

Agile Development A Revolutionary Paradigm

Anurag Sinha¹, Dr.Pooja Jha²

¹Department of computer science and Information Technology, Student, Amity University Jharkhand
Ranchi, Jharkhand, India

²Department of computer science and Information Technology, Assistant Professor and Program Leader,
Amity University Jharkhand, Ranchi, Jharkhand, India
EMAIL: anuragsinha257@gmail.com¹, pjha@rnc.amity.edu²

Abstract-Software engineering practices have been used for years to build software products. Choosing a software development methodology suitable for a specific project and harmonizing the methodology for clear requirements has been controversial ever since the discipline of the software development organization. Currently, the goal of most software companies is to produce software in minimal time at an insignificant cost and to produce software in an unbalanced and changing environment as Agile grows. Finding and addressing mechanisms to address and consider improving successful agile implementation, use, and delivery. In this paper, we will discuss some configurations that confirm why Agile is preferred over waterfall models. As the appeal of Agile methods grows, many software organizations are moving away from traditional systems to adopt Agile development methodologies. As an alternative to extrapolation, Agile is highly adaptable and people-focused. Support those who work closely and for joint teams. However, team size is limited by the workforce factor. This white paper provides a multidimensional observation of performance factors in agile software development projects. Then, evaluate the performance factors published in research papers, articles and technical reports related to real agile projects. To promote the indulgence of success factors, researchers organize them into five categories: organization, people, process, design, and technology. This is why in this white paper we will analyze some of the success factors of agile practices that are manipulated at the organizational level.

Keywords-*Agile methodology, waterfall model, Success factors.*

I. INTRODUCTION

Many programs are used in businesses and associations around the world. Today, because of the confusing and compelling nature of system structure development, the traditional approach becomes complicated and the system structure cannot be deceived. The Agile approach is designed to overcome the failures of traditional methods and provide a stupid system that helps organizations and teams react quickly and quickly. When a quick approach is taken, the first step an organization takes is the Agile Declaration, which declares an agile approach that incorporates 4 values and 12 principles. of success and failure.

The software problem was tested decades ago. Various techniques and phrases have been found to address the topic, but almost everything has fallen apart and left because it cannot be changed even with the many changes needed. A targeted approach emerged as a revolt around 20 years ago. It was adopted to show the characters as well as respond to the rapidly changing software industry. The object-oriented approach has evolved from a semi-structured structure to Objects in an integrated and agile way. The object-oriented approach has indeed survived the fury of change, but on the other hand, it has not provided a complete solution to the software problem. This white paper aims to assess the literature on Key Success Factors (CSFs) in agile software development over the past decade. This white paper will only focus on research that uses logical methods to identify aspects of successful agile software development over the

past decade. As a result, an agile approach has existed to alleviate the situation of these concerns. The transition from Waterfall to an agile environment is meant to save time and money, but it's not an easy process. Developers and managers are often frustrated with integrating an agile approach into existing systems development organizations. According to Cohn (2002), the renovation of the waterfall to the landscape is difficult because the successful change is neither the lowest nor the lowest, and the end result is unpredictable, the solidity is completely different, especially covered with ash, and change is common. As a result, organizations need a closer approach to the factors that influence the effective implementation of agile projects in the workplace, followed by a more traditional approach before. to software development (e.g. slip designs or programs).

II. BACKGROUND

2.1. Waterfall Approach

A waterfall model is a scale-based approach to development, scaling down development based on the needs analysis, design, implementation, test, and maintenance phases. The projects in the waterfall model are sorted in chronological order. Stress refers to the planning, schedule, target dates, financial reporting, and overall system performance. Extensive control is maintained throughout the life of the project through comprehensive written documentation, formal reviews and user approval/approval involved in most subsequent steps. The waterfall methodology generally does not allow for an absolute transition to the next step unless the

previous step has been fully completed and validated, but this methodology has several advantages and disadvantages. The waterfall model is a traditional engineering approach applied to software engineering. The Big Design Up Front approach has been widely criticized for its inability to meet budget requirements and implement some large government projects.

