IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 July 2022, accepted 30 August 2022, date of publication 1 September 2022, date of current version 12 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3203582

RESEARCH ARTICLE

RAPD: Rapid and Participatory Application Development of Usable Systems During COVID19 Crisis

YOUSEF-AWWAD DARAGHMI[1] **AND EMAN-YASSER DARAGHMI**[2]
[1]Department of Computer Systems Engineering, Palestine Technical University—Kadoorie, Tulkarem P305, Palestine
[2]Department of Applied Computing, Palestine Technical University—Kadoorie, Tulkarem P305, Palestine

Corresponding author: Yousef-Awwad Daraghmi (y.awwad@ptuk.edu.ps)

This work was supported by Palestine Technical University—Kadoorie.

ABSTRACT Software development methods have been evolved to enable producing usable systems rapidly while considering all requirements. Several studies have focused on the need to balance between rapid development and capturing requirements related to user experience and business workflow. This balance has become more urging during COVID19 because many businesses want to quickly transfer to usable electronic systems that are accurate, efficient, easy to learn, satisfy users and support remote work. Therefore, this paper proposes a framework by integrating Rapid Application Development (RAD) method with Participatory Design (PD) method for enabling rapid production of usable systems. Both RAD and PD consist of design stages that can overlap and generate new phases where users participate in the design process and accelerate the production. Five usability tests are also added to the framework to validate the usability of the design at all stages. The Action Research method is used to assess the framework empirically in a context of an urgent need to an electronic system, and qualitative data analyses were conducted. The results show that the framework can be adopted by software companies because it satisfies the requirements of adopting software development methods. Also, the system developed using the framework is usable. The paper concludes that COVID19 affects software development by emphasizing rapid development while maintaining workflow. Also, using video conference for remote design assists in meeting users more frequently and in creating concise requirement documentation.

INDEX TERMS COVID19, participatory design, rapid application development, software development methods, usability testing.

I. INTRODUCTION

COVID19 has affected all life aspects while lockdown policies and restrictions on movement caused a sharp jump in the use of digital technologies in different domains [1]. COVID19 has made the need to remote work clearer than ever. In fact, the lockdown left many businesses floundering due to the non-existence of systems supporting remote work. This has caused much financial loss to companies and put them in a critical situation trading off between health and business. In many countries, businesses depend on either traditional paper-based systems or on electronic on-site systems, and during COVID19, they have started to look for solutions to cope with urgent emerging work conditions. This motivates software development companies, who also suffer from work conditions imposed by COVID19, for producing solutions that enables work under pandemics. To enable rapid production, these companies shifted their development methods from slow methods, such as waterfall, to faster methods such as agile. Unfortunately, the fast development of new electronic systems worsened the situation because they do not fit with workflow and require complicated professional skills and security measure adding new burdens to businesses.

The new work environment caused by COVID19 has made the need for rapid development methods that can captures all requirements stronger than before. Traditional methods,

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

e.g. waterfall, collects requirements without involving users in the design, while Participatory Design (PD) overcomes old methods shortcomings by putting users in the center of the design process [2], [3], [4], [5]. However, both usually take long development time, [6], [7], [8]. The other methods enabling fast development usually do not capture all requirement details, e.g. Rapid Application Development method (RAD) [6]. The agile method, which is the most common now, was invented to enable the interaction with users for better requirements collection while building software rapidly [9], [10]. However, agile based methods are still criticized for not being able to understand the user point-of-view and the context of use [10], [11], and for not focusing on documentations which can be necessary for further development [6]. For overcoming these drawbacks, end-users should be involved in all development stages through user centered agile software development approach [10]. But, there is a lack of empirical studies as evidence of the success of this approach [10]. Moreover, the shift to work from home during COVID19 has reduced the possibilities to interact with users and analyze their work context effectively.

Therefore, this paper proposes an integration of software methods for enabling rapid development of software and capturing all user and work requirements, particularly during COVID19. The proposed integration utilizes RAD for the fast development and PD for user collaboration in the design process and capturing all details in the work context. We found that these two methods contain development stages that can overlap for strengthening their advantages and reducing their drawbacks. Also, we added usability testing for enabling verification and validation of system design along all stages. We refer to the integrated framework as RAPD.

To find empirical results, we used the action research method because it assists in examining a specific problem in a specific context and proposing solutions to that problem [12]. So, action research assisted us in examining RAPD practically by applying it to a context consisting of a software company and Jericho Central Vegetable Market which needed electronic systems urgently. The market includes different auctioning shops who receive farmers' products and sell them to traders who distribute the products to other cities. We also tested RAPD during mobility constraints by designing through video conference. We also interviewed the company manager, development team and end users to collect data related to their perspectives about RAPD, the development process and the final product. Qualitative and quantitative analyses were also conducted to assess RAPD in terms of factor influencing its adoption in the company and its usability in the workplace.

The results of the action research show that the integration of RAD and PD was successful from the company perspective and user perspective. The company stated that RAPD allows for rapid development with the required documentations and efficient team management. More importantly, RAPD organized the interaction between the development team and end-users in a way that is cooperative, creates concise user stories, eliminates conflicts and reduces post-development modifications. We found that RAPD satisfies the factors stated in [13] and [14] that enable the adoption in the company. The users stated that they accepted the system developed using RPAD because it supports remote work, and it is a usable system satisfying all factors affecting usability stated in [15] and [16]. Both, the company and the users stated that participating in the design even through video conference was time-saving and effective design process. They were able to make design meeting at homes more frequent, review and refine the design, and focus on the important elements without including unnecessary conversations. This research also found that COVID19 has affected the software development process and created a need for methods that focus on the new work style while urgently responding to the crisis.

The contributions of the paper can be summarized as:

- A Rapid Application Participatory Development (RAPD) framework which integrates RAD, PD and usability testing methods for the development of usable software that addresses user and work requirements,
- Application and Evaluation of RAPD through action research in an important domain,
- Qualitative analyses identifying the factors effecting the usability of new software methods used for software development during crisis from the company and end-user perspectives,
- Post-crisis quantitative analyses identifying the usability of RAPD as a development method and the usability of the system developed by RAPD.

The rest of the paper is organized as follows: Section II presents the theoretical background, Section III explains the methodology, Section IV presents and discusses the results, Section V presents the limitations of the research, Section VI concludes the paper.

II. THEORETICAL BACKGROUND

COVID19 has imposed new challenges to the software development by creating a critical situation for businesses due to new work environment. This situation has forced businesses to search for electronic systems that can be employed quickly and support remote work while maintaining workflow. This situation has also forced software development companies to search for methods that balance between rapid development and addressing all requirements. Therefore, the design framework proposed in this paper integrates three fundamental methods that are Rapid Application Development (RAD), Participatory Design (PD) and usability testing. These methods were successfully used in [17], but the integration details and influence of these methods on the development team and end users were not discussed.

A. RAPID APPLICATION DEVELOPMENT (RAD)

RAD is one of the agile software development framework and was found to allow developers to build high quality applications rapidly and iteratively [7]. In contrast to older methods, such as waterfall models, RAD involves iterative process during all software development four stages: requirements planning, user design, construction, and cutover [18]. Iterative development means creating increasingly functional versions of a system in short development cycles. Each version is revised with the end user to produce requirements that feed the next version. The process is repeated until all functionalities have been developed. In fact, RAD process consists of four stages [18], [19]:

1- Requirement planning stage: The designers meet with clients and form a team to analyze requirements, identify all entities, draw action diagrams, and define all interactions between functions and data.
2- Joint application development (JAD): the team revises the requirements for determining the core ones, develops the entities collected in the requirements planning into a data model and diagrams, develops test plans, and creates layouts and design for the system based on object oriented programming.
3- Construction stage: the team iteratively develops and tests the system, refines the requirements until the system is complete. The developers convert the data model into a functional prototype which is tested by the construction team using test scripts developed in JAD. During this stage, the designers meet also with users to refine the design.
4- Implementation stage: the system in deployed and the end users are trained on using the system.

