INFLUENCE OF THE NUMBER OF PEOPLE IN AGILE SOFTWARE DEVELOPMENT

Ashish Sanghvi[1], Parth Sehgal[2], Avnish Kapur[3], Suyash Singh[4]

[1]SCOPE, VIT University,Vellore, Tamil Nadu, 632014, India

Abstract: *With the increasing demand of agile methodologies, several software organizations are moving away from classical ways to adopt agile development methodologies. Rather than being fore telling, Agile is adjusting and people-focused. It advocates a tiny and cooperative team that tasks closely along. However team size may be an issue that's successively constrained by people factors. While implementing Agile, these key factors are usually unnoticed. This study aims at distinguishing the underlying people's factors to contemplate while adopting Agile for a team to be effective. The tactic utilized is the study of 3 completely different sized agile groups developing merchandise supporting identical technologies and utilizing scrum. Each objective and subjective measures were used and therefore the results are supported by a survey. The results clearly show that for agile methodologies to figure well, it's crucial to pick the proper people for the proper team.*

Keywords: Agile Methodology, Scrum, Agile Teams, Software Development, Project Management,.

1. INTRODUCTION

Agile project management may be an abstract framework in software engineering wherever software is constructed with a comparatively short amount of time and has many iterations that yield stable output of code (Hariyadi, 2011).As per the Agile declaration (2011), Agile is predicated on a group of principles that specialize in client price, unvaried and progressive delivery, intense collaboration, little integrated groups, selforganization and continuous enhancements. It's typically expressed that Agile Management works best with little groups. As per Bustamante and Sawhney (2011), the best Agile project team is little, collocated, communicate face to face on a routine and has a perfect team size not greater than 9 peoples. Beside this, agile ways like scrum advocate associate optimum team size of seven plus or minus two. But some business specialists claim that Agile may be a one size fits all methodology which it may be scaled up to a hundred and fifty person team. Normally Agile Management is long-faced with one key challenge: What should the optimal team size be for which Agile is to be applied for the team to be effective? Since Agile is incredibly individual focused, we ought to perceive how team size affects individual behavior and productivity at intervals in agile team and this rises to a different challenge. What people factors should be considered when managing an agile team to make it effective?

This paper analyses an Agile project management methodology, pilot it in observe with real world outcomes and aims at distinguishing the individual factors to be thought about for an Agile team to be effective. This study takes into thought some extent some psychological factors that may have an effect on team collaboration like memory, optimum expertise and suggested team size supported social behaviors. To attain this, three sample agile groups of various size are going to be used and managed with a similar agile principles. The results can facilitate verify that which team, providing all of them have all the required skills sets and similar team member profiles, which can work best and turn out the foremost optimum results. Agile recommends that, rather than having solely developers, agile groups ought to have all the specified skills sets. Although a few Agile methodologies like scrum have a suggested team size, it's a widely believed that Agile may be a conception of one size fits all and is thus scalable But the question then arises, however massive is just too much for an Agile team before it starts to lose its efficiency? In an exceedingly analytics performed by Ancona and Caldwell's (1992), the results indicated that team size might have an immediate impact on cluster performance and communication.

Agile Management is extremely individual focused whereas software system Project Management could be a rather technical discipline. Project Managers moving from a classical to agile strategies usually pay longer attention to managing outcomes instead of managing the individuals performing on the project. This suggests that human factors are usually not thought of once while managing Agile groups and that they might not be engaging at their best level. In spite of the growing quality of Agile Management (Agile pronunciamento, 2011) the speed at that which the software system are failing continues to be fearsome. As per the Standish 2009 Chaos Report (Standish, 2009), the common success rate is of all IT outcomes is 30%, four hundred and forty yards were challenged and therefore the failure rate is pure gold. The project success rate has remittent from previous years with solely thirty second being successful , whereby these were delivered on time, to budget and with all needed functionalities. 44% of the outcomes were challenged which implies that either they were delivered late, over budget, and/or with less functionalities that were supposed at the beginning. The remaining 24% of IT systems didn't mature and had to be put off before completion and were ne'er used. Furthermore in consistence with Scott W. walker (2010a), the success rates for Agile comes are 60% successful, 28% are unit challenged, and 12% are a unit failure. The success rate for Agile comes is 83% for little groups (less than eleven people), 70% for medium-sized groups (between eleven and twenty-five people), and 55% for big groups (more than twenty-five people). Based mostly from these figures it's clear that an agile team size includes a direct incidence on the success rate of the project.

The motivation for this paper is to produce an outline of the individual factors that are usually unnoticed while applying agile for setting-up of code development groups. In section 2, connected agile management concepts are mentioned. The analysis methodology is mentioned in section 3. Results and discussion from the study area are enclosed in sections 4. Section 5 presents a conclusion and a top level view for future work

2. LITERATURE REVIEW

This chapter analyses existing research and studies are performed on agile development teams.

