

AN ITERATIVE HYBRID AGILE METHODOLOGY FOR DEVELOPING ARCHIVING SYSTEMS

Khaled Ebrahim Almajed, Walaa Medhat and Tarek El-Shishtawy

Faculty of Computers and Informatics, Benha University, Information Systems
Department, Egypt.

ABSTRACT

With the massive growth of the organizations files, the needs for archiving system become a must. A lot of time is consumed in collecting requirements from the organization to build an archiving system. Sometimes the system does not meet the organization needs. This paper proposes a domain-based requirement engineering system that efficiently and effectively develops different archiving systems based on new suggested technique that merges the two best used agile methodologies: extreme programming (XP) and SCRUM. The technique is tested on a real case study. The results shows that the time and effort consumed during analyzing and designing the archiving systems decreased significantly. The proposed methodology also reduces the system errors that may happen at the early stages of the development of the system.

KEYWORDS

Requirement Engineering (RE), Agile, SDLC, Extreme Programming (XP), SCRUM, Archiving.

1. INTRODUCTION

Requirements engineering is a collection of methods related to describing and discovering the main objective of the system under developing. It works as the link between customers' real-world needs and other clients involved in the system such as software owner or software developer [1].

The title "requirements engineering" may appear a slightly heavy. The two words provide some poor meanings. Requirement points to somebody out there making the requiring: a special client who understands what he requires. In several projects, the client knows requirements as the list of needs. In tradition, very few systems have an individual client, but rather a different group of people who will be involved in one way or another by the software aka stakeholders. Stakeholders may have different and inconsistent needs or goals. Their goals or needs may not be clear or may be hard to explain. Maybe they do not know what they need or what can be achieved. Therefore, the inquiry about what they 'need or require' is not likely to be useful [2].

"Engineering" refers to an engineering system in its own right, considering it is really a part of a bigger method of engineering software-intensive systems. Engineering term also recommends that the requirements engineering process outputs required to be fully engineered, where those 'outputs' are normally known to be detailed specifications. In different projects, a big deal of care

is assured when assigning terms. That is because sometimes misunderstandings lead to major problems that could lead to a problem of security. It could be reasonable not to record complete terms or specifications in other projects. In various requirements engineering methods, it is the understanding that is reached from using the necessary systematic technique for analysis that is necessary, rather than the documented terms or specifications [3].

Agile is used for specifying means of planning and making work wherein it is agreed that doing modifications as required is an essential piece of the task. Agile in business indicates that an organization is regularly in a state to perform market changes. Agile in software development is adjusted to express the capability to respond to technology, people, and requirements changes. The most known agile methodologies are XP and SCRUM.

Extreme Programming [4] agile methodology founders pointed to develop a proper Object Oriented (OO) technique for projects where teams utilized by a little or dozen developers in one place. The technique is based on five underlying concepts: simplicity, communication, courage, respect, and feedback. SCRUM is a methodology for agile requirement engineering to ensure teamwork and iterative incremental progress that ensure reaching the client goals for developing any software project, SCRUM like any agile methodology can handle the multiple changes for the client requirement. SCRUM has three core concepts which are inspection, transparency, and adaptation.

Archiving Systems are very important nowadays specially for large and governmental institutions. They make use of huge structured and unstructured data [5]. The unstructured data or files need to be archived electronically to facilitate the automation of the work process. Since these institutions have many stakeholders, the process of gathering requirements is not easy.

This paper proposes a hybrid Agile methodology that combine the two best methodologies SCRUM and extreme programming. We called the methodology “An Iterative Hybrid Agile Methodology for Developing Archiving Systems”. HyArchive is an iterative and incremental hybrid agile software development framework merging between SCRUM and XP to solve the problem of gathering requirements to build archiving systems for large governmental institution.

The paper is organised as follows: An overview of the previous works related to our subject is presented in section 2. In section 3, a comparison between Agile methodologies is presented. In section 4, HyArchive Methodology is presented. In section 5, three different case studies are presented, before drawing the conclusion in section 6.

2. RELATED WORKS

In [6] the authors showed that the department of information technology is a critical and important factor that affects the practices of agile, and the current researches have not explored the relationship between the usage of agile practices and information technology department. They showed that how the main propose of their study is to explore/test two different types of agile practices (technical and social) and four competing culture forms. They also showed how the findings add value to the current literature by combining the competing values with the

literature which affects the information technology department level and agile development level. Authors concluded the research with listing the research limitations, as the sample consisted of

only the US-based firms. The researchers identified a potential pool of participants using LinkedIn, that's beside some departments of information technology that may only adopt agile practices for less information technology projects.

