Nazir et al LGURJCSIT 2020

ISSN: 2521-0122(Online)
ISSN: 2519-7991(Print)

LGU Research Journal for
Computer Science and IT

Vol.4 Issue4, Oct - Dec 2020

Software Quality Assurance and Android Application Development: A Comparison among Traditional and Agile Methodology

Mohsin Nazir[1,2], Amna Amin[2], Ovais Ahmad[3], Azeema Yaseen[4], Aneeqa Sabah[2]

[1] Asian Institute of Technology, Thailand

[2] Lahore College for Women University, Pakistan

[3] Karlstad University, Sweden

[4] Maynooth University, Ireland

[1]mohsin.nazir@lcwu.edu.pk

Abstract

This paper focuses on software quality assurance in android application development with focus on approaches to confirm software quality in android applications. In this work, we compare the waterfall model with agile-scrum methodologies to realize software quality assurance with special focus on mobile applications. This research allows us to answer, “Which process model and practices are available for android software development along with focus on quality assurance?” The other research focus in this paper is “How auditors, developers and testers play their defined roles to maintain quality in mobile application development?” Agility for software development has transformed into a preferred and favored method for mobile application development. The major aim of this research is to contribute for a better knowledge for mobile application development methodologies practiced by the organizations for the development of applications based on systematic literature review.

Keywords: Android Software Development, Waterfall Software Development Model, Agile Software Development Model, Software Quality Assurance.

1. INTRODUCTION

The success of speedily emerging mobile technology is due to the newer paradigm shift in the software industry. Software quality assurance (SQA) is an important part during software development and is a continuous process. Due to the rush, hidden errors and bugs roll up in the system. This obviously creates a poor user experience after the development of the product. As the quantity of versatile applications develops, programming quality turns out to be a more and more important issue right now. A convenient versatile quality structure would serve engineers as a rule for quality affirmation. Due to quality assurance, developers can maintain the standards of application. Verification is one example of assuring the quality. It is more about preventing software defects. It provides confidence about the product. Products meet the standards that are set and fully defined. It is a proactive function, prevent from defects [1].

The defect during a software package represents the lack and inefficiency of the software to satisfy the required requirements and principles. There are different types of defects in software development, such as Arithmetic defect, Logical defect, Syntax defect, Multi-threading defect, Interface defects, and Performance defects. The mobile market is currently growing and competing, that is why development cycles are kept brief. We need to implement quality assurance from the very start of application development. A bug free application is not enough but crash free application performance is important to make a positive user experience. SQA is a very essential organizational element. The logic for this occurs to add integration and independence. Integration involves good large numbers software project managers are short-term objective determined. Project-by-project basis, with a challenging timetable, not more money, a shortage of experienced workers to execute the attempt etc. Need to train all that are involved in developing the product. Independence that there are some items good not performed by the software developer like organizing and configuration of code, reviews and analysis and verification [2].

The purpose of this research is to describe the quality assurance practices in Waterfall and Agile Model and the relation of quality in both processes for android application development. Both are mature and widely used methodologies and each have its strengths and weaknesses. Now here we need to analyze by using the Agile or Waterfall model how to maintain Software Quality Assurance of the product at every phase and try to answer the question: Which model is more suitable in terms of Software Quality Assurance? Because in a mobile application development developer faces many challenges to assure software quality [3].

Our study attempted to look at the following research questions:

- ***Recommendation*****:** Which process model suits for mobile application development?
- ***Practice:*** Which Practices are available for Quality Assurance?
- ***Role:*** What is the role of software developers, testers and stakeholders?

This work presents how quality assurance of mobile applications maintain at different stages in a waterfall model and agile methodology. These findings are often used to decide a proper process for good software quality and suitable selection of working practices for development. Section 1 introduced the overall concept of the paper composition. Section 2 explains the research methodology. In Section 3, the description of literature review is given. Section 4 explains mobile application development and importance of quality. Section 5 highlights the reasons for failure of mobile applications due to business and development issues. Section 6 discusses the traditional (plan-driven) approach with detailed description of practices in the waterfall model. Section 6.2 describes how waterfall models work for large projects. Section 6.3 briefly explains why the waterfall model fails. Section 6.4 briefly describes the uncertainty in the waterfall model. Section 7 describes an agile manifesto, its four values and twelve principles. Section 8 highlights how the requirement prioritization techniques in agile software development helps. Section 9 discusses agile (scrum) framework values and its pillars. Section 9.1 highlights the benefits of using tools beneficial for analyzing work performance and its progress. Section 9.2 describes why organizations choose scrum practices for mobile applications. Section 10 highlights the reasons for choosing agile for mobile application development. Section 11 outlines recommendation briefly with explanation of the importance of choosing right methodology according to requirements. Finally, Section 12 is the conclusion, which sums up this paper with a summary of outcomes.

2. RESEARCH MTHODOLOGY

In this paper, the followed research methodology is shown in Figure 1*: Research Methodology* The aim behind this purpose is to classify the papers based on keywords in the topic. In the second step, thematic analysis is performed which helps in choosing keywords from different resources and build a better understanding from the selected papers. It also helps to identify categories for the classification. Systematic literature review supports an understandable research method, which is used accordingly, and provides enough details. Detailed documentation allows an in-depth assessment of study. This systematic literature review method is used for multiple purposes and here it is used for broad study of observed research.

In addition, here we use the backward and forward snowballing search approaches:

Backward Snowballing: Cited by the more other research studies and part of references [4].

Forward Snowballing: Studies and referred to in the current studies [5].

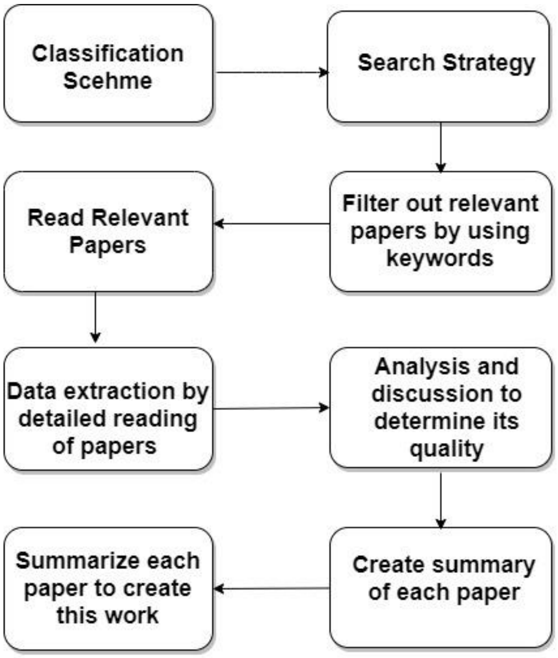

Classification Scehme
Search Strategy
Filter out relevant papers by using keywords
Read Relevant Papers
Data extraction by detailed reading of papers
Analysis and discussion to determine its quality
Create summary of each paper
Summarize each paper to create this work

Figure 1: Research Methodology.

3. LITERATURE REVIEW

In literature review, we mention many challenges from which mobile application development suffer and here the aim is to highlight a better approach with good practices to make our development process easy and successful. Mobile application development, highlights many issues like integration of software, common issues such as reliability, availability, efficiency etc. Limited time for developing an application, pressure on development team maintenance of mobile applications. Customer satisfaction is the main objective in developing a mobile applications [6]. The increasing expansion of mobile devices has completely changed the platform that is used for business, productivity, business, social, and marketing using software applications [7]. Konstantin Holl and Frank Elberzhager focused quality assurance in mobile applications in their research which reduces failures rates [8].

In another work analyze analytically and experimentally framework of software quality in mobile devices, which depend on methods, models, tools for testing applications. Poor design selections also affect the quality of software and its performance. Need to refactor the applications to improve the quality [9]. Josh Dehlinger and Jeremy Dixon both highlight some major issues defining requirements ambiguity, conducting the difficulties of applications when working on different platforms [10]. This work is going to attempt how we can choose an approach wisely to assure quality from the very first step of application development and which approach is best for development according to its scenario.

And how can different stakeholders play their roles in the whole development cycle. Krunal Bhavsar, Vrutik Shah, Samir Gopalan determined in that fact scrum helped organizations in revolution of organization by initiating honesty, accountabilities and responsibilities; by developing determination, respect and courage; by creating self-organization and collaboration; in among the members of team, that has concluded in boost of core principles of business, aspects of deliverable software features and also client satisfaction [11]. Another findings which proved the linear project uncertainty, the complication which begin coming out from the deficiency of conversation among the team members and also their conversation among the stakeholders in between the various stages of life cycle of project, deficiency of various different meetings with teams, the difference among the good trainings [12].

4. THE MOBILE APPLICTAION DEVELOPMENT & QUALITY

The goal is to enhance the standards of developed mobile applications. The reality of the matter is that the most effective mobile development approach is that the one, which will accommodate all of your requirements, optimize development time, and therefore the cost of the developed application. There are three features: requirements, estimates (cost), schedule (timeframe), and people fail to remember about the fourth i.e., quality. Either we use waterfall or agile, we need to be concerned with these four elements. Software quality assurance involves activities like requirements, design and code analysis, capturing the defects, and testing to accomplish product quality goals. The quality of software is obtained due to these three approaches: authentication (verification), constant audit and enhancement approaches. A special type of way in software quality commendations in four manners to resolve software quality: software quality standards by way of advance characteristics assessment, improve analysis, better processes and preferred tools. The development process of software must explain quality such as the important aim to increase the long-term quality of software products. Aspects of quality objectives need to be explained and written down by engaging the stakeholders. It ensures that the quality of the project obtainable and significant [13].

