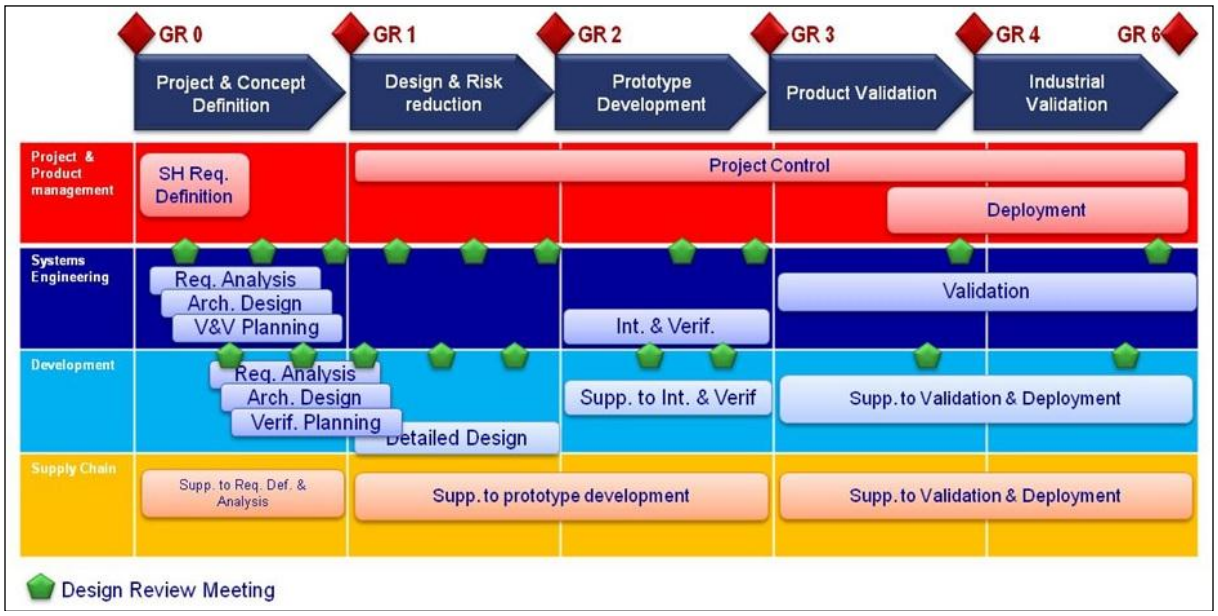

Figure 2:- Product Development process structure [7].

The previous picture represents the structure of the Product Development process. The other engineering process are very similar except for the phase number and scope. The Continuous Improvement process, as well as the other processes, is stage-gate and phase organized. However, the governance body may heavily vary depending on the scope and target of every initiative. Not being an engineering process, it does not rely on the Systems Engineering principles. On the other side, since it aims at improving the way people work, requirements are an important input as in the other processes. Following the Continuous Improvement process, there could be whole projects aiming at substituting or improving the tools used in the other processes, or at developing and spreading new methodologies. On the other hand there are mainly in place minor activities impacting on smaller sections of processes following a scaled version of the Continuous Improvement process. Given the variability characterizing the Continuous Improvement initiatives, only the various phase objective and pass criteria are established, together with scaling criteria. Thus, there are no flowcharts with activities nor competence areas identified. Finally, the governance is established based on the initiative scope. All these characteristics make the Continuous Improvement process less structured than the others, but also much more dependent on the team competence in terms of activity planning and organization.

Process Characterization:-
All the previously described process characteristics are summed in the following picture. According to different dimensions embodying what is stated, the impact of each dimension (or factor) is scored per each process in a High (3), Medium (2) and Low (1) scale. Those dimensions are also used, in a later section, to describe and compare the Agile methodologies.

	TD	PD	SD	PLC	CI
VOLATILITY OF REQUIREMENTS / HIGH SCOPE CHANGE LEVEL	High	Medium	Medium	Low	Medium
FORMALIZATION LEVEL IN PROCESS STRUCTURE AND GOVERNANCE	Medium	High	Low	Low	Medium
FORMALIZATION AND RELEVANCE OF DOCUMENTATION	Medium	High	Medium	Medium	Medium
TIME TO MARKET CRITICALITY	Low	High	Medium	Medium	Low
SOLUTION CRITICALITY	High	High	Low	High	Low
END-USER INVOLVEMENT	Low	Medium	Medium	Medium	Medium
NUMBER OF STAKEHOLDERS	Low	High	Medium	Medium	Medium
LARGE TEAMS	Medium	High	Low	Low	Low

Figure 3:- Characterization of the various processes

Volatility of requirements / high scope change level: this dimension is about how often requirements or scope are changed during a project life. Due to the scarce knowledge at the beginning of Technology Development process, often they are deliberately vague. In the case of Product Development and Service Development they are more stable already in the beginning, though some refinement usually follows. In PLC project requirements and scope are usually very clear because of the project tendentially limited impact in the product portfolio. Continuous Improvements projects usually suffer of a certain level of uncertainty mitigated by initial analysis phases.

