Maximising the potential of Agile practices

by

Shobhika Bharti
2017

A Dissertation presented in part consideration for the degree of
MSc Information Systems and Operations Management

ABSTRACT

Agile methods are popular choice in the technology sector for the flexible product delivery. There are many papers which researched the agile challenges from different perspectives such as culture, documentation, tools, team dynamics and human behaviour but it is important to know whether the people who work in the agile environment are satisfied with the agile practices. This study is aimed to find out the satisfaction level among the agile practitioners and the possibility to overcome the challenges using agile mindset. For this, the past papers were analysed to understand the concept of the agile, then studied the challenges in technology sector. Further, two methodologies were adopted to analyse the agile environment. First, interviews were conducted to know the perspective of the scrum master on agile practices. Second, the survey was conducted to know the perspective of the developers/testers towards the agile practices. The data collected from the survey is further analysed for patterns and current agile trend among the respondents. The study further asked the opinion on the happiness and stress factor in the agile to assess the satisfaction among the practitioners. In conclusion, in this study the agile is assessed from different aspects which can affect/contribute to the satisfaction factor.

Table of Contents

LIST OF FIGURES

Acknowledgments

I would like to express my sincere thanks to Dr Robert Pasley for his supervision to help me streamline the agile concepts on this dissertation. His guidance helped me visualise the qualitative content from a statistical perspective and understand the concepts better. Professor Pasley allowed this paper to be my work but steered me in the right the direction whenever I needed it. Also, I would like to thank all my professors to share their knowledge, to ignite curiosity to learn more and to help me connect the management concepts to the real-world scenarios throughout this year. Further, I would like to thank my friends and acquaintances who helped me conduct interviews and surveys to collect the data for this dissertation. Without their help, the findings would be a challenge to study the agile trends in the technology sector. Lastly, I would like to my family for motivating me to give my best performance throughout the masters' course.

Chapter 1: Introduction

The software industry, which has undergone more rapid and profound change than any other industry over the past 30 years, is revolutionized by the agile innovation (Rigby, et al., 2016). Agile software development (ASD) is trending in the technology sector because of its advantages over traditional methods of development not only in decreasing time to deliver the product but also in increasing development flexibility and product quality (Rodríguez, et al., 2014). In Agile methods, the focus is on lightweight working practices, constant deliveries, and customer collaboration over long planning periods, heavy documentation, and inflexible development phases. Though, it is still unclear whether it is agile which is a misfit for some of the projects or the inability of the projects to transform into an agile culture. Since agile is just a mindset in the form of project management technique called scrum, continuous improvement technique called Kanban, waste management technique called lean and coding technique called Extreme programming or XP (Stellman & Greene, 2015). Together these methods create an eco-system for the flexible system in the technology sector, which is penetrating every industry. However, agile may not be well suited for every project as there are many factors that can impact at different levels such as team, project or organisation (Ayed, et al., 2017). To adopt agile methods, one must know the difference and their usage before remodelling their organisation for agile transformation (Rigby, et al., 2016). Nevertheless, with so many success stories surfacing in the software industry, agile is a preferred choice for the project management because it provides a flexible platform for the dynamic projects in changing environment (Serrador & Pinto, 2015).

In 2001, the "Agile Manifesto" was written by practitioners who introduced 12 principles (see APPENDIX 3) of the agile software focusing on individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation and responding to change over following a plan (Sutherland, 2014). The principles were developed with an honest intention to address the gaps in the traditional methods which proved effective in increasing the number of successful software product deliveries and minimising the delay in software delivery process. One of the best features of the agile is its continuous customer involvement to keep client abreast of the challenges and the accomplishments in the software product (Serrador & Pinto, 2015). However, nothing exists in isolation, and often, the variables are satisfied at the expense of another variable in any operations. Likewise, the agile is also exposed to faults and failures when system consists of the requirement, communication, tasks,

meetings, team, organisation, people, culture and so on. The aim of the study is to know the satisfaction level among the agile practitioners who are working directly on the software product and to explore possibilities to improve the agile experience using agile mindset. Many papers researched agile from the different perspectives to solve the issues with the agile software development and to see whether it fits the organisation culture or not. But this study is targeted to find the satisfaction level among the key stakeholders responsible for successful delivery of the product, i.e. scrum master and the team, despite the challenges they face and what measures do they follow if they encounter a roadblock to the delivery process.

Chapter 2: Literature Review

For the further study of the topic, this paper adopts a literature review approach to identify the various challenges related to the agile software development. The issues are documented from different perspectives such as organisation, culture, team dynamics, and technology view. The first part of this section introduces to the hybrid methods of agile followed in the industry and presents arguments from scholarly articles and lays the foundation for the next sections. The second part discusses the challenges encountered in the technology sector by adopting agile and third part discusses the importance of coach to understand the agile methodologies.

2.1. What is Agile?

Agile is a set of methodologies that create a big difference in the way team thinks, works, or take decisions ultimately creating a flexible mindset. The agile transformation creates a culture which becomes the quintessential quality of the team to achieve stellar results by the successful execution. In the past, many projects adopted agile but only a few experienced exceptional results while others noticed the ordinary existence of the agile (Stellman & Greene, 2015). Though, agile is not new to the industry; its concepts are inspired by the Toyota Production System(TPS) from manufacturing industries where high quality products were produced at lowest price with minimum waste. The TPS started a manufacturing revolution that confounded American manufacturing enterprises for decades that how can high quality can be produced in cheap cost with almost no defects and no work in progress items on the product line (Ågerfalk & Brian, 2006). It is often debated that TPS is not meant for the impatient executives, who were trying to get the quick results, rather incorporating TPS in a organistation is a proactive approach of imbibing the cultural transformation, developing a habit so that the change doesn't seem like a change anymore (ZipKin, 1991). Likewise, the agile software development follows the principle of TPS to minimise the waste and the cost of the various IT operations. The need of agile transformation surfaced when the unproductive traditional methods of software development often resulted in software products that were out of scope by the time they were functional. The software industry adopted agile in the form of Scrum to limit the failures and delays in the industry (Sutherland, 2014). Methods for the agile comprise a set of practices developed by experienced practitioners which emphasise on focussing on things for quick delivery of the software product by making processes clear, concise and straightforward (Dybå & Dingsøyr, 2008). The people who developed the agile methodologies created

twelve principles to create a satisfactory product called Agile manifesto listed in the table in Appendix 3 (Beck, et al., 2001).

In Agile, the teams are empowered and allowed to participate fully in the following bottom up approach for estimations and direct participation in daily stand-up and retrospective meetings (Stellman & Greene, 2015). The worldwide adoption of agile encourages new possibilities with intriguing questions, but for successful implementation, the first step is to execute the steps accurately, implement the concepts carefully and wait patiently for the results (Rigby, et al., 2016). Extreme Programming or XP was the first popular agile method that was used in software industry. It mainly focussed on the programming practices, and the employees who worked on the XP were more satisfied working in this new agile environment compared to the traditional approaches, but there was no significant improvement in the product delivery (Dybå & Dingsøyr, 2008). Later, Jeff Sutherland with Ken Schwaber developed a project management technique called Scrum as a faster, reliable and efficient way to create a software in the Technology sector. As per Sutherland, the scrum is the evolutionary, adaptive, and self-correcting systems and it is designed to incorporate the software industry needs initially but since the scrum methodology is a project management technique, it can be used in any industry (Sutherland, 2014). In Scrum, teams are self-organised, follows bottom-up approach for duration estimates and are retrospective when it comes to software quality management (Elssamadisy, 2009). The team roles are divided into three parts: A Product Owner, the Scrum Master and the development team. There are three important events: the sprint planning, the daily stand-ups, and the sprint review/retrospective meetings. Then three artefacts: the master backlog, the sprint backlog and the burndown charts (Cole & Scotcher, 2016).

In Scrum, a self-managing team develops software in increments (called sprints); each sprint starts with planning, followed by performing tasks, and ends with a review. Then in daily stand-up meetings, the team coordinates, brainstorms and makes decisions daily. All the features to be implemented are registered in a product backlog which maintains everything that is needed in the final product based on current knowledge. The product owner prioritizes which backlog items should be developed in the following sprint. The backlog is a constantly updated list of business and technical requirements for the product being built or enhanced. Multiple stakeholders, such as clients, project teams, marketing and sales, management, and support, can participate in the planning phase to identify the product backlog items (Moe, et al., 2012)

Unlike Scrum, the third agile method, Lean is a way to identify and reduce waste (Rigby, et al., 2016). It focusses on the eliminating waste by recognising the work that doesn’t deliver value to the project. It fosters

learning process as every team member is empowered to look for the waste and enforces decision-making skills by advising to take decisions only when you have enough information about the problem. Finally, it asks to deliver as fast as possible to minimise the delay as time is money in business and finally, forces you to see a micro and macro picture of the project (Stellman & Greene, 2015). The seven principles of the lean are listed in the APPENDIX 4. Eliminating waste is the fundamental lean principle where the first step is to see waste then the second step is to eliminate the waste. For example, the unnecessary time spent in the communicating things can be a waste or the transportation time to the meeting to convey a single message to the other team member can be a waste. In software development, extra features that are not required in the application, switching time to another task, partially developed software that is waiting for the approval or technical delay due to the failure can be areas where the time can be saved. The main idea of the lean is to constantly look for these areas and think what can be done to minimise the waste (Poppendieck & Poppendieck, 2003).

Another agile methodology, Kanban is a Japanese word where "Kan" means "signal" and "ban" means "card". Initially, in a Japanese manufacturing environment, this card is used as a signal to tell an upstream step or a downstream step to produce or to stop the production. This tool has been fundamental to the "continuous improvement" or "kaizen" in Toyota for sustainability. The same concept is used in IT operations to reach a sustainable position as Kanban is not a software development methodology or an approach to the project management but a process improvement technique (Anderson & Reinertsen, 2010). The foundational principles of Kanban are rhetoric but resonating with the core idea of the team to work collaboratively and yet independently. The first principle simply states "start with what you do now" as one needs to be focused on the job. The second principle states "agree to pursue incremental evolutionary change" as one needs to be progressive to achieve some results. Third principle states "Initially, respect current roles, responsibilities & job titles" as one should respect the position to fulfil the assigned duties. Once the team member is comfortable with the principles, then the member must adopt the core practices: visualise, limit work in progress, manage flow, make process policies explicit, implement feedback loops, improve collaboratively and evolve experimentally using models and scientific methods (Stellman & Greene, 2015).

Kanban method uses the board to signal the team about the current work in progress. This board is a graphical representation of all the tasks that need to be completed from start to finish which would presumably have three columns; things to do, work in progress and completed. A big advantage of adopting Kanban is that it can be managed by individual or team to handle the workload or scale project. Also, Kanban encourages the team to begin with

the status quo and start building the board by contacting the people directly (Cole & Scotcher, 2016). However, in some organisations, Kanban is used as a project management technique where meetings are on demand, Kanban board is to spot bottlenecks and focus is on upstream quality. It has evolved into different project technique since it just uses the board to visualise team's progress and reduce the time on meetings (Brechner, 2015).

In summary, Agile is a collection of four methods: Scrum, Kanban, Lean and Extreme programming or XP. Depending on the nature of the project and current stage of the agile adoption and agile transformation, projects can choose to follow one of the agile methods or the fusion of the two or more methods. Hybrid methodologies are not the new concept in the agile practices where the projects take advantage of using a fusion of the concepts of agile such as Scrum and Kanban or Scrum and Lean. Scrum follows incremental software development through a sequence of iterations called sprints, whereas the primary goal of Kanban is to maximise the workflow and reduce the lead time (i.e., the average time to complete one item) by restricting the amount of work in progress. The fusion of Scrum and Kanban complement each other as scrum helps in the project management and Kanban removes the bottlenecks by identifying them as soon as possible, thus facilitating continuous improvement (MAHNIC, 2014). Other fusion of agile is Lean and Scrum which is not very successful on a large scale but intriguing enough for the scholars since it focusses on the waste management. Lean is a concept of the waste management, but there are very few cases in the software development where lean is cited as useful as the combination of the scrum and lean, but it is noted that the lean helped in reducing the bottlenecks whereas Kanban aided in keeping track of the roadblocks. The findings from the 30 experience reports stated that there is no one-type-fits-all solution for agile and lean software development. Thus, the organisation should reflect on its development context, project objectives, and other constraints before embarking on the journey of lean and agile software development (Wang, et al., 2012).

In agile, the team is empowered to work together on the issues to deliver the final product. Usually, the team size is restricted under 10 to enable all the members to be addressed in daily stand-up meetings. Many software development iterations follow 2-week iterations which involve the scrum practices of the product backlog, sprint planning, visual board, collective code ownership, daily stand up meetings, and sprint retrospectives (Drury-Grogan, et al., 2017). Stand-up meetings help to focus on the roadblocks from day to day basis. The bottom-up approach is followed where each team member comes with the estimated time to complete a task, then the estimated time to complete the work is decided as a collective task of the team members in the meetings (Sutherland, 2014). Shared code ownership gives the team have

the freedom to change other team members' solutions, whenever the developer finds it necessary. Finally, to focus on the quality issues, agile projects have retrospective meetings which take place at the end of the sprints to discuss deeper into the challenging issues faced by the team (Drury-Grogan, et al., 2017). The concept of pair programming is from the extreme programming, but it is often used in the scrum to foster the development process. Apart from the regular project management methods, agile has incorporated various other techniques such as pair programming. It helps developers work together with another developer, empower the partner's work, and learn collaboratively (Dingsøy & Lassenius, 2016).

The agile enforces on the concept of the team to be aware of all the operations for the product delivery and solve issues productively and progressively. However, some adoption patterns for agile transformation depicts a different story of the less stellar outcomes. Since the holistic approach of the agile considers all the major factors to developing an excellent method to satisfy the technology sector's projects, there can be various reasons that can be attributed to challenges that are faced in the industry from different areas. The next subsection will discuss the issues that are prevalent in the agile software development.

2.2. Challenges faced in the organisations

Agile is successful in many sectors, but as Jeff Sutherland repeatedly points out in his book, it can be dangerous in the hands of the wrong people (Sutherland, 2014). In a study of agile awareness, some executives were questioned about agile, they threw some agile jargons such as sprints," "time boxes" then claimed that their companies are becoming agiler, but their responses were not satisfactory to run projects on agile. Consequently, the executives inadvertently manage in ways that counteract the principles and practices of agile, hence undermining the effectiveness of agile teams (Rigby, et al., 2016). In practice, the organisations can easily adopt agile practices but the transition can take few years depending on the working environment and past practices as one can opt for training to know the agile theories and study case studies relevant to their organisation and project. Also, it is often claimed that agile is a mindset to be flexible and client-oriented, but some traditional practices from previous methodologies can be followed while working on agile methods to progress gradually into the agile transformation (Qumer & Henderson-Sellers, 2008). Further, in the next section, the various factors that are acting as a challenge for agile transformation are discussed from the perspective of the ones who are directly involved in delivering the quality software: the scrum master, the product owner and the team.

2.2.1. Cultural Change

An organisation can claim to have a "strong culture" if the trait is widely shared among employees. The most prominent "rituals" and "organisational stories" are anecdotes given to illustrate distinct cultural traits of the organisation. Sometimes, the strong traits in the organisation are the most important factor for financial success in the company (Lee & Yu, 2004). The philosophy of the agile culture is based on the autonomy and self-organised team while organisation culture is a different context. So, if any cultural change occurs in the organisation, it should fit in agile culture as well as organisation culture. The factors also rely on the type of agile methods used in the organisation. For example, if it single method Scrum or the fusion of the agile methods Scrum, XP and Kanban. Further, the cultural risk factors also include the impact of the change in the multicultural environment where employees come from a different background (M. R. R. Lazwanthi, 2016). Also, the project must be ready for the agile to work, if the team is not open-minded to change then the transition would take some time or worse, fail (Sutherland, 2014). Though it is slightly vague when it comes attributing failures to the cultural change as in many studies many other variables are considered when it comes to agile transition. In fact, in many studies, it is stated that it's hard to separate organisation culture from the agile methods, but to what extent they are related, no studies are confident on the matter (Iivari & Iivari, 2011).

However, the culture has been the center of discussion in the project management from a very long time and various studies have discussed the effect of culture on the project management techniques with the help of different variables, but nobody is sure if the variables will hold true for every organisation. For example, in the case study of Nokia's agile transformation, it was concluded that organisational culture is never fixed, it evolves or distorts with time. The viewpoints of various agile advocates and opponents were different, even when the conclusion was same; e.g. that the agile deployment is a challenge (Laanti, et al., 2011). Also, the opinions such as effectiveness and quality were widely discussed in the organisation, and there was no significant difference between the opinions of more and less experienced respondents. The opinions such as transparency and collaboration were different may be less discussed and thus less understood or simply not yet practised widely or properly enough. In another study, the agile culture factors were evaluated in term of implementation where it was noted that the more you resist agile and try to implement in the current environment, you try to achieve best of both worlds wherein agile revolutionary implementation must be followed in complete isolation (Waardenburg & Vliet, 2013). The figure below summarising the various such impediments in the agile cultural change transformation that can happen within the organisation.

