Various Universities

Mamadou Mouctar Diallo, Research Scholar
Déportement of Computer Science & Engineering
Vivekananda global University, Jaipur

***Abstract*— In light of this context, this thesis addresses the research question of how agile software development process can be scaled up within the context of various university. Agile development approaches build software that meets the customer's or user's needs. For many software applications,) that is usable adds value for the users. Recently, there has been some evidence that suggests that practicing agile methods alone does not ensure that an application's is usable for the user. As a result there has been interest in combining agile methods with niche model. This research presents the results of a qualitative empirical study to contribute to an understanding of how these two methodologies are being effectively combined, The results present a general process model, we call the niche model. Which will contribute to accomplish the project objectives, and will allow the students to approach development methodologies since the beginning of their studies.**

Keywords*—Niche model, Agile Methods, Software Application etc.

I. INTRODUCTION

In a cutting edge economy, smart items and administrations, just as the digitalization of data and substance, cause an expanding requirement for programming. While we have seen the abilities of microprocessors developing exponentially, the efficiency of programming improvement is just developing directly. This marvel, making programming ventures bigger, longer, and progressively mind boggling, was first distinguished in 1968, and is known as the Software Crisis. As it is characterized, programming frameworks take too long to even think about developing, cost excessively, and don't work great. There have been numerous endeavors to take care of this issue, fluctuating from specialized arrangements, for example, third era programming dialects that raise the dimension of reflection to different procedures and techniques, for example, object-situated dialects and even open source advancement. From the item improvement perspective, as items turn out to be increasingly unpredictable, the task intricacy expands, making ventures subject to an ever increasing number of complex
issues. Organizations endeavor to battle this unpredictability by employing experienced directors, just as building information inside the association. Nonetheless, we ought not think little of different angles affecting on progress, for example, the estimation of talented work force. A few examinations show an extent of contrast in execution between the most and least talented software engineers.

One of the real venture the executives choices, at that point, is the determination of the task's product procedure demonstrate. A fitting spry procedure model could help in adapting to the difficulties, and forestall numerous potential venture dangers and issues. The product business has turned out to be exceptionally aggressive. Hence the capacity to execute and test the correct highlights rapidly has turned into a key ability. On the off chance that changes to programming prerequisites occur at a steady rate and likelihood, amid a long improvement program there will be more strain to change the course of the program, i.e., a more drawn out advancement program will really require the capacity to change course easily significantly more than shorter ones. Customary necessities stop and change the executives will turn out to be excessively exorbitant and moderate. Organizations must purpose the hole of efficiency with a few methods; for instance with procedure models that empower them to wind up increasingly receptive to the inescapable changes. 30 In the '90s, a few new lightweight programming process models, for example, XP, Scrum, and Feature-Driven Development rose. These were later known as Agile Methods. The inception of the expression "light-footed" goes back to 2001, when a few of the designers of these new strategies met and examined these new wonders, just as the present condition of programming improvement at the time. The substance of these models was the capacity to grasp change, yet there is additionally more in these strategies, for example, the emphasis on individuals and correspondence. Schafer, among numerous others, contend that Agile Methods yield a few advantages. These incorporate decreased time-to-showcase, expanded quality, diminished waste, better consistency, and better assurance. Among other recommended advantages is, for example, an expanded capacity to react to dynamic market transforms It is remarkable that the two most generally utilized Agile Methods, to be specific Extreme Programming and Scrum both began from the inclination to take care of the profitability issue, a.k.a. the Software Crisis. Scrum can be considered to have been started at the Easel Corporation in 1993. It depends on the examinations on elite groups in the report by Borland, Takeuchi, and Nonaka (1986) in the Harvard Business Review, which proposes that ventures utilizing little, cross-utilitarian groups produce the best outcomes Sutherland' sex existences at the robot Corporation affected on how these key people were considering, i.e., there existed a comprehension and experience of Complex Adaptive Systems and the developing conduct behind Scrum. Outrageous Programming (XP) was conceived at DaimlerChrysler when Kent Beck endeavored to locate a superior method for doing programming advancement around 1990.

.