SQA actions for preventing the quality problems before occurring. Develop SQA plan, preparation, review the process and audits, problem reporting and corrective actions and methodologies. SQA should be involved from the beginning of the project and perform in each phase. The errors detect and reformed earlier, total life cycle expenditures are often decreased. SQA should be central and autonomous. The activities needed for software grow more and more great, continually challenging our abilities to satisfy them. SQA activity also requires an idea, which is named as SQA policies. The ambition of software quality assurance policy is preparation for methods and planning to make sure products are shaped, or the duty expressed by the management for outstanding characteristics. In strategy planning, Test Manager makes SQA policy and analyzes its schedule repeatedly. The benefits of SQA include audit and correcting the project management process, assuring the standards are followed in conducting procedures, and preventing serious software quality matters. Around each management has its individual internal principles that give a direction for analyzing and observing aspects. Standards boost the degree of accepting of the process by the entire team, so that encourages improved communication. In addition, managements require good explained quality models, which satisfy the customer requirements efficiently. This is valuable for management to promote a high standard model that the majority accurately goes with their concerns and implement it [14].

Developers and authorities always recommend best practices to assure quality in the entire development process. Effective development tools are used. Focused on its functionality and found the best solution to meet requirements. Scaling up is one from long-term challenges, recommending applicable techniques for managing rapidly increasing complex mobile application projects. Need to maintain usability, reliability of developed applications among users. Development of applications becomes more critical and complex, organizations need to choose the best development model wisely. The mobile execution environment presents many restrictions that are not introduced in desktop computing. These restrictions indicate extra attention to approach through the contrast steps of the SDLC [15]. A strong expertise in software engineering conditions can encourage governors to make conclusions that affect certainly the entire development process of application and support to produce mobile applications [16].

Applications for mobile are grown because of practices whose aim is to make applications, which are ready to operate sufficiently during a defined environment that even concerns high fulfillment on availability, efficiency and acceptance. Software quality assurance measures broadly in phased approach implementations, approximately as discussed. Minimal efforts involve the building a library of standards, technical publications, procedures, creating the adaptable set of guidance, building change control procedures, analyzing documents for completeness and agreement. Mid-level efforts include building configuration authority procedures, analyzing scripts for content, flexibility, and standards for quality, Acting like an expert and specialist all over the project, by engaging in establishing documentations, alter control boards, reexamine and analysis. In high-level efforts, the earlier also prepare inclusive system test plans, administrating autonomous, internal system integration tests, and adapting test audit reports. Balancing the script and code, keeping a test arrangements, evaluation and analysis of testing approach, maintaining currency with advanced audits, design, coding and testing automation. A good application for mobile devices, which have large scope, should design properly, manage and measured according to rules and standards and understandable according to development and product point of view [17] .

5. REASONS OF FAILURE OF MOBILE APPLICATIONS

Need of sufficient quality assurance practices for development to avoid failures in mobile applications. Failure of applications due to insufficient business strategies, difficulties in development and issues regarding users are discussed here. In the section highlight, many things based on experience and statistically experiment how the different factors lead to application towards failure [18].

5.1. Business Issues

In this category business, issues explain how the applications fail due to negligence of stakeholders, carelessness. Most of the applications failed due to poor team management and lack of experience. Selection and preplanning is the core part of any project. Right choice of strategy, team members, plans, and platform is the first step towards success but failure happens when selection of the wrong platform, insufficient environment for working [19].

- ***Poorly Developed Mobile Application Strategy***

Strong strategy is the key of success for any project without proper planning project cannot get success [20]. Right strategy according to requirements is very necessary to build successful applications. Sometimes the wrong choice of development life cycle creates problems in building a high quality application and produces delays in the development process. A choice of right strategy for system development is the first step towards application success because it helps to understand the flow of work. The best way to select a strategy is to firstly check out all the user requirements then choose an effective approach after knowing the application scope, features, time, resources [21].

- ***Poor Team for Job***

Achievement and deficiency of project depends on team [22]. Strategist, consultant, designers, developers, quality assurance experts, marketers sales teams all are the part of the team for a project each have its importance. Successful project without a good team is not possible. In a team, all members work together for the progress of application but inappropriate choice of team members, which do not have proper skills, and ability to do work in a team is a big reason, which leads an application towards failure. At each level in entire life cycle of application development, need of persons who have ability to work in team with good communication skills and expertise in their work. Deficiency of trust [23] in a team is big cause. The success of an application depends on hard work of team members. A right selection of team is more important for successful application [24].

- ***Not Constant Plans & Incompatible Business Strategies***

Needs to check plans and strategies which are made exactly working as we thinks and assume things are going as per desire but sometime in most cases application stockholders are not constant with their plans they not fully concentrate on things not surely know how much priority of which feature keep switching from one plan to another but application in results not get success due to poor planning and incompatible business strategies. If plans and strategies changed time to time then difficult developers, team members to work in this environment. If working with an incompatible business strategy that does not suit with your application development life cycle then definitely this is a step towards a failure of applications [25].

- ***Not Enough Resource, Cost, Timeline***

Cost estimation, time deadline, schedule, quality, Scope, resources [26] all are the key factors for success of any project. However, when the work is not properly scheduled, not considering the importance of quality factor, not knowledge about the market and the scope of the application. Therefore, properly doing work in those factors by taking in mind we can reduce the risk ratio of failure of mobile applications [27].

5.2. Development Issues

So many development issues in mobile applications many applications fail in the market due to not proper maintenance, working, and when not considering end user and client in mind team develop application and not consider market values in mind not assure the application fully according to expectations work properly as define its functional and non-functional requirements [28].

- ***Poor Quality Assurance***

The most important factor due to which most applications fail in the market is quality. Quality matters a lot while a highly competitive market exists and all are trying to give their best in this situation when an application is not tested properly from each and every aspect from user interface to functionality it leaves a bad impression. Need to assure each feature works properly as the end user expects. Many errors in an application can be resolved due to proper structured testing process. Role of the quality assurance team is very important in mobile application development. Check properly with testing, application meets the criteria according to standards without any bugs [29].

- ***Ignorance of Feedback and Customers***

Many applications fail when client involvement is not considered during development. Involvement of users is much more important because many times clients are not sure properly and have different expectations. By taking customer feedback regularly, opportunity to improve application and user satisfaction [30]. However, unfortunately many companies do not give priority to customer feedback or do not in touch with clients. They failed to develop an application that had success and in result, they developed applications in which the user has no satisfaction. After taking feedback periodically, performance of application can improve. Taking feedback from users and involvement of users during development is a very important factor. Due to involvement of clients, you can save your time and can develop low cost applications [31].

- ***Adding Too Many Features***

Simple and attractive user interface is always preferable by the user. Poor interface of applications is also a big issue. When trying to add too many things in the one application then it becomes complicated in development, developers face many difficulties and chance of many bugs in that type of application. Application becomes complex for designing and so much time consuming. Best thing is that we built a small iteration of the application, add new features after knowing the exact need of the user, and provide them an interface, which is easy to understand for end users. Application, which is easy to understand with a good interface always-preferable mobile application [32].

- ***Not Prepared for Updates***

Updates are most important part for mobile applications to combine new and modern functionality and features that improve user engagements. Releasing continuous updates maintains the performance of application. Updates are important to retain application and launch modern features, which boost the user engagement. Quality Assurance engineers throughout test the app before offering, although few bugs are extremely critical occur in later. Thus, the essential thing is to make essential application error fixing updates [33].

6. WATERFALL (PLAN-DRIVEN) APPROACH

Mostly used an approach in software development is Waterfall. Waterfall is a plan driven methodology. It has many phases. It is a methodical and sequential approach. A predictive process in which planning of the project is done initially, when the client knows properly what they want [34]. The major idea of this model is to break the software development process into separate phases and to implement them afterwards one after another beyond overlapping. Due to this reason, it is called a waterfall. A new stage starts when the previous phase is completed. It had a sequence of steps which are followed one at a time [35]. Figure 2 shows different phases of the traditional model. It is a mature and widely used model in the development industry. This model is very stricter due to its straight approach, and there could be limited user achievements since there is no space to make modification to the system. When the requirements are understandable and clear, the traditional waterfall model is conveniently accepted due to its clarity and linear processing approach [36].

Stresses methodical record keeping and documentation. Employee training or turnover is not much of a hassle. In this plan-driven approach, there is no permission for overlapping the phases. Once a phase is completed, we switch to the next phase and no option to revisit it again. Likewise, processes & data are generally isolated in this model, if the demands or needs of the user change, then surely need to change the code, which also refers to software coupling. Due to this reason, application and framework both are not reusable overhauled in light of the fact that the whole procedures will be changed to make any alteration, which can be bulky and costly. The overlapping of stages supports flexibility in the designing process of software. In this way, it ensures that the errors in software are removed in the phase of development, accordingly decreasing the budget of making modification to software earlier than the implementation phase. Additionally, we select the waterfall model because the stages are too linear than alternative methodologies. It not just used static techniques but also worked with dynamic techniques [37].

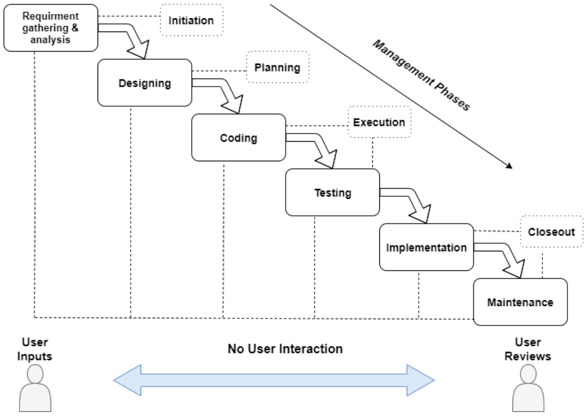


Figure 2: Waterfall Model.

