An-Najah National University

Faculty of Graduate Studies

AGILE-SCRUM BACKLOG CHANGES OPTIMIZATION IN SOFTWARE ENGINEERING ORGANIZATIONS

By

Azhar Omar Ghanem

Supervisor

Dr. Ahmed Awad

This Thesis is submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering Management, Faculty of Graduate Studies, An-Najah National University, Nablus, Palestine.

2022

AGILE-SCRUM BACKLOG CHANGES OPTIMIZATION IN SOFTWARE ENGINEERING ORGANIZATIONS

By

Azhar Omar Ghanem

This Thesis was Defended Successfully on 21/03/2022 and approved by

Dr. Ahmed Awad	
Supervisor	Signature
Dr.Yousef Daraghmeh	
External Examiner	Signature
Dr. Amjad Hawwash	
Internal Examiner	Signature

Dedication

I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears. My sisters and friends who have never left my side and are very special.

Acknowledgements

I would like to express my sincere gratitude to several individuals and organizations for supporting me throughout my study. Most importantly, I wish to express my sincere gratitude to my supervisor, Dr. Ahmed Awad, for his enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped me tremendously at all times in my research and writing of this thesis.

My sincere thanks also goes to my organization at work, they have been a valuable part of this research and helped me get answers to questions arose while completing this study also for their patience and understanding during the past year of effort that went into the production of this research.

A special thanks goes to Engineering Management department and their help during years of studying. Specially Dr. Mohammad Othman for his guidance through the thesis process and cooperation.

Further, yet importantly, sense of respect goes to my father and family for their strong support as well as regular encouragement in every step to make me in present stage. Similarly, my friends are also subjects to special thanks for their inspiration and cooperation in my study.

Declaration

I, the undersigned, declare that I submitted the thesis entitled:

AGILE-SCRUM BACKLOG CHANGES OPTIMIZATION IN SOFTWARE ENGINEERING ORGANIZATIONS

I declare that the work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's Name:	**Azhar Omar Ghanem**
Signature:	
Date:	**21/03/2022**

List of Contents

List of Tables

List of Figures

AGILE-SCRUM BACKLOG CHANGES OPTIMIZATION IN SOFTWARE ENGINEERING ORGANIZATIONS

By
Azhar Omar Ghanem
Supervisor:
Dr. Ahmed Awad

Abstract

Background: software engineering requirements are translation of the product needs or features requested by customers and stakeholders. Due to the evolution of technologies, these requests are always changing. These requirements should be managed carefully, in order to help the customers and organization achieving their goals. Agile-Scrum has been introduced as project management methodology that focuses on the customers first and team communications rather than requirements documentation.

Objectives: scrum methodology is all about delivering requests (i.e., backlog items) faster to customers and accepting changes to these requests, yet less reliable estimation of resources (e.g., time, money, manpower, etc.) exists. Backlog items depend on a lot of uncertainties inherited in the backlog management process using the scrum. Therefore, a need to reduce the changes in the backlogs is a must, by developing a new prioritization model.

Methodology: unstructured interviews were conducted with five product owners from two local organizations implementing the scrum to identify the factors that affect the process and lead to backlog changes. A prioritization model was developed to help software engineering organizations manage their backlog items effectively, and to minimize the losses due to continuous backlog changes

Results: results showed that the priority changes are the most dominant parameter that affects backlog changes. Unlike previous models in the literature, interdependencies between items, number of action words in the item description, and assigned developer features were found to significantly affect the priority ranking. In effect, the developed model serves as an effective tool to assign priorities during the agile-scrum planning phase for product owners.

Conclusion: we have validated our proposed model by having a case study in one organization that implements scrum and a benchmark project. The results pointed that change in time was reduces because of general prioritization methods used currently in the studied organization. In addition, our proposed model showed good impact on backlog prioritization in agile – scrum environment and help the product owners with their tasks related to backlog management.

Keywords: Agile, Scrum, Backlogs, Backlog Changes Management, Prioritization.

Chapter One
Introduction

In this chapter, an overview about the project management methodologies and scrum in specific.

1.1 Background

Leadership skills and management book, 2016 defined the project as a "Sequence of activities undertaken to accomplish a specified outcome at a defined time using a defined set of resources". The project key elements is the people (team) in addition to the project manager, each team member will has its own activities and takes during the project life cycle while the project manager is the connection bridge with the project sponsor who will give the team the approval or rejection of the project or part of the project.

Coping up with the speed of technological inventions and revolutions, in addition to the globalization and reaching up the industry competitors, especially software development companies, keeping up with the customers' needs and making daily changes on the requirements these companies offer to their clients is moving fast! And therefore, it's noticeable that some of the growing and big companies showing struggles to keep up the pace and to satisfy their clients (Sawsant 2020).

Both (ACM), (IEEE) and (Garmisch 1968) define the software engineering: "systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software to optimize its production, support, and quality [ISO/IEC 2382 2015.]

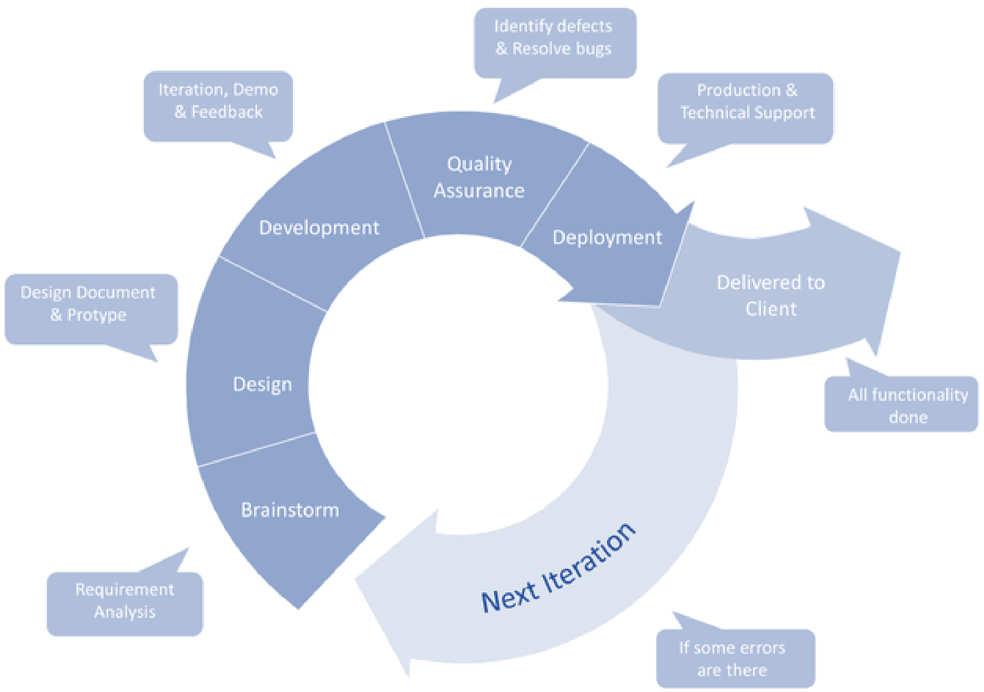
1.2 Introducing Agile

In the late 1690s until mid-1980s, there was the "Structured Method Era", this was the beginning of the software development era and it was about the codes of the program and the software system to be reflected into functions and then all the data should be saved. Most popular methods at that time SADT (Structured Analysis and Design Technique) and SA/SD (Structured Analysis/Structured Design). Later in early 1980s until early 2000s, a new era came up, which was named the "Component Era", unlike the before-mentioned era, the component era considered the software system as a set of interacting elements or components not functions or saved data. Currently since late 1990s the most popular trend that was spread all over the world, which is Agile or agility, and this era is called "The Agile Method Era", it refers to the software systems as a set of technical and people-related practices. Many methods were added under the umbrella of Agile like, Extreme Programming (Beck 1999), Scrum (Schwaber 1995), Kanban, Lean, FDD (Feature-Driven Development) (Palmer and Felsing, 2002), Crystal, DSDM (Dynamic Systems Development Method) (Stapleton 1999) and RAD.

Given a clearer definition about Agile/Agility; it is a way of managing the ability to act during creation or responding to the changes that might happen during the process, and reach best success can be done during the uncertain, risky and turbulent environment.

Figure 1

Agile development methodology for project management (Introduction To Agile Methodology)



1.2.1 Agile Manifesto

Back in 2000 the agile manifesto organization published four principles of Agile that will help creating and doing the software development better. In order to help the surrounding environment of achieving the best help they can get to ensure that the software they are managing is developed with less efforts but great value.

- Individuals and interactions over processes and tools

- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The value is divided by both parts of these sentences with more focus on the left side without neglecting the right part (Manifesto for Agile Software Development, 2022)

When trying to implement agile project management methodologies to companies, process needs a lot of time and effort to change and eventually be successful, working with agile is different relating the traditional managerial structure and traditional management of projects (Parente 2015). Each organization has its own way to manifest and organize their structure to meet the methodology followed, it may need to align three parts; the agile culture, team levels and management.

Through the years and the new leading way of product management, especially in the agile software development, many methods under the agile product management are now in practice (Abrahamsson et al, 2003) According to "State of Scrum 2020", Scrum Methodology is widely used as an agile project management methodology, around 56% of the organization are using it and followed by hybrid (multi-methodologies) with 14%, and the rest is divided into smaller percentages of other methodologies like SCRUMBan, Hybird, Kanban, Lean, XP and Iterative development (Ereiz & Music, 2019).

Software engineering requirements are translation of the product needs or features requested by customers and stakeholders. In other words, named "backlogs".

Backlogs are a list of items representing the customers' needs and product specifications and features. This list is studied and prioritized by the product owner with help from the stakeholders and the scrum team. However, the product owner is responsible for managing what items that comes first by its value, as the scrum is all about the most valuable work first" (Rubin, 2013). Furthermore, the items description, listing, availability, value and estimation. Therefore, the backlog item is explained clearly in the description to show its completeness when it's done, by all the needed information that is already filled (Scrum guide 2020).

Backlogs are known as items or user stories; both of these terms will be used in this research interchangeably.

Due to the evolution of technologies, these requests are always changing. These requirements should be managed carefully, in order to help the customers and organization achieving their goals. Agile-Scrum has been introduced as project management methodology that focuses on the customers first and team communications rather than requirements documentation.

1.3 Introducing Scrum

Scrum was first introduced in 1990s and it was referred as a management and control process (Schwaber & Beedle 2002), later it was defined as "it an iterative, incremental approach to optimize predictability and control risk". (Schwaber & Sutherland 2017)

Scrum was defined as an agile process that manages a project and allows us to focus on delivering the highest business value in the shortest time. An iterative framework within which people can address complex adaptive problems. Schwaber & Sutherland, mentioned in their Scrum Guide, that's it is closer to a framework than a procedure with defines steps and techniques. And therefore, the team may apply the needed techniques with variety of activities included. Moreover, this framework will help the product management efficiency by improving:

- Product.
- Team.
- Work environment.

Although scrum framework seems to be encouraging and easy to implement, this is a mistake for some to be considered. Scrum does not give you any specifications or clear questions on how to solve problems, since the main value of the scrum to continuous improvement, it encourages the teams to develop their own questions and expanding their imagination to generate the right questions and by then finding the answers (Rubin, 2013).

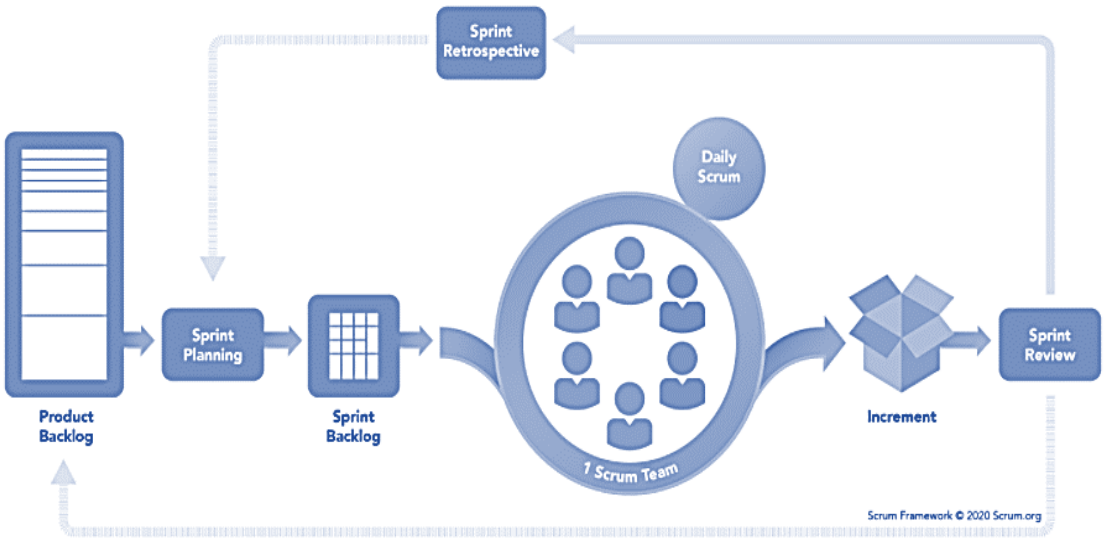
Following the Scrum components and processes is embracing the principles of the Agile Manifesto and its values; where it assists all parts of the organizations, from individuals to businesses. The journey any business managed by scrum will increase its efficiency and effectiveness through the communication and collaboration. Also reducing errors, unwanted functionalities and most importantly, helping organization to delivering the needed work frequently, and coping up with the business requirements changes. As other methodologies of agile, maintaining the customer satisfaction is the goal to achieve. It's mentioned as the first principle of the Agile Manifesto start with "Our highest priority is to satisfy the customer" (Miller and Larson, 2005; Moe et al., 2010). Despite the great advantages the scrum supports its organizations, its pitfalls could be disastrous to the projects (Common Scrum Pitfalls, 2012), like: excessive up-front planning, focuses on the tools more than the team and the processes, daily scrum event is used for solving problems only and not for keeping updated about the work, imposed deadlines and resources; all beforementioned are reasons of not following the scrum, this is defined as only restructure of the framework improperly. Organizations could restructure the scrum and its event based on the company vision and mission. A healthy scrum team should have the authority to decide development and needed work, teams should be distributed team-members, they can stretch the sprints goals. In terms of the process, they should not to focus on individuals over the team, compromising over quality, ceasing to learn and redesigning the teams.

1.3.1 Scrum Methodology

Scrum has many titles to be described as a methodology, process (Schwaber & Beedle, 2002), practices or a framework (Schwaber & Sutherland, 2020). The last update was to consider it as both an agile process and an iterative framework (Jacobson et al., 2019). Scrum is mainly followed to short iterations; it varies from one to four weeks while others may have it as eight weeks iteration. These iterations are called sprints. During these sprints, the requests of the new developments or enhancements to the product or the service are prioritized and divided into these sprints. After each sprint, a release is being launched with the new developments added without affecting the original functionality. The soul of the scrum team is a Product Owner (PO), as s/he represents the client during the implementation of the Scrum. Also, s/he is responsible for the maximum business value delivered to the clients (Schwaber & Sutherland, 2020) as well as the product backlog management.

Figure 2:

Scrum framework process (Scrum framework poster, 2020)



1.3.1.1 Scrum Team

The scrum team is defined to three parties, each with their own roles and responsibilities and they are defined in details in the followings.

1.3.1.1.1 Product Owner

The product owner is the soul of the scrum team, as the PO represents the client in the scrum. The PO is responsible for maximizing the business value delivered to the clients (the product) and the work that is going to done by the development team to complete the needed requirements during the sprints. The PO, creates a list of backlog items (user stories) to be done for the product with the help from the scrum master. At later stages before the sprint, the PO prioritize these user stories, to know which of them is going to be developed and when (at what sprint). Then, the PO collects the feedback from the clients and the development team in order to assess the progress and confirm delivering the items to the clients. At this stage, if the surroundings affected the items and changed them, then the PO could decide what to keep, what to be added and what to be dropped or delayed (Schwaber, 2004). The PO has the authority to make these decisions unlike the scrum master. Furthermore, the PO could do this backlog management by their own, or leave the call to the development team to do it. Some think that the product owner is leading or managing the overall scrum processes, nevertheless, Unger-Windeler & Schneider, explained in a study they made in 2019 about the scrum team expectations on the product owner role, results found out that the scrum team, expects the product owner to be the one responsible solely, to manage the backlogs by prioritizing and writing the users stories by herself/himself. That was confirmed by the Scrum guide 2020; it explains that the PO remains accountable for the backlog items management.

