Enhancing the Process of Requirements Prioritization in Agile Software Development — A Proposed Model

Mohammad Alkandari*, Asma Al-Shammeri
Computer Engineering Department, Kuwait University, Kuwait.

* Corresponding author. Tel.: 24985826; email: m.kandari@ku.edu.kw

Manuscript submitted April 5, 2017; accepted May 10, 2017.
doi: 10.17706/jsw.12.6.439-453

Abstract: Agile software development has been used widely because it best fits projects with dynamic environments. It is an iterative and continuous process that delivers product functionalities in small increments. That enables the stakeholders to better understand their needs hence and reach their satisfaction by building the right product. Requirements prioritization (RP) is an essential phase in requirements engineering. It helps in figuring out the real requirements that are necessary to be implemented. This is important due to the limitations and constraints of any project such as budget and release date. The goal is to select the requirements that definitely maximizing customer satisfaction simultaneously meeting the restrictions such as budget and deadline. This paper proposes a model that covers the most effective RP factors based on a comparison study for existing models. The study was conducted after reviewing the literature of three models. Our model combines the most beneficial factors to enhance the RP process. Beside the model, RP technique is proposed so it can be used within the proposed model.

Key words: Agile development, requirements prioritization, requirements prioritization factors, requirements prioritization techniques.

1. Introduction

Agile software development techniques have been used widely to support the dynamic environments of projects [1]. Agile methods are collaborative and iterative which respond properly to changing customer requirements during product life cycle. The idea is to deliver small increments of the software iteratively. It is a progressive process where iterations count on stakeholder's feedback. It depends heavily on the involvement of the customers and stakeholders from the early stages of the development process. This continuous and iterative process gives an opportunity for users to better understand their problems and hence, build a system that meets their real needs. In contrast, too many changes will definitely end up with costly and undesirable reworking especially in large projects [2].

Specification or requirement engineering is the process where the functionalities, constraints, risk, and the domain of the system are defined. Requirement prioritization is a main phase in requirement engineering. It is needed because not all the requirements have equal customer satisfaction [3]. From the viewpoint of the project manager, the requirements that can be implemented easily should be implemented first [3]. On the other hand, financial manager is in favor of the requirements that cost less [3]. The main goal of requirement prioritization is to deliver functionalities that achieve customer satisfaction, fit into the project constraints, and discard requirements that are not worth to be implemented [4]. Requirements are prioritized solely or by pairwise comparison of other requirements [5]. Another study proposed a multilayered dynamic prioritization technique that could work in both agile and non-agile software development projects [6]. Further, an iterative approach has been conducted to requirement prioritization with respect to quality factors to enhance the requirements engineering process [7]. Another study demonstrated the importance of involving requirements elicitation and prioritization in agile development for quality optimization [8].

In Agile software engineering, some users are recruited to be a part of the development team. The incremental development of requirements depends on the prioritization and elicitation of requirements gathered from those users [9]. Agile highly involves customers in the development process to create the value of the product [10]. Also, Agile methodology stratifies the technical decision to the business strategy [10]. The customer should be well prepared for prioritization to take the right decisions, as well as know the steps needed to prioritize the requirements as a model. Many researches explored RP techniques, however; few were dedicated to RP models and frameworks. Existing models either are based on assumptions used in RP techniques or based on resources that are most important to the product.

Our contribution in this research with respect to the area of requirements prioritization, is trying to come up with a more general model inspired by existing models. This model best works with small, medium or large sized project since Agile methods are used. The model enhances the RP process by helping the decision-makers in their task. In addition, the collected and filtered factors are used to propose RP technique based on merging and improving two available techniques. Future work explores options to validate the effectiveness of our model.

The paper is structured as follows: Section 2 introduces important concepts of RP process. Section 3 presents an overview of three models used in our study. Section 4 demonstrates the RP factors used to compare the models. Section 5 describes our analysis of the models including the evaluation, comparison and the proposed modification. Section 6 includes the proposed RP technique. Section 7 includes the validation plan to our model. Finally, the conclusion to this study is in Section 8.

2. Background

This section introduces some basic concepts in the area of requirement prioritization. The common categories of RP techniques will be demonstrated thoroughly in the following subsections.

2.1. Requirement Prioritization

Klaus defined Requirements prioritization as the act of prioritizing requirements according to specific criteria such as its importance, cost, damage, duration, risk, and volatility [5]. Prioritization can be applied in most of software development phases and is essential due to project constraints and limitation such as time, budget, and people. Requirement prioritization helps in finding ambiguity, inconsistency, and misevaluating of requirements by analyzing elicited requirements from different angles [11]. The complexity and the time needed to prioritize requirements depend on the size of the project, the number of requirements to be prioritized, the used techniques, and other factors.

In order to start a successful prioritization process, we have to follow these steps: (1) determine the stakeholders, (2) choose the set of requirements artifacts subject to the prioritization, (3) define a prioritization criterion, and (4) choose a prioritization technique. Klaus reviewed some RP techniques such as Ranking, Top ten, and One-criterion classification [5]. While applying any technique, the process of taking decisions is affected by several factors. Factors can be the number of involved stakeholders, the degree of the understanding of the requirement, and the prioritization criteria being used. Once overall process is completed, the prioritized requirements are defined, documented, and checked for correctness.

