Agile Framework To Transform Traditional Team

Abdulrahman Alsari
Department of Computer Science,
King Abdulaziz University,
Jeddah, Saudi Arabia
aalsari0003@stu.kau.edu.sa

Rizwan Qureshi
Department of Information Technology,
King Abdulaziz University,
Jeddah, Saudi Arabia
rmuhammad@kau.edu.sa

Abdullah Algarni
Department of Computer Science,
King Abdulaziz University,
Jeddah, Saudi Arabia
amsalgarni@kau.edu.sa

Abstract—**With the increasing computing power of processors, more complex web and mobile applications are being developed by the developers. To meet the consistent high standard software deliverables, software developers need to follow the software design life cycle as a standard practice. The conventional way suffers certain drawbacks in project management that need to be addressed. The newer approach called agile methodology is much efficient and improves the quality of the product if followed by the team members as per the agile values. Agile is one of the software development methodologies that have a lot of features and flexibility which play a vital role to bring improvement in educational as well as industrial sectors. As the traditional methodologies are not flexible with changes, whereas Agile methodologies main advantages are interacting with the customer, respond to changes and strong communication and collaboration. In this research, we have conducted a critical survey based on three fundamental modules including design and learning strategies, team building, and profiling. The consequences showed that the conducted survey brought up the current situation of organization and opening new research direction to bring Agility in existing systems.**

Keywords—Agile Methods, Traditional Frameworks, Transformation.

I. Introduction

Agile has different methodologies extreme programming (XP), Scrum, Dynamic systems development method (DSDM), Crystal, Feature Driven Development (FDD), Lean Development and Adaptive Software Development (ASD) are the main agile methodologies, these methodologies have much in common, including short iterative life cycles, quick and frequent feedback from customers, and constant learning. During the past decade, the most likely used of Agile was Scrum and XP/Scrum hybrid. The terminology Agile in software development refers to a set of specific principles where the solution to the design problem is achieved through an effective collaboration methodology. Here, the resolution to the problem is evolved by a matured collaboration between different cross-functional and self-organizing teams. The keyword agile in the English dictionary describes its flexibility or the capability to react and handle the fast-changing technology, demand of the software development sector and users. It can be called as an incremental and iterative process in which direct collaboration is happening with the ultimate customers. In this development process, each iteration lasts from one to three weeks. During this period, several validations are needed to pass before the final release. This includes multiple sets of actions that are carried out by the cross-functional teams.

In general, traditional methods are costly for smaller projects and smaller teams because they introduce a significant amount of administrative overhead. This is due to the effort required to support extensive documentation or planning every detail before acting. However, they have an advantage in understanding and implementing the project's lifecycle easily. There are different traditional methodologies such as Waterfall and Rational Unified Process (RUP).

Waterfall is a document-driven approach where the project phases are well-determined and executed sequentially. Practically, a phase starts only when the previous phase is finished. As a proof of the high orientation of this method to documentation, it suffices to recall the following (phase, output) pairs: (user requirements, User Requirements Document), (software requirements, Software Requirements Document), (software design, Architectural Design Document), and so forth [1].

Therefore, the detailed planning of each step facilitates easier identification of both deliverables and milestones. It is suitable for either large projects or critical/complex systems where fault tolerance resides at the highest level. In essence, the approach has two well-established principles: a) define work before design software and b) design software before coding application. It has longer turnaround times than more flexible development models because 1) it is difficult and time consuming identifying the requirements accurately during earlier project phases, 2) testing is done late in Software Development Life Cycle (SDLC), 3) changes are implemented with difficulty because the inflexibility of the design (which is already done when errors are identified), 4) risk management is not inherent (even if it is well-established), and 5) users' involvement is reduced [1-2].

Rational Unified Process (RUP) was proposed to overcome some shortcomings of the Waterfall model. Even if it still belongs to the group of traditional approaches, the hybrid combination between Unified Modeling Language (UML) and Unified Software Development Process has managed to avoid the lack of incremental/iterative exploration from the Waterfall model. It emphasizes the importance of 1) use cases, 2) task assignment, 3) project cycle, and 4) architecture. Practically, each of the four cycles (inception, elaboration, construction, transition) represents a distinct release. RUP uses predictability for Project Management constraints like budget and schedule. It still has a long turnaround time because the "classical" phases from the Waterfall model repeat during each of the four iterations (more or less) [1].

The combination between "when" (characteristic to phase modeling) and "what" (define the workflow within each phase) provides better control of the changes and gradual goal achievement. Not last, the requirements and components are also managed better in what regards software quality requirements. However, the documentation effort remains at a high value, while intensive planning is still a compulsory feature.

Both Agile and traditional methodologies define the workflow and organization inside software development work. However, the goals are achieved differently in each case: the traditional approach uses a linear model sequentially (non-overlapping) execution of the phases, while Agile environment uses iteration and short "sprints" (up to several weeks per each). Each sprint proposes a series of deliverables that can be either revised or rescheduled for the incoming sprint in case they are not delivered; the main objective of this philosophy is to release functional and complete components during all iterations [1], [3].

Agile best works when the overall scope of the project is not defined in advance and/or adaptation is possible; the iterations require more user involvement (scenarios focused) and a team-based approach (thus more collaboration) for meeting the project constraints (e.g. budget, schedule). Because of the iterative approach, both the requirements and changes are done in more flexibly. However, the difficulty level increases during the revision process. Even if Agile development model seems less focused on project organization, it is often preferable to use traditional approaches. As a result, the built-in flexibility can make possible adjustments anytime as long as the team members are valuable, and this will lead to a faster error discovery. The challenge is to avoid the "spaghetti code" effect that occurs in many iterations. But, if the project stays synchronized with the schedule, the effect will be reduced due to different components are delivered during each sprint [3].