1.3.1.1.2 Development Team

Scrum team consists of developers and quality assurance engineers and designers. The development team should be typically small, from five to nine members in general. Furthermore, Beedle (2002) mentioned that the best size of the team members should be seven, large number of members may lead to complexity in scrum and managing the projects and the team members, leads to efficiency. In addition to the size, the team member should be cross functional and have the appropriate skills to confirm the "done" increments are delivered to the client and will be done before the release version. There are other different operators that define the team and how fast they will move to effectiveness, "like geographical spread, frequency and duration of meetings, synergy of team types, stability of team membership, external influences and time pressures and the nature of its activities" (Ereiz & Music,2019). The development team will choose their prioritized preferred backlog items in the sprint planning meeting, and by committing the work, this does not mean that this backlog item, or increment will be done in that sprint.

The development team key personality treat is that they should be self-organized and they know what is the best to be done and how they are going to do it, so they can decide the estimated hours needed for a specific backlog item, and later once they perform the real work, these estimated hours are compared with the actual effort/hours needed to complete that specific item. They are aware enough to do their own decisions in order to make the product deliver its items/increments to the clients, no one can tell them what to do (Scrum guide 2020).

1.3.1.1.3 Scrum Master

The coach, guardian and servant of the scrum, the scrum master role is mainly about maintaining the implementation of the scrum and making sure that all the values, principles, rules and practices of the scrum are well performed and achieved. The scrum master is the light to guide all the other teams to understand the scrum and how their interaction may affect the scrum team, the scrum master is aware of what actions are helpful and what aren't in the process of implementing the scrum. The SM must always be the support to the scrum and all of its components (Scrum Guide, 2020; Beedle, 2002).

The SM main treat is the servant leadership, as the main responsibilities for her/him, is to help others in the scrum by being their "servant" for coaching and sharing the knowledge they gained by their expertise in that role and by following the scrum guide definitions. The SM could give lessons to the organization team to understand all about scrum. S/he is also considered as a communication bridge between the customers, project managers or stockholders. S/he also could be responsible for choosing/forming the scrum teams needed along with the management.

The SM is servant to the three parties: first, the PO; the SM will help the PO create the product backlog items in lists in addition to that, ensuring that the goal of the scrum is well known and understanding the process of planning and backlog items clearly and much more. Second, the SM is a coach to the development team, like coaching, facilitating the scrum when it's not implemented in its full version and removing any obstacles the team may face during the development. At last, the SM is the leader to the organization, by leading, coaching the adoption of scrum, planning the process of implementing the scrum, being supportive to encourage the productivity of the teams, moreover, ensuring the scrum is going to "increase the effectiveness of the application of scrum in the organization" (Ereiz & Music, 2019). Nevertheless, the teams go through five stages of development as "The five stages of project team development" mentioned which are, forming, storming, norming, performing and adjourning (Ereiz & Music, 2019).

The role of the scrum master is teaching and helping the team moving from the first stage (team formation) into the last stage (adjourning). This means that the team is mature now and the scrum values, rules or principles are well known and followed. Unlike the team in the forming stage, where the team is might not be aware of the scrum and knowledge is little about the scrum. Therefore, this stage is the most needed for the scrum master coaching. When the teams move to the mature status, the scrum master could have less efforts on the team, and there could be no scrum master. At the end, there is no scrum without a scrum master (Ereiz & Music, 2019).

1.3.1.2 Scrum Events

1.3.1.2.1 Sprint

“The heart of the scrum” (Scrum guide 2020) the sprint is defined into a time-boxed iteration. Each sprint should be no more than one month. Some organizations may use sprints ranging between one to two weeks or four weeks sprint. Each sprint is considered to be a stand-alone project, and by that we need to defend the project goal, scope, business value and the product/service delivered to the clients. The same will be entitled under the sprint. There is no change in the goal of the sprint, it needs to be determined through the planning at the start of each sprint and to make sure that during the sprint nothing happens that could affect the sprint goal. However, as for the sprint scope, it could be changed according to the decisions shared by the product owner and the development team during the sprint (Scrum guide 2020). They could add, remove or change some of the sprint backlog items.

The sprints indicate that there is amount of development work that is done through that specific duration and now ready to be released at the end of each sprint. Sprints are also sequential, so once the first sprint is over, the following sprint starts immediately once the first sprint is done (Scrum guide 2020).

During the sprint, there will be further scrum events or practices. The sprint starts with the sprint planning meeting at first, the sprint is started with the meeting called sprint planning, its duration depends on the sprint duration, if the organizations implemented the two-weeks sprint, then the planning meeting will be around four hours. If others chose to have four-weeks sprint, it may take at least eight hours to cover all the activities and developments needed to implemented under the sprint.

As the time of this meeting is relatively high, some of the coaching organizations defined the way to deal with this meeting, by dividing it into two meetings; the first meetings, where the product owner provides the list of the backlog items to be done during the sprint, with the help of the development team, and by forecasting what can and cannot be completed during the short period of time of the sprint. They help prioritizing the backlog items according to their business value, complexity and the time needed to complete the items, or the development needed to get this item done -if it depends on other items and could not be completed before other items-. The second meeting, the team gets in deep details and start commuting the items to be done or what they can complete during the sprint, and this means that the backlog items now are moved into sprint items. Both, the product owner and the development team also define how to develop these items, and assign estimation of the expected time these items will require to be done. Some scrum teams use estimation games just to make the scrum a little fun while trying to assign the correct value for estimation.

One of the estimation games named “Poker Estimation”: an agile estimating technique helps the team being gathered by the right estimation or the most agreed on estimation so it is a consensus-based technique. This technique was named the poker estimation because the cards used in the estimation is like to the pokers cards players. This happens after the product owner explains the items, defining the acceptance criteria for the user stories -the logic of how the item will be development in order to achieve the wanted goal from the user story. The team, will study the user stories for a short time and then each say its own estimation based on her/his own aspects. At the end, the most agreed card number will be chosen as the estimation efforts and will be added to the item. Later on, user stories will be distributed to the team members and will start creating sub tasks on to start the development.

The reason why some of the organizations adopted this new game or estimation, the poker estimation, is to make the estimation more dependently and encourages the team’s individuals to make the right estimation without noticing what others estimate to the sprint items (Cohn, 2005) (the traditional process of estimation), and this help planning to be unique by all the teams estimation and therefore, more accurate since non will know what the other is having under their cards which also lead to less “optimistic estimations” (Molokken-Ostvold & Haugen, 2007). The deck of cards, could have multiple with different sequential numbers, and this is defined by the team who will use this estimation technique to work with, so they define the write sequence they need in order to make most accurate estimations. Some numbers could be according to the “Fibonacci” sequence including the zero, 1,2,3,5,8,13,21,34,55 (Fibonacci sequence, 2020). Other sequence is the standard playing cards, with the Ace, 2,3,5,8, and king. Normally the highest card number indicates that the items is more complex and might not be done during the sprint, or it’s too big to estimate.

1.3.1.2.2 Daily stand-up

A new-short meeting is set up each 24 working hours during the sprint. Normally takes 10-15 minutes; others called stand up meeting. This quick meeting is held to update the team about the development's plans for the last 24 hours as well as the next 24 hours, since the last daily meeting. it's a way to make the process collaboration and performance smooth and easy to follow up as soon as possible in case of any issues. At the daily scrum, the team check their progress to reach the sprint goal and how they will continue the progress to finish all the items needed to be done until the sprint ends. It is also considered as an optimizer event since it helps the team probability to be at the highest value in order to meet the sprint goal.

Different teams work differently with the daily meetings. Despite that, they tend to reply specific questions and some tend to have more discussions about the work. Typically, the questions are already defined to be the following:

- What is the work done since the last daily meeting?
- What is the work needed to be done today?
- Is there any obstacles or impediments preventing the team from doing their duties?

All three above questions affect the sprint goal (Shwaber & Beedle, 2002; Scrum guide, 2020).

The daily scrum is gathering the development team and the scum master, it should be done always at a fixed hour and place, normally the team place, the members should attend the meeting all and non should be missed. In other cases where these are some circumstances that prevented the team member from being at the time and place for the meeting, they could join via the phone, online or having a team member representative from the same team.

The daily scrum could be a status update meeting on the project progress, when the team is gathered with the scrum master, they can work together to facilitate whatever causing the team from moving forward the sprint goal and this should be done only during those 15 minutes, if something comes up and needs to be discussed in more time, a spate follow up meeting could be done once the daily scrum is ended to discuss the issues raised (Shwaber & Beedle, 2002).

Before the end of each sprint there are two activities/events to be fulfilled. The first is the Sprint Review, where this meeting is about the product delivered to the clients, by presenting detailed increments that are done during this sprint with the efforts and the business value achieved, it is all about the product. However, the second is the Sprint Retrospective, which is mainly concerned with the process of how this product is built. (Ruben, 2013).

1.3.1.2.3 Sprint Review

The sprint review is the most important meeting (Rubin, 2013) as it helps the organization inspect its successful product and make sure to deliver it successfully to the clients/end users. Therefore, it is important to happen when the increments are done and merged to the product; because it helps the teams and everyone (stakeholders and project managers) included in the development of the product to be able to know what was developed and built. On the other hand, (Scrum guide 2020) clearly explained that the sprint review is “informal meeting”. The meeting takes no longer than four hours, depends on the sprint duration. This meeting is arranged by the scrum master (Schwaber, 2004) and s/he could choose specific individuals to show the results done in the sprint. In this meeting, all the backlog items are revised and the team could plan what backlog items could be done in the next sprint. In addition, the backlog items, that could be re-estimated and re-allocated to meet the new sprint goal and the new opportunities created.

1.3.1.2.4 Sprint Retrospective

This meeting occurs after the end of each sprint, could be done after the review or before, according to the organization. In this meeting many aspects are discussed however, the main purpose of it is to confirm the quality of the process followed to complete the sprint and what improvements could be done in the next sprint. it’s a loop where the team discuss the relationship of the people, process, the tools, how the product is built and how all of the before-mentioned affected the performance of the previous sprint. Moreover, create a plan to get things well done with more efficiency in the upcoming sprints.

This meeting could be a long meeting, since it takes the details of process of the development and it’s defined to be between two to three hours maximum, this meeting is arranged by the scrum master and s/he responsible for inviting all the parties included, could be the scrum team in addition to the stockholders if they want, where they discuss everything relating to the product and if anything went wrong or is well known that it will affect the process and may prevent the team reaching its goal for the product, this will be discussed including “including processes, practices, communication, environment, artifacts, tools, and so on”.(Rubin, 2013)

1.3.1.3 Scrum Artifacts

Scrum is managed using the following artifacts:

1.3.1.3.1 Burn down and up down chart

Burn down and burn up charts are commonly used in the scrum, and it could be an indication to the team and process progress. The burn down artifact is used to compare the story points or remaining work during the sprint by checking the story points and the remaining work day by day or by the sprint. The chart will draw the points (quantity of work) relatively with the sprints during the release. The chart will help the team to get to “no more work is needed” at the end of the sprint or the release, since the remaining work is decreasing daily or each release. While the burn up is used to compare the work progress towards the goal of the sprint or release. This chart will help the team know more about the needed outcomes to come closer to the line of the goal needed to be achieved.

1.3.1.3.2 Increment

Product increment is defined as the cumulative work of the product results and partial deliveries. Following the increment term will help the scrum team move towards the assigned vision or goal of the product and sprint (Scrum guide 2020).

1.3.1.3.3 Definition of done

Definition of done is “checklist of the types of work that the team is expected to successfully complete before it can declare its work to be potentially shippable”. Table 1 will present a sample of DoD Sheet.

Table 1

Definition of Done sheet (DoD) (Rubin, 2013)

Definition of done	
☐	Design reviewed
☐	Code Completed
	• Code refactored
	• Code is standard format
	• Code is commented
	• Code is checked in
	• Code is inspected
☐	End-user documentation updated
☐	Tested
	• Unit tested
	• Integration tested
	• Regression tested
	• Platform tested
	• Language tested
☐	Zero known defects
☐	Acceptance tested
☐	Live on production servers

In another scenario, the definition of done for an increment to be done means that the item is accepted, estimated, worked on and now it’s completed and ready to be delivered. (Scrum guide 2020). Each team may use a different definition of the term done. However, all the teams should clearly understand what does the DOD mean and how they can confirm it. All the terms add to the check list should be explained in details so the team can reflect the work done on the items accurately; this also helps record progress of the work since each point in the checklist – or any other criteria defined to ensure the increments are done – is checked at a the right time during the continuous work on the item and during the sprint, which will help keeping up to date with the health of the backlog items. If something went wrong, the team could interfere to do the fixes quickly. If one of these went missing or wasn’t checked, the feedback from the end use might be negative and that will make the item not considered as done, on other words, more work is required to complete the item and have it as done. This obstacle may lead to lagging in the sprint performance, and could affect other dependent items in the sprint (Rubin, 2013). Different factors can affect the definition of done mentioned below:

- Product backlog item nature.
- Technologies used to build the backlog item

- The organization to build the backlog item
- Obstacles and impediments that currently affecting building the backlog item

1.3.1.3.4 Backlogs

backlogs are a list of items representing the customers' needs and product specifications and features. This list is studied and prioritized by the product owner with help from the stakeholders and the scrum team. However, the product owner is responsible for managing what items that comes first by its value, as the scrum is all about the most valuable work first" (Rubin, 2013). Furthermore, the items description, listing, availability, value and estimation. Therefore, the backlog item is explained clearly in the description to show its completeness when it's done, by all the needed information that is already filled (Scrum guide 2020).

1.4 Problem Statement

Ideally, the product backlog is on constant change relatively with the new emerging requests that might change to make the product more valuable, competitive and useful (Schwaber & Sutherland, 2020). As the surrounding circumstances, like innovation and new technologies emerging to the software industry, in addition to the sub-industry market needs, like new features or new requirements that needed to be added to the value of the product, requirements are always changing and they will keep changing. This happens because of the feedback that product is receiving from the marketplace, industry, users, or even the business requirements, conditions and new technologies.

Agile -Scrum framework is implemented in more than half of the organizations around the world. Earlier work had explained how important is to manage the backlogs or customer requirements (CR). However, they identified some of the factors that resulting the changes of the backlog and how Scrum-team is working with it. Furthermore, they searched for a framework to predict these changes. Reasons causing these changes are many. The highest impact on backlog changes is two main reasons; the priority changes and live defect fixes (bugs fixes). During the sprint and due to the acceptance of changes in the scrum, priority has been stated as an issue in backlog management. Changing in priority causes the backlog order to change, which leads to loss in resources and it may cause additional unnecessary work. Priority changes can happen due to customers' feedback and stakeholders' decision in managing the product features and requests. It could depend also on the team’s familiarity, complexity and importance to the market. The problem is generated due to projects requirement’s changes because of priority changes, and in order to help minimize the changes in the priority, we’ve developed a new prioritization model based on some factor influencing the priority that will help the team minimize the changes in the backlog priorities and as a result reducing the backlog changes that are caused by priority changes.

1.5 Study Questions

The study objectives are to:

- What parameters that lead to backlog changes?
- What factors that affect the changes of the requirements in the backlog?
- How to minimize the backlog changes during a sprint during the Scrum?
- How to develop a conceptual model to reduce optimize the backlog prioritizing the backlog items during the sprint.

1.6 Study Scope

The study purpose is to identify the most important factors affecting the scrum backlog changes management and find a solution to help minimizing the backlog changes. Our scope headed to software organizations that are implementing the scrum process. Based on the local market news, two organizations were implementing the scrum in Palestine at the time of the study. To achieve this purpose, we conducted unstructured interviews with product owners from the two organizations, as the backlog management is their responsibility. The goal of the interviews was to define the factors affecting the backlogs and leading to changes in the backlog management. The two organization provide SAAS -Software as a Service- products. In addition, we investigated earlier studies backlog management and prioritization techniques, in order to help us define the reasons causing the backlog changes. As a try, to help us formulate a new prioritization model that is proposed to re-prioritize the backlog items.

1.7 Thesis Significance

This study will help the project managers, product owners whom implementing Agile practices for their organizations and projects in their decision making. More specifically in prioritizing their engineering requirements (requests) through our prioritization model.

Moreover, minimize the requirement changes during the planning phase or prior. This study will also involve the Agile-Scrum team in the process of decision making in assigning priorities to their work.

1.8 Thesis Structure

This thesis is organized as follows: Section 1 is an introductory chapter, Section 2 investigates the previous related works in the literature, Section 3 introduces data gathering methods used throughout the study, Section 4 shows the results analysis and discussion, Section 5 concludes the work.