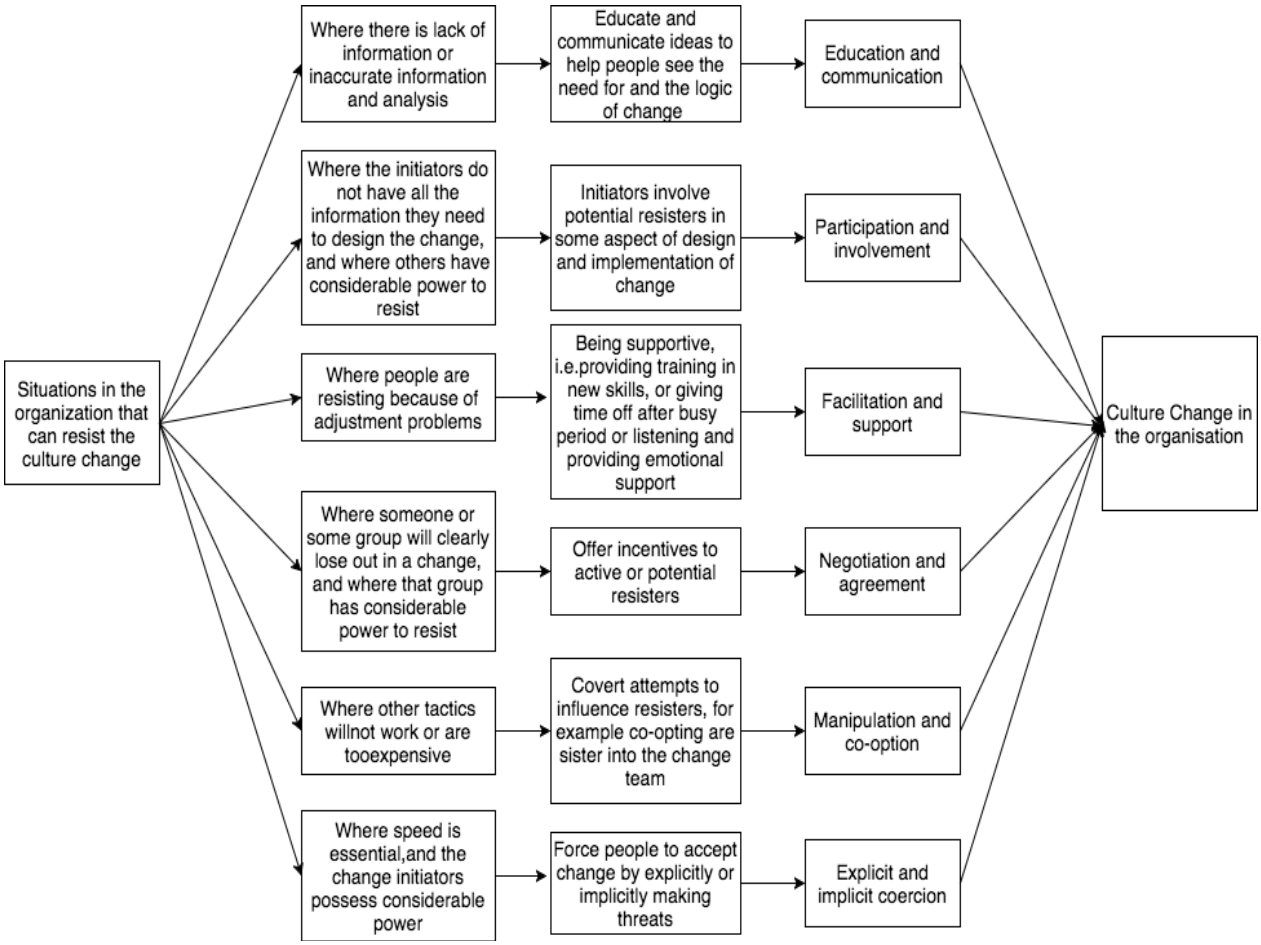

Figure 1: Factors influencing cultural change in an organisation Source: (Kuusinen, et al., 2016)

Further, in Cisco's in-depth study, every process was monitored to see how it was noted that the companies need to tackle two significant challenges. The first is to help teams or business units within the company to make the transition. Second, to develop new management practices that can support agile development method, but there were still gaps such as customers who do not collaborate and dependent team members working directly with the clients might act counter-productive to agile development process (Chen, et al., 2016). The resistance to change can be internal or external, and the organisation opting for agile transformation needs to assess the levels where agile transition can be impactful and thus, proceed to handle the change progressively.

2.2.2. Documentation

Documentation is another widely-debated topic among agile practitioners whether to accept documentation or ignore it altogether. There is a fine line between doing sufficient and committing too much. When it comes to

understanding the points in documentation, the case studies aren't considered for the contextual needs of the project. They probably missed the point that different application domain may have varying needs. For example, in healthcare, one might need a document that can be helpful during software maintenance. While in a small café software, the client may not require the documentation at all since the website can be a promotional tool for the personal business and it might change as per current trends (Ågerfalk & Brian, 2006). Agile methods do not support excessive documentation, focus mainly on user stories and product backlogs as key documents. The continuous requirements prioritisation in the agile helped in achieving minimum documentation and solving a major impediment in traditional method's requirements. Furthermore, overscoping is another issue that used to occur in traditional development where the changes happen after finalising the project scope. However, agile methods are successful in tackling the issues of overscoping and requirements gathering by frequent face to face communication with the client. But the impact of minimum requirements should be studied at the project level based on the team since the field is still immature. It has been noted that the documentation is a vital challenge for the development teams because sometimes, it can be insufficient to communicate requirements to the other team members in the form of verbal communication if the member was not part of previous communication. Likewise, it can be ineffective if it is large projects or teams are distributed in the different geographical area. In traditional methods, functional and non-functional requirements are both parts of the development, but in agile, to the considerable extent, the non-functional requirements are rejected (Inayat, et al., 2014).

Few agile practitioners coined the term as 'Documentation debt' to present a challenge that is raising concerns in agile software development. Because agile methods demand extensive communications, collaboration and trust that, when not present between the communicators, can result in poor product delivery mainly the communication is hindered if the team is placed at two different locations (Fernández, et al., 2015). Since documentation is a part of the software development process that acts a reference for the team members who haven't been part of the requirements engineering. It has been removed from the traditional development process to solve the issue of excessive documentation, but it has created other problems in the process such as when software moves into maintenance, the business rules are missing because the business rule was never a part of the user story. If any issue is encountered during maintenance, it is the job of the development to fix the software (Mendes, et al., 2016). In one of the proceedings paper, it was noted that the time spent on information seeking in the absence of the documentation is more than the time spent on the documenting in agile projects. Since existing form of reference such as sprint logs and backlogs, except the source code

documentation, do not cover the need for the information and information on the software architecture, which can be important to address the contextual requirements of the project (Voigt, et al., 2016).

In another study focused on retrospective meetings, it was observed that a common challenge for the decision was that resolutions made mid-iteration were not documented. Team members updated the centralised tracker with task and estimation decisions during initial phase but did not follow the practice in during the peak time after the initial phase. The missed log posed a problem for the developers who were dependent on the tasks/modules and continued working only to realise that something was missed out in the module (Drury-Grogan, et al., 2017). In agile, the documentation has become a center of discussion because the reference points to resolve an issue are limited in the form of sprint logs, product backlogs and some documentation.

2.2.3. Team dynamics

In agile, teamwork is everything, it starts with the bottom-up approach and ends at the retrospective meetings where brainstorming is a collective process. As per the creator of the scrum, if the Pareto principle is applied to the team dynamics, 20% of the team efforts result in 80% of the success factors. So, the focus area should be the precise selection of the people in the team and the team-building activities (Sutherland, 2014). Agile is a collaborative team work where decision-making process is a compound product of the individual member in a team; it becomes necessary to focus on the team dynamics and team orientation. There is a need to align decision and to implement a shared decision for the project's success. The challenges can differ from projects to projects but most problems can be same. In the figure below, challenges faced by the agile team are categorised into three main categories existing at different level such as in planning phase, in allocation of resources and working collaboratively on a task. Further, every challenge is impacted by various reasons that can be the hindrances in having effective team dynamics (Moe, et al., 2012).

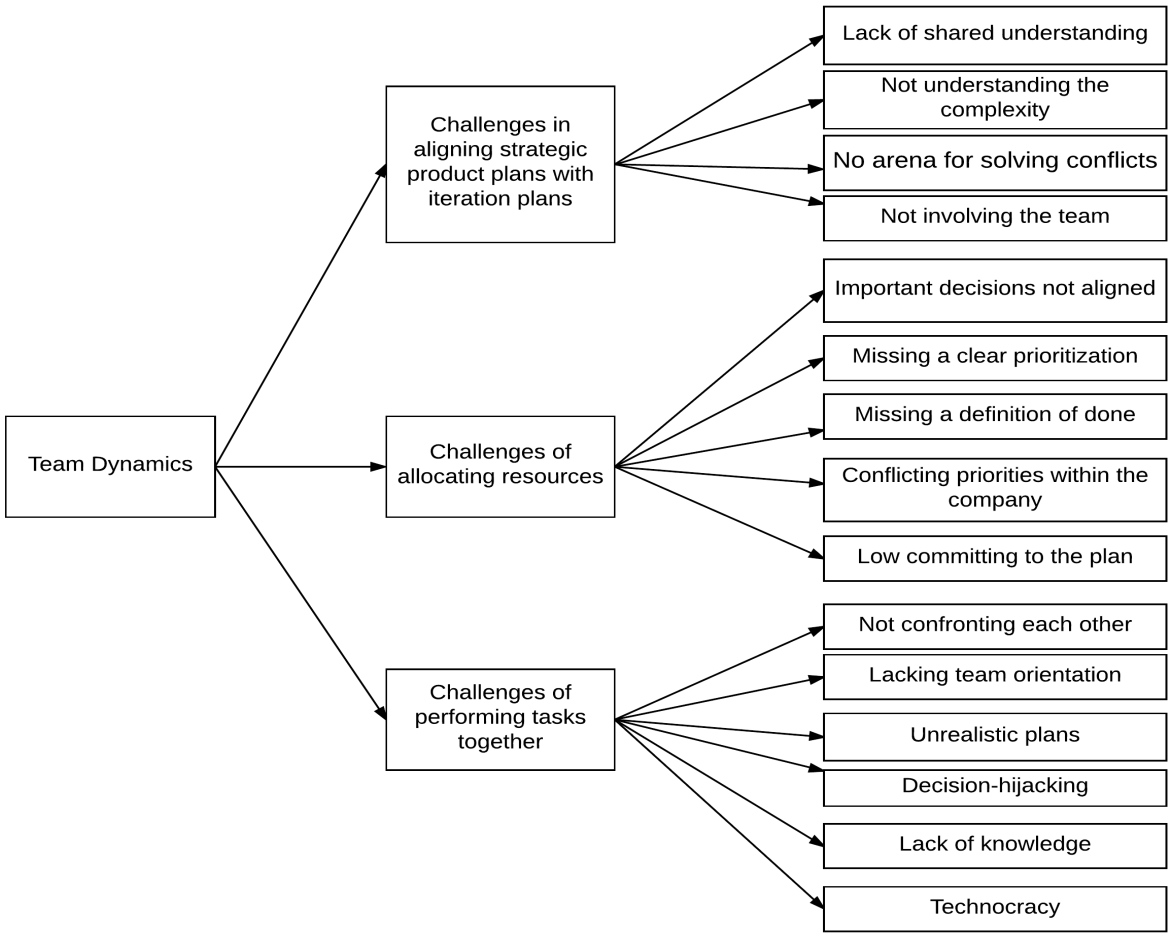

Figure 2: Challenges in agile due to the team dynamics Source: (Moe, et al., 2012)

The teamwork is one of the key aspects of the agile practices which is empowered by bottom-up approach asking the team to give the estimates, daily stand-ups where the team works both collaboratively and independently and pair programming where developers work together to develop code. The team work is fostered by effective communication, mutual support by helping other team members when they are stuck, common understanding when working on subtasks and willingness to share the workload. In agile teams, the workstations are often located close enough in open-plan offices to encourage informal and open communication. In a study of team-work comparison between agile and traditional methods of 477 respondents, it was found that there was no significant improvement in the team's performance (Lindsjørn, et al., 2016). For healthy team dynamics, to some extent, it is a job of the company to build a training system to train their employees regularly. The empowerment of employees at work has been beneficial to have a positive influence on work motivation and satisfaction. Software development is a discipline of knowledge work wherein one can anticipate

seeing these effects, and the idea of empowerment has become conspicuous in agile methodologies, in which advocates emphasise team empowerment and individual authority of the work activities as a primary concern (Tessem, 2014).

Though there can be other factors that can influence the team behaviour in a project. Also, the team factors can vary because of the perception of the author studying the agile system. In another study of the sustainable execution of the agile projects, the categories can be divided into the agile team factors, technological factors and organisational factors. Agile team factors consist of the agile experience, agile mindset, agile coach, result demonstrability. Technological factors comprise of agile practices in the project and the tool support. Organizational factors include the top management support, methodology champion and organisational structure (Senapathi & Drury-Grogan, 2017). The other factor that resists the agile change is strong team culture, the more rigid the team about the methods, the more difficult it gets to transform into the agile teams. Also, lack of knowledge about agile can aid to the agile resistance though this can be overcome by regular agile training and agile coaching (Jovanovic´, et al., 2017).

2.2.4. Tools & Skills

Agile is a mindset that focusses more on individuals and interactions over processes and tools, but it works around the tools, technology, and other operations. The agile transition is not a small step towards a better software development process but a gradual evolution towards sustainable development. The whole idea is to drop the pedantic IT operations to save time and make the team more productive. Tools and third-party software usually aid in the faster productive executive of the software process, saving time. However, with so many tools and software in the market, usually, the team selects the tool which works best for their part of the code. Sometimes, these tools can create compatibility issues consuming too much time of the team impeding their productivity (Martini, et al., 2013). It is essential that the agile mindset collaborates with the existing variables and slowly merge into the organisation culture with the evolution, not the revolution.

The other approach, projects follow is to maintain the work on the cloud. In a study of the fusion of agile and cloud computing, it was recorded that it enhanced excellence equality of products, improved efficiency of developers and fewer errors. It provided distributed application development enabling team members to access data about the application at any time, to share data and to update the priority level of the tasks simultaneously. Also, it comes with added benefits of the centralised deployment and quick accessibility to

the code (Butt, 2016). Apart from the cloud, there are software management systems that help to track the various operations such as JIRA and VersionOne. The recent advances to bridge the various gaps, the industry has launched products which fused the best practices followed in the industry into one single software. Example of such software is DevOps, which stresses more on the communication and collaboration between developers and operators rather than tools and processes. It can achieve Agile goals to reduce team working latency and extend agile principles to entire software delivery pipeline with extra features of simulation, maturity evaluation, production support and deployment planning (Ramtin Jabbari & Tanveer, 2016). With all the features organised at one place, it becomes easy to remember the terms and revise all the steps with just one mouse click.

2.2.5. Human aspects in agile transition and adoption

The study will not be complete without the study of the agile transition and agile adoption since agile is new in the software industry and with its stellar success results, it is creating new paths for the software. Though cultural change poses a serious impediment to the agile transformation since humans resist the change, which is expected but can be modified with the positive outlook towards change. However, negative outlook such as status quo, cynical behaviour, worried or indifferent to change can also impact the transformation negatively. Hence, it very important to identify the factors affecting agile transformation in the organisation. In the figure below, the factors are listed in three categories: identification of perceptions about the change, what is stopping the change and how to accelerate the change. Most of the factors in attitudes and impediments are self-explanatory. However, change is not easy, it takes an open mindset to believe in the decisions, but when an organisation decides to change, it must happen quickly. Here, accelerating change can prove beneficial and have people who are directly committed and working to bring the change. Then people from the management who has a positive outlook to the transformation because the majority of the organisation gets influenced by the strong/weak management. Then, comes the agile champions who have already worked on the agile and know the process thoroughly, so they don’t need extra training to get convinced that agile is adequate for their job. Finally, supporters who believe in the transformation and keeps environment optimistic during transformation by supporting the change (Gandomani, et al., 2014).

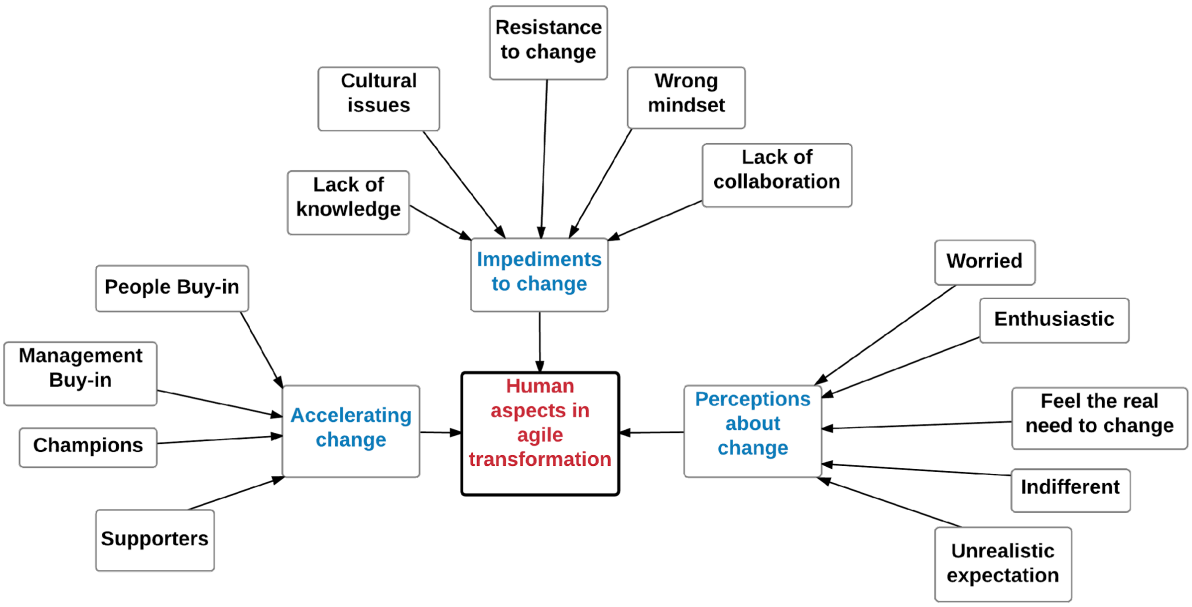

Figure 3: Key factors describing the human aspects in agile transformation Source : (Gandomani, et al., 2014)

Agile transformation is often blocked by the cultural challenges and factors are not even micro-analysed, they just exist. In the case study of cultural challenges in Asia and Europe, the set of problems encountered by Yahoo in a globally distributed project across Asia Pacific, Europe and the US were categorised into three areas: communication, control and trust wherein cultural differences made the challenges more problematic. However, in ThoughtWorks running an offshore development experience in Bangalore India discusses that the local culture reinforces deference to superiors which contradicts the value of team autonomy making the communication harder since team members might be discouraged from exposing problems, warning about non-feasible deadlines, or proposing alternatives to perceived directives from superiors (Ayed, et al., 2017). The study broadly categorised the challenges at the various levels and the key factors are analysed and synthesised in the figure below.