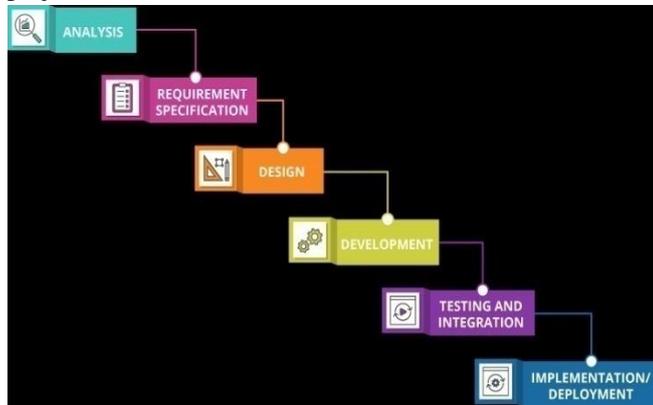


Figure 3: waterfall model

2.2 Agile Software Development

The Agile development model is based on a mutually indented iterative and incremental improvement approach to produce high-quality software in a timely and cost-effective manner, allowing plans to adapt quickly to changes. The Agile methodology focuses on providing the smallest functionality to work. We continuously improve as soon as possible and add additional features during the project lifecycle. Agile helps minimize and mitigate overall threats, keep your project on top of changes quickly, and unlike the waterfall model, you don't need to toughen up. The work is done several times and usually takes 1-6 weeks. The industry is primarily product-driven, so choose an agile development process. Agile methods focus on effective communication through written documentation. Project requirements are well documented in advance. Then, according to the business priorities, these functions are assigned to the drains attached to the iteration. The Agile method emphasizes the functioning of the software as the most important assessment of progress.

Agile software development methodology divides many functions and practices that make up a complete team, measurement, short-term release, test-based development, paired programming, customer collaboration, prototyping, refactoring, continuous integration, and reduced execution. Documentation for creating useful software. . Agile methodologies are Extreme Programming (XP), Crystal Method, Feature-Based Development (FDD), Systems

Development Method (DSDM), and Scrum.

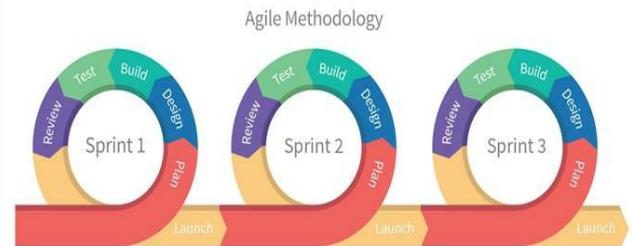


Figure 2: Agile Way of Development

2.3 Agile Manifesto

With the growing mastery of the software market in recent years, researchers are looking for more flexible ways to use software system requirements to regulate dynamic situations that change over time. The Agile Manifesto of 2001 initiated an evolution towards the succession of Agile software development, created by 17 personalities.

- People and interactions with processes and tools.
- Provides software that works with comprehensive documentation.
- Customer collaboration through contractual concessions.
- Respond to changes rather than follow the plan.

2.4 Agile Principle

The Agile organization has also documented the principles it pursues to inspire its manifesto. Therefore, the Agile process is based on principles and not on rules. Instead of predefined rules regarding roles, relationships and activities, groups and managers are guided by the following principles:

1. Our greatest concern is to quickly satisfy our customers and to continue to provide valuable software.
2. We welcome changing requirements and proceed smoothly later in development. Agile processes unite change to improve customer competitiveness.
3. Partially deliver software that typically operates in combinations of weeks to months on a shorter timescale.
4. Businessmen and developers should work together every day during the project.
5. Build projects around motivated individuals. Give them the environment, keep what they need and trust them to get the job done.
6. The most structured and efficient way to communicate information to the development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. Sponsors, developers and users must be able to maintain a fast and constant speed indefinitely.

9. Increased agility by continuing to focus on technical excellence and good intentions.
10. Simplicity, a technology that maximizes the amount of unfinished business is essential.
11. The best architecture, requirements and designs come from our team.
12. Periodically, the team thinks about how to be successful, then adjusts and adjusts their behavior accordingly.