Recently, RAD has been used for developing different applications that are based on mobile and cloud computing (e.g. [20], [21], [22]). In [20], RAD was used for developing a tool for electronic design automation. In [21] and [22], RAD was used for developing applications for prayer education and Islamic calendar. Extended versions of RAD were introduced for developing safety applications [23], [24]. An application to track suicide risks was developed in [24].

Although users are involved in the four stages, RAD allow users to only participate in the technical issues of the design and consequently non-functional requirements are not captured effectively [6], [7], [8]. While, user needs to explore other social, organizational and job issues to allow capturing all workflow details. Therefore, some attempts were made to make RAD focus more on user involvement for better requirements elicitation [23], [24], but empirical evidence is still needed, particularly from project management point of view. Further, RAD does not fully advise on how to build a project plan, set up a team, manage user-developer relations and document each design step [7], [8]. RAD, as other agile methods, is also criticized due to the lack of documentations, and this would make tracing changes during development or adding new components after development more difficult [6].

B. PARTICIPATORY DESIGN (PD)

PD allows for user participation in designing computer systems by creating intimate social atmosphere between end-users and developers [25], [26]. So, PD is a user centered design method in which users express their requirements for building software that is usable [4], user friendly and fits well with user culture, age, educational background and communities [27], [28], [29], [31]. Designers interacts with users through design meetings, interviews and participatory observation for collecting requirements and evaluating designs. The design process in PD passes through three main stages [26]:

1- Pre-design stage: designers and users decide the project plan, objectives and schedule, and they select representatives to perform the coming design tasks.
2- Requirement analysis and design stage: this stage is divided into three sub-stages:
    - Stage 2-A: designers and user representatives analyze the organizational workflow and feed the design with the output resulting from data collection and analysis. Documents are created to maintain the focus of all participants in the design process.
    - Stage 2-B: Then, developers build a prototype based on the earlier analysis, and each update is also documented.
    - Stage 2-C: After that, further technical issues are determined and the design can be revised to include new technologies.
3- Post-design stage: the prototype is implemented and tested against the project plan and objectives. The final specifications are also identified and documented.

PD has been used for developing different applications [27], [28], [29], [31]. In [27], PD was used for developing a system for clinical protocol writing. In [28] and [31], PD was used in the education domain for building an academic dashboard and mobile learning systems. In [29], PD was used in the entertainment domain to develop an interactive TV system for elderly people. In [30], PD was used for developing an information system in Ethiopia for rural communities.

However, PD is criticized because it neither provides a fully specified design process [32] nor structured and systematic assessment of the design concepts [33]. Therefore, PD requires integration with other practical development methods, such as RAD and agile, to achieve a sufficient influence on information system applications [5], [8]. PD also can be integrated with usability testing methods to enable assessing products at each stage effectively [33]. Also, PD puts emphasis on the early systems development phases which delays production of ready-to-use system and causes design sessions to consume much time [34].

C. USABILITY TESTING

The usability testing is borrowed from usability engineering and allows users to perform real tasks so that they can evaluate each developed block, and this helps developers produce a ready to use system earlier [35], [36]. In each development cycle, user problems, preferences, suggestions and work practices are applied, and a new design cycle will be issues if further design changes are needed [37]. Usability testing includes different types of tests which are performed during different design stages [38]:

1- The exploratory test is carried out early to test preliminary design concepts,
2- the prototype test is used during requirements gathering to provide iterative feedback into evolving design of prototypes or systems,
3- the validation test is conducted to ensure that completed software products are acceptable regarding predefined acceptance measures,
4- the comparison test is conducted at any stage to compare design alternatives or possible solutions.

The beauty of usability testing is that it can be integrated with different software development methods. For example, RAD is integrated with usability testing for developing websites as in [24], [39], and [40].

III. METHODOLOGY

In this research, we focus on solving the critical situation created by COVID19 for businesses by developing an electronic system that supports remote work and workflow in the business. As suggested by [13], when a software development method that is appropriate for a particular situation can not be found, compatible methods can be combined and used to develop a software. So, our solution includes integrating different software development methods so that the produced system can fit with the business requirements and can be accepted by users. But, would the integrated framework be adopted by the software development company, and would it be able to develop a usable system that satisfies users?

To answer these questions, we used the Action Research (AR) methodology. AR supports the seeking of improvements by solving real-life practical problem, and consequently it emphasizes the application of good practices and contributes to building new theories [12]. We used AR to investigate how RAD, PD and usability testing can be integrated and applied to real world practical problem. Therefore, AR enables the integration of RAD and PD by building a framework that shows how they work together and applying the integration to improve the context of digitization and remote work. Also, AR focuses on understanding how tasks are performed and why [41], so the researchers cooperates with a software development company and clients for analyzing tasks at the client side, and how these tasks can be digitized at the company side. This also enables testing the product through the entire development life, and the product is designed to be as usable as passible by adding usability testing at all design stages.

During the AR process, qualitative approaches including interviews and observations were used to determine if the new integrated framework is smoothly adopted by the company. So, we analyzed the factors affecting the adoption of the new framework focusing on factors stated in the literature. These factors include delivery time, cost of development, incorporation of changing requirements, size of team and communication with users [13]. Other classical factors also show that an innovative solution should be compatible with existing skills and practices, triable where results are experimented without extra effort and expenses, simple and used with low complexity, observable where results can be seen, and technically superior to its predecessors [14]. We also analyzed the factors affecting the acceptance of users to the systems developed by the integrated framework. We focus on the usability of websites as we aim to support remote work during crisis through web interface. Website usability include efficiency, learnability, memorability, accuracy and satisfactions [15]. We also focus on user satisfaction since we aim to make the system acceptable. User satisfaction can be measured by the website ease of use, content, delay, and customization [16].

The theory of AR states that the solution of a specific problem for the sake of any situation improvement requires understanding on both the real-world context and the methods used for achieving the improvements. So, this section presents the context in which AR is used, and the activities performed to understand the context and build a practical solution to the problem in the context. This section also explains the actions that should be made to integrate RAD, PD, and usability testing for producing a product that is usable and accepted by end users.

A. RESEARCH CONTEXT

The research context includes the company which is responsible for developing the software and the market which needs the software.

1) THE COMPANY

We refer to the name of the software company which participated in the research and in the development as ‘the company’ to ensure anonymity. The company is a software development company located in Palestine and started this business in 2003. The company produced several software products and participated in transforming the local commercial and financial business from paper-based systems to electronics systems. The company business size is small due to the limited software market in Palestine. In 2017, the company shifted its work from developing desktop applications to cloud services, and started focusing on software services.

The company previously depended on the waterfall software engineering method for developing most of its products. The company has started to adopt agile frameworks, mainly RAD, due to its simplicity compared to other agile frameworks. Also, RAD can be an answer to call for software solutions that support remote work due to the COVID19 pandemic. In the last three years, the company started to add PD to the development process, and PD was brought to the company by the manager who lived and worked in software development in other countries. The managers decided to use PD as the company found that RAD alone did not capture all requirements and the detailed workflow and therefore, the company had to perform extra RAD cycles to add new tasks to products and fix bugs.

To perform the development through the action research, the company allocated a team for managing the development process, and the team includes: a manager for signing contracts and participate in setting up plans, schedules and objectives; designers whose jobs focus on collecting requirements, designing prototypes, testing and validations; and developers who converts designs to working prototypes and products. All team members are familiar with RAD and usability testing, but PD was new to them. At the beginning of the research, the manager met with client side managers and agreed on work principles, trust, and mutual benefits. The team agreed on commitment to the design principles and ethical practices particularly because the PD involves collaborative interactions, verbal communications, sharing knowledge and maintaining client artifacts, such as user stories and other documents.