2.2. Requirements Prioritizing Techniques

As we review the importance of the prioritization in the previous section, the process of the prioritization cannot be performed well without the support of the techniques. Many RP techniques are introduced to enhance the meaning of the prioritization. "They also guide the decision makers in their task of analyzing requirements in order to assign them numbers or symbols reflecting their importance" [12]. Even the whole techniques have the same vision they differ in many things. Some techniques assume that each requirement has a priority, and others categorize requirements by a priority level [11]. Also, they could be categorized according to the scalability, time taken [13] or sophistication (very complex, complex, ...etc.) [14]. In some projects, the project managers use a combination of techniques to enhance the prioritizing process [14]. They are no such standard classification of the techniques. In another study, the techniques are classified into: nominal scale, ratio scale and ordinal scale, ignoring the absolute scale [15]. Another categorizing classifies the techniques into: ordinal scale, ratio scale and absolute scale [11]. In addition, ordinal scale and nominal scale seem to have convergent concept with little difference. This classification has the most logical classification of the techniques. In the following subsections, the techniques will be reviewed arranged ascending according to their power [11].

2.2.1 Ordinal Scale Techniques

The ordinal technique uses discrete values rather than continuous scale, where categories are used to distinguish between the requirements based on their importance, cost, etc. [16]. It differs from the ratio scale techniques and does not show the relatively between the requirements [15]. Bubble sort is an example of these techniques. It works by comparing two requirements and promote the important one, but not in a wide range like Analytic Hierarchy Process (AHP) [15]. This type of techniques is considered as the least powerful scale [11].

2.2.2 Ratio scale techniques

The result of the ratio techniques is an ordered list of the requirements, as it shows the relative difference between the requirements [15]. Thomas Saaty introduced analytic hierarchy process in 1980 [17] and it is considered to be a ratio scale technique. It is used in many decision-making fields, not only for requirements prioritization [17]. The idea of Analytic Hierarchy Process AHP is to compute the priorities of the requirements by comparing all pairs of requirements to approximate their relative importance [15].

2.2.3 Absolute scale techniques

In absolute scale technique, a fixed value is assigned to each requirement individually, disregarding to the dependencies between the requirements [16]. Weiger's method [18] and VOP [19] are examples of these types of techniques.

3. The Requirements Prioritization Process Model

Requirements prioritization is an important aspect in requirements engineering since it has a significant impact on the process of RE in any development project [20].

Requirements prioritization is a series of steps not just one-step. Few models are built to help the requirements specialists and projects' mangers in prioritizing the requirements. Three conceptual models were selected for the comparison study, which are: "conceptual model of Agile requirements prioritization" [21], "conceptual model for client-driven Agile requirements prioritization" [22] and its upgraded version [23]. A detailed description is provided for each model in the following subsections. Figures are used to illustrate the models as shown in figures. To ease understanding the figures, uniform shapes are used. In what follows of figures, the rounded rectangles describe a process or stage, the rectangles describe the incomes and outcomes between the stages, the ovals describe the factors, and dashed lines are used to facilitate viewing the models. Terms such as factors will be explained in detail in section 4.

3.1. Model # 1

The process model "conceptual model of Agile requirements prioritization" which was presented in [21] is created using descriptive coding and analysis of the collected information through the authors' literature reviews. The model is built as an input and output of the prioritization process. The model is shown in Fig. 1. The model started with a list of the whole requirements to be prioritized according to some factors. The business value is estimated by two factors. The first factor is the work break down structure (WBS), which is a hierarchal structuring of the requirements in the project [24]. It is used as a relative value of a requirement along with the other requirements. The second factor is knowledge, information, and experience. The star symbol in the model can be any of the RP techniques described in section 2.2. The factors to be considered in the RP technique are the business value, project constraints (milestones, available resources), new information during the project and the size measurement (effort estimation). The last three factors are external sources of information that are provided to the client to take his/her decision in prioritizing the requirements. The output of the model is a list of prioritized requirements. The model is executed in every iteration of the requirements prioritization.

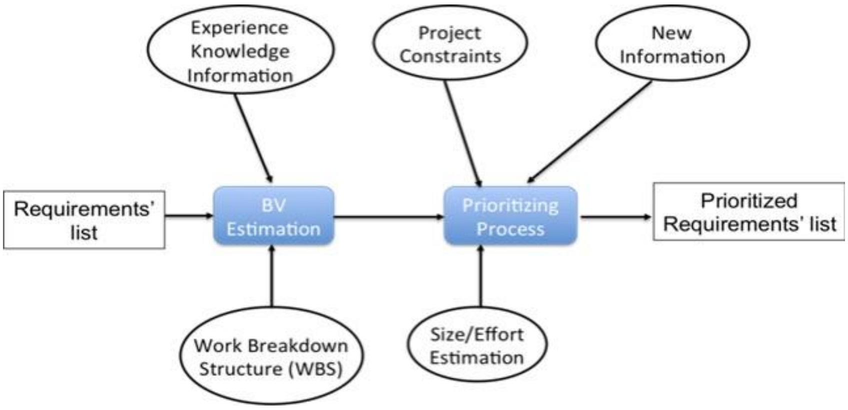

Fig. 1. Prioritization process of model # 1.