Chapter Two
Literature Review

Chapter Two represents earlier work related to the product backlogs, changes reasons and also prioritization methods used to manage the new requirements.

2.1 Requirement changes

Agility is a welcoming environment for change in general and requirements changes in specific. This area of requirement management is classified as one of the risks that software development is facing, with an observed effect and impact while managing projects using the agile term. Requirement's volatility and traceability for scrum environment has been studied in (Alsalemi & Yeoh, 2017).

2.1.1 Overview of requirements changes

The authors in (Hammad et al., 2019) mentioned that risks in software development encompass changeable requirements, scope sneak and categorizing the nonfunctional requirements as user stories. The stories poker values (points) are selected by considering the complexity and the risk; commonly, the outcomes are acceptable with fewer risks which will help in taking more solid choices for backlog (Gandomani et al., 2019). Shi et al. (2013) built a prediction model to predict the changeable requirements. The model works on two historical data-based hypotheses which are history of the requirements and the history topic. The model applied the recall and precision formulas to obtain the highest possible accuracy. In (Ghosh et al., 2013) they observe that User Interface (UI) requirements are the most volatile but the number of requirement changes is reduced as the project is progressing. The authors built a model to predict the effort variance for requirement changes in the several phases of a project to allow the product managers to take the best decisions on changing or deferring requirements.

The adoption of agile methodologies has its own challenges, but Scrum with best practices can solve most of the challenges (López-Martínez et al., 2016). Sedano et al. (2019) mentioned the issues that affect the teams from the backlogs' perspective; usually, product managers are not prioritizing the whole backlog. They are just focusing on the highest priority stories, which might delay the progress in some areas that are not business priority but will facilitate some other stories. On the other hand, miscommunication sometimes happen, which leads the development team to work on non-priority stories and delay high priority stories. The authors added another valuable note which is; the teams are forming the activities in two perspectives which are researching the project context and creating key concepts of user stories, and the second one is delivering a software product on the first perspective. A study was made on SaaS company by (Ananjeva, et al., 2020) about UX work integration in agile environment; the study suggests two interventions to user stories, being concise or deliberate user stories. They found that a story if concise can create a lot of uncertainty due to reduced verbal description and deliberation. Communication in deliberation was not helpful to the development team as they needed to have extensive context description. However, that will reduce other possibilities that user stories can imply to the product.

The nonexistence of the big picture of quality control and the issues related to documentations are the most challenges in agile software development methodologies. The incomplete documentation and design specifications, in addition to the lack of understanding of the system and business quality requirements are considered as main issues facing the agile methodologies (Behutiye et al., 2020a). Keeping the traceability of quality control depends on backlogs which are a key aspect for documenting them as agile-based teams, using several and complex backlog structures to document quality controls (Behutiye et al., 2020b). A key question was raised in (Alsalemi & Yeoh, 2015) which is "How do practitioners perceive the use of requirement traceability for product backlog and project risk management during requirement volatility in Scrum?". Based on feedback from practitioners, they end up with a result that requirements traceability is used as an acceptable practice used to do product backlog changes prediction smoothly. Given that, the complete requirement history was recorded, as requirement traceability is designed to track the connections between the different requirements and the affected areas for each one.

2.1.2 Requirement changes types and reasons

Requirements change has many types resulted from a study made by (Alsalemi & Yeoh, 2015) summarized as follows;

Change due to addition: this type of change represents the new features and developments that will be added to the software being developed.

- Change due to deletion: eliminating features and requirements from the developed software.
- Change due to modification: enhancements and adjustments relating any feature or requirement to the developed software.
- Change for addition modifications: adding new requirements and alteration for an already built feature or development.
- Change for deletion modification: omitting requirements and alteration of an already built feature or development.

In the same study, the authors surveyed scrum practitioners and explained 11 reasons for backlog changes;

- Fixing Change.
- Missing Requirements.
- Functionality Enhancements.
- Product Strategy.
- Design Improvement.
- Scope Reduction.
- Redundant Eliminating.
- Obsolete Requirements that are no more required.
- Erroneous Incorrect.
- Resolving Conflicts.
- Clarifying Changes.

2.2.3 Requirement change and backlog prioritization

Since the Agile is another term for embracing the change at any stage (Saeeda et al, 2020) the number of changes applied to the product features will impact the product life cycle. Changes should be also considered when adding new features specially in the initial stage of election, in order to reduce future iteration modifications (Hovorushchenko & Pomorova, 2018). Not only this, also requirement prioritization is also disturbing the process flow of software projects due to emerging changes. Sedano et al, studied the backlog and explained the main issue that affects the teams is the inappropriate or incomplete prioritization; usually, product managers do not prioritize the whole backlog, they are just focusing on the highest priority stories. On the other hand, miscommunication happens sometimes causing the development team to work on non-priority stories and delay high priority stories.

A lot of proposed methods were presented to define backlog priorities. However, the factors taken into consideration were limited to the business value, urgency and size of item in most cases (Zahraoui & Idrissi, 2015); in other terms, importance and effort. Priority ranks using the beforementioned factors only, may lead to insufficient real rank to description the importance and when to develop the backlog user stories. Other studies showed that these factors might be not enough to operate the sprint backlogs delivery. Despite the fact that there were a lot of studies about the backlogs in general, not enough studies are found that help to minimizing the backlog changes using the priority accuracy.

2.2 Requirements Prioritization

Earlier studies about the priorities used normally a general formula including the importance of the item, effort needed to complete the item and the size of the items. In Scheerer et al, dependencies and its effect were not considered when managing backlog prioritization and were not mentioned in the priority formula; they calculated the degree of freedom and the dependencies occurrence which helped eliminating some of the changes in priorities that may happen due to dependencies.

2.2.1 Popular Prioritization techniques

A lot of prioritizing techniques were published in earlier studies as

- MOSCow: This method helps stakeholders to understand the initiative in a specific release. This acronym stands for Must have, Should have, Could have and won't have. (Moscow product plan)
- Must have: this group requires the items (development) included to be completed before they go to the release.
- Should have: this group list of items (developments) is completed then it's going
- to add good value to the product or software. It will be good to have it.
- Could have: this list will contain items that it may be good to have them in the product or software, however, it's not mandatory and it will not reduce the value of the product of the software in case these were not done.

- Won’t have: this list will contain items that will not be done at this time or maybe the future. If these were missing it will not affect the product. items in this list mostly either risky or costly.

- Value-Oriented Prioritization: this technique focuses on the business value perceived from these backlogs and how it’s important to the stakeholders and market (Value based prioritization in Scrum). Dependencies might not be managed in this technique (SCRUMstudy, 2021).
- Quality functional deployment (QFD): it helps the team understand the “voice of the customer” and the customer needs. They will be able to check and discuss the features from multiple points of view (Yaseen et al, 2019).
- Cumulative Voting or “100 dollars method”: is a simple approach, where stakeholders use their shares or “100 dollars or points” to vote for the requirements. The votes are explained in points, hours or money. The requirements with the highest points will have the highest priority (Yaseen et al, 2019; Hudda et al., 2016)
- Cost-Value Ranking: The main aim of this technique is to produce effective systems, with quality measures at lower costs. Independence of the requirements will be difficult to manage as the requirements increase which leads to further complexity (Zacaria, 2020).
- Analytical Hierarchy Process (AHP); the well-known and the simplest method for prioritization. Each requirement is compared to every other requirement in a pairwise manner (Yaseen et al, 2019; Hudda et al, 2016). The result of computing the ratio of the value-cost ranking with the AHP will help the process of decision making. (Bakhtiar et al., 2015). This method is an effective tool for dealing with complex decision making and to set priorities.
- Binary Search Tree; in the BST, priority is not actually assigned, this gives only simple ordering of the requirements. Using this technique will focus on nodes in the tree. For every node, the left side is always less number, order or rank than the node itself, and the right side of the node is greater than the node itself. Therefore, the order is somehow predefined by left - node - right, in other terms, less priority, root, higher priority (Hudda et al., 2016)
- Kano Model: it simply describes and measures customer satisfaction. A key element of the Kano model is that, the more time, money, and effort invested into creating, innovating, and improving the features in each of those buckets, the higher level of

customer satisfaction achieved. (9 product prioritization frameworks for product managers, 2021)

- Wieger’s Matrix approach; the priority value of a requirement is calculated by dividing the cost of implementing it by the cost and risk associated with achieving it. Whether a requirement is valuable depends not only on the value provided by the client, but also on the penalties that are incurred if the requirement isn't met or missed. Using this technique helps the stockholders to manipulate the priority to reach specific goals (Saher et al, 2020).

Each one of the before-mentioned prioritization techniques focused on a specific aspect; missing the dependencies (Thakurta, 2016), ignoring the development team decision, not appropriate for large projects or big customers (Zacarias, 2020), time consuming, priority could be easily manipulated or consider only about the functionalities (Hudda, et al 2016) not the non-functional requirements (Masood et al., 2019).

The Literature mentioned many prioritization techniques, however, there is no evidence presetting some methods are the “right one”. Therefore, this research will cover factors that are affecting the project flow without categorizing the factors. This research will be the base for developing a new methodology that will help re-prioritizing the backlog items. New ordered items will help minimize the backlog changes, and allow the companies to achieve the optimum solution in order to help them complete the projects with less loss of resource.

Chapter Three
Methodology

This chapter contains a description of the methods followed in the thesis to determine the suitable solution for our problem

3.1 Basic terminology

3.1.1 An Overview: Backlogs

Sedano et al (2019) defined the product backlog is an “informal model of work to be done”. The product backlog contains a list a pre-defined items needed to create, fix or complete a project.

A backlog is a list of going items that are needed to be implemented during the product life cycle. Development could be needed in order to deliver/fix service or a new feature to the customers. Rodeghero et al (2017) explained their importance in defining the product requirements needed to be done and in what order in addition to other cases such as bug fixes. A product backlog is the general list of all items included to be estimated in a product or service. The product backlog could be used for new development items, features, bugs, technical debt items. These items contain a description of the needed outcome (deliverable results). This term is widely used in Scrum, XP and also Kanban. There were misused terms for the backlogs and mostly considered as requirement specifications, however (Sedano et al., 2019) confirmed that the requirement specifications are not the backlog items.

3.1.2 Definitions: backlogs artifacts

Backlogs are defined into three artifacts in scrum:

- Product backlog (PB): the product backlog is a list of all the items related to the product to be done. It could also indicate to the changes of the requirements needed to update and enhance a product.
- Sprint backlog (SB): the sprint backlog is the prioritized items from the backlog items list but divided into timely limited intervals to be done during the sprint.
- Release backlog (RB): the release backlog items are a sub list of the backlogs that are defined to be delivered once the new release or version is launched.

Rodeghero et al (2017) explained their importance in defining the product requirements needed to be done and in what order in addition to other cases such as bug fixes. A product backlog is the general list of all the items included to be estimated in a product or service. Sprint backlog is the list of items taken from the product backlog to be implemented through an iteration named sprint. These items named as user story or user cases; these will describe the functionality needed from part of the product from the user's point of view, they have further explanation to what is written in the backlog items. Backlog items should be used in reference of Bill Wake's INVEST acronym ("What does INVEST Stand For?", 2021) (Pokharel& Vaidya ,2020));

- Independent: user stories should be independent from one another; the team should be able to complete the items that are not dependent on another. If this is not achieved, the planning might be affected.
- Negotiable: user stories should be not be considered as a contract, a specific goal should be assigned to the items and in order to have the items negotiable, they should not be overly details. they should be clear and have a goal that may lead the discussion to be opened. This will help the team get the better shippable delivery.
- Valuable: each user story should have a value by defining the customer's perspective. Items that don't clearly implement the customer needs is not considered valuable.
- Estimable: user stories should be short and clear enough to help the scrum team estimate the effort needed for this item, vouge and unclear items are more likely to be risky and complex, therefore, priority will be less for such items.
- Small: user stories should not take a lot of time to be completed, shorter user stories assist the team in focusing on and delivering customer values, as well as receiving frequent feedback.
- Testable: user stories should be explained clearly in order to be tested carefully. Big or vouge user stories cannot be tested properly.

3.1.3 Backlog in Scrum

In Scrum, customers' needs and product features are translated into backlog item, as these backlogs get studied and estimated in order to draw a reasonable product road map, 80% of these items are estimated.

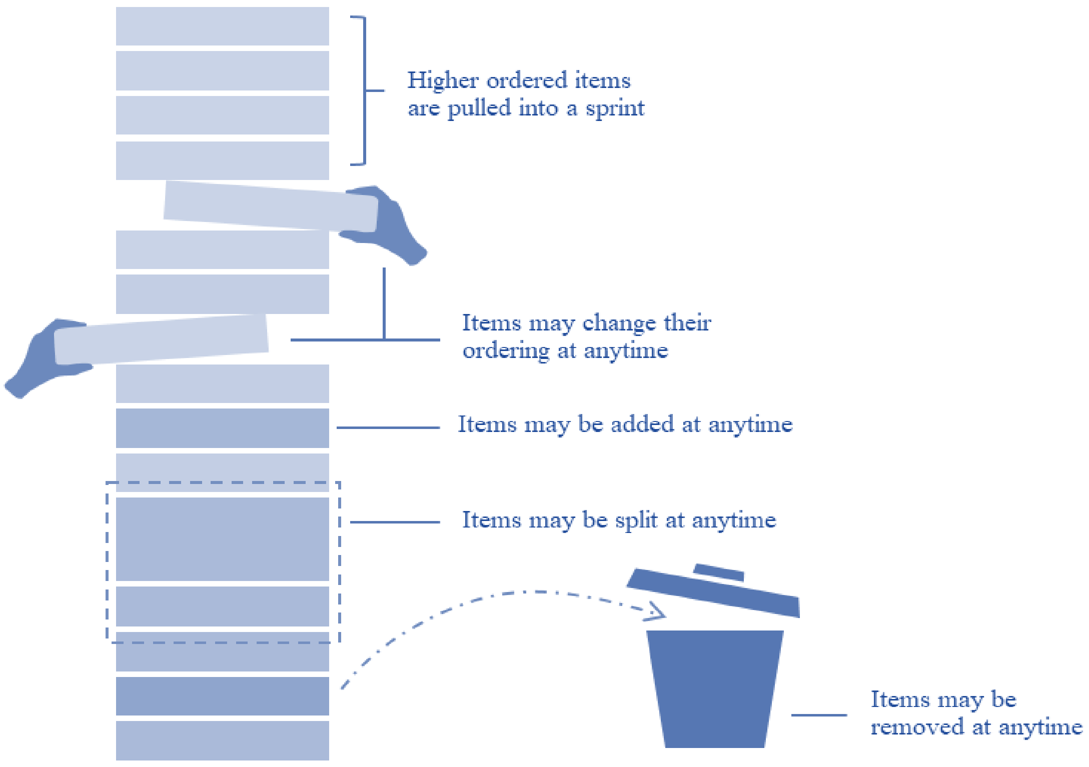
According to the priorities of the stories and bugs, the product backlog is ordered for the sake of being groomed and planned, sprint planning is controlled by combining the risk, density, and value of the user stories, in addition to constraints that are related to business, delivery, team skills, and other factors (Jansi & Rajeswari, 2015). The product backlog should be short and steady to ensure rapid feedback from users. In the sprint, the developers perform a comprehensive investigation for the stories, as a result, the team members should work on a group of stories that at the end of the sprint should be designed, implemented, and tested (Golfarelli et al., 2013). This is already managed in the INVEST set of characteristics that should be applied to stories. A study on user stories (Pokharel & Vaidya, 2020,) pointed that the story indicates three parameters, a user managing the role to allow reaching a goal in order to achieve a value (Chopade & Dhavase 2017).

3.1.4 Product backlog refinement

Backlog refinement is the responsibility of the Product Owner, the PO manage the product backlog by assigning priorities, preparing and changing the description of the items when new information or changes are emerged. These changes will affect the order of the product backlogs. In figure 3 below, the product backlog is divided into 3 imaginary sections. Highest priority items in the first section, next comes the moderate priority items, the last section contains less priority items and not likely to be implemented – moved to a sprint in order to develop it. In the product backlog there is a place of change before entering the items to the sprint. Normally, the highest prioritized items will be moved into the current of next sprint. Any time during the sprint, moderate items or changed items may be moved to the sprint. The decision depends on the stockholders, or customers in reference to organizations (Ruben, 2013).

