www.irbas.academyirmbr.com July 2021
International Review of Basic and Applied Sciences Vol. 9 Issue.3

Hybrid Integrated Lightweight Agile Framework for Quality Product by Managing Time and Cost

SYEDA FIZA RUBAB
Faculty of Computing, The Islamia University of Bahawalpur.
Email: fizbukharii@gmail.com
Tel: +923346909978

ZAIGHAM MUSHTAQ
Faculty of Computing, The Islamia University of Bahawalpur.
Email: zaigham@iub.edu.pk
Tel: +923357401242

NAJIA SEHAR
Faculty of Computing, The Islamia University of Bahawalpur.
Email: najiasaher@gmail.com
Tel: +923336387864

SAIRA LATIF
Department of Computer Science
COMSATS University Islamabad, Lahore Campus.
Email: saira.latif93@gmail.com
Tel: +92 332 5858333

Abstract

Performance, reuse, and high change costs are all issues that traditional approaches have dealt with successfully. Due to advancements in technology, projects are getting more complex and dynamic, offering new problems for development organisations in terms of adapting agile approaches to meet their requirements. Development; Quality Assurance; Testing; Planning, Risk Assessment, Designing, and Modeling. As a result, there are challenges at every stage of the SDLC. Agile was originally intended for short projects. Its usefulness prompted enterprises to organizations large scale software development. Managing large-scale agile projects can also be challenging. Software metrics analyze the overall health of the development process. Despite its benefits, practitioners do not employ SDLC measurements, which adversely effects on the quality of product. The findings of proposed agile framework illustrate new hybrid, customized agile methodologies that resolve the generated challenges about quality of the product. Metrics are used in the majority of the recommended framework's effective practices. The framework's effectiveness is demonstrated by an industrial-scale project and a published case study. The extent of the work that can be done with these frameworks and methodologies is much broader. Testing frameworks that produce better results than Test-Driven Development (TDD) do as well. The findings lend validity to the framework that has been proposed. Any firm can benefit from the proposed framework, which can help them save time, effort, and money while still producing a high-quality product.

Keywords: *Time and Cost, Hybrid Agile Framework, Quality Product, Risk Assessment, Metrics.*

Introduction

In today's world, the software now encompasses nearly every subject area and platform under the sun (desktop, mobile, online, etc.). Advance states rely heavily on applications of this type to carry out their daily activities. Aside from saving time and money, these programs were also able to execute jobs that a human could not. Software engineering is one of the most important fields in today's world, and it has numerous subfields. In order to provide the greatest software, software companies are constantly creating new methods and approaches. If the software meets the needs of the customer and is completed on time and under budget, it is called a success (Kumar, 2014).

Developing software products follows a set of procedures known as the Software Development Life Cycle (SDLC). There is a wide range of common activities, including planning, designing, developing, and testing (Hoda & Noble, 2017). In the past, corporations developed software using the waterfall model, a standard approach. Rather than working in a linear fashion, it's more like a step-by-step process where each stage builds on the one that came before it. Numerous difficulties were brought to light, including the high expense of rework, the difficulty of finding errors, the importance of involving customers, and the impact on performance and reusability (Sampietro, 2016).

In response to the shortcomings of more traditional approaches of sequential development, Agile was created with four standards and twelve principles. It has risen to the top of the field thanks to its capacity for adaptation and flexibility. More and more studies on agile began to take place as the industry shifted its focus to agile. Researchers started working on agile transition (Hoda & Noble, 2017), (Hussain et al., 2021) its different flavors (Sridhar Nerur et al., 2005), addressing various organizational issues (Stavru, 2014), guidelines, and enhancements for old and new organizations (Sampietro, 2016).

All agile research is directed at three distinct groups of people in the industry: Supporters, who embrace elegant in its purest form; Tailors, who combine elegant and traditional strategies to suit their unique circumstances; and Dilettantes, who combine a few ceremonial agile activities with a more conventional approach (Cram et al., 2017). Numerous agile techniques are available to meet the increasing complexity of modern software. Scrum, Feature Driven Development, and eXtreme Programming are just a few examples. Rather than relying on a single agile methodology, firms opt to combine two or more agile methodologies with meeting their demands. As a result, their consumers receive increased value rather than having to wait months for the project to be completed (Chauhan et al., 2017).

Certain Agile policies are classified as heavyweight, while others are classified as lightweight. In the last 15 years, researchers have focused on customizing and integrating agile approaches, as methodologies have an effect on the cost, time, and risk of a project. (Cahyana et al., 2019).

To determine quality, one must look at it from a variety of angles. Customers, developers, organizations, and other stakeholders all have various perspectives on this. From the perspective of quality assurance, it is critical to implement agile approaches in all projects. There are advantages and disadvantages to each agile technique. For all of these reasons, businesses have begun implementing hybrid approaches (Fahad et al., n.d.). Isn't it more important to look at how a product is made rather than how it's finished? In the eyes of the business, the most important quality trait is a happy customer. (Alsaqaf, Daneva, & Wieringa, 2017).

As further organizations started approving agile, they challenged several issues, and they faced several issues (Albadarneh et al., 2015), including risk management, requirements (Alam et al., 2017), development (Rauf & Alghafees, 2015), scaling, and quality (Hussain et al., 2021). The focus of this research is to take a step in resolving these issues.

Literature Review

In order to provide the greatest software, software companies are constantly creating new methods and approaches. If the software meets the needs of the customer and is completed on time and under budget, it is called a success (Fitzgerald & Stol, 2017). Developing software products follows a set of procedures known as the Software Development Life Cycle (SDLC).

To name just a few, there's always research, development, and testing to keep you busy. In the past, corporations developed software using the waterfall model, a standard approach. In this sequential model, each phase begins after the previous one, and there is no way to go back to a prior stage once it has begun. Many difficulties were highlighted, including the expense of rework and discovering errors, customer interaction, performance, and the reusability of the product (Chauhan et al., 2017).

In response to the shortcomings of more traditional approaches of sequential development, Agile was created with four standards and twelve principles. It has risen to the top of the field thanks to its capacity for adaptation and flexibility.

More and more studies on agile began to take place as the industry shifted its focus to agile. Researchers started working on agile evolution (Hussain et al., 2021), its diverse tastes (Nerur & Moe, 2012), addressing numerous executive matters (Stavru, 2014), strategies, and improvements for deep-rooted and novel administrations.

All agile research is directed at three distinct groups of people in the industry: Supporters, who embrace agile in its purest form; Tailors, who combine agile and traditional strategies to meet their unique circumstances; and Dilettantes, who combine a few ceremonial agile activities with a conventional approach (Cram et al., 2017).

It is imperative for developers to adhere to a disciplined development cycle in order to ensure timely delivery, feature complexity, and compatibility with various platforms and target markets. Software engineering is a crucial part of designing mobile applications, and this is undeniable. Scrum, Extreme Programming, and Feature-driven Development were all advocated as part of the study's Agile Approach. (Wang et al., n.d.)