In [7]the authors examined the previous work and found that forty-seven studies have examined the identification of stakeholder techniques in Requirements Engineering. They concluded that it is very important to consider that because the sources are unstructured and the high level of the analysed studies that are not heterogeneous, meta-analysis methods are not applied. Their findings, included an examination of the shortcomings found in this systematic literature review, produce definite evidence to support additional study in the development of new techniques to appropriately complete Stakeholder identification. They showed that all the analysed papers show the discrepancies between current stakeholders involved in software development as each one have various interests and priorities.

In [8]the authors showed that methodology selection of software development for all projects is important and has a high impression of client satisfaction and the selection mainly done by experience or in many cases a rational particularly in the third world countries such as Sudan. They showed that Sudan has various companies that still in the start-up phase or have small resources and they have a difficulty in software methodology choice. They also showed that Sudan government is the principal buyer of local software products and the government is shifting the industry of software to offer high-quality products to decrease the process of buying the software from foreign organizations. They showed how the small or medium companies use the Agile techniques on software development and how these techniques or methods is a way of getting a guaranteed high-quality product.In addition, how the Agile methods are the most known techniques nowadays that is because agile methods fit with the change of requirement and it only works to satisfy the clients' needs. They were mainly focused on filling the gap of information missing in existing literature reviews by making a review on most known and most used agile methods.

In [9] the authors constructed a case study on an e-business project to introduce trading based on web, logistics, and marketing to medium and small-sized enterprises. They showed that the project will help them in building a collection of new application modules combining with their legacy information systems, which will increase their IT platform and allow them to achieve their system aims. The authors found that the used method was more helpful and they also merged the advantages of various modeling method.Thus enhanced the outcome of modelling work and afford a full understanding of the process of business to the end user, and developer.

In [10] the authors tackled that Agile SCRUM methodology and agile extreme programming methodology when used on the project, the productivity is improved.They compared and analyzed the two methods to figure out the discrepancies and the relationship between them.They have applied the two methods on the same project and compared the acceptance of each method on the project.

In [11] the authors developed a methodology that supported the Enterprise Resource Planning system arrangement functions to client specifications. Authors developed a systematic approach that can help to recognize the new functions of the Enterprise Resource Planning system, through which a conventional combination of the requirements three parts are guaranteed. They have established a post-traceability connection between Enterprise Resource Planning functionality and requirements, giving an intended view of what the Enterprise Resource Planning system can do. They proposed the requirements process of matching that mainly Enterprise Resource Planning driven. They decreased the number of deviations from the Enterprise Resource Planning functionality in a provided installation besides taking the advantage of the full Enterprise Resource Planning functionality power.

In [12] the authors showed how the agile methodologies are widely used these days in IT industry and show how each method has its benefits and its limitations that need a solution. They clarified that SCRUM agile and extreme programming agile methods are the most agile used methods in the business and both of them can be used if the client did not have a clear requirement of his/her project. They introduced a new agile method that merges between SCRUM and XP and showed how the new method can be applied to increase the productivity and be useful if applied with small or large projects.

In [13] the authors showed how companies or organizations that work on software development can offer the production of a helpful software on short time with decreasing the cost. They showed how the agile methodologies improved the process of software production. They listed the benefits, advantages, and the disadvantages of the agile methodologies especially SCRUM and XP. They also showed the importance of agile methods and how these methods divide the large project into small tasks to finish and deliver each task in its time with a high quality.

In [14] the authors showed how requirement engineering is an important field and critical phase of the production process of any software. They showed how its importance increases more and more because of the changes in the project requirement during the production phase. How the traditional process and methods cannot solve the problem of changing requirement. They studied the agile methods such as SCRUM and XP clarifying their importance to the requirement engineering methods. They concluded that extreme programming method was beneficial to small projects and can be used with large projects but after defining its limitations.

In [15] the authors showed the importance of the SCRUM agile method and how it's critical when other researchers think about making or producing a new hybrid agile methodology. SCRUM is the best from the author's opinion based on a review they already made. Their research objective was to provide a full description of agile methods and how SCRUM method is the most used, merged, helpful, productive and beneficial method among the other agile methods. They also showed the importance of merging two or more agile methods to improve the performance of software production. Their conclusion showed how the SCRUM method was the most used one while crystal and Kanban ere not mentioned.

In [16] the authors showed how agile accelerated the software delivery process, addresses software changes and increases productivity. The authors' objectives were to design an approach

that can be combined with agile methodologies for software development in order to increase productivity. Their results showed how agile SCRUM is the most frequently integrated method and when SCRUM used, the usability and quality improved. They also showed the importance of using the DT approach and how it improved the communication between the customer and the development team during the whole cycle of software development.