Figure 3

The product backlog refinement



("How to Refine Product Backlog?", 2021)

3.2 Study Approach

In the present study, we used three mythologies. Firstly, we have observed the previous works about the backlog changes causes and effects. Secondly, we have conducted unstructured interviews with 5 product owners from two different local companies that implement the scrum framework as their project management methodology. lastly, a study case combined with a benchmark project to validate our study.

The base study was raised by (Alsalemi & Yeoh, 2015), discussing the reasons of backlog changes and how it has serious effect on the progress of the projects in Software development companies.

3.3 Study Sample

The study sample was a local software company named Company F. We have studied three sprints backlog items. Based on one of its old employees, Company F is the one of the first companies that started implementing the scrum in Palestine.

3.4 Study Design

The study methodology consists of the following six stages:

1. Defining the problem: finding that the product backlog changes in a Scrum environment is not managed carefully which led to loss in resources: we have studied the literature in this stage and gathered many information related to the backlog changes reasons and types.
2. Defining backlog changes reason using live experienced team, mainly product owners, as they are responsible about managing the backlog: we have conducted unstructured interviews with five POs from two local companies, questions were clearly focused on the backlog changes reasons.
3. Defining the dominant factor in causing the changes: from step 2 and 1, we have found the most accruing reason factor affecting the backlog changes.
4. Helping minimizing the effect of the dominant factor by proposing a new model
5. Validating our model by implementing a benchmark project and study it on a local case study in a local Company F.

3.5 Study Methods

The choice of the data collection is influenced by the type of study design used starting with defining the factors leading to backlog changes in Scrum environment.

3.5.1 Factors considered in the proposed methodology:

- Business value: defined by of the total business value gained by this item. In terms of competitiveness and industry needs.
- Customer retention: defined by the customer satisfaction which depends on customer needs, requirements and the quality of the product delivered to industry.
- Developers features (familiarity and experience): defined by the development team experience and knowledge or similar work and development.
- Dependencies: defined by the number of items dependent or independent of the item's discussion
- Backlog description language (action words): defined by the natural language actions. And number of actions works similar to the following; (View, edit, create, block, allow, send, receive and overwrite)

- Complexity; defined by the uncertainty and risk assigned to the item, this could be measured in terms of size and risk.
- Time estimation; defined by task efforts needed to complete the items in terms of hours.

3.5.2 Dominant factor: Priority

Based on (Popli & Chauhan, 2018) prioritization for the user stories in a sprint is dividing the importance by the effort in equation (1). Therefore, in order to calculate the factors affecting the prioritization in our model, they were inspired by the model proposed (Sher et al, 2014) the weighted sum model to calculate the weighted score for many aspects in a user story presented in equation (2).

$$\text{Priority} = \text{P}= \frac{\boldsymbol{Importance}}{\boldsymbol{Estimated\ Effort}} = \text{I/E} \geq 0 \ldots\ldots\ldots \qquad (1)$$

where;

- P: the priority rank for a specific user story,
- I: importance of the user story,
- E: the estimated effort of the user story

3.5.2.1 Calculating Importance (General)

$$\text{Importance (business)}= \sum_{\text{i}=1}^{\text{j}} \frac{\text{Weightji}}{\text{Scoreji}} \ldots. \qquad (2)$$

Where;

- i= user story ≥ 1
- j= importance factors defined as {business value (competitive), customer retention, Team familiarity}

Factors defined when calculating the effort

❖ Business value is proportional to the importance based on (Zahraoui & Idrissi 2015)

❖ Customer retention (satisfaction) is proportional to the importance based on (Sawsant 2020)

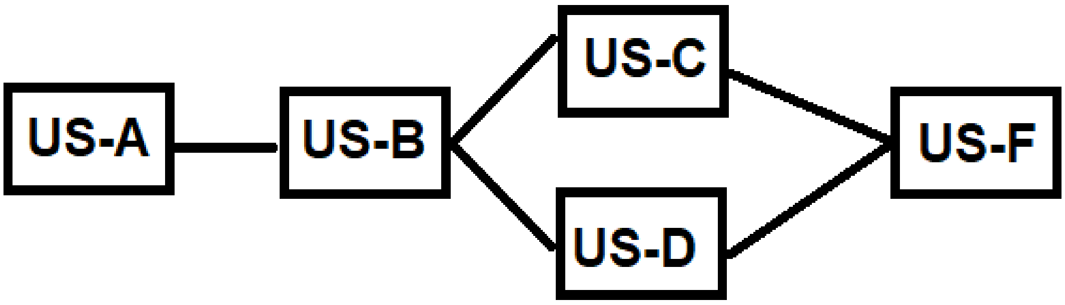
- ❖ Team familiarity is proportional to the importance as case study, when the team is familiar with the items details becomes clear, therefore, the resources to be spent are much less, and beneficial to the business

Another important factor found by this study is Dependency factor (Df)

As dependencies and their effects on backlogs were not previously emerged to the prioritization; dependency is considered as a dominant factor in the prioritization process (Scheerer et al, 2015). Independent items should be done before dependent items. Which means, if A depends on B, B should be done first, therefore, B should have higher value, in other terms, higher importance. Therefore, a dependency factor value should be assigned relating to the importance; i.e., figure 4 explains 5 user stories interdependencies. i.e., US-A →US-B → US-D+ US-C→ US-F

Figure 4

User stories interdependences graph



According to the dependency graph for the sprint items, as of figure 4 example; then US-A should have the highest DF of the US-B, US-B should have higher value than US-D or US-C, US-F should have low DF.

Df; the higher value of Df, means that many items are depending on this item, which leads to the fact, that it needs to be done before other items that are depending on this one. In other words, the higher the priority of that item

Df; the higher value of Df, the higher the priority of that item.

- ❖ Adding DF to equation (2) will result equation (3) that describes the total importance in terms of weighted score and dependencies.

$$*Ti = \text{Total importance} \sum_{i=1} \frac{\text{Weightji}}{\text{Scoreji}} *Dfi \geq 0 \ldots\ldots\ldots\ldots \quad (3)$$

Where;

- i and j as pointed in equation (1)
- Dependency factor ≥ 1

3.5.2.2 Calculating Effort (General)

Similar to the importance, the effort time estimated is divided by the importance in most of the prioritization techniques proposed earlier like equation (1), and defined to be the estimated time. In our proposed model, new defined new factors that may increase the effort unexpectedly, as a result, will lead to decreasing in the priority.

Factors defined when calculating the effort;

- ❖ Estimated Estimated time in hours; well-known factor that indicates the effort in time needed to complete the user story and its inversely related to the priority.
- ❖ Action Action words count; the description of the user story could indicate a lot of characteristics in the users' stories management. Based on (Castillo-Barrera et al., 2017; Chopade & Dhavase, 2017)), more action words in a user story, means more risk, which leads to less priority.
- ❖ Complexity Complexity factor; is an indication to the user story size combined with the uncertainty, bigger user story, have higher uncertainty and less priority.

As the complexity of the user story is indicated by the uncertainty and the understanding of the user story description, and as a reference for the literature, the more action words added to the user story description, the more complex become the user story. Therefore, equation (4) is established to combine the uncertainty factor (Uf) and the number of action words (Aw) in order to indicate the complexity factor (Cf).

$$\text{Complexity factor (Cfi)} = Awi * Ufi > 1 \ldots\ldots\ldots\ldots\ldots\ldots \quad (4)$$

where;

- i= user story ≥ 1
- Aw: the number of action words in the description of the user story ≥ 1
- Uf: the uncertainty factor, indicated the size and risk of the user story ≥ 1

Discussing the aforementioned factors and combine them with the effort needed to complete the user story actual effort. Calculating the "Total efforts (Te)", will require to use equation (5).

$$*Te = \text{Total Effort}i = (Cf * \text{Estimated Time}_i) \ldots\ldots\ldots\ldots \quad (5)$$

- i= user story ≥ 1
- Cf: complexity factor ≥ 0
- Time: estimated actual time needed to complete the user story ≥ 0

3.5.2.3 Proposed Priority Model

The re-prioritizing formula is combining the above factors based on (Popli & Chauhan, 2018) in equation (1) and resulting the new formula in equation (6);

$$\text{Priority} = P = \frac{\boldsymbol{Importance}}{\boldsymbol{Estimated\ Effort}} = I/E \geq 0 \ldots\ldots\ldots \quad (1)$$

Our model is based on gathering the most important and not known factors into one equation that will help the POs in their decision-making criteria. Therefore, filling equations (3) and (5) into equation (1) and representing them in equation (6).

$$* \text{Pi} = \sum_{i=1} \frac{(\text{Weightji} * \text{Scoreji}) * \text{Dfi}}{CXi * Ti(estimated)} > 0 \ldots\ldots\ldots\ldots\ldots\ldots \quad (6)$$

where:

- i: user story number,
- j: importance factors = business value (competitive), customer retention, Team familiarity} organization will define the weighted score.
- Df: dependency factor, organization will define the score rank.
- Cf: complexity factor = (Awi *Ufi) = (number of action words * uncertainty factor).
- Ti: estimated time needed to complete the user story in hours.

3.5.2.3 Proposed methodology and algorithm

Our proposed model was built based on the pre-identified factors that is affecting the progress of the scrum by lagging the scrum time because of the unsystematic backlog management as well as ignoring the backlog changes management. Therefore, this research will help the scrum teams and agile in general running their product sprint backlogs order smoothly with minimum changes caused in terms of prioritization. Using our proposed model, we have built a new methodology and algorithm steps that could be implemented into a prioritized backlog list or not prioritized backlogs:

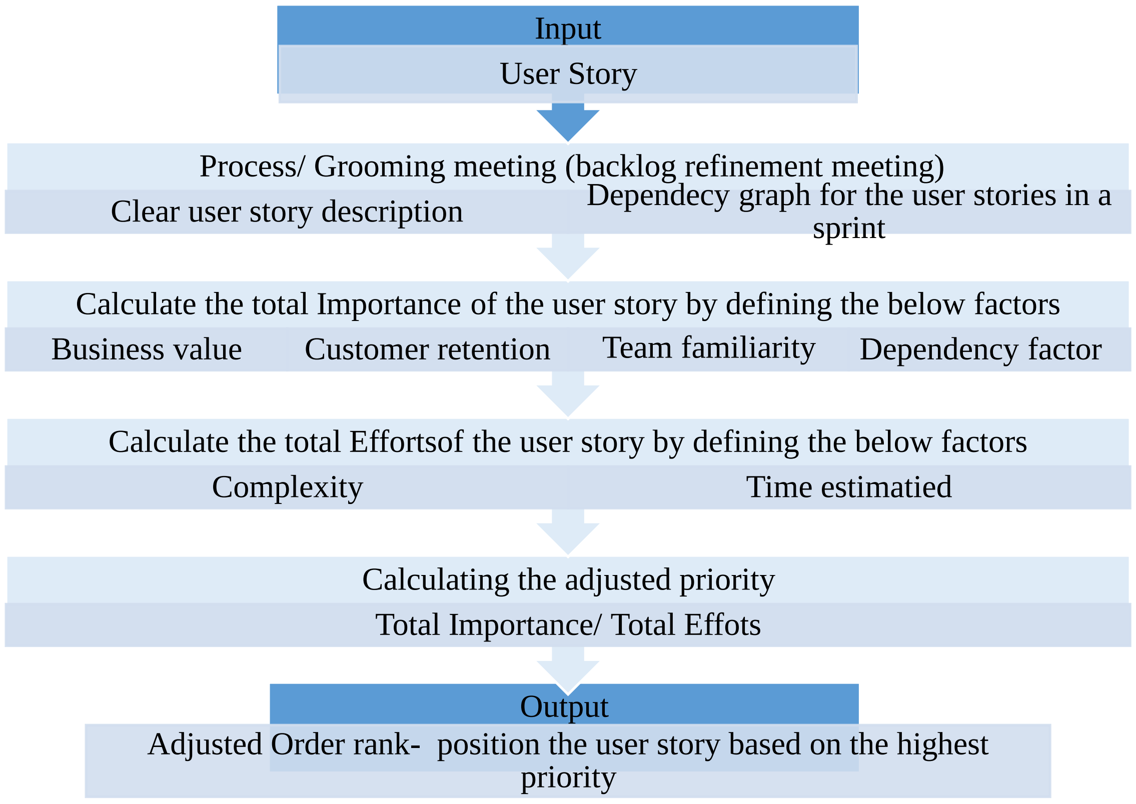
1- When adding a new requirement, item, bug, user story, make sure that the description is used in a clear format and defined number of action words. In addition to meeting the INVEST criteria.
2- The process starts in the refinement meeting (grooming). In this meeting the team discusses what items should be done in the next sprint. (Normally the highest orders and priority).
3- Draw a dependency graph gathering those items added in the sprint, in order to define the interdependencies between the items as in figure 4.
4- Checking those higher ordered items and start implementing our proposed model by adding the importance needed in terms of business value, customer retention and developers' features.
5- In planning meeting, the PO will assign the items to most experienced development based on their features, in terms of familiarity.
6- Calculate the weighted score as of equation(2).
7- Define the dependency factor, the team with help from the developer assigned should be able to rank the items' dependency factor. In addition to the complexity factor.
8- Calculate the total importance by using the sum of the business value ranks multiplied with the dependency factor as of equation(3).
9- Allow the assignee from the prior point to add the estimated time needed and negotiate the dependency graph. Since they are familiar with the items, they will be able to give the estimation needed as well as the complexity resulted in this user story.
10- Count the action words in the item's description. Use α for each verb and β for duplicated action words, then multiply by the uncertainty factor as of equation(4).
11- Calculate the total effort by the sum of the complexity factor and the number of action word multiplied with the time estimated as of equation(5).
12- Divide the total importance by the total effort to get the new priority as of equation(6).
13- Order the items based on the new priorities resulted from the proposed model ascending or descending.

3.5.2.4 Proposed methodology flowchart

This new order will help the team implement the scrum sprints in the minimum number of changes in terms of priority and dependency changes. The flowchart in figure (5) will explain in steps the procedure of this proposed methodology.

Figure 5

Proposed methodology chart flow


Input
User Story
Process/ Grooming meeting (backlog refinement meeting)
Clear user story description
Dependecy graph for the user stories in a sprint
Calculate the total Importance of the user story by defining the below factors
Business value
Customer retention
Team familiarity
Dependency factor
Calculate the total Effortsof the user story by defining the below factors
Complexity
Time estimatied
Calculating the adjusted priority
Total Importance/ Total Effots
Output
Adjusted Order rank- position the user story based on the highest priority

3.6 Validating the study

Validating the study was presented in two projects. The first, creating a benchmark and applying the proposed methodology. The experiment of the benchmark project used the dependency graph as a validation point to confirm the behavior of the results. The results were promising as we will be discussed in section 4. The second, applying our model into a local company, named “Company F”. In this project, the study used the time estimated as a validation factor to check the results, details will be discussed in section4.

Chapter Four
Results and Discussion

This chapter shows the results and the proposed prioritization model in order to minimize the backlog changes using dependencies graph.

4.1 Results

This study aims to optimize product backlog changes by proposing a new prioritization technique using dependency graph and other defined factors. This research will present the results of our experiment with and without our proposed model.

In section 3, we have formulated a new model to calculate the priority for the backlog items during the sprint, equation 6. We will use this model in our benchmark project and the case study then compare the results:

$$* \text{ Pi } = \sum_{i=1} \frac{(\text{Weightji} * \text{Scoreji}) * \text{Dfi}}{CXi * Ti(estimated)} > 0 \ldots\ldots\ldots\ldots\ldots\ldots \quad (6)$$

where:

- i: user story number,
- j: importance factors = business value (competitive), customer retention, Team familiarity} organization will define the weighted score.
- Df: dependency factor, organization will define the score rank.
- Cf: complexity factor = (Awi *Ufi) = (number of action words * uncertainty factor).
- T: estimated time needed to complete the user story in hours.

4.1.1 Experimental Results

In this research, we have implemented our model into a benchmark project and a case study. In both projects, we have gathered the information of the backlog prior our model and then implemented our new re-prioritization technique and the followed algorithm in order to validate our results.

4.1.1.1 Benchmark Project: HR Application

For further validating of our methodology and proposed model, we’ve developed a new small project with the help of one of the POs in Company F, named HR Application.

This project has some requirements and features to be developed in order to complete this project. Therefore, we’ve written this information as backlog items (user stories) defined in table 2. In addition, we’ve taken into consideration how to write the action words and their quantity in one item.

4.1.1.1.1 Benchmark project user stories:

Table 2

Benchmark project items (HR application) (not prioritized).