2.1. Agile and People Management People and team management are implemented based on unit supported numerous models like "The 5 dysfunctions" (Lencioni, 2001) and Tuckman's model (Tuckman, 1965). All of those models need that team members have lots of interaction. There are a lot of persons that's there on a team, then a lot of interaction is needed and therefore the tougher it shall be to manage such groups. In associate agile team, the project manager should outline the relationships between the roles to alter the effective coordination and management of the project. The subsequent rules ought to be applied once process organizational structures:

- Make sure that every member of the team reports to 1 and only 1 person (the "unity of command principle")
- Make sure that every person has no over seven individuals reporting under him or her (the "rule of seven" principle) (Fayol, 1917).

However, people with totally different personalities are usually expected to figure along as a cohesive team. Team potency is usually obsessed on the interaction between team members and therefore the coordination of the team leader. As per studies, team potency is at its peak once team size is of 3 to 7 and it starts to decrease once team size goes on the far side of 9 members (Abilla, 2006). Agile groups additionally work best during a common space or "war room," that alter team members to figure out working under a shared surroundings. This approach facilitates communication and collaboration and has well-tried to an efficient means of accelerating team productivity. In a code project the target is to deliver a project on time, to budget and inside the in agreement quality level. Thus code project managers have the responsibility for the success of outcomes.
Managers ought to be arch leaders and sensible at organizing problem-solving sessions that alter maximizing collaboration across departments (Johnson, 2008).

"AGILE like several other ways ... look nice on paper however fail to figure in point of fact as a result of which they forget the human issue. Any paradigm, that has human interaction at its heart, can fail if human science isn't understood and be brought into consideration. The key aspects of attribute that IT development/project management ways need to take into consideration aren't any totally different to those at the center of most up-to-date economic theories", (Brady, 2006):

- individuals can perpetually place their own interests earlier than the interests of the cluster
- individuals area unit self-interested
- business production choices are a unit supported rational expectations
- Karl Popper's "First law of collective action". You'll never get over five individuals to agree on something."

Cognitive studies tend to clarify the state of mind of team members operating in a very cooperative surroundings. This can be addressed by the speculation of Flow (Csíkszentmihályi, 1990). Flow (Optimal Experience), is the rational state during which people in activity is completely absorbed, altogether focus, have full engrossment, and eventually acquires success within the method. The state of flow is earned once someone has each on top of greater challenge and skills in a very specific activity. Not solely Flow is vital to people however it also can contribute to team potency and structure goals. Flow usually results in higher productivity, innovation, and worker development (Csikszentmihalyi, 1990). Whereas it's not without delay possible to spot the state of mind of workers inside an agile team. A psychological analysis is also needed so as to see the way to accomplish the optimum expertise for people in an agile team. Alan Baddeley diagrammatic the model of remembering is clearly incontestable and represents the Short Term Memory works (Baddeley and Sala, 1996). He stipulates that the mind processes differing types of data in numerous ways. The human mind is proscribed in its capability to retain short term data associated with this will have a control on associate degree individual's performance in an Agile team whereby work progressed is unendingly dynamical and briefly periods of your time.

One of the foundation causes of failure in comes is communication — either a scarcity there from, or miscommunication. Massive groups square measure inherently vehicles for unhealthy communication. This can be supported the amount of communication channels that grow geometrically, not linearly (Abilla, 2006). 3|a 3} member team has three communication channels and a 5 member team has 10, double the amount of persons. But a six member team has the maximum amount as fifteen communication channels. This will be diagrammatic as below:

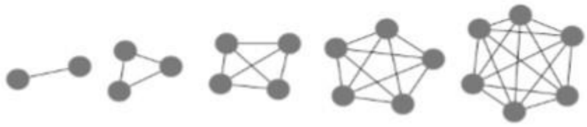

Figure 1. Overview of communication channel complexity (N x (N - 1) / 2)

Therefore the additional persons there's on a team the tougher it's to speak and share data among team members. So team size affects each potency and productivity of groups directly. For Agile to figure best every team member ought to have visibility of what the team is doing. Providing the human mind has such limitations as mentioned higher than then Agile might most likely work best with a little team size consisting of extremely driven people. Staffs have a way of happiness towards the corporate or the team if there's cluster project and social events and well-being. Recognition and appreciation of labour, promotion and job designation is that the accomplishment of esteem wants. Knowing that individuals reach best once they reach the esteem and self-creating by mental acts level, Agile Management might have to focus lots on guaranteeing that team members reach that level of satisfaction among the team.

In an analysis performed by Elizabeth Whitworth (Whitworth, 2006), she tackled the psychological facet of Agile development team members to work out their state once operating in associate degree Agile team. The outcomes of this study showed that Agile practices bring a precise level of motivation to people that create them tend to collaborate additional compared to classical strategies. This creates cohesion between agile team members and includes a positive impact on each personal and team productivity. The author doesn't come in detail to grasp the motivation and cohesion of agile team members supported the team size during which they need worked. This could be a key someone to work out the amount of satisfaction and motivation one has once operating in several team sizes.

In a study performed by Sudhakar et al. in 2011, it absolutely was found that there are a unit many soft factors that may have an effect on the performance of team members in a very software package of the development team. It's explicit that there are a unit four classes of things that have an effect on the performance of a development team: technical, nontechnical, structure and environmental.