The study discovered some previously undiscovered points: 40-hour workweek and on-site customer prohibits scattered teams from implementing integrated methods. Storing sprint backlog burndown charts helps scale. While a model is proposed to handle the challenges, there are significant concerns with this work: It is a heavyweight approach with limited usage. Large organizations can only use it. It excludes risk assessment and feasibility, which are vital for large-scale initiatives. (Jameel Qureshi, 2017).

The Scaled Agile Framework (SAFe) was selected for this study since it is one of the most extensively utilized Agile methodologies in the market. Qualitative case studies were undertaken within SAFe to discover possible intervention points for incorporating human values into the software.

Artifacts, roles, ceremonies, practices, and culture are all high-level scientifically supported SAFe interventions. Agile artifacts like user stories, roles like product owners, ceremonies like stand-up meetings, and techniques like business-facing testing may be adjusted in SAFe to facilitate the inclusion of human values in software development, as we explain in this paper. (Rafi et al., 2015), (Hussain et al., 2021).

Table 1: Strengths and Weaknesses

Year	Methods	Models	Strength	Weakness
2015	US-Scrum (Rafi et al., 2015)	Scrum, FDD	• Addresses certain quality attributes • Object-Oriented Analysis, Design	• Doesn't measure quality • Smaller teams cannot use it • Too many roles • No Coding Practice suggested • Quality is not measured • No validation of the model
2015	eXSRUP (Sharma & Wadhwa, 2015)	XSRUP	• Management, Development practices • Plan drove + Agile • Lightweight	• Complex • Expertise required • Quality not addressed • Suitable for large projects
2017	XScrum (Jameel Qureshi, 2017)	Scrum, XP	• Improves Scalability • Makes it usable for Distributed teams • Improve Efficiency, Productivity • Validated • Quality improved	• Eliminating 40-hours • Complex • No-Risk assessment • Expertise Required • Large Teams only • Heavyweight Methodology
2018	L-Scrumban (Abdullah Albarq & Qureshi, 2018)	Lean, Scrum, Kanaban	• Improve development • Eliminating waste • Practical Approach • Visual Board • Validated Model	• Separate boards • 5 tools were utilized for 1 methodology • Too many meetings • The team identify types of requirements • No progress showing to the customer
2018	C&C (Qureshi et al., 2018)	Scrum	• Communication, Coordination improved • Validated Model • Lightweight	• Quality not maintained • Communication channel not clear • Large teams only • Other factors remain untouched

Methods covering Highlighted Challenges

The following Table summarises which of the reviewed approaches attempted to address the issues raised in this research. The Table's columns are as follows: R for Risk Assessment, RE stands for Requirement, DE for Design, DEV for Development, S for Scalability, and QA for Quality.

Table 2: Proposed Solution Addressing Highlighted Issues

Method Name	R	RE	DE	DEV	S	QA
XSR (Gul Ahmad, Tariq Rahim Soomro, 2014)	-	-	-	-	-	-
L-Scrumban (Abdullah Albarq & Qureshi, 2018)				✓		✓
Tailoring for agile (Uikey & Suman, 2016)	✓	✓				✓
SXP (Anwer & Aftab, 2017)			✓	✓		

SFDD (Nawaz et al., 2017)			✓	✓	✓	
SCR-FDD (Tirumala et al., 2016)		✓		✓		✓
Hybrid Agile Model (Almasri, 2016)		✓				
XScrum (Jameel Qureshi, 2017)					✓	
TXP (Anwer et al., 2017)				✓		
Hybrid (Mushtaq & Qureshi, 2012)			✓	✓		
C&C (Qureshi et al., 2018)					✓	
IScrum (Ashraf, 2017)				✓		✓
FDMD (Filipe et al., 2015)		✓				
eXSRUP (Sharma & Wadhwa, 2015)					✓	
DXPRUM (Fahad et al., n.d.)	✓	✓	✓			

Methodology

Here, we'll go over the recommended structure. For simplicity, we've depicted it graphically in Section 1. A more precise understanding of the framework can be gained if the actions taken in each step of the framework have been outlined in a table. Each activity comes with a short description as well.

Quality Enhanced Agile Framework as feature-based

Various practical activities are incorporated into the agile framework's lifecycle to address the research questions. A glance at the visual and tabular representations will help us get started. The essential components of the model are summarised in the following. The principles of Scrum and Kanban are implemented using agile methods like these. Feature Assignment is akin to a class assignment in the Feature Driven Methodology. Several development-related tasks are part of XP. The essential components of the suggested methodology are summarised in the following paragraphs.

Visual Representation

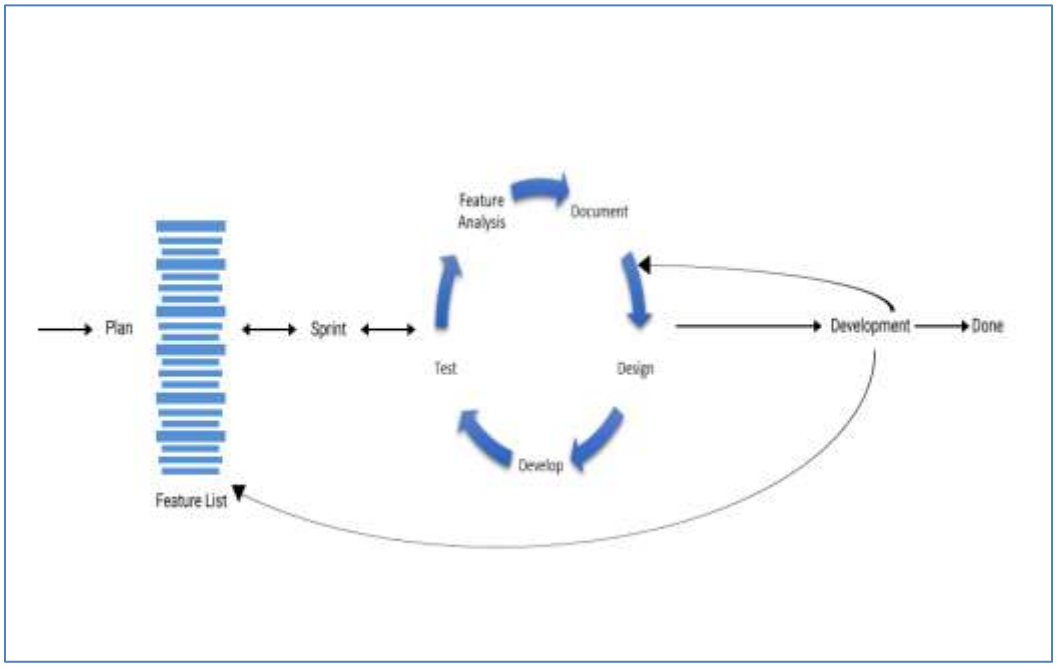

Figure 1: Visual Representation of feature-based Quality-Enhanced Agile Framework

The above image depicts a possible framework. An easy-to-integrate framework that is lightweight, adaptable, and extendable. Project planning comprises domain modeling, estimates, team selection, and the creation of a feature list based on customer needs. This is the first step in building the system. It's then utilized to select a set of Sprint features from a prioritized feature list. During the first phase of a sprint, it's important to extensively research the product. After that, the components are built, tested, and fine-tuned. A release is sent when the Sprint is complete, which is usually after a specified number of days. It can be made available as a service or installed on the computers of end-users. If the feature list grows or the project is declared finished, the following Sprint begins.