II. Background

In the late 1970s, the PC blast occurred. With it, the regular person accessed current processing. Buyer request drove development at a quicker pace than any time in recent memory seen. Organizations endeavoring to meet consistently switching client wants accelerated the pace at which business needs changed. Programming improvement expected to meet the pace and changes, and the inflexible approach that had controlled the SDLC world couldn't keep up. It essentially couldn't convey the product sufficiently quick or react adequately to necessity changes during advancement.

By the mid1990s, a little gathering of programming industry pioneers started creating and elevating imaginative ways to deal with SDLC which grasped rapidly responding and adjusting to changing prerequisites and advances. Rapid Application Development (RAD), Rational Unified Process (RUP), Scrum, and Extreme Programming (XP) turned into the new, exceptionally responsive, and adaptable programming advancement philosophies utilized.

In 2000, a little gathering of programming industry pioneers met in Snowbird, UT, to examine these new strategies. The term nimble programming improvement was authored in 2001 to depict the adaptable idea of programming created in iterative stages and turned into a sweeping term for the new systems.

A. *Scrum*

Scrum is recommended for those projects focused on business value (more than any other feature). It emphasizes the team organization and practices, but it does not neglect project management tasks as compared to other Agile development methodologies. The project is divided into packets, where each one of them is tested and documented continuously after the product is built. The customers receive "demos" frequently and can involve easier in requirements definition. Thus, Scrum provides a flexible work environment (advantage) for all stakeholders; at the same time, the disadvantages are those outlined for all Agile methodologies [4].

B. *Extreme Programming (XP)*

Extreme Programming (XP) is an agile method that highlights the technical practices and skills grouped in small and often releases. XP is focused on code review and intensive use of refactoring. It suits well when the project can be divided into many pieces having simple designs; this facilitates the coding portion for each chunk quickly. The projects which require onsite presence (colocation) are often dealt with using this method because unit tests are executed daily, while acceptance tests are designed by customers in most of the cases. The greatest advantage of this method is given by the technical skills, while their lack represents a major disadvantage for the team. It remains so far the widest Agile methodology [4].

C. *Lean*

Lean in an agile method that is focused on waste elimination and continuous optimization of the product. Even if the decisions are taken as late as possible, this does not influence the delivery dates. The process is amplified by learning and the major decisions are often given to first-line workers. This method uses thinking tools for simulating the process. It is suitable for those projects where ROI (Return of Investment) is a top priority. The greatest advantage is the complementarity of the approach (it can fill existing gaps), while the biggest disadvantage is given by a lack of decisional hierarchy in some points [4].

III. Related Work

Qualified software products must be developed in a systematic manner, and this is required more accurate development approach that monitors the quality of the work from the beginning to the last phase of development. Methodologies primarily focus on strong communication, collaboration between the team of the project and quality, which are considered as successful keys in the project. Jain et al. [5] reviewed different approaches to improve the quality process of agile methodology.

IT technologies are developed daily, and the organization needs to be up to date to keep up with the competitors. However, each chief information officer (CIO) has different experience, culture, qualifications, skills, and abilities that will impact the work organization in a different way. Padayachee et al. [6] proposed a framework that helped CIO's to take a decision in an Agile business environment that depends on the individual's characteristic, organization, and ecosystem.

Kim et al. [7] described an agile transformation of an electronic manufacturing company. This is done through three phases. Firstly, the process transformation that includes roles and responsibilities of the team. Secondly, engineering transformation for pair-programming, test-driven development and deployment. Finally, the organizational transformation for limitations to start the process and adopt engineering practices.

Lopez-Martinez et al. [8] presented a review of agile methodologies adoption with a focus on Scrum. They categorized several problems that could affect Agile adoption in four groups; these are people, process, project, and organization. All of the mentioned problems must be addressed for the successful adoption of agile and changing from the traditional process to agile method.

Agile has the ability to deal with the change of requirements while traditional methodologies are suited with defined requirements. Singhto et al. [9] presented an implementation of Scrum technique adaptation for the project of the human resource information system (HRIS). And that was good for a change of user requirements, delivery time and required budget as it was a small project.

Vijayasarathy [10] presented a study of the difference of software development methodology and the relation with organizational, project and team. One of the results is the number of teams has a significant relationship with the development approach while small team size, low budget required, and a small number of organizational employees' theses are applicable for Agile. Second for traditional methodology requires medium or multiple teams, high budget and a large number of employees.

Srivastava et al. [11] presented a research of Scrum model that is one of the Agile methodologies. Scrum has a different approach from the traditional methodologies, it is a combination of iterative and incremental model, Scrum focuses on productivity by communication and planning between teams. One of the most important features is accepting changes in each iteration. Increases the quality by ensuring all teams are aware of problems and changes; this will positively affect and keep the cost of the project as planned.

A lot of barriers faced Agile in a distributed environment, and these may affect the implementing of Agile methodology. Awar et al. [12] proposed a model for applying agile in a distributed environment and determine the critical factors. As the teams of these projects are in different locations and due to geographical distance, teams need to coordinate daily with their updates. The critical factors categorized into four factors; human, process, management and organizational factor.

Trimble et al. [13] presented a study for transformation from traditional to lean to Agile for finding optimal software engineering cycle. This study implemented since 2008 for five years as it was NASA project. After two years of improving the process with multiple iterations, researchers achieved three keys objectives; first is a close relationship with customers highly recommended, second is utilize the available resource with a focus of high priority targets, finally, verify progress and strategic map.

Keshta et al. [14] presented a comparison between traditional and Agile methodology based on team size and project domain. Agile has high flexibility for requirements change and it is more applicable with small team and software projects. While traditional is not flexible and fit on a large team and applicable for reusable or predictable project. Agile has four characteristics that make it more distinct, these are; Adoption of changing requirements, people are an important factor for success, self-organized team, and collaboration (customer is close to project team). The traditional methodology has a lot of limitations that are no flexibility, complex rules, teams have just to follow the documentation.