3.2. Model # 2

The conceptual models "conceptual model for client-driven Agile requirements prioritization", which were presented in [22] are created using Grounded Theory process by comparing information collected from 42 papers. The papers are selected carefully using some criteria such as adapting agile RP and reliability. The models A and B are illustrated in Fig. 2 and Fig. 3 respectively. Model A in Fig. 2 shows the entities of the agile RP process and the transmission between them, while model B in Fig. 3 works on the conceptual factors that affects the RP decisions. The two models are complementary to each other and take the viewpoints of the clients. Model A consists of 4 entities or phases: initial project backlog, prioritized project backlog, sprint backlog, and implemented requirements. Initial project backlog contains all the requirements in the beginning of the project. After applying any of the RP techniques discussed in section 2.2, the priorities are associated with each requirement in the prioritized project backlog. Small number of the highest priorities requirements in the prioritized project backlog will be forwarded to the sprint backlog. The status of these requirements will be changed to "implemented" after completing the iteration, which is the phase: implemented requirements. The requirements those are in the sprint backlog are not implemented for a reason; it will be returned back to the phase: prioritized project backlog. New information may be delivered to the projects, which force the whole requirements in the prioritized project backlog to be reprioritized. The transmission between the prioritized project backlog and the sprint backlog from the perspective of the client is expressed in Model B. This model is used to help the client focusing on the important concepts that should be considered in taking the decision. Model B consists of activities and artefacts, which are the input and the output to the activity. The arrows are the flow of the information and data. The model considers five aspects that may affect the decision of the RP. The aspects are: business value, risk, effort estimation/size measurements, learning experience, and external change. The first activity of the model is determining the business value of each requirement. Learning experience and external change aspects are considered when evaluating the business value. The second activity after computing the business value is prioritizing the requirements according to some aspects. One of the aspects is the risk. This risk is related to each requirement and different from the project's risks, such as schedule risk, budget risk or labor shortages. Furthermore, the size/effort estimation affects the decision of prioritizing the requirements. Also, External change and Learning Experience are considered in this activity as in estimating the business value. The output of this activity is a list of the whole requirements attached with each of them its priority. After that, an iteration planning is done to select the requirements to be implemented among the requirements in the prioritized project backlog. The project constraints, learning experience and external change affect the decision of this phase. The output of this activity is the sprint backlog, which contains number of the selected requirements to be implemented. In addition, learning experience affects both the project constrains and the size/effort estimation.

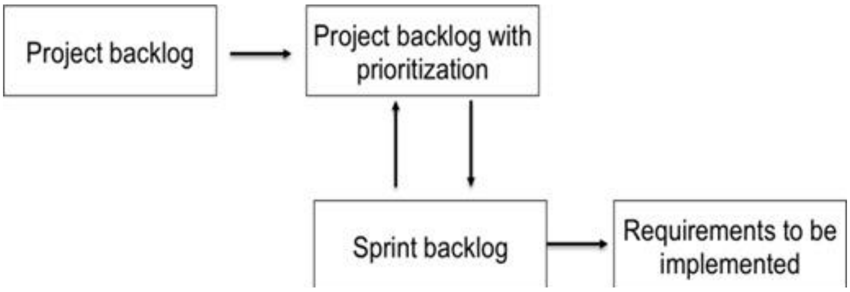
Project backlog
Project backlog with prioritization
Sprint backlog
Requirements to be implemented

Fig. 2. Model A of model # 2.

3.3. Model # 3

The model described in the previous section has been improved based on case study, which produce the third model. The model "conceptual model for client-driven Agile requirements prioritization" is demonstrated in Fig. 4. The authors came up with this model after several iterations of coding and comparing the information. They gathered information from a case study, which was conducted by questionnaire and interviews of eleven practitioners with different roles, and from ten projects distributed among seven organizations. They analyzed the data using Grounded Theory method. The model consists of a series of activities with their inputs and outputs. The value of the whole requirements has to be estimated according to some factors. First of all, the three components of the prioritization criteria, which are the business value, negative value and risk, have to be estimated for each requirement. These components depend on the project's context factor. Both of the prioritization criteria and the project context share in the value estimation activity. This activity is performed to estimate the value for each requirement individually and it is different from the business value. The remaining four aspects of estimating the value are: learning experience, external change, size/effort estimation and input from the developer. After performing this activity, any RP techniques can be used in the second activity, which is prioritizing the requirements. The same aspects used in estimating the value could be used here except the project context factor. The output of this activity is an ordered list of the requirements according to their priorities.

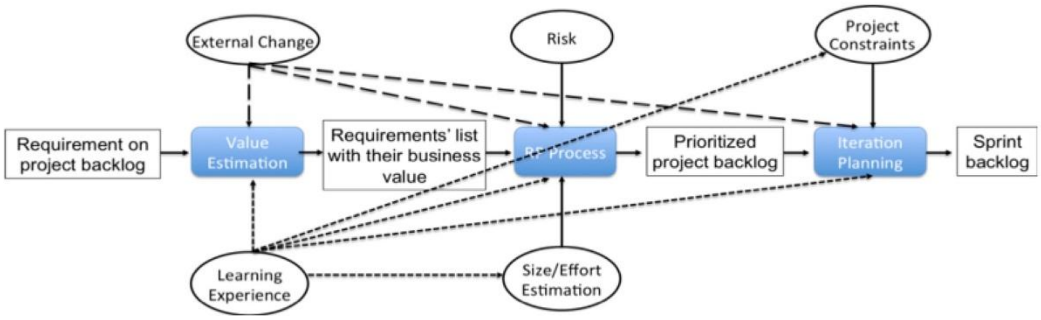

Fig. 3. Model B of model # 2.