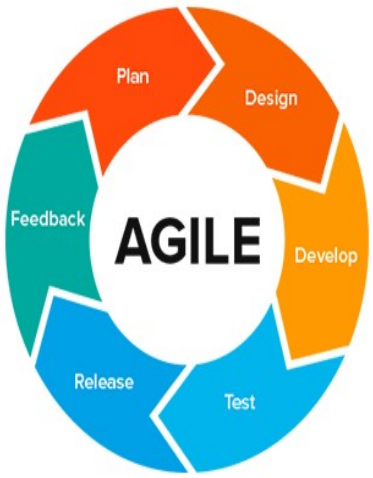

Figure1.Agile software

II. LITERATURE SURVEY

Kupiainen, Mika, and Itkonen (2014) introduced fundamental outcomes from a methodical writing audit. As indicated by their examination, measurements are centered around the accompanying regions: Iteration Planning, Iteration Tracking, Motivating and Improving, Identifying Process Problems, Pre-Release Quality Post-Release Quality, and Changes in Processes or Tools. At the point when Agile standards were contrasted or mapped and the discoveries, it was discovered that the measurements upheld the standards with a few deviations. Shockingly, little proof was found of the utilization of code measurements while there is much proof on the utilization of arranging and following measurements. It was likewise noticed that the utilization of measurements to propel and uphold process upgrades, just as material quality measurements, can be fascinating points for future research.

Fitzgerald, Stol, O'Sullivan, and O'Brien (2013) recognized the fundamental qualities of Agile methodologies and showed through an itemized contextual analysis how an Agile methodology was executed effectively in a managed domain. Among the intriguing ideas that rose up out of the examination were the thoughts of ceaseless consistence and living discernibility. The purpose behind this can be followed to the Agile Manifesto, which recognizes the four central offers for Agile to be .

- Individuals and their associations are esteemed in abundance over instruments and procedures.

- Functioning programming is favored over total documentation.

- Collaboration with the client could really compare to getting an agreement arranged.

- Reacting to change instead of inflexibly following an arrangement.

Harish, Madhu, and Lokesha (2012) played out a total report utilizing the methodology for Agile programming testing. The methodology for Agile programming mechanization testing and the prescribed procedures to be pursued to convey an ideal working item to the client were talked about. The paper incorporates the prescribed procedures to consider before executing computerized programming testing. These accepted procedures are key and are relevant paying little respect to the mechanization device utilized and give a reasonable way to test the product item in the Agile advancement process.

Melo, Cruzes, Kon, and Conradi (2013) directed a various contextual analysis for a half year in three huge Brazilian organizations that had been utilizing Agile strategies for more than two years. They concentrated on the fundamental profitability factors seen by colleagues through meetings, documentation from reviews, and non-member perception. A theoretical structure was created utilizing topical examination to comprehend the plausible procedures behind such efficiency factors. Nimble group the board was observed to be the most compelling element in accomplishing Agile group efficiency. At the intra-group level,B the fundamental profitability factors were group configuration (structure and work portion) and colleague turnover. At the between group level, the fundamental efficiency factors were the means by which well groups could be viably planned by legitimate interfaces and different conditions in this manner dodging delays in giving guaranteed programming to subordinate groups.

Moniruzzaman and Hossain (2013) fundamentally distinguished and depicted the main considerations to demonstrate that the Agile advancement approach enhances programming improvement procedure to meet the quickly changing business conditions. They likewise gave a concise examination of Agile advancement strategies with conventional frameworks improvement procedures, and talked about the present condition of embracing Agile approachs. They guessed that Agile programming advancement has risen as an option in contrast to customary arrangement based programming improvement strategies from the need of fulfilling the client right through ahead of schedule and unremitting conveyance of the valuable programming. The reason for this paper was to give a top to bottom comprehension of the significant advantages of the Agile improvement way to deal with the product advancement industry, and furthermore give a correlation of Agile programming advancement approaches over conventional programming improvement techniques (TSDM).