Items number and description
1-User level should be automatically defined using email, as employee, middle management, higher management, Intern
2-User should be able to create a profile using email or phone number. This profile should be verified to the user when the user clicks the verification code send to their email of mobile.
3-Users should be able to view and edit 4 tabs, Time-management, Work-sheet, Vacations and Requests sent to HR/Manager. Users will only see their data and should not access any other user's data
4-User should be able to read /view /download the company polices
5-User should be able to add/modify their vacations and the manager should be able to accept and send announcement in behalf of the user.
6-Users should be able to receive Birthday wishes as email and an announcement of the user birthday should be sent on behalf of all users.
7-User should be able to contact the HR manager in case of Salaries updates and requests of advanced payments from the salary. The user should receive updates as excel or PDF on the profile about his/her salary status and amounts with taxes calculations
8-Managers should be able to view the requests with the name of the requester on the requests screen. They should be able to approve the request then send a notification for the user about his/her requests status. Once the request is approved, the app will send an announcement to all users by clicking "announce to all".
9-A new tab should be created on managers screen to see reports. These reports will collect the users’ data based on time entered in the time sheet and the tasks related. The manager can modify the report based on the data inserted by the user.
10-Managers should be able to submit online forms for employees. Managers should be able to create online forms similar to excel and share it with users in order for the users to fill.
11-A shared tab for internal job vacancies should be viewed by the users. The users should be able to receive notifications about new opening and to apply online and the application should be sent to Managers for review.
12-External job vacancies screen should be created on the main screen, this screen will be shown without the need to be a user. The applicant can apply to these positions without creating a user.
13-HR manager can accept the new applicants’ applications and arrange the interview by sending emails from the HR profile.
14-Video conference calls available between Managers, employees and applicants through a link integrated with (Zoom).

4.1.1.1.2 Benchmark project backlog prioritization

This section consists of two parts;

- Part one: After preparing the list of the backlog user stories, we have managed to get help from one of our interviewees in Company F, the PO has assigned the importance and effort needed for the items to be completed without considering any dependencies, developers features or any other previously mentioned factors in our model.
- Part two of the benchmark project prioritization is re-prioritizing the user stories based on our model and methodology. Therefore, the next subsections will explain in details how we managed to achieve our results.

4.1.1.1.2.1 Benchmark project backlog prioritization– using the most common technique

Table 3 will present the general prioritization rank based on the mostly used technique taking into consideration two factors only, the importance in ranked numbers and estimated efforts in hours.

Table 3

Prioritized benchmark project items using importance and effort ranks only.

Item #	Importance	Estimated effort	Priority
	Rank 1-10	hours	
1	9	14	0.64
2	8	18	0.44
3	5	15	0.33
4	5	9	0.55
5	8	35	0.23
6	3	3	1.00
7	7	24	0.29
8	9	18	0.50
9	9	15	0.60
10	5	36	0.14
11	9	25	0.36
12	8	5	1.60
13	8	4	2.00
14	4	4	1.00

Where:

- Item # is the ID of the user story.
- Importance is the general importance of the item to be delivered to the customer.
- Estimated effort is the amount of time expected to complete the items.
- Priority is the order rank for implementing the items in a sprint.

POs normally use the priority rank in table 3 as a reference for user stories order when it's time to start the project.

4.1.1.1.2.2 Benchmark project backlog prioritization– using the proposed re-prioritization model

Following the methodology presented in chapter 3, in order to re-prioritize the backlog items to have the minimal backlog changes due to priority changes. We need to accomplish some steps before calculating the new priority.

4.1.1.1.2.2.1 Benchmark project dependency graph

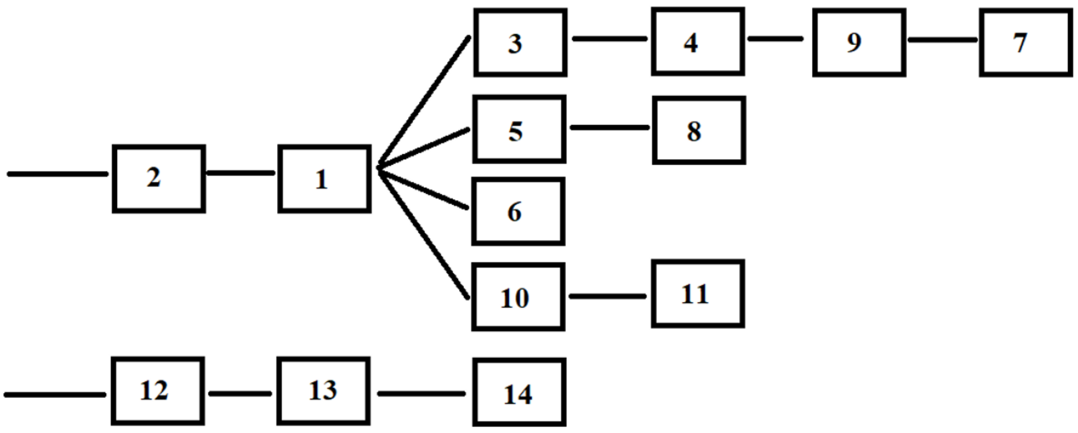
Once the project user stories description is completed, we've built the dependencies chart in figure 6 based on experienced PO from Company F. As the figure shows, that the higher dependency factor should be assigned to item 2 and 12, which means, they can be developed in parallel as long as the two branches do not depend on reach other.

If at some point item 12 was depending on items from the first branch, item 2 should have the highest DF.

- Branch 1 → 2 – 1 – (3 / 5 / 6 / 10) and others.
- Branch 2 → 12 – 13 – 14

Figure 6

HR Application dependencies



4.1.1.1.2.2.1 Benchmark project re-prioritization

Since the project items is prepared, description is clear, dependency graph is established. We can run the benchmark project into our model and help the team re-prioritize and re-order the items, mainly based on dependencies and other factors like developer features and action words number.

Excel sheets were used to manage and prepare our model table to make the process easier and faster. The PO needs to add the rankings and numbers values to the excel sheet and the new priority will be calculated. In this research, table 2 user stories were added to the excel sheet based on our proposed model based on equation 6. In addition, the methodology in section "3.5.2.3 Proposed methodology and algorithm". table (4) is generated:

Table 4

Prioritized benchmark project items using our model

Item #	BSV	CS R	TM F	DF	TOTAL I	Aw	UF	CX	$T_{estimated}$	Total E	New order
	4	5	3	from 1-5			from 1-5				
1	4	2	4	4	168	1	4	4	2	8	21.00
2	4	3	5	5	255	4	1	4	1	4	63.750
3	3	4	5	3	156	4	1	4	2	8	19.500
4	3	4	5	2	104	2	1	2	3	6	17.333
5	2	5	3	2	90	4	3	12	5	60	1.500
6	1	4	1	1	28	2	1	2	1	2	14.00
7	4	4	2	1	44	3	3	9	4	36	1.222
8	4	3	3	1	43	4	2	8	3	24	1.791
9	5	1	4	2	82	3	2	6	3	18	4.555
10	4	3	3	2	86	4	2	8	6	48	1.792
11	3	3	1	1	31	4	1	4	5	20	1.550
12	5	3	2	3	129	2	3	6	1	6	21.500
13	5	4	3	2	104	3	1	3	2	6	17.333
14	5	5	5	1	65	2	2	4	1	4	16.250

Where:

- #: User story ID.
- BV: business value.
- CR: customer retention.
- TF: Developer features (team familiarity).
- DF: dependency factor.
- Ti: total importance.
- Aw: number of action words.
- Uf: uncertainty factor.
- Ti: estimated time to complete the user story.
- Te: total effort needed.

Table 4 explains the details of the proposed methodology and the factors calculations in equation 6. The results or the new prioritizations ranks were compatible with figure 6.

The table 4 above shows us the most appropriate order for the backlog items with less changes in priority as we've considered the most needed factors in our model.

4.1.1.1.3 Benchmark project backlog prioritization comparison

In comparison between the benchmark project prioritization using the general formula of prioritization technique and prioritization based on our model proposed in this study Table 5 presents the difference in benchmark project items numbers.

Table 5

Order of benchmark items comparison based on priority ranks.

Item # (ID)	Priority using importance/effort	Priority using our proposed model
1	13	2
2	12	12
3	14	1
4	6	3
5	1	4
6	9	13
7	4	14
8	8	6
9	2	9
10	11	8
11	3	10
12	2	11
13	5	5
14	10	7

As noticed, column 2, meets our proposed dependencies in figure 6. The results were promising as the new order is logical based on the dependency graph. Only two items were odd, item 5 and 8, while 5 should be done before 8, the results in both methods showed the same behavior. Other items orders based on our model seems to fit the dependency graph more than general prioritization techniques. Dependency, have been already found that it's a major reason of change in the backlog's priority and lead to priority changes. Influencing factors used in our model, i.e., the developers features and number of action words in the description of the user stories was helping the new order validity. This can be noticed from table 5, items 2,1, these items have a lot of depend other items. Item 2 contains a higher number of action words, higher dependency factor and higher team familiarity, but lower time estimated. logically, point 2 is the most important point as it's the key to this project, therefore, it's needed to have the highest priority. Same factors applicable to point 1, 3 and so on.

Other items, like 11 have less dependency factor, however, it has higher complex factor and estimated time, which indicates lower priority in order. That also confirm our model by 3 factors; first, the developer features mean this item is not well known by the developer (s/he might not aware of all the needed work to complete this item). That leads the item into uncertainty and risk zone, a lot of unknown changes may occur while implementing this item. Secondly, average time is needed to complete these items despite the higher complexity considered relatively more than average amount of action words and uncertainty factor.

The user stories new orders seem more applicable for the team to work on the items, with less changes due to dependencies. Familiarity will allow the team to complete the work they have experienced first in order to accomplish it as soon as possible with no delays due to risks or unknown cases. That will save time for complex user stories more than the normal once. This will also be noticed in our study case on a local firm in Palestine (Company F) that has been implementing scrum for 5 years.

4.1.1.2 Study Case: Company F

We studied and implemented our proposed model into to a 3-sprint period at a local software development company in Palestine, nick-named as Company F; this is an interesting company to study the scrum process as they've been one of the first companies to implement the scrum for their operations since approximately 5 years -until this study is made. They develop a SaaS product for freight forwarding industry that went viral and global. They receive various requirements from their customers as well as new technologies emerging in their industry and its market; therefore, it will be a great material to run our model on. They use two weeks sprint iteration and each release is launched after five sprints developments. They also do just-in-time bug fixes and small work items developments that takes less than a day; using what is called as "maintenance board". They use Microsoft Azure DevOps as their Scrum management tool, where they can enter the backlogs items and manage it. In this company, they have 4 POs for 5 Scrum teams, no scrum master, the development team leader is considered as a Scrum master.

4.1.1.2.1 Company F backlogs management

In order to study the proposed method and its effect on the backlog changes and scrum, we have conducted a couple of unstructured interviews with the POs to understand the process and to gather their opinion on the factors that might affect the scrum process. The interviewees agreed that the most factors affecting the backlog changes are the priority changes as well as the live bug fixes. Based on their answers, we've investigated more in the prioritization and how it's measured in the company. As a result, we have proposed our new methodology to calculate the priority or re-calculate the priority during a sprint. Table 6 presents a 3-sprint backlog items and their order with the priority changes that has been changed on these items before committing the items to be done and before implementing our model.

Table 6

Backlog items in Company F and priority changes before our proposed model.

O	S	I#	Priority changes #
1	A	137390	6
2	A	138080	2
3	A	137873	3
4	A	137435	10
5	A	138231	2
6	A	138320	2
7	A	137617	6
8	A	137961	10
9	A	137703	4
10	A	134339	8
11	A	134764	6
12	A	136139	13
13	A	137965	-
14	A	138355	-
15	A	138213	-
1	B	137435	15
2	B	137961	12
3	B	137960	7
4	B	138941	4
5	B	138725	3
6	B	137963	9
7	B	138418	9
8	B	137734	6
9	B	137679	6
10	B	137735	8
11	B	137732	9
12	B	137962	9
13	B	138528	6
14	B	138827	5
15	B	139534	2
16	B	139156	-
17	B	139260	-

18	B	137941	4
19	B	137606	6
20	B	137688	6
21	B	136954	10
22	B	138280	-
1	C	140101	6
2	C	137435	18
3	C	139051	6
4	C	138013	8
5	C	140094	2
6	C	140090	3
7	C	139809	5
8	C	139762	7
9	C	138762	10
10	C	139691	2
11	C	139970	2
12	C	139692	6
13	C	137957	12
14	C	138948	7
15	C	139975	2
16	C	137868	9
17	C	139828	-
18	C	139829	-
19	C	139956	-
20	C	138280	-

Where:

- O: order of the user story
- S: sprint ID (A, B and C)
- I#: user story ID
- Priority changes #: total number of changes due to changing priority before the items are committed to be done.

4.1.1.2.2 Backlog order using general prioritization in Company F

Since we have already the order of the items, we have run our model into these items using Excel Sheets. The general prioritization technique was tested on a three-sprints backlogs with no dependency graph shown in the table 7.

Table 7

Backlog items in Company F using general prioritization model.

O	S	I#	Importance	Estimated Effort	Priority
Sprint A					
1	A	137390	32	26	1.230
2	A	138080	48	11.5	4.173
3	A	137873	32	4	8.000
4	A	137435	48	40	1.200
5	A	138231	16	3	5.333
6	A	138320	16	4.5	3.555
7	A	137617	16	3.5	4.571
8	A	137961	64	25	2.560
9	A	137703	16	2.2	7.272
10	A	134339	32	19	1.684
11	A	134764	64	16	4.000
12	A	136139	16	4	4.000
13	A	137965			
14	A	138355			
15	A	138213			
Sprint B					
1	B	137435	48	40	1.200
2	B	137961	64	25	2.560
3	B	137960	64	23	2.782
4	B	138941	80	15	5.333
5	B	138725	16	2	8.000
6	B	137963	48	9	5.333
7	B	138418	16	4	4.000
8	B	137734	32	2.5	12.80
9	B	137679	16	7	2.285
10	B	137735	16	4	4.000
11	B	137732	16	5	3.200
12	B	137962	48	9	5.333
13	B	138528	16	3	5.333
14	B	138827	16	3	5.333
15	B	139534	32	2	16.00
16	B	139156			
17	B	139260			
18	B	137941	16	2.5	6.400
19	B	137606	16	6	2.666
20	B	137688	16	3	5.333
21	B	136954	16	3	5.333
22	B	138280			
Sprint C					
1	C	140101	80	44	1.818
2	C	137435	48	40	1.200
3	C	139051	48	13	3.692
4	C	138013	32	15	3.200
5	C	140094	48	22	1.454
6	C	140090	48	10	4.800
7	C	139809	16	5	9.600
8	C	139762	16	3	5.333
9	C	138762	32	8	2.000
10	C	139691	80	21	1.523

11	C	139970	80	17	4.705
12	C	139692	64	9	8.888
13	C	137957	16	5	12.80
14	C	138948	16	5	3.200
15	C	139975	16	5	3.200
16	C	137868	16	6	2.666
17	C	139828			
18	C	139829			
19	C	139956			
20	C	138280			

Where:

- O: order of the user story
- S: sprint ID (A, B and C)
- I#: user story ID
- Importance: total importance based on our model and methodology in section 3.5.2.3
- Estimated effort: total efforts based on our model and methodology in section 3.5.2.3
- Priority: priority rank based on general prioritization formula (Importance and Estimated efforts in hours)

4.1.1.2.3 Backlog order using the proposed prioritization model in Company F

Table 8 will present the data when we implemented our model into the same three-sprints items in Company F. Data in the table is added by the PO responsible for a specific team, and he is responsible for managing these backlogs. The PO was very cooperative to work with us on this study.

Table 8

Backlog items in Company F using our prioritization model with no dependency graph.