The authors planned the subsequent soft factors that may have an effect on software package team performance:

- Team climate: shared perceptions and objectives to attain organizational goals.
- Team diversity: The variation of team member skills, levels of expertise, qualifications, gender and race as an example.
- Team innovation: new approaches to downside finding and price accessorial skills.

- Team member competencies and characteristics: Technical and private competencies of individuals on team that impact familiarity and collaboration.
- Team leader behavior: Less micro-management approach and additional people management and a help role.
- Prime management support: Commitment from management to the project

From this study, 2 key factors were found to be the foremost prestigious on package team performance, namely: trust and effective communication. The study is generalized and fails to spot the connection between the project management methodologies used with these soft factors. It should be that completely different people's factors influence the success of associate degree agile team.

Shane Hastie (2004) discusses how Agile differs from classical ways by golf shot rather than putting more stress on team work, cooperation and selforganization. One amongst the key to the success of Agile is trust that has to be there between each of the leader and also the team and amongst the team members themselves.

The paper will provides a smart insight on the factors which will influence the productivity of agile groups. But the factors weren't investigated with respect to team size associate degreed failed to build adequate thought on however this might influence people's factors in an agile team. As seen from previous studies, team size may be a key issue to the success of a package project. It also can influence individual's behavior and probably make to alternative soft factors which will want thought.

In a paper by Turner and Boehm in 2003, they have a tendency to look at the people's factors that influence each Agile and arrange based mostly on planned ways. It's been evidenced that individual's factors are the foremost essential success factors for development groups. There's enough proof from previous analysis in science of package groups that recommend that individuals factors have an immediate influence on the value. In a paper by Turner and Boehm in 2003, they tend to examine the people factors that influence both Agile and Plan Based methods. It has been proven that people factors are the most critical success factors for development teams. There is enough evidence from previous research in psychology of software teams that suggest that people factors have a direct influence on the cost and quality of software. In a software project, the key areas which are impacted by people factors are:

- Staffing: The right people ought to be chosen to work for the advancement of a product extend. In wide terms, it implies that the client ought to be spoken to by gifted staffs who are community oriented, agent, approved, submitted, and proficient (CRACK) (Turner and Boehm, 2003). The attributes of designers inside an agile group ought to incorporate agreeableness, ability, aptitude, and correspondence.
- Culture: Agile requires a genuine social change from plan based approach, not just a basic change in the procedures utilized.
- Values: One of the most disregarded difficulties in programming improvement is the combination also, estimation of prerequisites from all partners included.
- Correspondence: Agile requires more successive and serious correspondence, the more people on a group the additionally difficult and dangerous it gets to be to convey adequately.
- Overseeing Expectations: Software improvement groups regularly neglect to oversee desires and this can bring about issues between the groups and the client.

While the paper covers the significant territories affected by individuals' figures an improvement group and indeed, even say correspondence unpredictability and effect of group size, the creators don't investigate the perfect group measure that can influence the adequacy of an agile group. There is additionally constrained reference to the genuine individuals' figures that ought to be considered inside Agile yet more on the ranges affected.

In a study distributed in 2011 (Conboy et al., 2011) it was found that the expanding utilization of Agile methodologies and developing weight to embrace Agile Management, add to the requirement for human asset offices and project directors to address individuals challenges. There is a need to distinguish the issues that the agile move may bring about. A rundown of the most essential individuals challenges in Agile was proposed as takes after: engineer fear brought on by straightforwardness of expertise lacks, the requirement for designers to be an 'ace of all exchanges', expanded dependence on social abilities, an absence of business information among designers, the need to comprehend and learn values furthermore, standards of Agile, not only the practices, absence of engineer inspiration to utilize Agile techniques and the requirement for Agile consistent execution assessment.

There is one faulty issue however; the requirement for a dexterous designer to be multi gifted is relative. Albeit Agile suggests that the group ought to have all the aptitude sets required to execute the project, it doesn't essential imply that every single colleague ought to have this adaptability. This is liable to translation and rather, an agile group can comprise of various parts that give the ability fulfillment that agile backers.

3. PROPOSED METHODOLOGY

The accompanying section concentrates on the technique used to distinguish the general population calculates that can influence an agile group's execution.

3.1 Organization foundation

This study has been made with the advancement groups of an association that has practical experience in transportation and cargo administrations. The association is a noteworthy player in Europe and is completely operational over the Middle East, Asia, South America and Australia. The IT bureau of the association exist for well more than 30 years and has achieved a sound level of development in the programming industry. It has been utilizing a few improvement techniques including a completely tweaked Rational Unified Process and as of late moved to Agile Methodology utilizing Scrum. The product improvement groups are situated in Mauritius with the head office situated in the Joined Kingdom.

3.2 Investigation of Agile Teams

This study has been performed utilizing three distinctive estimated projects utilizing Agile Methodology over a time of six months. Not every one of the projects have taken the full six months for fulfillment; as some have been finished inside a shorter time span. Fulfillment is considered as the stage at which the project is effectively sent into creation and closed down by the client. In light of the topic of this study and to expand the rate of precision, no other parameters separated from the project size and group size were radically unique. Each of the three lithe groups had similar parts, expertise sets and utilizing the same agile procedures i.e.: Scrum. For privacy purposes, the projects utilized will be alluded to as A, B and C inside this study.