Formalization level in process structure and governance: this dimension is about how formalized and strict the process workflow and the governance bodies are. The governance bodies are involved into this dimension because they are those deciding also on how projects could scale the various processes. TD process is not too formalized because it must allow freedom to explore and evaluate concepts. Product Development projects suffer the highest level of formalization in terms of activities and governance: it is harder to scale compared to other processes and activities and tasks are described and established. Service Development and PLC processes allow a lot of freedom to projects in terms of creating their own activity work-streams, and the activities as such leave some freedom in terms of detailing what to do. Continuous Improvement process stands in between PD and PLC. For its own characteristic activities are not described or formalized, but scaling decision and the approach in general should be discussed with governance members.

Formalization and relevance of documentation: this dimension is about the documentation level that is required at each milestone in order to proceed with the process. Though the freedom level characterizing the Technology Development process, outcomes and findings must be reported to build-up knowledge. Again, Product Development process results being the most complex one in terms of documentation level. That is both to ensure the solution is addressing initial requirements, to describe the solution, and to secure the business case relevance and risk minimization. Then there is the full set of hygienic and safety related declarations and documentations that are in charge of the projects' members. Service Development and PLC put less emphasis on the documentation to be provided. That is mainly due to their smaller scope and minor criticality in terms of the final outcome. In the case of PLC projects already in place documentation has to be extended while in the case of Service Development, the set of

documents to be provided is smaller than in Product Development. That is mainly because they do not impact in parts in food contact. Continuous Improvement process requires a similar amount of documentation as well. That is mainly to describe the approach and the outcomes of the activities.

Time to market criticality: this dimension is about how important is to be fast in delivering the final product. When it comes to TD projects, the knowledge and technology outcome of the projects are very important, but Technology Development projects should be considered as close as possible to the pure research: it may take years to have something as output. Similarly, the Continuous Improvement does not suffer from market pressure because it aims at internal customers' needs. Moreover, its outcomes usually cope with mindset and change issues. This makes the time dimension less relevant than the people one. Service Development and PLC projects face a higher time pressure. Even though they do not deliver new products, they extend the life-cycle or add features of existing ones, or they add side services to the portfolio. It means that there is the need of them from the market. Finally, Product Development projects are those suffering the highest time pressure. That is because new products should be in the market as soon as possible in order to anticipate competitors.

Solution criticality: this dimension is about how critical and complex are the solutions developed with those processes. Criticality implies a security and aseptic dimension because of the company core business. In this perspective, Technology Development, Product Development and PLC processes deals with physical equipment or parts of equipment that are going to interface with food. Moreover, they mainly focus on systems with many interfaces within themselves and characterized by a high engineering and technology level. Thus, those processes are very important from a criticality point of view. On the opposite side, there are Service Development and Continuous Improvement processes that usually do not have any impact on the food itself and are characterized by a smaller engineering level.

End-user involvement: this dimension is about how much the end-users of the developed solution are involved into the solution development as such. In case of Technology Development projects, end-user might be involved close to the end of the process, when the solution is tested. Otherwise, having such a wide and volatile scope during a given period means that is hard to take into consideration the needs and opinions of a group that may change during the project itself. In the case of all the other processes, end-users are involved in validation and user acceptance tests towards the process end. Their needs are also captured as part of the requirements definition activities. That is why this dimension is set as "Medium" for Product Development, Service Development, PLC and Continuous Improvement processes.