4. Requirements Prioritization Factors

Requirements can be prioritized according to many various aspects. In this paper, we will use the terms aspect and factor interchangeably. The aspects could be attributes to the requirements [14], or some external effects such as competitive opportunities. It is a straightforward process if the prioritization is done based on one factor but, it becomes more complex when consider many factors [14]. Some Agile methods such as Extreme Programing XP prioritize the requirements with respect to one factor, which is the business value [25]. It is better to consider many factors in order to raise the likelihood of the project's success. Furthermore, some factors depend on each other [14]. In addition, not all the aspects are suited to every project. The collected factors from the different references are many. Asking the customer, requirements specialist, developer, and project manager to estimate the values for these whole factors will repel them from using the requirements prioritization process. Some factors are not very affected or could be merged with another closer factor. Moreover, the factors' roles are distributed among the process of the prioritizations. Some factors play role in the used technique, while others in the iteration-planning phase. Also, the factors can be categorized based on the stakeholders that who will evaluate it. The most important and valuable factors are collected and filtered from different literatures, and they will be demonstrated in the following subsections.

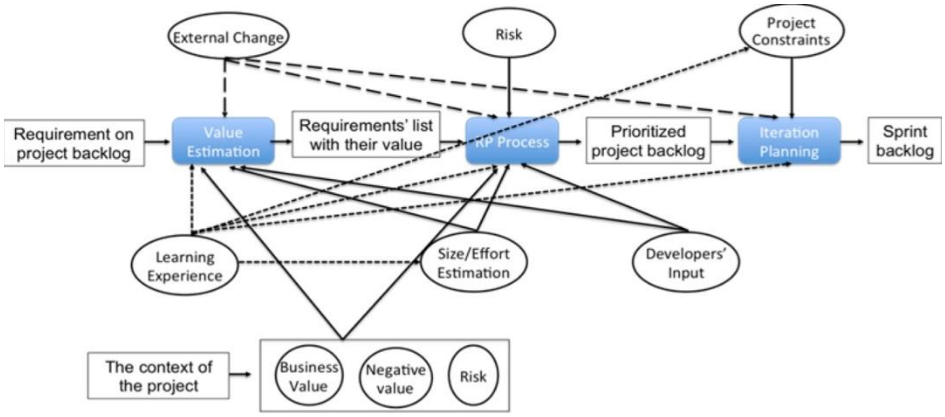

Fig. 4. Prioritization process of Model # 3.

4.1. Cost

Cost is considered as one of the critical prioritization factors. It imposes constraints where only affordable requirements can be implemented. The cost is affected by the complexity of implementing requirements. Financial recourses that each requirement needs should be covered within the project budget [26]. For a project with a limited budget, cost could be a control factor when prioritizing the requirements [26]. Cost is a major factor in evaluating the business value.

4.2. Size/Effort Estimation

Size/effort estimation is the act of predicating available and overall needed effort to deliver or maintain the product. The development team is responsible for implanting requirements as well as creating and recording effort estimates for each and every requirement. Moreover, the technical leader of the project and the architectural team lead the whole process of estimation [27]. Further, the client considers the estimated size based on the functional size [24].

4.3. External Changes

External changes are the "events that occur during the project and impact the company, the business environment or the product under development" [23]. Even though the external event occurs outside the project, it can have a significant influence on its current operations [29]. Examples of the external change are labor market competition, government policy [29], merging client organization with another organization [24] or time to market. As an example, consider the competition between the two leading companies in producing the smart phones, which are Apple and Samsung. Some requirements could be selected so that it will be implemented first because of this factor.

4.4. Importance

Importance is the key factor for the customer in building the product. Some literatures use the term "benefit" instead of importance. The entire customer can tell you "how this requirement is important", regardless of the cost and how to implement it. The customer needs to be isolated from the details and be stepped away from the technical issues of the project while evaluating the importance of a requirement. For commercial product with open-budget, importance could be the key factor of a requirement.

4.5. Business Value (BV)

In many Agile project, business value is considered to be the key requirements according to a study [10]. Many authors provided various definitions for the business value term. Barnett defined business value as "...business value, as measured in business revenue, stock price, market share, or other business metrics. Value is in the eyes of the customer..." [28]. However, the term "value" alone is closer to "benefit" and "important". But, adding the "business" to it forces us to consider other factors that affect it. An enhancement to this result is what shown in [30], "The business value an IT solution tends to be dependent on non-IT business processes. Our observations from the reviewed publications suggest that business value might well be related to other aspects and processes of the business." A more general definition for BV is presented by Patton, which said, "Business value is something that delivers profit to the organization paying for the software in the form of an increase in revenue, an avoidance of costs, or an improvement in service" [31]. Thus, BV does not only consider the importance of the requirement, but also it takes care of its cost. Removing the confusion between the earned business value and the business value is important here. Earned business value is the ration of what will be delivered and the cost spend on it [30]. Business value is influenced by combination of many factors, neither of them is a technical factor. Hence, evaluating the business value for a requirement is considered as a management task more than technical task. Estimating a requirement's business value is affected by time, cost and surrounding environment behind its importance. The main factors that affect the business value are the importance of the requirement, the external change and the cost of implementing it. A mathematical equation will be proposed to set the business value of a project in section 6.