2) JERICHO CENTRAL VEGETABLE MARKET

Jericho Central Vegetable Market is the client who has a special context that needs practical improvement. Each county in Palestine has a vegetable market that is responsible for buying vegetables and fruits products from farmers and selling them to traders who distribute these products to other places. Jericho Central Vegetable Market which consists of several auctioning shops, and each shop contains employees mainly auctioneers, accountants and managers. The business model in this market depends on that each shop inside the market provides annual funding for farmers to enable them buy plants, water, electricity, fertilizer, irrigation pipes, plastic, and other materials. Farmers pay back funds after selling their products to traders through the auctioning shop who also gets a commission for each transaction. Also, the auctioning shops have contracts with supply stores to provide farmers with all needed materials. This market was selected because the auctioning shops urgently needed a commercial and financial software that supports remote work during the COVID19 pandemic.

B. INITIAL RESEARCH PROCESS

The research started by investigating the work environment by the company manager and designers. This investigation process lasted one week in which workflow observations and 12 semi-structured interviews were performed. They shared the recorded interviews, observations and artifacts with other designers and developers. The recorded materials were transcribed and analyzed.

The analysis of the targeted market showed that it has some distinguishing characteristics. The work culture where clients in the vegetable market have always emerging requirements due to the changing work conditions, they do not have sufficient computer skills, and they do not accept computer systems that bring new workflow. This was clear because at least four computer accounting systems were bought and only used for a short time. Evidence in literature shows that a large number of software in developing countries fail to give the desired results [42], [43], [44], [45]. The clients in the vegetable market justified that by the inability to deal with new workflow imposed by new software. They stated that farmers, traders, and other suppliers will be confused when shifting from the existing work style, which they are used to, to another style. These results agree with those in [46] about software that fits with the industrial countries may not suit the developing ones. While investigating this case, we found that clients, farmers, traders, and suppliers want the same format of bills, financial balance, and other documents. The clients stated that is more useful if the company let them participate in the design.

On the other hand, clients insist to see a prototype as fast as possible for two reasons: they want to ensure that their requirements are satisfied before paying any money for the software producers, they find learning a complete system very difficult so they prefer a step by step training learning fashion. The need for rapid development is also emphasized during crisis such as COVID19 which has imposed restrictions on the market as shop workers, farmers and traders can not meet face to face inside the shops to discuss transactions, billing, and payments. These processes usually depend on a long discussion between all parties and include paper printing of all documents. For example, farmers and traders always request paper bills containing all transactions of that day. So, all these processes have to be transformed to digital format during COVID19.

These findings were shared with the clients, and again the company and the clients agreed on these finding and decided to start the development of the software. The company explained that the framework, which will be used during the development, requires the client participation and involvement, and users agreed on that.

C. RAPD FRAMEWORK: METHODS INTEGRATION

The company team formulate the software development framework that will be used in the context based on the AR theory and on the initial process findings. The team names the framework 'Rapid Application Participatory Development' (RAPD) because it integrates both RAD and PD by overlapping the design stages so that advantages are strengthened and disadvantages are eliminated. Also, RAPD allows adding testing at any design stage.

1) RAPD STAGE 1

RAD requirements analysis stage is integrated with PD predesign stage and PD stage 2-A to form RAPD stage 1. In this stage, designers and users determines objectives, plans, and schedules, and they form a team of designers and user representatives to collect the requirements, identify all entities and draw all diagrams and interactions between functions and data. To avoid delay in this task, a team leader is selected to coordinate tasks among other team members who are divided into members for collecting requirements, members for putting the initial designs and diagrams, members for documenting all details, and others for reviewing related projects and using similar features for the new design.

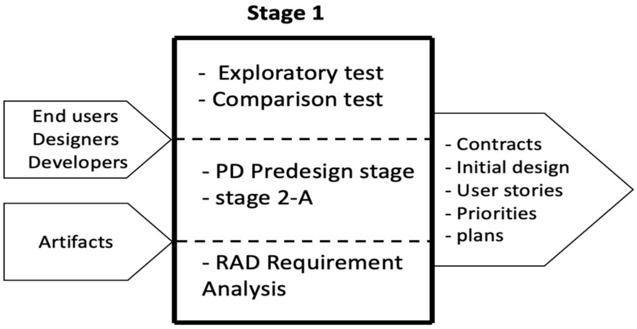

FIGURE 1. RAPD stage 1, its input, activities and output.

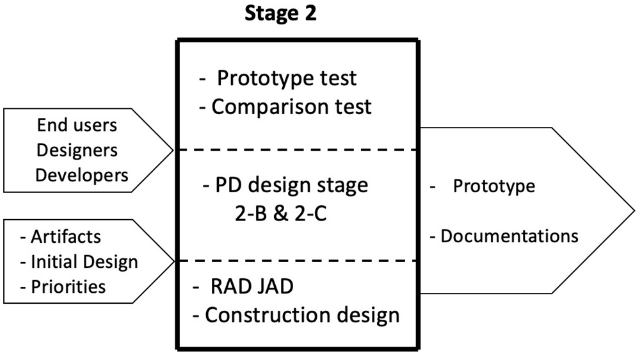

FIGURE 2. RAPD stage 2, its input, activities and output.

This stage is accompanied by exploratory testing for testing the collected requirements, observations and other initial design. This stage is shown in Fig. 1.

2) RAPD STAGE 2

PD stage 2-B and 2-C are integrated with RAD stage 2 (JAD) and RAD stage 3 (construction stage), as shown in Fig. 2. This reduces the development time if all documentation from stage 1 are adequately available, and if reusable components from related projects can be found and reused. In this stage, all requirements are refined, tests cases are developed, and a prototype containing layouts and data based on model, view, controller (MVC) is developed and tested. But if no reusable components are found, this stage is divided into sub-stage A which is for refining requirements, layouts, diagrams, and test cases; and sub-stage B which is for building a prototype, testing it and refining the design. All steps and changes are documented. The team here involves designers who interact with developers and representative users. All tasks are coordinated among members.

This stage also contains prototype testing to assess the functionality of the prototype and feedback the results for refining the requirements, modifying the prototype and updating the documentations.

3) RAPD STAGE 3

RAD implementation stage and PD post-design stage are integrated to implement the prototype and test the final system, as shown in Fig 3. All documents are reviewed to ensure all requirements have been considered. Because representative users are involved in the design and testing, no training is needed from development team for the client side.

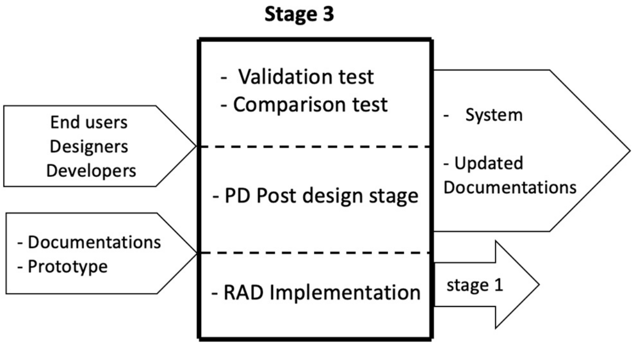

FIGURE 3. RAPD stage 3, its input, activities and output.

At the end of this stage, a validation testing is performed to evaluate the final system and ensure that all objectives and plans have been sufficiently addressed. Further, each stage also contains comparison testing for allowing participants to look for alternative designs and solutions.

At the end of stage 3, all major requirements are addressed in the design, and functions that are necessary to complete tasks are ready. So, end users can perform their work and send feedback to the design team. The team also revisits stage 1 to consider the requirements that were postponed and consider any emerging requirements.

D. APPLICATION OF RAPD

This section presents how the RAPD framework is applied to the Jericho Central Vegetable Market. The process started in April/2020 when the market made a call for a system that transforms the traditional work routine to electronic one and supports remote work. Four auctioning stores were selected, and two of them have electronic desktop system and the others are paper based stores. Three levels of users; managers, accountants, and data entry employees, cooperate to build the system. We refer to these users as direct users because they are responsible for the management of the system. Meanwhile, other users such as traders, farmers, workers, and supply stores are indirect users because they can view transactions and interact with the system after the direct users initiate transactions. In total, the direct users include nine auctioning shop owners and accountants, and indirect users include 120 farmers, 70 traders and nine agriculture supply stores.