3.2.1 Project size

The tasks were at first evaluated as takes after:

Table 1. Overview of Project Size.

Project	Initial Estimate – MD
A	1800 (8.2 man yrs)
B	800 (3.6 man yrs)
C	120 (0.5 man yrs)

3.2.2 Design and Level of intricacy

Table 2 underneath frameworks the advancements utilized and the level of unpredictability for every project which has been assessed according to the association measures.

Table 2. Overview of technologies and complexity.

Project	Language/Technology	Complexity
A	Java Services, WebSphere Portal, Oracle, Spring MVC, Free-Marker	High
B	Java Services, WebSphere Portal, Oracle Database, Spring MVC	High
C	Java Services, WebSphere Portal, Oracle Database, Spring MVC, Free Marker	High

Every one of the three tasks utilized for this study have a similar engineering, comparative multifaceted nature yet diverse scale. The applications have been produced utilizing WebSphere Portal innovation for the User Interface and Java back-end administrations which thusly converse with Domain items and Oracle database.

3.2.3 Group Profile and Structure

Every group is lead by a Project Leader and the entire project is possessed by a Project Manager.

The table underneath gives a diagram of the different parts that were assigned inside every group:

Table 3. Overview of Project Teams.

Project	Business Analyst	Analyst Programmer	Test Analyst	Technical Architect	Total
A	2	12	6	2	22
B	1	7	3	1	12
C	1	2	1	1	5

3.2.4 Strategy for Measurement

A project result can be measured utilizing both target and subjective measures (Kemerer, 1989; Bahli and Bu¨yu¨kkurt, 2005; Ong et al., 2005). A group's execution can be measured by assessing whether the yield delivered is on calendar and on spending plan (Boehm, 1981). This incorporates measuring the nature of programming delivered and consistence to calendars and spending plan (Huckman et al., 2009). Nature of programming can be measured utilizing number of deformities found in testing (Boehm, 1981). With the end goal of this study, the target measures utilized are measurements based while the subjective measures depend on a group study and administration perceptions. The accompanying measurements were measured for every group.

Budget

The financial plan for every project is dispensed by the Project Board. This advisory group supports the financial plan in view of the business estimation of the project and the assessments as gave by the individual Project Manager. Real spending plan spend is ascertained in view of two measures: sum of work time signed in JIRA against every assignments and the measure of time signed in Clarity, the association time sheet framework. Worldwide Journal of Software Engineering and Applications (IJSEA), Vol.3, No.1, January 2012.

On Time Delivery

Extend arranging is performed on JIRA and it gives a guide diverse discharges and their arranged discharge dates. The genuine discharge dates are upgraded at whatever point the discharge goes out to the buyer. Toward the end of the project, any discharge due dates that are not accomplished can effortlessly be followed on JIRA.

DEFECTS

Deformities are arranged as non-satisfaction of prerequisites according to characterized documentation or undesired usefulness because of coding mistake. Deformities can have four sorts of seriousness:

- Critical: Resulting in an aggregate loss of the usefulness of the framework or an incomplete misfortune in a key range of the framework that makes the product unusable or temperamental.
- Major: Resulting in a center prerequisite not being satisfied and affect a fundamental practical stream or process. Frequently causes information disparity and disappointment of a few option streams.
- Blocker: A coding blunder or a pending change in light of necessities that effects the substitute streams of a necessity and keeps the use of a specific usefulness which is viewed as low effect.
- Minor: An insignificant deformity that can relate to graphical UI, fundamental approval with non-breaking impact.

Adjust Level

Modify is computed as the real man-days spent on a project to settle surrenders that have been raised by framework or reconciliation testing. This figure is then partitioned by the financial plan real to decide the rate of revise:

$$\text{Rework Level} = \frac{\text{Time spent fixing defect}}{\text{Time spent on the development}} \text{ X } 100 \qquad (1)$$

JIRA is used as the tool to track defects and time spent fixing a defect. This provides the data to calculate rework level.

Deformity Rate

Deformity rate depends on the quantity of imperfections identified on a product over the aggregate sum of genuine time spent on the advancement of the product. For example: Number of days spent on a project: 100 man-days Add up to number of deformities found on the
project: 150
Deformity rate of the group: 1.5 So for each
day of work delivered, around 1.5
imperfections are presented.

Release and alter Management In a team wherever the amount of problems mounted or functionalities being developed on a daily basis are high, nightly builds facilitate in distinguishing any potential problems with the code as early as possible so a stable release is formed obtainable to the test team as the finish of every sprint. This has been the case principally for Project B whereby a high range of defects were raised by test team and required to be fixed as shortly as doable. Table four provides a summary of the release frequency for every project furthermore that the frequency for testing cycles and also the forms of testing that were done.

Table 4: Release Frequency.