4.6. Volatility

Volatility can be defined as "the tendency of requirements to change over time in response to the evolving needs of customers, stakeholders, organization, and work environment" [32]. The requirement that tend to be changeable will affect project's implementation either in terms of cost, time, or both. Volatility of a requirement is considered as part of risk factors [14].

4.7. Learning Experience

In some projects, the development team needs to learn a new programming language or a study new technology. Some requirements require using a new programming language in their implementation that the development team did not use it before. Learning a programming language or a new technology requires time, cost or both, which will affect on the implementation time and budget. Based on that, a learning experience factor is considered as a risk that could be avoided when choosing the requirements for the next iteration.

4.8. Risk

Risk is the probability of failing the project. Risk is a normal event in any project. Projects' risk could be schedule risks, budget risks, etc. [11]. However, what we consider in the requirement prioritization is another type of risk. As the project has risks, the requirement also has different risks. "Requirements risks are those which are associated directly to specific requirements" [33]. The term risk is a very general term, which may confuse the analysts when they evaluate the priority of the requirements. Another notice about risk is that not all the risks have negative impacts. Some risks can actually benefit the requirements and are known as positive risks or opportunities [34]. Most of the RP techniques evaluate the risk with a negative scale [22]. It is obvious that those techniques ignore the impact of the positive ones. The positive risks will give the team an experience for the upcoming projects. In our case, we are interested in learning experience as positive risk, since it is already has been used in the three models. Learning experience is considered as a risk because the developer will need more time to learn a new programming language or understand a new technology. At the same time, the developer will gain an experience in this field, which is the positive side of the risk. As a result, we should not deal with risk as a single factor. Instead, it should be evaluated early in terms of the most valuable factors mentioned earlier, which are changeability and learning experience. The first factor will maximize the probability of the risk, while the second one will minimize it.

4.9. Dependencies

A set of requirements depends on other requirements. The likelihood of having dependency increases in the iterative development process [26]. The best way of noting and describing the dependencies is using the Work Breakdown Structure (WBS). A Difference is found between the chronological and architectural dependencies. More specifically, "A chronological dependency from clients' point of view could be different from chronological dependency from developers' perspective, as the first one is based on business reasons– and – processes, while the second one has technical reasons and might require completely different sequence of implementing features" [21]. Hence, the customer can express the business dependency between the requirements via the importance factor. Also, the dependency factor is important in deciding the number of requirements to be implemented, so that no children requirements will be selected without their parent requirements. Many strategies are proposed to handle the dependency issue. One of these strategies is grouping the dependent requirements, in which and each group is totally independent of the others [16] as well as gives the option to the project manager to take them all or leave them all.

4.10. Project/Iteration Constrains

Project constrains contain the information that are related to the project such as duration, release date and available resources. Also, the iteration velocity is considered in this factor. The iteration velocity could be changed from iteration to other based on management. The change in the velocity of the iteration i.e. duration, will affect the number of the requirements to be implemented [35]. The project constrains is part of the iteration planning stage [22]. Estimating the project and iteration constraints and picking the requirements based on them in the iteration planning is the job of the project manager job. Table 1 summarized the factors that have been discussed in the subsections 4.1 to 4.10, their usage and perspective.

Table 1. The Used Factors in the Proposed Model, their Usage and Perspective

Factor	Stage	Perspective
Cost	BV estimation	Project manager
Size/effect estimation	RP process	Project manager
External change	BV estimation	Customer
Importance	BV estimation	Customer
BV	RP process	Requirements specialist
Volatility	Risk estimation	Requirements specialist
Learning experience	Risk estimation	Developer
Risk	RP process	Requirements specialist
Dependencies	RP process/iteration planning	Requirements specialist
Project/iteration constraints	Iteration planning	Project manager

5. Models' Analysis

5.1. Models' Evaluation

In this section, we evaluate the three models based on the prioritization factors discussed in section 4. The stages of the models have also been analyzed.

5.1.1 Model # 1

The model performs the requirements prioritization process in two stages. The first stage is evaluating the business value and the second stage is prioritizing the whole requirements using any RP techniques. The weak point of this model is that it did not specify how many requirements will be implemented in the iteration and what the criteria are of selecting this number of requirements. The BV is estimated with respect to two factors, which are the WBS factor and knowledge, information, and experience factor. We noticed that, the customer should not estimate the first factor and the second factor is vague. Moreover, the BV in this model misses the cost and importance factors. The factor called "new information during the project" used in the prioritization process is ambiguous. The model creator should determine if the information is internal or external.

5.1.2 Model # 2

The model applies the requirements prioritization process in three phases. The weak point of this model is considering the risk without specification. Also, the model ignores the role of importance and cost in estimating the BV. However, it judges the BV based on the learning experience, which is out of the customer scope.

5.1.3 Model # 3

The model applies the requirements prioritization process in three steps. As unusual, this model assigns values to the requirements based on the BV. The model estimates the BV, negative value and risk based on general information about the project such as the size of the project or the number of clients in the organization. This way of estimation will lead to unexpected and incorrect priority for the requirements. Moreover, the same aspects used to estimate the value will be used again to estimate the priority for the requirements, except one factor, which is the project context. As a result, the total priority of a requirement will not differ a lot from the value of this requirement. In addition, the model defines extra factor called "input from the developer" such as the importance of the project to the organization. This factor should have a general effect for the project not necessarily for requirements.

