A Framework for Requirements Prioritisation Process in an Agile Software Development Environment: Empirical Study

Rami Hasan Al-Ta'ani[#], Rozilawati Razali[*]

[#]Research Center for Software Technology and Management, Faculty of Information Science and Technology,
Universiti Kebangsaan Malaysia, Bangi, Selangor, 43600, Malaysia
E-mail: 1ramitaani@yahoo.com, rozilawati@ukm.edu.my

Abstract— **User requirements are in high demand due to advancements of technology and business needs. Therefore, it is extremely difficult to achieve user requirements at one single time. Agile methods have been introduced to mitigate such issues by implementing user requirements incrementally and iteratively. The top significant aspect in agile development is selecting a group of prominent requirements to be implemented in iteration. This could be derived from continuous requirements prioritisation (RP). To assign which requirements should be selected by practitioners is considered one of the obstacles faced in the process of implementing the inter-iteration period. This study aimed to identify the RP elements and factors that should be considered through the RP process in agile software development. The elements and factors were identified through a review of previous studies and in-depth interviews involving 18 agile practitioners from different software development organizations in different countries (i.e., Malaysia and Jordan), over a period of 2 years. To get the best result, the data is analyzed by using grounded theory techniques (Open, Axial and Selective coding). The elements and factors from the RP process will guide practitioners to an effective RP process that can produce high-quality requirements, which are expected to meet the needs of the intended users.**

Keywords— **requirements prioritisation; requirements analysis; agile methods**

I. INTRODUCTION

Organisations have to respond to new opportunities and markets, dynamic economic conditions, and the emergence of competing products and services. Software systems are created to help organisations respond to such environments. To ensure the systems are ready to operate as and when needed, the conventional phase-to-phase software development is no longer practical. Instead, a methodology that enables rapid development and delivery is required [1] and for this, agile methods are proposed to cater to organizational needs.

Agile methods employ an iterative and incremental approach to software development [2]. They respond quickly to changing requirements by involving stakeholders in the process [3]. However, rapidly developing software systems to cope with the incoming changes imposes some challenges [4]. When several requirements change simultaneously, they become volatile and disorganised [5]. This complicates the prioritisation process that determines which requirements to be developed [4], particularly when it involves multiple stakeholders [4], [6].

Ensuring that the correct requirements are prioritised is essential in agile methods because they focus on implementing the most valuable features for customers [7].

This means practitioners must have an in-depth understanding of the RP process in order to ensure that the correct requirements are prioritised [8]. In other words, the elements and factors that need to be taken into consideration during RP process in agile development should be known. Thus, this paper is aimed towards articulating the process in which the proper stakeholders and requirements can be identified and subsequently prioritised in the agile methods environment.

The paper consists of six sections. It begins with an introduction followed by the related works on the RP process. Then it is followed by a brief explanation of the method and how it was used. After that, it presents the empirical results of the study. In addition, it shows and discusses the proposed framework. Lastly, the paper concludes the main findings and offers suggestions for future research.

II. MATERIALS AND METHODS

Requirements prioritisation is significant in traditional requirements engineering and more so in agile requirements engineering as it is one of the major methods used to produce value in a timely manner [9]. In other word, one of the core action within the agile methods is to employ requirements prioritisation that helps in assigning the correct

requirements [10]. The high demand from customers and the time consumption as well as limited resources such factors need to be considered and the product should consist of the most important functions [11]. In addition, the most difficult point is creating a new software and ensuring that it functions as expected, moreover, the hardest is to modify further [12].

Agile methods constitute requirements prioritisation [7]. Establishing priorities early in the project will offer greater options for obtaining better results and enhancing customer satisfaction [11], [13]. Most of the vital requirements will be implemented based on the RP process [7]. The one value of agile method is responding to change [3]. Due to continuous changes in requirements, the prioritization process in agile methods becomes more difficult [14]. Moreover, prioritisation is an ongoing process, which is performed at the beginning of each iteration, or in the course of the iteration [15]. Thus, the process requires a high level of organisation and systematisation.

The RP process has been defined as an action during which the most significant requirements for the software (or release) should be acknowledged [16-18]. Additionally, in order to deliver the most important function, the highest priority requirement has to be implemented in prior [19]. Thus, requirements prioritization is aimed at identifying the most vital requirements for a system.

As an insight into the RP process, it begins by stakeholders determine the requirements that should be implemented [20]-[22]. Given their knowledge of the system, developers have the most influential say in RP process [23] but they do not act alone. The opinions of the customers are nearly equally as important since they are the ones who require the system as well as have the authority towards it [22]. Thus, developers and customers must work together to prioritise the requirements [22], [24]. Due to differing natures of stakeholders, the RP process becomes rather complicated, particularly when that they come from two very different perspectives and operate in what is often a hectic and disoriented situation [6].

Choosing the right stakeholders to be involved in RP process is important in agile development [25].The stakeholders should possess significant knowledge about the agile development [8] as well as customers' needs and interests [23]. The contemplating process in prioritisation relies on joined learning experience of the stakeholders with respect to systematic (explicit) and anecdotal (tacit) knowledge [23], [26]-[27]. Moreover, trust between stakeholders is crucial in order for the RP process to be successful [5]. This can be cultivated through direct communication [28].

After establishing the team dynamic between stakeholders, project constraints are then considered in identifying the appropriate requirements. An important consideration in this regard is cost, that is the requirement that costs the least is often prioritized [10]. To determine the most value for the requirement, cost-benefit analysis is performed [29]-[30]. The analysis is important as the accurate blend of requirements for iteration would exploit values for the business [31]-[32]. The RP process also depends on the available personnel to implement the iteration and the project schedule [29]. The agile development process operates within rigid delivery dates because its introduction into the market is crucial. Delaying iterations will certainly lead to postponing its introduction which is unacceptable in agile development process [33]. Moreover, the possibility of the requirements to cause project failure is also considered during the prioritisation process [32], [34].