3. AGILE METHODOLOGIES

There are many Agile methodologies in the literature as illustrated in Figure 1. In the next subsections, we will tackle the best used methodologies from the literature Scrum and XP then, we will compare between them.

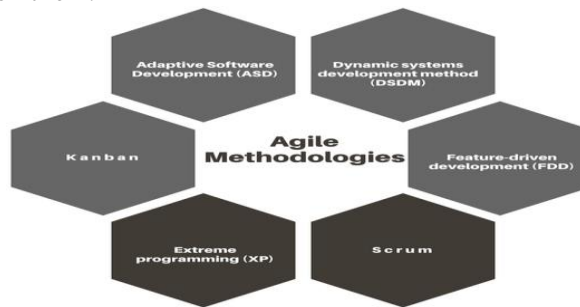


Figure 1. Agile methodologies

4. SCRUM

The Scrum process that carry out the requirements with a very high quality in compliance with the most used standards is shown in the following steps [17]:

Planning the tasks for the product owner: when planning the tasks for the product owner any newly formed team found a challenge on how to find its ability/capacity for the Sprints development or how many items can be scheduled in a Sprint. Therefore, it is recommended to run very short Sprints at the first, which gives the team the ability to adjust the time for the Sprints in order to finish in a timely manner. On the other side, the Product Owner works with his Stakeholders so that they can maximize the benefits and works with the team to find the best solution for a requirement. A requirement must be worked on so that it can deliver a return at the end.

Organization of the product backlog: The Multi-Level Product Backlog is an important tool for the Scrum built-in requirements engineering process. An effective Product Backlog must have a good organization and structure and this can be used for both the development and organization of Sprints. The product backlog has three different stages. Stage 3 where the requirements are collected, Stage 2 where the requirements specified, and Stage 1 where the requirements

transferred into a state through quality gates between the levels, which meet the requirements of the Team regarding estimates that are more precise.

Quality gates: there are quality gates between the three-staged product backlog these gates are checklists that verify the quality of requirements, for example in Stage 3 the product owner collects unspecified features, ideas, and change requests. The product owner should have a minimal set of information attached when an item enters the product backlog at this stage, then he works on the refinement of this information if it has considered important for him. At the end when an item is important enough to enter the stage of detailed requirement engineering, the following quality gate has to be passed.

5. EXTREME PROGRAMMING (XP)

Extreme Programming (XP) is an agile software development framework that develop a methodology suitable for Object Oriented (OO) projects and aims to produce a software of a high quality using teams of a dozen or fewer programmers in one location. Extreme Programming is the most specific of the agile frameworks and it is based on five underlying values [18].

Extreme programming consists of small, extended development team that works in the same place. Software requirements are dynamically changing. Extreme Programming five values are simplicity, communication, courage, respect, and feedback. Extreme programming main players are the Customer, the Developer, the Tracker, and the Coach. Although XP is based on specifying particular practices for the development team to follow, it is not establishing specific roles for any of its players.

Extreme Programming planning releases and also different concepts are shown in the following points:

Planning Game

The planning starts at the beginning of each iteration where the stakeholders of the project meet to define, estimate and prioritize the “User Stories” (i.e. requirements) for the next release.

Small Releases

There are two types of version releases: initial version and working version. The initial version is produced after a few iterations and it does not implement all features, but only essential ones. While, a working version is produced after a few weeks and contains most of the features.

Metaphor

It is used in the modelling of the software system and is constructed by all the stakeholders.

Simple Design

It is the base of the XP methodology and applies to the requirements gathering, system design, coding, and communications with the customer.

Tests

In XP methodology, testing is considered one of the major activities to ensure high quality product in addition to high customer satisfaction. Testing begins before the coding phase, where developers are required to prepare the test functions prior writing the code itself. While, customers are required to prepare the functional test scenarios for each iteration.

Refactoring

It means that any changes made on the system must uphold the simplicity feature.

Pair Programming

Coding of the software system is carried out in a group of two developers.

Continuous Integration

New parts of the software system are integrated as soon as they pass both the unit and functional test cases.

Collective Ownership

The ownership of the code produced belongs to all developers.

On-Site Customer

Someone from the customer side must work with the development team at all times.

40-Hour Weeks

The maximum working hours per week for developers must not exceed 40-hours, which implies that the requirements must be revised too.

Open Workspace

All work, including coding and development must take place in a common environment.

XP and SCRUM Differences and Similarities

Differences

- SCRUM team works on a higher priority than extreme programming team in backlogs.
- Extreme programming main player is the product owner but in SCRUM is not.
- SCRUM is not engineering practice field but XP is an engineering practice.
- Extreme programming method takes less time than SCRUM method. Extreme Programming method takes two weeks at the most but SCRUM takes up to one month in order to finish the process.
- The maintenance and improvements for extreme programming are more easily than SCRUM method where the changing not allowed through the iteration process if sprint begins.