We have some issues with the Waterfall model. This model depends on its initial stage requirement and specifications because sometimes customers are not sure about specific details and functionalities whereas this stage is meaningful with respect to this approach. Not possible to understand the fully complex project and its scope in the initial step. Also without working prototype users not able to figure out exactly what they want. A risk rate in this approach is high because there is no option to go back. If a developed application is not according to the user's expectations, then definitely it will be a waste of cost and time. Applications are easy to develop in which the customer is properly sure about its requirements because the life cycle is straightforward and for a pre-defined scope of project easy to add or remove the features [38].

In Waterfall customers are commonly involved in requirements and software designing although they are not involved as enough and do not support the higher as they are supposed strongly in agile methodology. In waterfall development, model integration performs very later. All phases must be finished to conclude a clear conclusion. A result requires a scope document to work up. It involves System boundary, expense profit inquiry, usefulness study and planning of management. When the process has been properly accepted, the Manager of the project establishes a plan for a project that consists of scope, schedule and timeline. Planning and designing becomes straightforward. Growth of a project can check easily because scope of the project is defined at an early stage. Developers and customers both agree with working and performance but this type of approach is used for specific types of application projects in which workflow is performed linearly [39].

6.1. Practices, Verification and Validation in Waterfall Approach

Fundamental aim of a successful project to meet the user assumptions. To obtain this aim practices are performed from an initial step. Each practice has its importance, which is performed to assure the quality of project and user expectations. In Quality Assurance activities focused on error prevention in the start stages, on error removal during coding and testing, and on defect containment in support [40]. Defect containment activities are performed for error prevention. This step was performed earlier because it is a very important step in software development phases because of developers sometimes they make conceptual mistakes, not have proper knowledge in the domain, not good experience or expertise in the development technique, which is used. Whereas fault avoidance activities like fault tolerance and safety assurance are continuously performed in the center of phases to avoid failures and check measures. Planning, designing and implementation performs during the development of software. To make fault tolerant adding some tools or references in methodology [41].

Figure 3 shows different phases of the waterfall model. At the initial step, it involves business and system requirements specification. Designing involves high and low level designs, after the completion of the first two stages move towards coding, testing, implementation and maintenance. It involves well explained techniques for quality assurance like validity and suitability of product, inspection of code and proper testing. This approach deals with many steps such as requirement gathering, specifications, implementation, unit and integration testing also perform release alpha beta testing version of application for fix issues if any find then finally release the product. However, in it every step is final. Changes are not expected in this approach [35].

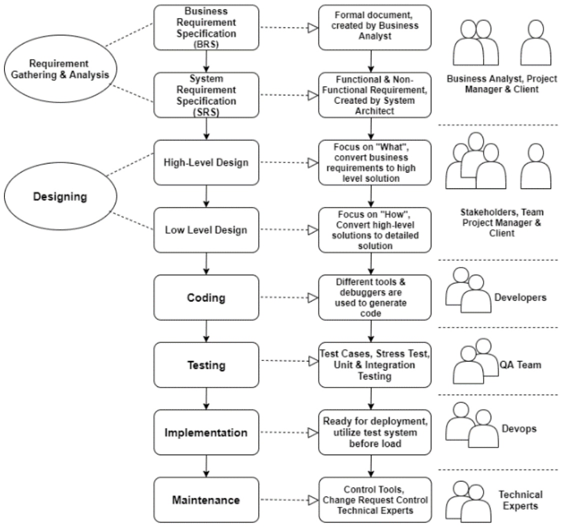

Requirement Gathering & Analysis
Business Requirement Specification (BRS)
Formal document, created by Business Analyst
Business Analyst, Project Manager & Client
System Requirement Specification (SRS)
Functional & Non-Functional Requirement, Created by System Architect
Designing
High-Level Design
Focus on "What", convert business requirements to high level solution
Low Level Design
Focus on "How", Convert high-level solutions to detailed solution
Stakeholders, Team Project Manager & Client
Coding
Different tools & debuggers are used to generate code
Developers
Testing
Test Cases, Stress Test, Unit & Integration Testing
QA Team
Implementation
Ready for deployment, utilize test system before load
Devops
Maintenance
Control Tools, Change Request Control Technical Experts
Technical Experts

Figure 3: Phases and the roles of stakeholders in Waterfall Model.

- ***Requirement Gathering and Analysis***

In the first stage requirement gathering, business analysts make sure the business and functional needs satisfy the Stakeholders desires. Collect entire requirements in this phase with the help of flow charts, user cases, test cases and written script. Business requirement specifications created by business analyst and system requirement specifications created by system architect. Formal documentation is done and finds out the application functional and non-functional needs. This initial stage is the very important phase because project success and failure fully depend on this stage [42]. At this stage Requirement understanding document, Functional & non-functional requirements and prototypes are deliverables as shown in Figure ***4***. In waterfall model project success depends on this stage, effective communication between project manager and stakeholders gives good results if communication is not good and requirements not fully understood at this stage then in last it may be a great loss of effort and then more cost and time is required to rebuild the application. Sometimes a client is not good at conveying their thoughts but the best practice is to effectively use prototypes to give ideas about applications [43].

- ***Designing***

In this phase designing a high and a low-level design document involves. High and low level designs focus on "What" and "how". High-level design describes the architecture for developing an application while low level respectively. High-level design document describes method and relation between classes and program specifications. It is a component level design process. In this phase, we are able to know how the application requires to be established. It consists of tables, hierarchy diagrams, pseudo-code and ERD for a project. ERD shows relationships between entities whereas hierarchy diagrams describe the relationships among several layers of elements. Pseudo code is written before the actual coding of the application for sketching out the structure of the program. This stage helps to describe how the application creates experts who approve the best design [44].

- ***Coding***

Developers start writing code after the first two phases are completed when system analyst and architecture finalize the design specifications. Coding is the longest phase of a development cycle. Java is a very popular language for development in mobile applications. Many famous applications are built using this language. Kotlin is an alternative to Java. It is a cross platform programming language for android application development. Development kits C++, C#, Python, corona that is used for building android applications utilizing Lua. HTML, CSS, javascript utilize the Adobe PhoneGap framework. There is no language that is known as the correct language for an application development of android. It depends on skills and writing choice of language according to project requirements and scenarios [45].

- ***Testing***

In this phase developers verify that, the application works according to design specifications. In quality assurance activities, software testing is the major one part. Additionally, it is a compulsory part of this development process. Mostly used unit testing, system testing and acceptance testing. For quality assurance techniques black box testing is very generally adapted on it. It needs not broad awareness of the internal structure of the application. Problems, found in testing, should be accordingly written down and reported to the developer's team. In Testing, Business Analyst confirms that the application works according to desired requirements. This verification process assures entire applications are running as designed. Different methods of testing like test cases test and defect reports workflow diagram unit testing. It's difficult for developers to go back for changes [46]. In android app development, it is a bit hard to follow a waterfall model. Using this model, it is hard to perform testing at the last. As you develop, also test the application and fix the bugs as you get them. That is the best practice we follow because during testing we may get few errors in functionalities also so again we need to go back to the coding stage, which contradicts the waterfall model. Best practice promotes a test system in development. The ultimate consumer requires assuring that the application fulfils their requirements. It is the validation of an application [47].

- ***Implementation***

After the successfully functional and non-functional testing the next phase is implementation in this phase a very good practice firstly load the application on different test systems to check its working before deployment. When the team approves the application then it is time to implement it in the production environment. It is the validation part in the waterfall model. It is also possible applications successfully pass when verified but flops during validation. User training is also a part of this stage [48].

- ***Maintenance***

This step also includes modification to alter attributes or components or need to improve performance. List of new features are implemented. Need of modifications due to change of requests or defects, which are uncovered. If any issue is found and resolved, make sure the code is updated at both sides in the development area and in the coded environment. Maintenance is the last stage in waterfall model user manuals are very beneficial at this phase. Among this phase, users are using the developed application. Technical experts are at this stage to resolve any technical issue or problem. Three types of maintenance, which includes corrective maintenance, perfective maintenance and adaptive maintenance, are done [49].

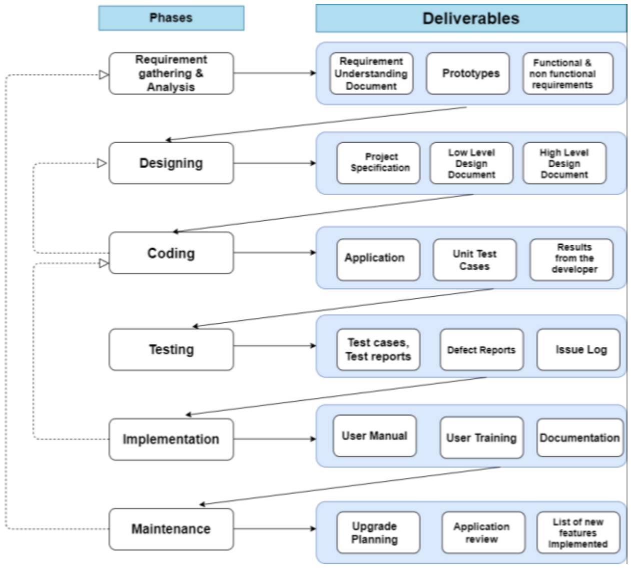


Figure 4: Deliverables at each phase in Waterfall Model.

6.2. Waterfall Model for Large Projects

As we know the waterfall model is a slow process because it is working in a linear way but on the other hand, especially in the largest project it has many benefits, which are remarkable. Used in large projects [50].

Strong Documentation and Team Members

In planning and documentation stages, all the detailed design structures specifications robust scope is maintained. If the team members of a project change then easily new team members can continue working. Strong detailed documentation with full concepts and designs helpful for team members. This model accommodates shifting team members [51].

· ***Focus on User Requirements***

Changing the design is very difficult in the last if in the beginning stages focus on specifications and detailed analysis before going in the development phase then it helps to develop the right project with concentration and in results always positive but here is the point client should exactly know about their requirements then it will happen.