Mahmud and Abdullah [1], discussed the approaches of the development teams to achieve the best of the web and mobile apps. The case studies were undertaken with a small team of fewer than 10 employees. In modern fast-changing technology and requirements, it is important for the developers to use agile methodology to meet the demands of the customers. They presented the different categories of the mobile application development environment and the uniqueness in terms of performance, also studied the prior researches undertaken towards mobile app development framework and processes. They analyzed the potential and its effectiveness in bridging the gap in the development processes. The manufacturing company developed software for consumer electronic products using the waterfall model. The software plays an important role in the performance of any hardware product. It needs to be properly synced with the hardware. Poorly developed software can give disastrous results on even a high performing costly hardware. To achieve the successful transformation in the company, a plan was created with several significant phases. The first step was a roadmap creation. In the second step, the processes were revised. In the third phase, an integrated master plan was worked out by the team members to adopt the standard engineering approaches and high-performance optimized coding style. In the fourth phase, cross-functional teams were formed. Researchers concluded DSDM as an effective method to be used in large scale mobile app development. This approach is capable enough to handle custom development requirements as demanded by the customer and effective project management to achieve the desired results on time. The mobile application development environment suffers several challenges during their development phase. The cross-compatibility nature of mobile apps poses the biggest complexity for the software development team. The applications must be compatible with all the OS like windows, iOS, and android. Simultaneously, the customer may ask for certain UI or functionality changes in the wireframe. The UI may need to be replaced completely or modified once the customer checks the beta release. For a successful transition from the conventional to agile methodology and its adoption by the team members several agile training programs, whitepapers and newsletters were published by the company. In spite of the rigorous efforts and guidelines, the transformation was difficult to produce positive results. A pilot project was also undertaken to sync the team members including UX, developers, and QA.

Filho et al. [15] described an Agile Software Development learning through Open Hardware Project. They proposed the study and implementation of an agile approach in product development and introducing a simplified learning environment for the students. Team size and experience, there were seven members in the team, and they were beginners in this industry as they were students. The research work was undertaken on a hardware-software co-design project i.e. a Plug Meter. The researchers used an open hardware project to illustrate the agile methods in application development for teaching the student community instead of using the agile development learning method through software. The development cycle was divided into two different phases. The first phase was developing a fully operational electric charge meter. The second phase was to develop the communication mechanism i.e. the connection module for the mobile devices. The product was a Plug meter that has the capability to measure energy consumption and report the total consumption using the Arduino hardware development platform. This study pointed out the use of Arduino development platform for the agile approach in embedded hardware application design. The Plug Meter design team was fully satisfied with the development of the project. It enabled them to learn the product development life cycle. The biggest challenge was to introduce a real-time learning environment for the students. The students were mostly dependent on professors, friends, and families to have user feedback.

Bose [16] discussed her experience in the transition from Classic/Traditional to Agile in the field of software development model through critical steps for smooth transition. Bose recommended several transformation steps from the transition of the classical waterfall model to the agile. The most important factor is the initial plan and timing estimation needed to complete the project as per the specified deadline.

Kirmani [17] presented a comparative analysis of Agile methods for mobile application development. One of the used studies in mobile application development, team size distributed as per the product development phase which includes product development and testing. The popular agile approaches discussed are scrum, lean and XP. Transformation is applicable to the complete software development life cycle. It starts from project inception and gathering the requirements from the client. The next phase is the architecture design and coding phase. Finally, system integration and testing in different environments are done. The popular agile methods are DSDM, Scrum, ASD, and XP.

As a result, and recommendations, using agile practices will increase the speed of product development. The product development speed quite becomes important as there are several new applications available on the app store on a daily basis. The main challenge, usually, developers are acquainted with the conventional waterfall model, so the transformation becomes quite difficult for them.

Filip et al. [18] conducted a thorough research review on agile to Lean Software Development Transformation. They studied the transformation processes to get an idea regarding the performance metrics, benefits and other challenges during the transformation phases. The conclusion regarding the transformation benefits was improved design flow, improved bug fixes and reduced lead time. The basic problem encountered during the transformation was the 'lean mindset'. The major part of the transformation criteria is to use the adaptive transformation framework [19] which can reproduce higher performance metrics.

Lombardi [20] discussed how to Being Business Agile focusing on flow efficiency. The environment of the case study, a system under consideration was an integrated model with hardware, software, firmware process development. Team size is over 300 as the operational strategy for the project extends to 3 to 4 design houses across the globe. In any circumstances, the team i.e. neither the development team nor the testing team should be demotivated regarding the perceived quality instead the focus should on identifying the factors which reduce the throughput. An organization on the path of transforming itself to business agile should be quite flexible in responding to the demands of the development and testing team. The transformation from existing methods to new methods must be unruffled. The biggest challenge is the introduction of large-scale programs in a shorter span of time.

With respect to educational systems, agile methodology plays a vital role in improvement. Numerous scientists had the instinct of fitting Agile approaches to the instruction condition [21]. Stewart et al. [22] introduced a first survey of the writing planned for demonstrating how nimble techniques were applied to instruction. Additionally, they gave a mapping between the qualities and standards of the Agile Manifesto to explicit instructive techniques and exercises. Table 1 shows the qualities characterized in the Agile Manifesto and the applied mapping, which comprises of the interpretation of programming improvement figures and jobs to the instruction condition.

TABLE I. MAPPING AGILE OVER REAL EDUCATION SYSTEM [23]

Sr#	Agile manifests	Agile Manifests in Education
1	Person to person interactions over processes and technologies	Students and teachers over processes and tools.
2	Software utilization over detailed documentation.	Working on projects over detailed documentation
3	Clients quislingism, over contracts	Students and teachers interaction over teaching courses
4	Reaction to change over the plans	Students feedbacks instead of following plans

IV. PROPERTIES OF TRADITIONAL METHODS VS AGILE METHODS

Before moving toward 'how to transform development organization from the traditional methodologies to Agile', it is necessary to know firstly that what are the primary factors that should be considered while transforming from traditional methodologies to Agile. However, below table II describes the top ten properties that should be considered and understand before transforming from conventional to Agile methods.