COVID19 has also imposed restrictions on designers and developers as they can not meet users face to face. So, designers and developers shift to digital forms of communication such as video conferencing for most of meeting and workshops. This was not easy at the beginning due to lack of technical experience of end users. But, once a prototype is shown to users, they started interacting with designers remotely for testing and providing feedback.

The RAPD framework passed through the three stages:

RAPD stage 1:

Participants: direct users, designers, and developers

Prerequisites: artifacts: user stories, paper documents of all types of transactions, screenshots of existing electronic system (interfaces), and video of workflow

Activities:

1- Managers signed contracts to initiated the development process during a face to face kick-off meeting for three hours, and they agreed on the development team which involves users, developers and designers,

2- a video conference environment was selected to facilitate the communication, and all participants were informed that the meeting lasts for two hours,

3- designers and direct users met virtually to setup objectives, plans, schedules, and checklists,

4- designers analyzed artifacts and met virtually with direct users to collect the functional and non-functional requirements and identify priorities, and perform exploratory testing for refining the requirements and other observations,

5- designers and developers put initial design including all needed diagrams, entities, and initial interfaces, and then met virtually with direct users to perform comparison testing for the design so that all alternatives can be identified.

Outcomes:

1- contracts, objective document, work plan documents, schedule, and checklists,

2- concise user stories, specifications including functional and non-functional requirements documentations,

3- priorities of tasks based on dependency,

4- initial design (models and Documentations) including diagrams interfaces, database entities and communication entities.

The development team decided to give high priority to functions that are related to direct users. Other requirements related to indirect users were postponed. This task lasted for 21 days and include one physical meeting and three virtual meetings. The developers build the initial design and identified the reusable components at the end of this period. These reusable components can accelerate the development time. To allow remote work, the development team agreed on using web-based system supported by cloud services. Examples of services that are necessary to direct users are shown in Table 1 and prioritized based on dependency.

TABLE 1. Examples of direct user services prioritized based on dependency.

stage	Priority	Service
1	1	Data entry of accounts for customers, such as farmers, traders, suppliers, and employees
1	2	Settings to configure all needed parameters, such as tax, commission, box prices, employee's month salary, and others,
1	3	Data entry for daily transactions: inputting daily sales, daily payments, and daily receipts,
1	4	Daily auditing of all transactions,
1	5	Basic query
1	6	Printing service customized to end user formats (language, size and shape),
1	7	Billing and balancing accounts,
1	8	Basic query service including daily or accumulative customer balance, daily or accumulative bill, store daily and accumulative balance, commissions, taxes,

TABLE 2. Services developed in stage 2.

stage	Priority	Service
2	1	Direct users: service to create user authentication credentials
2	2	Direct/indirect users: Log in service with defined user privileges
2	3	Indirect users: Interacting with functions, such as viewing bills and balances, requesting changes, and modifying entries
2	3	Indirect users: billing and balancing queries
2	3	Direct users: advanced queries

RAPD stage 2

Participants: development team: direct users, indirect user representatives, designers and developers

Prerequisites:

1- Initial design, diagrams, entities, interfaces and reusable components,

2- artifacts: concise user stories, paper documents related to indirect users, screenshots of existing electronic system (interfaces), and video of workflow,

3- priorities of tasks.

Activities:

1- Designers analyzed artifacts and met virtually with direct and indirect users to collect the functional and non-functional requirements for the rest of the requirements, and updated the checklist,

2- designers and developers refined requirements, built test cases, developed and tested MVC prototype as in comparison test to check all alternatives, and then met virtually with direct and indirect users to discuss and modify the design,

3- the development team performed prototype testing and used the results to refine the prototype and update the documentations,

4- the development team setup and tested the cloud environment.

Outcomes:

1- prototype that runs all functions related to direct users and interfaces to those related to indirect users,

2- documentations for requirements, design, diagrams, test cases, cloud and results of testing.

This stage lasted for 14 days including three virtual meetings, and the development team was able to build a prototype for 90% of the entire system. The services related to indirect users were also added to the prototype, and the services developed in this stage are shown in Table 2.

Other low priority services, such as data analytics and forecasting, were postponed because these services require data. So, the development team waited until users inputted some data. This postponing also ensured that the end users are satisfied with the highest priority services before upgrading to more advanced ones.

RAPD stage 3

Participants: development team: direct users, indirect user representatives, designers and developers

Prerequisites:

1- prototype,

2- design documentations.

Activities:

1- The development team implemented and tested the prototype, and then met virtually with direct and indirect users to perform comparison testing and check all alternatives,

2- They reviewed the documentations to check whether all requirements have been considered or not and updated the checklist,

3- The development team met virtually with direct and indirect users for validation testing and for ensuring all objectives and plans were sufficiently considered.

Outcomes:

1- A system that runs on cloud,

2- Updated documentations that include all requirements designs, testing and results.

This stage lasted for 12 days including three virtual meetings. During this stage, the last 10% of the system was designed and implemented. With another short round by revisiting all stages, more advanced services related to data analytics and forecasting were added.

E. POST COVID-19 ANALYSIS

To ensure the validity of RAPD and its usage after the crisis is over, we conducted questionnaires focusing on both the software development company and the users. The questionnaire is justified at this point because the number of end users of the system developed by RAPD has increased. The questionnaire objectives are divided into two categories that focus on the development analyses and usability analyses. The development analyses targeted the persons who participated in the development process and other developers, and this questionnaire has four categories:

1- Effectiveness of RAPD during similar crisis conditions or normal conditions: This questionnaire targeted direct users, indirect user representatives and the development team from the company. Three questions tested if RAPD is effective during the crisis and even after the crisis, and if the team are convinced that RAPD achieves good results in other scenarios.

2- RAPD benefits over other methods: Ten questions tested if the development team think that RAPD is better than other development practices in terms of cost of development time and team size, delivery time, commitment to schedules, collecting user stories, interaction with users, avoiding conflict with users, clear plans, writing documentations, and addressing requirements.

3- Willingness to learn RAPD development process: This questionnaire targeted company developers. Five questions tested the reputation of RAPD among developers' peers, their skills in PD and RAD, and their motivations to learn RAPD even if they do not know PD and RAD.

4- Willingness to participate in remote design based on RAPD: four questions tested if the team is confident that they master RAPD and ready to practice it for developing software for other markets. This is important because most companies in Palestine develop software for other international companies. This also tested the willingness of developers to cooperate with users face to face or by video conference after the end of COVID-19.

On the other hand, the usability analyses targeted direct and indirect users, particularly users who did not participate in the development process. This tested if the system developed using RAPD during COVID-19 is still usable after COVID-19. The questionnaire focused on:

1- testing the willingness of user to work remotely.

2- testing the usability factors stated in [15]

3- testing user satisfaction factors stated in [16].

The questionnaires were written in Arabic, and each statement in the questionnaires has five scales: Strongly Agree (SA), Agree (A), Neutral (N), Disagree (D) and Strongly Disagree (SD). There are also questions with open answer where respondents can write short paragraph about the most and least favorable features of RAPD. A total of 23 respondents; nine direct users and 14 developers from different companies participated in the development analysis questionnaire. Also, a total of 51 respondents, 14 direct users and 37 indirect users including farmers, traders, and supply store workers participated in the usability analysis questionnaire. We ensured that all end user respondents knew about RAPD and experienced the system developed by RAPD. For example, each user has at least three months of experience in using the system.

IV. RESULTS AND DISCUSSION

This section presents and discusses the effect of RAPD on the company and end users. The section discusses the factors affecting the adoption of RAPD in the company. Then, it discusses the factors affecting the usability of system developed using RAPD. Finally, the section discusses the quantitative analysis results for the use of RAPD in normal conditions.