Number of Stakeholders: this dimension is about the number of different stakeholder impacted and interested in the projects. It is a proxy of the complexity of the stakeholder management efforts required by team members, especially by project managers. Technology Development projects have to face a small amount of stakeholders, both because of their uncertainty and because those few are mainly internal groups that will receive and own the outcome. Product Development projects, on the other hand, are affected by a high amount of stakeholders. That is because they must take into consideration also legal and governmental aspects, end-users, customers, retailers and consumers' voices. Furthermore, also internal organizations are impacted and bring their voices to the projects. Service Development projects usually face a smaller quantity of stakeholders. From one side because of the smaller outcome criticality, and from another side because of the novelty of the type of outcome in the product portfolio. PLC projects suffer of a smaller quantity of stakeholders than PD because they mainly deal with minor developments, targeting very specific market needs trying to minimize the impact on the internal interested organizations. Continuous Improvement projects face a higher amount of stakeholder compared to TD projects. They only consider internal stakeholders, due to their scopes, but they affect the whole company. This means that several organizations are impacted and willing to provide their needs and inputs.

Large teams: this dimension is about how big teams are when dealing with those processes. Technology Development projects usually have a medium size project team composed by a project manager, a systems engineer, some experts about the technology under development and, eventually, supply chain representatives. Product Development projects usually are driven by a team composed by project manager, systems engineers and representatives of equipment development, packaging material development and supply chain. Then, typically, each of them coordinates an external team that is referring to the main one, making the whole approach much more complex than in other processes. In the case of Service Development and PLC teams are typically small, due to the lowest effort and competences required for the specific project. Finally, when it comes to the Continuous

Improvement process, projects in this category are characterized by small teams driving activities and discussions, that are then undertaken in specific networks or forums. However, the people in those networks are there for their role, and not because they were appointed as project contributor in a project charter. This makes them not-part of the project team as such.

Agile methodologies comparison:-

Agile methodologies were born for the software industry and, given their success and delivered benefits, they are also spreading to non-software industries [14], [5]. Even considering this fact, all the studies and researches agree on the Agile methodologies applicability. There is no one size fits all. Given the characteristics, artefacts, routines and rules required by each of them, and given the specific project and organization context, some of them may be applicable, and some others not. Some may provide good results while others may be struggling to adopt [14], [18], [17], [21], [5], [9], [13], [16], [10], [2], [12], [20], [4]. This means that the adoption of each of them must be considered carefully, and that some guidance about how to evaluate this aspect is required. Different criteria were proposed so far to guide the choice. One of them could be the importance given to the documentation. According to Phil [17] and to Swaraj [19], methodologies like XP, Scrum and Crystal don't put emphasis on the documentation at all. DSDM has some requirements on certain documentation, like feasibility studies. FDD, on the other hand is the method requiring the highest documentation level than others [17], [19]. A further criteria to be used is the level and possibility of involvement for end users. In Phil [17] and Swaraj researches [19], in XP end users are directly and actively part of the team all the planning, development and testing long. In Scrum end users are represented through the project owner, that is the interface between them and the requirements. They also participate in the sprint review. This makes the end user involvement less critical for Scrum that for XP. In the meantime, end users are even less involved in DSDM and Crystal, where they participate mainly during the releases, or in FDD via reports [17], [19]. Then, another important dimension to consider is the emphasis on team meetings [19]. XP and Scrum requires daily brief meetings, while in FDD and DSDM the information are shared via reports. In the between there are Crystal methods, with no explicit rule for team meetings. Another criteria to take into account is the project size measured in terms of number of team members. XP and Scrum are best suited for projects with a limited number of members (maximum 10) [19], [4], while DSDM is even more restrictive posing the threshold at 6 members [8]. On the opposite side of the continuum, there is Crystal, where its practices are useful and valuable also in larger projects. Phil [17] does not fully agree on this stating that both Scrum and DSDM are applicable in teams of any size. In case of bigger projects, they are divided into sub-projects guided from a central one, in an structure like called Scrum of Scrum. Thus, in the end, this confirms Swaraj [19] and Coffin and Lane [4] statement about the limited team size. One last criteria that can be useful to evaluate, according to Phil [17] and to Swaraj [19], is the length of the development cycle. On average, all the development activities are organized in cycle lasting 4 weeks. In FDD they tend to be 2 weeks long, while in XP they can vary from 1 to 6 weeks, though 2-3 is far more frequent [17], [4]. In DSDM the time boxing technique is used, with an overall length varying from days to some weeks. The following Figure summarizes the outcomes of the above considerations. Based on the specific conditions and needs, companies can understand which methodology is most suitable in their own context and what aspects of each one take to tailor their own way to Agile. For instance, if it is possible to have users into development teams, XP and Scrum could fit. On the other hand, if the average size of a team is more than ten members, than Crystal or FDD could be more suitable then the others. Thus, tailoring means applying the right mix of the most fitting methodologies available, i.e. the proper mix of practices.