S	Item #	BS	CS	TF	DF	TOTAL I	# Aw	Uf	CX	$T_{estimated}$	TOTAL E	New Order
	Weight/rank	4	5	3	1-5			1-5				
Sprint A												
A	137390	8	10	9	1	118	3	2	6	40	240	0.491
A	138080	12	10	6	3	366	1	1	1	13	13	28.15
A	137873	8	20	12	1	180	1	2	2	2	4	45.00
A	137435	12	15	12	2	342	1	2	2	50	100	3.420
A	138231	4	10	12	1	114	1	1	1	3	3	38.00
A	138320	4	5	9	1	77	1	3	3	8	24	3.208
A	137617	4	15	15	1	151	1	1	1	2.5	2.5	60.40
A	137961	16	20	9	2	400	8	3	24	35	840	0.476
A	137703	4	15	12	1	139	1	1	1	2.5	2.5	55.60
A	134339	8	10	9	1	118	3	3	9	25	225	0.524
A	134764	16	15	6	1	163	6	3	18	20	360	0.452
A	136139	4	15	9	1	127	1	2	2	5	10	12.70
A	137965											
A	138355											
A	138213											
Sprint B												
B	137435	12	15	12	2	342	1	2	2	50	100	3.420
B	137961	16	20	9	2	400	8	3	24	35	840	0.476
B	137960	16	20	9	1	200	5	3	15	45	675	0.296
B	138941	20	20	12	1	228	11	3	33	27	891	0.255
B	138725	4	15	12	1	139	1	1	1	2.5	2.5	55.60
B	137963	12	20	12	3	588	2	2	4	5	20	29.40
B	138418	4	15	12	1	139	1.5	1	1.5	4	6	23.16
B	137734	8	10	12	2	260	2	1	2	3	6	43.33
B	137679	4	5	9	1	77	1	1	1	5	5	15.40
B	137735	4	10	9	1	102	1	2	2	4	8	12.75
B	137732	4	5	9	1	77	1	2	2	5	10	7.700
B	137962	12	10	12	3	438	2	1	2	10	20	21.90
B	138528	4	15	12	1	139	1	1	1	2	2	69.50
B	138827	4	15	12	1	139	1	1	1	2	2	69.50
B	139534	8	20	9	1	168	1	3	3	2	6	28.00
B	139156											
B	139260											
B	137941	4	10	12	1	114	1	1	1	2.5	2.5	45.60
B	137606	4	5	9	1	77	1	2	2	5	10	7.700
B	137688	4	5	12	1	89	1	2	2	4	8	11.125
B	136954	4	5	12	1	89	1	1	1	2	2	44.50
B	138280											
Sprint C												
C	140101	20	20	3	1	192	8	5	40	45	1800	0.106

C	137435	12	15	12	3	513	1	2	2	50	100	5.130
C	139051	12	10	12	2	292	4	4	16	15	240	1.216
C	138013	8	20	9	1	168	1.5	3	4.5	25	112.5	1.493
C	140094	12	15	6	2	294	3	3	9	10	90	3.266
C	140090	12	10	6	1	122	5	3	15	20	300	0.406
C	139809	4	15	12	3	417	1	2	2	7	14	29.78
C	139762	4	15	12	1	139	1	1	1	2.5	2.5	55.60
C	138762	8	15	9	1	143	4	2	8	27	216	0.662
C	139691	20	20	12	1	228	12	3	36	23	828	0.275
C	139970	20	20	12	1	228	8	3	24	25	600	0.380
C	139692	16	15	6	1	163	6	4	24	7	168	0.9701
C	137957	4	5	9	1	77	1	1	1	2	2	38.50
C	138948	4	5	9	1	77	1	1	1	3	3	25.66
C	139975	4	5	9	1	77	1	1	1	5	5	15.40
C	137868	4	15	12	1	139	1	1	1	3	3	46.33
C	139828											
C	139829											
C	139956											
C	138280											

where:

- #: User story ID.
- BV: business value.
- CR: customer retention.
- TF: Developer feature (team familiarity).
- DF: dependency factor.
- Ti: total importance.
- Aw: number of action words.
- Uf: uncertainty factor.
- CX: Complexity factor.
- TOTAL I: estimated time to complete the user story.
- TOTAL E: total effort needed in hours.

Note: bold items are booked for review, clean code and trainings were not included in this study.

During the implementation of the new re-prioritization method, maintained the same order of the items organized in each sprint, in column "O". Then we performed the steps described in the method flow chart to reprioritize the sprints items expect drawing the dependency graph, dependency factor was defined by the PO. A new order (priority) for the items is clearly found in table 8 above.

As a comparison between table 7 and 8, the results are presented in table 9.

Table 9

The new re-prioritized items using our model compared with the old order.

S	Item number	General Priority Value	General Priority Item Order	Proposed priority Item Order	Proposed priority Value
			Sprint A		
A	137390	8.000	137873	137617	60.40
A	138080	7.272	137703	137703	55.60
A	137873	5.333	138231	137873	45.00
A	137435	4.571	137617	138231	38.00
A	138231	4.173	138080	138080	28.15
A	138320	4.000	134764	136139	12.70
A	137617	4.000	136139	137435	3.420
A	137961	3.555	138320	138320	3.208
A	137703	2.560	137961	134339	0.5244
A	134339	1.684	134339	137390	0.491
A	134764	1.230	137390	137961	0.476
A	136139	1.200	137435	134764	0.452
			Sprint B		
B	137435	16.00	139534	138528	69.50
B	137961	12.80	137734	138827	69.50
B	137960	8.000	138725	138725	55.60
B	138941	6.400	137941	137941	45.60
B	138725	5.333	138941	136954	44.50
B	137963	5.333	137963	137734	43.33
B	138418	5.333	137962	137963	29.40
B	137734	5.333	138528	139534	28.00
B	137679	5.333	138827	138418	23.16
B	137735	5.333	137688	137962	21.90
B	137732	5.333	136954	137679	15.40
B	137962	4.000	138418	137735	12.75
B	138528	4.000	137735	137688	11.125
B	138827	3.200	137732	137732	7.700
B	139534	2.782	137960	137606	7.700
B	137941	2.666	137606	137435	3.420
B	137606	2.560	137961	137961	0.476
B	137688	2.285	137679	137960	0.296
B	136954	1.200	137435	138941	0.255
			Sprint C		
C	140101	12.80	137957	139762	55.60
C	137435	9.600	139809	137868	46.333
C	139051	8.888	139692	137957	38.50

C	138013	5.333	139762	139809	29.78
C	140094	4.800	140090	138948	25.66
C	140090	4.705	139970	139975	15.40
C	139809	3.692	139051	137435	5.130
C	139762	3.200	138013	140094	3.266
C	138762	3.200	138948	138013	1.493
C	139691	3.200	139975	139051	1.216
C	139970	2.666	137868	139692	0.970
C	139692	2.000	138762	138762	0.662
C	137957	1.818	140101	140090	0.406
C	138948	1.523	139691	139970	0.380
C	139975	1.454	140094	139691	0.275
C	137868	1.200	137435	140101	0.106

where:

- General priority value: Rank value of priority using (importance/estimated efforts).
- General priority order: Order of items using (importance/estimated efforts).
- Proposed priority value: Rank value of priority using our model.
- Proposed priority order: Order of items using our model.

Note: items 137435 and 137961 were odd as these are remined items that could be moved within more than one sprint to be completed and this case was not covered in our current study.

As the above information in table 9 was shared with Company F, in order to study the new re-prioritized items. Our results concludes that our model might help with re-prioritizing the new ordered items as numbers of order is clearly changed. Questioning the results, can we confirm that our model positively affects the study case?

In fact, if we excluded the odd items, like (137961), the patterns move from the lower changes to the higher changes based on priority changes in Sprint A and B but not Sprint C. That can be explained as the new order of the item is mostly affected by the changes performed before applying our model. The results present enhancements in some of the items, for example, in sprint item 137617 moved from order 7 originally to order 1 after our model. That is accepted since has the higher team familiarity rate (assuming it's 15), neutral dependency factor and lower action words results in lower complexity and less estimated time effort. That means, less risk and higher reliability. Another example, values of item 137617 are similar to item 137703 from the same sprint. Nevertheless, the developer features rank in this item has a lower rank (12) which means less familiarity with the work and may lead to higher risk or complexity. Therefore, the model was helpful in having item 137617 then item 137703 (if we exclude the dependencies between these items. It will be a safer choice for development team to manage this item at first.

Another example is item 138080, this item has DF= 3, the highest value in sprint A, based on the data entered from the PO, this item should be done at earlier stages or before other items are doing to be done. After implementing our model, the item 138080 downed from order 2 into 5. The value seems odd as this should be at the top of the list since it has the higher rank of interdependencies; the value is considerable though, because as it's reported by the PO, item 137390 depends on item 138080.

We hoped to allow Company F to implement our results and check the difference between organized technique to manage the prioritization and checked their backlog changes. However, due to limitation of time, they were not ready to implement our model. However, we shared our model to the PO and during the time of the study these items were considered completed. In order to check if our model help the PO manage the backlogs to minimize the loss or resources -i.e., time, we have created table 10 below to explain the difference between the estimation done before our model -using consensus estimation method and estimation based on the developer features.

Table 10

Difference in items estimation consensus method and our model vs the actual time needed.

S	item #	Estimated Time $_{\text{Consensus}}$	Estimated Time $_{\text{Developer}}$	Actual Time	Actual VS Time $_{\text{Consensus}}$	Actual VS Time $_{\text{Developer}}$
A	137390	26	40	42.74	16.74	2.74
A	138080	11.5	13	14.53	3.03	1.53
A	137873	4	2	1.71	-2.29	-0.29
A	137435	40	50	77.24	37.24	27.24
A	138231	3	3	3	0	0
A	138320	4.5	8	9.45	4.95	1.45
A	137617	3.5	2.5	2.5	-1	0
A	137961	25	35	36.77	11.77	1.77
A	137703	2.2	2.5	2.5	0.3	0
A	134339	19	25	30.83	11.83	5.83
A	134764	16	20	11.58	-4.42	-8.42
A	136139	4	5	6.5	2.5	1.5
B	137435	40	50	77.24	37.24	27.24

B	137961	25	35	36.77	11.77	1.77
B	137960	23	45	80.14	57.14	35.14
B	138941	15	27	27.95	12.95	0.95
B	138725	2	2.5	2.83	0.83	0.33
B	137963	9	5	5.5	-3.5	0.5
B	138418	4	4	4.17	0.17	0.17
B	137734	2.5	3	3.17	0.67	0.17
B	137679	7	5	3.5	-3.5	-1.5
B	137735	4	4	2.33	-1.67	-1.67
B	137732	5	5	8.25	3.25	3.25
B	137962	9	10	11.77	2.77	1.77
B	138528	3	2	2.33	-0.67	0.33
B	138827	3	2	1.75	-1.25	-0.25
B	139534	2	2	2.5	0.5	0.5
B	137941	2.5	2.5	2.5	0	0
B	137606	6	5	4.33	-1.67	-0.67
B	137688	3	4	4.5	1.5	0.5
B	136954	3	2	1.63	-1.37	-0.37
C	140101	44	45	45.42	1.42	0.42
C	137435	40	50	77.24	37.24	27.24
C	139051	13	15	16.28	3.28	1.28
C	138013	15	25	36.64	21.64	11.64
C	140094	22	10	25	3	15
C	140090	10	20	9.9	-0.1	-10.1
C	139809	5	7	8.5	3.5	1.5
C	139762	3	2.5	2.25	-0.75	-0.25
C	138762	8	27	25.22	17.22	-1.78
C	139691	21	23	22.66	1.66	-0.34
C	139970	17	25	26.8	9.8	1.8
C	139692	9	7	11.38	2.38	4.38
C	137957	5	2	1.67	-3.33	-0.33
C	138948	5	3	3.25	-1.75	0.25
C	139975	5	5	4.75	-0.25	-0.25
C	137868	6	3	3	-3	0

Where:

- Estimated Time $_{\text{Consensus}}$: Estimated items time in hours based on consensus method.
- Estimated Time $_{\text{Developer}}$: Estimated items time in hours based on assigned developer -our model.
- Actual Time: Actual items time in hours at completion.

- Actual VS Time $_{\text{Consensus}}$: Actual time – Time $_{\text{Consensus}}$.
- Actual VS Time $_{\text{Developer}}$: Actual time – Time $_{\text{Developer}}$.

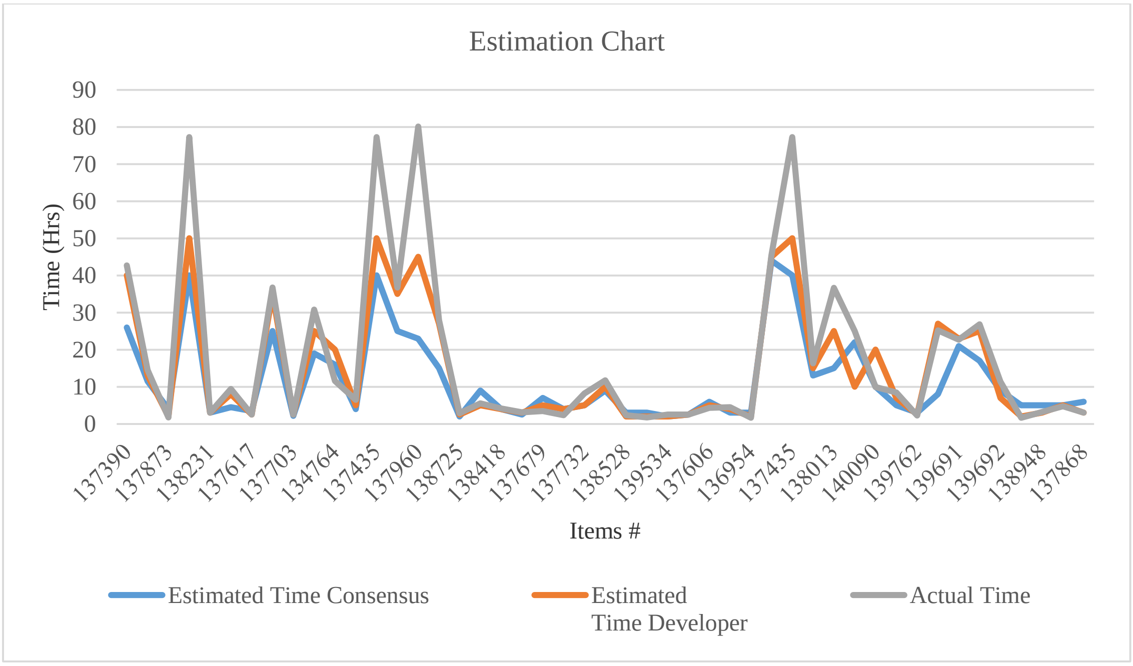
All times based on hours.

Table 10 will prove that our model could be more reliable since we are considering the assigned developer knowledge with the description of the items, consensus estimation does vary a lot and it has a difference gap for some of the items, like item 137390, it has a difference of 16.74 hours, on the other hand, the gap was reduced to 2.74 based on our model estimation. The majority of the items that have difference is zero, where the actual time is equal to the time estimated is based on our model. Nevertheless, there will be some items that will not follow the pattern as item 138013, as it has a relatively difference gap using the two estimation methods.

The estimation chart in figure 7, presents the data of all items estimated times based on consensus method in blue, using our model (focusing on the assigned developer to give the estimation) in orange and the actual time in grey.

Figure 7

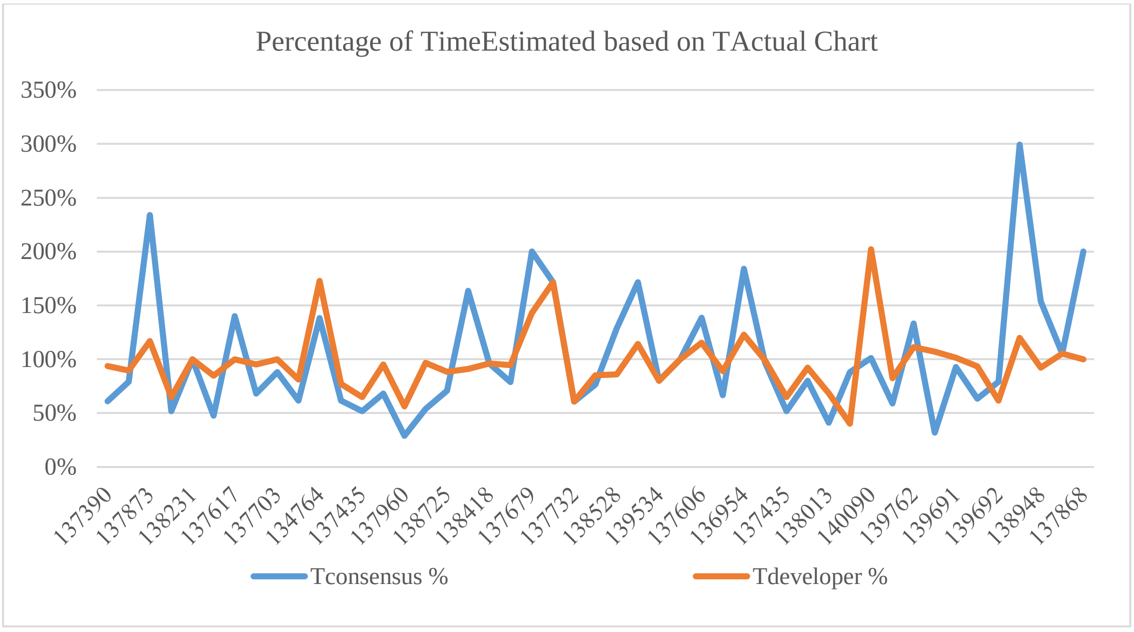
Estimation comparison chart



Given the information of table 10 and chart 8, a percentage charge is created to present the percentage of the Time estimated to the actual time based in consensus method in blue. Also, the percentage of the Time estimated to the actual time based in our model method in orange. It’s noticeably shown that our model is more consistent and the change in time at least is managed to have the closest values to the actual time. That results in less waste of resource based on prioritization general technique.