Similarities

Both SCRUM and Extreme Programming can work iteratively but with a different time for each one, and the work based on priority depends on the process that needs to be done.

In general, we will compare the agile methodologies and describe the agile limitations in the following subsections.

Agile Limitations

- The availability of the customer may not be possible.
- Methods rules must be given by experienced teams.
- Defining a proper iteration' functionality required.
- Communication skills required for the teams.
- Iterations must be designed properly which needs special design skills.

Comparison between agile Methodologies

Table 1 shows a comparison between Agile Methodologies while Table 2 lists the traditional techniques that are used in different agile methodologies.

Table 1. Comparison between Agile Methodologies

#	Agile Method	Requirements Representation	Customer Role
1	XP	<ul style="list-style-type: none"> • User Stories (written cards and conversations) • Requirements are not assumed to be complete or sharply stated. • Destroy user stories after implementation. 	On-site customer is required to participate in requirements
2	Scrum	<ul style="list-style-type: none"> • User stories are used. • The actual requirements are defined based on the discussion of user stories between software owner and software developers. 	Software owner plays the lead role in defining the
3	FDD	<ul style="list-style-type: none"> • UML diagrams are used. • List of features is used to manage the functional requirements. • Requirements are first represented in a high-level 	Undefined.
4	Lean Development	<ul style="list-style-type: none"> • Just-In-Time methodology is applied in requirements gathering. • User stories (cards) are also used by the customer to specify initial requirements and sample screens 	Provide input on initial user stories.
5	ASD	<ul style="list-style-type: none"> • Requirements' gathering is part of the speculation phase. 	Undefined.
6	Kanban	<ul style="list-style-type: none"> • User stories are used to define each sprint main goal. • Each sprint handles a single user story. • Each story is divided into the server-side and 	Undefined.
7	AUP	<ul style="list-style-type: none"> • Requirement phase consists of the following activities: <ul style="list-style-type: none"> - Identifying stakeholders - Understanding problem - Establishing the estimation basis. - Defining user interface. 	Undefined.
8	DSDM	<ul style="list-style-type: none"> • Requirements are gathered and prioritized during the feasibility phase. 	Undefined.

Table 2. Traditional technique exist in scrum and XP

	Scrum	XP	Crystal	ASD	Kanban	DSDM
Brainstorming	✓ 1	✓ 1	✓ 1			
Interview	✓ 1	✓ 1	✓ 1			
Use cases	✓ 1	✓ 1				
Modeling	✓ 1	✓ 1	✓ 1		✓ 1	
Observation	✓ 1	✓ 1	✓ 1	✓ 1		✓ 1
Meeting		✓ 1	✓ 1			
Analysis meetings	✓ 1				✓ 1	
Scenarios	✓ 1	✓ 1				
Prototypes	✓ 1		✓ 1	✓ 1		
Social analysis	✓ 1	✓ 1	✓ 1	✓ 1		✓ 1
Documentation	✓ 1					
Reviews	✓ 1			✓ 1		

HyArchive Methodology

We propose a hybrid Agile technique that merges the two most known agile methodologies: extreme programming (XP) and SCRUM. The main concept is to integrate agile methods XP, SCRUM with the software development life cycle (SDLC) methods to build an efficient archiving technique. Hy Archive is a technique for managing an archiving system' development process. It is designed for teams of developers who: break their work into actions like SCRUM that can be completed within fixed duration cycles (called "sprints"), and collaborate to deliver workable Archiving software every sprint. Hy Archive is an iterative and incremental hybrid agile software development framework merging between SCRUM and XP for managing the development process of an archiving system.

HyArchive Main Players

There are four core players in the HyArchive methodology:

- Product Owner
- HyArchive Development Team
- HyArchive Analyst
- HyArchive Master

These are ideally co-located to deliver potentially shippable an archiving system increments every sprint.

Product Owner: The product owner represents the archiving system customer or stakeholders and is responsible for ensuring that the development team delivers value to the business. The

product owner defines the product in customer-centric terms (typically user stories), adding them to the archiving system backlog, and prioritizes them based on importance and dependencies. Teams should have one product owner. This role should not be combined with that of the HyArchive master. The HyArchive product owner should focus on the business side of the archiving system development and spend the majority of their time liaising with stakeholders and should not dictate how the team reaches a technical solution.

HyArchive Development Team: The HyArchive development team is accountable for producing potentially shippable product increments every sprint (the sprint goal). The HyArchive development team execute all needed tasks to produce the product increments.