Despite the useful insights provided, AUP and Lean are not considered into Swaraj [19] comparison. Others researchers have approached the same problem and came up with different results. For instance, Coffin and Lane [4] suggest a different set of criteria to use. They evaluated the appropriateness of the Agile methodologies considering the project size, evaluated considering the team dimension, and the project criticality, evaluated from the system safety point of view.

	XP	SCRUM	DSDM	CRYSTAL	FDD
DOCUMENTATION RELEVANCE	Low	Low	Medium	Low	High
END-USER INVOLVEMENT	Very High	High	Medium	Medium	Low
TEAM MEETINGS	High frequency	High frequency	Low frequency		Low frequency
PROJECT MAX SIZE	10 members	10 members	6 members	Any size	Any size
DEVELOPMENT CYCLE	1-6 weeks	2-4 weeks	80% solution in 20% time		2 weeks

Figure 4:- comparison between different Agile methodologies [8], [17], [19])

The following picture is the result of their analysis, showing if the methodologies are favorable (and fitting) to the circumstances and conditions in a context. In particular, "V" means that the condition is positively impacting the implementation is a methodology, "X" means that there is a negative impact and "-" states that there is no impact.

CONDITION	XP	SCRUM	LEAN	FDD	AUP	CRYSTAL	DSDM
SMALL TEAM	V	V	V	X	X	-	V
HIGHLY VOLATILE REQURIEMENTS	V	V	V	V	-	-	X
DISTRIBUTED TEAMS	X	V	V	V	V	X	X
HIGH CEREMONY CULTURE	X	X	-	-	V	-	V
HIGH CRITICALITY SYSTEM	X	-	-	-	-	V	X
MULTIPLE CUSTOMERS/ STAKEHOLDERS	X	V	V	-	-	-	X

Figure 5:- conditions favorable or not to Agile methodologies adoption [4]

It is worth underlying that in many cases the Agile implementation is not dependent on the product and its criticality in terms of safety, except for Extreme Programming and DSDM. Moreover, in case of DSDM and AUP, companies with a high level of ceremonies are more successful in their adoption, while Scrum and XP will struggle in bringing in their own ceremonies. Other important observations are about team size, distribution and the uncertainty in the requirement sets.

For instance, Scrum and XP have lot in common: both are most suitable in small teams and they both help in dealing with uncertainty and volatile requirements. According to Coffin and Lane [4], Scrum is favorable in case of distributed team. However, one of the Scrum building principles is the colocation of team. That is why it is possible to state that both XP and Scrum are not favorable in contexts where there are distributed teams. Considering that XP consists of a collection of programming practices and that Scrum is a management framework to organize activities, they can be easily combined together. Similar considerations can be drawn also for Lean and FDD. It is thus possible to state that, among the Agile methodologies listed in the picture, the leftmost ones are the most agile ones, while those in the right part are characterized by a lower agility level.

Considering all the previous inputs and considerations, for a subset of methodologies a further comparison is made based on an internally established set of dimensions. Those dimensions are deemed relevant and characterizing the various processes, i.e. different uses, in the company. That choice is thus functional for the matching between the Agile methodologies and the company environment because they describe and characterize the various processes as well.

In the following figure, with the same structure used by Coffin and Lane [4], per each dimension or characteristic it is stated if it is favorable, not favorable or neutral to the specific Agile methodology implementation. The statement is done using a "-1", "0" and "1" scale, where "1" means positive impact or favorable condition. "0" is for conditions that are neither an impediment or favorable, whilst "-1" points towards a negative impact or unfavorable condition for the Agile methodologies implementation. The assessment is done based on the knowledge and comparison previously summarized.

	XP	SCRUM	DSDM	CRYSTAL	FDD
VOLATILITY OF REQUIREMENTS / HIGH SCOPE CHANGE LEVEL	1	1	-1	0	1
LOW FORMALIZATION LEVEL IN PROCESS STRUCTURE AND GOVERNANCE	1	1	-1	0	1
LOW FORMALIZATION AND RELEVANCE OF DOCUMENTATION	1	1	0	0	1
TIME TO MARKET CRITICALITY	1	1	1	0	0
SOLUTION CRITICALITY	-1	0	-1	1	0
END-USER INVOLVEMENT	1	1	0	0	0
HIGH NUMBER OF STAKEHOLDERS	-1	1	0	0	0
HIGH TEAM SIZE	-1	-1	-1	0	0

Figure 6:- new set of dimensions and impact assessment on the favorability of the various Agile methods implementation

The meaning of each dimension is hereby described.