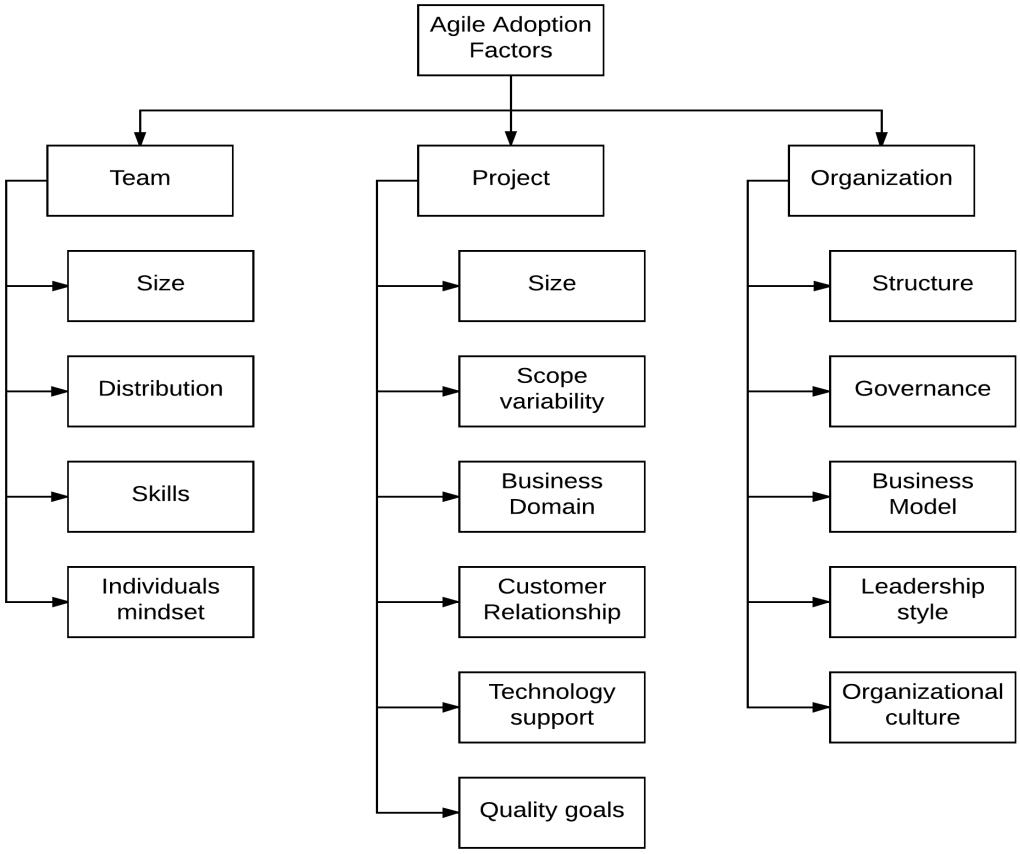

Figure 4:Key factors influencing agile development Source: (Ayed, et al., 2017)

In Samsung Electronics' agile transformation study, the electronic product has been developing software based on the waterfall model. When they adopt agile software development methodologies, they concluded the successful transformation with some challenges such as more agile coaches are needed, experience on engineering practice is insufficient, personal performance evaluation should be improved and keeping small team is important. But their strategy was evident as they first we developed a road map which contains three phases. Then, they revised their process then established a master plan to adopt engineering practices, then, tried to build cross functional teams. Finally, cultivated our agile culture by training employees on agile courses, conducting conferences, and publishing agile newsletters (Kim, et al., 2016). This transformation took place in the presence of the agile coach, who specialises in agile and monitors the team's activities to help them achieve productive results. Further, he/she motivates the employees to adapt the agile by success stories and helps them build gaps by reasoning. In next section, the study will continue to analyse the importance of the agile coach in the agile adoption and transformation.

2.3. Agile Coach and its importance

As in one's professional success, a mentor plays an important role. Similarly, in agile adoption, Agile coach plays an important role. There can be a lot of questions on agile that can arise due to overthinking, anxiety and scepticism. For example, 'what if you don't already have a mindset that works for Scrum, XP, or another agile methodology?' or 'What if you're working in an environment where it's difficult to succeed with the agile values?' or 'What if individual contributions on your team are rewarded far more than team effort?'. The agile coach helps bridge the gaps who helps a team adopt agile by training them to embrace a new attitude and mindset, working with every individual on the team to help him or her to understand "how" and "why" to adopt agile. The goal of the agile coach is to help the team attain a better, more agile mindset. A good coach helps the team choose a methodology that best fits their current mindset, and introduces them to the values, principles, and practices of a method in a way that works for them (Stellman & Greene, 2015).

The agile coach can prove efficient to the organisation. Depending on the variables, they can choose from external and internal agile coaches. External Agile Coaches can provide an impartial view on the company and diverse experience, whereas internal Agile Coaches have a good understanding of the company's business and processes. Agile Coaches significantly reduce the risk of failure of agile adoption and speed up the adoption process and tailor of agile practices to company's needs, highlighting dysfunctions and waste in processes, sorting out industry related agile adoption challenges, etc. The common questions in the agile transformation where agile coach can be required are how to do agile manages requirement, how to get rid of the documentation, how to apply agile to a legacy code, how to keep the quality of code high, how to do incremental design, how to track progress of an agile project, how to get testing done within a short iteration and what are the things that should happen before items hit the backlog (O'Connor & Duchonova, 2014). Besides, the importance of agile coach is very less studied, and some projects do not employ a coach because they just can't afford the fees.

Though the importance of agile coach is also a debatable topic but projects, who opted for the coach witnessed some significant improvement. In a case study of a medium size development company, the regular update of the burn-down chart by the agile coach was a major motivating factor for the project teams; team members were racing to finish tasks before update time. Over the period, charts location turned to "meeting points" for project team members. They used to discuss issues related to the project while checking the project status and progress – thus, team intercommunication was

significantly improved (Anwar, et al., 2016). This result can be expected from the agile coaches because they are the ones who were once agile practitioners and now, took a job to spread the word. Every success seems to create a group of passionate evangelists who are excited to let others know the benefits of agile in the organisation (Rigby, et al., 2016).

Agile methodologies were introduced in 2001, and the principles are so simple that they can be applied to any project in any sector. Many projects are adopting agile because of the success stories associated with the agile. The team plays a significant role in the product delivery, and it is important to analyse the agile practices' satisfaction level among employees since the behaviour influences the culture in the organisation. If the team is open to the working collaboratively, they submit themselves to the continuous improvement mindset becoming part of the agile system and further, minimising the gaps progressively.

Chapter 3: Methodology

After the thorough review of the literature, it is required to know the agile practitioners' perspective on the challenges. To further study the topic, the interview is conducted to know the view of the scrum masters and the survey is carried out to know the perspective of the developers and testers. The product owner is not selected for this study because he/she can work on multiple product deliveries that can introduce many external factors such as client exposure and internal factors such as multitasking, organisation policies and product delivery which would have extended the study. For now, this study will focus on the key stakeholders who are developing the product i.e. scrum master and the team. The questions in the interview and survey were targeted to know the satisfaction factor of the scrum master and the agile developers/testers in the agile projects respectively.

3.1. Interview

For interviews, 24 questions were prepared from the literature review to ask the feedback on the agile practices in projects where 30 people were approached, 10 responded, but due to the busy schedule of interviewees, only 2 participants agreed for a phone interview while other 3 participants agreed to open-ended surveys. The aim was to collect the feedback from the scrum master's point of view since they are ones who exposed to the IT operational issues and work closely with the team. This method recorded responses from the managers who developed software using the agile methodology to explore the challenges and opportunities for the estimation process following agile practices. The audio recording of one of the interviews conducted was transcribed, and other was not recorded but transcribed to maintain the anonymity clause. The questions focus on challenges they face while working on agile practices, what is their experience of agile transformation and what challenges they usually face in the agile adoption to know their satisfaction level compared to traditional approaches. Also, the questions addressed the problems from the literature review on team dynamics, cultural impact, documentation and behavioural effects on the delivery process.

3.2. Survey

The survey was developed to collect data from the developers/testers to gather the feedback on the agile practices in their projects from their perspective since they are working to develop the product in defined time limit. Also, if the assigned task is not completed in the current sprint then it is carried to the next sprint adding more to the assigned work on the

individual. Hence, it was necessary to know the attitude of the team member in the agile project. Also, the survey recorded opinions of the developers/testers towards the agile projects in technology sectors using Likert scale method i.e. strongly agree, agree, neutral, disagree and strongly disagree. The aim of the survey was to understand the agile adoption practices and how satisfied the developers/testers were in the agile environment. The survey examined the respondent's role in the organisation, which agile practices the respondent's organisation had adopted and how they perceive the challenges studied in the literature review section. The questions focused on different issues such as team orientation, culture issues, documentation, productivity and self-motivation. Further, the respondents were given a choice of 'not applicable (N/A)' at the end of each question but if the question still is irrelevant to their area of expertise they can leave the comment. The questionnaire consists of thirteen multiple choice and three matrix scale questions on the survey. The matrix scale questions used Likert scales to know the sentiment of the respondents in the documentation, culture, coding practices, and agile practices to perform sentiment visualization to analyse opinion trends in the agile projects (Author, 2016). The 'Not applicable' option was also included to capture situations if the question is not relevant to the respondent's job role. The survey was developed iteratively with the help of the case studies and followed the university protocol of keeping the responses anonymous. It was created using SurveyMonkey, and an invitation was sent to more than 80 agile developers/testers, but 48 people responded. Further, the survey data was analysed using data visualisation software for various trends. The sentiment visualisation is performed for all the matrix scale questions i.e. on 12 statements from question 14 to question 16 mentioned in the APPENDIX 2 to see the inclination of the respondents' opinion on the challenges in the agile practices.

Chapter 4: Findings/Results

4.1. Interview/Open-ended survey's Findings

The questions were asked from the scrum master's perspective and aimed to find the challenges faced in the agile practices in the technology sector. The findings are from a macroscopic perspective since the scrum master's job is to help the team plan, and to get barriers out of their way throughout the project so they can deliver the software product (Stellman & Greene, 2015). As per findings, all respondents followed the agile practices regularly such as having daily stand-ups, having retrospective meetings, maintaining visual board to track progress of the tasks, and tracking burndown chart as per literature review. Four out of five scrum masters stated that they follow industry standard of 15 days or two weeks' sprint and one scrum master (from fifth interview) follows 1 week sprint for the delivery. Though fifth interview is working on the data analysis but findings are kept for this study because the agile practices were almost same and the scrum master had a similar perspective compared to other scrum masters. The list of the interview questions is attached in APPENDIX 1 and five interview transcripts can be seen in APPENDIX 6. As three findings are from the open-ended surveys, I asked the explanation of the statements which needed more clarification, then corrected the statements as per the reply and transcribed in the final findings. Every sub-section (4.1.X) states the findings and attaches the transcripts from the interview related to the topic and below are the transcripts on the sprint duration followed in the agile projects of the interviewees.

> *"Sprint duration will be decided by a number of releases in a year. If the client expects a release for every two months, then we usually have two weeks' sprint. Sprint duration is also depending on the complexity of requirements that are going to be developed. If they are complex, then most of the scrum team will go with two weeks' sprints. If they are simple, then one week sprint also works." -Interview 1, Q4*

> *"Team decision. Our default is two weeks unless there is a good reason to change this. Sprint 0s are often shorter." –Interview 2, Q4*

> *"I take the industry best practice of 15 days as my benchmark." –Interview 3, Q4*

> *"We follow industry practice of 15 days." –Interview 4, Q4*

> *"Weekly because we focus on data analysis" –Interview 5, Q4*

4.1.1. Cultural Change

Instead of asking every challenge separately, the interviewees were asked for the significant challenge that they face while working in agile practices since some of the challenges may not be applicable to their projects. Every scrum master stated different challenges but two of them stated that the culture issues are the major challenge they face. The first scrum master emphasized on the training to mitigate the issue and it should not be biased towards product delivery only. Second scrum master explained the cultural change from the meetings perspective which requires the team members to allocate the time for the brainstorming sessions for retrospective meetings and regular discussions on the progress of the task in daily stand-ups.

> *"The biggest challenge is Cultural Change. Employees must get trained properly / brainstorm before they start working in the agile environment. It is not easy for the employees who worked in traditional projects and moved to agile, suddenly. They will feel that they are getting pressurised for deliverables on the daily or weekly basis. Demotivated most of the times. As a Scrum Master, we must guide/encourage them with Agile /Scrum Principles which are to be followed, as part of this cultural changes. Sometimes, clients will pressurise developers to take more work and deliver more. This is not fair, as agile is against to that. Agile states that 'Let team decides how much they can commit and how much they deliver in the sprint." –Interview 1, Q7*

> *"Developers do not like the additional meetings early on, especially the honest nature of retrospective and regular stand-ups. Senior management does not like receiving nothing more concrete for planning that estimates and support - they want levels of commitment regarding scope and timeframe that we cannot provide. Scrum requires a genuine cultural change that not everyone/everywhere is ready for or happy with." – Interview 2, Q7*

4.1.2. Documentation

One of the scrum master stated "Documentation" as the biggest challenge as the time spent on the information seeking took more time than the document itself. On the contrary, the same scrum master stated that the documentation slows down the developer's productivity because without documentation, the developer working on the issue must reverse engineer the code to understand the problem to provide solution.

"User guides are updated, developer documents aren't maintained as that may slow down the development as every requirement might not need strict documentation, system architecture is maintained." –Interview 4, Q6

"Productivity is lost when the developers had to do reverse engineering on code which was written by other developers, refactoring can be done." – Interview 4, Q7

4.1.3. Team Dynamics

Every scrum master agreed to some approach to address the challenges in team dynamics and they suggested resolving the issues by conducting training and workshops. Sometimes, if the issue needed more time with the team member, the problem is resolved in one to one meetings where the elaborate discussions are aimed to find the solution for the abrupt behaviour. The scrum masters also stated that they use team programming and pair programming techniques to give lagging person as chance to enhance the skills, which was also stated by (Dingsøy & Lassenius, 2016). When asked about "do you conduct any workshops for behavioural issues to enhance team dynamics", one scrum master in interview 4 replied that "it would be beneficial for the team".

"There will be workshops at the organizational level to handle this kind of issues. We recommended employee to undergo those training. It happens monthly once." -Interview 1, Q16

"We discuss these issues at retrospectives where possible." -Interview 2, Q16

"No, but I think it would be beneficial for the team" -Interview 4, Q16

"It is extremely important to have the team on the same page. If any unexpected behaviour is triggered during the meeting, then it is handled in the separate one to one meeting with the person to resolve the issue." -Interview 5, Q16

4.1.3.1. Training

Scrum masters mentioned various training that are conducted at the team and organisation level to train their team members. However, the training are not conducted regularly and usually, the training takes place before project's initiation phase to get everybody on the same level. One scrum master also agreed to cross-train the team members to increase the efficiency of the team (Interview 5, Q5).

"Team must learn while working. In general, we provide/conduct technical training, once in every two months." –Interview 1, Q17

"Not as regularly as we would like" –Interview 2, Q17

4.1.3.2. Behavioural Workshops

To address the behavioural issues, three scrum masters stated that the workshops are conducted at the organisation level to maintain the effectiveness. One stated that we don't but it can be beneficial for the team.

"Emotional Quotient (EQ) plays a very important role in extracting good output from the Developer. There are several ways to handle it. Intelligent Quotient (IQ)- It can't be changed by Scrum Master." –Interview 1, Q15

"Peer review and regular communication. We attempt to allow people to take tasks to which they are best suited whenever possible." –Interview 2, Q15

"You must instil in them the confidence that they can achieve any task no matter how difficult it is, till the time they have the technical expertise. A lot of coaching is required." –Interview 3, Q15

"No, but I think it would be beneficial for the team." –Interview 4, Q16

4.1.3.3. Meetings

All scrum masters were consistent on the daily stand-ups as most of the issues are resolved and roadblocks are exposed in these meetings. They follow agreed sprint and if anything, that is not delivered in the sprint is conveyed to the client then sent in next sprint. Then retrospective meetings are conducted after every sprint to improve the process. The meetings are divided into two parts: internal meeting with the team and external meeting with the clients.

a. **Team Meetings**

To have effective interaction among the team members, the progress on the current sprint is discussed in the daily stand-ups. The things that must be improved are discussed in the retrospective meetings. If any

exceptional behaviour, such as topics that are irrelevant to the forum, triggered during the meetings are addressed in separate one-to-one meeting with the scrum master.

"The daily stand-up call lasts for 15 mins, max. Yes, we will have the daily stand-up call, during heavy work load too, without fail." – Interview 1, Q3

"Daily stand up is strictly for 15 mins, unless there are items that require immediate attention and take more time." –Interview 3, Q3

"Yes, we organise daily stand-ups for 10-15 minutes and we adhere to them daily meetings as they help assess roadblocks." –Interview 4, Q3

"Daily stand meetings are time tracked for 10-15 minutes" –Interview 5, Q3

b. **Client Meetings**

The clients did not pose any challenge and helped the interviewed agile practitioners when the scrum masters were stuck in their tasks. The client and the team understood the common goal and worked together towards the product delivery and the client did not pose a challenge in this study.

"We don't have this challenge at the moment, but I have done in the past. Ultimately, it's the responsibility of the PM to ensure the client is communicated with how they want to be - I have in the past woken up at 6 am every day to send status reports. If necessary, it's part of my job." –Interview 2, Q19

"Set up meeting on a common agreed upon the time which is convenient to both the parties." –Interview 3, Q19

4.1.4. Tools

Every scrum masters agreed to have central control system such as JIRA, VersionOne, Microsoft Visual Studio Team System (VSTS) and Microsoft VSO where they maintain virtual visual board to the track progress of the tasks. Nobody maintained the repository on the cloud as they do not require it for their project. However, on DevOps suggestion, one scrum master agreed that they can use it but currently, they are not.

"DevOps - It can be used only when the organization has sufficient tools to implement it. For example, Continuous build / Continuous integration / Continuous testing / Continuous Delivery/Continuous Deployment. If an org is good at least one are of the above, then they think about implementing DevOps solutions to reduce the time to market and get benefited with in short time span." –Interview 1, Q12

4.1.5. Transformation & Adoption

Two out of five scrum masters agreed that they faced issues as developers during the agile transition phase since they must deliver results in shorter sprints compared to the traditional methods. Also, one scrum master stated that the change can be taxing for the introvert as they would resist the agile change.

"Agile is a cultural change, all together. If it is the product based company, then I recommend implementing Agile top to bottom. Instead, it implements it partially. Implementing partially results in cultural differences (difference in opinions / stress etc. between employees (agile / non-agile)." –Interview 1, Q10

"Depends on the nature of the organisation. Developers need to be bought in, as do senior managers. They need to understand why we do things like this and benefits they bring, or there will be resistance. Let the dynamic team form; not everyone needs to be vocal and a leader. Support leadership in long term planning with estimates and limited commitments but don't make promises on behalf of the team they cannot keep. Avoid sprints become mini waterfall sessions - get things into the test as soon as possible." –Interview 2, Q10

"For developers, it can be difficult as introverts want to work on code all by themselves." –Interview 4, Q10

4.1.6. Agile coach

This question is not answered because this is a part of the training process, only one agreed to receive the 18 hours training before the initiation phase of the project. This scrum master wanted agile coach as a regular part of the team because sometimes, product owner doesn't know the solution where the client should be contacted for the solution.

"That's right, but I have now moved beyond the role of Agile Coach to that of the Scrum Master. And coming from a technical background, I fully empathise with the development team in the sense that they are under

constant pressure to develop small software packages and demo to the client every 3-4 weeks, and at the same time continue with the respective backlogs too. So, it's a very high-pressure environment they are working in." –Interview 3, Q24

"No dedicated coach but we had training for three days for 18 hours." – Interview 5, Q20

"Yes, we have gaps, and we may get benefit from the agile coach as sometimes, we don't have solutions as product owner also doesn't know. In the end, we raise the issue with the client, and he/she resolves." –Interview 5, Q24

4.1.7. Others

The interview questions also asked whether they find it difficult to work with a client in a different time zones to which everybody agreed working collaboratively with the client. However, one respondent reported issues but stated that it is responsibility of the person in-charge to communicate important things to the client whenever it is necessary.