TABLE II. TOP TEN PROPERTIES WHILE TRANSFORMING FROM CONVENTIONAL TO AGILE METHODS [23]

Sr.#	Properties	Traditional Methods	Agile Methods
1	Attitude	Predictive	Adaptive
2	Project size	Large	Small
3	Team size/mindset	Large/disciplined	Small/innovative
4	Project management model	Autocratic	Decentralized
5	Change attitude changes	Resistant against	Embracing changes
6	Documentation	Comprehensive	Light and abstract
7	Upfront planning	Comprehensive	Limited
8	Life cycle	Tied and bound	Unlimited iteration
9	Organizational culture	Command and control	Leadership and collaboration
10	Return of Investment	At the end of the project	Early stages

V. Challenges while Transforming from conventional to Agile methods

While Agile transformation process, the challenges under four primary domains that companies may face and should be resolved.

A. Organization and management

Management Style: *To changing from conventional to agile strategies, the executives style ought to be changed from "order and control" to "administration and cooperation" that can cause to acquire adaptability and responsiveness association, and gives favorable circumstances of synergism all the while* [24].

Role of PM: *Undertaking Manager ought to be changed from organizer and controller to chief and facilitator* [25].

Group Decision Making: *Cooperative choice creation is likewise an issue in portion of improvement assets, arrangements of vital product offering and performing advancement and upkeep assignments in groups* [26].

Documentation: *In customary techniques information the board depends on heavyweight documentation yet in light-footed strategies, documentation is restricted and information is for the most part comprehended and live in the leader of the improvement colleagues* [27].

B. People

Pair programming in XP for senior developers: *Right now, should choose suitable staff and give them fundamental preparing, coaching and making a lot of work rehearses that advance procedure greatness* [28].

Customers Role: *Clients assume a basic job in accomplishment of spry techniques and they ought to be responsive, cooperative, approved, submitted and educated. Right now, conventional undertaking supervisors couldn't adjust with new circumstance* [29].

Awareness to Team member: *Absence of enough preparing, instructing and tutoring is a basic issue right now. Supervisors should give enough consideration to allot an accomplished and expert mentor in their groups.*

C. Process

Predefined Standards and measurements: *In customary techniques forms depend on characterized/standard exercises and estimation while forms in deft strategies depend on unsure exercises that help fast improvement and top notch creation* [30].

Implementing agile activities: *Executing some agile exercises, for example, nonstop coordination, creating forthright test code and continuous testing in conventional programming engineers is difficult.*

choosing appropriate agile method: *Agile strategies are distinctive in needs, execution, venture and group size, cycle time, code proprietorship and different elements. Lamentably, there is no bound together deft methodology, in this way, associations ought to choose about their proper lithe strategy.*

D. Tools

Measuring tools: *here and there desire for customary engineers to finding sufficient and archived estimating apparatuses in lithe techniques causes confounding* [31].

Using non-flexible tools and hardware: *Organizations should utilize devices that can supply gradual development, consistent coordination, re-working, rendition the executives and other deft innovations* [32]. *They should also focus on agile metholdogy implementations with respect to the most emerging technologies such as fog computing, parallel computing, GPU technology hardware, and supercomputing technologies* [33].

VI. Research Questionnaires

A. Design & Strategic learning based questionnaires

Based on design and strategic learning, we conducted our survey based on given list of questionnaires as follows. The answers for design and strategic learning based questions were categorized into four options including 1) Categorically Yes, 2) Yes, 3) Categorically No and 4) No.

1. Does the group normally produce an incentive for their partners?
2. Does the group approve their work as well as could be expected?
3. Are partners effectively included?
4. Is the group self-sorting out?
5. Does the group endeavor to improve their procedure?
6. What level do you have for certainty about the external environment we're in?
7. What level do you have of consensus on the way ahead?
8. Do you work through a network of small, empowered teams?
   (it means are you working a small team or having large setup)
9. Do you use rapid decision and strategic learning cycles?
10. Do you have dynamic people models based on a community, passionate about working cohesively with common intent?
11. Do you work in hybrid teams combining employed staff, contractors and outsource vendors?
12. Do you move toward innovative technology with the ability to disrupt the existing business models and ways of doing things?
    (it means, with the passage of time, the business models are required to be improved, however, you do apply any strategy to move toward the improvement in your business/system)
13. Is your company ready to commit a full-time business master, called an item proprietor?
14. Is your company ready to commit a full-time conveyance group?
15. Is your company ready to give a business examiner to inspire in the just in time (JIT) prerequisites?
    (it's about asking to a company that; do you have strategy to analyze your business to increase the in time requirements provisioning)
16. Is your company ready to time box every emphasis?
17. Is your company ready to place the perfect individuals in the correct jobs?
18. Is your company ready to help a collective situation?
19. Is your company ready to apply the essential order?
20. Have you assigned the right people in right roles?