A. RAPD COMPANY PERSPECTIVES

From the analyses of the qualitative data obtained from the interviews and observations, we identified the factors affecting the adoption of RAPD as a new software development framework. We mapped these factors to the factors identified in [13] and [14].

1) RAPD PROJECT TIME

In the company context, the project time refers to the time from the start of the project to the moment users become familiar with the developed system and stop demanding new modifications. This time includes the development time, debugging time, modifying by including new requirements, and training time. In [13], the project time is referred to as delivery time.

At the beginning of the project, developers and designers resisted PD and thought that PD will increase the project time. They justified their thought as they are not expert in PD, and user involvement in the design to a degree more than that of RAD would cost much time working with users. The thoughts about PD were seriously considered by the company, and a project manager who has experience in PD was assigned to the project. The project manager put the project time in the first priority. To ensure effective time management, the PD principles were explained to the developers, designers, and clients. The project manager facilitates the use of PD and made the team feel comfortable with the whole process. In fact, because RAPD contains RAD which is used in the company for other projects, the manager found RAPD **compatible** to a moderate degree with the developers' skills and company practices. RAPD compatibly accelerated the acceptance of the development team to this new framework.

In total, RAPD did not increase the project time. The company stated that, in previous systems developed by RAD only, they used to add more RAD cycles for bug fixing, follow up review, and modifying the system according to user emerging requirements. In RAD, the developers focused on developing what they got from end users iteratively, and in many cases the development time was overdue, this agrees with [13]. But, when PD is introduced to the development life cycle, users were involved in the design and felt as they were in brainstorming sessions and obligated to explain all work details. So, the integration of RAD, PD and usability testing reduced the **delivery time** in the company. This made the company accept RAPD and consider it for future projects, which agrees with [13].

2) TEAM MANAGEMENT

The company used to follow RAD as an agile framework more than the other agile frameworks. In RAD, the company used to have five team members, and in RAPD the number became six because a team member was added for writing the documentation. Because most design meetings were online, one member was enough for the documentations. The video conference with voice to text conversion made collecting user stories easier than hand writing.

The members stated in the interviews that they were previously acting according to semi-management plans at which schedules and overall objectives were not clear. When PD is integrated with RAD, the project manager ensured clear management strategy, plans, schedules and objectives. The project manager initiated meetings, supervised design and documentations, and served as mediator between the company and the clients. According to the management strategy, tasks were distributed among members based on their experience. Each member has to participate in the design meeting, continuously report the progress of the development and commit to the time schedule. At the end of the project, the team members stated that they experienced less pressure during the RAPD processes, and they were fully aware of the purpose of tasks under-development and the final goals. This clarity of objectives, tasks, and final system made the **development cost** less than before in terms of team size and project time. This made the company adopt RAPD easily, which agrees with [13].

3) TEAM TO USER INTERACTION

The authors in [13] show that the **communication with users** is one of the factors affecting adoption of software development methods. So, the interaction between the development team and the users was also studies through observation during the development and interviewing participants. The analysis shows that the interaction was limited at the first meeting but after that the interaction became smoother. The project manager revealed important issues about the interactions with users. In older development project, developers used to resist new changes demanded by users. In some cases, conflicts occurred between developers and users and between developers and designers because developers do not want to keep changing the product. The conflicts resulted from erroneous assumptions made by designers who depended on verbose user stories that were interpreted differently from time to time. Also, users expressed 'what' functions they need without caring about 'how' these functions should be made, and designers had to extract 'why' these functions were needed and then build a design to show how functions could be in the digital form.

In contrast, the RAPD made the interaction more cooperative, direct and clearer than before because developers were aware of the project objectives, and they also aware that user role in the design would lead to an optimized product. During the workshops and meeting, the researcher observed that the company team members were consistently working on performing tasks and harmonically interacting with each other and with the end-users. According to the development team, the roles of each team member was clearer in RAPD than before. Even the end users were participating in the design according to PD principles and were aware of the final goals. They also benefited from RAD as they could try the prototype, give feedback and suggest alternative solutions. Therefore, the end users started to care about 'how' the function should be made and to suggest ways for doing that. Also, the end users started to appreciate the effort spent on developing every single function, and they could estimate the required time for transforming one task from paper to digital. So, the end users became more considerate about requesting changes or modifications.

The video conference also made the interaction more cooperative and useful because all participants knew that the time for the virtual meeting is limited and they avoided unnecessary talks. There were nine virtual meetings with two hours per each. In the vegetable market culture, auction shop owners and customers usually use much conversation about different topics while making a specific deal which costs much time.

In RAPD, the conversations were minimized during the video conference. The project manager had more control on the virtual meeting than the face to face meeting. The manager acted as a moderator of the meetings and mediated the discussion between the company side and the client side, and pushed to avoid unnecessary talks. So, PD is performed via the video conference to create artifacts, and the rapid nature of RAD assisted in minimizing unnecessary communications which create verbose stories. PD made user stories clearer than what they used to be before because PD allows the user to focus on the design not only describing the requirements. Thus, the video conference enabled creating concise documentations and concise participation and consequently clearer design with less conflict. The video conference is found useful for creating participatory design during crisis, and it can help big companies to organize participatory design session with customers in different places

To show the advantage of using the video conference in the design meetings, we give the following example. Printing farmer balance is a function that was developed using different software methodologies and users experienced it on different systems. In previous systems, the user story was recorded as 27 minutes video for this function. This video was converted to a document of four pages. But with RAPD video conference, the user story for that function is nine minutes and converted to a concise document of two paragraphs. Both the end users and the company were happy about the interaction during the project development which made RAPD to be easily adopted by the company.

4) RAPD OBSERVABILITY AND TRIABILITY

RAPD enabled the rapid development of the system and users could see the prototype in early stages. Also, the usability testing included in all RAPD stages reduced the number of bugs and modifications in the final system. These testing accelerated the production of usable services that have higher priorities. So, RAPD observability was high as its results were clear to users and developers. Further, RAPD compatibility and the existence of a member with high skills of PD and RAD made the developers aware how to use the framework. Also, user participation and the documentations facilitated the process of trying RAPD. On the client side, users could try the system services easily because they already had their hands dirty on testing the prototypes and the system. So, the RPAD triability was also high, which made RAPD attractive to the company and easily adopted. This agrees with results in [14].

5) RAPD INCORPORATION OF EMERGING REQUIREMENS

This factor is stated in [13] as an important factor for considering a software development method in any company. Most modern development methods, such as Agile and RAD, focus on user involvement in the design process to enable capturing all requirements not only at the start of the project but also during all design stages. So, RAPD by including both RAD, PD and usability testing makes user involvement clearer and more effective because users become a major player in the development team. The users were observed very active in expressing their requirements and in adding new changes to the system after testing developed services. The RAPD documentations consisting of concise user stories allowed the developers to refer to these changes and address them one by one. These features of RAPD made it acceptable by the company.

6) RAPD COMPLEXITY

The developers at the company initially thought that changing the method, which they are used to, will complicate the development process. Their lack of experience in PD strengthened their argument. However, their behavior towards RAPD started to change when the PD principles were explained to them and when they saw that PD and RAD stages can overlap. Their knowledge of RAD and the usability testing made them easily understand RAPD. Also, the COVID19 conditions motivated the company and the developers to think more seriously about finding a way to incorporate all requirements effectively. The video conference also made the developers less reluctant to use RAPD because they did not need to meet users face to face, which reduced travel burdens and infection possibilities. By the time, RAPD complexity was reduced, and the company accepted it which agrees with [14]. But, this required the existence of a member who has strong experience in PD and RAD.