Pathak and Saha (2013) depicted another sort of model for programming advancement, i.e., Agile programming improvement. In this paper, they introduced their investigation of three Agile programming forms. They additionally depicted the issues looked amid the execution of Agile programming advancement. Their goal was to help programming engineers comprehend the key qualities of these procedures and in this manner select the most appropriate procedure as for the sort of programming ventures created by them. As a future exercise, there is a

need to survey other Agile procedures not shrouded in this paper, for example, Lean Software Development (LSD) and Dynamic Systems Development Method (DSDM).

Imreh and Raisinghani (2011) endeavored to reply in their work. Lithe programming improvement, the spearheading "advancement", significantly affects organizations which center around programming improvement and furthermore speaks to a couple of hindrances that at last influence quality inside IT organizations. This examination explores the impact of Agile programming improvement on quality inside the legitimate, exact, and social structures of IT organizations and gives industry standard proposals to alleviate such impacts.

Singh and China (2012) made norms that have their place in business, for instance, Agile programming advancement, reuse based improvement, and part based improvement. Then again, different programming headway models disregard to satisfy the different needs of the item business. All strategy models go for achieving thing quality, diminishing time for advancement, expanding benefits, and diminishing costs, yet no singular technique exists that is finished. The programming business is moving towards Agile programming improvement. Facilitated headway does not plainly fit well to fabricate reusable ancient rarities. On the other hand, with wary thought, and imperative modifications made to Agile procedures, it may be possible to adjust successfully and put on Agile techniques for the advancement of reusable items. The model proposed here joins the traits of Agile programming improvement and reusability.

Qwaider (2012) concentrated on Agile in programming improvement techniques and practices. Be that as it may, powerful techniques for use got less consideration. This paper examines the reception and dimension of understanding of the utilization of Agile practices in three programming improvement organizations in Jordan.
Zhang and Dorn (2012) contemplated Agile practices in a little scale, time-concentrated web advancement venture at a school level IT rivalry. In light of the perception of the advancement procedure, meetings of the venture groups, and the investigation of pertinent records, they depict how Agile practices, for example, day by day scrums, accumulations, and dashes, were effectively received to extend improvement. They additionally depict a few supporting exercises that the group utilized, including cross-leveling of information, socialization, and different correspondence modes. At long last, they talk about the advantages and difficulties of actualizing Agile practices for the situation venture announced, just as commitments and confinements of their discoveries.

Agarwal and Majumder (2011) portrayed Agile undertaking the board, a basic factor that comes while executing Agile system, genuine Agile condition, and Agile task arranging. Unimportant information of the advancement procedure isn't adequate for compelling programming improvement. Knowing moreover how to deal with the procedure will guarantee that the procedure is done successfully. For instance, while arranging a get-away, simply having a point by point schedule of spots to visit isn't adequate.

Sousa and Almeida (2011) concentrated on the scholarly setting: the Laboratory SAPO/UA, explicitly in the SAPO Campus venture. The working of this specific situation, right now diminished in the executives conventions, was investigated to additionally build up an Agile venture the board show. The structure show was actualized among the undertaking group and therefore approved. Thus, they expect a genuine enhancement of the task and a social event of contentions that reinforce an unmistakable viewpoint on the adjustment of Agile venture the board to the setting of sight and sound generation

III. AGILE METHODS

A few creators have introduced diverse models of an Agile methods. Leffingwell (2007 and 2011) exhibited The Agile methods Big Picture;