- ***Achieve Milestone***

This is very suited approach for progress every work in different stages is divided in different team members who have good skills and experts in their field everyone prepare for its working area and do their work in timeline for the whole process [52].

- ***Structured Organization***

It involves a detailed process to maintain each element in the very well-structured way detailed feasibility study to design, development and testing so on. This way is very helpful in large projects because larger projects demand all the things in the structured way. Therefore, this model pushes organizations to be organized. However, developing shorter projects through this way can be exhausting [53].

6.3. Reasons for Failure of Waterfall Model

Reasons due to which this waterfall model fails are discussed here. It has some deficiencies like one way street, no revisit of phases, less interaction with stakeholders, not much flexibility, less communication with customers, not feedback until the project is complete. These factors affect the performance of waterfall models [54].

- ***On Way Street***

The waterfall model is similar to a one-way street. In this methodology working on a project in linear fashion, that is step by step. It is a problem due to which sometimes the waterfall model fails. There is no option to go back. If we go back then it is a big loss of effort, time and cost. It does not give permission to go back and revise [55].

- ***Overlapping***

Waterfall model works very well, when the project description is stable and strong, not uncertain requirements because this methodology does not offer overlapping of phases once the phase is completed then move to the next stage. However, in real projects need to work parallel to boost the performance and decrease the cost phases may overlap in projects [56].

- ***Less Interaction with Customers***

Less interaction and not feedback by the customer during the development process and after the development phase changes cannot be modified easily. The customer in the development of an application does not take feedback. Involvement of customers is very low. Face difficulties to compute progress of a product because less client interaction is included in the development of the product [57].

- ***Not Flexible***

Difficult to adjust changing requests. According to the waterfall model customer involvement only in the beginning of the project this model demands the client give its requirements accurately and completely because this model worked in a linear way step by step but usually the customers are not sure of their requirements and fail to explain. This model does not accept changes [58].

- ***Delays***

In many large projects there will be unpredictable delays if the requirements are not clear in the start firstly need to determine the requirements and specifications clearly then move to the development phase otherwise the project goes towards failure [59]. Sometimes more time is spent in analyzing the requirements and specifications of users. Time delays occur in the waterfall model [60].

6.4. Uncertainty in Waterfall Model

In this model testing concentrates on internal efficiency and external effectiveness of the project. The aim of internal testing is to check out code efficiency, standardization and documentation whereas external testing checks that software is working according to designs and specifications, doing work exactly, and performing fundamental functionality. However, a problem arises when the client is not sure and changes their requirements so it is leading to redesigning, redevelopment of code and so on that increases the cost of the project. Nonetheless, activities in the development life cycle do not always have achievements. If the requirements and specifications are not fully defined then the development process revolves from one phase to another as shown in Figure ***5***. It creates great difficulty and makes the development process and application progress towards failure [61].

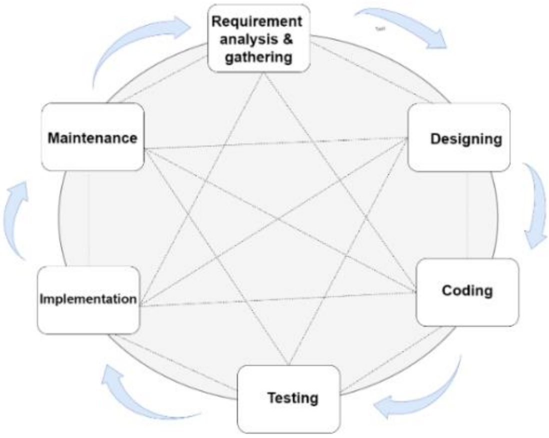


Figure 5: Uncertainty in Waterfall Model.

7. AGILE IN SOFTWARE DEVELOPMENT

Agile methodologies play an essential and significant performance in the progress as correlated to alternative approaches. Agile in software development allows for an adaptive process in which the team and developers deal with modifications in requirements and specifications of clients, despite lateness in the development procedure. Every organization and company wants to achieve high quality systems, in a less time, within budget or loss cost. Therefore, large numbers of companies started to adopt agile methods. Nowadays agile is used in every field of life due to its flexibility and atmosphere, which is very comfortable for working. Agile offers many practices to make work better with good performance. With the help of agile, very flexible and dynamic applications provided to customers. Agile project management principles have been applied to enhance team communication, skillful development, participations and also everyone exchanges their ideas [62]. It involved the stakeholders throughout the complete life cycle of development process. Stakeholders from upper level to lower level all are involved in it for better performance. Nonetheless, agile practices were developed to deal with many of the issues obtained within the traditional approach. Agile methodologies encourage flexibility, light documentation, changes at any phase [63].

Design quality can enhance due to pair programming and help to decrease defects. Team members work together so that chances of defects are less than in it. Dedication of team members plays an important role in agile. It works better and shows good outcomes. Continuous client involvement during the development process helps to meet the user expectations. The Agile process had many practices, which are used to assure the quality in development of the project. In general, integration is most used practice during using agile methods. Needs to integrate all the time. Not wastage of time due to continuous integration. Because due to this practice they caught bugs which reduce their time and help to work in a better way. Rigidity and mobility are two issues due to incremental requirements because these two factors have high risk and a lot of effort. Team members not just have the knowledge of working but also know business knowledge and good communication skills. Business knowledge is important to deliver valuable products to customers [64].

7.1. Values, Principles of Agile

According to the agile manifesto, it consists of 4 values and 12 principles. As shown in

Figure 6 first value explains selection of right processes and right tools for work are most important to develop a project. Fundamental key point is that every member of teamwork together to increase work performance and building a successful software depends on teamwork and individual participation and sincerity with work. Second value shows some organizations spent a lot of time on documentation. In agile primary aim to build software, not documentation. Repeated communication with team members and customers, collaboration among clients and developers support team to know what client asked for and what the actual needs. Commitment agreements have their position but this builds a barrier with clients that no help develop successful software. Must be capacity to favor for adjustment otherwise planning for the project turns irrelevant. Normally in traditional development models, project works according to planning. This value focused on flexibility. It welcomes changes throughout the development process [65].

As stated in principles

Figure 6 Agile depends on high-level involvement of customers during the review sessions because it gives opportunity to go through all over the working of the project before delivered and make changes and decisions on time. Development team organizes meetings with customers for deliverable iteration. Members of the team communicate with each other for better work in iteration. After deliverable iteration, customers provide feedback related to current working and additional features need to add in the system. For each iteration, there is a time limit [66]. Dealing with uncertain requirements during the development, project under budget constraints and short time period delivery of project most significant characteristics of this approach. It is a practical approach, which implements adoptability, flexibility, productivity. It has been refined for controlling and organizing system development processes [67].

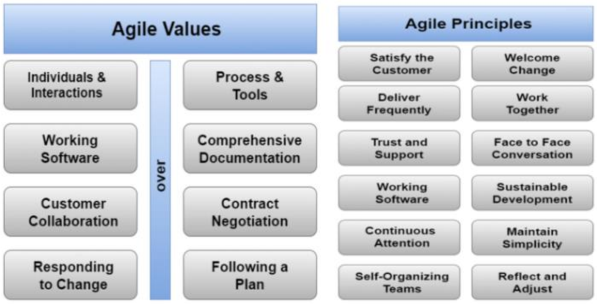


Figure 6: Agile Values and Principles.

Agile concentrates on time-boxed commitment and periodic reprioritization, so this is possible the priority delivery of iteration needs some extra time also may be adding some project cost. Due to continuous customer engagement, engagement may be needed to add additional features during the project. Therefore, this factor can add the additional cost and time. New challenges and changes are allowed in it. Agile process requires valuable communication between teams due to its modular nature. Satisfaction of customers is the highest preference. Software team works to increase the production and achievement of the software. Prefer small teams but must be dedicated with a good coordination and synchronization. In addition, work is very efficient even if the scope of the project is not defined properly in advance. Assurance of quality at each level and effective communication meetings in business between management and developers. Approve the performance of the application repeatedly. For quality assurance, the issue of good skills persons rapidly increase to perform the task effectively. Need to execute fundamental verification, programmed suitcases and recommend new concepts. The major aim is to satisfy the customers; usually it concentrates on particular needs with different iterations. New features added easily in agile by adopting different iterations [68].

8. REQUIREMENTS PRIORITIZNG TECHNIQUES IN AGILE SOFTWARE DEVELOPMENT

Requirement Prioritization is very necessary for agile software development. The procedure of prioritization in agile development unlike the traditional development process [69]. In agile first priority customer satisfaction, even the requirement change in the development. Allocate the priorities on the base of customer demands. Requirement prioritization techniques help to figure out requirements based on their importance. Many constraints including time, professional skills, budget, resources and client satisfaction. For prioritization, the need of effective communication between stakeholders and the needs of experts. There are many various techniques in what way manners prioritize the requirement [70]. Moscow is a popular method that lies under nominal scale, which generates specialized classes in which objects can be classified. Requirement preferences on the basis of the critical business advantages [71]. It is convenient, lower/ fewer restricted, low struggle needed and capable to deal with alternatives. Conveniently Flexible, sufficient for small large demands. Not required complicated calculations and wide knowledge. It builds up understanding among team members and stakeholders. Fast and clear scheduling, friendliness, easy to understand and simplicity as shown in Figure 7. Requirements, which are nominated to categorize, based on the priority of implementation. Entire requirements in a group represent the same priority for all that are in one group. Challenging in Moscow to know which should or could requirements are much more meaningful than rest. Less consistency of practice. Not have sequencing of tasks and have lack of planning [72].