It's possible to see all of the suggested framework's actions in this graphic form.

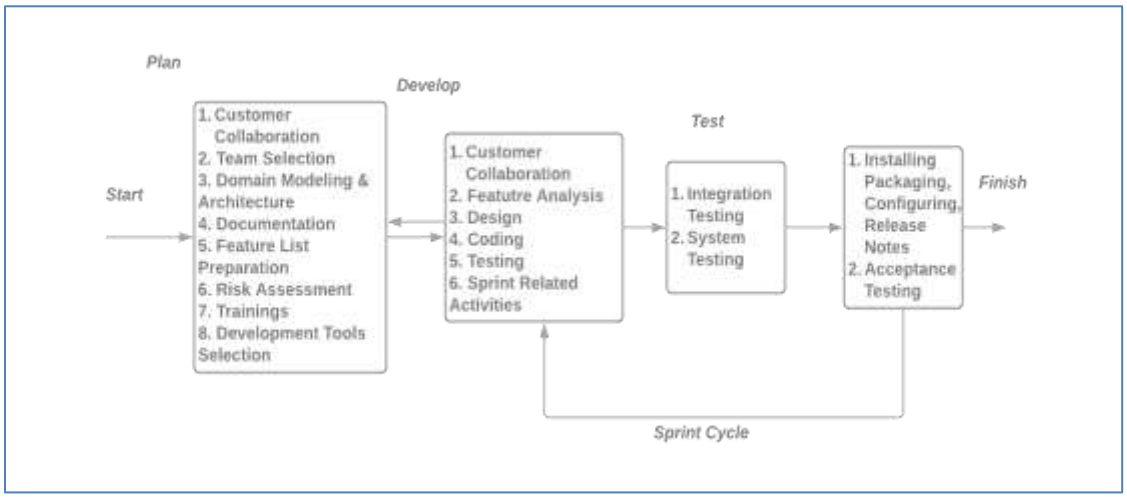

Figure 2: Visual Representation of Framework

Case Studies Overview

Most recommended agile approaches have not been proven via the development of real-time projects. Thus, it is impossible to assess their strengths and drawbacks after reading the solutions presented by various academics. There were just a few projects in which the offered solutions were tested, but they were modest to medium-sized products. The suggested framework is tested in two real-world scenarios. In the first case study, a product was created employing the framework that was proposed. "Netsol Financial Suite" was developed and compared to prior system versions to see how well it worked. Because the product is produced using proprietary software, it is not possible to share the product's detailed information. Canada is now implementing the created project. Another case study was selected from the literature and compared to a real-time telecommunications project to propose an agile methodology. A medium-sized project was involved. In the case study 2 section, you'll find more information about that case study.

Validation

Projects were used to validate the proposed solutions; however, the products are of the small to medium range. The suggested framework is tested in two real-world scenarios.

Case Study1

A product was created using the proposed framework in this case study. "Netsol Financial Suite" was developed and compared to prior system versions to see how well it worked. Because the product is

produced using proprietary software, it is not possible to share the product's detailed information. Canada is now implementing the created project.

Netsol Financial Suite (NFS)

Netsol Technologies is the biggest IT Company in Pakistan, with more than 2500 employees all over the globe. Its products are working in more than 12 countries. NFS is their main product, a lease, and finance software for companies. It is used by several big Lease and Finance firms in the world. This product usually works based on dealer company relationship like Toyota Motor Corporation is a Japanese automotive company with many dealers in many countries. Netsol provides a single solution (combination of 6 applications) to handle the business between these two people. This product is used to save time and cost.

Application Selection of NFS

NFS development is used to evaluate the proposed framework. It was the primary purpose of this development to rearrange the product's architecture in order to increase its performance and to include the most recent technology breakthroughs. Targeted were NFS's POS (Point of Sale) and CMS (Customer Management System) software because they manage 80% of the company's activity. And save the time of an organization.

NFS Versions over the Years

NFS has made three different versions over the years with different tech sets so that it can keep up with changes in lease and finance. Here is a table that tells you about each one of them. R1 was the old system. After it worked well, the project leaders decided to move the technology to a newer one. This new technology moves costs Netsol 1 Year, and the same is true for the rest of the versions. So, Netsol had to make a move. It's the only company in Pakistan that has both ISO and CMMI Level 5. So, they need to change their processes, toolsets, technology stack, and people skillset to meet the needs of their global clients and meet the standards set by the international standardization bodies. The researchers at the company told the project managers to make sure the system was up to date with the most recent trends. Then, they started thinking about updating the system instead of revamping their old one. A lot of money had already been lost because the company had invested in Java Technology. That's why they didn't want to take this risk with so much money. A senior project manager was asked to form an R&D team to make it happen without having to give up anything.

Description of the Activities Performed

The project is now in its conceptual stage. A number of actions will be carried out at this phase to ensure that the team's comprehension of the project is crystal clear. Here's a quick rundown of what's going on during this stage:

Customer Collaboration

Keeping in close contact with the client is critical at every stage of development. "On-site customer" availability can't be achieved owing to a variety of reasons. The concept of business analysts was established to address this issue. As an organization's employees, they are responsible for communicating with clients about the application's features and keeping track of relevant paperwork. Using business analysts, customers are involved in both planning and development. Various modes of communication are employed to maintain constant contact as and when required. Skype, WhatsApp, and Outlook were among the communication technologies employed. Images, audio, and video can all be transferred using these

technologies. There was a team WhatsApp group set up in the event that a resource was unavailable in the office so that he could still communicate with the rest of the team.

Team Selection

Choosing a team is a difficult process. To make this decision, a project manager with considerable domain and management experience should be in charge, as he is well-versed in all aspects of the business (e.g., managing resources). Team selection criteria are defined in order to put this framework into action. Team members with these roles and experiences should be included:

Table 3: Team Selection

Role	Domain Experience	Other Experience
Business Analyst	2+ Years	NA
System Analyst	4+ Years	2+ Year
Pr. Software Engineer	2+ Years	3+ Years
Sr. Software Engineer	1+ Years	2+ Years
Software Engineer	6+ Months	1+ Years
Sr. Quality Assurance	2+ Years	2+ Years
Quality Assurance	6+ Months	1+ Years

Some conventions have been utilized in the Table above, such as Pr for Principal and Sr. for Senior resource. All of the above positions can be game-changers for the company if they are filled by the rightpeople.

Project Level Reusability

Even though this concept was not depicted in the visual representation of the proposed framework, it is valuable due to its advantages. When coding a task within an existing project or starting from scratch, promoting reusability is a great way to cut down on the amount of time and effort it takes to complete the work. The only need is that it be carried out at the appropriate time. In this step of planning, reusability is utilized to select an architecture from a repository of available architectures in the organization. Rather than building a new design, it's best to first employ these architectures so that the time, effort, and expense of this procedure can be reduced. This framework can only be implemented by an organization after creating a repository of popular software designs that they have made themselves or gathering some architectures from open-source coding platforms like GitHub. As a way to speed up the development and deployment of new software and other tasks like testing and tracking progress, logging bugs, and keeping track of changes, a number of software applications were used. In the Table below, you can read about each one.