B. Profile mapping

1. Agile Experience: I will be answering this survey from the point of view of:

   1. I am currently on a project team where we believe we are agile
   2. I am currently on a project team where we are trying to become more agile
   3. I am not currently on an "agile team", but have been in the past (and will discuss the most recent one)
   4. I have never been involved with an agile team

2. Position: Which best describes your current position?

   1. Agile Team Member
   2. Architect/Architecture Owner
   3. Business Analyst
   4. Business Stakeholder
   5. IT Manager
   6. Operations/Support
   7. Product Owner
   8. Programmer
   9. Project Manager
   10. QA/Test
   11. Scrum Master/Team Lead
   12. Other

3. Department Size: Number of IT/Systems/Development people in your organization

1. 1 to 10
2. 11 to 50
3. 51 to 100
4. 101 to 500
5. 501 to 1000
6. 1000+

4. Sector: which sector is your organization primarily in?

1. e-Commerce
2. Financial
3. Government
4. IT Consulting
5. Manufacturing
6. Retail
7. Shipping
8. Technology (inc. Software)
9. Other

5. Location: Where are you located?

1. North America
2. South/Central America
3. Europe
4. Asia
5. Australia/NZ
6. Africa

VII. Results and Discussion

In this section, we have presented the results of conducted survey based on questionnaires presented in last section. The questionnaires were divided primarily into three major sections including Design & Strategic learning, profiling and team information of responded person. The volume of overall survey responses was up to 25 from different countries. The method we used to conduct this survey was through google form. However, firstly we present and analyze the profile based report as follows in figure 2 (a), (b), (c), (d) and (e). According to the questionnaire response, most of the people didn't used to with agile team methodology but interested to follow. Further regarding position, 27% of the people were holding PM position while working in agile network. We also noticed that most of the organization strength were containing in the range of 1 to 50 where majority where belonging to software houses. IT and government sectors. As most of the respondents were belong to Asian counties however, survey report shows the majority of usage from Asian countries.

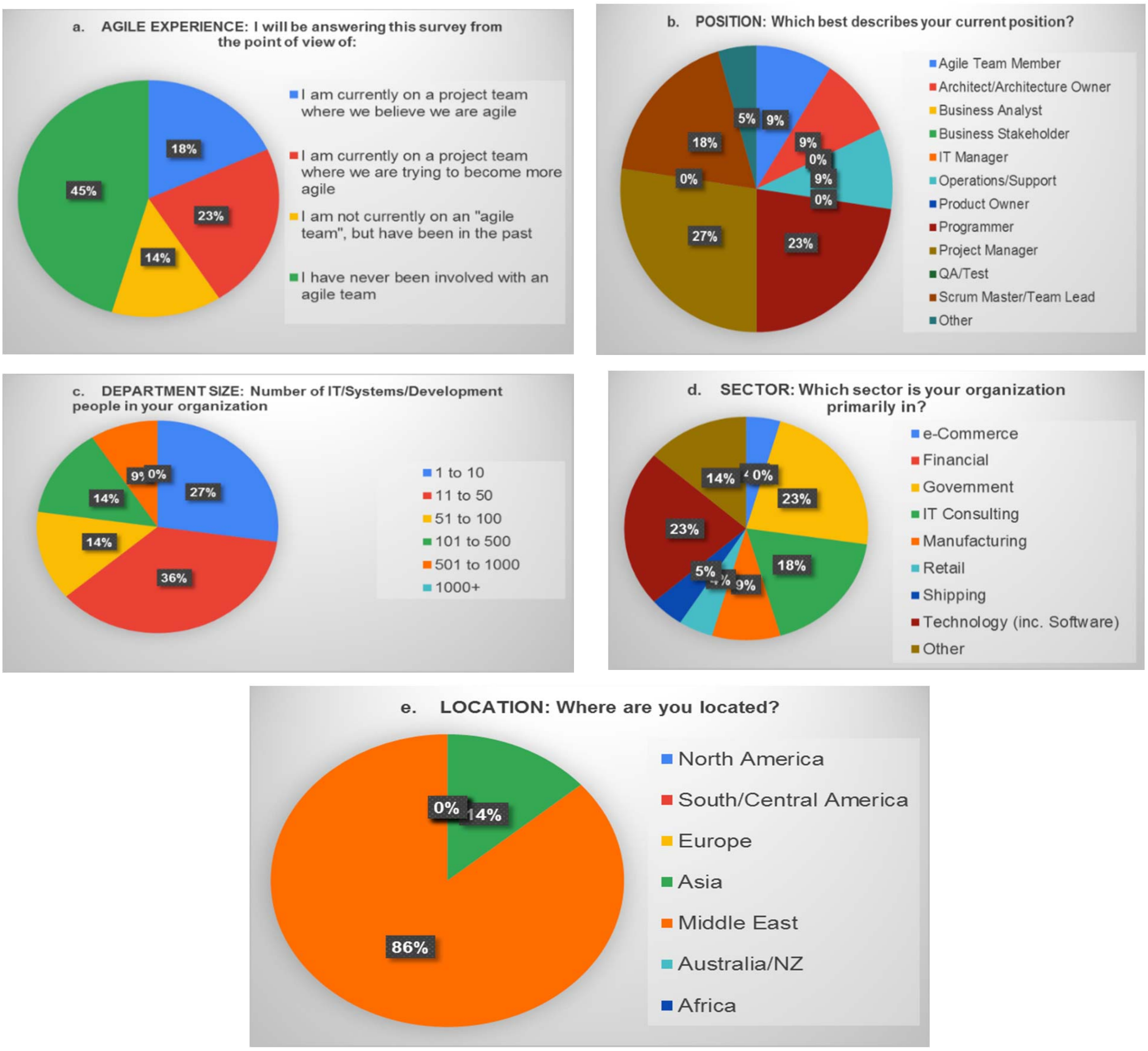

Fig. 1. Profile survey of (a) Experience, (b). Position, (c). Department Size, (d). Sector, (e). Location