HyArchive Analyst: As we know one of agile weakness is the design needs that should be kept simple and maintainable. Thus requiring effective design skills. Therefore, the HyArchive analyst plays an important role and can't be one of the team members. The analyst main rule is designed to sketch and to design data structure and then write the algorithms in a step-by-step way with the development team. He also works with the HyArchive master to guarantee the efficiency of data structure, OOP, and testing concepts on our technique.

HyArchive Master: The HyArchive master facilitates the HyArchive technique. The HyArchive master is accountable for removing impediments to the ability of the HyArchive development team to deliver the archiving system goals. That's besides ensuring that the HyArchive methodology implementation is correctly followed. The core responsibilities of a HyArchive master include:

- Enabling the HyArchive development team to know the progress of the product, along with the input from key stakeholders.
- Arrange with the product owner to schedule a meeting with the HyArchive development team to address if what planned is the best for the project or not.
- Deliver the archiving system with good-quality features.
- Guarantee the best progress.
- Helping the stakeholders to learn HyArchive principles.

HyArchive Algorithm

Figure 2 shows the flowchart of the HyArchive algorithm.

The HyArchive algorithm is listed here from the start until publishing the system.

- 1- Collect Requirements from customer (Rx).
- 2- Collect Requirements from S/W owner & S/W developers (Ry).
- 3- Compare (Rx) and (Ry) and get matched Requirements.
- 4- Store requirements into three ordered classes:

- (a):contains $(R_x) \cap (R_y)$.
- (b):contains $(R_x) \text{ --- } (R_y)$.
- (c):contains $(R_y) \text{ --- } (R_x)$.

- 5- Work on requirements of class (a).
- 6- Get customer feedback & store results in class (b).
- 7- Work on requirements of class (b).
- 8- Get user feedback & store result in class (b).
- 9- Work on requirements of class (c).
- 10- Get user feedback & store result in class (b).
- 11- Work on requirements of class (b).
- 12- Get user feedback & store results in class (b).
- 13- Review requirements in class (b) by S/W owner & developer.
- 14-Discuss R(b) by S/W owner & developer & customer.
- 15- Decide final set of requirements to be implemented.
- 16- Release the system to review team.
- 17- Release the system to testing team.
- 18- Publish the system.

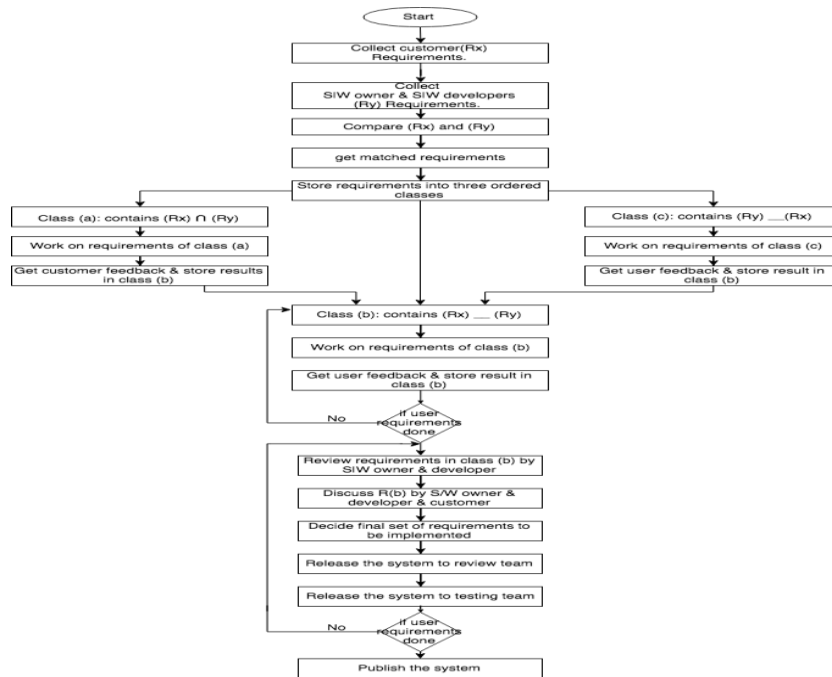


Figure 2.HyArchiveAlgorithm Flowchart

HyArchive Work Flow

Figure 3 shows the flowchart of the HyArchive methodology which contain the following main steps.

- A sprint is a core unit in HyArchive methodology.
- The sprint is a time-boxed effort. It is restricted to a specific duration.
- The duration is fixed in advance for each sprint.
- The sprint backlog is defined at the beginning of each sprint, after identifying each sprint work.
- Ensures the best for what planned through a meeting in each sprint between the HyArchive analyst and the product owner.
- Make a full review of the end of each sprint to the stakeholders to list improvements and lessons for the next sprints.