"Daily, through meetings. Face to Face - Yearly Once. In general, they travel to India for two weeks and spend time with all the team." –Interview 1, Q2

"Internal clients 2-3 times a week, external clients around once a year." – Interview 2, Q2

"Once in 2 weeks" - Interview 3, Q2

"Sales group contact the client and we get the requirements for the development" - Interview 4, Q2

"Product Owner talks daily and team talks weekly in a time bomb which lasts for 2-3 hours" –Interview 5, Q2

The agile method Scrum was dominant method in the interviews. The scrum masters followed daily stand-up meetings, kept the stand-ups limited to the current sprint discussion and followed retrospective meeting after every meeting. The challenges, such as documentation, culture, behavioural aspects, in the interview are similar to the challenges mentioned in the literature review. However, the presence of the agile coach in the project is limited in the interviewees' projects.

4.2. Survey Findings

The survey captured 48 distinct responses. The findings are analysed in the section individually to present absolute numbers or percentage of the collected responses. But some statements, which were relevant to the one section in literature review (i.e. 2.2.X), are further analysed for the trends in the agile projects. Some findings are tracked for the individual opinions (in section 4.2.3) and others for the pattern by combining certain results from one or more questions. For every combination, the parent statement is plotted on the x-axis or y-axis and then the bar plot is analysed for the segments (i.e. stacks on the bar plot) for the second/child statement. The combinations are analysed to spot the trend in the agile projects. For example, for meetings, the respondents answered the daily stand-up meetings questions but one may want to know whether they follow the retrospective meetings after the sprints. Since retrospective meetings are long duration meetings aimed to find the improvement areas to deliver better quality product next time. Multiple option questions are further analysed in the form of combination. For example, if the questions had 3 options, the combination would give 8 outputs but the data is visualised for the relevant combinations i.e. excluded combinations which had no data. Further, every figure has the formula/ technique used to reach to the visualization of the survey data which can be seen below the bar plots in each chart. These visualizations were essentially useful in analyzing the agile trend among the respondents. For example, for agile methods' trends (figure 5), the survey showed that the scrum is used by 91% of the respondents but after data visualization analysis it was found that the only scrum is used by 64.58% of respondents and rest is distributed as a combination of the scrum & Lean, Scrum & Kanban, and Scrum, Kanban & Lean as illustrated in the left-hand side graph below. The sprint duration did not follow any regular pattern and is contradictory to the interview's findings where it was stated that 15 weeks' duration is the industry standard. Please refer to the right-hand side chart in figure 5 for the sprint duration pattern among the respondents.

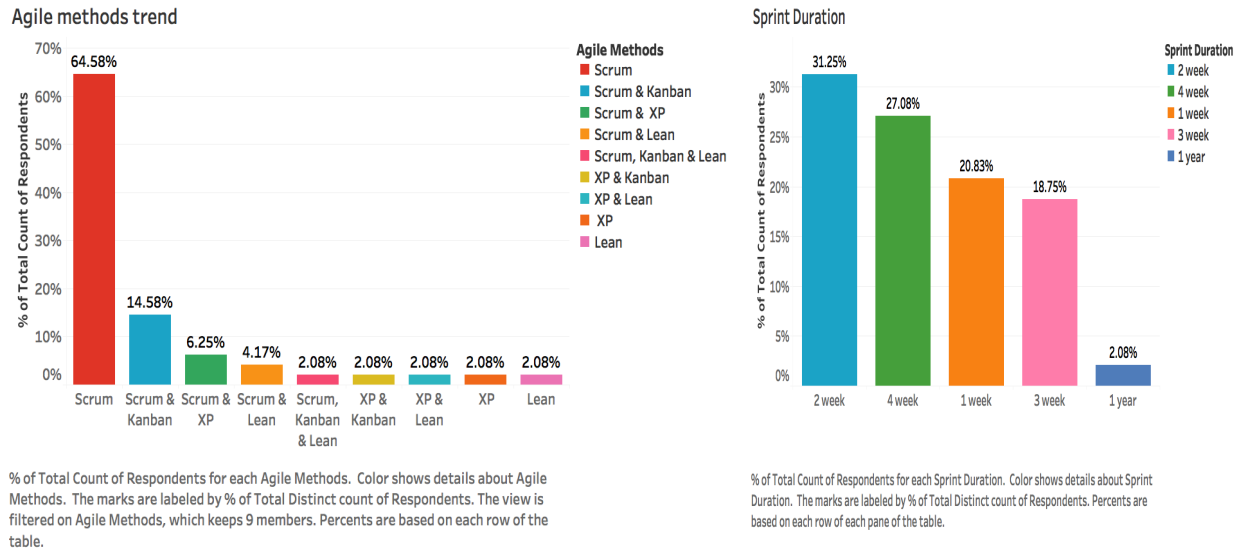

Figure 5: The left graph illustrates the various Agile method trends among respondents & the right chart shows the sprint duration followed among respondents

4.2.1. Cultural Change

The cultural challenge question was created using "Matrix/Rating Scale Question" to know the sentiment among the survey takers. Out of 48 respondents, 5 strongly agree, 17 agree, 15 are neutral, 6 disagree, 1 says it is not applicable and 4 did not answer the question to the statement "I feel culture issues are common" (refer figure 7). Although among 44 respondents, 22 agreed to the cultural issues and 6 disagreed to the challenge, but with only a few respondents disagreeing to the cultural challenge, the result suggests the culture issue as a challenge that people face in the agile.

4.2.2. Documentation

The documentation question was created using "Matrix/Rating Scale Question" where the respondents had to rate the question based on their experience. The survey asked the respondents how strongly they agree or disagree with the documentation being part of the agile. This question aimed to understand the generic feedback among the agile practitioners, what is the impact on their productivity and to know to which extend the documentation helps in information seeking. Only 44 respondents answered the question; the result showed that the 58.33% want documentation as a part of the agile practices and 66.67% completely agree that the documentation makes it easier to understand the code but 41.66% agree that the documentation consumes a lot of time (refer figure 6).

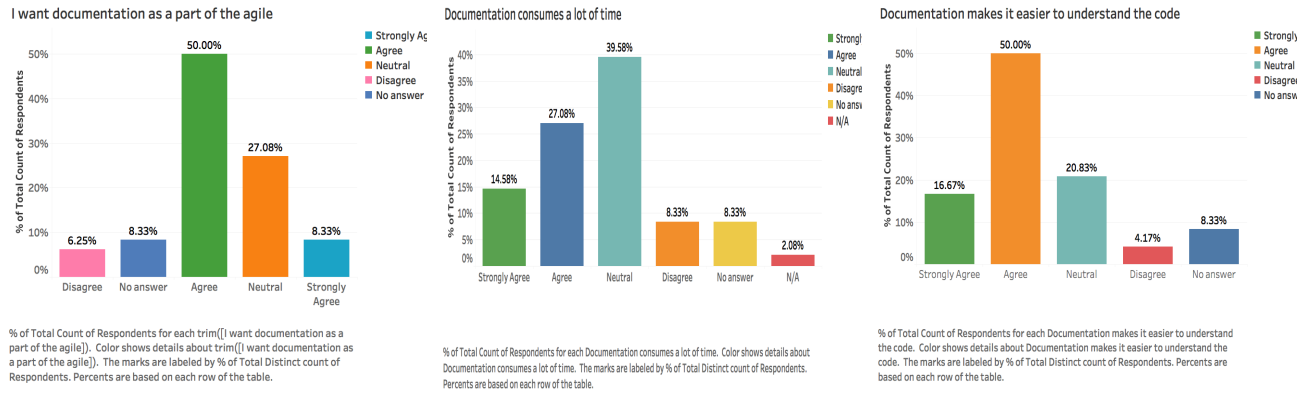

Figure 6: The first graph is the generic feedback on the documentation, the second graph is the feedback from the productivity perspective and the third graph illustrates the usability of the document in understanding the code in the decreasing order.

4.2.3. Team Dynamics

Only 44 respondents answered the matrix scale questions. These statements aimed to understand the opinion on the agile environment among the developers/testers. The questions were asked from many perspectives such as standard industrial practices for code, cultural issues, technical environment awareness, stress factor, sprint backlog, confidence to communicate their ideas in the open forum, agile satisfaction level and ability to adjust to the technical issues. These questions aimed to understand the various sentiment of the respondents and to understand the agile, technological and cultural environment awareness among the respondents. The individual responses captured are illustrated in absolute numbers in the charts in figure 7,8, 9 and 10.

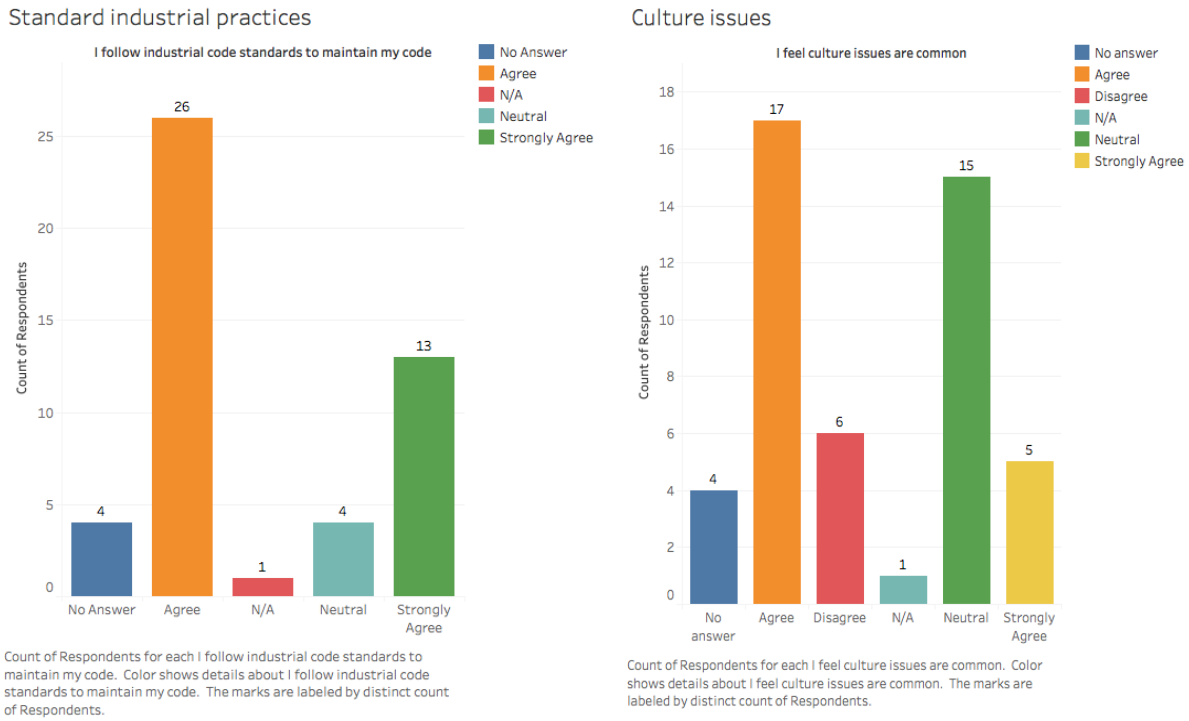

Figure 7: The left graph illustrates the usage of standard coding practices among the respondents for the daily tasks & the right chart shows the opinion on the cultural issues

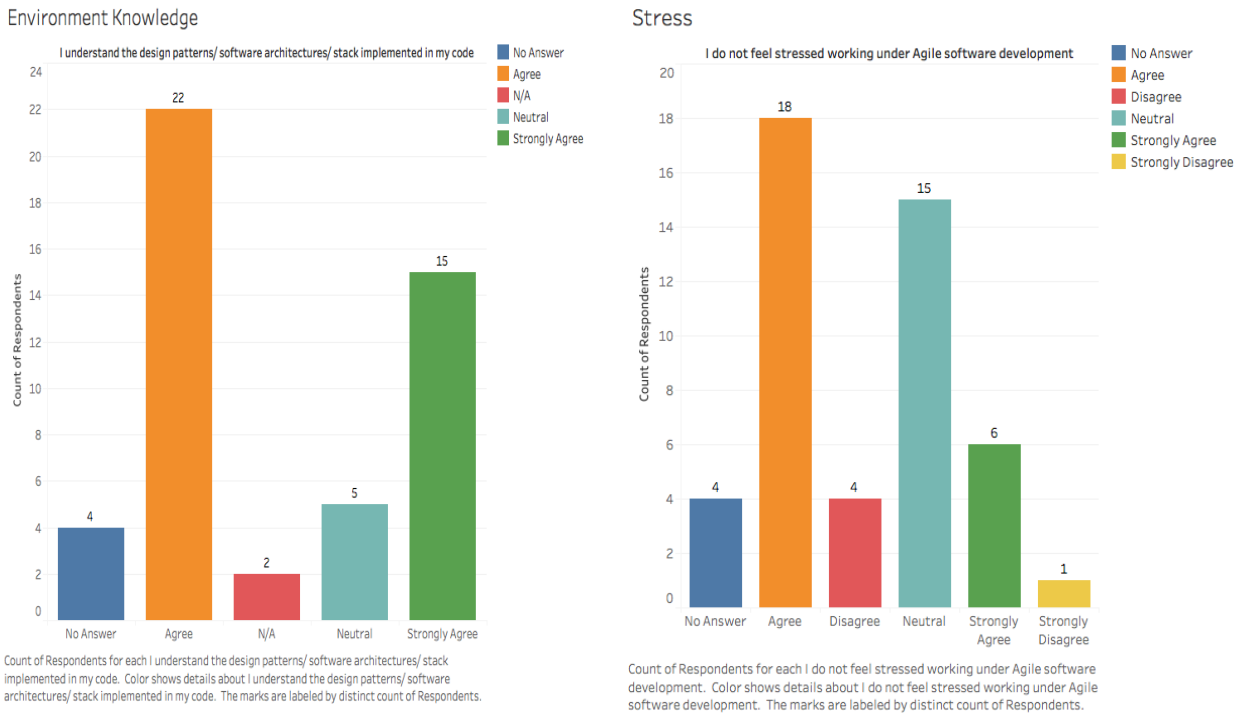

Figure 8: The left graph illustrates the technical environment awareness and the right chart shows the stress pattern among respondents

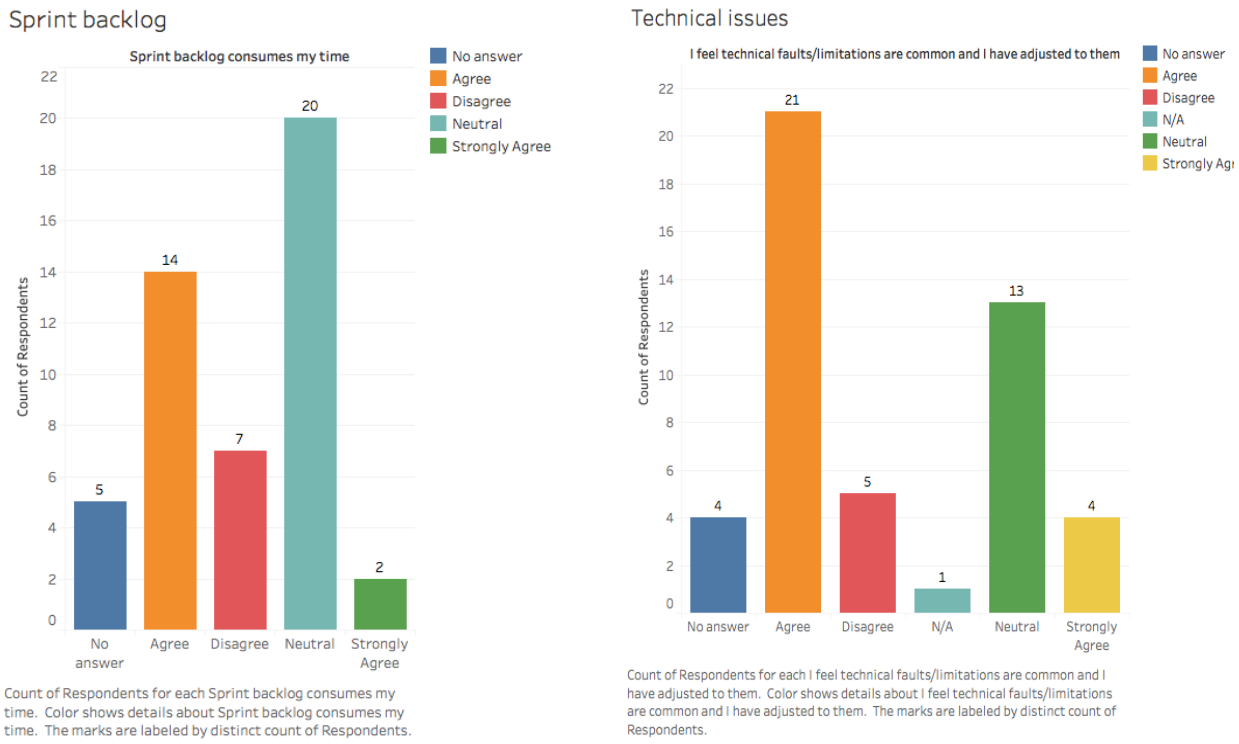

Figure 9: The left graph illustrates the feedback on the sprint backlog & the right graph illustrates the feedback on the technical issues

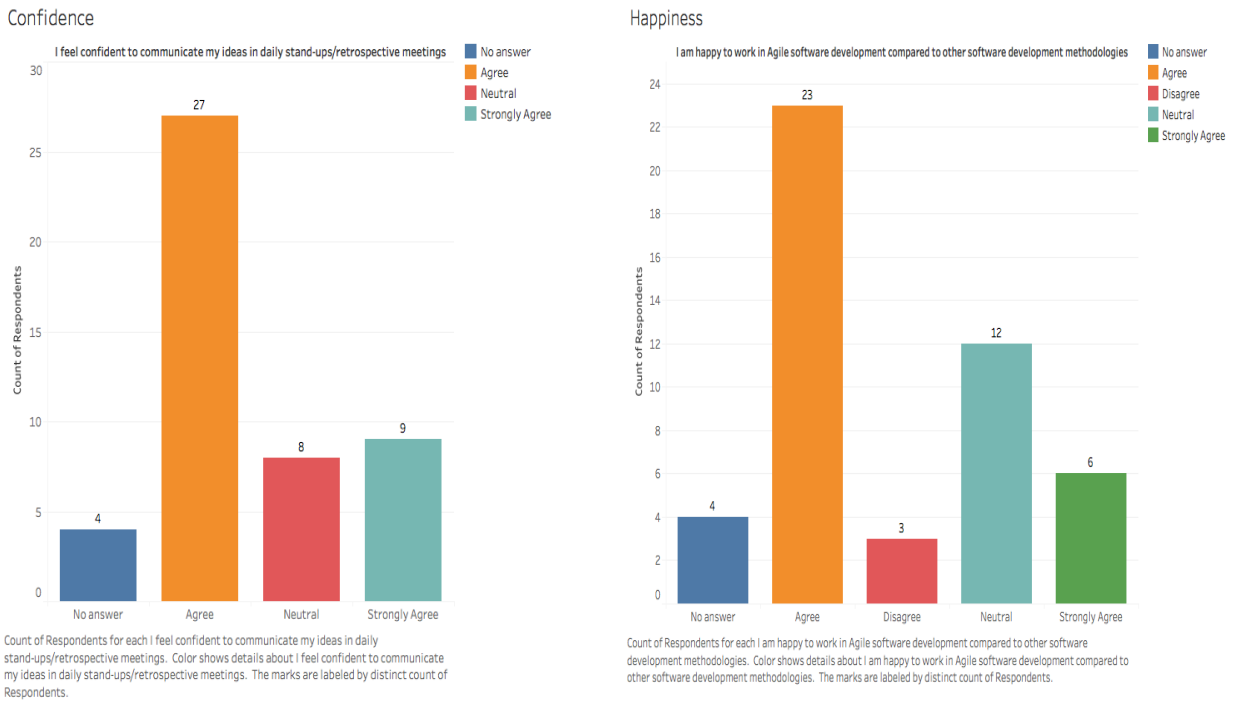

Figure 10: The left graph illustrates the confidence level of respondents in communicating their ideas & right chart shows the Happiness factor

4.2.4. Tools

Out of 44 respondents who answered this question, 41 agreed to use the versioning tools such as JIRA & GitHub for version controlling and 33 out of 47 respondents agreed to track the progress of the tasks on the visual board. The trend is illustrated in the left and centre graph in figure 11. Also, 44 out of the 48 respondents agreed that they have a central repository for their projects. To see if the respondents follow any other productive measures by incorporating vision board. The relation is drawn between central repository and vision board in right graph in figure 11 where one respondent checked that vision board is not applicable, but they use version tool whereas two respondents who did not provide an answer for the version tool says that they have a vision board in their project.