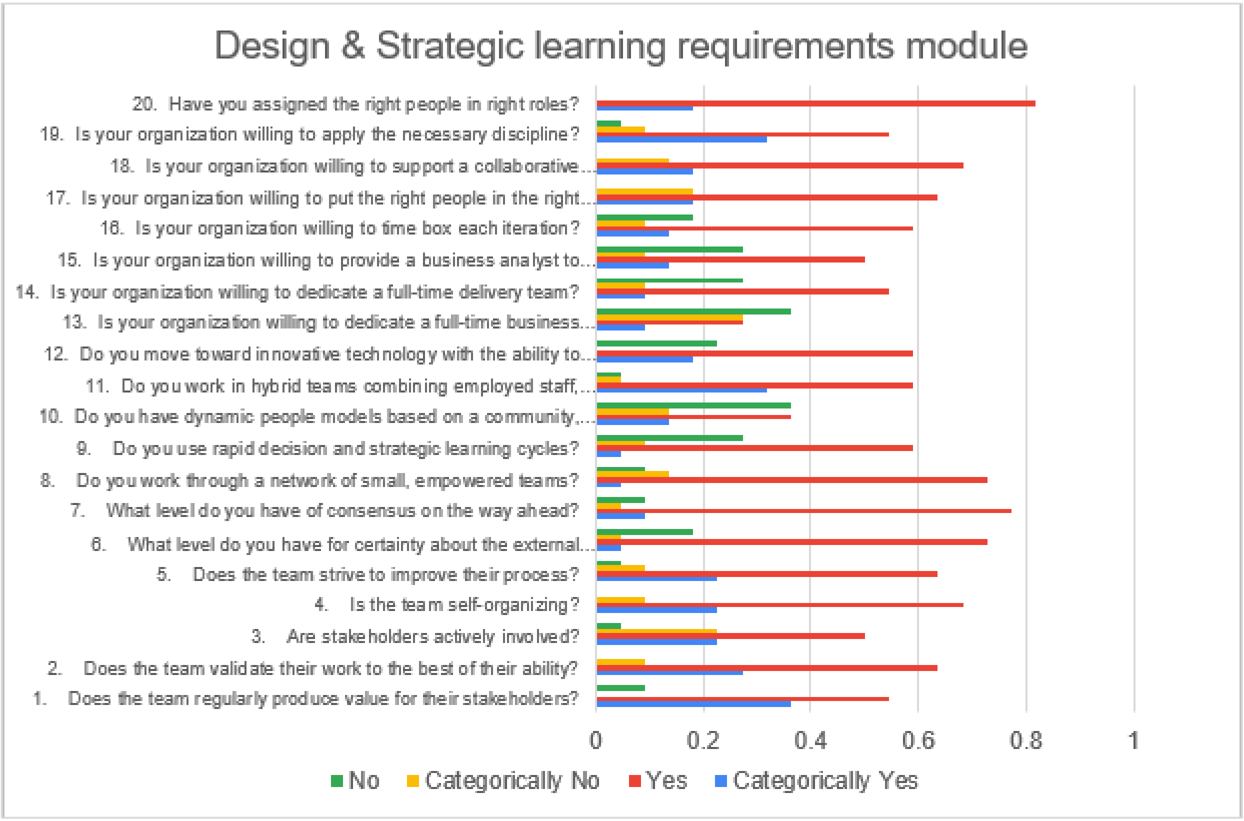

Fig. 2. Design & Strategic learning requirements survey report

Figure 2 showed the second category of our survey to determine the design and strategic learning requirements with respect to team regularity production for their stakeholders. According to this survey report, our first point was to know about team regularly produce value for their stakeholders where maximum peoples responded as 'yes' and 'categorically yes' that means most of the organizations are producing regularly for their stakeholders. In our next concern about work validation by team and involvement of stakeholders, majority were responded yes but few of them conflicted and said no because of not properly involvement of stakeholders. In next, we asked about the external environment and consensus on the way ahead, at this time the maximum people responded as categorically yes. In contrast, with respect to hybrid team collaboration environment, most the people said they don't have such collaborative environment which shows clearly that a strong collaborative environment is the primary requirement for agile implementation. Further we asked some question core related to the organizations willingness to 'dedication for full-time business expert', 'dedication for a full-time delivery team', 'to provide a business analyst for JIT', 'to time box each iteration', 'to assign right person at right role', 'support a collaborative environment', and 'to implement the mandatory discipline'. We noticed that for full time business expert dedication, around 50% of the people were not agree to dedicate for full time.

Similarly, we conducted the third survey report on team factor of our questionnaires and evaluated five factor including value, validation, stakeholders, self-organization, and reflective improvement as shown in figure 3 (1,2,3,4,5). With respect to Value concern, we observed that 68% people were engaged with key stakeholders at the start of their project. In contrast, 77% of the people believe in agility and adopting to bring improvement in their business processes. Moreover, in term of "validation of what strategy is followed by the team", 55% were agree to release at the end of the project, "final" testing is performed before releasing the system by an independent test team. We also noticed that while working directly teams with stakeholders, 59% people said that their team has a product owner who represents the stakeholder community. On other hand, we observe that just 9% of the people were agree in working with stakeholders with business analysts who provide requirements to the team directly. In next evaluation parameter 'self-organization', we asked peoples to know the adopting strategies in order to work together, where around 60% people were agree that project manager of the organization should assign the tasks directly to the team instead of self-efforts. Similarly, 50% people said, they are organizing the daily stand-up meeting in start of working hours, discuss the agenda based on 'what is done' and 'what to do', then assign the responsibilities accordingly to the team. In contrast, we noticed that only 9% people leverage and develop with respect to collaborative environment infrastructure.

In our last parameter ''reflective improvement' of team building module, we inspected about what strategy is to be followed while improving the way of working together. In response, 50% of the people measure and track the progress on daily based at the end of the day. Another 45% said, they actively make attempts to resolve the challenges that are identified in retrospective sittings throughout the project. According to 18% of the people, external auditors should review the team work and identify the parameters required to bring a potential improvement in the environment.