TABLE 1: FACTORS

	METRICS	SUB SUCCESS FACTORS
Organizational	Corporate culture	Team management
		Diversity
People	User involvement	Stakeholder politics
	Team strength	Decision time
Process	Project management process	Simplicity in process
	Project definition process	Time management
	Active testing	Risk management
Project	Clear objectives	Project type
	Appropriate schedule	Nature
	Well estimated budget	Team size
Technical	Selecting agile method	Essential tools
	Paired with new technology	Technical familiarity

III. TERMINOLOGIES

Extreme Programming (XP) is location-based practices such as pair programming, client placement, and client satisfaction. XP is probably the best known and most used agile method.

Crystal Methods Crystal techniques are a simple method based on the premise that people influence software development projects more than tools or processes.

FDD (Feature Driven Development) FDD is considered an iterative, incremental, and lightweight software extension process. It is a combination of many of the best-recognized products in the industry. It all comes from a customer value function that puts this into action. Its main goal is to deliver tangible and functional software in a timely and repeatable table 1: FACTORS manner.

Dynamic System Development Method (DSDM) This is primarily a structure over a process. A vigorous way to develop a system is to fix quality, cost and time. DSDM is used to equip software and non-IT companies.

Scrum Scrum is an iterative, incremental process for writing articles or getting things done. Scrum focuses on how its peers work to make the framework adaptable in an environment that changes often. At the end of each cycle, he gives potential consent for the utility. We take a look at some of the key Scrum examples below and the process is illustrated in Figure 3.

Product backlog: a systematic summary of all elements and changes that have not yet been applied to the framework envisioned by other performing artists such as clients, advertising and trading, and venture capital groups. Product owners keep an order book.



Figure 3: Scrum Process

IV. LITERATURE REVIEW

4.1 People Factors

The realization of an ASD project is often linked to the human factor. The human resources component is also presented as an important component of a successful ASD project. Some of the performance factors of the ASD project can also be classified as human factors. However, the researchers will simply go into more detail rather than get stuck in a trivial problem of the class to which they belong.

4.2 Factors influencing the Success of Agile Software Development Projects

4.3 Competency

Competence refers to whether an individual has experience in the field of technology, has developed similar systems in the past, and has good communication and interpersonal skills [10]. Competency also means that you have acquired the

skills and competencies to perform overtime well on a similar system. By using the skill requirements matrix as a competency model, organizations can access the skills gap of ASDT members and identify those who need additional training for a specific skill [2]. The skills of these team members are consistent with the conclusions of [10].

4.4 Organizational Factors

The organization ended the success of the project. Several researchers have shown that organizational factors are also important success factors for the success of agile projects. These factors are also taken into account by customer engagement, decision time, team implementation, company culture, planning and control, the importance of business/safety, etc. According to Cao's study [15], the factors of organizational success are: a) strong management support; (b) special sponsors or managers; (c) the ethnicity of the cooperative organization as a hierarchical alternative; (d) an oral culture which places a high value on the face; (e) communication; (f) organizations for which the agile methodology is universally accepted; (g) overall team placement; h) Facilities with an appropriate agile working environment; And (i) an agile compensation system (p. 963).

4.5 Factors that influence the Failure of Agile Software Development Projects

Vijayarathy and Turk (2008) point out that some of the factors that mark the agile project disruption path include lack of training and support, an unattended agile approach, lack of facilities for pair programming, personal confrontation and the addiction. Economic evaluation criteria. Another concern that has been raised is managerial indifference and organizational resistance to change (Vijayarathy & Turk, 2008). Much like the factors of success, Chow and Cao (2008) discuss the factors of failure in four dimensions. These are summarized in Table 2.

TABLE 2- Failure factors of Agile

Developers are not collaborating on stories
Lack of discipline
Ambiguous design and code
Blaming the process
Having unclear customer requirement
Non-supportive management
Treating agile as a checklist
Incomplete work by the end of the iteration

4.6 Technical Factors

Technical factors are factors that Associate in Nursing effect on} the operation of a project and relate to the software, hardware or technology used within the project development process. Varied researchers have confirmed that technical factors conjointly play a key role in the success of

the ASD project. These technical elements, as well as requirements, development and testing, are coated in additional detail. in line with the Cao study [15], the factors of technological success are as follows: a) well-defined cryptography standards; (b) pursue a straightforward design; (c) rigorous refactoring activities; (f) an adequate quantity of documentation; (g) offer regular software, (6) provide the foremost important practicality first. (h) correct integration tests; And (i) applicable technical coaching for the team (p. 963).