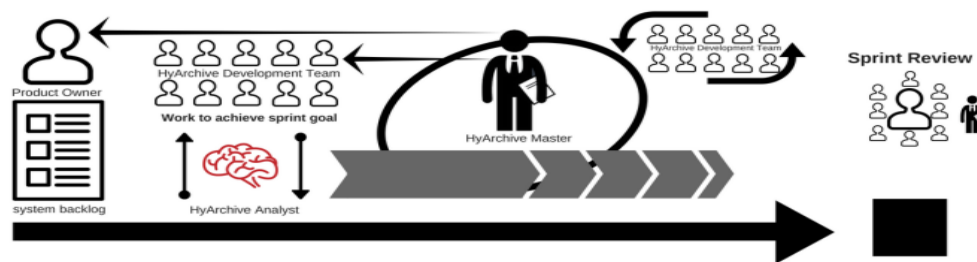


Figure 3. HyArchive Work Flow

4.3. HyArchive Sprint

The HyArchive sprint contains the following:

- Exploring Requirements.
- Development team meeting with the product owner.
- Algorithm design.
- Scheduling.
- Coding.
- Testing.

Case Studies

Three case studies are conducted, one for Extreme programming method, one for the Scrum method, and one for our new method HyArchive. All the studies were conducted in order to make an archiving system for three different departments. These methods applied on three departments that were suffering from the large amount of documents and the un-automated workflow in a governmental Organization.

Three teams were chosen to compare the effectiveness of the proposed approach with existing approaches. Each team is assigned to implement the Archiving system using one methodology. Three departments used to apply the implemented techniques. The first team took an assignment of implementing the archiving system in the HR department. The second team took an assignment of implementing the archiving system in the training department. The third team is assigned to implement the archiving system in the Administration and organization department. Table 3 shows the comparison between departments and illustrates the amount of documents in each department.

Table 3. Comparison between departments of case studies

#	Department name	number of document (per year)	Number of PCs required (per room)
1	HR	789100	4
2	Training	60925	2
3	Administration and organization	100524	4

Meetings were held with the personnel in charge and employees in three different departments. It was agreed to build an electronic archive system with the following features:

1. Provide spatial space, dispense the archived papers and piles of transactions and files.
2. Linking the parts of the institution no matter how far away between places.
3. Easy and fast transfer of messages and electronic documents between branches of the organization or outside the scope of the organization.
4. Easy access to electronic documents.
5. Regardless of the location of the beneficiary or user of these documents.
6. Monitoring the documents and their transformation.
7. Follow-up, development, and knowledge of the course of transactions within the system.
8. Multiple access points for documents electronically saved which facilitates the retrieval of documents.
9. Fast access to the beneficiaries and their services, minimizing errors and violating regulations.
10. Minimize reviews of government departments by beneficiaries.
11. Centralization in dealing with documents and thus not inconsistency and repetition in issuing orders or decisions.
12. Standardization in dealing with documents linking information in more than one database both inside and outside the parent organization.
13. Increase the performance of government institutions as well as private sector institutions.

HyArchieve Case Study

The main purpose of conducting the case study is to develop an archiving system following the new proposed method HyArchieve. The total time for this case study was eight weeks not

including the training time of the new proposed methodology HyArchieve. MySQL, MS Test, and crystal report, are tools used during the work. The HyArchieve team was consisting of eight members, 4 developers/designers, 3 testers, and one HyArchieve master that deals with the team and acts as the product owner too.

Scrum Case Study

The main purpose of conducting the case study is to develop an archiving system following the principles of Scrum method. The total time for this case study was eight weeks not including a training program that includes the principles of the Scrum methodology. The Scrum team was

consisting of eight members, 4 developers/designers, 3 testers, and one Scrum master that deals with the team and acts as the product owner too.

Extreme programming Case Study

The main purpose of conducting the case study is to develop an archiving system following the principles of the extreme programming method. The total time for this case study was eight weeks not including a training program that includes the principles of the extreme programming methodology. The main propose of the training was to ensures that the team understands the concept "keep it simple", and to work in pairs correctly. The XP team was consisting of eight members, four developers/designers, 3 testers, and XP master that deals with the team and acts as the product owner too.

Results

All the studies were conducted in order to build similar archiving systems for three different departments. All case studies were conducted in Kuwait country, and a team of eight members is selected to work on all the case studies in order to generate the results. More details provided in the below table. In comparing the three methodologies, the same features were used in [19].