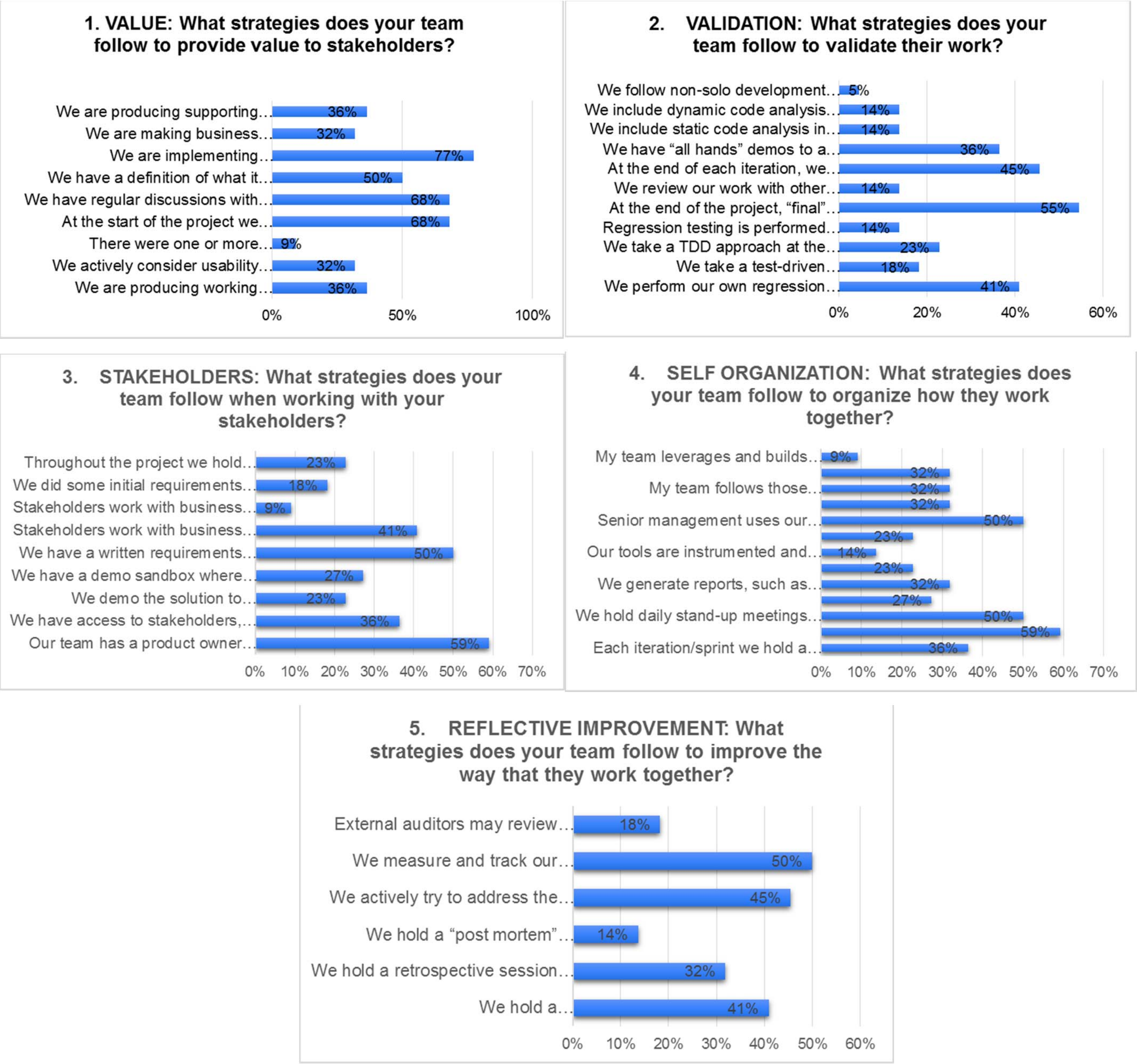

Fig. 3. Team survey report on 1). value, 2). Validation, 3). Stakeholders, 4). Self-Organization, 5). Reflective Improvement.

VIII. Conclusion

The traditional methodologies are not flexible with changes, whereas Agile methodologies overcome these challenges by interacting with the customer, respond to changes and strong communication and collaboration. In current study, we conducted a survey and presented a critical analysis report. Leading to transformation from current traditional methodology to Agile, we prepared the questionnaires based on three agile methodologies XP, Lean and Scrum. Further we categorized these questionnaires into three major modules including Design & Strategic learning requirements, team building and profiling modules. Each module contained several questions against which we received the responses. In our survey report we observed that 45% never involved in agile activities and following the current traditional systems. Only 18% organization were working under agile methodology where 23% play role of programmer as well as project managers in software IT companies. In term of design and strategic learning, majority of the organizations were not willing to dedicate full time business experts and delivery teams. Further in team building module, we noticed that 77% people were implementing different strategies to improve the business processes which clearly shows that organizations are fully interested to improve their businesses by following the advance strategies. In order to overcome this gap, we will provide a full-fledged adaptive framework which is actually the need of agile adoption that will be followed by all interested organizations to improve their business process as well as teams counseling.

References

[1] Mahmud, Dewi Mariati, and Nur Atiqah Sia Abdullah. "Reviews on agile methods in mobile application development process." 2015 9th Malaysian Software Engineering Conference (MySEC). IEEE, 2015.

[2] I. Sommerville, "Software Engineering 9th edition - Software Process Models" 2010.

[3] https://theagileblueprint.wordpress.com/2011/03/02/comparing-waterfall-and-rational-unified-process/ [Accessed on Jul 2019]

[4] http://www.base36.com/2012/12/agile-waterfall-methodologies-a-side-by-side-comparison/ [Accessed on Jul 2019]

[5] P. Jain, L. Ahuja, and A. Sharma, "Current State of the Research in Agile Quality Development," Comput. Sustain. Glob. Dev., pp. 1877–1879, 2016.

[6] R. Padayachee, M. Matthee, A. Merw, " Disruptive technologies and IT decision making in an agile business environment " , IEEE Africon 2017 Proceedings, pp. 1078–1083, 2017.