B. RAPD USER PERSPECTIVES

The developed system during the action research through the RAPD framework is a web-based system running on cloud and support several services. These services include data entry, data review, billing, balancing, queries, and creating accounts for customers and users. The interaction of users with the new system was observed during the design and performing tasks from the workplace and remotely. Also, direct and indirect users were interviewed at the end of the research. The analysis focused on the usability of the system and user satisfaction. The analyses of the observations and interviews are mapped to the usability factors stated in [15] and the user satisfaction factors stated in [16].

1) REMOTE WORK

This factor is a new one and emerged due to COVID19 to maintain safety. The system utilizes cloud computing technologies for hosting the data and process the functions efficiently. The users interact with the system through a responsive web interface which supports desktop computers and mobile devices. Therefore, the users are able to work ubiquitously; from the office, home or other places at any time. To enable secure access to the system, the users need only to login to the system and start any process based on the privileges given to them.

The users stated that the remote work was very useful to them during the COVID19. When the mobility constraints were very tight, the direct users, such as the shop owners and accountants, were able to perform the basic data entry at office, which is at the moment of auctioning, and then could do other data processing remotely. On the other hand, indirect users, such as traders and farmers, were able to view their accounts, bills, and balance online without the need to visit the auctioning shops and demand paper bills. This reduced the face to face negotiations between all users and minimized the possibility of infection.

An important point was expressed by all users is the satisfaction from the remote work. Once users got used to the remote work functionalities, they started to depend on that because the remote work enabled reviewing the billing before adding bills to the total financial balance. In the traditional system, bills had to be written manually and added to the balances and customers had to wait before taking their bills. The RAPD framework allowed the users to participate in the design remotely and build experience of the system step by step. So, the adaptation to remote work was easier than they thought at the beginning.

2) TIME SAVING AND EFFICIENCY

In [15], efficiency affects system usability. The developed system saved the time for users on different levels. The time for the daily work routine was sharply reduced for users who depended on the paper work. They used to spend eight hours on average for writing bills and reports manually. They had to check different documents and made several calculations when a customer made a query about a specific financial issue. For the other types of users, who had electronic systems, the time was also reduced because RAPD enabled the users to modify the systems to be more customized to their work and more efficient.

We also found that the remote work saves users time. In the vegetable market, the remote work saved time because customers became able to view their accounts, bills, and financial balance without visiting the market which saves their time, especially, some farmers and traders come from far places. Customers could also review their billing remotely and demand for modification before adding bills to the financial balance. This reduced the revision time and negotiation time by enabling fixing calculations errors easier than before.

3) LEARNABILITY AND MEMORABILITY

Learnability and memorability are important for evaluating user satisfaction [15]. When the qualitative data were analyzed, we found that users did not separate between learnability and memorability. They stated that "what I learn I can remember". RAPD enabled the system to be developed rapidly and to be exactly customized to the daily workflow in the market by considering the sequence of operations, the output formats and all query services. Also, end users stated that the customization through RAPD allows them to learn how to use the system and memorize that faster and easier than other systems developed by other methods. Further, users including direct and indirect users participated in designing and testing the system. So, the degree of learnability and memorability were higher than other systems purchased by the market before.

4) ACCURACY

Accuracy is very critical for the auctioning shop because any mistake in the calculation causes conflict between the direct users and the indirect users. Also, mistake causes time delay to determine the reasons of the mistake and solve it. The users stated that the system is very accurate because all calculation errors were solved during the development stages. The users actually participated in making the system accurate because they explained the workflow, dependencies and show all parameters clearly. User participation in the system design made the system easy to use without causing any mistake.

In contrast, the users stated that they have bad experience with other systems developed by other methods. They were not used to workflows added by the other systems and they took much time to learn how to use these systems. Also, they were not satisfied with the format of the output reports and bills. So, they did many mistakes during the learning phase and received many complaints from indirect users. They gave up using these systems because of the low accuracy and new tasks needed to be added to make the systems more accurate.

5) USER SATISFACTION

We also analyzed the responses of users towards the factors affecting user satisfaction, and these factors include:

a: WORKFLOW AND CUSTOMIZATION

The customization of the system to users and business requirements makes users more satisfied with the system [16]. In RAPD, users participated in the design and expressed their requirements effectively. They refined the design at each workshop according to the artifacts. The usability testing also assisted the users in verifying and validating the produced services. To users, the most important points were the sequence of operations and the output format. Previously when using RAD alone, the developers decided where to start the programming process. But with RAPD, the developers had to build the functions that can achieve the sequence of operations based on dependency priority so that the users can test them and give feedback.

b: CONTENT

The well consideration of the workflow and output formats made the users satisfied with the content of the system. An important issue that was raised during the development is the number of queries and their output reports. When more functions were generated and the users saw the strength of the system, they kept on adding new queries for generating useful reports. For example, more data management, items tracking and forecasting functions were demanded by the direct users. The developers followed the style of expressing the power of the developed queries, and if the users were not satisfied, they considered the requested queries in the new stage. The developers also explained the dependency priority to users and both side committed to the dependencies. When the third design stage was completed, the users found the query services were enough. RAPD helped the company produce a system with appropriate content and therefore the system was accepted by users, which agrees with [16].

c: EASE OF USE AND COMPLEXITY

Users participated in designing the interface of the system, and the output format, such as bills and financial reports. The reports were exactly the same as the paper-based formats so they are simple and easy to understand. The users ensured that any complex feature has to be simplified. So, the resulting interface was simple and the direct users who use all of the services found the system easy to use. They stated that they felt familiar with the structure of the system, and they easily understood the tasks represented by each service provided by the system. According to [16], users can be more satisfied when the system is easy to use.

d: DELAY

Delay of retrieving information negatively affects user satisfaction [16]. In the developed system using RAPD, users experienced delay in a creating weekly and monthly account balances due to the large amount of data. This issue was solved in RAPD stage 3 as developers found they need to use aggregation of data. After that, the users were satisfied with the speed of information retrieval. We also found that direct and indirect users expressed their satisfaction from the product delivery time which was less than two months (47 days). The reusable components assisted in generating all services without costing much development time or effort. In total, the product became usable in a short time compared to other methods. For example, one auctioning shop stated that they had experience with developing a similar system using other methods and the delivery time was almost five months that is longer than RAPD.

C. RAPD POST COVID-19 RESULTS

The results of the development analysis questionnaires show that RAPD performed well during COVID-19 and can maintain its effective performance after the crisis, as shown in Table 3. All respondents agree that RAPD was effective during COVID-19 and can be used in normal conditions for building other applications. The benefits of RAPD are shown in Table 3, where developers agree that RAPD is useful for the company. However, 4% of respondents think that RAPD may increase team size because documentations require extra staff. This issue was resolved using the video recording feature. Also, while most respondents think that RAPD supports clear planning, collecting user stories and interacting with users, only 4% do not agree because some developers were not familiar with RAPD before. A high percentage of developers stated that they did not know RAPD, RAD or PD before, as shown in Table 3. But, the most important issue is that most developers are willing to learn RAPD even they do not know RAD or PD.

Table 3 also shows that developers preferred the video conference tool during the design, and they think they are qualified enough to participate in RAPD if it is used for outsourced software development in other countries. Most respondents agree that RAPD does not need modifications to fit with design in normal life conditions. This is because RAPD rapid and participatory nature assist developers in building usable software. A low percentage of respondent think that video conference tool can be replaced by face to face workshops.