- Volatility of requirements / high scope change level: this dimension is about how good are the methodologies in dealing with changes in the requirements list and/or in the scope of the project. DSDM is the only methodology resulting not that suitable in coping with change because of the analysis and feasibility activities in the pre-project. Any change in the requirement may have a big impact on analysis results.
- Low formalization level in process structure and governance: this dimension is about the rigidity of the methodologies in terms of activity sequences, roles and ruling bodies. In this dimension, less structure means more agility. Again, DSDM results being more formal than other methodologies.
- Low formalization and relevance of documentation: this dimension is about how rigid and how much importance is given to project documentation. According to the Agile values stated in the manifesto [1], less structure and importance means more agility. XP and Scrum result to be the methodologies giving less attention to the documentation, focusing more on the delivered product.

- Time to market criticality: this dimension is about how supportive are the methodologies in delivering valuable products in a fast way. Short development cycles, like in XP and Scrum, are a possible way to achieve earlier product releases
- Solution criticality: this dimension is about how supportive are the methodologies in the case of critical products. Criticality is especially about complex products with an impact on people safety. It is important to note that Crystal only seems to be suited to critical product development, whilst XP and DSDM are not.
- End-user involvement: this dimension is about to what extent end-users are involved in project activities. In XP they are considered a sort of extension of the project team, while in Scrum they can also be part of the cyclic review. Other methodologies don't put particular emphasis on this topic.
- High number of stakeholders: this dimension is about how supportive are the methodologies in dealing with a high number of stakeholders. With the Product Owner role, in Scrum there is a specific person appointed in dealing with stakeholders and prioritising their requirements. XP struggles in dealing with a lot of different voices, while other methods are not particularly supportive, nor impediments on this. For instance, it is possible to prioritise on different stakeholder voices using standard methods like MOSCOW [8].
- High team size: this dimension is about how applicable are the methodologies in case of large teams. Some Agile methodologies are specifically designed for small teams, such as XP, Scrum and DSDM, while others, like Crystal, are not particularly affected by this factor.

As further consideration, it is possible to see that this comparison is in line with others already presented. This means that different methodologies are characterized by different factors, resulting thus in different agility levels. One more time, leftmost methodologies are those characterized by a higher agility level than others. However, it does not mean that rightmost methodologies are not agile. DSDM and Crystal, for instance, result being less agile than XP and Scrum, but still they are more agile then classical project management approaches.

Processes and methodologies matching:-

In this section the previously introduced processes (i.e. Technology Development, Product Development, Product Life-Cycle Management, Service Development and Continuous Improvement) are matched with the most famous and common Agile methodologies. The matching is meant to support the understanding about which methodology is better suited for a specific purpose, i.e. process.

The matching is achieved by comparing the affection level ("High", "Medium", "Low") of each characterizing dimension for each process with the corresponding dimension used to assess the favourability for the various Agile methodologies (described in previous paragraph). In order to allow the comparison, the "High", "Medium", "Low" scale is translated into numbers according to the following rule: High = 3, Medium = 2, Low = 1. In this way every dimension, either positive or negative for an Agile method, impacts the fit to the process based on how much it characterizes the processes (or applications).

	TD	PD	SD	PLC	CI
XP	3	4	5	2	5
Scrum	8	13	10	9	10
DSDM	-9	-8	-3	-4	-5
Crystal	3	3	1	3	1
FDD	7	8	5	4	6

Figure 7:- matching between common Agile methodologies and development processes

The most outstanding result of this matching is the unfit between the DSDM methodology and the various processes. This means that DSDM is hardly applicable in the R&D context within the company. That is because the most characterizing processes' dimensions and the strong point of DSDM do not match each other. This means that DSDM does not support the agility level needed by those processes in terms of scope and requirements change level

and in terms of applicability in case of critical solutions and products. Therefore, DSDM confirms its fame of least agile amongst the Agile methods.
Considering the other methodologies, further considerations can be drawn from the following picture. It is graphical representation of the matching.