Figure 8

Percentage of Time$_{Estimated}$ based on T$_{Actual}$ Chart



Company F commented that the values seem encouraging if they started with our model to run their backlogs for the future. They pointed out some of the points in the sprints in general, for example, item 137390, they have changed the priority because of the stakeholder decision, and that left the item to be at the top of the older list though it was dependent on item 138080. Another sample from Sprint B, item 137960, the list went up because of the stakeholder decision only, even it has lower “actual priority as of business value or urgency”.

They also confirmed that this methodology proposed will help them a lot to document the dependencies between the user stories and their description. As a first step in managing the backlog changes due to priority changes resulted from dependencies change and description changes as well. They mentioned that this has been an issue for a while as the scrum does support less documentation, which may lead to weaknesses in managing the backlogs by the development teams or the product owners themselves.

As a summary of our study on this case, we have found that difference could be managed, however, in the introduction it was mentioned, there is no “perfect technique” to manage the priority. We tried to gather the most important factors that were neglected before, or at least not included in formula. Another important finding, a dependency graph is a helpful tool to validate the work for software organizations.

4.2. Discussion

4.2.1 What reasons lead to backlog changes?

Backlog changes is rising issue in software development companies that implement the Scrum or any other Agile methodology. This is caused by the agile environment that welcome change. Reason causing the change are many, and most of them are mentioned in the literature. In addition to the literature, we have asked a couple of product owner about reasons for causing the backlog changes and to define them. first, comes the priority of the user stories, then bug fixes (live bug fixes), then cross functional teams, then stakeholders' decisions.

Other reasons for change mentioned by the development teams, are unclear description of user stories, incorrect estimations. Miscommunication between the team and the PO or between the PO and the stakeholders. Starting the project without notifying the customer with the plan. In correct feedback on completed user stories after time.

As number one change reason we discussed in the priority of the items, as it's the main rank given to the user stories to indicated their importance that also include details of the size or risk of these user stories.

4.2.2. Parameters that affect the changes of the requirements in the backlog

The dominant factor that was found causing the major impact on backlog changes is the priority. Therefore, in order to reduce these changes due to priority changes, we needed to investigate further about the factors impacting the priority. The followings were found:

1- Business value: this value will indicate how much the item is important for the business market and if this will add competitiveness to the product. The business value is defined after the market study done by the PO, in addition to studying other competitors in the same field. The higher the business value, the higher the importance of the item which lead to increase the priority.

2- Customer retention: customers are the most critical factor when maintaining a product or service to succeed. The end user for the product will have feedback about your product and what it needs to be done to make it running for their needs. If they did not find the items applicable to them and the new developments created will help optimizing their process, soon will stop using that product. In addition to the customers' requests that they wish to have and sometimes, they will be willing to pay for development is an important factor for keeping the customer attached to your product. This is also considered when the customer is urgently needs a specific feature. The higher the customer retention value is the higher the priority.

3- Developers features (familiarity and experience): the scrum team is a valuable factor in the process of setting estimations. In sprint planning, the team will only decide the expected efforts, and this is agreed by the majority. Then the team lead will assign the backlog items to the developers. Once missing point here is the estimation and characteristics of the developer or QA team assigned to the item. The experience and familiarity of the items can help the team performing the items in less time and effort, as the developer is already aware of all the cases and may use to build this development before; which means, less errors or bugs and lost time on the known items. Therefore, the higher the developer feature value leads to less uncertainty effort and risks.

4- Dependencies: as scrum supports the less documentation operations, interdependencies within the items may lead to mass loss of resources. There is a lot of dependencies to be considered, specially, with cross functional teams, if two teams' developments is dependent on one another, if it is not discussed while planning and prioritizing the items. Since there are a couple of types of dependencies, each item can affect the priority differently. In this study, direct dependency is considering, as item1, item 2 cannot be done before item 3. Therefore, item 3 should have higher priority than item 1 and item 2. Interdependencies between items have higher priority that independent items.

5- Backlog description language (action words): studies explained that the language used to describe the backlog items can indicate complexity. And mostly recommend to carefully right the item's description. As it's the only documented information related to develop the deliverables to customers. Further action words may mean further uncertainty, this is affecting the total estimation that will be assigned to the items. This also can define the developer’s understanding of what this item needs to be completed. Therefore, more action words, means higher efforts needed.

6- Complexity: the value of complexity is defined by the effort estimated needed which is affected by the size of the item. Big items will definitely take more efforts to be completed. In addition to higher risk, as anything might happen since the team is not aware of the uncertainty, dependencies and bugs that will be added if we did not complete the item correctly. Complexity leads to less priority.

7- Time estimation: wrong estimations is normal to happen when working in scrum environment, efforts needed to complete the items. Normally this process is done in planning when assigning the development of QA tasks. Estimations can be wrong if we did not include the risk, size, the importance of the item or even the features of the development team. Unexperienced developer will not give the approximately accurate estimation as an experienced one. Large estimation leads to less priority. Since the world is moving fast towards new management methodologies as customers’ requests are changing, this definitely will lead to less reliable estimation. Scrum is all about delivering more shippable items faster to customers, which means less reliable estimation as depends on a lot of uncertainties accompany the management of backlogs using scrum.

4.2.3 A conceptual model to reduce the backlog changes during a sprint

In order to propose a new prioritization model for scrum backlog items, we’ve taken into considerations the factors mentioned in Q2 and merged them to a general prioritization technique proposed by (Popli, 2018), following this algorithm:

1- When adding a new requirement, item, bug, user story, make sure that the description is used in a clear format and defined number of action words. In addition to meeting the INVEST criteria.

2- The process starts in the refinement meeting (grooming). In this meeting the team will discuss what items should be done in the next sprint. (Normally the highest orders and priority).

3- Draw a dependency graph gathering those items added in the sprint, in order to define the interdependencies between the items as in figure 4.

4- Checking those higher ordered items and start implementing our proposed model by adding the importance needed in terms of business value, customer retention and developers’ features.
5- Instead of the planning meeting, assign the items to most experienced development based on their features, in terms of familiarity.
6- Calculate the weighted score as of equation (2).
7- Define the dependency factor, the team with help from the developer assigned should be able to rank the items’ dependency factor. In addition to the complexity factor.
8- Calculate the total importance by using the sum of the business value ranks multiplied with the dependency factor as of equation (3).
9- Allow the assignee from the prior point to add the estimated time needed and negotiate the dependency graph. Since they are familiar with the items, they will be able to give the estimation needed as well as the complexity resulted in this user story.
10- Count the action words in the item’s description. Use α for each verb and β for duplicated action words, then multiply by the uncertainty factor as of equation (4).
11- Calculate the total effort by the sum of the complexity factor and the number of action word multiplied with the time estimated as of equation (5).
12- Divide the total importance by the total effort to get the new priority as of equation(6).
13- Order the items based on the new priorities resulted from the proposed model ascending or descending

4.2.4 Backlog conflicts

A question has been raised in this research about backlogs conflicts in user stories. Will our model help managing user stories conflicts in the development process using scrum?

Our study results showed that based on the conflict type between the user stories is managed using our model. In case of dependency conflicts, our model clearly presents an important step in prioritizing the backlog based on dependencies between items. Let’s assume item A and item B; item A is more important that item B (business-wise), however item A depends on B, then based on our model, item B should be completed before item A. This will help the development team choose the items from the backlog that is more important (dependencies-wise). Another example, let’s assume item C and item D, item C has higher priority based on the end user’s feedback, while item D, is more logical for the market study of industry, meaning, it will help a lot of customers not only that end user. In this case, the PO decide to keep item D and might remove item C from the product backlog, as it's specific and not generic. Other cases may arise like, conflict in technologies, the team and PO will need to decide which items will help their product better.

4.3 Limitations

The study was limited to a three- sprint data, knowing that the release is considered to be 5-sprints. Measuring our proposed model results in more sprints will assist us in understanding the frequency of change related during one release development. The team was very helpful in explaining the items and since the product owners were not following any priority complex model, even there is more than 80 software organization locally, we were not able to reach part of them and others were not implementing the scrum. That made our study a little bit hard to check its results and see how much impact was related to the backlog changes. Further limitation is our capability of implementing our model actively on new projects, that was not authorized by the local company which left us with the benchmark project to validate our results, in addition to action words factor generalization, we only took into consideration the number of action work without defining the level of the action word; some actions are more complex.

Chapter Five
Conclusions and Recommendations

Chapter five contains the study conclusions and introduces the recommendations the researcher comes up with after applying the methodology of the study, making sure that the objectives of the study are fully attained.

5.1. Conclusions

In software development project, the customer requirements are the main engine to run the operations. The process of gathering information from client and then managing it throughout the scrum process mainly depends on the product owner or stakeholders with the end user's communication. The main role of the product owner is to receive the requirements requested by the customer or the stakeholder, translate them into user stories or backlog items and then the most important task is to prioritize them based on one of the priority techniques or methods we've mentioned in the literature

In our research, we studied a local software development firm and previous work in order to study the factors affecting the scrum process and leading to backlog changes. Based on our study, the main two reasons for backlog changes are the priority changes and bug fixes. Therefore, we've investigated further to allocate the factors relating the priority and code fixes in order to formulate a new model that will help the product owner re-prioritize the items, or using this formula to prioritize the items directly in the sprint planning phase in addition to involve the scrum team while assigning the priorities to their work, nevertheless it was mentioned that the scrum team expect this task to be done solely by the product owner.

Factors like business value, customer retention, developers' features like familiarity and experience, dependencies, backlog description language (action words), complexity and time estimation. We've implemented our model into a benchmark project, the results were correlating of the dependency graph we've built with the help of an experienced PO. A precise priority and defined dependencies, with the right development tasks assigned to the team members; backlog changes amount will be decreased in reference to priority changes and bug fixes (by assigning the right work to the right developer).

The new model helped the scrum team to achieve the sprint shippable requests in minimum changes in backlogs management.

As future works, we intent to investigate further into scrum artifacts and events that affecting the backlog management changes, and define the dependencies types into the model. Defining a new priority model that is related to less factors but major impact on backlog changes, using more specific category of action words from the natural language in the backlog description.

5.2 Recommendations

After all, the researcher would like to identify the most important recommendations based on the abovementioned findings of the study.

Since the world is moving fast towards new management methodologies as customers' requests are changing, this definitely will lead to less reliable estimation. Scrum is all about delivering more shippable items faster to customers, which means less reliable estimation as depends on a lot of uncertainties accompany the management of backlogs using scrum.

Software organization implements the scrum following their own values and needs. However, it could be following a parent organization, if these are categorized as outsourcing firms. Scrum was studied and implemented to a local firm in (Hannoun & Adhalhaq, 2013) in Palestine with a detailed information about introducing the scrum framework to a local firm. And how the Scrum framework is managed to maintain the agility and help them understand the scrum in depth. During this time, all of the scrum artifacts take a perfect place of timing to make sure at the end the most valuable requests are done.

Despite that the scrum showed its ability to maintain and achieve the project goals perfectly and by the end of each iteration, there will be some certain deliveries to be added to be live to the end user, in which we call product/service release. The process of getting the customer requests and then develop them into increments to be added to the product/service the organizations offering to their clients, could be take a defined amount based on iterations, from 1 to four weeks for each sprint. during this time, all of the Scrum artifacts takes a perfect place of timing to make sure at the end the most valuable requests are done. Therefore, this research will cover the framework of the scrum and what factors are affecting the project flow. In addition, to develop a conceptional model to help software organizations that are using the agile-scrum framework achieve the optimum solution to help them complete the projects with the highest values in return, by proposing a new re-prioritizing model.

1- Listen to the scrum team

Companies focus more on the process than the people and this is one of the main concepts of the agile manifest, Scrum means focusing on communication more that documents, even through less documentation has an effect of quality or requirements, however, scrum organization tend to focus more on running the development that considering the team managing the development. As pointed in the INVEST acronym, items should be negotiable, the this is normally managed between the customer, the PO and the stakeholder, the scrum team is obligate to do the items based on the design provided by the product management team and without consideration of the architecture of the product.

2- Prioritize special developments

Prioritizing income over customer satisfaction is not always leading the companies to win-win state. Paid big developments will take form the customer’s retention developments time. Considering a new development for a new customer that may take 50 developments day, even if the customer is paying for that special development, the other customers using the product may not be interested in that, and it may be not helpful for them. Therefore, this will make the customer who requested the development ready, meanwhile, other customers may feel unattached and no satisfied at this time is already lost doing 1 special development while could has been done a lot of small development helping all customers.

3- Add some error time for uncertainty during a sprint

Due to the scrum nature of accepting changes, and always updating the products; managing risk and uncertainty make it a complex task for the team to measure it. Therefore, add some error time to the sprint capacity in case of any unwanted results without affecting the current development items in the sprint. This process will help relief the team from the stress happened with the done-done rate and allow them sometime to fix older issues in case of no issues discovered during the sprint.

4- Meetings' arrangement

Normally the scrum consists of three basic meetings, the sprint planning, sprint review, sprint retrospective and the review planning. Normally the most critical and longest meeting is the planning meetings, a grooming meeting is needed in this case. Grooming meeting will help reduce the time of the sprint planning meeting which is considered to minimum 4-8 hours based on the sprint iteration. Grooming will help the team discuss the future items to be considered to developments (the highest priority items). Although this meeting reduces the time of the planning, it still takes a lot of time during the scrum process. The scrum team needs to consider only the important items and the most complex ones to be discussed. Familiarity of this point is important in reference of the developer features. Once the team is well known of the item, it will take less time to be discussed which means it's clear and understood, therefore, a result of a better product.

List of Abbreviation

Abbreviation	Meaning
SADT	Structured Analysis and Design Technique
AHP	Analytical Hierarchy Process
Aw	Action words
BST	Binary Search Tree
DF	Dependencies factor
DOD	Definition of Done
DSDM	Dynamic Systems Development Method
DT	Development Team
E	Effort
FDD	Feature-Driven Development
I	Importance
MOSCow	Must have, Should have, Could have, Won't have
P	Priority
PO	Product Owner
QFD	Quality functional deployment
SA/SD	Structured Analysis/Structured Design
SM	Scrum Master
ST	Scrum Team
Te	Total Effort
Ti	Total Importance
Uf	Uncertainty factor
US	User story
VOP	Value-Oriented Prioritization
XP	Extreme Programming

References

[1] product prioritization frameworks for product managers. (2021). Retrieved 25 November 2021, from https://2u.pw/qBqff

[2] Alsalemi, A., & Yeoh, E. (2015). A survey on product backlog change management and requirement traceability in agile (Scrum). 2015 9Th Malaysian Software Engineering Conference (Mysec). https://2u.pw/D5aDA

[3] Alsalemi,A. M. & Yeoh, E. (2017). "A Systematic Literature Review of Requirements Volatility Prediction," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), 2017, pp. 55-64, doi: 10.1109/CTCEEC.2017.8455174

[4] Ananjeva, A., Persson, J., & Bruun, A. (2020). Integrating UX work with agile development through user stories: An action research study in a small software company. Journal Of Systems And Software, 170, 110785. doi: 10.1016/j.jss.2020.110785

[5] Bakhtiar A., Hannan A., Basit A., & Ahmad J., “Prioritization of Value Based Services of Software By Using Ahp and Fuzzy Kano Model,” Int. Conf. Comput. Soc. Sci., no. August, pp. 48–56, 2015.