The nature of requirements influences the RP process. It determines their complexity, which subsequently affects the selection of iteration [23]. Dependencies can be chronological or architectural. A requirement that is judged to be very complex to develop and where the risk of project failure is too high may not be implemented [29]. Nevertheless, importance and urgency may outweigh complexity and thus be included in the iteration depending on the stakeholders perspective [29]. Important requirements are functions that are required early and bring strategic business values to the organisation [35]-[36]. The stability or volatility of requirements is another important consideration, as the uncertainties associated with instability may prove costly and affect the project schedule [25].

The above reviews indicate that previous studies have attempted to identify the contributing factors and elements for requirements prioritisation process in agile development. However, these factors and elements are isolated. None of the studies have clearly depicted the interrelation among those factors. This study, therefore, aims to empirically integrate the factors and elements as a framework which assists multiple stakeholders in determining which requirements to be developed.

This study identifies the factors and elements that influence the selection of requirements during the prioritisation process in an agile development. The identified factors and elements could help people work in RP process to determine the RP and consider which requirements should be implemented first. To ensure the mentioned factors and elements are holistic and practical, the study applied both theoretical and empirical approaches. This comprised a review of past studies and interviews with practitioners from the software industry. Fig. 1 outlines the research design containing the core activities involved in the study.
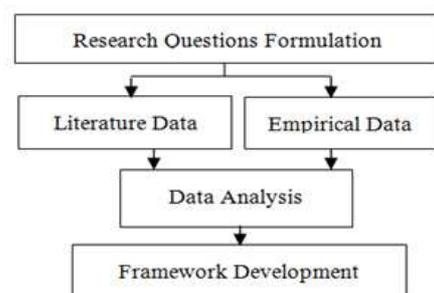

Fig.1 Research design

A. Formulate Research Questions

The study aimed to answer the following research questions. The questions were derived based on a preliminary study of the subject matter.

What are the factors that influence the selection of requirements during the prioritisation process in agile development environment? How do these factors and elements relate to each other during the requirements prioritisations process?

B. Literature Data – A Review

The aim of the review was to determine the factors and process that should be considered during the RP process in an agile software development project. The keywords used in searching the articles included "requirement prioritisation", "agile method", "requirement prioritisation technique", "agile development environment", "prioritisation process", "stakeholders selection", "agile software development", "requirement analysis", "agile requirements prioritisation", and "requirements prioritisation process". The analysis was conducted on previous related work. The reviews were based on articles concerning RP process in agile development published within a period of twenty years. The articles were searched in online databases such as IEEE, Elsevier, and Springer. The search covered both journal articles and conference proceedings. The results of the review have been discussed in the earlier study [37].

C. Empirical Data – In Depth Interview

In order to verify the factors and process elements found in the literature, interviews with several domain experts and practitioners from the industry were conducted. The face-to-face interviews with experts sought to collect information on the specific subject matter [38]. The approach was selected because it is well suited to a research that requires an understanding of deeply rooted, delicate phenomenon, responses to complex systems, processes or experiences. The in-depth interviews offer the opportunity for clarification and detailed understanding [39].

The interviews used semi-structured questions, which were constructed based on the RP process elements found in the review. Prior to the real session, a pilot study was conducted with four persons; one of whom is an agile software development researcher. The pilot study helps to validate the accuracy and completeness of the questions and determines the feasibility of the session. The feedback drawn from the pilot study was used to improve the planning of the real session.

The defined selection criteria of participants were set in order to guarantee that the gathered data is meaningful. The potential participants must possess some experience in agile software development and RP process. To fulfil this requirement, the study employed purposive sampling [40].

The researcher adopted a flexible trend as to the interview place and time to ensure the participation of respondents. While some interviews were conducted face-to-face, others were conducted through Skype. After the approval was obtained, invitations were sent emails to the interviewees along with the interview questions and interviews commenced from October 2013 and lasted until April 2015. The informants were provided with the study purpose and their role as informants and they were informed of the confidentiality of the information obtained from and their anonymity. Each interview session was recorded through an audio recorder with the informants approval – the interview sessions lasted for approximately 30-70 minutes each.

As mentioned, the entire interviews were audio-recorded, with the interviewees' permission obtained prior to the interview after which the recording was transcribed into a text document. According to Morgan and Guevara (2008), audio recording is important for in-depth interviews as it provides an accurate record of the interview and according to other studies (e.g., [41]-[43]), the audio recording can be replayed, transcripts accurately documented, the record permanently preserved in its sequence and there it is an enriching detail of evidence. The researcher also made follow-ups through email and telephone to seek clarification of what was recorded.

D. Analysis of Data and Construction of the Framework

Data analysis was carried out with the use of grounded theory coding techniques [44]. Coding refers to a process wherein data is fractured, conceptualized and integrated to create a theory [45]. With regards to grounded theory, data analysis is initiated by open coding and progresses to axial coding and selective coding [46]. The three mentioned phases enable the analysis of qualitative data in a deductive and inductive manner [45]. The coding methods are deemed to be technical analytical procedures [47] involving the encapsulation of insights, production of themes and creating a theory from data [48].

III. RESULTS AND DISCUSSION

In the following sections, this paper presents the results collected from eighteen interviewees from different countries. The interviewees are from different software development companies with over one year experience, fifteen from a private software company, while three from public government agencies. Out of these, 9 were from Jordanian organizations, 4 from Malaysian organizations, and 1 from Palestinian organization, 1 from Singaporean organization, 1 from a U.S. organization, 1 from an Indian organization and one from a Chinese organization. In order to respect their confidentiality, we refer to the participants by numbers R1 – R18. All were using agile methods, specifically scrum and extreme programming (XP) – two of the most popular agile methods today [49].