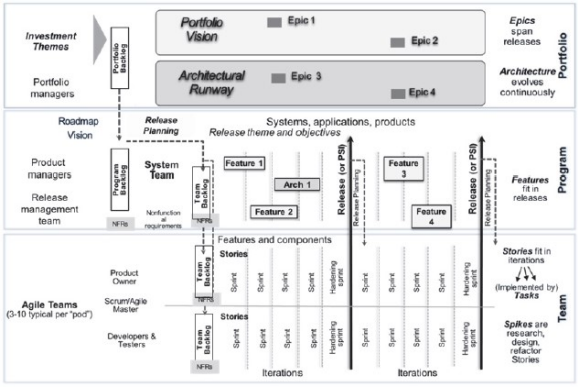

Figure2. Agile methods

IV. THE SCOPE OF THE RESEARCH

This paper focuses on the applicability of Agile Methods, such as Scrum and Extreme Programming, in large-scale software development university, and on developing both a model for an Agile Enterprise and a model by which the adoption of the above-mentioned methods can happen on a large scale. In this context, a large scale means several thousand software developers. The selected viewpoint is how to create university abilities for the application of these selected Agile Methods and not specifically the impacts obtained. The most important Agile Methods applied were Scrum (Schwaber 2004 and 2007), scaled Scrum, as defined by Leffingwell (2007 and 2011), along with Continuous Integration (Humble and Farley 2011), and elements of Lean Software Development (Poppendieck 2003 and 2010), Agile Portfolio Management (Vähäniitty and Rautiainen 2008), and Practices for Scaling Lean & Agile Development (Larman and Vodde (2009 and 2010).

V. PRINCIPLES OF AGILE SOFTWARE

The twelve principles of agile software are:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity--the art of maximizing the amount of work not done--is essential

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

VI. APPLICATION SOFTWARE

These are set of programs that are designed to carry out operations for a specific application (itsavy.com). These applications allow end user to accomplish one or more specific non-computer –related tasks. Application software ranges from word processing software, database software, educational software, computer games to even medical software(osait.com)

VII. PROBLEM STATEMENT

The problem statement is: “To develop a thinking framework for achieving three things namely:

1) Characterization of software development projects in order to determine the suitability of using agile software development methodologies in a given project,

2) Selecting the most appropriate set of practices from the family of agile methodologies,

3) Tuning the selected set of practices into a feasible way of working ” Based on the problem statement the following section explains the process and route to be followed for the successful implementation of the proposed solution.

VIII. RESEARCH METHODOLOGY

Research Methodology This section outlines the research approach followed in this thesis. The research starts by building theory through a literature survey of existing methods of project characterization in order to determine:

- The purpose of project characterization
- Ascertain the existing methods of agile methodology selection
- Establish the need for a methodology selection technique for matching a given project to a set of agile methodology practices. As part of the theory building stage the author also participated in a project at a software development company. Knowledge gaps were identified and research questions described in this chapter were formulated. It is then followed by the building of a proposed solution to the identified problems and a partial validation of the solution. The approach is essentially qualitative research based on the underlying philosophies of positivist and critical theories and implemented through grounded method and the instruments of interviews and questionnaires.

Representation of Agile Methodologies

Schuh (2004) summarizes agile development by stating that in order to understand what characterizes agile methodologies it is important to realize that the agile practices are not new, what is different and original about the agile approach is that the Agile Alliance has published these practices, fused them with core values about people and project environments and stated the way to build software better. Such a view assumes that people behind agile practices have provided proof that these practices are better than the traditional practices. However, the lack of this proof is one of the major limitations of agile development cited by other authors (Cohen, Lindvall and Costa, 2003; Abrahamsson et al, 2002; Lindvall et al, 2002; Turk, France and Rumpe, 2002 and Maurer and Martel, 2002) all show that most of the evidence available is only anecdotal. It is however, important to note that more recent research work (2004 to 2005) reveals more and more empirical evidence confirming success in the use agile

practices, (Eckstein, 2004; Highsmith, 2004, and Cockburn, 2004a).

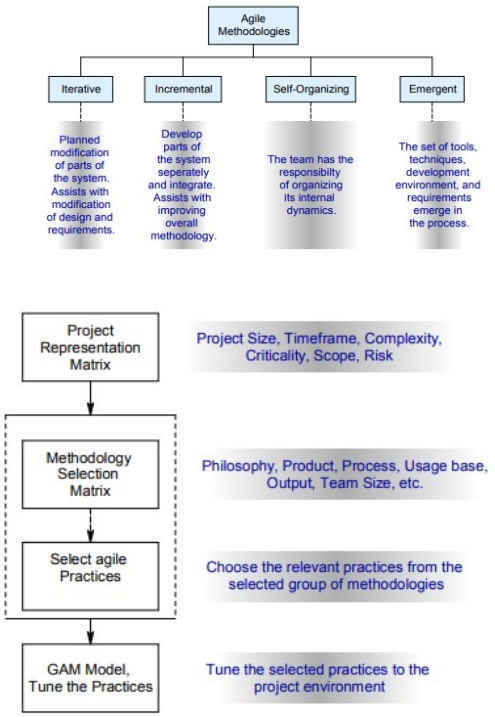

Figure 3. Agile Methodologies Selection Framework

Each rectangle represents a stage of the framework; the rectangle with dotted sides represents a phase that is composed of two steps. The arrows represent the movement in between the stages.