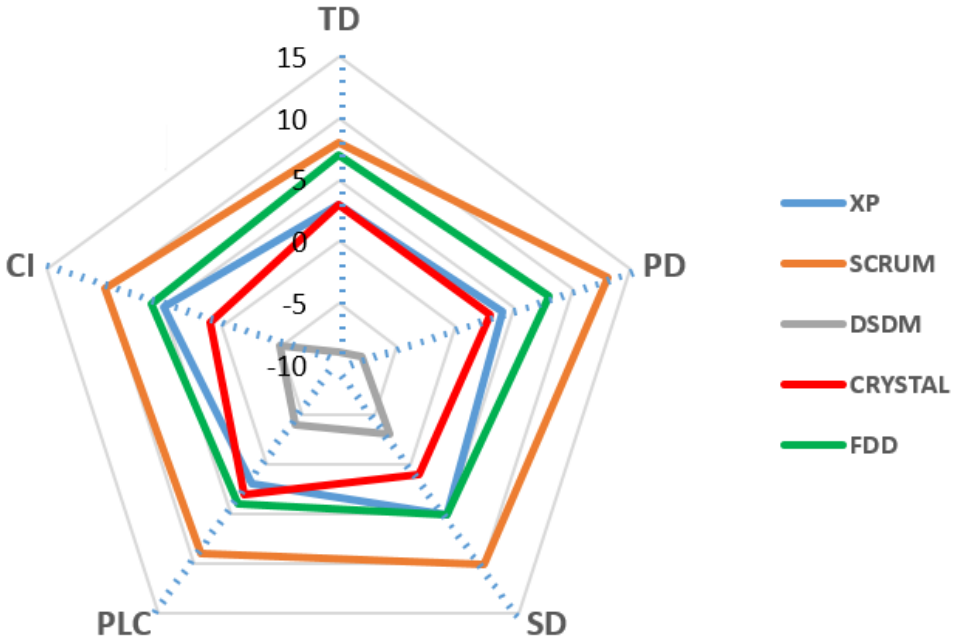

Figure 8: spider chart displaying the fit between methodologies and processes

Together with the DSDM unfit, another important result is the fit between the Scrum methodology and all the processes. From the spider-chart it is possible to notice that Scrum has the best fit with all the processes. That could be due to the fact that it is a project management approach, thus a way to organize activities, to empower people and to release products. It is also important to remember that, for this characteristic, Scrum deals well with other and more practical methodologies, such as Extreme Programming or Feature Driven Development.

Considering FDD, it appears to be the second best choice for all the applications, especially for Technology Development projects. That is because FDD allows low formalization in terms of activities and documentation. Furthermore, due to the incremental development of the final outcome, FDD supports in dealing with changes in scope and requirements.

Against the expectations, though a high agile level from the change and formalization dimensions, Extreme Programming is not fitting that much into the R&D context. That is because it is limited by its inner difficulties in dealing with multiple stakeholders and not suitability when developing critical systems. However, in Service Development and Continuous Improvement projects, XP could be a decent choice. It is important to add that XP is suggested when there are extensive software development activities. It could be the case of a new service product enabled by a new software, or the case of new IT tools development for internal activities. An improvement possibility could be to use both XP and Scrum in those activities in order to achieve the maximum from both of them.

A particular comment is deserved by Crystal methodology. From the comparison it appears that nothing will change if Crystal is used. For all the five processes there are no positive nor negative impacts if Crystal is implemented instead of other methodologies. The main explanation could be its adaptability to all contexts. Moreover, Crystal is a collection of practices, whose application is suggested in certain contexts rather than in other. This means that, when it comes to Crystal, it is better to dig down into details and consider partitions of it as objects of the analysis, instead of considering the methodology a single item.

Conclusions:-

Since the Agile Manifesto [1] times many Agile methodologies were proposed and spread. Some of them became more popular than others, often due to the perceived benefits. It is important to remember that the incubator of the Agile methodologies was the software industry, but they are now spreading also outside that context [5]. However, it is very hard to understand which one (or what mix of methodologies) to pick-up and implement. All the researchers agree on the fact that there is no one size fits all. It is thus important to understand, given their intrinsic differences and strengths, what are the most suitable methods and methodologies on a case by case basis [17], [21], [19], [4].

