

Universiteit Leiden

ICT in Business and the Public Sector

Scrumban from a practitioners' point of view

Name:

Moaid Homsi Student-no: S2355442

Date: 28/10/2022

1st supervisor: Dr. Werner Heijstek 2nd supervisor: Prof.Dr.Ir. Joost Visser.

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS) Leiden University Niels Bohrweg 1 2333 CA Leiden The Netherlands

List of Figures	
List of tables	
Acknowledgments	5
Abstract	5
Chapter 1 Introduction	
1.1. Overview	
1.2. Problem Statement	9
1.3 Research gap	
1.4 The research question	
1.5 Research Objectives	
1.6 Research relevance	
Chapter 2 Research Methodology and Design	14
2. The Research design and methodology	
2.2. Literature research study	14
2.3 The primary research	
2.4 Interviews design	
2.5 Sampling and Expert Selection	
2.6. The Analysis:	
Data Reduction	
Data Display	
Drawing and verifying conclusions	
Chapter 3. Theoretical Background and literature review	20
3.1 Software development process evolution:	
3.2 Scrum methodology	
3.3 Kanban methodology	
3.4 Scrumban	
3.4.1 Scrumban practices and applied principles in literature	
3.4.2. Related Work	
Chapter 4 The Results	45
4.1- Scrumban Application	45
4.1.1. Teams' compositions	
4.1.2. Delivery mechanism	
4.1.3. Meetings and ceremonies	
4.1.4. Working mechanism:	
4.1.5. Metrics	

Table of Contents

4.1.6. Conclusion on Scrumban Application	55
4.2 Scrumban components representation model	56
4.3 Scrumban perceived benefits	59
Chapter 5 Discussion	61
5.1. Selector's definition and work mechanism	61
5.1.1 Scrumban Team Roles selector	62
5.1.2 Scrumban team specialized or cross-functional selector	64
Chapter 6 Conclusion	68
6.1- Conclusion	68
6.2 Summary	68
6.3 Future work	68
6.3.1 Applying quantitative studies and analysis	68
6.3.2 Scrumban (influencing) success factors studies	69
6.4 Outcome Vs Objective	69
6.5 Research contributions	69
6.6 Limitations	
Bibliography	72
Appendix A: Interview questions	
Appendix B: The practice, praxis matrixes	81
Delivery mechanism 2	81
Meetings and ceremonies	
Working mechanism	83
Metrics	
Appendix C: The selectors	85
Team size adjustor	85
Sprints/ Workflow selector	
Estimation selector	86
Meetings selector	87
Metrics selector	87

List of Figures

Figure 1: Agile frameworks adoption. Adapted from. Adapted from [13]	. 10
Figure 2: Agile method adoption indicator values between 2014 and 2021 in State of Agile"	
reports [13]	. 11
Figure 3: Research methodology stages	. 14
Figure 4: practice theory framework, praxis, practices, and practitioners. Adapted from [26]	. 18
Figure 5: The matrix used Praxis, Practices matrix	. 19
Figure 6: The waterfall model. Adapted from [29]	. 20
Figure 7: Comparison between Agile and Waterfall Approaches. Adapted from [3]	. 21
Figure 8: Scrumban metrics. Adapted from [6].	. 35
Figure 9: practice, praxis teams' compositions matrix	. 47
Figure 10: practice, praxis delivery mechanism (1) matrix	. 51
Figure 11: The allocation model of Scrumban key components between Scrum and Kanban	. 58
Figure 12: Scrumban role selector	. 64
Figure 13: Scrumban specialized / Cross functional team selector.	. 67
Figure 14: practice, praxis delivery mechanism matrix	. 81
Figure 15: practice, praxis meetings matrix	. 82
Figure 16: practice, praxis working mechanism matrix	. 83
Figure 17: practice, praxis metrics matrix.	. 84
Figure 18: Team size adjustor	. 85
Figure 19: Sprint/ Workflow selector.	. 86
Figure 20: Estimation selector.	. 86
Figure 21: Meetings selector	. 87
Figure 22: Metrics selector.	. 87

List of tables

Table I: Commonly used agile frameworks [3]:	8
Table II: participants of the interviews:	17
Table III: Related work overview:	38

Acknowledgments

As Prophet Muhammad, peace and blessings be upon him, once said, "Whoever does not thank people has not thanked Allah." (Ahmad, Tirmidhi). So, I would like to thank people who without their support, the journey of obtaining this Master has not been possible.

First and foremost, I would like to express my deep sense of thanks to my primary supervisor Dr. Werner Heijstek and to my second supervisor Dr. Joost Visser, for not only their continuous support and guidance while writing this thesis, but also for the learning opportunities provided by them.