Table 4: Tool Selection

	Name	Description
1	Visual Studio Professional	Write, debug, run desktop, mobile, web applications
2	SQL Server & Management	Write, debug, run, manage databases
3	Internet Information Service	Writing SQL queries, managing databases
4	Tortoise SVN	Manages code versioning
5	JIRA	Issue tracking, progress reporting
6	Jenkins	Producing automated builds
7	Deployment Utility	It extracts the packages from Jenkins produced to build and deploy them on the client machine
8	Release Notes Builder	It produces the list of completed tasks in the current build and shares their copies to the stakeholders i.e. testing teams, managers, leads, business analysts

The data about the roles in previous versions weren't available, so we could not measure the overall skillset of the teams and compare it with the current version.

- Team Velocity
- Lead Time
- Individual Velocity
- Team Morale

Sprint-related Activities

Assessment of the team's performance, the Sprint's execution, blunders, and so on are all part of this process. Pre- and post-release issues can be addressed more quickly by creating nested sprints inside this structure. This operation adds a small amount of expense, but the benefits it provides more than offset that. The sprint duration in NFS 1.0 was 30 days, while in NFS 2.0, it was 15, and in NFS 2.0 Revamp, it's only seven days. The idea behind this short-term Sprint is the short-term delivery.

Nested Sprints

In this methodology, pre-and post-release faults are addressed through the use of nested sprints. A release's pre-and post-production periods are often stressful, and many resources lack the confidence to tackle these periods. Before releasing a final production version, it is recommended that two-child releases be made.

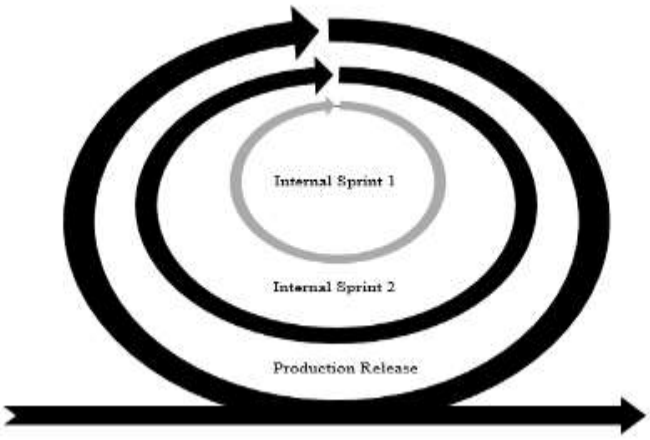

Figure No 3: A figure of nested sprints

Resource Optimization

The purpose of resource optimization is to make sure that only those people are given access to features which are qualified to use them. A fresh developer is given an assignment, and, for whatever reason, he or she is unable to do it on time or with the quality required, the task is given to the most senior resource on hand. Other times, if a developer is unavailable or on vacation, their responsibilities will be transferred to other resources.

Sprint Meetings

For each week, three meetings are held to uncover problems, diagnose them, and develop solutions. The team leaders assign tasks to the resources, who then assess those assignments and describe them in detail during the sessions that follow.

Documentation

There are different versions of domain modeling documents because the project is going on in different countries, like China, the UK, Canada, and others. That's why domain modeling is different in each country. There were four different versions of the document to deal with businesses that were spread across the world. That's why there are two types of documentation, one for the project and one for each Sprint. There are a lot of useful metrics that can be used to measure the documentation. The following metrics are used:

Table 5: Documentation Metrics

Name	Description	Scale
Accessible	Accessible to the intended audience	Scale 1-5
Secure	Secure from unauthorized people	Scale 1-2
Versioning	The document is properly versioned	Scale 1-2
Understandable	The document is understandable by everyone	Scale 1-3
Guidelines	The user is guided throughout the document	Scale 1-2
Layout	The document is properly styled	Scale 1-4

Risk Assessment

The following are some of the metrics that can be used to quantify expected risks:

Table 6: Risk Assessment

Name	Description	Measurement
Infrastructure	Measures various types of infrastructure	Scale 1-4
IT	Measure security of IT related equipment's	Scale 1-3
Workplace	Measure Workplace Safety	Scale 1-3
Process Quality	Measure Customer Satisfaction	Scale 1-2
Business	Measure the risks associated with the business.	Scale 1-4

Assessments

As the name implies, this is where all of the project's many aspects are evaluated. The following activities are included:

a. Iteration Goal Assessment

After we finish the iteration, we will find out whether we were successful or not. We will also find out if the iteration-related goals were met or not.

b. Feature Marking

It means marking the parts that are done. Update their statuses and any relevant documentation. If there are any problems, fix them and put them on the feature list for the next Sprint, if there are any at all.

c. Change Management

Next, features are added to the feature list for the next iteration or Sprint. They are reprioritized, resources are placed, and features are given a priority for the next time around.

Time-related Matrics

These metrics include information on the team's progress, time spent, and so on from JIRA. These indicators are as follows:

a) Acceleration

This metric is used to track the team's progress during each release. It's quite good at forecasting the amount of work that a team can complete in upcoming updates.

b) Required for Delivery

It is the time period between receiving requirements and converting them into deliverable software. It includes cycle time

c) Duration of the cycle

It includes the time required to entertain and deploy new software. Continuous delivery enables teams to complete this task in a matter of seconds.

d) Velocity of Individuals

TheNNumber of days a resource works, theNNumber of lines he writes, and theNNumber of issues he reports are the metrics used to determine his velocity.

e) The morale of the Team

Team morale is the most critical metric of all because if resources are pleased while working for the firm if they care about the environment and the rewards, they will undoubtedly deliver high-quality work. Otherwise, all other metrics will be in vain. According to research, a happy employee is more collaborative, creative, and loyal.

f) Rates of opening/closing

It quantifies theNNumber of reported and resolved production issues over a certain time period.

g) Planned-to-do Ratio

ThisNNumber indicates how many features were created in accordance with a plan and how many were not.

Sprint Velocity

The current methodology keeps the team focused on the tasks at hand. Only the last Sprint completed feature was 4. That's because there was an issue logged related to that particular feature at the moment of the system testing cycle, but that was completed well before the production release.

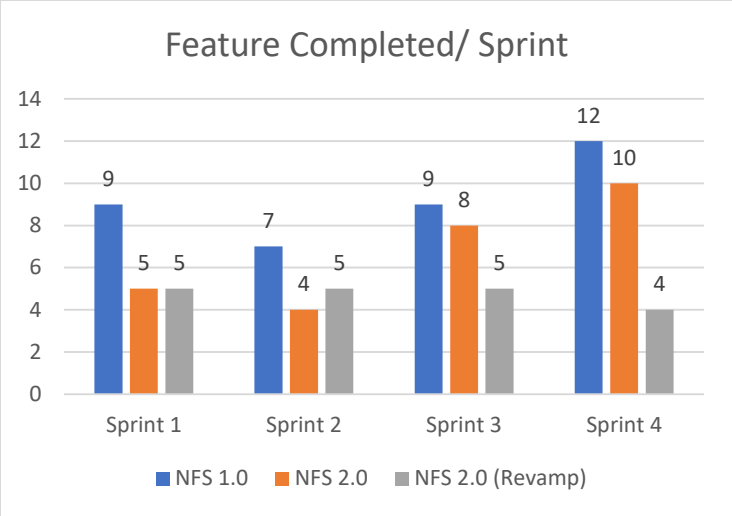

Figure 4: Sprint Velocity

Sprint Goals

As mentioned earlier, sprint goal measurement wasn't part of previous versions. This measure had been made to understand whether all the resources for a particular task are on the same page or not. Team Lead gives a point after every meeting about this metric, and it's calculated after the end of each Sprint so that this process can further be improved based on this measure. The scale was set for this measure from 1-3.