Table 4 and Table 5 illustrate that HyArchieve was conducted by a team consists of eight people. They took 35 days to complete their mission. The system was released five times each one with an updated issue. The numbers of tasks of the system were 50 tasks come from 56 user story. It seems that 56 is a large number but in our system, it is preferred to gather a large number of tasks to increase accuracy. The overall work takes 800 working hours. The productivity comes equal to 50% and the defects are 10% based on the user's opinions collected from prepared survey. That leads to company satisfaction equal to 95.2%.

Table 4. Definition of the criteria of evaluation

	HyArchive	Scrum	XP
Information source	Archiving	Archiving	Archiving
Team Size	Number of Teams Members		
Time (days)	Day to complete their mission		
Releases	System releases		
Tasks defined	The number of tasks of the system		
number of user stories/tasks	Tasks comes from user story		
Total work hours	Total work hours		
Productivity	The productivity estimated by Master of Each Method by %		
Defects at post release	Based on the user's opinions collected from prepared survey		
Organization Satisfaction	$(\text{Work Hours} / \text{Times of days for project} * 24 \text{ hr}) * 100 = \%$		

Table 5. The results of evaluating the three conducted Case Studies

	HyArchive	Scrum	XP
Information source	Archiving	Archiving	Archiving
Team Size	8	8	8
Time (days)	35	42	45
Releases	5	5	5
Tasks defined	50	50	50
number of user stories-tasks	56	44	39
Total work hours	800	850	900
Productivity	50	45	40
Defects at post release	10	18	20
Organization Satisfaction	95.2%	84.3%	83.3%

As shown in the results, the performance of the projects is calculated by dividing the total effort of the entire project over total work hours. We can say that using the hybrid method HyArchive is much better for customers than using XP or scrum. This is because by comparing the value 51 % (for HyArchive method) with the value of scrum 41 % or XP 44 %, it is correct to use the hybrid method that gives a value much close to the value resulted from each individual method. The performance of the project is calculated by the following equation:

$$\text{Project performance} = \frac{\text{total effort of the entire project}}{\text{total work hours}} \quad (1)$$

Table 6. Comparison between the existence of traditional techniques methods HyArchieve, Scrum, and XP methods

Criteria	HyArchieve	Scrum	XP
Software requirements specification	√	√	
Interview	√	√	√
Brainstorming	√	√	√
Use cases	√		√
Scenarios	√	√	√
Meeting	√		√
Prototypes	√	√	
Priorization	√	√	√
Modeling	√	√	√
Analysis Meetings	√		
Observation	√	√	√
Social analysis	√	√	√
Documentation	√	√	
Reviews	√	√	

Table 7 and Table 8 show that HyArchive was conducted by a set of requirements equal to 16. At least 2 users participate in any requirement. The effort is the amount of work unit required to complete any given task. The effort done for each requirement was 42.77 unit of work, the effort done for each user story is 27.3 unit of work and this number is the mean of the values that represent the unit of works in each iteration. Total effort is calculated as the result of adding effort per requirement and the other efforts.

Table 7. The Data description of the three conducted Case Studies

Metrics	HyArchieve	Scrum	XP
# of requirements	Number of requirements		
(# of user stories/tasks)	The number of tasks of the system		
user stories per requirements	User stories per requirements		
effort per requirement	Effort is the amount of work unit required to complete any given task		
(effort per user story/task)	The effort done for each user story		
Total effort all user stories/tasks	Total effort is calculated		
Other effort	Other effort come from another source		
(total effort entire project)	Total effort is calculated as the result of adding effort per requirement and the other efforts		

Table 8. Results of evaluating the three conducted Case Studies

Metrics	HyArchieve	Scrum	XP
# of requirements	16	16	16
(# of user stories-tasks)	56	44	39
user stories per requirements	2	3	5
effort per requirement	42.77	36.77	23.04
(effort per user story/task)	27.3	17.5	6.51
Total effort all user stories/tasks	390	344.6	300.5
Other effort	11	20	102
(total effort entire project)	401	364.6	402.5

Table 9. Comparison between the three used Methodologies

#	Agile Method	Requirements Representation	Customer Role
1	XP	User Stories (written cards and conversations) Requirements are not assumed to be complete or sharply stated. Destroy user stories after implementation.	On-site customer is required to participate in requirements definition, estimation and prioritize.
X	Scrum	User stories are used. The actual requirements are defined based on the discussion of user stories between software owner and software developers.	Software owner plays the lead role in defining the requirements
3	HyArchieve	User Stories (written cards and conversations) Requirements are assumed to be complete or sharply stated. Keep user stories after implementation. The actual requirements are defined based on the discussion of user stories between software owner and software developers	Customer, Software owner and software developer plays the lead role in defining the requirements.