All of them were experienced in basic Agile practices namely iterative and incremental development with various iteration lengths, iteration planning, estimation and planning of user stories and tasks, testing, status report meetings (daily stand-up), and frequent release of working software. They were also engaged in requirements prioritisation process in varying phases – with some of the participants even being certified, Scrum Masters.

On the basis of their experience and work on agile projects, the eighteen participants held different levels of experience, with some fresh (first time in Agile project), others experienced in working in many other Agile projects. Some others had over 5 years of experience in such projects, with an aggregate of 183 years of software development experience and 69 years of experience in the development of agile software.

In the following sections, this paper selected quotations drawn from interviews that shed light on the concepts. Due to space limitations, we cannot describe all the underlying key points, codes, and concepts from our interviews that further ground the discussion.

A. Human

The analysis revealed that humans are one of the important areas of agile software development methods for

the requirements prioritisation process. Based on empirical studies, there are four factors identified namely top management, product owner, development team and clients.

1) Top Management: The participants identified several characteristics for top management to ensure that projects are implemented properly, which are empowering, availability, supporting. In the following paragraphs, these characteristics are described in detail.

- *Empower*: The authority given is important from the top management to the product owner. This is due to the fact that sometimes the project might be reaching a critical stage, so urgent decisions need to be made. As mentioned by a consultant: "if we found that the project is in danger we can talk with high executives or managers to make some decisions or give the authority to the product owner" R17.
- *Availability*: The availability of top management at the beginning of the project is considered important especially during the requirement identification for the project as people need answers with regards to the requirements and the requirements need to be more visible for the team. A product owner said: "So, there are people who give you high-level requirements that are very vague and general, and they are often absent"R6.
- *Supporting*: Top management plays crucial roles in the success of the project using agile methodology by supporting the people working on the project. As mentioned by a general manager: "I am always outside the project team but as one of the people who are involved in making the project succeed. So, I am always in the supporting role" R10.

2) Product Owner/ Project Leader: Product owner characteristics are a crucial key success of any agile software development because he/ she plays a different role in the team as he/she has a connection between the development team and the clients. As the study revealed, there are critical characteristics that a product owner should have to be one of the team members. It is as follows: Authority, Availability, Experiences, Knowledge and Leadership. Therefore, the aims of the next points are to explain the characteristics of the product owner as unveiled by interviewees.

- *Authority*: The product owner should have the authority. Thus, he/she can give the response to questions within seconds or minutes. He/ she also do not need to ask the top management about the decision or what he/she should do, as mentioned by a programmer: "the authority and power should be given to the product owner" R16, R1.
- *Availability*: The top management should select the product owner who is available to be close to the development team and to solve any issues they face very fast and understand what is going on regarding the process. A system analyst and developer pointed out that: "the product owner is close to the team and is always available with the team. So, if there is any problem, we remedy it in the same moment." R7.

- *Experiences*: The product owner's experience in the process and prioritisation using agile software development methods is one characteristic that should be selected. It takes into account the top management's perspective once a product owner is selected to be one of the team members - this is as mentioned by project leader "project leader should have sufficient experience to know the required features; furthermore, he must have the experience on how to prioritize the release" R15.
- *Knowledge*: A product owner should be a knowledgeable person to play a role as a product owner. There are two types of knowledge that a product owner has namely business and technical knowledge. Business knowledge is a deep understanding of the business and what is required from the user. Furthermore, he/she should understand the technical knowledge to understand what is going on. As mentioned by the consultant: "A product owner has to understand both the business aspect and the technical aspect" R17, R12.
- *Leadership*: Leadership is a comprehensive concept that has many attributes that a human should possess. A product owner is, after all, a human being, so he/she brings his/her values to the project. Therefore, the top management should take into account the leadership characteristic once a product owner is selected for the project. They should be able to establish clear vision, guide the team, solve the problem, demonstrate high communication skills, influence the people and collaborate. Therefore, in the following points, each point is discussed separately as it appeared in the empirical study.
- *Having Vision*: Product owner should have a vision and he/she is preferable to be chosen from the top management because he/she is capable of sharing his/ her vision for the people working on the project as stated by a product owner: "he must be able to help define a vision as to why we are doing something" R12.
- *Guide The Team*: The ability to guide team members without force can leave them feeling empowered and accomplished. Thus, the product owner who has the ability to guide the team is suitable as a product owner, as a general manager mentioned: "the project leader should be able to guide the development team around the requirement" R10.
- *Solve Problems*: The ability to solve any problem, which faces any team members in the project, is required out of a product owner. Therefore, the top management should take into account the capability of solving the problem between the team members once the product owner is selected. As claimed by a developer: "product owner must have the ability to solve problems" R2.
- *Communication Skills*: Good communication skills are also important in conveying information between the product owner and all people who have interests in the project especially with the client to get as much as he can from them about the product requirements. Also, he must be capable of conveying the requirements to

the development team clearly without any misunderstanding. As stated from the scrum master: "He should be capable of communicating with clients to gain the features from them" R14. This is also supported by a product owner who said: "Communicate very clearly to your team as to why we do this" R6.

- *Influencing People*: In order to be an effective product owner with good leadership he/she must be able to distinguish from the various forms of power and the top management should select the one most consistent with his/her leadership style, and has the characteristics to be able to influence the people working in agile software environment even he/she does not have enough authority. As stated by a product owner: "the product owner influences people or who has that ability is preferable to be selected" R12.

3) Development Team: As the analysis revealed, there are critical characteristics for a person to be a team member of a development team. These characteristics include experiences, availability, self-organized, communication skills and knowledge. Therefore, the aims of the next points are to explain the characteristics of the development team as mentioned by interviewees.

- *Experiences*: Experience is an important characteristic one should have when the top management is going to select team members. Particularly for those who have the experience in software development, there is no need to teach them how the software is developed using agile methodology and that will help deliver a project on time. As mentioned by a product owner: "experience in basically doing development work. I mean if you take someone who is a fresh graduate, he may take some time to understand the process and how things go" R6.