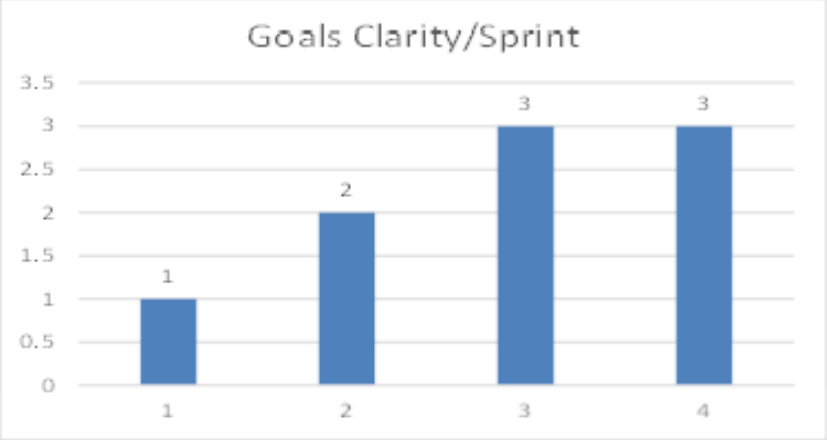

Figure 5: A figure of sprint Goals for clarity

As you can see, the first two meeting goals weren't clear to the team, or they had some questions in mind that were addressed later. Within sprints 3 and 4, the team improved their understanding of the Sprint, the tasks in hands, and the feature information.

Overall Sprint Results

This section presents the results of sprint-related measurements collectively. Sprint feature and sprint goals metrics were not available in other versions of NFS. That's why they are omitted. See the table below.

Table 7: Sprint Results

	R1	NFS 1.0	NFS 2.0	NFS 2.0 Rev
Sprint Duration	NA	30	15	7
Feature Division	NA	NA	NA	Yes
Sprint Velocity	NA	9.25	6.75	4.75
Nested Sprints	NA	NA	NA	Yes

Customer Satisfaction

Measures customer satisfaction based on the comments received from them after each release. Six possibilities are included in the table below, and each of them is graded on a scale from 1 to 5, as shown.

Table 8: Customer Satisfaction

Please rate the quality of the product you reviewed	
1	Poor
2	Fair
3	Good
4	Great
5	Excellent

Time spent in Documentation

It shows how much time is spent on each version throughout the course of a sprint. The amount of time spent is expressed in days. Compared to alternative frameworks, the proposed one took the least amount of time versions.

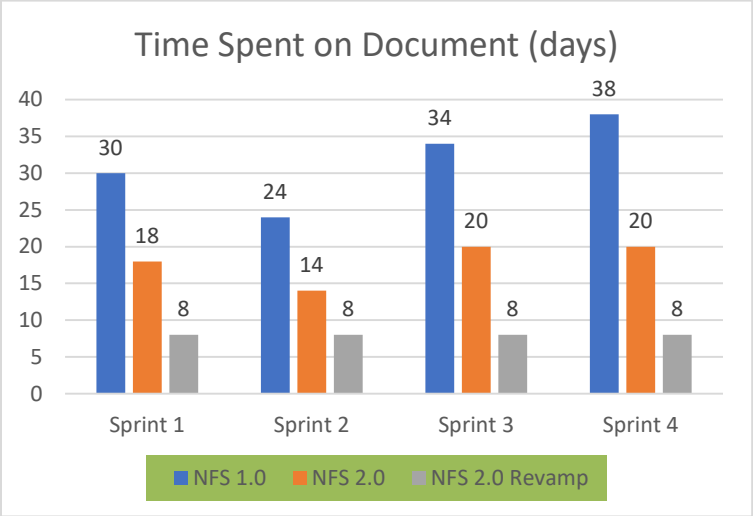

Figure 6: A figure of fixing issues according to time

Overall results of documentation

In this part, all the measurements made on paperwork are consolidated into a single location. The following tables detail documentation metrics at the project and sprint levels:

Table 9: Project Level Documentation Results

3	Metric	4	R1	5	NFS 1.0	6	NFS 2.0	7	NFS 2.0 Rev
8	Accessability	9	2	10	2	11	3	12	4
13	Security	14	0	15	0	16	1	17	2
18	Versioning	19	0	20	0	21	1	22	2
23	Understandability	24	1	25	1	26	2	27	4
28	Styling	29	1	30	2	31	3	32	4
33	Obtained Score	34	4	35	6	36	10	37	16
38	The total Score was 16, and the obtained Score shows the quality of the documents produced								

Table 10: Sprint Level Documentation Results

Project Versions	Doc Add	Pag Add	Tech Doc Add	Time Spent (days)
NFS 1.0	63	756	0	16
NFS 2.0	29	432	0	72
NFS 2.0 Revamp	20	180	12	32
Doc Add= Documents Added, Pag Add = Pages Added, Tech Doc Add = Technical Documents Added, Time Spent = Time spent in creating the documents				

Team

The methodology section outlined the criteria for selecting the 27 people who would be involved in the implementation of the framework. They are well-versed in both the industry and the underlying technology. As a result of the newness of the people and procedures engaged in the development, the earlier versions of the system's implementation took far too long due to the large team sizes selected. You can see this in the graph.

Table 11: Team Selection

Role	Size	Description
Project Manager	1	Plan, Budget, Monitor the progress, and satisfy the upper management
Team Lead(s)	2	Responsible prepare a building plan within the allocated timeline, distributing the work to the team according to their skills, and follow-up on their progress.
Quality Assurance	4	Responsible for testing the developed solution that it actually works
Software Engineer(s)	5	Responsible for building the solution that the customer needs
Sr.Software engineer(s)	6	Capable of handling complex tasks without any supervision.
Pr.Softwar Engineer (s)	4	Responsible for smart resource utilization and meeting strict deadlines.
Business Analyst(s)	5	Responsible for guiding directing the resources regarding business concerns and keeping them safe from distractions. Represents customer needs.