Project	Sprint Length	Release Frequency/Test Cycles
A	One Week	In the end of each Sprint – System Testing Every month – Integration Testing
B	Two Weeks	Every Day – Automation testing In the end of each Sprint – System and Integration Testing
C	Two Weeks	In the end of each Sprint – System Testing At project completion – Integration Testing

Changes in every project are managed through the organization modification method that consists of the following stages:

- Initiation – Changes are documented and raised by the Business Analysts and created on JIRA
- Analysis and estimation – This stage involves the approval from Technical designer for the change to be performed from an design perspective. Changes are then reviewed by the development team with the risks and impact assessed and estimates are made.
- Approval – Business Analyst and Project Manager approve each budget and coming up with on the change
- Implementation and delivery – Changes are planned within the acceptable Sprints and delivered as per agreed timelines and budget spends. Whenever changes are approved and implemented, the calculable prices are added to the general estimate of the project.

Questionnaire Design

For the purpose of this study, a questionnaire is developed to receive feedback in 3 key areas, namely Management, People and Product. A copy is attached in Appendix A.

- Section A – Agile Management. This section consists of 5 questions and aims at understanding how the team members felt regarding the manner their various projects were run from an Agile Management perspective.
- Section B – Team Work. This section of the form can facilitate understand the mindset of the team members and motivation. It'll additionally facilitate in distinguishing some key people factors that are mentioned in the previous analysis.
- Section C – Product. The questions in this section are aimed to grasp the extent of challenge that the individual has got to face within the project and overall perception of the end product to determine the satisfaction level.

Each team members are going to be required to fill within the questionnaire at the end of the project and the data are going to be collected on a per project basis.

4. RESULTS AND DISCUSSION

It should be noted that the results that came out of this analysis mustn't be thought-about as wide practice or as a regular across the software industry. But they do provide an insight of what's the most effective possibility for an agile team size and also the factors that require to be thought of.

4.1 Budget

The budget spent on each projects are collated following their completion from tools such as JIRA and Clarity. The results are:

Table 5. Budget.

Project	Initial estimate(MD)	Budget Spend(MD)	Percentage Deviation
A	1800 (8.2 man yrs)	3200 (14.1 man yrs)	66
B	800 (3.6 man yrs)	950 (4.3 man yrs)	18
C	120 (0.5 man yrs)	130 (0.6 man yrs)	8

As seen within the table above, the over spend has been largest in Project A with a deviation of over 65%. This has been principally attributed to the subsequent factors:

- Team Communication and Collaboration: There was very little interaction between the 2 Agile sub-teams at the beginning of the project partly because of lack of familiarity between the team members. This has caused many dependant functionalities not properly developed.

- There was lack of clear communication of requirements and queries raised by the Mauritius team weren't being answered in a timely manner by the United Kingdom team. This didn't allow smart relationship to be developed between each agile sub groups early within the project life cycle. The problem was partly resolved by having every member of the Mauritius team travelling to the United Kingdom workplace to figure alongside the onshore agile team for a given amount of time that ranged from four to six weeks. The • collaboration and communication gradually increased however a lot of time was spent fixing problems that were already raised during the early stage of the project. This had a major impact on the project budget as not only time was needed to fix unexpected issues however also to make sure correct integration when these fixes were made.

- Trust: it had been discovered that there have been trust problems amongst team members. •
In some situations, people wouldn't assign tasks to others on the idea that they didn't think the other person could complete the work in an efficient manner and on time and hence took up these tasks themselves. This has caused delays on sure areas.

- Team Interaction: every sub team had to carry • its own daily meeting and pass any information relevant to the opposite team through the project leaders. But the coordination between the 2 groups wasn't invariably straightforward. It was not very sensible to carry the daily scrum meetings more than often concerning an hour was spent within the daily updates. Some members who had already provided their updates for the day were keen to depart the meeting to be able to continue working on their allotted tasks. They didn't think that they the information being discussed was relevant to them and felt that they were being unproductive.

- Code Integration: There are a large range of code integration problems between the two agile sub groups that caused lots of rework to make the integration of components work evidently. Every team were playing their own daily Scrums and aiming at completing tasks allotted to their respective team. But there was very little visibility on what the opposite groups was doing, mainly because there wasn't enough time for team members to dig into the tasks allotted to the opposite team. There have been just too several tasks for a team to appear into the opposite groups work basket. Lots of functionalities weren't in synched, for instance components didn't match the expected behavior owing to developers making changes in their component without correct evaluation of impact on its dependencies or didn't keep in mind communicating the changes to everybody or perhaps did not communicate data to those who they do not get on that well.

Last minute defects: Integration problems were typically uncovered at the last minute when performing a release as each sub Agile groups would begin integrating their work as least one day before a planned release date. This was performed on a weekly basis and a release made to system testing team. But since the problems were uncovered late, this caused lots of process and quite a range of releases got delayed.

Defect Rate: the actual fact that lots of functionalities were being delivered on a weekly basis meant that continuous testing had to be performed. This in turn gave rise to a significant amount of defects to be raised and need fixing. The impact was that rework level increased and wedged on the budget.

Requirements Clarification: the rate at which the groups were operating was quite high. But Business Analysts were unable to produce clarifications on time for queries that were raised to them. There have been too many queries raised and many individuals raising them. The work time eventually impacted the time to fix problems and delay releases. Despite the fact that clarifications were obtained, they weren't consistently passed on to the opposite members of the team chiefly because developers would typically forget to incorporate everyone in their communication or they might suppose that the knowledge wouldn't be relevant to fellow colleagues.