IV. METHODOLOGY

5.1 FACTORS LEADING TO THE SUCCESS AND FAILURE OF AGILE PROJECTS IMPLEMENTED IN TRADITIONAL WATERFALL ENVIRONMENT

Case Study: 1

Company A (AMDOCS PUNE) was a code services company. IT had a national expression and stamp. Employees loving the reviews ranged from six,000 to 12,000 staff with well-known PMOs. Company A transferred a project to renovation code employed by agents and also the public (customers). This table enclosed moving applications to the cloud, portals and different changes (a total of quite ten code applications, together with mechanisms like tax). The explanations for the agile approach of company A relate to Business selections (fast delivery of the project and former negative experiences with the waterfalls). In-house code development antecedently reached a falls. more modern comes are dead with an agile loom. At the time of the study, the project had been current for 2 years and section one was nearing completion. The project went well and still among all measuring/chase metrics applied.

Table three provides an outline of the categories of respondents, the quantity of every sort interviewed by Company A. Respondents were known when aggregation the background of the project through discussions with project managers and PMO managers. throughout the clean-up method, Company A attended 2 retrospective and demonstration sessions. This resulted in multiple interactions seen over 2 2-hour retrospective sessions, together with 2 agile groups and a three-hour demo that includes agile groups and business associates.

Case Study 2

Organization B was a solid producer. Organization B composed shows and subsequent meet-ups the nation over and had more than 10,000 representatives with built-up PMOs. Pay has surpassed 15 billion rupees for every year. Organization B required information about the executive's framework with the work process and notice abilities. The inspiration for presenting Agile in Company B was the intricacy of the prerequisites (indistinct and reasonable), the timetable was feeling the squeeze, and new advancements were utilized. Deft establishments have been distinguished to

empower the cycle. These include Repetitive weaving machines, continuous development, a welcome change in gracefully, commitment to the movement for some time.

Table 4 portrays the sorts of respondents and the quantity of each kind met by Company B. What's more, Company B partook in 5 prototyping meetings and gatherings, which required a normal of 1 to 2 hours of cooperation and of inquiries and answers. This meeting was gone to by venture chiefs, business investigators, frameworks examiners and business pioneers.

DESCRIPTION OF RESPONDENT TYPE	SAMPLES
PMO manager	1
Business analyst	1
Project manager	2
Project resource manager	1
System analyst	1

5.2 PROPOSED SUCCESS FACTOR FRAMEWORK

Organizational factors The organization has a huge impact on the realization of a project. Conventions can control a lot in agile software development. As shown in the structure in Figure 4, the corporate tradition includes top management support in the agile software development process, which can influence how customers engage in the software development process, and ultimately, the software group decides on the 'environment. Development works.

Corporate culture

- Management assistance
- Involvement of users, team environment

The human factors The exhibition of a product advancement venture is regularly connected to the human factor. The HR factor is additionally hypothesized as a significant factor for undertaking spry programming improvement ventures. To this degree, we have the fundamental components of client significance and group capability, and under every one of them is an alternate sub-component. Under the capability of the group are the successful aptitudes expected to deal with the task, the capacity of the group, the venture director to deal with the unpredictability of the undertaking, completely deal with the time required for ends and correspondence.

Description of respondent type	Number sampled
Office manager	1
Project manager	2
Project manager	1
Business analyst	1
Business analyst	1
Developer	1
Stakeholder	1

User participation

- Treatment under commercial pressure
- Stakeholder policy

Competence of the team

- Effective project management skills
- Ability to manage the complexity of the project
- Decision time

Effective communication and feedback pic-4

Process elements are people who are concerned with the progress or usefulness of the project work itself. Software code, project location description and risk organization review and review Below is a breakdown of the primary and secondary factors of the process success factors.

Project management process

- Minimal changes in requirements
- The simplicity of the process
- Good look for project categories

Project definition process

- Crisis management
- Allocation of time

An accurate estimate of project resources

Active test
Code review

Technical factors are the reasons why there is a conflict in the way the project operates and relate to the software, experience or hardware used in the project expansion process. We have mentioned two of the most important reasons for choosing a suitable agile method depending on the conditions and constraints of the project as well as the high-tech process of the project.