Different studies on that topic pointed out that the appropriateness of the agile methods can be evaluated considering the project size and project criticality dimensions [4]. By taking into consideration XP, AUP, Scrum, Lean, DSDM, FDD and Crystal, Coffin and Lane [4] argued that XP is generally good on small and highly dynamic projects, but it can be combined with other management methodologies (and even scaled up). AUP provides routines that can be good in larger teams, distributed teams and in projects with high criticality. It is also a good method in those cases where the corporate culture is slow in changing from waterfall standard model: it works as an introduction to the Agile mindset [17], [21], [4]. Scrum and Lean are approaches to the overall process and project management aiming at the business value maximization and waste reduction. None of them suggests practical techniques. In this way, they can complement other methodologies, both already in place in the context or newly introduced and Agile ones [11], [17], [4]. DSDM is heavier and more formal than others, but it is very business centric: it focuses on current business value opposed to risk. It can be compared with AUP. Crystal provides a set of methodologies to choose from, depending on project size and criticality. With their increment, new mechanisms and practices are added in order to support the burden of larger teams and higher degrees of criticality required [8], [4]. Finally, according to Coffin and Lane [4], FDD can be used as complete agile method by itself, or it can be combined with other frameworks like Scrum, Lean or XP. They state also that it is important to remember that no methodology should be taken as it is on the paper. They must be adapted to the specific context, so that there are no identical implementation of the same methodology.

In order to compare the various methods, different criteria, or set of criteria, can be used. Many authors described and compared the most known Agile approaches according to the importance and relevance given to the documentation, the End-user involvement level into the project, the emphasis on team meetings, the project size calculated in number of team members and the length of the development cycle. The Agile methodologies that usually were compared are Scrum, XP, DSDM, Crystal and FDD [8], [17], [21], [19], [4], thus they were used for the purpose of this paper as well.

The comparing dimensions used in the paper differ from those just listed. That is to better assess the matching between the methodologies with the specific applications into the context. The context is the one of multinational food packaging company with already established and structured development processes for its portfolio. Moreover the company heavily invests in research and development. Within the R&D department there are several processes in place to be used depending on the various projects' scope. To develop new technologies for future new products and keep the competitive edge there is a Technology Development in place. A Product Development process is implemented in order to exploit current technologies and satisfy current customers' needs. With the aim of developing service-products, there is a Service Development process. Services are then used to enrich the product portfolio and to diversify the value proposition based on customer need. The Product Life Cycle Management process is used to solve product issues and claims from the market and to extend current products' lifetime with new features or performance improvements. Finally, there is a Continuous Improvement process used to identify improvement areas in the other process and to design and implement improvement solutions. These processes represent the various applications where Agile methodologies can be implemented within the R&D department. They are thus described with a set of dimensions that were deemed meaningful for the specific purpose. Thus, similar criteria are used to describe the chosen set of Agile methodologies. From the process perspective, those dimensions are the volatility of initial requirements and scope change level, the formalization level of the processes in terms of governance and structure, the formalization level and emphasis given to the project documentation, the time to market criticality in each application, the criticality of the solutions outcome of the various processes, the end-user involvement in the project activities, the number of stakeholders that typically have interest and a voice into the projects and how big teams typically are in the various applications. Per each process and per each dimension the evaluation is made considering how important or how affecting/characterising that aspect is in a "1 - Low", "2 - Medium" and "3 - High" scale.

Agile methodologies are described using the high requirements volatility and scope change level, the low formalization level in terms of process structure and governance, the low formalization level end emphasis given to the project documentation, the time to market criticality, the solution criticality expressed considering its complexity and safety issues, the End-user involvement level, the high number of stakeholders that could possibly be involved into the project and the high team size of the project team. All those characteristics were classified, per each Agile methodology based on if they would positively, negatively or not affect the implementation of that specific practice in a context characterized by that factor. The used scale was "-1 – Negative impact", "0 no impact", "+1 – favourable impact". In this way, when matching the methodologies with the processes based on the various dimensions, higher weight is given to those characterising each process. Moreover, it is taken into consideration if they are favourable to each methodology implementation or if they will be an obstacle. The overall outcome of the matching is the map of how suited are the various Agile methodologies in the current company R&D context for the different applications. The results are summed with a spider chart. There are two main outstanding results: Scrum methodology results as the best fitter to all the applications and, on the other side, DSDM results to be the choice to avoid. Concerning Scrum, the result is given to the fact that it is a project management approach and that it can be implemented and applied together with other practices. About DSDM, it does not support the required change level and the development of critical solutions.

Other considerations can be drawn about the other methodologies. FDD results to be the second-best choice after Scrum, especially in the case of Technology Developments, due to its low formalization. XP results not to be a solid choice because it does not support the development of critical systems and does not foresee the involvement of a high number of different stakeholders. Finally, considering the Crystal methodology, it results as not being particularly suited or unsuited to any application. The reason for this result can be found in its being a collection of different practices to be used in certain conditions rather than in others. This highlights a possible improvement area for this paper, that is to dig deeper into Crystal's practices and to use clusters of its composing practices into the matching. Another interesting improvement area is about the set of dimensions used to describe the processes and the Agile methodologies. Other interesting insights could be collected by changing some or all the criteria used, so to have different matching results in the end. Those matching could also be compared each other. Still on the same improvement idea, the describing dimensions could be more strictly related to the Agile principles and values described into the Agile Manifesto [1].