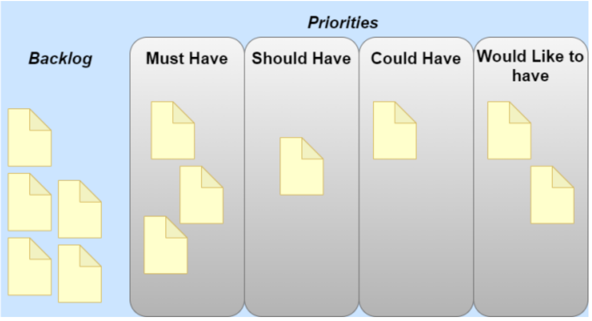


Figure 7: Agile Backlog Prioritizing Technique: Moscow.

9. AGILE (SCRUM) FRAMEWORK VALUES AND ITS PILLARS

Scrum consists of five values and three pillars that are the base of this framework. Scrum values include commitment, courage, focus, respect, and openness. In Figure 8 lists a scrum pillar and its values. In scrum transparency can be performed with the help of scrum tools like product backlog, burn down charts, task boards, sprint reviews, definition of done, daily standup meeting, retrospective etc. Scrum provides us this favor it allows visibility with respect to work progress and team. Inspection in scrum performed with the help of activities like use of scrum board to check out current working of project. Feedback from clients and stakeholders. Built prioritized product backlog and strategies for release. Approval and inspection of the deliverables by the owner of product. For improvement in the project team adapt changes, in scrum meaning of adaptation is that which could perform correctly or better. Use the outcomes from the inspections to choose which practices and procedures to run next like in daily stand up meeting inspection is [performed, in sprint review scrum team ask for feedback from stakeholders and then adapt changes according to analysis and measures. In sprint retrospective, team discuss about issues problems, and for opportunity to increase the efficiency of work. Adapt new strategies for better results and improvements [73].

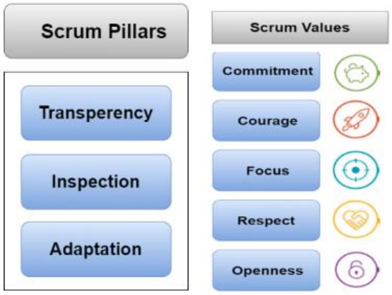
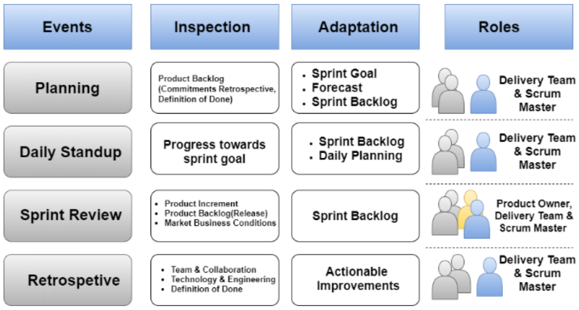


Figure 8: Scrum Pillars and its Values.

Commitment is an essential part, which is self-decision of every person in a team for success of product in every sprint. In a team, respect is a key factor being a part of a scrum team; everyone gives respect to others skills, knowledge and their expertise. Due to focusing on goals, applications can deliver on time and avoid wastage of resources and lower rate of risks. Courage facilitates potential to team members to deal with all types of difficulties and problems in the sprint. Openness empowers transparency and belief in the squad by discussing thoughts, ideas and issues with the team. As stated in “The Scrum Guide”, product owner, scrum master and development teams are involved in the scrum team as shown in Figure 9. Three roles included in scrum: Project Owner who is responsible for managing product backlog, Scrum Master assures appropriate systems and working Scrum Team involving each member responsible for high quality project development. Planning of sprint, Daily meetings, Sprint Review and Retrospective all are the part of scrum in which members communicate with each other to make work performance better and better. Scrum involves planning of project, delivery and feedback from stakeholders and team members. Daily scrum meetings are helpful for efficient growth of projects in the right way. Sprint review has its importance in the development process. Audit of each sprint performed instead of auditing and testing in the last. Whole team focuses on work. Allows for regular reflection and feedback of working on projects. Interaction on a daily basis increasing engagement, energy and clarity. Collective team ownership and accountability. All helps to give structure to this approach.

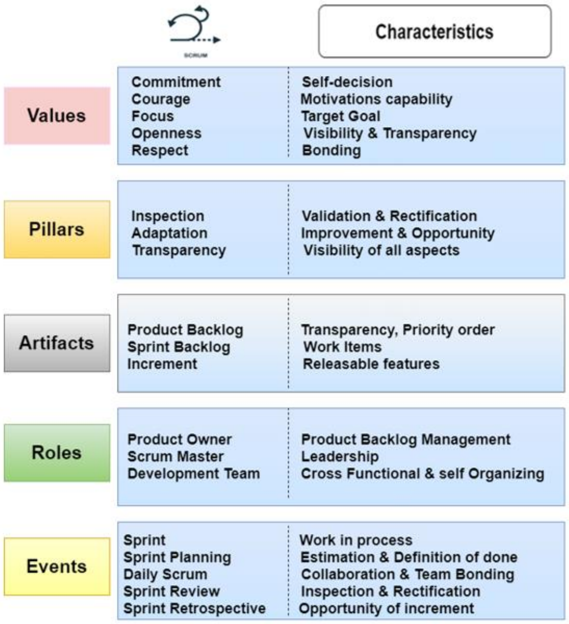


Events	Inspection	Adaptation	Roles
Planning	Product Backlog (Commitments Retrospective, Definition of Done)	• Sprint Goal • Forecast • Sprint Backlog	Delivery Team & Scrum Master
Daily Standup	Progress towards sprint goal	• Sprint Backlog • Daily Planning	Delivery Team & Scrum Master
Sprint Review	• Product Increment • Product Backlog(Release) • Market Business Conditions	Sprint Backlog	Product Owner, Delivery Team & Scrum Master
Retrospetive	• Team & Collaboration • Technology & Engineering • Definition of Done	Actionable Improvements	Delivery Team & Scrum Master

Figure 9: Scrum Events and roles of stakeholders.

All are equally responsible for the success or failures of developed applications in a team-based approach. This approach gives importance to speedily delivery of an application with functional requirements. Generally, works in sprints, and normally every sprint has a limited time duration planned at the beginning. Deliverables are arranged according to its priority level and business values. In addition, if the outlined task for the sprint is not completed at time, the task is prioritized again and completed first successfully. Normally a choice of agile methodology for managing development projects can boost the team. The possibilities of success strongly become high. This model is very popular in development due to its valuable practices, which are available in the development life cycle of application [74].

Agile depends on high-level involvement of customers during the review sessions because it gives opportunity to go through all over the working of the project before delivered and make changes and decisions on time. Agile development demands the continuous user involvement for better performance but sometimes a customer is a person who does not have enough knowledge or not enough time for meetings or not interest in participation creates a problem for teams. Development team organizes meetings with customers for deliverable iteration. Members of the team communicate with each other for better work in iteration. After deliverable iteration, customers provide feedback related to current working and additional features need to add in the system. For each iteration, there is a time limit. Dealing with uncertain requirements during the development, project under budget constraints and short time period delivery of project most significant characteristics of this approach. It is a practical approach, which implements the adoptability, flexibility, and productivity. It has been refined for controlling and organizing system development processes [75]. Figure 10 Shows characteristics of scrum values, pillars, Artifacts, roles and its events.



SCRUM		Characteristics
Values	Commitment Courage Focus Openness Respect	Self-decision Motivations capability Target Goal Visibility & Transparency Bonding
Pillars	Inspection Adaptation Transparency	Validation & Rectification Improvement & Opportunity Visibility of all aspects
Artifacts	Product Backlog Sprint Backlog Increment	Transparency, Priority order Work Items Releasable features
Roles	Product Owner Scrum Master Development Team	Product Backlog Management Leadership Cross Functional & self Organizing
Events	Sprint Sprint Planning Daily Scrum Sprint Review Sprint Retrospective	Work in process Estimation & Definition of done Collaboration & Team Bonding Inspection & Rectification Opportunity of increment

Figure 10: Scrum Values, Pillars, artifacts, Roles, Events and their Characteristics.

Agile concentrates on time-boxed commitment and periodic reprioritization, so this is possible the priority delivery of iteration needs some extra time also may be adding some project cost. Due to continuous customer engagement, engagement may be needed to add additional features during the project. Therefore, this factor can add the additional cost and time. New challenges and changes are allowed in it. Agile process requires valuable communication between teams due to its modular nature. Satisfaction of customers is the highest preference. Software team works to increase the production and achievement of the software.

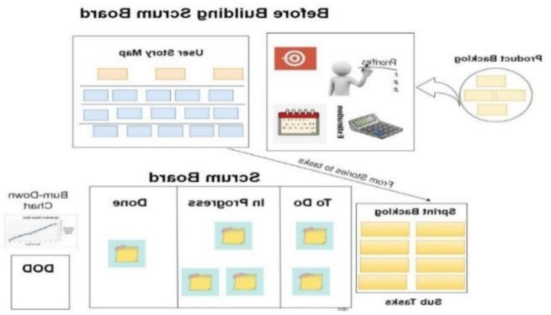
Prefer small teams but must be dedicated with a good coordination and synchronization. In addition, work is very efficient even if the scope of the project is not defined properly in advance. Assurance of quality at each level and effective communication meetings in business between management and developers. Approve the performance of the application repeatedly. For quality assurance, the issue of good skills persons rapidly increase to perform the task effectively. Need to execute fundamental verification, programmed suitcases and recommend new concepts. The major aim is to satisfy the customers; usually it concentrates on particular needs with different iterations. New features added easily in agile by adopting different iterations [76].