Management views on Project A are that's was difficult to manage the team using Agile approach and there have been several instances where team members focused principally on their own deliverable instead of team output and forgot to pass info to everybody. Project B was eighteen over budget and this was principally attributed to:

- Communication and Collaboration: Communication wasn't effective within the team as a number of team members wouldn't share info obtained from the Business Analysts either because they have a tendency to forget to pass the data to any or all team members or they thought that the data wouldn't be relevant to the other areas being developed. Based on this, quite a number of requirements weren't properly implemented and required some rework.
- Requirements: a number of the development team members weren't acquainted enough with the business processes that were being implemented and so more time was spent in requirements clarifications with the business analyst. There have been also very little or no documentation for some areas of the project and therefore the information resided with some key team members only.
- Quality {of info|of data|of knowledge} shared: There was usually inaccurate information being provided when clarifications were requested. This caused a large variety of defects to be raised by testers as they'd check per documentation out there whereas developers would perform development supported information and knowledge gathered from the Business Analyst.
- Conflicting views: Daily scrum meetings would sometime overrun as it was difficult to get agreements on some problems and this meant that separate meetings had to be held to confirm that a agreement is reached in order that development might run smoothly.

Management views are that the project was delivered late and over budget however it matched all requirements requested by the business.The budget deviation for budget for Project C was around 8 may 1945 and the main causes were:

- Underestimated areas of work: owing to the very fact that needs weren't properly analyzed at the beginning of the project, some areas of development weren't properly estimated and therefore the impact on sub parts not known. Throughout development the impact became obvious and impacted to initial budget.

The allowable deviation from budget for all projects based on the organization standards is 10%. Thus solely Project C was within the standards whereas Project A was well outside.

4.2. On Time Delivery

The rate at which releases were performed on time was measured as follows:

Table 6. On Time Delivery.

Project	Releases performed	Releases on time	Percentage on time
A	40	12	30
B	25	15	60
C	10	9	90

From Table six, it will be seen that Project A had the biggest number of releases and owing to the fact that most of them were performed late and testing and deployment were successively delayed. The project itself wasn't implemented on time and manner off its target delivery date. The main reasons for late deliveries were the massive number of defects that were raised due principally to integration and coding issues. Project B conjointly featured some similar problems as Project A. Releases weren't on time owing to integration issues and also lack of coordination with external groups such infrastructure meant that environments were often not operating as expected. Deployment and configuration were not performed in a good manner because the team were centering primarily on completion of pure development tasks as a priority and sometimes meant that putting in place of environments were requested late. The organizations standards allow for a minimum of 90th on time delivery for all projects. Based on the data collected, solely Project C satisfied this criterion.

4.3 Rework

The rework level for each project has been calculated as below:

Table 7. Rework Level.

Project	**Actual Spend – MD**	**Rework – MD**	**Rework Level - %**
A	3200 (14.1 man yrs)	800	25
B	950 (4.3 man yrs)	150	16
C	130 (0.6 man yrs)	20	15

The rework on project B and C were low but were more than required range as per organisation standards which is around 10%. Highest amount of time was spent on Project A fixing defects against the total time spent on development the software. 800 out of 3200 days were spent on pure defect fixing, which is a quarter of the total budget.

4.4 Defect Rate

The defect rates recorded have been as follows:

Table 8. Defect Rate.

Project	**Actual Spend**	**Defects Raised**	**Defect Index**
A	3200 (14.1 man yrs)	1633	1.9
B	950 (4.3 man yrs)	305	3.0
C	130 (0.6 man yrs)	40	3.0

Project B and C have higher defect index than A. In project C, a new defect was obtained for every 3 days of work whereas for project A, a defect was produced every 1.9 days.

Based on the statistics for rework level and defect count, in general, half day is required to fix a defect on all three projects, i.e.:

- Project A: 1633 defects requiring 800 man days rework
- Project B: 300 defects for 150 man days rework
- Project C: 40 defects for 20 man days rework

This indicates that the extent of quality for of these projects is comparable and therefore confirms that the projects used were ideal candidates for this study.

4.5 People Aspect

During this study, except the worker survey and subjective measures like observations, soft factors of the agile groups weren't measured with any tools or methods.

As a part of the instinctive measures, some observations were created on how the people behaved during the lifetime of the projects. Specific attention were created to make use of individual's best practices of management during all the three projects. A number of problems were encountered in some domains because of inherent quality in managing specific groups. Such issues couldn't be quantified on a scale but from an objective view only as described below:

- Authorization: All people required on the groups were given satisfactory specialized preparing where required and the power to make dependant JIRA assignments where fundamental. These undertakings were then either appointed to other colleagues or to outside groups to advance. Initiators of such undertakings were required to screen the advance until fruition.

- Supervision: The groups were given guiding when required and blockers or hindrances that were raised instantly amid the everyday Scrums must be advanced either by the Project Leader or Project Manager. This guaranteed administration went about as a facilitator to the groups while they concentrate on the conveyance of the item.