TABLE 3. RAPD development analysis results.

category	Question	SA %	A %	N %	D %	SD %
RAPD in different condition	effectiveness during COVID	91	9	0	0	0
	usage after COVID	87	13	0	0	0
	usage in other applications	78	17	4	0	0
	Reducing development cost	87	9	4	0	0
	Reduces team size cost	78	13	4	4	0
	On-time delivery	87	4	9	0	0
RAPD benefits	Team commitment to schedule	87	9	4	0	0
	Clear planning	91	4	0	4	0
	Collecting user stories	87	9	0	4	0
	Interacting with users	87	4	4	4	0
	Avoiding conflict with users	83	13	4	0	0
	Enabling documentations	87	9	4	0	0
	Addressing requirements	87	13	0	0	0
RAPD learnability	Familiarity of RAPD	70	13	0	17	0
	Familiarity of PD	35	17	4	30	13
	Familiarity of RAD	61	17	0	13	9
	Learning RAPD while knowing RAD and PD	78	13	9	0	0
	Learning regardless RAD and PD	78	17	0	4	0
RAPD remote design	Cooperative design globally	83	13	4	0	0
	Using video conference for design	83	13	0	4	0
	RAPD need changes for post COVID-19	2	3	0	8	87
	RAPD remote work helps collecting requirements	91	9	0	0	0

The respondents answered the open answer question that the video conference supporting remote cooperative design is the most important feature, particularly nowadays as remote work has become a daily routine. They added that the least favorable feature of RAPD is incorporating many users in the design which may complicate the process, particularly when users have low computer proficiency degree. Others, who were using software methods other than RAD and PD, added that they do not mind learning RAPD but this requires time.

The results of the usability questionnaire show that remote work has become one of the usability factors as most users, Table 4, require that the system should support remote work although they sometimes prefer to work onsite. Most users also are stratified with the system developed using RAPD and agree that the system is usable, as shown in Table 4. The reason why some do not agree is that some users need more

TABLE 4. RAPD usability analysis results.

category	Question	SA %	A %	N %	D %	SD %
Remote work	Work remotely all the time	80	6	4	6	4
	Work remotely sometimes	90	6	2	2	0
RAPD usability	Efficiency against paper based system	82	16	2	0	0
	Efficiency against desktop application	88	8	4	0	0
	Easy to learn	90	4	4	2	0
	Easy to memorize	92	4	2	2	0
	Accurate data input	88	8	2	2	0
	Accurate results	92	6	2	0	0
RAPD user satisfaction	Customized to work	92	8	0	0	0
	Query services are satisfying	88	6	2	2	2
	Reports are satisfying	92	4	2	2	0
	Functions are not complex	90	8	2	2	0
	Delay in submitting queries	82	6	8	2	2
	Delay in viewing query results	80	8	4	4	4
	Reports are easy to understand	90	8	2	0	0
	Well-structured display /pages	94	6	0	0	0
	Easy to use	96	4	0	0	0

query services, some have a very low computer proficiency, and the slow internet increases the latency of submitting queries and viewing results.

The most important summary of this quantitative results is that a high percentage of users agree that the system is usable, and easy to be understood and used with low complexity. On the other hand, a high percentage of developers agree that RAPD is more useful for the company than the current method, which is RAD, and RAPD can be used for developing different applications at different times.

V. LIMITATIONS

The limitations of our research concern the used research method which is action research. Because action research allows researchers to study a specific problem in a specific context to find a suitable solution, this process limits the generalizability of the results. Therefore, further research is needed to examine RAPD in different context. The other limitation is assuming that the company has a team whose one or more members have experience in participatory design which is new to the Palestinian software development culture. Also, assuming that end users would accept participation and having the skills to participate in video conferencing. So, further research is needed to study the factors influencing adopting new software development methods in larger companies, for larger projects, and with different cultures of end users.

VI. CONCLUSION

Software development during crisis, such as COVID19, faces new challenges identified by the creation of usable and accepted systems in a short time. To save time, businesses adopted on-shelf systems that do not fit with their requirements. This caused new challenges, such low accuracy long training time on using these systems. On the other hand, using agile methods that do not focus on capturing user experience also made the produced systems inappropriate for business workflow. Therefore, this paper has proposed a framework called RAPD that integrates two well know software development methods. RAD allows for rapid development and PD allows for user participation in the design process. To make the produced system more acceptable, usability testing methods are also added to all design phases. For the company, RAPD was useful as it offers sufficient management plan, reduces development time and cost, and eliminates conflicts with end users. The company adopted the new framework easily as it is a small new company with young development teams, which agrees with [47]. For user, RAPD made the developed system usable and acceptable as it is customized to the workflow, accurate and time saving. We also found that software development is affected during COVID19, and video conference enabled PD to create more concise design documentations. The post COVID-19 analyses show that RAPD, its tools and process can still be used to develop different applications in normal conditions without crisis. Future work will include using RAPD for software development in another context so that it can be generalized.

ACKNOWLEDGMENT

The authors would like to thank Palestine Technical University – Kadoorie for their support in funding this publication.

REFERENCES

[1] H. Mansour, "How successful countries are in promoting digital transactions during COVID-19," *J. Econ. Stud.*, vol. 49, no. 3, pp. 435–452, 2021.

[2] T. Mettler and R. Pinto, "Serious games as a means for scientific knowledge transfer—A case from engineering management education," *IEEE Trans. Eng. Manag.*, vol. 62, no. 2, pp. 256–265, May 2015.

[3] S. Alyahya, R. Bin-Hezam, and M. Maddeh, "Supporting remote customer involvement in distributed agile development: A coordination approach," *IEEE Trans. Eng. Manag.*, early access, Jan. 6, 2022, doi: 10.1109/TEM.2021.3131964.

[4] A. Addone, R. De Donato, G. Palmieri, M. A. Pellegrino, A. Petta, V. Scarano, and L. Serra, "Novelette, a usable visual storytelling digital learning environment," *IEEE Access*, vol. 9, pp. 168850–168868, 2021.

[5] G. Boella, A. Calafiore, E. Grassi, A. Rapp, L. Sanasi, and C. Schifanella, "FirstLife: Combining social networking and VGI to create an urban coordination and collaboration platform," *IEEE Access*, vol. 7, pp. 63230–63246, 2019.

[6] G. Islam and T. Storer, "A case study of agile software development for safety-critical systems projects," *Rel. Eng. Syst. Saf.*, vol. 200, Aug. 2020, Art. no. 106954.

[7] I. Sommerville, *Software Engineering*, 7th ed. Reading, MA, USA: Addison-Wesley, 2004.

[8] P. Beynon-Davies and S. Holmes, "Integrating rapid application development and participatory design," *IEE Proc. Softw.*, vol. 145, no. 4, pp. 105–112, Aug. 1998.

[9] A. Cockburn, *Agile Software Development: The Cooperative Game*. Reading, MA, USA: Addison-Wesley, 2006.

[10] M. Brhel, H. Meth, A. Maedche, and K. Werder, "Exploring principles of user-centered agile software development: A literature review," *Inf. Softw. Technol.*, vol. 61, pp. 163–181, May 2015.

[11] A. Ananjeva, J. S. Persson, and A. Bruun, "Integrating UX work with agile development through user stories: An action research study in a small software company," *J. Syst. Softw.*, vol. 170, Dec. 2020, Art. no. 110785.

[12] J. McKay and P. Marshall, "The dual imperatives of action research," *Inf. Technol. People*, vol. 14, no. 1, pp. 46–59, Mar. 2001.

[13] C. V. Geambaşu, I. Jianu, I. Jianu, and A. Gavrilă, "Influence factors for the choice of a software development methodology," *Account. Manag. Inf. Syst.*, vol. 10, no. 4, pp. 479–494, 2011.

[14] E. M. Rogers, *Diffusion of Innovations*. New York, NY, USA: Free Press, 2003.

[15] J. Nielsen, "Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier," *Comput. Sci.*, 1994.

[16] D. T. Green and J. M. Pearson, "A confirmatory factor analysis of two website usability instruments," in *Proc. 3rd Annu. Workshop HCI Res. MIS*, 2004, pp. 35–39.

[17] Y.-A. Daraghmi, M. Daadoo, and D. Eleyan, "User centered method for enhancing the adoption of software systems in Palestine," in *Proc. Int. Conf. Future Netw. Distrib. Syst.*, Jul. 2017, pp. 1–11.

[18] J. Martin, *Rapid Application Development*. New York, NY, USA: Macmillan, 1991.