9.1. Benefits of using Tools

Story mapping is a cooperative practice that trains members of an agile team in the establishment of product backlog. The aim of story mapping exposes the specifications and details of what is undoubtedly mandatory to deliver client value. Story mapping is an important practice for each agile team. It is a powerful technique for assuring that the team perceives their customer needs. Definitely expressive solution and concentration on delivery. Maintain the story map continuously up-to-date according to progress of work. A scrum board is a very helpful tool, which makes backlog items apparent. Traditionally the board is divided into three partitions [77]. These partitions are to do, in progress and done as shown in Figure 11. In planning the sprint, every user story is defined as a task and inserted priority wise into the category. Team members choose which feature they like to accomplish first according to requirement. In addition, move post-it from to do category to in progress category list. After completing the task move post-it from in-progress list to done. Whole team observes tasks on a daily basis and members

Table 1 **:** Difference between Physical and Digital Board.

Features	**Physical Scrum Board**	**Digital/Virtual Scrum Board**
Work for remote teams	-	Yes
Automatic Alerts	-	yes
Easy to visualize	yes	-
Face to face interaction	yes	-
Work for local teams	yes	yes
Facilitates daily scrum	yes	-
Synchronous Multiple users use	-	yes
Integrate with other tools and documents	-	yes
Automatic Updates	-	yes



*F*igure 11: Scrum Board.

Two types of scrum boards one is physical other one is digital or virtual board. The difference between physical and digital boards is shown is **Table *1.*** To track stories scrum teams mostly used scrum board, which placed in organization physical board. Physical boards good for smaller teams and local teams. Whenever you adopt a procedure for physical board, confirm that solid communication scenario in place thus no any task is missed, task position is continually up-to-date. On the other hand, teams also used online board to manage sprints. The advantages such that live updates, flexibility, customization. In addition, tools make easy to accumulate entire project information. Online scrum board best for teams, which have remote team members [79].

9.2. Using Scrum for Mobile Applications

Scrum determined methodology introduces an iteration-based achievement that revised and customized revolving around on the outcome accomplished by preceding iterations. Design phase to build an overview to process the product backlog. It relies on implementation of the sprint by the team members. It implies that test endeavor should remain in each sprint. Agile concentrates on providing continual dispatch of functional or working software versions. Measurement is an essential in achieving the awareness on what results over the development process. Gather information, establish metrics and draw conclusions. Scrum suggests "Sprint Retrospectives" and prepares the outcomes of the affected effort and also analyzes lessons experienced and opportunities of advancement and improvements for the subsequent sprint [80]. Agile methodology builds their important values in the prospect of concentrating on the engaged product, acknowledging continentally to the adjustment as shown in Figure 12.

Agile scrum introduces sprints minor duration committed to multiple steps in application development. It offers a convenient plan to handle workload. In scrum, the roles for mobile application and tasks are broken down as if all the team members get approximately equal work. Scrum always welcomes changes for satisfaction of the customer. Product owner represents the concerns of the client. Product owners assure the product according to the client even with many unplanned changes in the development time. Scrum increases adaptability that is very helpful in mobile application development. Scrum commits the strength to manage each step of development, easily discuss with each other and modifications according to need. This decreases risk without any doubt and all stakeholders participate in the process and perform their better [81].

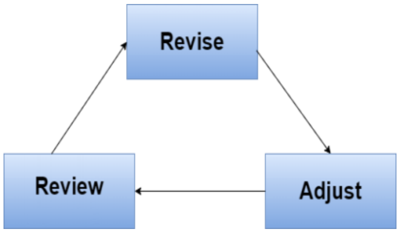


Figure 12: Flexibility in Agile.

Product owner knows about future applications. Team members work with the product owner. Perfect teamwork is the key advantage of scrum methodology to build a successful mobile application [73].

Implementing agile provides readjust processes, procedures and practices for the unstable requirements. Mobile applications need to be refined fast and maintain a less cost elected to rewarding and strong in the market of large numbers of customers. Agile effectively deals with business models, learning the market, designing the system and release in instant time [82].

10. Results

Some are specific reasons for choosing Agile for Mobile application development. By practicing agile methods, an application has essentially entirely software affirmation. Agile approaches are good for the mobile application development process. Agile is a convenient method for improvement in mobile application development. These methods support teams of software development to acknowledge instantly to adjustments [83]. Agile practicing Scrum which stated the perfect long-term satisfaction [84].

- ***Faster Development***

In the situation of agile methodology, the completely mobile application project is split within smaller components which are considered like separate sub-projects. Diverse teams manage the particular sub-projects individually, along not much interdependence on one another. Moreover, every person has a transparent idea of what individual contributions will also correlate resources and time limit, which speed up the development procedure [85]. Self-organizing teams each team member focus on work for better performance and give their best towards achieving their part in the mobile application development project, a result of which is a higher modernized application development process with quick delivery [86].

- ***Best Quality Product***

Agile acknowledges testing the application at each stage in development. Before launching, the testing is necessary part. Whereas in traditional application development approaches, the application is tested at the last. Thus, the application developed under agile practices assures quality and it is according to client needs. Due to repeatedly testing at many stages, errors and issues can be forecast earlier that is why flawless applications are developed [87].

- ***Reduce Risks***

In the dynamic market with different requirements and trends, it is entirely uncertain to launch an application. Mostly, it takes place that the market statistics taken within examination although developing application achieves out-of-date by the era launch application. The effect is low ROI and a dark expectation in the market. In that situation, agile is a tool, which supports taking estimated risks and improvement in projects according to market scope. It helps to reduce risk factors. Particularly, this methodology allows a mobile application development company to adjust changes to any specific sprint without disquieting the code of the preceding sprints. Therefore, building their application is much appropriate for the market [88].

- ***Better Quality***

Dissimilar to the traditional application development, agile does not perform testing at the last in development phase. Instead, it promotes testing of each particular module at the underdeveloped level. This decreases the uncertainty of experiencing a bug at the stage of affirmation testing of the entire project. It also facilitates mobile application development to audit the application elements at each moment of the development process, accomplish adjustments according to client's requirements, ultimately serving in delivering a greater excellence of services [89].

- ***Enhanced Customer Experience***

The agile in mobile application development concept brings a powerful significance to people and associations that contribute the development squad with favorable chances to work intimately with clients and also consider their ideas [90]. In addition, the projects are handed over to the users in the form of many sprints that take clarity to the process. Also facilitates the team to regulate whether participants are on the similar page or note, enabling them to build the needed changes earlier. This reduces the risk of launching an application, which does not achieve the concept behind, so it assures better customer experience [91].

- ***Cost for Development***

Considering each step is well structured, accomplished, delivered then easily estimate the cost of building an application and so confirm application budget. In addition, in case that at any stage in development, the requirement to increase the application cost, resources can conveniently be achieved in agile methodology. So avoid leaving behind the project unfinished because of shortage of needed budget and resources [92].

- ***Customization***

In agile mobile application development, also support developers with circumstances to modify their development procedures. No rules and regulations to build an application in a specific way. Experts keep considering multiple approaches to build and launch the mobile application, accommodate leading position technologies, mechanisms and tools. Analyze that agile processes facilitate developers to modify development schedules according to their preference and deliver a user-focused outcome.

- ***Higher ROI***

Mobile application companies enter the market with the basic version of application and update the application with every iteration. This arranges easier for the application owners to analyze their ideas, deliver the better features according to customer requirements and market situation. So the application owners and companies associated with mobile application development can make better decisions to achieve ROI in the market [93].

- ***Seamless Project Management***

Agile approaches support with the convenience of managing projects efficiently by converting the entire application development project into many individual modules. Easily allocate the tasks to distinct teams and shorten the dependencies. Team members are responsible for fast response on changing requirements. Also can manage a log of the activities executed on every mini project so through this way resolve if anything is missing or either not working according to the proposed plan. Also can analyze the productivity of each team member [94].

- ***Efficient Tools for Application Development***

It supports tools, which are very efficient, and boost the work provided facility to focus on the goal. Goal of scrum delivering new, acceptable, profitable and valuable software to clients quicker. Tools achievable come with different features that provide favor in planning, executing, and ship software [95].

- ***Earlier Market Reach***

Due to partitioning of the entire application project into modules, the agile mobile application development process boost to deliver each module with the specified deadline. Underdeveloped later than deadlines is not possible. The result is that the entire application project is structured and delivered at time or even preceding to it, that's earlier market reach. This methodology is amazing for an application development. The need is that everyone should be aware of end goals and perform their best for better performance [96].

11. RECOMMENDATIONS

Selection of SDLC according to the scenario is very important because all the project success depends on it. In studies we find out that waterfall is applicable when requirements are well defined, straightforward. Waterfall model works better when you know the proper scope of the project. It gives better results in these types of scenarios. It is excellent for those applications having a specific goal. Because the risk factor of failure is very low when all the requirements are well known. However, if requirements change frequently it is risky to develop an application by using the waterfall model [97]. Agile consists of flexible problem-solving procedures. Generally, it is very visible that agile and scrum project management processes are very helpful in facilitating an achievement in a project. This is a lightweight framework for managing and controlling in a distributed and speedily changing environment. The sprint was a useful tool for division the application into multiple tasks. In mobile applications, we also have need of updates with time. Agile works better in this environment because consumers already utilize the facility of application. Scrum helps to build a goal-oriented mobile application. It facilitates concentration on quality affirmation activities. Quite than only building an application from starting point to ending, the approach of accomplishing work in the order of sprints is the understanding of flexibility in designs [98].

One of the most important things in agile method is that it is satisfied by dividing the entire development of the project into workable chunks, conveniently proficient in a precise time. Working systematically more comfortably provides you to design a practical schedule, which is not too discouraging in a complete development life cycle. If the development requires improvement, redevelop, modernize then it is a best approach to follow for android mobile applications. Confidentially agile helps to achieve the goal. If not have good planning and documentation, development environment is unsteady, resources are limited, need to launch the app earliest this approach will also fulfills your expectations. Agile is a flexible model for working. It gives relaxation when the client is not properly sure about its requirements or fails to define properly. Due to continued meetings, testing and good practices application development becomes easier because this provides the environment in which team members work in pair programming and easily achieve the goal defined by the customer request. In agile methods members of a team feel easiest to communicate and share their ideas when they are physically working in the same place or office but sometimes this is not possible, there are other good ways to interact with each other by using technology. Due to continuous improvement in the project, many technical issues were addressed instantaneously. In addition, ideas regularly reconsider, evaluate, and deliver the product according to client needs. Many applications face failure due to poor strategies and development issues, we can resolve these issues by using agile. Agile gives support to mobile application development [99].