1. Methodology Techniques and Tools

This parameter helps the user to identify the techniques and tools applicable to the methodology. Tools may be software applications that can be used to automate some tasks in the development process, or they can be whiteboards and flip charts. In fact it is the use of tools that makes the implementation of a methodology enjoyable. Organizations therefore tend to spend a lot of money acquiring tools and training staff on tools. As technology evolves and new tools emerge more acquisitions and training are usually done.

2. Develop a Niche Model Using Agile Methods

The supposition for the choice of an appropriate agile methodology is to know well its comparative advantages in relation to all the available alternatives. The simplest way to make decision is to do it by the analysis of reports generated and based on the other experiences in their applications. However, researches show that there are not a critical number of such reports, connected to agile methodologies, in order to compare them. Therefore, the need to compare them scientifically is evident. Any kind of comparing agile methodologies, without traditional-formal methods, is extremely susceptible to subjectivity. Introducing the quasiformal approach of comparison, the problems caused by subjectivity are prevented, the problems appearing with the non-formal approach. It is possible to establish the quasi-formal method of comparison in many ways: Describing conditionally ideal methodologies, and then by comparing and evaluating the selected methodologies relating to it. 2. Identifying the set of basic characteristics deduced from the set of known methodologies, and then by comparing every methodology with the identified set. 3. Formulating a priory hypothesis about the requirements connected to the methodology. Then, it is necessary to test the formulated hypothesis by practical evidence from comparative methodologies. Some of agile methodologies have been concisely represented in the previous text. A general conclusion is that they are based on the same or similar concepts; still the methods used are very different. Therefore, there are similarities and significant differences, too. To compare them, the following criteria are the most often used in literature: a. Methodologies supporting activities of project development management, b. Methodologies supporting the phases of the software life cycles, c. Level of concreteness/abstractness these activities have, on which these methodologies are based and how they use some resources, d. Level of adaptability in the concrete situations of applications, e. Level of empirical foundation.

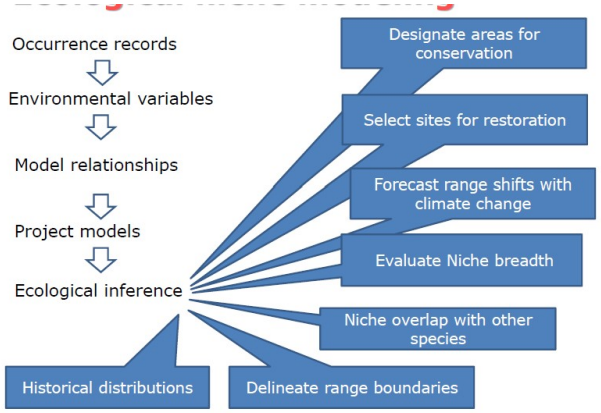

Figure 4 Niche Model

IX. CONCLUSION

Beside the use of these proposed methods, there is a requisite to explore other agile approaches that can also be investigated and integrated with the development of mobile applications. This can be done by means of industrial surveys, interviews with mobile software developers, mobile project managers, concrete discussions with Agile experts and planning other experimental studies. It is expected that this review will assist in planning future work on investigating mobile application development process using hybrid collection of best agile approach which could further undergo experimental validation. Later, the conclusions from those studies could be recommended to mobile software development companies and researchers that would suitably adopt agile methodologies as a generic development culture without worrying about specific agile methodologies. In addition, it is anticipated that these studies can be successfully implemented to enhance and evaluate the overall quality and performance of mobile software development process that brings forth many

observations which indicate substantial scope for further research in the domain

X. REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.