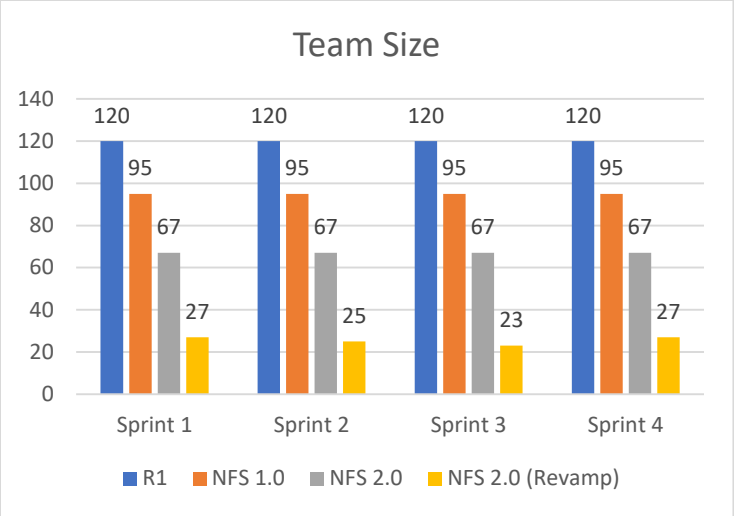

Figure 7: Team Sizes in 4 Sprints

To figure out how well a team was at their jobs and compare it to how well they were at their jobs now, we didn't have any information about how roles worked in previous versions.

Number of features completed per Sprint

Previous versions had a different number of features than the current version, but we stayed consistent in our efforts to add and complete features on schedule. Because of a bug of low severity, only one feature was reopened, and it was only partially approved as completed. Take a look at the completion rate in the figure below:

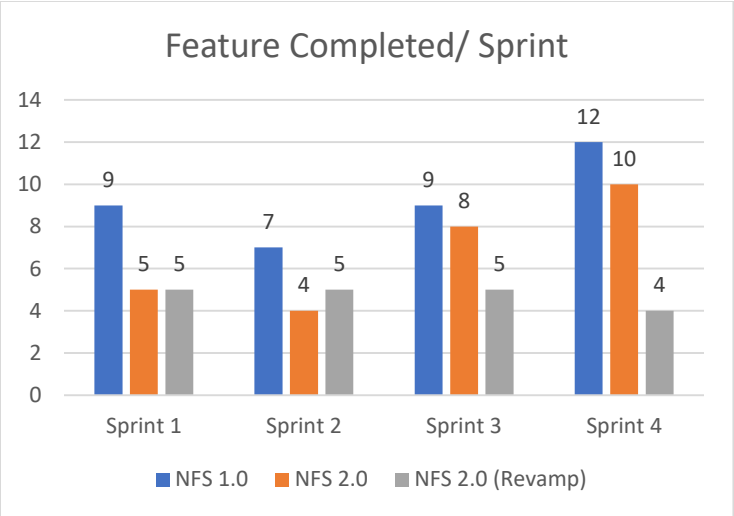

Figure 8: Features Completed/Sprint

Number of features reopened per Sprint

This metric tells us how many features were not declared complete throughout testing, which is valuable information. In the earlier version of NFS, there is an alarming number that appears. Because some of

Netsol's clients are still utilizing NFS 1.0 and NFS 2.0, it is necessary to fix this issue. In sprint 4, one issue was raised in relation to the feature, and it was resolved as soon as it was identified and documented.

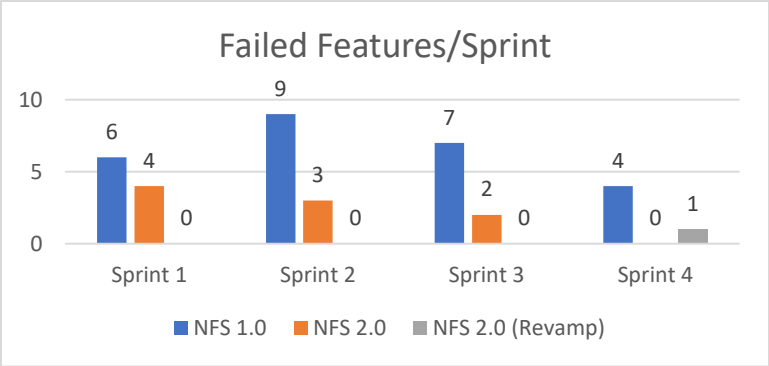

Figure 9: Features Failed/Sprint

Overall Feature Results

This section brings together the results of some of the measures that were taken in the previous sections. Because of their nature, some of those metrics cannot be merged; however, the remaining measurements are listed in the following Table:

Table 12: Feature Results

	R1	NFS 1.0	NFS 2.0	NFS 2.0 Rev
Fteatures Added	NA	63	36	20
Features Completed	NA	37	27	19
Features Failed	NA	26	9	1

Time spent on code review

The amount of time that senior resources spend evaluating the code is captured in this metric. It's the junior's job to check out the code written by the senior. This section does not include the time spent in pair programming. In order to develop the skills of junior resources, this method is utilized to write and review code efficiently. In addition to improving readability, this exercise also helps identify code gaps that could lead to production errors if executed. Time is shown in hours on the graph below.

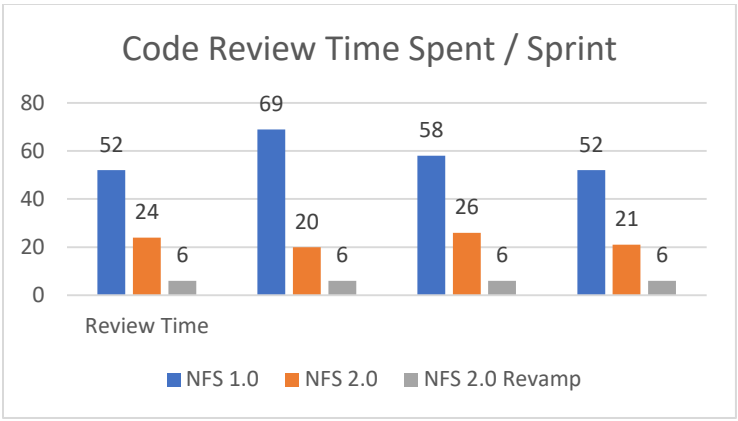

Figure 10: Time Spent on Code Review

Code Quality

The task of determining the code's quality is a complex one, as it depends on a wide range of factors. We can assess this attribute by devising a meter of our own. As a whole, we looked at things like the Maintainability Index, Coupled Codes, Code Complexity, Depth of Inheritance, and Defects Caught to gauge the overall quality of the code. The quality is rated on a scale of one to five. The scale's description is presented in the following paragraphs:

Table 13: Code Quality

Scale 1-5	
1	Low
2	Normal
3	Average
4	High
5	Very High

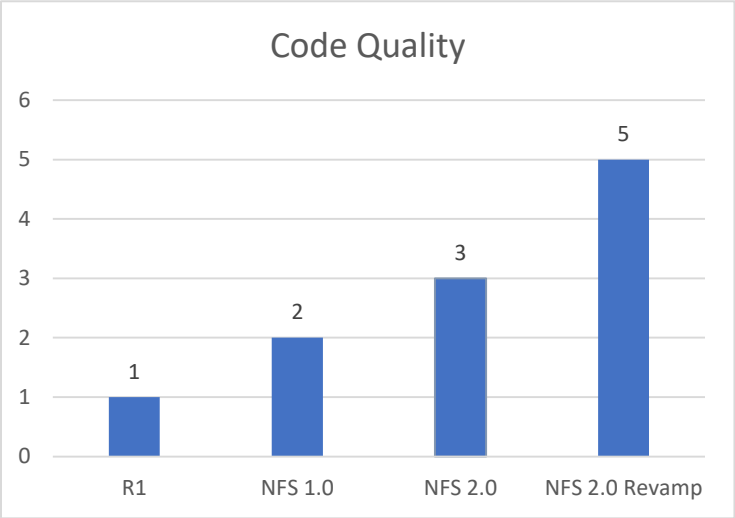

Figure 11: Code Quality

Overall Results

This section computes the totalNNumberof measurements taken during the coding process. The measurements that fluctuate depending on the Sprint are not combined, but rather the measurement from the final Sprint is taken instead. Please take a moment to check the following naming convention table before proceeding to the measurement: All the related code measurements are presented in tabular form below:

Table 14: Coding Results

	R1	**NFS 1.0**	**NFS 2.0**	**NFS 2.0 Rev**
CC	NA	1192	570	142
MI	NA	30	71	83
COM	NA	4992	3719	2264
DIN	NA	11	7	4
INT	NA	47	31	12
CLS	NA	126	58	108

BCLS	NA	57	29	14
CCLS	NA	69	29	96
LOC	NA	29473	26118	18499
KLOC	NA	29.473	26.118	18.499
CDUP	NA	15.6	18.8	10.9
TMCR	NA	52	21	6
DDPS	NA	NA	NA	14 avg
DDH	NA	NA	NA	4
NPE	NA	3380	945	119
DFPL	NA	NA	NA	328 avg
CODQ	1	2	3	5

Customer Satisfaction

Customer satisfaction is a measurement that determines how happy customers are with a company's products, services, and capabilities. Customer satisfaction information, including surveys and ratings, can help a company improve or change its products and services.

Case Study 2

Case study 1 was unavailable to the general audience because of the confidentiality agreement. Because of this, the researcher used a previously published case study to suggest an agile technique he termed "DXPRUM" (Muhammad Fahad, Salman Qadri, Saleem Ullah, Mujtaba Husnain, Rizwan Qaiser, Shehzad Ahmed Qureshi, Waqas Ahmed, 2017). The new method was based on DSDM, but it was improved. A real-time project approach was used, and the outcomes were compared to those of DSDM to demonstrate the effectiveness of the suggested technique. DXPRUM results are then compared to the proposed approach, as seen in this example.

Table 15: Case Study 2 Comparison

	Parameter	**DXPRUM**	**Proposed Framework**
1	Size of project	Medium	Large
2	No of Iterations	4	4
3	ED (in Weeks)	7	4
4	AD (in Weeks)	6	4
5	Team Members	4	27
6	Programing Language(s)	HTML, CSS, jquery, Ruby O Rails	Angular, C#, SQL, XAML
7	Database	MySQL	MS SQL Server
8	Web Server	Apache	IIS
9	Customer's Satisfaction	92%	95%
10	Post-Release Defects	13	8
11	Performance / Quality (Post-Release Defects / KLOC)	0.237	0.081

The proposed approach is compared in the above Table. Despite the fact that more data was available for the generated case study, all of the parameters remained the same. However, the same parameters were used to make an exact comparison.

Despite the fact that this was a massive project, the comparison to a medium-sized one shows that the recommended methodology may be used for projects of all sizes.

There is a noticeable change in parameters in the above Table. Every project has a significant impact on the schedule. Despite being a major project, the "Actual Time" parameter is shorter than DXPRUM's actual duration. Because this was a major project, the team size was more than the team size of DXPRUM.

The character played a different part as well. DXPRUM didn't have a product owner, but they did have the function of business analyst. The role of the product owner is similar to that of the business analyst in many ways. "Customer Satisfaction" was found to be 95 percent in this case study and 92 percent in DXPRUM, which is an improvement in the proposed solution. It is critical for project evaluations to obtain regular input from the BAs in order to gauge their level of satisfaction.

Quality is still another critical consideration. In addition, the proposed method improves the situation because it provides a visual representation of each work. They know how to fix a bug and how much it affects other sections of the system since they have a deep understanding of the system.

As previously stated, "Time to fix" and other parameters were not included in the DXPRUM case study comparison.

Table 16: Frameworks support metrics

Frameworks	Espresso	Mockito	Robo electric	Monkey runner
Integration Testing	Yes	No	Yes	No
Use Espresso methods	Yes	Yes	Yes	Yes
Scripting Language	Java	Java	Java	Java
Supported API Levels	Gingerbread or higher	Jelly Bean or higher	Jelly Bean and higher	Astroid and higher
Test Creation Tool	Espresso recorded tool	Android studio	Android studio	CLI

We have discovered that the Monkey Runner tool needs activity and a count of the Numberof hits after implementing and running it. These hits will be applied at random to various widgets (components) that are part of the action you specify. Using the random impact will also put the program in airplane mode, turn on the Wi-Fi, and do other actions while the tool runner is running, as shown below. In Android GUI testing, monkey runner is a common practice. Using the monkey runner tool is only necessary for the interface. Espresso testing for GUI is slowed down by the need for a real android environment to perform the tests, making them more time demanding to write. For espresso GUI testing, Roboelectric is an alternative. It uses the Java Virtual Machine (JVM) to run tests. Manually written test cases are now being created for each new release. Currently, there is no way to add assertions to the various components of activity because espresso has been provided as a recorder. Mockito-like functionality is included in Android. Uses for mockito include unit and system tests. We use mocks to test different scenarios, such as when we are integrating the UI with data from the database, for example. Using the JVM backend, the Mockito framework is utilized to load resources and test many user interface components. As a mocking framework, Mockito is not the best choice for integrating testing.

Future Work and Conclusion

The Monkey runner is one of the greatest tools to utilize if you need to test a single activity and the aim of your testing is to test the UI components presented on your activity randomly. The Espresso recorder for android applications, on the other hand, is a better choice if your testing involves adding assertions to

various UI elements to validate their correctness. It can be used in Roboelectirc if you want to run the UI test rapidly and have a continuous integration environment. It simulates the Android environment on local PCs to execute UI tests. The JUnit and Mockito frameworks have been shown to be the best fit for testing at the unit and integration testing levels, based on our research. Researchers can use this study's findings in the development of frameworks for GUI testing, such as UI Automator and other fake frameworks. The extent of the work that can be done with these frameworks and methodologies is much broader. Testing frameworks that produce better results than Test-Driven Development (TDD) do as well. It is built on Android apps, but we also look into operating systems like iOS. A focus on the in-depth study is therefore essential, and future researchers should take this advice to heart. Frameworks given by iOS for GUI and TDD should be researched.

References

Abdullah Albarq, A., & Qureshi, R. (2018). The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development. *International Journal of Information Engineering and Electronic Business*, 10(3), 23–35. https://doi.org/10.5815/ijieeb.2018.03.04

Alam, S., Nazir, S., Asim, S., & Amr, D. (2017). Impact and Challenges of Requirement Engineering in Agile Methodologies: A Systematic Review. *International Journal of Advanced Computer Science and Applications*, 8(4). https://doi.org/10.14569/IJACSA.2017.080455

Albadarneh, A., Albadarneh, I., & Qusef, A. (2015). Risk Management in Agile Software Development : a Comparative Study.