Choosing the right Agile method

- Necessary configuration of the lot and infrastructure
- Use of advanced knowledge
- Relationship with technology

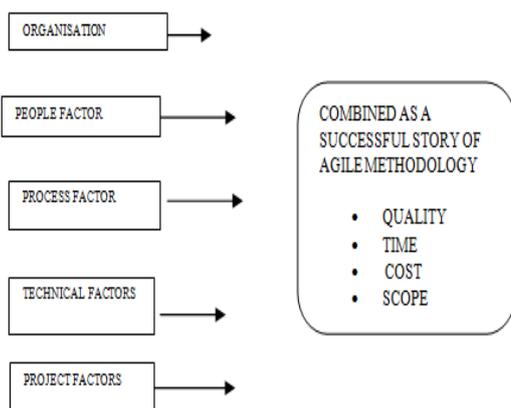


Figure 4: The Framework of Research Model

VI. FINDINGS

The following research topics are considered a logical grouping of success and failure factors in Agile project releases. Culture, client engagement and delegation, stakeholder engagement and purchasing, team structure and team logistics, types of projects and project planning, and skill levels and attitudes of team members. Each is examined in turn to discuss the results and the literature search. The feeling that the success factors are applicable in many cases and consist of conflicting failure factors assumes that the success and failure aspects are closely related. In other words, sometimes it becomes a success factor. Tools and methods should support agile practices and include the concept of failure factors. Unfavorable skills and tools within agile freedom. Thus, the success and failure factors are grouped into clear examples where it makes sense.

- • culture
- • Customer engagement and delegation
- • Stakeholder engagement
- • Team structure and team logistics
- • Project type and project plan
- • Skill level and attitude of team members

VII. DATA COLLECTION

This study is employed a survey and data collection from an organization PUNE, AMDOCS.PVT.LTD. In this survey, an analytical study is done by gathering the information through some source. It is in the form of a demographic information collection in which entire agile way of working is analyzed i.e. reason behind .success factors of agile, how the team are structured as an agile team, how organization follow the agile principals and manifesto, how product vision is made and road mapping is done, how requirements are gathered, customer satisfaction factors in agile.

VIII. FUTURE SCOPE

Another article is about four ways to deal with programming designing. Part based programming designing, light-footed technique, sidelong direction and concoction (i.e., the nimble

approach is more reasonable for little frameworks) Component and viewpoint-based philosophies are appropriate for huge frameworks. ; The blended strategy can be utilized on little and huge frameworks. These at least four innovations can assist you with lessening costs. Regarding security, smashups have security concerns and the part base in some cases appears to be significant. As far as reusability, the nimble system is not, at this point an issue, yet different strategies increment reusability. Moreover, the work components of dexterous programming improvement and task the executives - this examination intends to distinguish the fundamental work components to consider while receiving: Agility to keep your group successful. The technique utilized was an investigation of nimble groups of three sizes utilizing Scrum to create items dependent on a similar innovation. Both target and abstract measures were utilized and the outcomes are upheld by reviews. The outcomes unmistakably show that picking the correct individuals for the correct group is fundamental for the dexterous approach to work.

A top to bottom investigation of the customary approach of programming advancement assumes a significant part in the improvement of programming in present-day society. Among the different techniques of programming improvement, the dominant part allude to two approaches, hefty and light. Conventional strategies for programming advancement, alongside complete arranging, point by point documentation, and costly plan, are viewed as awkward philosophies. Dexterous displaying, otherwise called lightweight philosophy, has collected significant enthusiasm for the product designing network lately. The Agile philosophy utilizes short emphasis cycles and depends on understood information inside the group, not documentation. This exploration paper portrays the qualities of a conventional and spry approach generally utilized in programming improvement. The diverse achievement elements of the spry methodology, the achievement pace of nimble ventures and an examination of customary programming advancement with dexterous programming improvement are likewise portrayed.

IX. CONCLUSION

The rule of this examination was to contemplate the components controlling the distribution of dexterous programming improvement ventures. All the more explicitly, he considers the variables adding to the disappointment of a coordinated venture and recognizes the achievement factors adding to the accomplishment of a spry task. For reasons unknown, associations should be watchful, particularly when working with naturally separate colleagues and providers. It is likewise essential to comprehend that neglecting to embrace nimbleness, in any case, doesn't imply that deftness doesn't work, however why it doesn't work, and that proper shows ought to be held to survey the exercises. learned. It will work next time. Partner commitment and agree is fundamental to the achievement of a deft task. It is comprised of customers, backers, chiefs and colleagues. It's additionally critical to take note of that there is a colossal contrast

between a customary task administrator and a spry venture chief or scrum ace. The predominant and overpowering customary way to deal with the executives has been inadequately picked and the correct style of versatile administration should be adjusted.