[19] S. McConnell, *Rapid Development: Taming Wild Software Schedules*. Washington, DC, USA: Microsoft Press, 1996.

[20] T. Rabuske, "Polymath: A platform for rapid application development of modular EDA tools," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.

[21] R. Rosyad, A. Syukur, and R. Rahim, "Multimedia prayer application for education with rapid application development method," in *Proc. 7th Int. Conf. Cyber IT Service Manag. (CITSM)*, Nov. 2019, pp. 1–4.

[22] H. Qodim and R. Rahim, "Islamic calendar: Prototype of Hijri calendar application using rapid application development method," in *Proc. 7th Int. Conf. Cyber IT Service Manage. (CITSM)*, Nov. 2019, pp. 1–4.

[23] S. Hassan, U. Qamar, and M. A. Idris, "Purification of requirement engineering model for rapid application development," in *Proc. 6th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Sep. 2015, pp. 357–362.

[24] T. Wahyuningrum, G. F. Fitriana, A. C. Wardhana, M. A. Sidiq, and D. Wahyuningsih, "Developing suicide risk idea identification for teenager (SERIINA) mobile apps prototype using extended rapid application development," in *Proc. 9th Int. Conf. Inf. Commun. Technol. (ICoICT)*, Aug. 2021, pp. 92–97.

[25] S. Bødker, C. Dindler, O. S. Iversen, and R. C. Smith, "Participatory design," in *Synthesis Lectures on Human-Centered Informatics*, vol. 14, no. 5. San Rafael, CA, USA: Morgan & Claypool, 2021.

[26] S. Kuhn and M. J. Müller, "Participatory design," *Commun. ACM*, vol. 36, no. 41, pp. 26–28, 1993.

[27] C. Weng, D. W. McDonald, D. Sparks, J. McCoy, and J. H. Gennari, "Participatory design of a collaborative clinical trial protocol writing system," *Int. J. Med. Informat.*, vol. 76, pp. S245–S251, Jun. 2007.

[28] A. Hume, N. Ferreira, and L. Cernuzzi, "The design of a privacy dashboard for an academic environment based on participatory design," in *Proc. 47th Latin Amer. Comput. Conf. (CLEI)*, Oct. 2021, pp. 1–10.

[29] V. Widyanuri and J. R. Octavia, "Participatory design of interactive TV user interface for elderly people in Indonesia," in *Proc. 3rd Int. Conf. Inf. Commun. Technol. (ICoICT)*, May 2015, pp. 300–303.

[30] A. Zewge, Y. Dittrich, and R. Bekele, "Adapting participatory design to design information system with rural Ethiopian community," in *Proc. AFRICON*, Sep. 2015, pp. 1–5.

[31] M. M. Hassan, A. Moreno, E. Sutinen, and A. Aziz, "On the participatory design of Jeliot mobile: Towards a socio-constructivist mlearning tool," in *Proc. Int. Conf. Learn. Teaching Comput. Eng.*, Apr. 2015, pp. 120–123.

[32] S. Pilemalm, P.-O. Lindell, N. Hallberg, and H. Eriksson, "Integrating the rational unified process and participatory design for development of socio-technical systems: A user participative approach," *Des. Stud.*, vol. 28, no. 3, pp. 263–288, May 2007.

[33] M. M. Hassan, A. Moreno, and M. Tukiainen, "Participatory consensual assessment technique: End-users assessing outcomes of participatory design," in *Proc. Int. Conf. Comput., Eng., Design (ICCED)*, Nov. 2017, pp. 1–5.

[34] S. Pilemalm and T. Timpka, "Third generation participatory design in health informatics—Making user participation applicable to large-scale information system projects," *J. Biomed. Informat.*, vol. 41, no. 2, pp. 327–339, Apr. 2008.

[35] J. Nielsen, *Usability Engineering*. New York, NY, USA: Academic, 1993.

[36] J. Rubin, *Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests*. New York, NY, USA: Wiley, 1994.

[37] B. Bygstad, G. Ghinea, and E. Brevik, "Software development methods and usability: Perspectives from a survey in the software industry in Norway," *Interacting Comput.*, vol. 20, no. 3, pp. 375–385, May 2008.

[38] A. W. Kushniruk and V. L. Patel, "Cognitive and usability engineering methods for the evaluation of clinical information systems," *J. Biomed. Informat.*, vol. 37, no. 1, pp. 56–76, Feb. 2004.

[39] A. M. Joseph, N. Jali, A. J. R. Jupit, and S. K. Jali, "EMarket for local farmers," in *Proc. IEEE 19th Student Conf. Res. Develop. (SCOReD)*, Nov. 2021, pp. 30–35.

[40] K. Ayuningtyas and N. Z. Janah, "Development and UI/UX usability analysis of pinjemobil web-based application using user satisfaction model," in *Proc. Int. Conf. Appl. Eng. (ICAE)*, Oct. 2018, pp. 1–6.

[41] D. E. Avison, R. M. Davison, and J. Malaurent, "Information systems action research: Debunking myths and overcoming barriers," *Inf. Manag.*, vol. 55, no. 2, pp. 177–187, Mar. 2018.

[42] A. G. Maail, "User participation and the success of development of ICT4D project: A critical review," in *Proc. 4th Annu. Workshop*, 2011, pp. 1–14.

[43] R. Heeks, "Do information and communication technologies (ICTs) contribute to development?" *J. Int. Develop.*, vol. 22, no. 5, pp. 625–640, Jun. 2010.

[44] R. Heeks, "ICT4D 2.0: The next phase of applying ICT for international development," *Computer*, vol. 41, no. 6, pp. 26–31, Jun. 2008.

[45] L. L. Dodson, S. R. Sterling, and J. K. Bennett, "Considering failure: Eight years of ITID research," in *Proc. ACM Int. Conf.*, 2012, pp. 56–64.

[46] R. Tongia and E. Subrahmanian, "Information and communications technology for development (ICT4D)—A design challenge?" in *Proc. Int. Conf. Inf. Commun. Technol. Develop.*, May 2006, pp. 243–255.

[47] K. A. Kozar, "Adopting systems development methods: An exploratory study," *J. Manag. Inf. Syst.*, vol. 5, no. 4, pp. 73–86, Mar. 1989.

YOUSEF-AWWAD DARAGHMI received the bachelor's degree in electrical engineering from An-Najah National University, Palestine, in 2002, and the master's degree in computer science and information engineering and the Ph.D. degree in computer science and engineering from the National Chiao Tung University, Taiwan, in 2007 and January 2014, respectively. He is currently an Associate Professor with the Computer Systems Engineering Department, Palestine Technical University—Kadoorie. He has published several papers about traffic prediction, ITS crowdsourcing, and VANETs in highly distinguished journals and conferences. His research interests include software engineering, blockchain, intelligent transportation systems (ITS), and vehicular *ad-hoc* networks (VANETs). He received the Best Paper Award from International Conference on Intelligent Transportation Systems Telecommunications, in 2012. He serves as a Technical Program Committee Member for the International Conference on Connected Vehicles and Expo, International Conference on Intelligent Transportation Systems Telecommunications, International Conference on Signal Processing, International Conference on Electric Vehicle and Vehicle Engineering, and Asia-Pacific Network Operations and Management Symposium. He is a Reviewer of highly distinguished journals, such as IEEE TRANSACTION ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *IEEE Communication Magazine*, *IEEE Network* Magazine, and others.

EMAN-YASSER DARAGHMI received the B.S. degree in communication and information technology from Al-Quds Open University, in 2008, and the M.S. degree in computer science and the Ph.D. degree in computer science and engineering from the National Chiao Tung University, Taiwan, in 2011 and 2015, respectively. She is currently an Associate Professor with the Department of Applied Computing, Palestine Technical University—Kadoorie (PTUK). Her current research interests include distributed computing, cloud computing, distributed systems, the internet technologies, and blockchain. She serves as a technical program committee member for several conferences and a reviewer for highly distinguished journals.