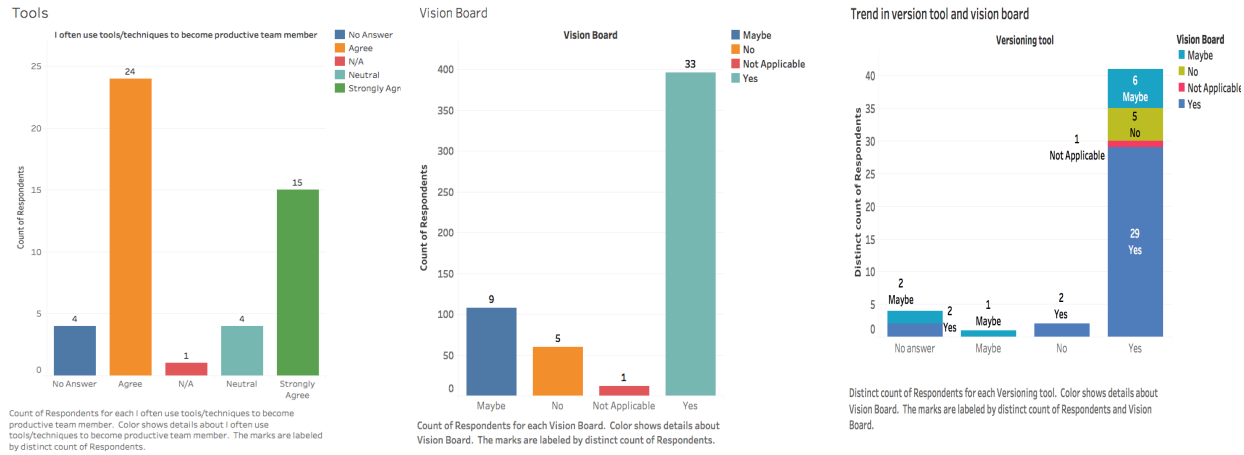

Figure 11: The left graph illustrates the regular usage of the tools used in their daily activities, the centre graph shows the vision board usage among the respondents and the right graph illustrates the relation between the version tool and vision board from the survey with version tool on the x-axis and the vision board stack on the bar plot.

4.2.5. Meetings

The respondents were asked to express their attitude towards daily stand-up meetings in four statements: "Daily stand-up interrupts my work", "Daily stand-up is necessary as they help me in solving the challenges", "Daily stand-up helps me to understand the big picture of the software product", "Daily stand-up is necessary". The results of the four statements were combined to analyse the patterns among the responses which can be seen in figure 12. Similarly, the respondents were asked to express their attitude for towards the retrospective meetings in four statements: "Every week we conduct retrospective meeting to focus on the backlogs", "Retrospective meetings help us achieve quality results", "Retrospective meetings are helpful for reducing the backlogs in next sprint", "Retrospective meetings are not useful". The results of the four statements were combined to analyse the patterns among the responses which can be seen in figure 13. Then the combined analysis is

performed to visualise the retrospective meeting pattern for the respondents who selected certain combination in the daily stand-up meeting question (see figure 14).

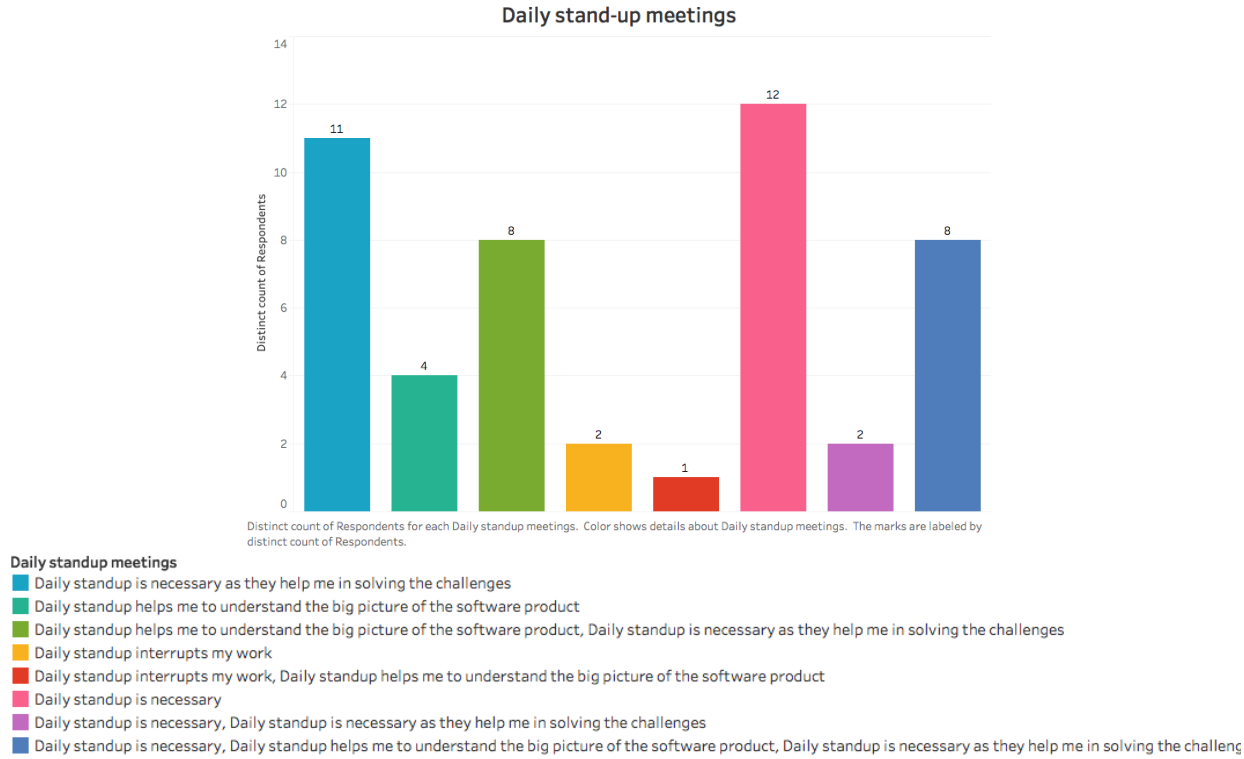

Figure 12: The graph illustrates the daily stand-up meeting pattern followed among the respondents

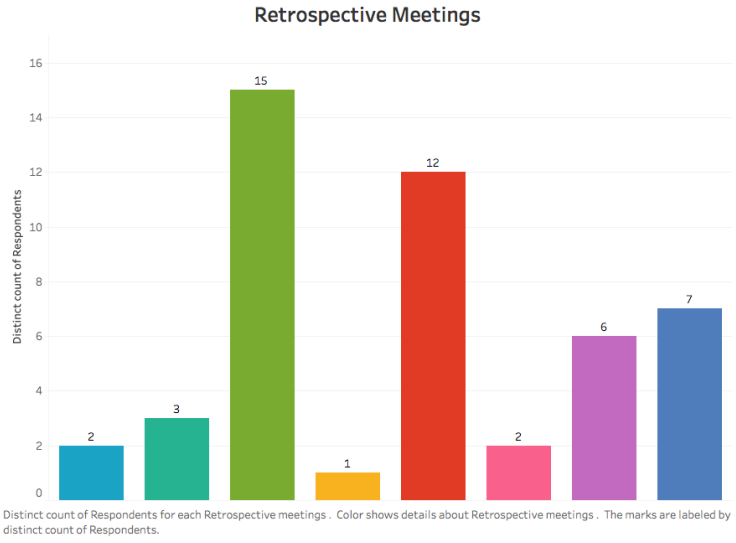

Figure 13: The graph illustrates the retrospective meetings pattern followed among the respondents

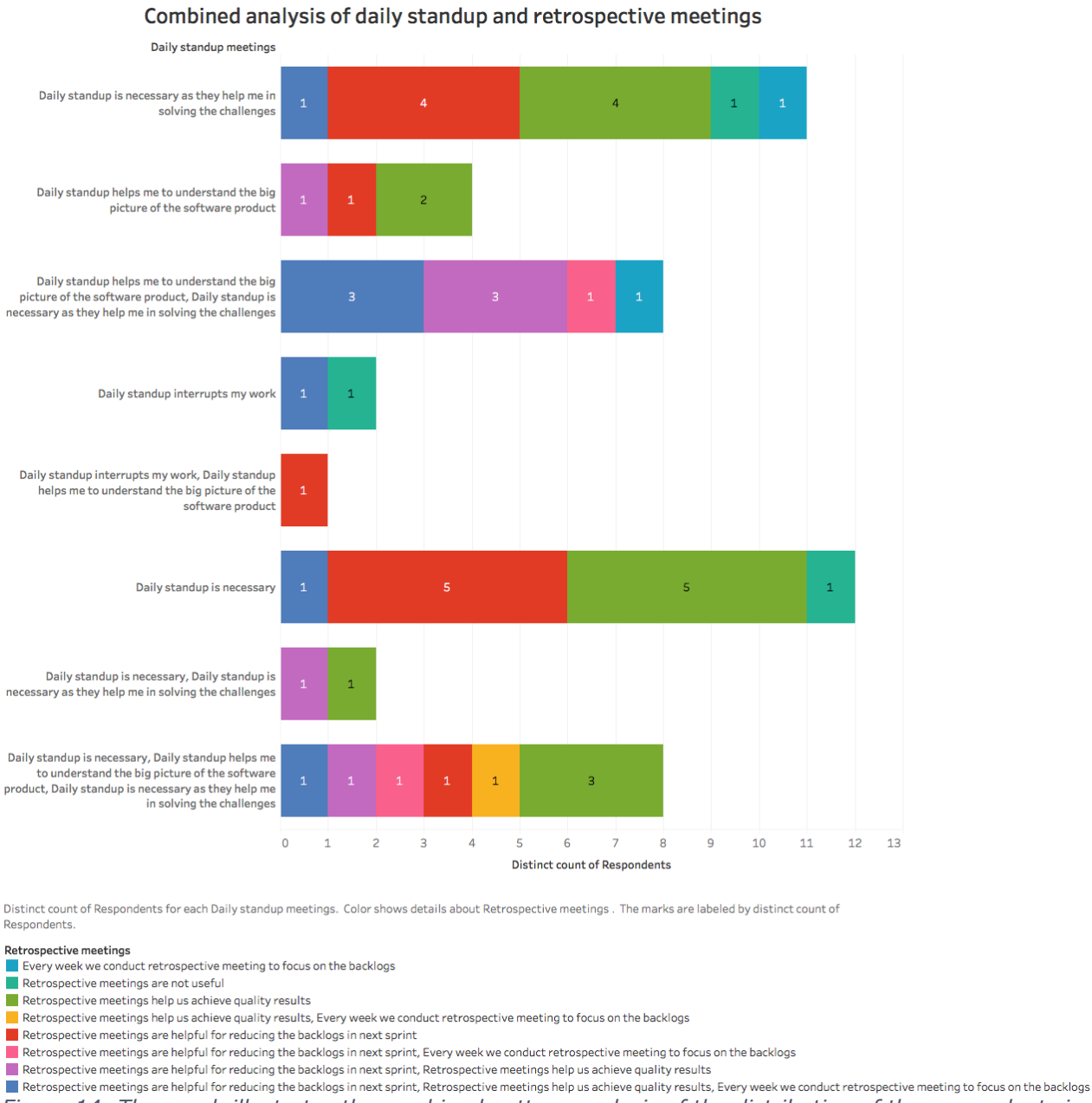

Figure 14: The graph illustrates the combined pattern analysis of the distribution of the respondents in the daily stand-up (bar plot) and retrospective meetings (stack on the daily stand-ups bars)

Two respondents chose to leave a comment (see below) for the daily stand-up question.

> *"Daily Stand-ups should be short and precise, otherwise they interrupt the work" – on daily stand-up meetings*

> *"Should be kept short though." – on daily stand-up meetings*

4.2.6. Transformation & Adoption

The survey asked the respondents to select the training process in their organisation during the induction phase, i.e. the training that is conducted when an employee joins a new project and the regular training trend in their organisation. In induction training question, the respondents were asked to select from "product", "code" "agile" and "not applicable" options. In regular training question, the respondents were asked to select from "tools", organisational training", "agile methods", "domain knowledge" and technical training. The respondents also had a choice to skip the question or leave a comment. The left graph in figure 15 illustrates the combination of the induction period training trend and the right graph shows the combination of the regular training trend among the respondents. Figure 16 illustrates the trend among the respondents who went through an induction training then categorised for the different regular training combinations.

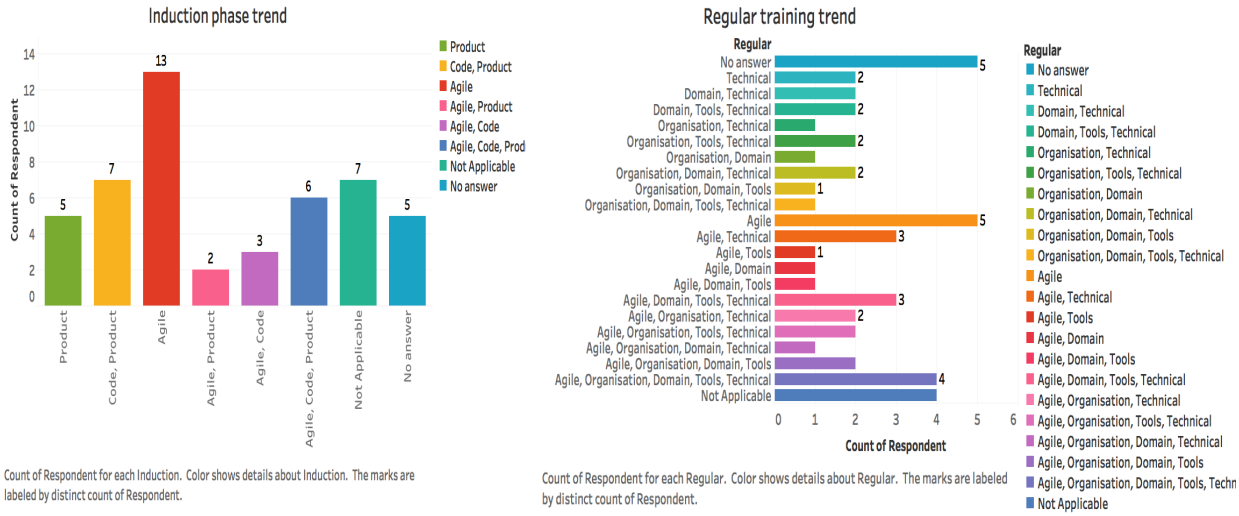

Figure 15: The left graph illustrates the training pattern during conduction phase, and the right chart shows the regular training pattern among the respondents.

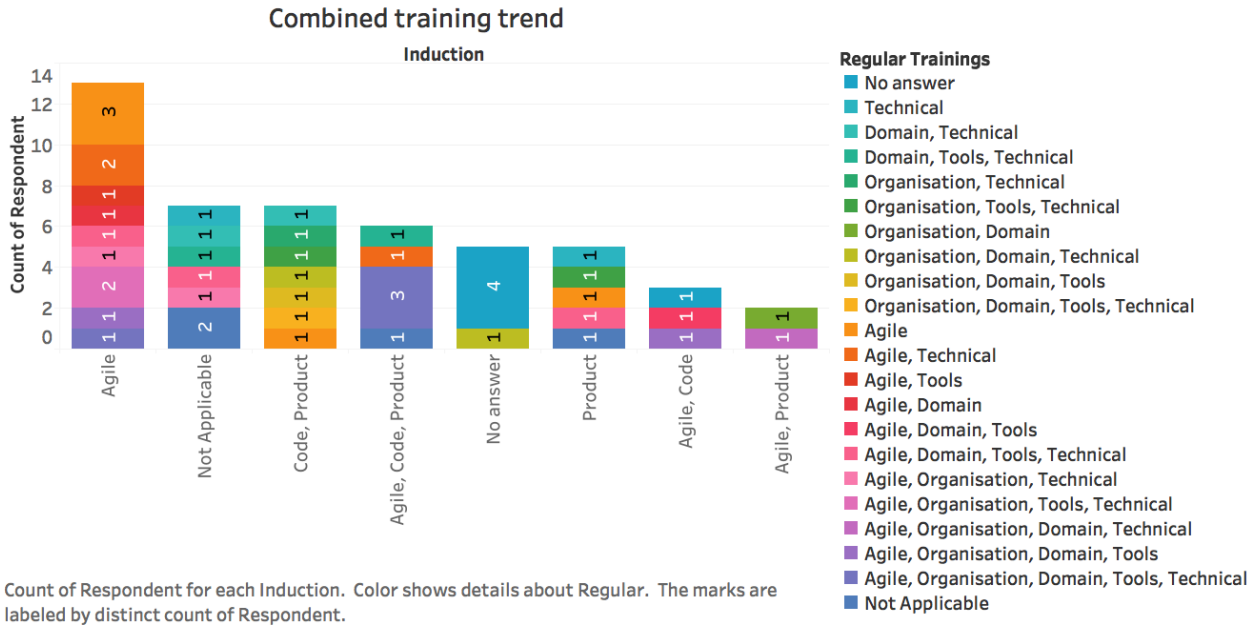

Figure 16: The graph shows the combined analysis of the training during induction phase and then the regular training stacks are distributed on the induction bar plot

One of the respondents stated that they do not have any regular training in their projects and other commented they do not have training.