- *Availability*: Availability for the development team during the project is important and this is must be considered by the top management before selecting any member of the development team. As mentioned by experts we face some issues regarding the availability of the developer once they start the project. Some developers leave a company and that gets us trouble in terms of delivering the feature to the customer. As mentioned by a product owner "They work on the project half way and the critical developers or team members will tell me that they've resigned and they want to move on. So, that impacts the speed of the project, and we fail to finish it on time" R6.

- *Self-Organized*: Self-organization is considered a critical success characteristic of the team member in agile software development. as supported by the general manager when she said: "The team will work out between them, how they are going to manage the delay and how they are going to continue to work because that's one core thing of self-organized agile teams" R10.

- The collaboration and motivation are critical characteristics for self-organizing a development team

and every member should know what is going on, in the environment, they are working. As mentioned by a product owner "there are developers who are very outspoken, they are quite communicative but not very cooperative, they wait too long to a point that the target cannot be reached and they will get upset and they lose motivation very fast" R6.

- *Communication Skills*: Agile software development is a highly communicative environment, and in this regard, all participants strongly agree that all people working in this environment should have very good communication skills to easily communicate with each other and understand what they have to work on and deliver. As mentioned by a developer: "communication skill should be high because some development team member may have the authority and knowledge, but he/she may not have the communication skills to deliver his/ her ideas. Thus, this might lead to misunderstanding" R9.

- *Knowledge*: The selection process for the specific development team to be involved in a project must take into account the different areas and degrees development team knowledge. Individual knowledge could be categorized into two main categories: business and technology.

- Regarding the development team's business knowledge, every member of the team should have excellent business knowledge about the project to be implemented and the way the company works. As mentioned by a developer who claimed: "if we have a project about Islamic banking and the development team should know about Islamic banking from A to Z" R8.

- Beside the business knowledge, the development teams should have to have a good understanding of technical knowledge to achieve the tasks as required on time as stated by the developer: "They should have technical knowledge about the work process in order to achieve the tasks on time without delay" R9.

- *4) Clients Characteristics*: Engaging the clients in the software development life cycle is considered as an important thing in agile software methods. Therefore, the companies that are going to implement a product should have correct persons to play the role as clients. However, the differences between human skills and characteristics do exist and should be considered when selecting the clients as this could affect project implementation. Thus, the analysis revealed some of the characteristics the clients should have including authority, availability, communication skills and knowledge. Thus, the next points are going to discuss each of these characteristics as mentioned by the participants in this empirical study.

- *Authority*: Authorized client is a compulsory characteristic for one to become a team member who is going to implement the product until the delivery. Therefore, it is preferable for someone quite senior client to play that role to make a decision as soon as possible and the one who doesn't need to ask a top management certain situation as claimed by delivery manager when he said: "for any group you represent,

you should have the right to make a decision within that group. You can't be someone who says -I need to check with my boss. So, whoever is coming has to be the decision maker of that domain or that section of the business that they represent." R13.

- *Availability*: Client's availability means the client should be close to the product owner and to the development team as well. This is because when they need him/ her, he/she has to be on hand and accessible to answer any questions coming from the product owner or development team. Therefore, the client's availability is important as the analysis revealed that the participants consider the availability of the clients is important according to a product owner: "the clients who are close to us can provide feedback on how well the project is progressing and where change is needed very fast" R4.

- *Communication Skills*: Communication skills in the workplace are all about being able to explain information to team members clearly and simply, in a way that means requirements are understood and get done. It is for transmitting and receiving feedback clearly. Thus, clients who have good communication skills are preferable to be a one of the team members for agile software development to explain clearly what he/she wants from the team. As claimed by a product owner: "clients who are able to explain the work to the people who will do work are much preferred to be close to our team" R4.

- *Knowledge*: Clients come from different backgrounds and areas that reflect their knowledge. There are two types of knowledge that clients who will be team members in the project should have, which are business and technical knowledge. Business knowledge is needed by all businesses, while technical knowledge reflects on specific practicality. Combining between technical and business knowledge is considered to be important required during the agile software development, especially, in the requirements prioritisation process reflected in the decision making, the working environments so he/she is able to describe what he/ she wants exactly. As mentioned by the Scrum master: "the client usually doesn't know what he wants in detail because he/she lacks business knowledge" R14.

- On the other hand, the developer mentioned that the lack of client's technical knowledge can lead to something not wanted or meant by the client. When he said: "the client lacked the understanding in the technical matter. Therefore, he/she asked about something illogical" R11.

B. Process

This study links the factors and elements identified in the requirements prioritisation process. The process is a sequence of interrelated activities to achieve a requirements prioritisation systematically. The requirements prioritization process involves a sequence of activities by people (Top Management, Product Owner, Clients, and Development Team) from the companies that are going to implement a product requested by the clients. The analysis revealed that

the agile development methods for software development include five phases, namely Product Conception and Initiation, Release Planning, Sprint Planning, Sprint Execution and Sprint Review. Each phase contains certain contributing factors and elements that are explained in the next section.

1) Product Conception and Initiation Phase: Product conception phase refers to activities carried out before and at the beginning of the project. Six contributing factors identified in this phase are the Pre-Product Conception, Defined Product Goals and Objectives, Determine Product Scope, Determine Product Schedule, Determine Product Budget and Determine the Human Recourses.

- *Pre-Product Conception*: The analysis revealed that the prioritisation process is initiated when the top management determines the priority for the products they have. Therefore, there are a lot of requested functionalities and requirements coming from the business people to make sure that their business runs smoothly and easily. Therefore, the business people include a justification about why they need this functionality or system. Thus, all those requirements are placed in strategic planning for the organization then the top management should prioritize those products and what should be implemented first. This is argued by a programmer when he said: "Business people apply for a request from the director accompanied by reasons for his/ her (their) needs and what are the benefits from this project, then the top management will collect the needed projects and hold a meeting with the head of departments. This is in order to determine the priority for those projects and choose the project that should be implemented this year or next year" R16.