Almasri, A. K. (2016). A Proposed Hybrid Agile Framework Model for Mobile Applications Development. *International Journal of Software Engineering & Applications*, 7(2), 1–9. https://doi.org/10.5121/ijsea.2016.7201

Anwer, F., & Aftab, S. (2017). SXP: Simplified Extreme Programing Process Model. *International Journal of Modern Education and Computer Science*, 9(6), 25–31. https://doi.org/10.5815/ijmecs.2017.06.04

Anwer, F., Aftab, S., & Ali, I. (2017). Proposal of Tailored Extreme Programming Model for Small Projects. *International Journal of Computer Applications*, 171(7), 23–27. https://doi.org/10.5120/ijca2017915112

Ashraf, S. (2017). IScrum: An Improved Scrum Process Model. *International Journal of Modern Education and Computer Science*, 9(8), 16–24. https://doi.org/10.5815/ijmecs.2017.08.03

Cahyana, N., Khomsah, S., & Aribowo, A. S. (2019). Improving Imbalanced Dataset Classification Using Oversampling and Gradient Boosting. Proceeding - 2019 5th International Conference on Science in Information Technology: Embracing Industry 4.0: Towards Innovation in Cyber Physical System, ICSITech 2019, 217–222. https://doi.org/10.1109/ICSITech46713.2019.8987499

Chauhan, B. D., Rana, A., & Sharma, N. K. (2017). Impact of Development Methodology on Cost & Risk for Development Projects.

Cram, W. A., Newell, S., Cram, W. A., & Newell, S. (2017). Mindful revolution or mindless trend ? Examining agile development as a management fashion Mindful revolution or mindless trend ? Examining agile development as a management fashion. 9344. https://doi.org/10.1057/ejis.2015.13

Fahad, M., Qadri, S., Shah Muhammad, S., & Husnain, M. (n.d.). Software Quality Assurance of Medium Scale Projects by using DXPRUM Methodology.

Filipe, J., Maciaszek, L., Institute for Systems and Technologies of Information, C. and C., IEEE Computer Society. Technical Council on Software Engineering, SIGAPP., ACM Sigsoft., Association for Computing Machinery. Special Interest Group on Management Information Systems., Association for Enterprise Information., SCITEPRESS (Firm), Institute of Electrical and Electronics Engineers., International Conference on Enterprise Information Systems (17th : 2015 : Barcelona, S., & GISTAM (Conference) (1st : 2015 : Barcelona, S. (2015). FDMD: Feature-Driven Methodology Development. 2015 International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE).

Fitzgerald, B., & Stol, K. (2017). The Journal of Systems and Software Continuous software engineering : A roadmap and agenda. *The Journal of Systems & Software*, 123, 176–189. https://doi.org/10.1016/j.jss.2015.06.063

Gul Ahmad, Tariq Rahim Soomro, M. N. B. (2014). XSR: Novel Hybrid Software Development Model (Integrating XP, Scrum & RUP). *International Journal of Soft Computing and Engineering (IJSCE)*, 2(3), 5. https://doi.org/10.5815/ijitcs.2012.08.05

Hoda, R., & Noble, J. (2017). Becoming Agile : A Grounded Theory of Agile Transitions in Practice. https://doi.org/10.1109/ICSE.2017.21

Hussain, W., Shahin, M., Hoda, R., Whittle, J., Perera, H., Nurwidyantoro, A., Shams, R. A., & Oliver, G. (2021). How Can Human Values Be Addressed in Agile Methods ? A Case Study on SAFe. 1–18.

Jameel Qureshi, M. R. (2017). Evaluating the Quality of Proposed Agile XScrum Model. International *Journal of Modern Education and Computer Science*, 9(11), 41–48. https://doi.org/10.5815/ijmecs.2017.11.05

Kumar, G. (2014). Comparative Analysis of Software Engineering Models from Traditional to Modern Methodologies. https://doi.org/10.1109/ACCT.2014.73

Muhammad Fahad, Salman Qadri, Saleem Ullah, Mujtaba Husnain, Rizwan Qaiser, Shehzad Ahmed Qureshi, Waqas Ahmed, S. S. M. (2017). A Comparative Analysis of DXPRUM and DSDM. *IJCSNS International Journal of Computer Science and Network Security*, 17(5), 259–264.

Mushtaq, Z., & Qureshi, M. R. J. (2012). Novel Hybrid Model: Integrating Scrum and XP. *International Journal of Information Technology and Computer Science*, 4(6), 39–44. https://doi.org/10.5815/ijitcs.2012.06.06

Nawaz, Z., Aftab, S., & Anwer, F. (2017). Simplified FDD Process Model. *International Journal of Modern Education and Computer Science*, 9(9), 53–59. https://doi.org/10.5815/ijmecs.2017.09.06

Nerur, S., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213–1221. https://doi.org/10.1016/J.JSS.2012.02.033

Qureshi, R., Basheri, M., & Alzahrani, A. A. (2018). Novel Framework to Improve Communication and Coordination among Distributed Agile Teams. *Information Engineering and Electronic Business*, 4, 16–24. https://doi.org/10.5815/ijieeb.2018.04.03

Rafi, U., Mustafa, T., Iqbal, N., & Zafar, W.-U.-I. (2015). US-Scrum: A Methodology for Developing Software with Enhanced Correctness, Usability and Security. *International Journal of Scientific & Engineering Research*, 6(9).

Rauf, A., & Alghafees, M. (2015). Gap Analysis between State of Practice & State of Art Practices in Agile Software Development. 102–106. https://doi.org/10.1109/Agile.2015.21

Sampietro, M. (2016). The Adoption and Evolution of Agile Practices. *In PM World Journal The Adoption and Evolution of Agile Practices*: Vol. V.

Sharma, N., & Wadhwa, M. (2015). eXSRUP: Hybrid Software Development Model Integrating Extreme Programing, Scrum & Rational Unified Process. *Indonesian Journal of Electrical Engineering and Computer Science*, 16(2), 377–388. https://doi.org/10.11591/IJEECS.V16.I2.PP377-388

Sridhar Nerur, B., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of Migrating to Agile Methodologies. 48(5).

Stavru, S. (2014). A critical examination of recent industrial surveys on agile method usage. *Journal of Systems and Software*, 94, 87–97. https://doi.org/10.1016/J.JSS.2014.03.041

Tirumala, S. S., Ali, S., & Babu, A. (2016). A Hybrid Agile model using SCRUM and Feature Driven Development. *In International Journal of Computer Applications* (Vol. 156, Issue 5).

Uikey, N., & Suman, U. (2016). Tailoring for agile methodologies: A framework for sustaining quality and productivity Service oriented computing View project Tailoring for agile methodologies: a framework for sustaining quality and productivity. *International Journal of Business Information Systems*, 23(4), 432–455. https://doi.org/10.1504/IJBIS.2016.080216

Wang, X., Flora, H., Heland, J., Ortega, J. C., Zosa, M. G. M. De, & Carpio, J. T. (n.d.). Blended Agile Approach : A Mobile Application Approach Related papers Blended Agile Approach : A Mobile Application Approach.