"we don't implement any regular training" – on regular training

"no trainings" – on induction training

4.2.7. Agile coach

The survey asked the respondents if they have "internal/external "Agile Coach" in your organisation to help you adopt agile practices" but only 45.83% agreed to the agile coach at their work place. The trend is illustrated in figure 17. To analyse further, the responses from the opinion questions: "I am happy to work in agile software development in comparison other software development methodologies" and "I do not feel stressed working under agile software development" were analysed on the bar plot of the agile coach responses. The relation is illustrated in figure 18. The distribution of respondents who felt happy in comparison to the traditional approaches are present in the area where respondents chose 'No', 'Maybe' or 'I never came across the term agile coach' (illustrated in left graph of figure 18). Similarly, the stress statement is analysed on the agile coach bar plot in right chart of figure 18.

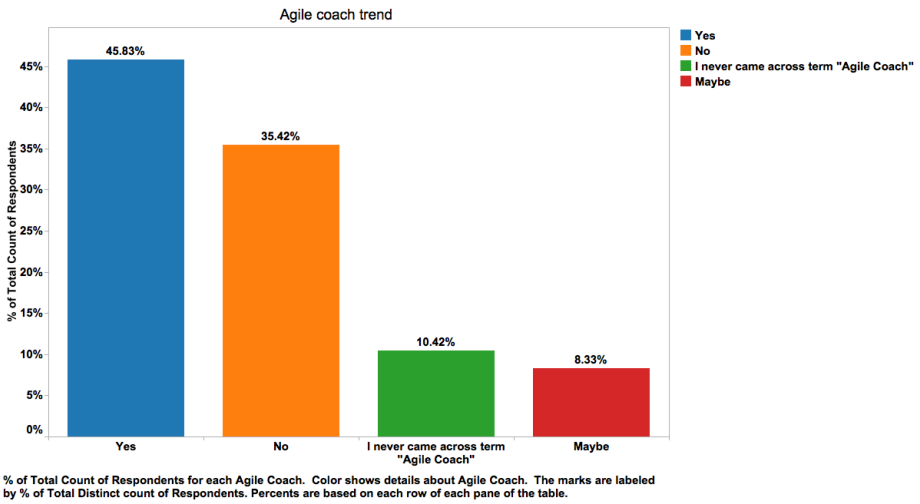

Figure 17: Agile coach trend among respondents

Figure 18: The left graph illustrates the relation between happiness factor and the agile coach. The right chart shows the relation between the stress and the agile coach.

4.2.8. Other Correlations

The bar graph is plotted for the statement "I am happy to work in Agile software development compared to the other software development methodologies" and the opinion segments are distributed for the statement "I do not feel stressed working under agile software development" on the bar plot to visualise the outliers. For example, 23 people are happy to work on the agile but there are 2 respondents who feel stressed under agile practices. Also, in neutral bar, there is one respondent who strongly agree that he/she does not feel stressed in the agile environment. As per the figure 19, one respondent is neutral on the former statement but strongly agrees with the latter statement that he/she does not feel stressed in agile.

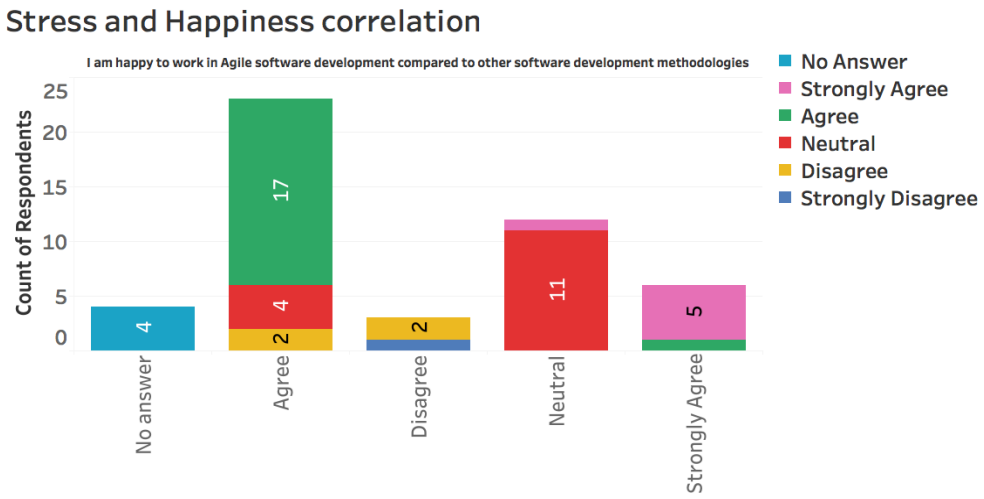

Figure 19: The graph illustrates the stress and happiness correlation from the survey

Further, all the matrix scale questions were analysed for the sentiment visualisation to see the likelihood for the various statements among the respondents which are illustrated in figure 20. As per figure, respondents are neutral on the statements "Sprint backlog consumes my time" and "Documentation consumes a lot of time" in high percentage. Interestingly, both statements have time factor in it and time management. Also, "sprint backlog consumes my time" has the highest number of negative responses. The last four statements have the highest percentage of the positive responses, and they were asked to know the technical environment awareness, confidence and self–motivation factors among the respondents. The trend on the stress factor in agile, happiness factor in agile and sentiments on documentation are also illustrated in the graph.

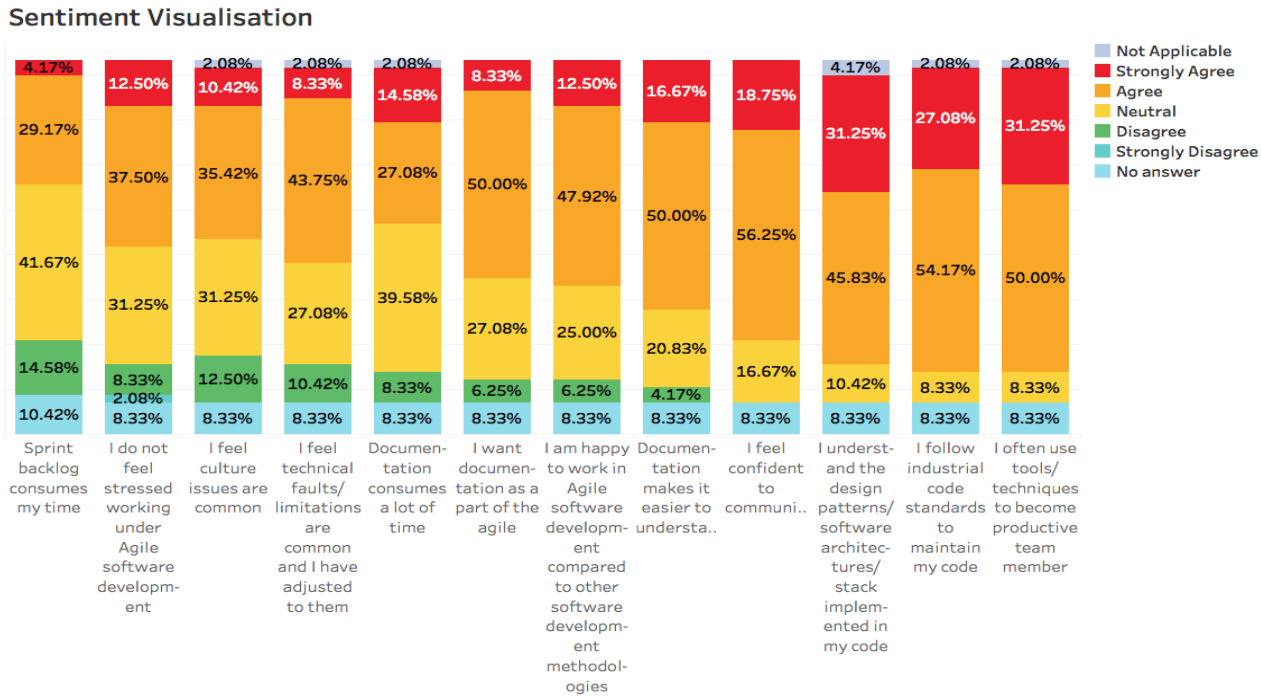

Figure 20: Sentiment visualisation for all the Likert scale questions on the survey

Apart from conducting objective choices, the survey asked for subjective answers at the end of multiple choice questions and in the end for further comments. 11 respondents chose to answer the further comments from which three elaborative answers are quoted below.

> *"Agile Methodologies eases out the work pressure if followed religiously or else it may create chaos."*

> *"I find it apt methodology as per the day to day requirements of industry"*

> *"1. Although daily meetings do consume time but yeah they are beneficial for the whole team to get an idea what everyone is working on. Plus, to check if we are on track. 2. I have never seen any company follow true agile in my 7 years of experience so far. 3. There should be a proper training on agile before you start working on it which usually doesn't happen."*

The survey findings section found some interesting revelations on the culture, documentation, agile coach, training, tools and self-motivation among the respondents. The findings from both interview and the survey are discussed in detail in the next section.

Chapter 5: Discussion

From the combined analysis of the interview and the survey, the cultural issues, documentation and the team dynamics were the challenges among the practitioners. It is revealed that in comparison to the traditional practices, agile methodologies are a popular contemporary choice among practitioners. Though, there are diverse perspectives on the stress in the agile methodologies where scrum masters stated that the developers feel stressed initially but later they adjust to the agile culture wherein survey suggested that only 11% agree with the statement that they feel stressed while working in the agile software development. It was also noted that the training and the agile coach trend is limited among the interviewees and the survey respondents. The agile practices in the industry is discussed in detail in this section focusing on the challenges mentioned in the literature review with the help of both interviews and the survey results.

The collective finding in both interview and survey is the cultural change and documentation is identical to the finding in the 11th annual state of the agile report. In VersionOne report, 63% respondents agreed that "the company philosophy or culture at odds with core agile values in the challenges experienced in adopting and scaling agile". While other factors, such as "lack of experience with agile methods", "lack of management support", and "general organisation resistance" etc. were next on the list with approximately 40% responses in each category and 34% agreed to the "insufficient training" (VersionOne, 2017). In comparison to VersionOne report, the survey 45.84% agreed that the culture issues are common and 58.83% wants documentation to be a part of agile (refer figure 20). Though these factors can further be analysed for the lack of experience on agile methods and management support since as per literature review, every challenge is getting impacted by many factors which can be seen in figure 1-4.

Further, in figure 19, the fusion of the happiness and stress factors revealed some interesting findings but collectively, it was important to see if the respondents who is happy to move from traditional development must not feel stressed in agile. Because the primary need of any project is to deliver the product on time as per the client's needs because it is the customer who brings business to the organisation. But one must not neglect the welfare of the employees to satisfy the customers because if the organisation takes care of the employees, then employees will take care of the customers (Walton & Huey, 1992). It is important to know if the agile practitioners are satisfied with the adoption & transition because if the team members are supportive of the

agile adoption, the issues identified later in the transition can easily be fixed if the team members have a positive mindset (Gandomani, et al., 2014).

The **cultural challenge** is not unique to only agile, it can be a roadblock to any change within the organisation, but the agile transformation is a big step to transition in software development (Sutherland, 2014). The change can affect at any level in the organisation which makes it difficult for the anybody to implement consistent innovation as many large organisations are sceptical to demonstrate the leadership demanded to change their legacy cultures and allow innovation to happen. It is possible that executives may face a challenge in transforming their legacy cultures, employees may lack the experience to make the transition (Denning, 2016). The obstacles to the adoption of agile methodologies are sometimes hidden under the organisational culture. Understanding the culture of a company is necessary to know it more sincerely (Waardenburg & Vliet, 2013). The perception of cultural levels can make the cultural analysis more efficient in the context of adopting an agile method. Sometimes, software team neglect the organisational success in favour of technical and personal achievements to deliver the product and sometimes, when people talk about the organisation's culture, they are referring to its dominant culture. Once an organisation/project/team identifies various cultures, then it is always easier to start mitigating the dominant culture first (Tolfo, et al., 2009). Though from the findings, it is difficult to find the dominant culture in the organisation because this study is not focussed on one organisation or project and to know the ruling culture, the study needs to be focused.

Even if the transition is completed, maintaining organisational change is difficult because the transformation is an ongoing process. To succeed at lasting transformation, one must constantly remind himself/herself the principles in the manifesto of what needs to be achieved. The organisation must examine and re-examine the culture when a member of the team is added; a team is moving to agile or client/process is at different locations, as Agile is never done, it’s a continuous journey (Maples, 2009). In the interview with Dutch banking group, ING, two executives shared their experience on the agile transformation where they stated that agile culture revolves around ownership, empowerment and customer centricity. ING's transformation journey was based on four pillars where first was agile itself. The second pillar was their organisation culture, third was the integration of the product development and IT integration with the help of DevOps, and fourth is the people model where the performance is rated at the organisation level instead of on the project level. Further, the culture change must be reflected in everything that one undertake as an organisation and as individuals (Mahadevan, 2017). Clearly, from the success stories of agile transformation,

it is important that one needs to regularly connect agile principles to the organisational principles repetitively to stay agile.

The **documentation** is agile is the most misunderstood area where people sometimes interpret that no or less documentation is one of the benefits of the agile. But ignore the fact of documenting relevant information can be beneficial when the team member is revisiting the code, then he/she does not have to devote excess hours searching for information in the sprint logs (Voigt, et al., 2016). Although, the question on a survey limits the answer to the likelihood of the scenario with the opinion based questions and with limited sample size (refer figure 6), it is hard to conclude what will happen if the study had bigger datasets because the sentiments can be distorted negatively or positively. Nevertheless, documenting vital information is crucial as it may be required in challenging times, but for that, one must identify at project level what information needs to be recorded as a reference. Also, one can opt for unit test case which serves as for the expected behaviour of the code as a reference for the developers (Stellman & Greene, 2015). For example, in interview, one scrum master contradicted his statements on the documentation (refer section 4.1.2). The documentation is a management that relays from employee to employee; it can be personal preference to tidy up everything and keep a record of every information. But it is important to log relevant information easily understood by the fellow colleagues' reference.

For effective **team dynamics**, regular training and workshops are important for the improved quality of the agile understanding since the transformation is a cultural change. A proper training can help in leveraging agile benefits and they have training when they feel it is required. However, the training that is focused only from the agile perspective can also prove a disadvantage when the newest member of the team is not just new to the agile but to the development and product knowledge as well. The organisation must not neglect other factors such as ability-related factors, motivation-related factors, and opportunity related factors, as they can potentially impact acceptance of agile methodologies among their systems developers. Besides educating developers on the characteristics of agile methods such as its usefulness, ease of use, compatibility, results in demonstrability, and maturity, organisations can also take initiatives such as providing adequate training, external support, etc. to improve the ability of their developers in using agile methodologies (Chan & Thong, 2009). Further, the organisations can also create a platform to develop a knowledge culture (e.g., teamwork, communication, shared understanding, interpersonal relationship) by establishing meeting rooms, rewarding those who pursue learning and who share knowledge with others what they know. The organisations must establish a set of cultural norms, processes, and practices to have outstanding performance.

Another important aspect of the agile is **cross-training** of the team members to manage the workload collaboratively and efficiently (Kim, et al., 2016). However, many executives believe that just creating more cross-functional teams necessitate the agile transformation, which is true, but the first step is to train the team to work together. Also, it is important in agile to have only one boss to approve team's decisions, commissioning a cross-functional team, and selecting the team because people cannot have multiple bosses which mean multiple decisions that can conflict or burden the team member. The important thing is to invest in agile teams since they are self-managed as they continually improve their collective performance by clarifying roles, exercising conflict resolution techniques, and assuring that team members contribute equally (Rigby, et al., 2016). Team dynamics to a vast extent is dependent on the behavioural understanding and effectively managing the disputes or status quo in time.

The Daily Stand-ups and Retrospective **Meetings** are the platforms where the team interacts, the team members are open for a conversation, and everybody is devoted to solve the challenges as well as to improve the quality of the product. Here, it is required that every team member focusses only on one topic i.e. the product. In the interview, one scrum master stated that it is possible that sometimes in daily scrum, people bring their personal agenda, or they arrive late for the meetings, but these behavioural issues are regulated by the scrum master because daily scrums are limited to 15 minutes and must be focussed on the product's progress. In extreme case, if the member's behaviour is eccentric, then it is handled in one to one meeting. The face-to-face communications and meetings serve as the strategy by Agile teams to dealing with problems both during transformation and after adoption. Also, management support, qualified people on agile, agile champions and agile coach are part of the agile process, but they act as facilitators for the agile transition and adoption (Zulzalil, et al., 2014). The twelve principles of the agile manifesto focus on the client and sustainable development process but the agile facilitators can impact the process positively. Essentially, the retrospective meeting is quality management technique followed in agile to improve the delivery process by brainstorming to eliminate the previously encountered errors and have a consistent delivery next time.

The **software tools** used as a part of the agile development process contains best features from the agile development and specifically developed from agile perspective to keep everything on track. The apparatus to record every activity in the agile eco-system can facilitate the management of agile software development projects (Davidov, et al., 2017). Almost every respondent agreed to use agile management system such as Microsoft's Team Foundation Server, Rally, VersionOne, or JIRA to track the progress. These

systems contain a roadmap for more than sprints while maintaining information of visual boards for tracking information, sprint logs, backlogs and burndown charts. Each Scrum master in the interview agreed to use one of the agile management systems but in the survey, 4.55% denied of using any tools to track the progress and 2.7% respondents were not sure. Although, 7% respondents who disagreed using the agile management system are required to be researched to track the complete process of the automatic builds, continuous integration, continuous bug tracking and continuous deployment to have the exact percentage of how efficiently the projects are utilising the capabilities of the software. Further, DevOps is a management system, a critical enabler of agile software development, incorporating culture perspective and extending the agile management process that automates the to maintain the flow of continuous integration, continuous delivery and continuous deployment processes but as per the study, nobody is currently using it (Bossert, et al., 2015).

Lastly**, stress** is a challenge that can be a problem area in any industry and any project management technique. In the study of well-being in agile teams, it was discovered that feeling of empowerment leads to better control of stress. But there was a little population who felt worse, and their performance decreased under agile (Laanti, 2013). Likewise, in this study, approximately 5 respondents agreed that they feel stressed and 15 respondents are neutral on the stress in the agile project management in the survey (refer figure 8). But on the contrary, in interview, scrum masters agreed to the fact that they often hear complaints about the stress in the team. Though, the sample size in the survey was greater than the sample size of the scrum masters. However, the scrum master responded on behalf of their team which again contradicts the scenario since he/she was responding on behalf of the larger sample: their team. In the interview, one scrum master stated that the scrum team regularly interact in the form of daily stand-ups, pair programming and team programming which enables lagging member to reach at the same level as others. Similarly, in a study by (Lindsjørn, et al., 2016) it was stated that the team work is fostered by effective communication, mutual support by helping other team members when they are stuck, common understanding when working on subtasks and willingness to share workload. This could be counted as a reason of reduced level of the stress in the agile team but this can be further researched in comparative study of agile versus traditional project management techniques.