5.2. Models' Evaluation

In this section, we compare the three models according to the factors and stages discussed in section 4. Table 2 shows the Decision Matrix (DM) of the models versus the factors. Decision Matrix (DM) was used to identify, analyze and rate the relationship. The following steps were conducted to form our DM. First, we defined the alternatives, which are the models. Second, we identified decision/selection criteria, which are the stages and the factors. Third, assigning weights for the criteria. In this research, we care about the occurrence of the stages and factors, where the weight for each of them is not covered here. We will use the colored cell instead of the weight [36].

1. If the stage/factor is fully existed/implemented in the model, the cell will be colored by dark blue.
2. If the stage/factor is existed/considered in the model but used incorrectly, the cell will be colored by light blue.
3. If the stage/factor does not exist in the model, the cell will be colored by white.

Table 2. DM of the Models vs. the Factors

Criteria	Models	Model # 1	Model # 2	Model # 3
		■	■	▒
	Risk Estimation			
	RP Process	■	■	▒
	Iteration Planning		■	■
	Importance			
Business value	Cost			
	External Exchange		■	■
Size/ Effort estimation		■	■	▒
Project/ Iteration constraints		▒	■	■
Dependencies		▒		
Risk	Learning Experience	▒	▒	▒
	Volatility			

5.3. Suggested Improvement

Based on our observations, we will propose some improvements for the best model among the three evaluated models. According to the used comparison tool in section 5.2, the suggested improvement will be applied to model #2. Model # 2 has the maximum number of dark blue cells. So, this model is the easiest one among the others to be modified. The new improved model should contain all the factors explained earlier in section 4. Consequently, a new activity will be added to work simultaneously with the BV estimation activity. The new activity is used to estimate the value of the risk based on the two factors, which are changeability and learning experience. The proposed improvement is shown in Fig. 5.

6. The Proposed Technique

As described in section 2.2, the absolute scale techniques are the most powerful techniques in requirements prioritization, “With higher levels of measurement, more sophisticated evaluations and calculations become possible” [11]. Absolute scale techniques are suitable with any number of requirements, where comparisons are not needed.

Wieger’s method [18] is an example of this technique. We can conclude the concept that the founder followed from the formula being used in the technique. As shown in equation 1 (Wieger's Formula), the formula is built based on the ratio concepts, the factors that maximize the priority are placed and summed in the numerator and the factors that minimize the priority are placed and summed in the dominator. Wieger includes benefit and penalty as factors that increase the probability of selecting the requirement. On the other, it includes the cost and risk as factors which increase the probability of discarding the requirement. In equation 1, the subscript i is denoted to the requirement, any requirement. The formula will be used as many requirements as we have.

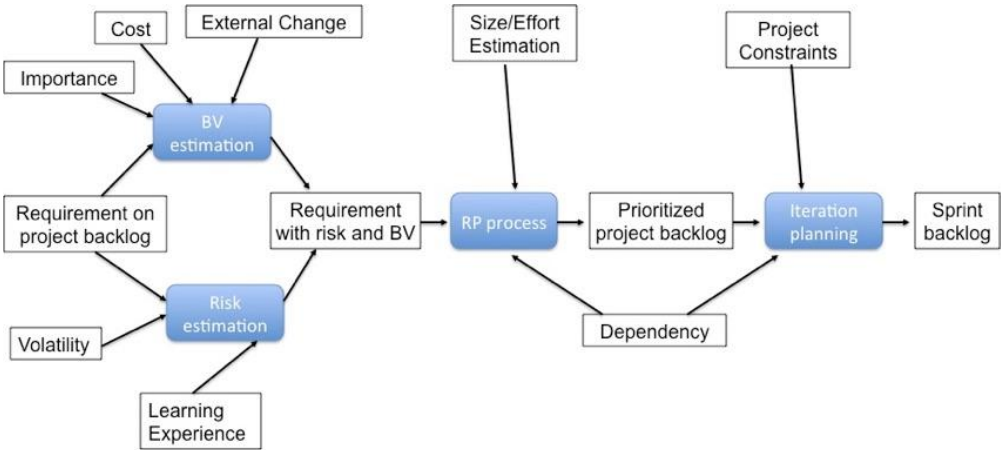

Fig. 5. The proposed model with the whole factor.

$$Priority\ of\ R_i = \frac{benefit_i + penalty_i}{cost_i + risk_i} \qquad (1)$$

In addition, he gave the project manager the ability to assign the factors weight based on his estimation. As a default value, he assigned the benefit and cost factors weights that are double the penalty and risk weights, respectively [18].

Another absolute scale technique is VOP. “VOP evaluates requirements according to their impact on specific business values an organization recognizes” [19]. It was established and used by an organization called TBI (Technology Builders Incorporate) to be used by various stakeholders. For business values, they limited the categories to fit in the size of their organization. In addition, these business values are dynamic and can be changed depending on the project and the organization [19]. Moreover, the risk factors are changeable. The VOP formula used to calculate the priority of a requirement is shown in equation 2. In equation 2, w_i is referred to the weigh assigned to the factor, either business value or risk, v_i referred to the business value factor and r_i to the risk factor.

$$Priority\ of\ R_i = \sum w_i * v_i - \sum w_i * r_i \qquad (2)$$

VOP is built to evaluate the priority, as it is the difference between the factors that improve the priority i.e. business values, and the factors that decrease the priority, i.e. risks. Weights are assigned to every factor based on the estimation of the stakeholders.