- Developer Skills and Fear: On project A three of the developers who were bad communicators did not raise concerns with respect to specialized zones where they not by any means great at. This was ascribed to the way that they would not like to show in the day by day Scrum gatherings that they were in fact behind the other colleagues. This prompted to additional time being spent on errands allocated to them. A similar issue was experienced on Project B whereby one developer did not raise his worries. In any case, on Project C there were no such issues and when engineers required training, this was raised immediately with no dread in day by day Scrum gatherings. In view of this response, no doubt people dread to bring their aptitude inadequacies up in an extensive gathering while this makes no worries when in a much littler gathering.

- Liability: Members of all 3 groups got clear tasks possession which was monitored via JIRA effortlessly. A corresponding JIRA price tag was created and allotted to a member of the team for every assignment of the project. During this manner there was clarity of who does what and who owns what. About 20% of the tasks of project A, that changed ownership, as developers were not sure which sub teams had the responsibility to produce the work. This implied administration needed to intercede to dispense such errands to the proper groups or people. There was a reasonable absence of clarity and correspondence that was exhibited by the way that some colleagues did not have an unmistakable learning of who does what or who claims what.

- Performance management: JIRA gives the office to see a project's advance and status by means of a dashboard. This was checked on an everyday premise to guarantee that the individual and thus the group is executing not surprisingly and any inconsistencies were promptly noticeable and remedial moves could be made. For example if a colleague were to remain too long on a JIRA ticket this will be seen on the dashboard as beneath and empower provoke move to be made amid the everyday Scrum gatherings.

- Successful Communication: The fundamental part of the Project Leader was not just to guarantee that any blockers from outside conditions are advanced however too to

  guarantee that all data got were fell in a viable way to the group. Changes in necessities were constantly fell and general input on how the group is performing was given.

As prescribed by Agile, the project groups worked in a mutual workspace in an open office structure that went for encouraging correspondence and joint effort. This was not effortlessly achievable with project A whereby the two sub Agile groups were situated in various topographical areas consequently obstructing correspondence to some degree.

The way the colleagues cooperate were some way or another diverse over the three groups and there were a few perceptions made:

- Association: At slightest five colleagues from Project A, in spite of the fact that their work had some reliance on that of others, were seen to work in solo and would not give enough data to other colleagues on the effect of their work on other areas. They were extremely focussed in the conveyance of their own errands on time and inside the evaluations even at the cost of not guaranteeing that joining is mulled over, in any event not until the very late when it is really required. This was likewise seen with a slightest two individuals from group B. However for group C the camaraderie was very extraordinary and individuals were imparting effortlessly. On project B a similar conduct was seen with just two colleagues who did not work together successfully and bringing on their finished work to be incongruent with those of other colleagues. However no such issues were raised from Group C.

- Commonality of purpose: This normal for an Agile group was seen in both Projects B and C yet as specified above there was insufficient inspiration for the gathering to function as one group for project A. There was a great deal of independence in whereby a portion of the designers were focussed on conveying their own particular work as opposed to seeing the general picture and where it fits in the entire project.

- Discernment of a group: While this was effortlessly acknowledged in Project C and Project B, it was significantly more troublesome in Project A. The way that there were two sub Agile groups, every part went about as though they were their very own piece sub group and couldn't without much of a stretch fit themselves in the general Agile group. This implied the groups worked autonomously from each other and frequently prompted to correspondence holes and reconciliation issues. The majority of the colleagues who were situated in the UK were emanant pioneers and frequently took responsibility for and eventually even framed smaller than expected groups.

- Group Interaction: Albeit the vast majority of the groups had guide up close and personal correspondence once a day, some colleagues from Project A and B would make the utilization of visit apparatuses to speak with each other amid the day regardless of the possibility that the other was in a similar office inside a couple meters. This implied correspondence was kept to strict least between these people, just when obliged to do as such through group gatherings. For Project C the connection was vastly improved and the group reinforced all the more effectively. Individuals would collaborate socially by going out to lunch and convey verbally frequently amid the day and this enhanced their nature. The general perception was that for groups A and B, the greater part of the colleagues did not permit much time to build up the social connections between each other yet rather focused on the deliverables and regularly appeared to work under stretch. Agile has a high dependence on social abilities.

- Inspiration: The work on Project B was sorted out in a manner that particular ranges of improvement were doled out to a colleague. One of the issues experienced however was that some colleagues convey what needs be on the way that they didn't feel that the work designated to them were testing enough.

4.6 Communication Channels

One of the key difficulties that the agile colleagues needed to face was the communication unpredictability. For Project C, it was not troublesome for the colleagues to impart viably. However for groups A and B this was a great deal more troublesome.

Table 9. Communication Channels.

Project	No of people on team	No of communication channels
A	22	231
B	12	66
C	5	10

The above table plainly demonstrates 231 correspondence channels for Project A which gives a knowledge on how troublesome and poor correspondence could have been among the colleagues. Spry standards require than colleagues work in serious coordinated effort and have up close and personal correspondence. This was accomplished with no issue on Project C whereby the group was little and demonstrated exceptional cooperation. In project B, regardless of the possibility that the group worked firmly together there were occasions where colleagues will demand to skip gatherings in order to finish their undertakings. In project A however eye to eye correspondence was not generally conceivable between the two topographically scattered groups even with the utilization of Video Conferencing frameworks. It was hard to arrange the groups to cooperate and the separation did not permit coordinate correspondence.