12. CONCLUSION

In the waterfall model, more time is spent in documentation to clear the specifications and goals, while in the agile model, developers and team spend more time in developing the application on a daily basis with meetings helping to check and maintain the progress of mobile applications. Waterfall is good for applications but agile methodology has flexibility and is ready to accept challenges all the time, and the customer gets a progressive outlook of their polished work in prototyping in the development cycle. It helps the team members to improve the requirements of clients due to continuous communication. It works as iterative as compared to compromising the overall structure composition of application. Basic version of functioning software can be built rapidly for its earliest launch in the market. Due to its iterative approach it helps better to compete with market demands to perform better according to end user requirements. Consequently, "Good quality processes provide great quality products". These results present the particular methods that are used in mobile applications. The major aim of that research is to contribute for a better knowledge for mobile application development methodologies practiced by the organizations for the development of applications.

REFERENCES

[1] D. Franke and C. Weise, "Providing a software quality framework for testing of mobile applications," in *Proceedings - 4th IEEE International Conference on Software Testing, Verification, and Validation, ICST 2011*, 2011, pp. 431–434, doi: 10.1109/ICST.2011.18.

[2] F. J. Buckley, S. Member, and R. Poston, "Software Quality Assurance," *IEEE Trans. Softw. Eng.*, vol. 10, no. 1, pp. 36–41, 1984.

[3] G. K. Hanssen, "A longitudinal case study of an emerging software ecosystem: Implications for practice and theory," *J. Syst. Softw.*, vol. 85, no. 7, pp. 1455–1466, 2012.

[4] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, 2014, pp. 1–10.

[5] S. Jalali and C. Wohlin, "Systematic literature studies: database searches vs. backward snowballing," in *Proceedings of the 2012 ACM-IEEE international symposium on empirical software engineering and measurement*, 2012, pp. 29–38.

[6] S. Kaleel and S. Harishankar, "Applying Agile Methodology in Mobile Software Engineering: Android Application Development and its Challenges," in *Computer Science Technical Reports*, 2013, pp. 1–11.

[7] J. Trimble *et al.*, "Mobile Applications Softwrae Engineering: Chellanges and Research Directions," *Univ. Edinburgh, Edinburgh*, vol. 6, no. December, pp. 362–366, 2014, doi: 10.1109/WiMOB.2011.6085388.

[8] K. Holl and F. Elberzhager, "A mobile-specific failure classification and its usage to focus quality assurance," in *Proceedings - 40th Euromicro Conference Series on Software Engineering and Advanced Applications, SEAA 2014*, 2014, pp. 385–388, doi: 10.1109/SEAA.2014.19.

[9] G. Hecht, O. Benomar, R. Rouvoy, N. Moha, and L. Duchien, "Tracking the Software Quality of Android Applications along their Evolution," in *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2015, pp. 236–247, doi: 10.1109/ASE.2015.46.

[10] C. Peskersoy and M. Tanintara-Art,

"Extending web applications to mobile platforms: a software engineering view." 2013.

[11] K. Bhavsar, V. Shah, and S. Gopalan, "Scrum: An Agile Process Reengineering in Software Engineering," *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 3, pp. 840–848, 2020, doi: 10.35940/ijitee.c8545.019320.

[12] D. Hilčíková, M. Dohnanská, D. Lajčin, and G. Gabrhelová, "Agile Project Management As One of the Critical Success Factors in Project Risk Management in Machinery Industry in Slovakia," *Int. J. Econ. Financ. Issues*, vol. 1, no. 1, pp. 37–54, 2020.

[13] S. J. Khalaf and M. N. Al-Jedaiah, "Software quality and assurance in waterfall model and XP - A comparative study," *WSEAS Trans. Comput.*, vol. 7, no. 12, pp. 1968–1976, 2008.

[14] D. Galin, *Software quality assurance: from theory to implementation*. Pearson Education India, 2004.

[15] Dongsong Zhang and Boonlit Adipat, "Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications," *Int. J. Hum. Comput. Interact.*, vol. 18, no. 3, pp. 269–292, 2005, doi: 10.1207/s15327590ijhc1803.

[16] N. Shiratuddin and S. M. Sarif, "The m d-matrix: A learning tool in the mobile application development course," *Int. J. Mob. Commun.*, vol. 7, no. 4, pp. 494–514, 2009, doi: 10.1504/IJMC.2009.023696.

[17] L. Bass, P. Clements, and R. Kazman, *Software architecture in practice*. Addison-Wesley Professional, 2003.

[18] R. Johnson, *Expert one-on-one J2EE design and development*. John Wiley & Sons, 2004.

[19] J. M. Verner and L. M. Abdullah, "Exploratory case study research: Outsourced project failure," *Inf. Softw. Technol.*, vol. 54, no. 8, pp. 866–886, 2012.

[20] Wahidmurni, "The risk factors affecting to the software quality failures," in *arXiv preprint arXiv*, 2017, pp. 1–14.

[21] P. E. Bierly III and S. Gallagher, "Explaining alliance partner selection: fit, trust and strategic expediency," *Long Range Plann.*, vol. 40, no. 2, pp. 134–153, 2007.

[22] M. Wächter, "Mobile Strategy," *Mob. Strateg.*, pp. 1–19, 2016, doi: 10.1007/978-3-658-06011-4.

[23] V. N Inukollu, D. D. Keshamon, T. Kang, and M. Inukollu, "Factors Influncing Quality of Mobile Apps: Role of Mobile App Development Life Cycle," *Int. J. Softw. Eng. Appl.*, vol. 5, no. 5, pp. 15–34, 2014, doi: 10.5121/ijsea.2014.5502.

[24] H. Wi, S. Oh, J. Mun, and M. Jung, "A team formation model based on knowledge and collaboration," *Expert Syst. Appl.*, vol. 36, no. 5, pp. 9121–9134, 2009.

[25] T. H. Davenport, *Mission critical: realizing the promise of enterprise systems*. Harvard Business Press, 2000.

[26] K. Kakkar, R. Shah, and M. Kakkar, "Risk analysis in mobile application development," *IET Conf. Publ.*, vol. 2013, no. 647, pp. 429–434, 2013, doi: 10.1049/cp.2013.2351.

[27] R. Schmidt, K. Lyytinen, M. Keil, and P. Cule, "Identifying software

project risks: An international Delphi study," *J. Manag. Inf. Syst.*, vol. 17, no. 4, pp. 5–36, 2001.

[28] C. Giardino, S. S. Bajwa, X. Wang, and P. Abrahamsson, "Key challenges in early-stage software startups," in *International Conference on Agile Software Development*, 2015, pp. 52–63.

[29] D. A. Garvin, *Managing quality: The strategic and competitive edge*. Simon and Schuster, 1988.

[30] M. A. Subih *et al.*, "Comparison of agile method and scrum method with software quality affecting factors," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 5, pp. 531–535, 2019, doi: 10.14569/ijacsa.2019.0100569.

[31] M. Warkentin, R. S. Moore, E. Bekkering, and A. C. Johnston, "Analysis of systems development project risks: An integrative framework," *ACM SIGMIS Database DATABASE Adv. Inf. Syst.*, vol. 40, no. 2, pp. 8–27, 2009.

[32] D. A. Norman, *The invisible computer: why good products can fail, the personal computer is so complex, and information appliances are the solution*. MIT press, 1998.

[33] J. S. Brown and P. Duguid, *The social life of information: Updated, with a new preface*. Harvard Business Review Press, 2017.

[34] M. McCormick, "Waterfall and Agile Methodology," in *MPCS Inc*, 2012, pp. 1–8.

[35] M. Huo, J. Verner, L. Zhu, and M. A. Babar, "Software Quality and Agile Methods," in *28th Annual International Computer Software and Applications Conference*, 2004, pp. 520–525.

[36] A. Adetokunbo and A. Basirat, "Software Engineering Methodologies: A Review of the Waterfall Model and Object- Oriented Approach," *Int. J. Sci. Eng. Res.*, vol. 4, no. 7, pp. 427–434, 2014.

[37] M. Kramer, "Best Practices in Systems Development Lifecycle: an Analyses Based on the Waterfall Model," *Rev. Bus. Financ. Stud.*, vol. 9, no. 1, pp. 77–84, 2018.

[38] N. M. Devadiga, "Tailoring architecture centric design method with rapid prototyping," in *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, 2017, pp. 924–930.

[39] S. Blomkvist, "Towards a model for bridging agile development and user-centered design," in *Human-centered software engineering—integrating usability in the software development lifecycle*, Springer, 2005, pp. 219–244.

[40] Y. Kamei and E. Shihab, "Defect Prediction: Accomplishments and Future Challenges," *IEEE 23rd Int. Conf. softwrae Anal. Evol. enginerring(SANEER)*, vol. 5, no. December 2017, pp. 33–45, 2016, doi: 10.1109/saner.2016.56.

[41] M. Lane, "The lean distributed scrum (LDScrum) model: investigating the relationship between lean and agile software development in a distributed development context," 2013.

[42] C. Courage and K. Baxter, *Understanding your users: A practical guide to user requirements methods, tools, and techniques*. Gulf Professional Publishing, 2005.

[43] R. G. Cooper and A. F. Sommer, "The agile–stage-gate hybrid model:

A promising new approach and a new research opportunity," *J. Prod. Innov. Manag.*, vol. 33, no. 5, pp. 513–526, 2016.