- *Define The Project Goals and Objectives*: The analyses revealed that, after prioritizing the projects by top management during the executive steering committee or executive management committee meeting, they determine the projects that should be implemented this year and define the project goals and objectives. As stated by a general manager: "the portfolio management group for the executive steering committee or executive management committee gets together and talk about the priority of their business and the goals and objectives for their projects" R10.

- *Determine The Human Resources (HR)*: Human resource refers to the individuals or personnel within an organization responsible for performing the tasks given to them for the purpose of achievement of goals and objectives of the project, which are possible only through proper selection. The human resource is important in delivering the project and can be determined based on the needs of the project, its size, and duration. In addition, human resource determination is also made based on the team members' ability to handle the task. Human resources need to be planned carefully to avoid the resources from being unused or redundant by the organization. As stated by consultant: "basically, before the requirements are determined, we need to conduct a

project initialization and identify the stakeholders (user, developer, owner of application)" R3.

- *Determine Product Scope*: The analysis shows that the scope of work is important to determine the coverage of the work to meet the goals and objectives of the project. In determining the scope of work, the establishment of the party responsible for a particular task or activity should be made clear in order to control the powers inherent in the implementation of the task. This was strongly supported by a delivery manager who elaborated that: "top management always says you can do the top fifty things and nothing else and that's your scope" R13.

- *Determine Product Schedule*: The product schedule is important in ensuring the smooth running of the project and the tasks get to be distributed and delivered by the development team according to the product schedule. The participant said that the determination of schedule should be made carefully by top management so as not to affect the course of the next project and so the objectives of the project can be met. As mentioned by system analyst and developer: "top management holds a session meeting to study and analyse the project and to determine the schedule for it" R7.

- *Determine Product Budget*: The informant said that the determination of costs should be made carefully at the beginning of the project to meet the project objectives. As claimed by a general manager: "we need to know our financial focus for the project such as how much we need to spend on it" R18.

2) Release Planning Phase: The analysis shows that the product owner creates a release plan. The release plan identifies high-level of product requirements of working software, which is (Group discussion for everyone, Product owner sits with the business people or Business people identify the requirements). After that, an agile project will have many releases, with the highest-priority features launched first. Therefore, the requirements prioritisation process should be conducted in the first place with high-level requirements depending on several criteria as claimed by participants. As revealed by an analysis there are two major factors where the people focusing on prioritizing a requirement, and these project constraint (schedule, budget, and scope) and requirement nature (such as visibility, business value, dependencies and complexity).

- *Define The Product Requirements (Product Backlog)*: The analysis revealed that selecting the project from different projects depends on its benefits and goals. A set of procedures should be done including identifying the project schedule, cost, and human resources. After that, the first step is to gather product requirements (backlog). This is supported by a developer who stated: "Based on group discussions with the people involved in this project (business side) and also under the supervision of people from the Department of IT all agreed on the requirements, which led to the productivity of the project" R2.

- *Prioritize The Product Requirements (Product Backlog)*: The analysis appeared that after gathering the requirements from different ways, there is a set of prioritisations that should be carried out for the requirements to determine which requirements should be implemented first depending on different criteria. The requirements priority happens in two rounds depending on several criteria. As claimed by a delivery manager: "requirements prioritisation comprises multiple passes on your requirement list" R13.

- *Prioritize Product Requirements (Product Backlog) Criteria*: Prioritization requirements criteria are almost the same regardless of which way to set the priority. As revealed from the analysis there are two main factors that are taken into account from people once they set a requirement priority, which are project constraints (schedule, budget, and scope) and requirements nature (visibility, business value, dependences, and complexity). As claimed by a developer: "The backlog prioritisation process is a group discussion. This happens first between the product owner and the client to prioritize the backlog depending on several criteria" R9.

- *Product Constraints*: The analysis from the empirical study showed that product constraints are considered an important factor, where the people set the priority for the requirements that were identified from the previous stage. The product constraint factors consist of a sub factor which are schedule, budget, and scope. As a product owner claimed: "we should trade off between time (deliver date) and cost. These are the main things that need to be taken into account when prioritizing the feature" R14. This is also supported by the product owner who mentioned about the scope: "I think what is also very important is the scope, because very frequently when you allow the stakeholder to be involved in processes like this, they will ask for more and more changes which lead to feature creep" R6.

- *Requirements Nature*: The nature of requirements is another important consideration when the people set the first rounds of requirements prioritization process. The analysis revealed that the nature of the requirements that the people focused on are (visibility, business value, dependencies, and complexity).

A developer stated that unclear requirements should be postponed especially that requirements do not have any dependencies on it when he claimed: "Unstable requirement and unclear one should be delayed naturally unless another requirement depends on it" R5.

On the other hand, another participant mentioned the business value when he stated: "we give the highest priority to some requirements that achieve the optimum size of the project" R2.

In addition, other system analyst and developer considered the dependency as a crucial factor in requirements prioritisation process and there should be no mistake on it when he stated that: "The requirements priority process is focused on dependencies between the

requirements and there should not be any mistake on it" R7.

The analysis also revealed that some requirements tend to have big details and they are very complex. The people in the requirements prioritisation process must divide it into the sub-requirements to determine the dependencies and to get the business value from it. As mentioned by the scrum master: "The most important thing is to know the complexity and to get more details about it. So, when it is detailed, the dependencies become clearer and we get the important feature" R14.

3) Sprint Planning Phase: The analysis demonstrated that the product owner and the development team did sprints planning, also called iterations, and they started creating the product within those sprints. Sprint planning sessions take place at the start of each sprint, where the team divides the product backlog (requirements) into sprints and then determines what sprint will be in the upcoming iteration depending on some criteria that they have considered. Therefore, the next points explain how they divide the requirements and what are the criteria considered when they prioritize the sprint with support given by the respondents.