5. CONCLUSIONS

This study has been made solely in view of the Agile Methodology and inside an association that is making utilization of Scrum. It didn't mull over any tasks utilizing different sorts of advancement procedures. Subsequently it is basic that before any association chooses to embrace an Agile Method, it needs to evaluate whether the organization culture, working structure, business procedures and projects are suited for the utilization of an Agile Extend Management Methodology. This study was gone for assessing the general population figures that can influence the execution of an agile group. The exploration was made with a little gathering of comparable activities that varied by size and group measure as it were. This work will give significant understanding to group pioneers to choose the suitable number and kind of individuals on every project. It will likewise be exceptionally valuable to programming association which will be in a superior position to evaluate whether lithe could work for them. Future work should be performed to assess whether the same result can be gotten with activities of differing association, multifaceted nature and innovation.

6. REFERENCES

[1]. Abilla, P., 2006. Team Dynamics: Size Matters Redux. [Online]. Available: http://www.shmula.com/team-dynamics-sizematters-redux/182/. [Accessed: 27th April 2011]

[2]. Agile Manifesto (2011). The Agile Manifesto. Online]. Available : www.Agilemanifesto.org. [Accessed: 10th August 2011)

[3]. Ambler, S.W., 2010a. 2010 IT Project Success Survey. [Online]. Available: http://www.ambysoft.com/surveys/success2010.html. [Accessed: 1st April 2011]
International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.1, January 2012
135

[4]. Ancona, D.G., Caldwell, D.F., 1992. Work-groups; Organizational-climate; Organizationaleffectiveness. Administrative Science Quarterly. 37(??)p. 634-65.

[5]. Baddeley A, Della Sala, S., 1996. Working memory and executive control. Philos. Trans. R. Soc. Lond., B, Biol. Sci. 351 (1346): 1397–403.

[6]. Boehm, B., 1981, Software Engineering Economics, Englewood Cliffs, N.J., USA, PrenticeHall.

[7]. Bahli, B. and Bu¨yu¨kkurt, M.D., 2005. Group performance in information systems project groups: an empirical study, Journal of Information Technology Education, Vol. 4.

[8]. Bustamante, A., Sawhney, R., 2011. Agile XXL: Scaling Agile for Project Teams, Seapine Software, Inc.

[9]. Brady, K., 2006. AGILE/SCRUM Fails to get to grips with Human Psychology. [Online]. Available: http://www.claretyconsulting.com/it/comments/Agile-Scrum-fails-to-get-to-grips-withhumanpsychology/ 2006-08-17/. [Accessed : 3rd August 2011]

[10]. Conboy, K. Coyle, S. Lero, X.W. Pikkarainen, M., 2011. People over Process: Key Challenges in Agile Development July/August 2011 (vol. 28 no. 4) pp. 48-57

[11]. Csíkszentmihályi, M., 1990. Flow: The Psychology of Optimal Experience. New York: Harper and Row, ISBN 0-06-092043-2.

[12]. Fayol, H., 1917. (French), Administration industrielle et générale; prévoyance, organisation, commandement, coordination, contrôle. Paris, H. Dunod et E. Pinat.

[13]. Hastie, S., 2004. The Agile Mindset: what does it take to make this stuff work? Software Education Associates Ltd, Agile Development Conference Wellington & Sydney, September 2004

[14]. Huckman, R.S., Staats, B.R. and Upton, D.M., 2009. Team familiarity, role experience, and performance: evidence from Indian Software Services, Management Science, Vol. 55 No. 1, pp. 85- 100.

[15]. Johnson, R., 2008. Six Principles of Effective Management. [Online]. Available:

http://ezinearticles.com/?Six-Principles-ofEffective-Team-Management&id=1803062.
[Accessed: 28th March 2011]

[16]. Kemerer, C., 1989. An agenda for research in the managerial evaluation of computer-aided software engineering (CASE) tool impacts, Proceedings of the 22nd Annual Hawaii International Conference on System Sciences, Hawaii, pp. 219-28.

[17]. Lencioni, P., 2002. The Five Dysfunctions of a Team. Jossey-Bass

[18]. Ong, A., Tan, G.W. and Kankanhalli, A., 2005. Team expertise and performance in information systems development projects, Proceedings of the 9th Asia Pacific Conference on Information Systems, Bangkok, Thailand, July 7-10.

[19]. Standish Group., 2009. Chaos Report 2009. [Online]. Available from: http://www1.standishgroup.com/newsroom/chaos_2009.php. [Accessed 20th May 2011].

[20]. Sudhakar, G.P. Farooq, A. Patnaik, S., 2011. Soft factors affecting the performance of software development teams. Team Performance Management Vol. 17 No. 3/4, 2011 pp. 187-205

[21]. Tuckman, B., 1965. Developmental sequence in small groups. Psychological Bulletin 63 (6): 384–99

[22]. Whitworth, E., 2006. Agile Experience: Communication and Collaboration in Agile Software Development Teams. M.A Carleton University, Canada.

[23]. http://airccse.org/journal/ijsea/papers/3112ijsea09.pdf