[6] Behutiye, W., Seppänen, P., Rodríguez, P., & Oivo, M. (2020). Documentation of Quality Requirements in Agile Software Development. Proceedings Of The Evaluation And Assessment In Software Engineering. https://2u.pw/D5aDA

[7] Castillo-Barrera, F., Amador-Garcia, M., Perez-Gonzalez, H., & Martinez-Perez, F. (2017). Agile Evaluation of the Complexity of User Stories Using the Bloom's Taxonomy. 2017 International Conference On Computational Science And Computational Intelligence (CSCI). doi: 10.1109/csci.2017.182

[8] Chopade, M., & Dhavase, N. (2017). Agile software development: Positive and negative user stories. 2017 2Nd International Conference For Convergence In Technology (I2CT). doi: 10.1109/i2ct.2017.8226139

[9] Ciric, D., Lalica, B., Gracanin, D., Tasic, N., Delic, M., & Medic, N. (2019). Agile vs. Traditional Approach in Project Management: Strategies, Challenges and Reasons to Introduce Agile. Procedia Manufacturing, (39). https://2u.pw/zhH8E

[10] Cohn, M. (2005). Agile Estimating and Planning. Upper Saddle River, NJ, USA: Prentice Hall. ISBN: 0131479415

[11] Common Scrum Pitfalls | SCRUMstudy Blog. (2022). Retrieved 8 February 2022, from https://blog.scrumstudy.com/common-scrum-pitfalls/

[12] Company F, Ramallah, Palestine (April -June 2021)

[13] Ereiz, Z., & Music, D. (2019). Scrum Without a Scrum Master. 2019 IEEE International Conference On Computer Science And Educational Informatization (CSEI). doi: 10.1109/csei47661.2019.8938877

[14] Fibonacci sequence. Math is Fun. (2020). Retrieved February 8, 2022, from https://2u.pw/vqbrS

[15] Gandomani, T., Faraji, H., & Radnejad, M. (2019). Planning Poker in cost estimation in Agile methods: Averaging Vs. Consensus. 2019 5Th Conference On Knowledge Based Engineering And Innovation (KBEI). https://2u.pw/TeE5X

[16] Ghosh, S., Ramaswamy, S., & Jetley, R. (2013). Towards Requirements Change Decision Support. 2013 20Th Asia-Pacific Software Engineering Conference (APSEC). https://doi.org/10.1109/apsec.2013.30

[17] Golfarelli, M., Rizzi, S., & Turricchia, E. (2013). Multi-sprint planning and smooth replanning: An optimization model. Journal Of Systems And Software, 86(9), 2357-2370. doi: 10.1016/j.jss.2013.04.028

[18] Gonen, B., & Sawant, D. (2020). Significance of Agile Software Development and SQA Powered by Automation. 2020 3Rd International Conference On Information And Computer Technologies (ICICT). https://2u.pw/FGDPP

[19] Hammad, M., Inayat, I., & Zahid, M. (2019). Risk Management in Agile Software Development: A Survey. 2019 International Conference On Frontiers Of Information Technology (FIT). https://doi.org/10.1109/fit47737.2019.00039

[20] Hannoun, A., & Abdalhaq, D. (2013). Introducing Agile Software Development Methodology (Scrum) Into a Software Development Project at a Local Firm. An-Najah National University. Retrieved from https://2u.pw/PK72U

[21] Hector, C. (2020). Introduction to Scrum. Code Magazine. Retrieved from https://2u.pw/l9plm

[22] Hovorushchenko, T., & Pomorova, O. (2018). Methodology of evaluating the sufficiency of information on quality in the software requirements specifications. 2018 IEEE 9Th International Conference On Dependable Systems, Services And Technologies (DESSERT). doi: 10.1109/dessert.2018.8409161Software: Evolution And Process, 32(7). doi: 10.1002/smr.2247

[23] How to Refine Product Backlog?. (2021). Retrieved 14 October 2021, from https://2u.pw/ldSHM

[24] Hudda, S., Mahajan, R., & Chopra, S. (2016). Prioritization of User-Stories in Agile Environment. Indian Journal Of Science And Technology, 9(45). https://doi.org/10.17485/ijst/2016/v9i45/105069

[25] Increment. Scrum Dictionary. (2018, November 30). Retrieved February 8, 2022, from https://scrumdictionary.com/term/increment/

[26] Introduction To Agile Methodology. (2021). Retrieved 13 November 2021, from https://chercher.tech/jira/agile-methodology

[27] J. Stapleton, DSDM: dynamic systems development method, in: TOOLS '99: Proceedings of the Technology of Object-Oriented Languages and Systems, IEEE Computer Society, Washington, DC, USA, 1999. p. 406.

[28] Jansi, S., & Rajeswari, C. (2015). A Greedy Heuristic Approach for Sprint Planning in Agile Software Development.

[29] K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley Longman Publishing Co., Inc., Boston, USA, 1999

[30] K. Molokken-Ostvold and N. C. Haugen, "Combining Estimates with Planning Poker--An Empirical Study," 2007 Australian Software Engineering Conference (ASWEC'07), 2007, pp. 349-358, doi: 10.1109/ASWEC.2007.15.

[31] K. Schwaber, SCRUM development process, in: Proceedings of the Conference on Object-Oriented Programing Systems, Languages, and Applications Workshop on Business Object Design and Implementation, 1995, pp. 117–134

[32] Lopez-Martinez, J., Juarez-Ramirez, R., Huertas, C., Jimenez, S., & Guerra-Garcia, C. (2016). Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review. 2016 4Th International Conference In Software Engineering Research And Innovation (CONISOFT). https://2u.pw/FGDPP

[33] Manifesto for Agile Software Development. (2022). Retrieved 8 February 2022, from https://agilemanifesto.org/

[34] Masood, M., Azam, F., Anwar, M., & Amjad, A. (2019). Defining Meta-Model for Value-Oriented Requirement Prioritization Technique. Proceedings Of The 2019 7Th International Conference On Computer And Communications Management. https://doi.org/10.1145/3348445.3352739

[35] Miller, K.W., Larson, D.K., 2005. Agile software development: Human values and culture. IEEE Technol. Soc. Mag. 24, 36–42. http://dx.doi.org/10.1109/MTAS.2005.1563500.

[36] Moe, N.B., Dingsøyr, T., Dybå, T., 2010. A teamwork model for understanding an agile team: A case study of a Scrum project. Inf. Softw. Technol. 52, 480–491. http://dx.doi.org/10.1016/j.infsof.2009.11.004

[37] MoSCoW Prioritization. (2021). Retrieved 25 November 2021, from https://www.productplan.com/glossary/moscow-prioritization

[38] Nuseibeh, B., and Easterbrook, S., “Requirements engineering: a roadmap,” Proc. Conf. Futur. Softw. Eng. - ICSE ’00, vol. 1, pp. 35–46, 2000

[39] P. Abrahamsson, J. Warsta, M.T. Siponen, J. Ronkainen, New directions on agile methods: a comparative analysis, in: Proceedings of the International Conference on Software Engineering, 2003, p. 244.

[40] Parente, “Bridging the Gap: Traditional to Agile Project Management,” Pm.Umd.Edu, vol. IV, no. Ix, pp. 1–12, 2015

[41] Paul E McMahon., Pan-Wei Ng., Harold Bud Lawson., Ivar Jacobson., & MIchael Goedicke. (2019). The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons!. Association for Computing Machinery and Morgan & Claypool.

[42] Pokharel, P., & Vaidya, P. (2020). A Study of User Story in Practice. 2020 International Conference On Data Analytics For Business And Industry: Way Towards A Sustainable Economy (ICDABI). doi:10.1109/icdabi51230.2020.9325670

[43] Popli, R., & Chauhan, N. (2018). A Sprint Point Based Tool for Agile Estimation. Advances In Intelligent Systems And Computing, 63-72. doi: 10.1007/978-981-10-8848-3_6

[44] Rahim, S., Hasan, M.H., Chowdhury, A.E., & Das, S. (2017). Software Engineering Practices and Challenges in Bangladesh: A Preliminary Survey. Journal of Telecommunication, Electronic and Computer Engineering, 9, 163-169.

[45] Rida, A., Nazir, S., Tabassum, A., & Asim, S. (2017). The Impact of Analytical Assessment of Requirements Prioritization Models: An Empirical Study. International Journal Of Advanced Computer Science And Applications, 8(2). doi: 10.14569/ijacsa.2017.080240

[46] Rodeghero, P., Jiang, S., Armaly, A., & McMillan, C. (2017). Detecting User Story Information in Developer-Client Conversations to Generate Extractive Summaries. 2017 IEEE/ACM 39Th International Conference On Software Engineering (ICSE). https://doi.org/10.1109/icse.2017.13

[47] Rubin, K. S. (2013). Essential scrum: A practical guide to the most popular agile process. Addison-Wesley.

[48] S.R. Palmer, J.M. Felsing, A Practical Guide to Feature-Driven Development, Prentice Hall, 2002

[49] Saeeda H., Dong, J., Wang, Y., & Abid, M. (2020). A proposed framework for improved software requirements elicitation process in SCRUM: Implementation by a real- life Norway- based IT project. Journal Of Software: Evolution And Process, 32(7). doi: 10.1002/smr.2247

[50] Saher, N., Baharom, F., and Romali, R., (2020). Guideline for the Selection of Requirement Prioritization Techniques in Agile Software Development: An Empirical Research. International Journal Of Recent Technology And Engineering, 8(5), 3381-3388. doi: 10.35940/ijrte.e6634.018520

[51] Saurabh, S., Sagnika, S., Mishra, S., & Das, M. (2017). A Systematic Review on Software Cost Estimation in Agile Software Development. JOURNAL OF ENGINEERING SCIENCE AND TECHNOLOGY REVIEW, 10(4), 51-64. doi: 10.25103/jestr.104.08

[52] Scheerer, A., Bick, S., Hildenbrand, T., & Heinzl, A. (2015). The Effects of Team Backlog Dependencies on Agile Multiteam Systems: A Graph Theoretical Approach. 2015 48Th Hawaii International Conference On System Sciences. https://doi.org/10.1109/hicss.2015.606

[53] Schwaber, K., Beedle, M., 2002. Agile software development with Scrum, Prentice Hall

[54] Schwaber,K., Jeff Sutherland, J., 2020. The Scrum Guide, Scrum.org

[55] Scott, E., & Pfahl, D. (2018). Using developers' features to estimate story points. Proceedings Of The 2018 International Conference On Software And System Process. https://doi.org/10.1145/3202710.3203160

[56] Scrum Alliance. (2021). STATE OF SCRUM 2017-2018. Scrum Alliance.

[57] Sedano, T., Ralph, P., & Peraire, C. (2019). The Product Backlog. 2019 IEEE/ACM 41St International Conference On Software Engineering (ICSE). https://2u.pw/FGDPP

[58] Sheemar, H., & Kour, G. (2017). Enhancing User-Stories Prioritization Process in Agile Environment. 2017 International Conference On Innovations In Control, Communication And Information Systems (ICICCI). https://doi.org/10.1109/iciccis.2017.8660760

[59] Sher, F., Jawawi, D., Mohamad, R., & Babar, M. (2014). Requirement's prioritization techniques and different aspects for prioritization a systematic literature review protocol. 2014 8Th. Malaysian Software Engineering Conference (Mysec). doi: 10.1109/mysec.2014.6985985

[60] Sommerville, (2004). Requirements Engineering Processes," Software. Engineering. 7th Ed. Chapter 7, pp. 1–52.

[61] The scrum framework poster. Scrum.org. (n.d.). Retrieved February 8, 2022, from https://www.scrum.org/resources/scrum-framework-poster

[62] Unger-Windeler, C., &amp; Schneider, K. (2019). Expectations on the product owner role in Systems Engineering - A Scrum team's point of View. 2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). https://doi.org/10.1109/seaa.2019.00050

[63] Value based prioritization in Scrum | SCRUMstudy. (2021). Retrieved 25 November 2021, from https://www.scrumstudy.com/whyscrum/scrum-value-based-priotirization

[64] What does INVEST Stand For?. (2021). Retrieved 3 October 2021, from https://www.agilealliance.org/glossary/invest/

[65] Yaseen, M., Ibrahim, N., & Mustapha, A. (2019). Requirements Prioritization and using Iteration Model for Successful Implementation of Requirements. International Journal Of Advanced Computer Science And Applications, 10(1). doi: 10.14569/ijacsa.2019.0100115

[66] Zacarias, D. (2020). 20 Product Prioritization Techniques: A Map and Guided Tour | Briefings | career.pm. Retrieved 14 October 2021, from https://2u.pw/Y5rl9

[67] Zahraoui, H., & Janati Idrissi, M. (2015). Adjusting story points calculation in scrum effort & time estimation. 2015 10Th International Conference On Intelligent Systems: Theories And Applications (SITA). https://2u.pw/FGDPP

جامعة النجاح الوطنية

كلية الدراسات العليا

تقليل التغيرات في متطلبات مشاريع هندسة البرمجيات في الشركات المستخدمة لنظام الأجايل سكرم

إعداد

أزهار عمر غانم

إشراف

د. أحمد عواد

قدمت هذه الأطروحة استكمالاً لمتطلبات الحصول على درجة الماجستير في الإدارة الهندسية بكلية الدراسات العليا في جامعة النجاح الوطنية في نابلس –فلسطين.

2022

تقليل التغيرات في متطلبات مشاريع هندسة البرمجيات في الشركات المستخدمة لنظام الأجايل سكرم

إعداد

أزهار عمر غانم

إشراف

د. أحمد عواد

الملخص

خلفية الدراسة: تتمثل متطلبات هندسة البرمجيات في ترجمة إحتياجات المنتج أو الميزات المطلوبة من قبل العملاء وأصحاب المصلحة. وبسبب التطور التكنولوجي، فإن هذه الطلبات تتغير دائماً. لذا يجب إدارة هذه المتطلبات بعناية لمساعدة العملاء والمؤسسات على تحقيق أهدافها. قد استُحدث نظام "الأجايل – سكرم" كمنهجية لإدارة المشاريع تركز على إتصالات العملاء أولا وعلى إتصالات الفريق بدلا من التوثيق المُركّز لهذه المتطلبات.

أهداف الدراسة: منهجية "سكرم" تتميز في تقديم الطلبات و الاحتياجات بشكل أسرع إلى العملاء وقبول التغييرات التي تطرأ على هذه الطلبات، إلا أنه يوجد قدر أقل موثوقية للموارد (مثل الوقت والمال والقوى العاملة وما إلى ذلك). تعتمد معلومات هذه المتطلبات على الكثير من المعلومات و النتائج الغير مؤكدة الموروثة في عملية إدارة المتطلبات باستخدام منهجية "سكرم". لذلك، يجب تقليل هذه التغييرات في المتطلبات عن طريق تطوير نموذج لتحديد أولويات المتطلبات البرمحجية و ترتيبها.

منهجية الدراسة: أجريت مقابلات غير منظمة مع خمسة موظفين كمُسمّى "برودكت اونر – مالكي المنتج" في شركتين محليتين تنفذان إجراءات "سكرم" كمنهجية لإدارة عملية تطوير البرمجيات.

ونتيجة لذلك، تم تحديد العوامل التي تؤثر على التغييرات في معلومات المتطلبات، ثم تم تطوير نموذج لتحديد الأولويات لمتطلبات البرمجيات و ترتيبها لمساعدة مؤسسات هندسة البرمجيات على إدارة المتطلبات لديها بشكل فعال، وللتقليل إلى أدنى حد من الخسائر الناجمة عن التغيرات المستمرة في إدارة المتطلبات.

نتائج الدراسة: أظهرت النتائج أن التغييرات في الأولوية هي أكثر العوامل هيمنة التي تؤثر على التغيرات في المتطلبات. على عكس النماذج السابقة لحساب متطلبات الأولويات، وُجِد أنّ التبعيات بين العناصر، و عدد كلمات الفعل في الوصف اللغوي للمتطلبات و أيضاً ميّزات الشخص المبرمج المسؤول تُؤثر بشكل كبير على ترتيب الأولوية. فإنّ النموذج المطوّر هو بمثابة أداة فعّالة لتعبين الأولويات خلال مرحلة التخطيط في إجراءات منهجية "سكرم".

الاستنتاجات: لقد قمنا بإثبات نجاح ومصداقية النموذج الذي اقتُرِح بإجراء حالة دراسية و تجربة في إحدى الشركات المحلية التي تقوم بتطبيق منهجية "سكرم" و مشروع مرجعي. أشارت النتائج إلى أن التغيير في الوقت يقلّ بشكل ملحوظ عند تطبيق النموذج المتقرح على الطرق السابقة لتعيين الأولويات للمتطلبات على النموذج العام المُتّبع عامةً. بالإضافة إلى ذلك، أظهر النموذج المقترح تأثيرا جيدا على مهام عملية إدارة المتطلبات للقائمين بها.

الكلمات المفتاحية: أجايل، سكرم، ادارة تغيير المتطلبات، اولويات المتطلبات.