- *Divide Product Requirements into Tasks (Sprints Backlog)*: As revealed by analysis, the identified initial requirements through different ways serve as a first step as mentioned in the previous sections. Then after that, they prioritize it in the first round by business people depending on several criteria. The product owner will bring the prioritized requirements into the development team to implement it as supported by a delivery manager (R13). But, this will happen after classifying each one and prioritizing it from the technical perspective. A developer said: "the prioritized features will be passed on to the development team by the product owner, where they are delivered accordingly and the development team would divide the feature into sprints" R9.

- *Prioritize Sprints Backlog Criteria*: The second round of prioritisation takes place between the product owner and development team depending on several criteria. The criteria most voted are schedule, size, visibility and technical risk. As claimed by a consultant: "the PO introduces the requirements and the team tries to discuss in more detail and ask questions, that are the most technical and after the conversation they will confirm the acceptance criteria and understand how much effort in the size and understand how much effort to be exerted. The product owner makes a decision to consider all those attributes like schedule pressure and technical risk as the team tells him" R17.

4) Sprint Execution Phase: Execute phase refers to activities carried out after prioritizing a backlog and dividing it into prioritized sprints. Two contributing factors are identified in this phase, which are distribute the task and implement the sprint.

- *Distribute The Tasks to The Development Team*: Distributing the tasks among the development team members is a product owner's responsibility. It means

that he/ she takes into consideration many things when he/she distributes tasks for the development team such as development team experience, availability and the ability to implement the task. This comes from the development team's side. Moreover, he/she has to look for the projects schedule and requirement dependencies. As mentioned by the developer and analyst expert: "In every project, there is a team leader, project manager or product owner and they are responsible for distributing the tasks for the team, depending on the technical differences between the development team" R16.

- Implement the sprint: Implementing a task by the development team is the next action after task distribution by a product owner as stated by a delivery manager (R9). Moreover, a developer stated: "the first step is implementing the highest priority task" R2.

5) Sprint Review Phase: The analysis shows that the development team holds a sprint review. In the sprint review, at the end of every sprint, the working product created during the sprint is demonstrated to the clients. Depending on the sprint review, there are two actions that occur. The first action is the clients giving their feedback about the product and the second one is the re-prioritisation might occur depending on several reasons as revealed by the analysis such as bugs (errors), requirements misunderstanding, dependencies, or time estimation.

- *Feedback*: The feedback session is an action that will be held after finishing a sprint implementation, which will be between all people and there are a lot of communications during this session between them. As stated by a developer: "The meeting with the client takes place once every two to three weeks, where we inform the client about what we have accomplished by using Demo and the feedback is given by the client" R9.

- *Re-Prioritisation*: The analysis revealed that the feedback from the clients during the session will determine the next actions in the agile software development which are implementing the next sprint or re-prioritizing the features depending on several factors. As claimed by a scrum master: "if the feedback reveals that the clients are unsatisfied. The team member will be called for a session to do re-prioritisation" R14.

- *Bugs or Error*: The bugs or error is considered as a cause for doing reprioritisation of the features especially when that error affects the whole project or system. As mentioned by a project leader: "if there was an item executed and after the test an error happened and at the same time there was a feature that needs to be implemented so if the error was critical we should delay the next iteration and give the critical error the highest priority" R15.

- *Requirements Misunderstanding*: Sometimes, there are misunderstandings about the features from the development team or the clients whereby the development team implements features not required by the clients or the client meant something else. Therefore, the reprioritisation session will be held to

determine the correct priority. A developer stated that: "when the feedback was unsatisfactory from the clients because we had developed requirements that they did not require we had to conduct re-priority" R9.

- *Dependencies*: dependencies sometimes lead to reprioritized requirements after implementing some requirements, if the development team discover there is something else that should be implemented before it; they have to reprioritize the requirements accordingly. As stated by a general manager: "if we have 10 requirements then we have a task to queue this requirement one by one - let's say in the fifth requirement we realize that there is something else we need to complete then we need to re-visit and reprioritize this and release it" R18.

- *Time Estimated*: Estimating the delivery time for each sprint is an important thing the development team should do for every task and how long and the time of delivery; for this, they can consult someone more experienced to estimate the time that because if there is any mistake on delivery date estimation, this will lead to reprioritizing of a requirement and in turn, it will affect the project schedule. As claimed by a project leader: "if we proposed that the task take 1 week but it takes 2 weeks, it will affect the environment we work in and this frequently happens. So, we have to extend the deadline of the project or iteration. And all these will affect the project delivery to clients. This leads us to re-prioritize the requirement and reschedule the project deadline" R15.

Based on the analysis of theoretical and empirical studies that have been conducted, Fig. 2 below elaborates a new framework after extracting information from experts. The proposed framework contains two essential components, namely human and process. Each component has several factors that need to be considered holistically during the RP process. The factors are interconnected during the process. The arrows between the factors in the framework indicate their relationships with each other. For instance, Development Team (DT) is involved in phase three, four and five while clients are just involved in phase two and five.