The two previous absolute techniques use some of the factors that we explained in section 4, not all of them. In addition, Wieger use the risk as a single factor, while VOP use it as compound of unspecified factors.

The two techniques can be merged and customized to suit the whole factors discussed in section 4. The proposed technique uses the ratio and difference to evaluate the priority.

As we conclude in section 4.4, business value is affected by many factors, which are: importance, cost and external change. Importance and external change increase the probability of implementing the requirement in iterations. On the other hand, the requirement with low cost is preferable to be implemented earlier. Thus, a ratio will be used to scale the importance and external change with the cost. A requirement that has high importance, high external change and low cost will probably be implemented in the coming iteration.

Although business value is a good estimation of the requirement's priority, risks should also be considered also. Risk is explained in section 4.8. To have a good estimation of risk, specification is needed. Two risks factors are considered in this technique, volatility and learning experience. A requirement with low volatility and high learning experience is preferable more than a requirement with high volatility and low learning experience. We are looking for the risk to be low; this is achieved by placing the factor that increases the risk in the numerator, and the factor that decreases the risk in the denominator, i.e. positive risk.

Based on that, VOP is used again, so the difference between the business value and composed risk is the priority of a requirement. The proposed technique use equation 3, where i denoted the requirement, any requirement.

$$Priority\ of\ R_i = \sum \frac{Importance_i + External\ Change_i}{Cost_i} - \sum \frac{Volatility_i}{Learning\ Exprience_i} \quad (3)$$

Following the criteria used in the study "Selecting an Appropriate Framework for Value-Based Requirements Prioritization" to evaluate the different techniques of requirements prioritization, weights could be assigned to the factor based on the project type [37]. For example, a project with a very limited budget, a large weight should be assigned to the cost. In addition, to simplify viewing the requirements corresponding to their values, using a table is recommended. Further, this table is useful to filter the requirements according to a specific factor. Using this feature, we can ignore the result of the formula expressed in equation 3 to implement the requirements that have risks lower than certain value or the requirements that do not cost more than the particular value [11].

7. Validation Plan

A future research is required to validate the proposed improvements for both the model and the technique. So, we plan to use different methods other than the used one, which is based on gathering the information from literatures. It is necessary to have interviews with experts in software engineering especially in Agile development, which can support in identifying more issues, stages or factors to the proposed model. Also, case studies from different organizations will be useful, where different types of stakeholders can participate and contribute with us to prove our logical analyzing of the effective factors. As the development and marking fields are increasing, new factors could be initiated which may affect the decision of prioritizing the requirements. The proposed model and technique should be judged based on how it facilitates and enhances the RP process for the clients.

8. Conclusion

Requirement prioritization is an essential process for any developed project. RP factors and techniques are well known and being used widely. In contrast, there are few concerns about a comprehensive framework or models enhance this process. RP in Agile needs more focusing since Agility involves stakeholders heavily and deal with intensive changing in requirements. The goal is to achieve as much as business value in the software development process. Our contribution to this area is to find a more general model that enhances RP process effectively. A comparison study was conducted to come up with a model that includes possible RP factors that affect RP process significantly. This study was based on the exiting RP models from the reviewed literature. The defects and the missing items of the reviewed model where taken in to account when designing our model. Beside the model, a proposed RP technique of type absolute scale is introduced based on improving two existing techniques to include all the factors. The future work explores options to validate and apply this model.

References

[1] Iqbal, R., & Abbasi, A. (2014). Requirement engineering process in agile software development: Review. *Research Journal of Computer and Information Technology Sciences, 2(5)*, 1–15.

[2] Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and agile software development. *Proceedings of the Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprise* (pp. 308–313).

[3] Ritu, & Gill, N. (2012). A comparison among various technique to prioritize the requirements. *International Journal of Computer Science & Management Studies, 12(3)*, 601–607.

[4] Robertson, S., & Robertson, J. (2014). Measuring the requirements process: Getting requirements right.

[5] Pohl, K. (2010). Requirements engineering: Fundamentals, principles, and techniques.

[6] Gapta, V., Chauhan, D. S., Dutta, K., & Gupta, C. (2013). Requirement reprioritization: A multilayered dynamic approach. *International Journal of Software Engineering and its Applications, 7(5)*, 55–64.

[7] Singh, D., & Jatain, A. (2013). An interactive approach to requirements prioritization using quality factors. *International Journal in Foundations of Computer Science and Technology, 3(6)*, 25–40.

[8] Asghar, A., Bhatti, S., Tabassum, A., Sultan, Z., & Abbas, R. (2016). Role of requirements elicitation & prioritization to optimize quality in scrum agile development. *International Journal of Advanced Computer Science and Applications, 7(12)*, 300–306.

[9] Sommerville, I. (2015). *Software engineering* (10th ed.). Pearson.

[10] Racheva, Z., Daneva, M., Sikkel, K., Herrmann, A., & Wieringa, R. (2010). Do we know enough about requirements prioritization in agile projects: Insights from a case study. *Proceedings of the 18th IEEE International Requirements Engineering Conference* (pp. 147–156).

[11] Aurum, A., & Wohlin, C. (2005). Requirements prioritization — Engineering and managing software requirements.

[12] Karlsson, J., Wohlin, C., & Regnell, B. (1998). An evaluation of methods for prioritizing software requirements. *Information and Software Technology, 39(14-15)*, 939–947.