As per (Drury-Grogan, et al., 2017) study, in the agile eco system, **agile champion** and **agile coach** are main influencers in promoting agile. In findings, though there was no direct reference to the agile champions in the team but people agreed to the presence of the agile coach. In fifth interview, the scrum master stated that the internal agile coach might prove beneficial

for the team because sometimes product owner has no solutions and we need to contact client for the solution. In the survey, 35.4% respondents do not have agile coaches, and 10.4% haven't heard a term called agile coach (refer figure 17). Also, the happiness and stress factor in figure 18 could not reveal any pattern which can conclude the importance of agile coach. Though in opinion questions, many respondents are neutral to a very large extent. Perhaps, the agile coach could have helped them understand the transition which is also a big cultural change (Parizi, et al., 2014). Considering the factors that can affect an individual in the agile management, one can opt for the agile coach who can help to bridge the gap and eliminate ambiguity related to the processes. However, any change can take considerable amount of time to reflect the transformation. When it comes to the agile, the steps are simple but the challenge is to align the agile technique with the IT operations local to the project to deliver the software. It can impact not only an individual team member but also a team and the factors can vary from project from to project. Here the agile coach's consultancy can help mitigate a situation.

Chapter 6: Limitations and Conclusion

6.1. Limitations

1. Due to lack of time, data is collected from few stakeholders which limits the sample size and restricts the analysis on few topics such as tools and agile coach.
2. The survey is carried to capture the generic satisfaction level among agile practitioners and do not distinguish among small scale, medium scale and large scale projects.
3. The study has not focussed extensively to segregate the satisfaction among the agile practitioners in hybrid agile methods because of the limited available data in each category.
4. The study presents the analysis captured at one point of time.

6.2. Conclusion

The study intended to know the agile environment by understanding the agile concepts, to what extent the agile practitioners are implementing the agile principles, how the scrum masters and the team perceive the challenges faced in agile projects and how satisfied they are with the agile practices. To understand the concept of agile, the paper reviewed the history of the agile methodologies to review the agile concept then the paper examined previous works for the challenges faced in the agile practices and finally examined the role of the agile coach in the agile projects. Further, the interviews and surveys were conducted to know how well the industry perceive the agile principles, what is the outlook on various challenges reviewed in the previous works and what is the role of the agile coach in bridging the gaps. From the interview, it was revealed that the cultural change, documentation and team dynamics pose a challenge for the agile practitioners. From surveys, the developers/testers' perspective is brought into the analysis to reveal the interesting behaviours among the practitioners apart from the validating the challenges. Despite challenges, majority of the respondents are happy to work in agile, feel confident while communicating their ideas in meetings and do not feel stressed in the agile projects. The future research areas are endless as every challenge in the study comprised of the multiple factors expanding into the sub area that can impact the satisfaction among the practitioners. Further, the impact of the agile coach is another area which can be researched to know in which cases the agile coach is required in the projects or organisations.

Appendices

APPENDIX 1: **Interview Questions**

1. Your client is from which sector?
2. How often do you have the meeting with clients?
3. How often do you have the daily stand-up meetings? Do you adhere to the daily stand-ups routine in stressful time?
4. How do you decide the Sprint duration?
5. How much difference do you notice in agile in comparison to other software development method?
6. How do you accommodate to request the change?
7. What are the challenges faced in agile development?
8. What tools do you use to monitor the progress?
9. Do you suggest any practices that you follow for every agile project?
10. How difficult was the agile transition for you? What were the challenges faced?
11. How do you manage actual and expected burndown?
12. Do you use agile management systems such as JIRA, DevOps to organise the software delivery process?
13. How do you perform version controlling in the agile software development?
14. Do you maintain code on the cloud?
15. How do you maintain to get consistent performance from the developer considering everybody's Emotional Quotient and Intelligence Quotient level is different?
16. How often do you conduct workshops for behavioural issues to enhance team dynamics?
17. How often must you provide training to the team on the technical and functional aspect of the project?
18. Do you think that the technical, functional or behavioural issue interferes with the behavioural issues? If yes, how do you manage the delivery process?
19. How do you manage to interact with the client when the client is not working in the same time zone?
20. Do you have Agile Coach in the project? If yes, how do you plan the visit?
21. Which of the agile methods do you follow from the Kanban, Scrum, XP and Lean? How do you implement the fusion of the methods?
22. Do you have any visual board to track the completed task?
23. Do you consider it interesting?
24. Do you consider the role of the agile coach is important to bridge the gaps since agile is new to other project management techniques?

APPENDIX 2: **Survey Questions**

a. Please select the appropriate sector you are currently working for client. Check all that applies**.**
b. How long have you been working in the agile project?
c. Please enter the team size in number.
d. How long have you been using agile methods or agile practices?
e. How many agile methodologies do you use in your project?
f. What is the sprint duration in your project?
g. Do you have internal/external "Agile Coach" in your organisation to help you adopt agile practices?
h. Do you think "Daily Stand**-**up" are necessary to know roadblocks or to keep things on track?
i. Please select relevant option in the case of the retrospective meetings.
j. We maintain the visual board in the project to track the progress of the tasks (example: maintaining Work in progress, to-do, completed, pending etc. on the board for every team member)
k. Do you use any versioning tool such as JIRA or GITHUB to maintain all the copies of the code?
l. Please select the training sessions you have had when you joined the team.
m. Please select all the options that are part of regular training curriculum in your organisation.
n. Please provide your response to the statements below

i. I want documentation as a part of the agile
ii. Documentation consumes a lot of time
iii. Documentation makes it easier to understand the code

o. Please provide your response to the statements below

i. I follow industrial code standards to maintain my code
ii. I understand the design patterns/ software architectures/ stack implemented in my code
iii. I often use tools/ techniques to become productive team member

p. Please provide for the responses for the statements below

i. I do not feel stressed working under Agile software development
ii. I am happy to work in Agile software development compared to other software development methodologies
iii. Sprint backlog consumes my time
iv. I feel technical faults/limitations are common and I have adjusted to them.
v. I feel culture issues are common
vi. I feel confident to communicate my ideas in daily stand-ups/retrospective meetings

APPENDIX 3: **Twelve principles of the Agile Manifesto**

#	Principles
1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4	Business people and developers must work together daily throughout the project.
5	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7	Working software is the primary measure of progress.
8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9	Continuous attention to technical excellence and good design enhances agility.
10	Simplicity--the art of maximizing the amount of work not done--is essential.
11	The best architectures, requirements, and designs emerge from self-organizing teams.
12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Source: (Beck, et al., 2001)

APPENDIX 4: **Lean Principles**

Core Values	Description
Eliminate waste	Find the work that you're doing that doesn't directly help to create valuable software and remove it from the project.
Amplify learning	Use feedback from your project to improve how you build software.
Decide as late as possible	Make every important decision for your project when you have the most information about it — at the last responsible moment.
Deliver as fast as possible	Understand the cost of delay, and minimize it using pull systems and queues.
Empower the team	Establish a focused and effective work environment, and build a whole team of energized people.
Build integrity in	Build software that intuitively makes sense to the users, and which forms a coherent whole.

See the whole	Understand the work that happens on your project — and take the right kind of measurements to make sure you're actually seeing everything clearly, warts and all.

Source: (Stellman & Greene, 2015)

APPENDIX 5: **Agile Resources**

Agile Alliance (https://www.agilealliance.org/) For guides to agile practices, links to "The Agile Manifesto," and training videos.

Scrum Alliance (https://www.scrumalliance.org/) For a "Scrum Guide," conference presentations and videos, and the "State of Scrum" research report

ScrumLab Open (https://www.scruminc.com/scrumlab-open/) For training presentations, videos, webinars, and published papers

Annual State of Agile Survey (http://stateofagile.versionone.com/) For key statistics such as usage rates, customer benefits, barriers to adoption and success, and specific practices used

APPENDIX 6: **Interview Transcripts**

Interview 1

Q1. ***Your client is from which sector?***
Healthcare

Q2. ***How often do you have the meeting with clients?***
Daily, through meetings. Face to Face - Yearly Once. In general, they travel to India for two weeks and spend time with all the team.

Q3. ***How often do you have the daily stand-up meetings? Do you adhere to the daily stand-ups routine in stressful time?***
The daily stand-up call lasts for 15 mins, max. Yes, we will have the daily stand-up call, during heavy work load too, without fail.

Q4. ***How do you decide the Sprint duration?***
Sprint duration will be decided by a number of releases in a year. If the client expects a release for every two months, then we usually have two weeks' sprint. Sprint duration is also depending on the complexity of requirements that are going to be developed. If they are complex, then most of the scrum

team will go with two weeks' sprints. If they are simple, then one week sprint also works.

Q5. ***How much difference do you notice in agile in comparison to other software development method?***
Using Scrum / Kanban/ or Agile methodologies, we can deliver minimum viable product to the customers, with in short span of time (once in every two months) …there by, the business can see return on investment quickly, as customers will be happy with availability of new functionalities, with in less time frame. In normal traditional waterfall / v- model software development methods - We used to wait years to get all the functionality implemented and release to the production (go live), at one go. That too, we are not sure whether all the implemented functionality will work as expected at the end of the phase. This is time/money waste, and the customer will not be happy, if they can't see the fruit, after waiting a long time. Whereas Scrum/Agile, things will be decided in short span of time (release for every two months). Whether the implemented functionality is working or not. If that does not work, then it will be fixed in next couple of days or during next release.

Q6. ***How do you accommodate to request the change?***
Agile recommends to allow/accept last minute changes if it is important to get deliver during that release. Provided, a team has enough band width to take-up that change. If not, then the team will re-prioritise their work by replacing least important requirement with this new change.

Q7. ***What are the challenges faced in agile development?***
The biggest challenge is Cultural Change. Employees must get trained properly / brainstorm before they start working in the agile environment. It is not easy for the employees who worked in traditional projects and moved to agile, suddenly. They will feel that they are getting pressurised for deliverables on the daily or weekly basis. Demotivated most of the times. As a Scrum Master, we must guide/encourage them with Agile /Scrum Principles which are to be followed, as part of this cultural changes. Sometimes, clients will pressurise developers to take more work and deliver more. This is not fair, as agile is against to that. Agile states that 'Let team decides how much they can commit and how much they deliver in the sprint."

Q8. ***What tools do you use to monitor the progress?***
JIRA or Version One

Q9. ***Do you suggest any practices that you follow for every agile project?***
Agile Methodologies can be tailored based on the situation. In general, we all start with basic guidelines/principles given by Agile…and tailor per our needs in that project, keeping fundamentals are in place. For example, Backlog

grooming Meeting will be conducted once in every sprint. But, sometimes, we will have that meeting twice in a sprint, if more requirements (User Stories) are there to discuss OR complexity in understanding requirements (User Stories).

Q10. ***How difficult was the agile transition for you? What were the challenges faced?***
Agile is a cultural change, all together. If it is the product based company, then I recommend implementing Agile top to bottom. Instead, it implements it partially. Implementing partially results in cultural differences (difference in opinions / stress etc. between employees (agile / non-agile).

Q11. ***How do you manage actual and expected burndown?***
Hours Burndown: Team must burn/consume hours on a daily basis, before they leave for the day. Then actual and expected burndown will be in sync.

Q12. ***Do you use agile management systems such as JIRA, DevOps to organise the software delivery process?***
DevOps - It can be used only when the organization has sufficient tools to implement it. For example, Continuous build / Continuous integration / Continuous testing / Continuous Delivery/Continuous Deployment. If an org is good at least one are of the above, then they think about implementing DevOps solutions to reduce the time to market and get benefited with in short time span.

Q13. ***How do you perform version controlling in the agile software development?***
Using GIT or Selenium tools

Q14. ***Do you maintain code on the cloud?***
We are not yet moved to the cloud. Can't answer it.

Q15. ***How do you maintain to get consistent performance from the developer considering everybody's Emotional Quotient and Intelligence Quotient level is different?***
Emotional Quotient (EQ) plays a very important role in extracting good output from the Developer. There are several ways to handle it. Intelligent Quotient (IQ)- It can't be changed by Scrum Master.

Q16. ***How often do you conduct workshops for behavioural issues to enhance team dynamics?***
There will be workshops at the organizational level to handle this kind of issues. We recommended employee to undergo those training. It happens monthly once.

Q17. ***How often must you provide training to the team on the technical and functional aspect of the project?***
Scrum/Agile - Team must learn while working. In general, we provide/conduct technical training, once in every two months.

Q18. ***Do you think that the technical, functional or behavioural issue interferes with the behavioural issues? If yes, how do you manage the delivery process?***
Sometimes, Yes. Again, it's a perception / experience from one scrum team to other scrum teams. Technical / Functional issues can be figured out and be fixed, by taking help from other team mates or pulling Subject Matter experts to support the team, temporarily. But, behavioural issues- One of the reason could be the cultural change, as agile expects continuous delivery/output on end of each day...so, the developer who is new to this agile must face to some struggle during the initial period, until he gets acquainted with the process. If any other behavioural issues - it will be handled by management - will look for a replacement for that developer.

Q19. ***How do you manage to interact with the client when the client is not working in the same time zone?***
We plan in such a way that client / development team will have proper handshake (4 hours), on a daily basis. For example, USA and India. The offshore team will share all the info through meetings during those 4 hours and then leave for the day. Rest of the time, communication happens through emails.

Interview 2

Q1. ***Your client is from which sector?***
Legal

Q2. ***How often do you have the meeting with clients?***
Internal clients 2-3 times a week, external clients around once a year

Q3. ***How often do you have the daily stand-up meetings? Do you adhere to the daily stand-ups routine in stressful time?***
I organise all scrum related catch-ups and meetings. We always do the daily scrum.

Q4. ***How do you decide the Sprint duration?***
Team decision. Our default is two weeks unless there is a good reason to change this. Sprint 0s are often shorter.

Q5. ***How much difference do you notice in agile in comparison to other software development method?***
Significantly increases the chance of delivering a product the client needs, and increases their satisfaction. Less significant issues occurred compared to waterfall.

Q6. ***How do you accommodate to request the change?***
We don't need to. As part of the scrum framework, we do not commit to anything more than the next sprint, and the content for that is frozen, and cannot be changed. Regarding the longer-term planning, we offer guidance and estimates but do not commit to delivery. As such, the client is always welcome to change his or her mind.

Q7. ***What are the challenges faced in agile development?***
Developers do not like the additional meetings early on, especially the honest nature of retrospective and regular stand-ups. Senior management does not like receiving nothing more concrete for planning that estimates and support - they want levels of commitment regarding scope and timeframe that we cannot provide. Scrum requires a genuine cultural change that not everyone/everywhere is ready for or happy with.

Q8. ***What tools do you use to monitor the progress?***
Visual Studio Team Services primarily, and notes/drawings on walls/white boards

Q9. ***Do you suggest any practices that you follow for every agile project?***
Yes - scrum is the framework rather than a process so does not dictate exactly how everything should be done. Therefore, it is important to find what works for your team - for example. We combine the demo and retro sessions into a single meeting, but planning is the next day to ensure productivity. We also ensure regular backlog grooming sessions and have approaches to support product owners. We run a fortnightly session to look at our practices and improve them

Q10. ***How difficult was the agile transition for you? What were the challenges faced?***
Depends on the nature of the organisation. Developers need to be bought in, as do senior managers. They need to understand why we do things like this and benefits they bring, or there will be resistance. Let the dynamic team form; not everyone needs to be vocal and a leader. Support leadership in long term planning with estimates and limited commitments but don't make promises on behalf of the team they cannot keep. Avoid sprints become mini waterfall sessions - get things into the test as soon as possible.

Q11. ***How do you manage actual and expected burndown?***
We track through VSTS, as above. Keeping this accurate can be a challenge

Q12. ***Do you use agile management systems such as JIRA, DevOps to organise the software delivery process?***
We use Scrum where appropriate, but have several projects/BAU pieces that are not appropriate.

Q13. ***How do you perform version controlling in the agile software development?***
VSTS

Q14. ***Do you maintain code on the cloud?***
Yes, VSTS

Q15. ***How do you maintain to get consistent performance from the developer considering everybody's Emotional Quotient and Intelligence Quotient level is different?***
Peer review and regular communication. We attempt to allow people to take tasks to which they are best suited whenever possible.

Q16. ***How often do you conduct workshops for behavioural issues to enhance team dynamics?***
We discuss these issues at retrospectives where possible.

Q17. ***How often must you provide training to the team on the technical and functional aspect of the project?***
Not as regularly as we would like

Q18. ***Do you think that the technical, functional or behavioural issue interferes with the behavioural issues? If yes, how do you manage the delivery process?***
I'm not entirely sure what this means. A huge amount of reasons can affect/create behavioural challenges. It is my role as scrum master to coach and helps people overcome these. Everyone reacts differently, and the scrum master must be good enough for people to understand this.

Q19. ***How do you manage to interact with the client when the client is not working in the same time zone?***
We don't have this challenge at the moment, but I have done in the past. Ultimately, it's the responsibility of the PM to ensure the client is communicated with how they want to be - I have in the past woken up at 6 am every day to send status reports. If necessary, it's part of my job.

Interview 3

Q1. ***Your client is from which sector?***
HVAC

Q2. ***How often do you have the meeting with clients?***
Once in 2 weeks

Q3. ***How often do you have the daily stand-up meetings? Do you adhere to the daily stand-ups routine in stressful time?***
Daily stand up is strictly for 15 mins, unless there are items that require immediate attention and take more time

Q4. ***How do you decide the Sprint duration?***
I take the industry best practice of 15 days as my benchmark.

Q5. ***How much difference do you notice in agile in comparison to other software development method?***
The agile method gives more value to the clients in a lesser time frame, but at the same time is harsh on the development team because the timelines become very strict and pressure rises exponentially.

Q6. ***How do you accommodate to request the change?***
Can you please clarify more on this question?

Q7. ***What are the challenges faced in agile development?***
1) Strict timelines, 2) Micro management of development team, 3) Customer demo almost every month

Q8. ***What tools do you use to monitor the progress?***
Version 1/JIRA

Q9. ***Do you suggest any practices that you follow for every agile project?***
Yes, I do, and try to include best practices from the industry as well.

Q10. ***How difficult was the agile transition for you? What were the challenges faced?***
NA

Q11. ***How do you manage actual and expected burndown?***
I manage it through Version 1/JIRA tools. I keep track of this daily and ensure that whatever people are working on, they reduce them To Do hours daily from that object.