This study consists of two different components as mentioned previously. To summarize, human has four categories; Top management, product owner, clients and development team, while process consists of five phases, namely the product conception and initiation, release planning, sprint planning, sprint execution, sprint review. There are six factors contained in the phase of product conception and initiation namely, pre-product conception, define the project goals and objectives, determine the human resources, determine product scope, determine product schedule, determine product budget. The release planning phase involves the following factors; define the product requirements (product backlog), prioritize the product requirements (product backlog), prioritize product requirements (product backlog) criteria. Further, sprint planning phase has two factors namely, divide product requirements (backlog) into tasks (backlog sprints) and prioritize sprints backlog criteria. Moreover, the sprint

execution contains two factors, which are; distribute the tasks to the development team and implement the sprint. Sprint review, which is the last phase, consists of two factors, namely, feedback and re-prioritisation. The proposed framework shows how each of these factors and elements contained in different aspects or phases correlate with each other.
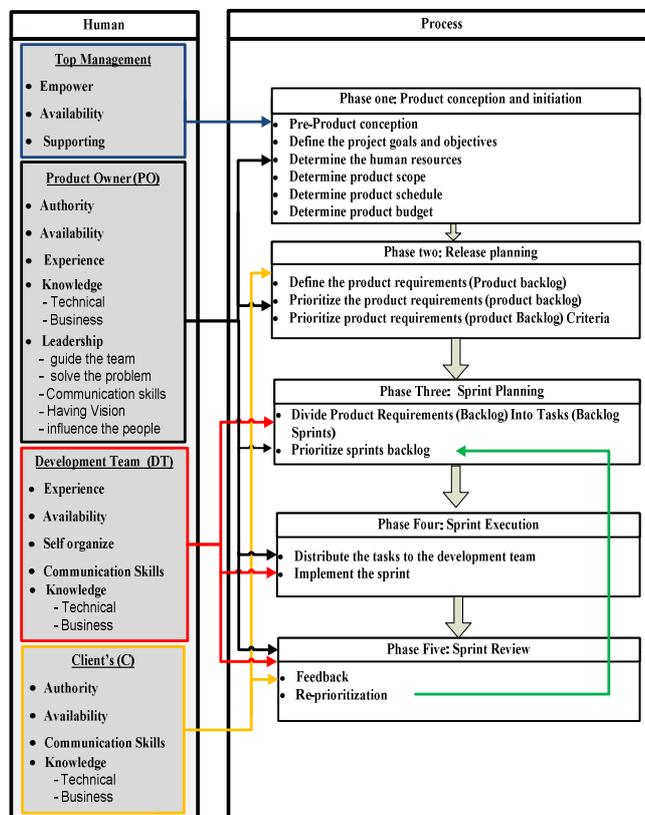

Fig. 2 A Framework of RP process in agile methods

IV. CONCLUSIONS

Requirements prioritisation in agile method is a core process to decide which requirement should go first. This paper discussed the RP process factors along with the corresponding elements concerning requirements priority, which should be considered during the RP process in agile methods. They were gathered through a review of previous studies and interviews with experts. The factors and elements in the framework would constitute a guideline for the practitioners in selecting the requirements. Better than relying on conventional wisdom, practitioners can execute a framework to be more systematic. The study is based on both theoretical and empirical approaches. To obtain the best results, the data were analyzed by using grounded theory techniques (Open, Axial and Selective coding). The findings indicate that there are two aspects involved in the PR process in agile development, namely human and process. The human consists of four factors, namely top management, product owner, clients, and development team. The process outlines the activities involved in the RP process, which were executed based on the identified factors. The framework emphasizes the significant elements that are worthwhile for further exploration. For example, future research may investigate the methods or techniques that can

help in selecting stakeholders and prioritization techniques as well as conducting systematic requirements prioritizing and reprioritizing process in agile development.

REFERENCES

[1] R. Popli and N. Chauhan, "A mapping model for transforming traditional software development methods to agile methodology," International Journal of Software Engineering & Applications, vol. 4, 2013.

[2] R. V. Anand and M. Dinakaran, "Popular Agile Methods in Software Development: Review and Analysis," International Journal of Applied Engineering Research, vol. 11, pp. 3433-3437, 2016.

[3] K. Beck, et al., "Manifesto for agile software development," 2001.

[4] K. Petersen and C. Wohlin, "Issues and advantages of using agile and incremental practices," Software Engineering Research and Practice, 2008.

[5] L. Cao and B. Ramesh, "Agile requirements engineering practices: An empirical study," Software, IEEE, vol. 25, pp. 60-67, 2008.

[6] I. Sommervile, Software Engineering, 9th edition, Addison Wesley, 2010.

[7] I. Inayat, S.S. Salim, S. Marczak, M. Daneva and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," Computers in Human Behavior, vol. 51, pp. 915-929, 2015.

[8] R. Harris and M. Cohn, "Incorporating Learning and Expected Cost of Change in Prioritizing Features on Agile Projects," Extreme Programming and Agile Processes in Software Engineering, pp. 175-180, 2006.

[9] V. T. Heikkilä, D. Damian, C. Lassenius and M. Paasivaara, "A Mapping Study on Requirements Engineering in Agile Software Development," in 2015 41st Euromicro Conference on Software Engineering and Advanced Applications, 2015, pp. 199-207.

[10] K. A. Khan, "A systematic review of software requirements prioritization," Unpublished master's thesis, Blekinge Institute of Technology, Ronneby, Sweden (October), URL http://citeseerx. ist. psu. edu/viewdoc/download, 2006.

[11] K. Wiegers, "First things first: prioritizing requirements," Software Development, vol. 7, pp. 48-53, 1999.

[12] F. P. Brooks Jr, The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition, 2nd Edition, Pearson Education India, 1995.

[13] S. Girase, "Comparison of various Elicitation Techniques and Requirement Prioritisation Techniques," International Journal of Engineering, vol. 1, 2012.

[14] Á. Szőke, "Decision support for iteration scheduling in agile environments," Product-Focused Software Process Improvement, pp. 156-170, 2009.

[15] A. M. Ozaki, E. Vasconcellos and M. Bengtsson, "Agile Roadmapping: How Brazilian Software Companies Evolve Their Products," in ISPIM Conference Proceedings, 2015, p. 1.

[16] T. Bebensee, I. Van D. Weerd and S. Brinkkemper, "Binary Priority List for Prioritizing Software Requirements," Requirements Engineering: Foundation for Software Quality, pp. 67-78, 2010.