CONCLUSION

We have proposed a new Agile methodology for building archiving system that tries to fill gaps in agile and guarantee the high-quality archiving system. It is called HyArchive: Hybrid Agile Archiving Technique. HyArchive merges between most known agile methodologies named extreme programming (XP) and SCRUM. The technique is implemented to reduce the time and effort consumed during analyzing and designing the archiving systems. The proposed methodology reduce the system errors happen at the early stages of the development of the system. HyArchive helps the organizations in designing their archiving system that satisfies their needs in minimum time and effort. Hyarchive was implemented and compared to other

methodologies Scrum and XP and show a superior performance. It depends on values of courage, correspondence, criticism, simplicity, adaptability, flexibility, and efficiency. As shown in the results, using the hybrid method HyArchive is much better for customers than using XP or scrum as it gives 51 % performance of the project using HyArchive method) with the performance of the project while scrum gives 41 % performance and XP gives 44 %. The Hyarchive show a beneficial use in requirement engineering process to build archiving systems for governmental institutions.

REFERENCES

- [1] Nuseibeh, B.; Easterbrook, S. Requirements engineering: a roadmap. ICSE'00. Proceedings of the conference on the future of Software engineering. pp. 35–46, 2000.
- [2] Kotonya, Gerald; Sommerville, Ian (September 1998). Requirements Engineering: Processes and Techniques. John Wiley & Sons, 1998.
- [3] Chemuturi, M. (2013). Requirements Engineering and Management for Software Development Projects, 2013.
- [4] Sriram, Rand Mathew, S.K Global Software Development Using Agile Methodologies: A Review of Literature. 2012 IEEE International Conference on Management of Innovation and Technology, Bali, 2012.
- [5] Medhat, W, Fouad, KM, Yousef, AH, and Moawad, IF, published in 12th international conference of Computer Engineering and Systems (ICCES), 2017
- [6] Manjul Gupta, Joey F. George, and Weidong Xia, “Relationships between IT department culture and agile software development practices: An empirical investigation”, “International Journal of Information Management 44 (2019) 13–24”, 2019.
- [7] Pacheco, C., and Garcia, I. A systematic literature review of stakeholder identification methods in requirements elicitation. Journal of Systems and Software, 85(9), 2171-2181, 2012.
- [8] Amani Mahdi Mohammed, Hisham Mohamed Abushama, “Popular Agile Approaches in Software Development: Review and Analysis”, Researchgate Conference Paper • August 2013.
- [9] Shen, H., Wall, B., Zaremba, M., Chen, Y., & Browne, J. Integration of business modelling methods for enterprise information system analysis and user requirements gathering. Computers in Industry, 54(3), 307-323, 2014.
- [10] Mohammad Almseidin, Khaled Alrfou², Nidal Alnidami³ and Ahmed Tarawneh, “A Comparative Study of Agile Methods: XP versus SCRUM ”, International Journal of Computer Science and Software Engineering (IJCSSE), Volume 4, Issue 5, May 2015
- [11] Khaleel, Y., Abuhamdah, A., Sara, M. A., & Al-Tamimi, B. Components and Analysis Method of Enterprise Resource Planning (ERP) Requirements in Small and Medium Enterprises (SMEs). International Journal of Electrical and Computer Engineering, 6(2), 682, 2016.
- [12] Farrukh Musa, and Muhammad Ali Tariq, “Agile Methodology: Hybrid Approach Scrum and XP ”, International Journal of Scientific & Engineering Research, Volume 8, Issue 4, April-2017.
- [13] Ankita Sharma, and Manav Bali, “Comparative Study on Software Development Methods: Agile vs Scrum ”, International Journal of Emerging Research in Management & Technology, June 2017.
- [14] Apoorva Singh and Dharendra Pandey, “Implementation of Requirement Engineering in Extreme Programming and SCRUM ”, International Journal of Advanced Research in Computer Science Volume 8, No. 5, May – June 2017.

- [15] Sara Ashraf, Shabib Aftab, “Scrum with the Spices of Agile Family: A Systematic Mapping”, I.J. Modern Education and Computer Science, 2017.
- [16] Julio Cesar Pereira, and Rosaria de F. S. M. Russo, “Design Thinking Integrated in Agile Software Development: A Systematic Literature Review”, “international Conference on Project Management / HCist– International Conference on Health and Social Care Information Systems and Technologies”, 2018.
- [17] Sultania, A. K. (2015, February). Developing software product and test automation software using Agile methodology. In Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT) (pp. 1-4). IEEE.
- [18] Kniberg, H. (2015). Scrum and XP from the Trenches. Lulu. com.
- [19] Kevin Thompson, Ph.D. “How to Estimate Capacity for Work in Agile Teams”, 2012.