[44] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," *ACM SIGARCH Comput. Archit. News*, vol. 28, no. 2, pp. 83–94, 2000.

[45] J. A. Hoffer, *Modern systems analysis and design, 6/e*. Pearson Education India, 2012.

[46] S. Balaji, "Waterfall vs v-model vs agile : A comparative study on SDLC," *Int. J. Inf. Technol. Bus. Manag.*, vol. 2, no. 1, pp. 26–30, 2012.

[47] M. J. Harrold, "Testing: a roadmap," in *Proceedings of the Conference on the Future of Software Engineering*, 2000, pp. 61–72.

[48] J. Humble and D. Farley, *Continuous delivery: reliable software releases through build, test, and deployment automation*. Pearson Education, 2010.

[49] R. R. Sambasivan *et al.*, "Diagnosing Performance Changes by Comparing Request Flows.," in *NSDI*, 2011, vol. 5, p. 1.

[50] M. A. Cusumano and S. A. Smith, "Beyond the waterfall: Software development at Microsoft," 1995.

[51] J. R. Hackman and R. J. Hackman, *Leading teams: Setting the stage for great performances*. Harvard Business Press, 2002.

[52] A. Singh and P. J. Kaur, "Analysis of software development life cycle models," *Lect. Notes Electr. Eng.*, vol. 476, no. 2, pp. 689–699, 2019, doi: 10.1007/978-981-10-8234-4_55.

[53] W. Van Casteren, "The Waterfall Model and the Agile Methodologies : A comparison by project characteristics," in *Research Gate*, 2017, no. February, pp. 1–6, doi: 10.13140/RG.2.2.36825.72805.

[54] D. Huizinga and A. Kolawa, *Automated defect prevention: best practices in software management*. John Wiley & Sons, 2007.

[55] M. D. Rosenau and G. D. Githens, *Successful project management: a step-by-step approach with practical examples*. John Wiley & Sons, 2011.

[56] H. Mooz and K. Forsberg, "A Visual Explanation of Development Methods and Strategies including the Waterfall, Spiral, Vee, Vee+, and Vee++ Models," *INCOSE Int. Symp.*, vol. 11, no. 1, pp. 610–617, 2001, doi: 10.1002/j.2334-5837.2001.tb02348.x.

[57] Y. Leau, W. K. Loo, W. Y. Tham, and S. F. Tan, "Software Development Life Cycle AGILE vs Traditional Approaches," *Int. Conf. Inf. Netw. Technol.*, vol. 37, no. 1, pp. 162–167, 2012.

[58] L. Hao, J. Zhang, and X. Ma, "Development Trend of Software Process Model," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 569, no. 5, 2019, doi: 10.1088/1757-899X/569/5/052077.

[59] L. Cocco, K. Mannaro, G. Concas, and M. Marchesi, "Simulating kanban and scrum vs. waterfall with system dynamics," in *Lecture Notes in Business Information Processing*, 2011, vol. 77, pp. 117–131, doi: 10.1007/978-3-642-20677-1_9.

[60] G. E. Iyawa, M. E. Herselman, and A. Coleman, "Customer interaction in

software development: A comparison of software methodologies deployed in Namibian software firms," *Electron. J. Inf. Syst. Dev. Ctries.*, vol. 77, no. 1, pp. 1–13, 2016, doi: 10.1002/j.1681-4835.2016.tb00560.x.

[61] H. J. De Vries, *Standardization: A business approach to the role of national standardization organizations*. Springer Science & Business Media, 2013.

[62] R. Paul and L. Behjat, "Using Principles of Scrum Project Management in an Integrated Design Project," in *Proceedings of the 15th International CDIO Conference, Aarhus University, Aarhus, Denmark, June 25 – 27, 2019*, 2019.

[63] O. Ktata and G. Lévesque, "Agile development: Issues and avenues requiring a substantial enhancement of the business perspective in large projects," in *proceedings of the 2nd Canadian conference on computer science and software engineering*, 2009, pp. 59–66.

[64] D. Leffingwell, *Scaling software agility: best practices for large enterprises*. Pearson Education, 2007.

[65] K. S. Rubin, *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley, 2012.

[66] S. Sharma, D. Sarkar, and D. Gupta, "Agile Processes and Methodologies: A Conceptual Study.," *Int. J. Comput. Sci. Eng.*, vol. 4, no. 5, pp. 892–898, 2012.

[67] J. Highsmith, *Adaptive software development: a collaborative approach to managing complex systems*. Addison-Wesley, 2013.

[68] M. Ibrahim *et al.*, "Exploring the Agile Family: A Survey," *IJCSNS*, vol. 20, no. 10, p. 163, 2020.

[69] N. Saher *et al.*, "A Review of Requirement Prioritization Techniques in Agile Software Development," in *Knowledge Management International Conference (KMICe)*, 2018, pp. 25–27.

[70] M. Vestola, "A Comparison of Nine Basic Techniques for Requirements Prioritization," in *Helsinki University of Technology*, 2010, pp. 1–8.

[71] A. Hudaib, R. Masadeh, M. H. Qasem, and A. Alzaqebah, "Requirements Prioritization Techniques Comparison," *Mod. Appl. Sci.*, vol. 12, no. 2, p. 62, 2018, doi: 10.5539/mas.v12n2p62.

[72] C. L. Cooper and J. Marshall, *Understanding executive stress*. Springer, 1978.

[73] J. Sutherland and K. Schwaber, "The Scrum Papers," *Nuts, Bolts Orig. an Agil. Process*, 2007.

[74] D. Leffingwell, *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. Addison-Wesley Professional, 2010.

[75] L. Cao and B. Ramesh, "Agile requirements engineering practices: An empirical study," *IEEE Softw.*, vol. 25, no. 1, pp. 60–67, 2008.

[76] Z. A. AL HAMMADI, "Being Agile: The Influence of Agile Project Practices on the Project Team Productivity." The British University in Dubai (BUiD), 2018.

[77] D. Leffingwell, *SAFe 4.5 Reference Guide: Scaled Agile Framework for Lean Enterprises*. Addison-Wesley Professional, 2018.

[78] T. DeMarco and T. Lister, *Peopleware: productive projects and teams*. Addison-Wesley, 2013.

[79] M. Cohn, *Succeeding with agile: software development using Scrum*. Pearson Education, 2010.

[80] L. Corral, A. Sillitti, and G. Succi, “Agile Software Development Processes for Mobile Systems: Accomplishment, Evidence and Evolution,” in *In International conference on mobile web and information systems*, 2013, pp. 90–106.

[81] K. Schwaber, *Agile project management with Scrum*. Microsoft press, 2004.

[82] C. Larman, *Agile and iterative development: a manager’s guide*. Addison-Wesley Professional, 2004.

[83] M. Mousaei and T. J. Gandomani, “OVERVIEW OF ADVANTAGES , CHALLENGES , AND ADAPTATION OF AGILE METHODS IN MOBILE,” *J. Softw. Eng. Intell. Syst. ISSN 2518-8739*, vol. 5, no. 1, pp. 45–54, 2020.

[84] B.-A. Andrei, A. Casu-Pop, S.-C. Gheorghe, and C.-A. Boiangiu, “A Study on Using Waterfall and Agile Methods in Software Project Management,” in *Journal of Information Systems & Operations Management*, 2019, pp. 125–135.

[85] S. Hassani-Alaoui, A.-F. Cameron, and T. Giannelia, “‘We Use Scrum, but …’: Agile Modifications and Project Success,” in *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020, pp. 6257–6266, doi: 10.24251/hicss.2020.765.

[86] M. C. Layton, S. J. Ostermiller, and D. J. Kynaston, *Agile project management for dummies*. John Wiley & Sons, 2020.

[87] W. E. Lewis, *Software testing and continuous quality improvement*. CRC press, 2017.

[88] G. H. N. Laursen and J. Thorlund, *Business analytics for managers: Taking business intelligence beyond reporting*. John Wiley & Sons, 2016.

[89] N. K. Hanna, *e-Transformation: Enabling new development strategies*. Springer, 2009.

[90] M. Elkhateeb, A. Shehab, and H. El-Bakry, “Mobile learning system for egyptian higher education using agile-based approach,” in *Education Research International*, 2019, pp. 1–12, doi: 10.1155/2019/7531980.

[91] R. Banfield, C. T. Lombardo, and T. Wax, *Design sprint: A practical guidebook for building great digital products*. “ O’Reilly Media, Inc.,” 2015.

[92] B. S. Blanchard, D. C. Verma, and E. L. Peterson, *Maintainability: a key to effective serviceability and maintenance management*, vol. 13. John Wiley & Sons, 1995.

[93] D. J. Anderson, *Agile management for software engineering: Applying the theory of constraints for business results*. Prentice Hall Professional, 2003.

[94] B. Meyer, *Agile!: The Good, the Hype and the Ugly*. Springer Science & Business Media, 2014.

[95] A. Cockburn, *Agile software development: the cooperative game*. Pearson Education, 2006.

[96] K. Fowler, *Developing and managing embedded systems and products:*

methods, techniques, tools, processes, and teamwork. Elsevier, 2014.

[97] F. Ji and T. Sedano, “Comparing extreme programming and Waterfall project results,” in *2011 24th IEEE-CS Conference on Software Engineering Education and Training, CSEE and T 2011 - Proceedings*, 2011, vol. 5, pp. 482–486, doi: 10.1109/CSEET.2011.5876129.

[98] T. J. Gandomani and M. Z. Nafchi, “An empirically-developed framework for Agile transition and adoption: A Grounded Theory approach,” *J. Syst. Softw.*, vol. 107, pp. 204–219, 2015.

[99] B. C. Sichone and B. C. Sichone, “The Quality Quest: In search of the best in the best Managing Quality in project and Human Resource.”