[13] Khari, M., & Kumar, N. (2013). Comparison of six prioritization techniques for software requirements. *Journal of Global Research in Computer Science, 4(1)*, 38–43.

[14] Fernandes, J. (2009). Requirements prioritization. Retrieved March 19, 2016 from: http://alba.di.uminho.pt/por/content/download/2663/13999/file/unit3-prioritization.pdf

[15] Vestola, M. (2010). A comparison of nine basic techniques for requirements prioritization. Retrieved March 23, 2016 from: http://www.mvnet.fi/publications/software_development_seminar.pdf

[16] Herrmann, A., & Daneva, M. (2008). Requirements prioritization based on benefit and cost prediction: An agenda for future research. *Proceedings of the 16th IEEE International Requirements Engineering Conference* (pp. 125–134).

[17] Ahl, V. (2005). An experimental comparison of five prioritization methods – Investigating ease of use, accuracy and scalability. Master Thesis, Blekinge of Institute of Technology, Sweden

[18] Wiegers, K. (1999). First things first: Prioritizing requirements. *Software Development, 7(9)*, 24–30.

[19] Azar, J., Smith, R., & Cordes, D. (2007). Value-oriented requirements prioritization in a small development organization. *IEEE Software, 24(1)*, 32–37.

[20] Batra, M., & Bhatnagar, A. (2016). Requirements prioritization: A review. *International Journal of Advanced Research in Science, Engineering and Technology, 3(11)*, 2899–2904.

[21] Racheva, Z., Daneva, M., & Buglione, L. (2008). Supporting the dynamic reprioritization of requirements in agile development of software products. *Proceedings of the 2nd International Workshop on Software Product Management* (pp. 49–58).

[22] Racheva, Z., Daneva, M., Herrmann, A., & Wieringa, R. (2010). A conceptual model and process for client-driven agile requirements prioritization. *Proceedings of the 4th International Conference on Research Challenges in Information Science* (pp. 287–298).

[23] Racheva, Z., Daneva, M., & Herrmann, A. (2010). A conceptual model and process for client-driven agile requirements prioritization: Results of a case study. *Proceeding of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Management*.

[24] Hillson, D. (2003). Using a risk breakdown structure in project management. *Journal of Facilities Management, 2(1)*, 85–97.

[25] Firesmith, D. (2004). Prioritizing requirements. *Journal of Object Technology, 3(8)*, 35–47.

[26] Iqbal, A., Khan, F., & Khan, S. (2009). A critical analysis of techniques for requirement prioritization and open research issues. *International Journal of Review in Computing, 1*, 8–18.

[27] Siddiqui, S., Beg, M., & Fatima, S. (2013). Effectiveness of requirement prioritization using Analytical Hierarchy Process (AHP) and Planning Game (PG): A comparative study. *International Journal of Computer Science and Information Technology, 4(1)*, 46–49.

[28] Barnet, L. (2007). Agile projects must measure business value. Retrieved from: http://www.agilejournal.com/content/view/211/76/

[29] Education Portal. How internal and external factors drive organizational change. Retrieved July 13, 2016 from: http://education-portal.com/academy/lesson/how-internal-and-external-factors-drive-organizational-change.html

[30] Racheva, Z., Daneva, M., & Sikkel, K. (2009). Value creation by agile projects: Methodology or mystery? *Proceedings of the International Conference in Product-focused Software Process Improvement* (pp. 141–155).

[31] Patton, J. (2008). Ambiguous business value harms software products. *IEEE Software, 25(1)*, 50–51.

[32] Nurmuliani, N., Zowghi, D., & Powell, S. (2004). Analysis of requirements volatility during software development life cycle. *Proceeding of the 2004 Australian Software Engineering Conference* (pp. 28–37).

[33] Modern Analyst. Requirements risk management. Retrieved July 28, 2016 from: http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/212/Requirements-Risk-Management.aspx

[34] Right Hub PM. How to respond to positive risk. Retrieved August 7, 2016 from: http://www.brighthubpm.com/risk-management/48400-how-to-respond-to-positive-risks/

[35] Logue, K., & McDaid, K. (2008). Handling uncertainty in agile requirement prioritization and scheduling using statistical simulation. *Agile Conference* (pp. 73–82). IEEE.

[36] Triantaphyllou, E. (2000). Multi-criteria decision making methods: A comparative study.

[37] Kukreja, N., Boehm, B., Payyavula, S., & Padmanabhuni, S. (2012). Selecting an appropriate framework for value-based requirements prioritization. *Proceedings of the 20th International Requirements Engineering Conference* (pp. 303–308)

Mohammad A. Alkandari is an assistant professor of computer engineering at Kuwait University, Kuwait, where he has been on the faculty since 2012. He received his Ph.D. degree in computer science at College of Engineering from Virginia Polytechnic Institute and State University (Virginia Tech). He was the director of the office of engineering education technology at Kuwait University, College of Engineering and Petroleum for 3 years. He is currently the coordinator of software and systems engineering research group at Computer Engineering Department. He is a researcher in software engineering, requirements engineering, software project management, software quality assurance, privacy and data protection, and human-computer interaction.

Asma Z. Al-Shammeri is currently working as a senior computer engineer in IT Department at Kuwait National Assembly. She graduated from Kuwait University, Computer Engineering Department in January 2010, and got her master degree in January 2014. She is interested in software engineering, especially in requirement engineering.