[17] J. Karlsson and C. Wohlinb, "An evaluation of methods for prioritizing software requirements," Information and software technology, vol. 39, pp. 939-947, 1998.

[18] R. Thakurta, "Understanding requirement prioritization artifacts: a systematic mapping study," Requirements Engineering, pp. 1-36, 2016.

[19] F. Paetsch, A. Eberlein and F. Maurer, "Requirements engineering and agile software development," in Proceedings of the Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003, pp. 308-313.

[20] Z. Racheva, M. Daneva and A. Herrmann, "A conceptual model and process for client-driven agile requirements prioritization," in Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on, 2010, pp. 287-298.

[21] R. Razali and F. Anwar, "Selecting the right stakeholders for requirements elicitation: a systematic approach," Journal of Theoretical and Applied Information Technology, vol. 33, pp. 250-257, 2011.

[22] R. B. Svensson, T. Gorschek, B. Regnell, R. Torkar, A. Shahrokni , R. Feldt and A. Aurum, "Prioritization of quality requirements: State of practice in eleven companies," in Requirements Engineering Conference (RE), 2011 19th IEEE International, 2011, pp. 69-78.

[23] Z. Bakalova, Z. Bakalova, M. Daneva, A. Herrmann and R. Wieringa, "Agile requirements prioritization: what happens in practice and what is described in literature," Requirements Engineering: Foundation for Software Quality, pp. 181-195, 2011.

[24] T. McDaniels and M. Small, Risk analysis and society: an interdisciplinary characterization of the field: Cambridge University Press, 2004.

[25] F. A. C. Pinheiro, "Requirements honesty," Requirements Engineering, vol. 8, pp. 183-192, 2003.

[26] K. Petersen and C. Wohlin, "A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case," Journal of Systems and Software, vol. 82, pp. 1479-1490, 2009.

[27] Z. Mansor, R. Razali, J. Yahaya, S. Yahya, and N. H. Arshad, "Issues and Challenges of Cost Management in Agile Software Development Projects," Advanced Science Letters, vol. 22, pp. 1981-1984, 2016.

[28] D. S. Nguyen, "Success Factors for Building and Managing High Performance Agile Software Development Teams," International Journal of Computer (IJC), vol. 20, pp. 51-82, 2016.

[29] C. Wohlin and A. Aurum, "What is important when deciding to include a software requirement in a project or release?," 2005, p. 10.

[30] C. Wohlin and A. Aurum, "Criteria for selecting software requirements to create product value: An industrial empirical study," Value-Based Software Engineering, pp. 179-200, 2006.

[31] N. Riegel and J. Doerr, "A systematic literature review of requirements prioritization criteria," in International Working Conference on Requirements Engineering: Foundation for Software Quality, 2015, pp. 300-317.

[32] J. Favare, "Managing requirements for business value," IEEE software, vol. 19, pp. 15-17, 2002.

[33] T. Gorschek, A. Gomes, A. Pettersson and R. Torkar, "Introduction of a process maturity model for market‐driven product management and requirements engineering," Journal of software: Evolution and Process, vol. 24, pp. 83-113, 2012.

[34] A. Herrmann and B. Paech, "Practical challenges of requirements prioritization based on risk estimation," Empirical Software Engineering, vol. 14, pp. 644-684, 2009.

[35] L. Lehtola, M. Kauppinen and S. Kujala, "Requirements prioritization challenges in practice," in Product Focused Software Process Improvement, ed: Springer, 2004, pp. 497-508.

[36] R. Lutowski, Software requirements: encapsulation, quality, and reuse: CRC Press, 2005.

[37] R. H. AL-Ta'ani and R. Razali, "Prioritizing Requirements in Agile Development: A Conceptual Framework," Procedia Technology, vol. 11, pp. 733-739, 2013.

[38] M. C. Harrell and M. A. Bradley, "Data collection methods. Semi-structured interviews and focus groups," DTIC Document 2009.

[39] H. Kallio, A. M. Pietilä, M. Johnson and M. Kangasniemi, "Systematic methodological review: developing a framework for a qualitative semi‐structured interview guide," Journal of Advanced Nursing, 2016.

[40] U. Flick, Introducing Research Methodology: A Beginner's Guide to Doing a Research Project: Sage, 2015.

[41] D. L. Morgan and H. Guevara, "Audiorecording," in The Sage Encyclopedia of Qualitative Research Methods vol. 1 & 2, L. M. Given, Ed., ed. Los Angeles: SAGE, 2008, pp. 40-41.

[42] R. M. Isa, "Social Capital And Tacit Knowledge Sharing In Organisational Project In Malaysian ICT Companies," PhD Thesis, Aston Univesity, Birmingham, United Kingdom, 2008.

[43] D. Silverman, Interpreting Qualitative Data: Methods for Analysing Talk, Text and Interaction, 2nd ed. London: Sage, 2001.

[44] L. Benaquisto, "Codes and Coding," in The Sage Encyclopedia of Qualitative Research Methods vol. 1 & 2, L. M. Given, Ed., ed. Los Angeles: SAGE, 2008, pp. 85-88.

[45] A. L. Strauss and J. M. Corbin, Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, Second ed. Thousand Oaks: SAGE Publications, 1998.

[46] T. Male, Doing Research in Education:Analysing Qualitative Data, Theory and Practice, SAGE, p. 177, 2015.

[47] A. Böhm, "Theoretical Coding: Text Analysis in Grounded Theory," in A Companion to Qualitative Research, U. Flick, et al., Eds., ed London: SAGE, 2004, pp. 270-275.

[48] M. Birks and J. Mills, Grounded Theory: A Practical Guide: Sage, 2015.

[49] Version One, "10th Annual State of Agile Development Survey," http://www.agile247.pl/wp-content/uploads/2016/04/VersionOne-10th-Annual-State-of-Agile-Report.pdf, vol. (4 August 2016), 2015.