Q12. ***Do you use agile management systems such as JIRA, DevOps to organise the software delivery process?***
No

Q13. ***How do you perform version controlling in the agile software development?***
The development workbench has inbuilt functions to cater to the version management

Q14. ***Do you maintain code on the cloud?***
Yes.

Q15. ***How do you maintain to get consistent performance from the developer considering everybody's Emotional Quotient and Intelligence Quotient level is different?***
You must instil in them the confidence that they can achieve any task no matter how difficult it is, till the time they have the technical expertise. A lot of coaching is required.

Q16. ***How often do you conduct workshops for behavioural issues to enhance team dynamics?***
NA

Q17. ***How often must you provide training to the team on the technical and functional aspect of the project?***
As and when am required to give training. No timeline for this.

Q18. ***Do you think that the technical, functional or behavioural issue interferes with the behavioural issues? If yes, how do you manage the delivery process?***
Yes, it does. I have seen people with excellent functional/technical knowledge messing up things because of their negative/bossy attitude. You need to make them understand that they need to work in/with a team/set of people and they aren't working in isolation. Hence their bad behaviour is going to affect them as well as the team environment which will not be beneficial to anyone in the long run.

Q19. ***How do you manage to interact with the client when the client is not working in the same time zone?***
Set up meeting on a common agreed upon the time which is convenient to both the parties.

Q20. ***Do you have Agile Coach in the project? If yes, how do you plan the visit?***

No Agile Coach in my project.

Q21. ***Which of the agile methods do you follow from the Kanban, Scrum, XP and Lean? How do you implement the fusion of the methods?***
Only scrum.

Q22. ***Do you have any visual board to track the completed task?***
Yes, in our company we have a tool called Version One where we track the progress of the backlogs in each sprint

Q23. ***Do you consider it interesting?***
Not sure how interesting it is. But yes, its harsh on the project team. Too much of micro-management.

Q24. ***Do you consider the role of the agile coach is important to bridge the gaps since agile is new to other project management techniques?***
That's right, but I have now moved beyond the role of Agile Coach to that of the Scrum Master. And coming from a technical background, I fully empathise with the development team in the sense that they are under constant pressure to develop small software packages and demo to the client every 3-4 weeks, and at the same time continue with the respective backlogs too. So it's a very high-pressure environment they are working in

Interview 4

Q1. ***Your client is from which sector?***
Finance/Healthcare

Q2. ***How often do you have the meeting with clients?***
Sales group contact the client, and we get the requirements for the development

Q3. ***How often do you have the daily stand-up meetings? Do you adhere to the daily stand-ups routine in stressful time?***
Yes, we organise daily stand-ups for 10-15 minutes and we adhere to them daily meetings as they help assess roadblocks.

Q4. ***How do you decide the Sprint duration?***
The industry practice of 15 days.

Q5. ***How much difference do you notice in agile in comparison to other software development method?***
Huge difference in delivery but it stresses out team.

Q6. ***How do you accommodate to request the change?***
User guides are updated, developer documents aren't maintained as that may slow down the development as every requirement might not need strict documentation, system architecture is maintained

Q7. ***What are the challenges faced in agile development?***
Team members working individually which might slow them down. Code sharing, libraries, templates, code is not written very well, productivity is lost when the developers had to do reverse engineering on code which was written by other developers, refactoring can be done.

Q8. ***What tools do you use to monitor the progress?***
Agile Central

Q9. ***Do you suggest any practices that you follow for every agile project?***
We try to restrict team size to 10 where developer to tester ratio is 7.5:2.5 as QA/testing work can be shared with other team member, but development can't be shared with tester.

Q10. ***How difficult was the agile transition for you? What were the challenges faced?***
For developers, it can be difficult as introverts want to work on code all by themselves

Q11. ***How do you manage actual and expected burndown?***
It is automatically updated, depends on story points and acceptance of stories. If anything comes as a roadblock, one might do overtime to compensate the loss. If something is not part of the plan, there is a slack that should be communicated to the client to discuss further step whether to add the functionality or not.

Q12. ***Do you use agile management systems such as JIRA, DevOps to organise the software delivery process?***
NA

Q13. ***How do you perform version controlling in the agile software development?***
Yes, TFS

Q14. ***Do you maintain code on the cloud?***
No

Q15. ***How do you maintain to get consistent performance from the developer considering everybody's Emotional Quotient and Intelligence Quotient level is different?***
In team programming, we let everybody come to same level, pair programming, code walkthroughs and training. Also, each person come up with estimates, we try to understand challenges then build up their knowledge.

Q16. ***How often do you conduct workshops for behavioural issues to enhance team dynamics?***
No, but I think it would be beneficial for the team

Q17. ***How often must you provide training to the team on the technical and functional aspect of the project?***
Yes, immediate product group

Q18. ***Do you think that the technical, functional or behavioural issue interferes with the behavioural issues? If yes, how do you manage the delivery process?***
"In our delivery process, everything is maintained in six weeks, nothing is maintained for the integration testing, regression testing or deployment process. For issues, we do what we can do if roadblocks occur. We usually deliver everything we planned for."

Q19. ***How do you manage to interact with the client when the client is not working in the same time zone?***
Our clients are usually in same time zone

Interview 5

Q1. ***Your client is from which sector?***
Telecommunications

Q2. ***How often do you have the meeting with clients?***
Product Owner talks daily and team talks weekly in a time bomb which lasts for 2-3 hours

Q3. ***How often do you have the daily stand-up meetings? Do you adhere to the daily stand-ups routine in stressful time?***
Daily stand meetings are time tracked for 10-15 minutes

Q4. ***How do you decide the Sprint duration?***
Weekly because we focus on data analysis

Q5. ***How much difference do you notice in agile in comparison to other software development method?***
Huge difference because we communicate client on a daily and weekly basis. We follow pair programming, and cross training helps to complete work as team member anybody can do any work. We maintain product backlog, sprint meeting, task list, user stories. We estimate the task priority centrally where everybody raise ticket size till everybody ticket size is same then resolve the issue within the team. Then question is written on the page then scrum master and team decides whether to make task or sub task or epic. First we create an Epic then each epic will have list of tasks which will have its owner. When everybody closes all the tasks, epic will be closed. If somebody has a problem in a task then member can raise separate task with high priority.

Q6. ***How do you accommodate to request the change?***
Product owner decides the priority and sprint will have the task and no other work will come in that week, team will finish that work on priority. Our team is divided into product owners and all Product owners sit together to decide the value out of the project. Our sprint divided into three teams which manages the requirements.

Q7. ***What are the challenges faced in agile development?***
In the task, if the task is 80% completed and 20% is not done then the task will be revised with the help of the client. Client and product owner and scrum master decide what to do next and the new task will be raised with high priority and old item will be raised as super high (P0, P1, P4). Priority is set by product owner; scrum master maintains the things in the team and product owner.

Q8. ***What tools do you use to monitor the progress?***
VSO Microsoft

Q9. ***Do you suggest any practices that you follow for every agile project?***
We keep our team small. Small teams are easy to manage.

Q10. ***How difficult was the agile transition for you? What were the challenges faced?***
Everybody should have domain knowledge which is an advantage and a disadvantage but you must cross train team members.

Q11. ***How do you manage actual and expected burndown?***
If any sprint item is left, then new item is created and added to the epic after team decision. 4-hour sprint meeting, retrospective review meeting

Q12. ***Do you use agile management systems such as JIRA, DevOps to organise the software delivery process?***
NA because we do data analysis

Q13. ***How do you perform version controlling in the agile software development?***
NA because we do data analysis

Q14. ***Do you maintain code on the cloud?***
NA because we do data analysis

Q15. ***How do you maintain to get consistent performance from the developer considering everybody's Emotional Quotient and Intelligence Quotient level is different?***
In sprint meeting, everybody will be asked for user stories to how and why. Scrum master will resolve behavioural issues by stopping the person and reliving in the separate meeting. If somebody is dependent on the other member, then the person must pay overtime, and agile does not give any room.

Q16. ***How often do you conduct workshops for behavioural issues to enhance team dynamics?***
It is extremely important to have the team on the same page. If any unexpected behaviour is triggered during the meeting, then it is handled in the separate one to one meeting with the person to resolve the issue. Because daily scrums are limited to 15 minutes and must be focussed on the product's progress.

Q17. ***How often must you provide training to the team on the technical and functional aspect of the project?***
If product comes, the team is decided, agenda is made then shared with the team. Every team member reads and time is given to every team to go through the workshops. Everybody is brought to the same level, and tests are conducted internally. If failed, retake until you pass. Then individual meeting happens, asked for reviews, product backlog

Q18. ***Do you think that the technical, functional or behavioural issue interferes with the behavioural issues? If yes, how do you manage the delivery process?***
In agile, everybody must self-managed and take responsibility independently. Take credit for your work and all work on the same level. Scrum master changes every week to give person a chance to get the flavour of the whole team. For technical issues, we have main code on cloud and automated build. First, we perform unit testing then integration testing. The checklist is followed

and the logs are maintained. Though, there are roadblocks such as communication gap but we record everything and manage everything daily.

Q19. ***How do you manage to interact with the client when the client is not working in the same time zone?***
Yes, all over the world but client manages to talk to us in our time zone, and no overtimes happen.

Q20. ***Do you have an Agile Coach in the project? If yes, how do you plan the visit?***
No dedicated coach but we had training for three days for 18 hours

Q21. ***Which of the agile methods do you follow from the Kanban, Scrum, XP and Lean? How do you implement the fusion of the methods?***
VSO Microsoft

Q22. ***Do you have any visual board to track the completed task?***
VSO Microsoft

Q23. ***Do you consider it interesting?***
Yes

Q24. ***Do you consider the role of the agile coach is important to bridge the gaps since agile is new to other project management techniques?***
Yes, we have gaps, and we may get benefit from the agile coach as sometimes, we don't have solutions as product owner also doesn't know. In the end, we raise the issue with the client, and he/she resolves.

Bibliography

Ågerfalk, P. J. & Brian, F., 2006. FLEXIBLE AND DISTRIBUTED SOFTWARE PROCESSES: OLD PETUNIAS IN NEW BOWLS?. *COMMUNICATIONS OF THE ACM,* October.49(10).

Anderson, D. J. & Reinertsen, D. G., 2010. *Kanban: Successful Evolutionary Change for Your Technology.* s.l.:Blue Hole Press Inc..

Anwar, A., Kamel, A. A. & Ahmed, E., 2016. *Agile Adoption Case Study, Pains, Challenges & Benefits.* New York, ACM, pp. 60-65.

Author, G., 2016. *How to Visualize Sentiment and Inclination.* [Online] Available at: https://www.tableau.com/about/blog/2016/1/how-visualize-sentiment-and-inclination-48534

Ayed, H., Vanderose, B. & Habra, N., 2017. *Agile Cultural Challenges in Europe and Asia: Insights from Practitioners.* s.l., ACM.

Beck, K. et al., 2001. *Principles behind the Agile Manifesto.* [Online] Available at: http://agilemanifesto.org/principles.html [Accessed 29 July 2017].

Bossert, O., Ip, C. & Starikova, I., 2015. *Beyond agile: Reorganizing IT for faster software delivery.* [Online] Available at: http://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/beyond-agile-reorganizing-it-for-faster-software-delivery

Brechner, E., 2015. *Agile Project Management with Kanban.* Washington: Microsoft Press.

Butt, S. A., 2016. Study of agile methodology with the cloud. *Pacific Science Review B: Humanities and Social Sciences,* Volume 2, pp. 22-28.

Chan, F. K. & Thong, J. Y., 2009. Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems,* Volume 46, p. 803–814.

Chen, R. (., Ravichandar, R. & Proctor, D., 2016. Managing the transition to the new agile business and product development model: Lessons from Cisco Systems. *Business Horizon,* Volume 59, p. 635—644.

Cole, R. & Scotcher, E., 2016. *Brilliant Agile Project Management: A Practical Guide to Using Agile, Scrum and Kanban.* s.l.:Pearson.

Davidov, B., Klein, R. & Weiss, M., 2017. *System, apparatus, and method to facilitate management of agile software development projects.* United States, Patent No. US9619208 B2.

Denning, S., 2016. The age of Agile. *Strategy & Leadership,* 45(1), pp. 3-10.

Dingsøy, T. & Lassenius, C., 2016. Emerging themes in agile software development: Introduction to the special section on continuous value delivery. *Information and Software Technology,* Volume 77, pp. 56-60.
Drury-Grogan, M. L., Conboy, K. & Acton, T., 2017. Examining decision characteristics & challenges for agile software development. *The Journal of Systems and Software,* Volume 131, p. 248–265.
Dybå, T. & Dingsøyr, T., 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology,* Volume 50, pp. 833-859.
Elssamadisy, A., 2009. *Agile Adoption Patterns.* Boston: Pearson Education Inc..
Fernández, D. M. et al., 2015. Naming the Pain in Requirements Engineering Comparing Practices in Brazil and Germany. *VOICE OF EVIDENCE,* SEPTEMBER/OCTOBER.
Gandomani, T. J. et al., 2014. How Human Aspects Impress Agile Software Development Transition and Adoption. *International Journal of Software Engineering and Its Applications,* Volume 8, pp. 129-148.
Iivari, J. & Iivari, N., 2011. The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology,* Volume 53, pp. 509-520.
Inayat, I. et al., 2014. A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior,* 2 December, Volume 51, pp. 915-929.
Jovanovic´, M., Mas, A., Mesquida, A.-L. & Lalic, B., 2017. Transition of organizational roles in Agile transformation process: A grounded theory approach. *The Journal of Systems and Software,* pp. 1-21.
Kim, S. et al., 2016. *Our Journey to becoming Agile: Experiences with Agile transformation in Samsung Electronics.* Seoul, IEEE.
Kuusinen, K., Gregory, P., Sharp, H. & Barroca, L., 2016 . *Strategies for doing Agile in a non-Agile Environment.* Ciudad Real, ACM.
Laanti, M., 2013. *Agile and Wellbeing — Stress, Empowerment, and Performance in Scrum and Kanban Teams.* Hawaii, IEEE.
Laanti, M., Salo, O. & Abrahamsson, P., 2011. Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology,* Volume 53, pp. 276-290.
Lee, S. K. J. & Yu, K., 2004. Corporate culture and organizational performance. *Journal of managerial psychology,* 19(4), pp. 340-359.
Lindsjørn, Y. et al., 2016. Teamwork quality and project success in software development: A survey of agile development teams. *The Journal of Systems and Software,* Volume 122, p. 274–286.
M. R. R. Lazwanthi, A. A. P. P. S. S. A. E., 2016. *Cultural Impact on Agile Projects: Universal Agile Culture Model (UACM).* s.l., IEEE, pp. 292-297.
Mahadevan, D., 2017. ING's agile transformation. *McKinsey Quarterly*, January.

MAHNIC, V., 2014. Improving Software Development through Combination of Scrum and Kanban. *Recent Advances in Computer Engineering.*
Maples, C., 2009. *Enterprise Agile Transformation: The Two-Year Wall.* s.l., IEEE.
Martini, A., Pareto, L. & Bosch, J., 2013. *Communication Factors for Speed and Reuse in LargeScale Agile Software Development.* s.l., ACM.
Mendes, T. S. et al., 2016. *Impacts of Agile Requirements Documentation Debt on Software Projects: A Retrospective Study.* s.l., ACM, pp. 1290-1295.
Moe, N. B., Aurum, A. & Dybå, T., 2012. Challenges of shared decision-making: A multiple case study of agile software development. *Information and Software Technology,* Volume 54, pp. 853-865.
O'Connor, R. V. & Duchonova, N., 2014. *Assessing the Value of an Agile Coach in Agile Method Adoption.* Luxembourg, Springer.
Parizi, R. M., Gandomani, T. J. & Nafchi, M. Z., 2014. *Hidden Facilitators of Agile Transition: Agile Coaches and Agile Champions.* Malaysia, s.n.
Poppendieck, M. & Poppendieck, T., 2003. *Lean Software Development: An Agile Toolkit: An Agile Toolkit.* s.l., Addison-Wesley.
Qumer, A. & Henderson-Sellers, B., 2008. A framework to support the evaluation, adoption and improvement of agile methods in practice. *The Journal of Systems and Software,* Volume 81, pp. 1899-1919.
Ramtin Jabbari, N. & Tanveer, B., 2016. *What is DevOps? A Systematic Mapping Study on Definitions and Practices.* Edinburgh, ACM.
Rigby, D. K., Sutherland, J. & Takeuchi, H., 2016. *Embracing Agile.* [Online] Available at: https://hbr.org/2016/05/embracing-agile
Rodríguez, P., Partanen, J., Kuvaja, P. & Oivo, M., 2014. *Combining Lean Thinking and Agile Methods for Software Development A Case Study of a Finnish Provider of Wireless Embedded Systems.* Hawaii, IEEE.
Senapathi, M. & Drury-Grogan, M. L., 2017. Refining a model for sustained usage of agile methodologies. *The Journal of Systems and Software,* Volume 132, pp. 298-316.
Serrador, P. & Pinto, J. K., 2015. Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management,* Volume 33, p. 1040–1051.
Stellman, A. & Greene, J., 2015. *Learning Agile: Understanding Scrum, XP, Lean, and Kanban.* Sebastopol(California): O'Reilly Media Inc..
Sutherland, J., 2014. *Scrum: The Art of Doing Twice the Work in Half the Time.* s.l.:Crown Business.
Tessem, B., 2014. Individual empowerment of agile and non-agile software developers in small teams. *Information and Software Technology,* Volume 56, pp. 873-889.
Tolfo, C., Wazlawick, R. S., Ferreira, M. G. G. & Forcellini, F. A., 2009. Agile methods and organizational culture: reflections aboutcultural levels. *JOURNAL OF SOFTWARE MAINTENANCE AND EVOLUTION,* Volume 23, pp. 423-441.
VersionOne, 2017. *11th annual state of the agile report,* s.l.: VersionOne Inc.

Voigt, S., Garrel, J. v., Müller, J. & Wirth, D., 2016. *A Study of Documentation in Agile Software Projects.* s.l., ACM.
Waardenburg, G. v. & Vliet, H. v., 2013. When agile meets the enterprise. *Information and Software Technology,* Volume 55, pp. 2154-2171.
Walton, S. & Huey, J., 1992. *Sam Walton, Made in America: My Story.* New York: Bantam.
Wang, X., Conboy, K. & Cawley, O., 2012. "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *The Journal of Systems and Software,* Volume 85, p. 1287–1299.
ZipKin, P. H., 1991. *Does manufacturing need a JIT revolution?.* [Online] Available at: https://hbr.org/1991/01/does-manufacturing-need-a-jit-revolution
[Accessed 29 July 2017].
Zulzalil, H. et al., 2014. Exploring Facilitators of Transition and Adoption to Agile Methods: A Grounded Theory Study. *JOURNAL OF SOFTWARE,* July.9(7).