With this paper an approach on how to assess the fit between methodologies and application is proposed and used to evaluate the suitability of a given set of Agile methodologies. The Scrum methodology results to be the most fitting one in the context of the R&D department in a food packaging company. However, the approach could be refined and validated if applied in other companies, both into the same market or outside of it. This type of study could provide interesting results and insights about the different suitability for the Agile methodologies. Having then those information, it will be possible to provide a qualitative understanding of how impacting differences into the context are when implementing Agile methodologies. Therefore, it will be possible to address specific course of actions aiming at mitigating or levering the effects of such factors.

References:-

1. Agile Manifesto (2001), Manifesto of Agile Software Development, http://www.agilemanifesto.org/ (accessed April 2016);
2. Base36 (2012), a Compunel Company, Agile and Waterfall methodologies – a side by side comparison, http://www.base36.com/2012/12/agile-waterfall-methodologies-a-side-by-side-comparison/ (accessed February 2017);
3. Buede D. M. (2009), The Engineering Design of Systems, Models and Methods, Wiley;
4. Coffin R., Lane D. (2006), A practical guide to 7 Agile methodologies, http://www.devx.com/architect/Article/32836/0/page/4 (accessed February 2017);
5. Conforto E.C., Salum F., Amaral D.C., da Silva S.L., Magnanini de Almeida L.F. (2014), Can Agile Project Management be adopted by industries other than software development?, Project Management Journal Vol.45, Iss.3;
6. Cutrona G., Margini A., Fantuzzi C. (2016), Value Chain v.s Life Cycle approach for Product Extensions, 2nd CIISE - Conferenza INCOSE Italia su Systems Engineering;
7. Cutrona G., Margini A., Fantuzzi C. (2015), Structured Product Development Process Implementation for a Packaging Company, 2nd IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control CESCIT, IFAC-PapersOnline, Vol.48, Is.10;

8. DSDM Consortium (2016), Agile Project Management v2;
9. Ghilic-Micu B., Stoica M., Mircea M. (2014), Collaborative environment and agile development, Informatica Economica Vol.18, Iss.2;
10. Ivan I., Ciurea C., Vintila B., Nosca G. (2013), Particularities of verification processes for distributed informatics application, Informatica Economica Vol.17, Iss.1;
11. Kaleshovska N., Josimovski S., Pulevska-Iovanoska L., Postolov K., Janevski Z. (2015), The contribution of SCRUM in managing successful software development projects, Economic Development;
12. Lee G., Xia W. (2010), Toward Agile: an integrated analysis of quantitative and qualitative field data on software development agility, MIS Quarterly Vol.34, Iss.1;
13. Lindgren O., McAllister J. (2014), Agile teams: do's and don'ts in agile software development, Journal of Socioeconomic Engineering;
14. Margini A., Cutrona G., Fantuzzi C. (2016), A lean AGILE approach to Service Products Verification and Validation – a case study from a packaging company, VVT Passion and Deployment Challenges, AISE and INAF conference on Verification, Validation and Testing;
15. Margini A., Cutrona G., Fantuzzi C. (2015), Product Service System Design: How to Design Humans, Application of a methodology in a PSS development with high human involvement, Journal of Engineering and Architecture, Vol.3, Iss.2;
16. Muller E.M. (2014), Self managing teams in agile project management – pain or gain?, Holistic Marketing Management Journal Vol.4, Iss.1;
17. Phil M. (2015), Comparative analysis of different agile methodologies, International Journal of Computer Science and Information Technology Research Vol3. Iss.1;
18. Smartsheet (2016), What's the difference? Agile VS Scrum VS Waterfall VS Kanban, https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban (accessed February 2017);
19. Swaraj G. (2014), Comparison of key methodologies in Agile, Quotium, http://www.quotium.com/performance/comparison-of-key-methodologies-in-agile/ (accessed February 2017);
20. Waters (2007), What is Agile? (10 key Principles of Agile);
21. Wirf-Brock (2015), Changing your Agile practices: a story of shifting from Scrum to Kanban;