[7] S. Kim, H. Lee, Y. Kwon, M. Yu, and H. Jo, "Our journey to becoming agile: Experiences with agile transformation in Samsung electronics," Proc. - Asia-Pacific Softw. Eng. Conf. APSEC, pp. 377–380, 2017.

[8] J. Lopez-Martinez, R. Juarez-Ramirez, C. Huertas, S. Jimenez, and C. Guerra-Garcia, "Problems in the adoption of agile-scrum methodologies: A systematic literature review," Proc. - 2016 4th Int. Conf. Softw. Eng. Res. Innov. CONISOFT 2016, pp. 141–148, 2016.

[9] L. R. Vijayasarathy and C. W. Butler, "Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?," IEEE Softw., vol. 33, no. 5, pp. 86–94, 2016.

[10] A. Srivastava, S. Bhardwaj, and S. Saraswat, "SCRUM model for agile methodology," 2017 Int. Conf. Comput. Commun. Autom., pp. 864–869, 2017.

[11] K. B. Awar, M. S. I. Sameem, and Y. Hafeez, "A model for applying Agile practices in Distributed environment: A case of local software industry," Proc. 2017 Int. Conf. Commun. Comput. Digit. Syst. C-CODE 2017, pp. 228–232, 2017.

[12] J. Trimble and C. Webster, "From traditional, to lean, to agile development: Finding the optimal software engineering cycle," Proc. Annu. Hawaii Int. Conf. Syst. Sci., pp. 4826–4833, 2013.

[13] N. Keshta and Y. Morgan, "Comparison between traditional plan-based and agile software processes according to team size project domain.," 2017 8th IEEE Annu. Inf. Technol. Electron. Mob. Commun. Conf., pp. 567–575, 2017.

[14] M. Yadav, N. A. Professor, J. Yadav, and A. Professor, "Agile Methodology Over Iterative Approach of Software Development –A Review," Comput. Sustain. Glob. Dev. (INDIACom), 2015 2nd Int. Conf., pp. 542–547, 2015.

[15] A. Gomes Filho, "Agile Software Development learning through Open Hardware Project", 6th Brazilian Workshop on Agile Methods, vol. 16, no. 2016, pp. 978-1-5090-1377-7, 2016.

[16] L. Bose, "Transition from Classic/Traditional to Agile: Critical Steps for Smooth Transition", International Journal of Advanced Research in Computer Science RESEARCH PAPER Available, vol. 3, no. 0976-5697, pp. 136-139, 2012.

[17] M. Kirmani, "Agile methods for mobile application development: A comparative analysis", International Journal of Advanced Research in Computer Science, vol. 8, no. 5, 2017.

[18] F. Kiss, "Agile to Lean Software Development Transformation: a Systematic Literature Review", Proceedings of the Federated Conference on Computer Science and Information Systems, vol. 15, no. 2300-5963, pp. 969–973, 2018. Available: 10.15439/2018F53.

[19] Qureshi, M. Rizwan Jameel, and M. Kashif. "Adaptive framework to manage multiple teams using agile methodologies." International Journal of Modern Education and Computer Science 9.1 (2017): 52.

[20] G. Lombardi, "Being Business Agile Focusing on Flow Efficiency: Tale of a Practitioner's Approach," 2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC), Lisbon, 2016, pp. 113-117.

[21] Dewi, Deshinta Arrova, and Mohana Muniandy. "The agility of agile methodology for teaching and learning activities." 2014 8th. Malaysian Software Engineering Conference (MySEC). IEEE, 2014.

[22] Stewart, John C., et al. "Evaluating agile principles in active and cooperative learning." Proceedings of Student-Faculty Research Day, CSIS, Pace University (2009): B3.

[23] Salza, Pasquale, Paolo Musmarra, and Filomena Ferrucci. "Agile Methodologies in Education: A Review." Agile and Lean Concepts for Teaching and Learning. Springer, Singapore, 2019. 25-45.

[24] Yang, Haibo, Sid Huff, and Diane Strode. "Leadership in software development: Comparing perceptions of agile and traditional project managers." AMCIS 2009 Proceedings (2009): 184.

[25] Moe, Nils Brede, T. Dingsyr, and O. Kvangardsnes. "Understanding shared leadership in agile development: A case study." 2009 42nd Hawaii International Conference on System Sciences. IEEE, 2009.

[26] Moe, Nils Brede, Aybüke Aurum, and Tore Dybå. "Challenges of shared decision-making: A multiple case study of agile software development." Information and Software Technology 54.8 (2012): 853-865.

[27] Levy, Meira, and Orit Hazzan. "Knowledge management in practice: The case of agile software development." 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering. IEEE, 2009.

[28] Srinivasan, Jayakanth, Kristina Lundqvist, and Christer Norström. "Studying software organizations: in search of a method." Proceedings of the 3rd India software engineering conference. 2010.

[29] Conboy, Kieran, et al. "People over process: key people challenges in agile development." (2011).

[30] Singh, Narinder Pal, and Rachna Soni. "Agile software: Ensuring quality assurance and processes." International Conference on High Performance Architecture and Grid Computing. Springer, Berlin, Heidelberg, 2011.

[31] Javdani, Taghi, et al. "On the current measurement practices in agile software development." arXiv preprint arXiv:1301.5964 (2013).

[32] Brockmann, Patricia Shiroma, and Thomas Thaumuller. "Cultural aspects of global requirements engineering: An empirical chinese-german case study." 2009 Fourth IEEE International Conference on Global Software Engineering. IEEE, 2009.

[33] Ashraf, Muhammad Usman, Fathy Alburaei Eassa, Aiiad Ahmad Albeshri, and Abdullah Algarni. "Toward exascale computing systems: An energy efficient massive parallel computational model." International Journal of Advanced Computer Science and Applications 9, no. 2 (2018).