At last, it has created a ton of examination zeroed in on recognizing the components that prevent the accomplishment of spry programming improvement. This white paper plans to evaluate the basic achievement elements of coordinated programming improvement research over the previous decade. Eight past examinations were chosen since they utilized experimental technique to approve CSF. Chosen considers have assessed numerous achievement factors for the advancement of dexterous programming. Eight things were chosen in this article since they were affirmed by more than one examination. The eight components are: dispatch approach, group potential and preparing, coordinated advancement abilities, client commitment, venture the executive's measures, authoritative culture, correspondence and the board uphold. The chose things have been gathered into a thing grouping containing classifications of innovation, association, cycle and individuals.

REFERENCES

- [1] Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). Manifesto for agile software development.
- [2] Moniruzzaman, A. B. M., & Hossain, D. S. A. (2013). Comparative study on agile software development methodologies. *arXiv preprint arXiv:1307.3356*.
- [3] Rao, K. N., Naidu, G. K., & Chakka, P. (2011). A study of the Agile software development methods, applicability and implications in industry. *International Journal of Software Engineering and its applications*, 5(2), 35-45.
- [4] Ullah, M. (2014). Comparison and problems between Traditional and Agile software development methods.
- [5] Coad, P. (1999). Feature-Driven Development. Object International. Retrieved April 8, 2001, from the World Wide Web: <http://www.togethersoft.com/jmccu/chapter6.PDF>.
- [6] Cockburn, A. (2000, July/August). Selecting a project's methodology. *IEEE Software*, 17(4), 64-71.
- [7] " Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). Manifesto for Agile Software Development Manifesto for Agile Software Development.
- [8] Eppler, M. J. (2006). A comparison between concept maps, mind maps, conceptual diagrams, and visual metaphors as complementary tools for knowledge construction and sharing. *Information visualization*, 5(3), 202-210.
- [9] Belassi, W., & Tukel, O. I. (1996). A new framework for determining critical success/failure factors in projects. *International journal of project management*, 14(3), 141-151.
- [10] Mahanti, A. (2006). Challenges in enterprise adoption of agile methods-A survey. *Journal of Computing and Information technology*, 14(3), 197-206.
- [11] Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of systems and software*, 81(6), 961-971.
- [12] Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D. B. (2013). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software*, 86(6), 1663-1678.
- [13] Wan, J., & Wang, R. (2010). Empirical research on critical success factors of agile software process improvement. *Journal of Software Engineering and Applications*, 3(12), 1131.
- [14] Charette, R. N. (2005). Why software fails [software failure]. *IEEE spectrum*, 42(9), 42-49.
- [15] Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, 34(11), 131-133.
- [16] Melo, C., Cruzes, D. S., Kon, F., & Conradi, R. (2011, August). Agile team perceptions of productivity factors. In *2011 Agile Conference* (pp. 57-66). IEEE.
- [17] Asnawi, A. L., Gravell, A. M., & Wills, G. B. (2011, May). Empirical investigation on agile methods usage: issues identified from early adopters in Malaysia. In *International Conference on Agile Software Development* (pp. 192-207). Springer, Berlin, Heidelberg.
- [18] Kumar, A., & Goel, B. (2012). Factors influencing agile practices: A survey. *International Journal of Engineering Research and Applications*, 2(4), 1347-1352.
- [19] Sheffield, J., & Lemétayer, J. (2013). Factors associated with the software development agility of successful projects. *International Journal of Project Management*, 31(3), 459-472.
- [20] Oluwole, Dele. "Agile X Factors." <https://www.scrumalliance.org.Web>. <<https://www.scrumalliance.org/community/articles/2013/december/agile-x-factors>
- [21] Coram, M., & Bohner, S. (2005, April). The impact of agile methods on software project management. In *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)* (pp. 363-370). IEEE
- [22] Sahota, M. (2012). *An Agile Adoption and Transformation Survival Guide*. Lulu. Com.