Also, my thanks are to all the Agile experts that I have interviewed during this research for their useful insights and incredibly interesting conversations we had, and especially to my friend and example of success Rasheed Raya who introduced me to this amazing Agile community.

Whatever I say, I cannot begin to express my sincere thanks to my family, especially my father Riad and mother Malak, for the unconditional support, prayer they have sent my way along this life.

Also, my genuine gratitude to my amazing siblings Huda and Homam, and my friends Ammar Hussam and Malik for their motivation and care throughout this journey. This gratitude is extended to my loving wife Hadeel, who has been always devotedly by my side, not only during this mater's journey but also so far in this life, in rain and shine. To my children, Riad, and our newborn Amir, being in my life has always given me the strength I need to realize my goals.

24 October, 2022.

Moaid Homsi.

Abstract

Context - Agile methodologies have generally been used in order to develop software frequently, in small increments, done in cycles or iterations with frequent feedback loops. Recently, there has been an increased interest in hybrid software development methodologies, which combine different practices from two or more frameworks, such as applying Scrum events with Kanban workflow visualization. The aim is to have a final enhanced framework, in order to keep up with the ever-changing markets and customer requirements. Scrumban is a hybrid methodology that combines Scrum and Kanban, where Scrumban team members select useful practices from both methodologies to meet the team-specific requirements.

Problem - Scrumban is introduced in some theoretical sources as a method that combines the best practices of Scrum and Kanban regarding the team's situation, and a promising framework based on its flexibility in dealing with the ever-changing requirements. However, although Scrumban is a commonly used framework, it has a low adoption level of 9%, in comparison with other frameworks like Scrum 66%, according to the latest "State of Agile Report". Moreover, during the literature study, it was found that Scrumban has no definitive guide or definition and that experts do not agree on what Scrumban really is. Additionally, problematic points in the literature were discovered. Thus, it is safe to say that Scrumban is inadequately understood throughout Lean and Agile communities.

Objective - The objective of this study is to explore the available theoretical and literature sources and to identify the key problematic areas and difficulties around Scrumban, that may confuse Scrumban application and may hinder its adoption both theoretically and practically. Afterwards, to explore how Scrumban can be applied successfully by agile experts and teams using agile practitioner's points of view. Also, to find a way that can be used to group the identified problematic contradictory areas in Scrumban theory against the practitioner's viewpoint, then design a framework that can be used as a tool to contribute to improving Scrumban adoption level.

Method - In this thesis a literature study was performed using relevant trusted sources. That includes most of the problematic areas. After that, a qualitative approach was adopted, and seven semi-structured in-depth interviews took place, with Scrumban practitioners and agile experts, in order to discuss those seven dimensions. Then seven matrixes were established using the "Strategy

As Practice" as research approach, to compare the definition of the under-consideration practice in the theory sources (practice) against the interviews (praxis).

Findings - Based on the literature study, seven dimensions of Scrumban were identified: Team, delivery mechanism, working mechanism, meetings, planning, documentation, and metrics. Similarly, it was found that most of the Scrumban research reported by the previous literature was applied within special organization types and for specific applications. The established Scrumban matrixes, have disclosed some disagreements between the considered theory sources, and have displayed some differences between practitioners' viewpoints as opposed to the theory sources. For example, Scrumban theory sources do not require team roles, whereas practitioners emphasize keeping Scrum roles. Another example is where some of Scrumban's theory references indicate that the estimation should be optional and other resources suggest that it even can be problematic. However, practitioners' interviews show that estimation is used.

Moreover, and also based on the established Scrumban matrixes, it was found that Scrumban comprises a high degree of flexibility and a large variety of possible combinations. This flexibility can be worrying, especially for teams that are new to agile and do not have an internal coach. Based on this need, "Selectors" were created, in order to provide Scrumban practitioners with a tool that can be used as an assistance to support optimal selection between Scrum and Kanban practices.

Keywords — Scrumban; Kanban; Scrum; Agile methodologies, framework.

Chapter 1 Introduction

1.1. Overview

In modern software development, it became apparent that techniques like Waterfall have been pushed out of the mainstream, while new processes based on Agile have gained traction [1]. Agile methods are defined as iterative methods that deliver the whole product in increments, and it aims to strengthen the interaction among the self-organizing team [2]. Based on the essential concepts of Agile many approaches have emerged. Examples are XP (Extreme Programming), Scrum, Lean Development, Crystal and Dynamic Systems Development Method (DSDM) [3]. Those approaches can be broken down into three types of agile methodologies as shown in the following table:

Group Name	Frameworks		
	Feature Driven Development (FDD), Kanban,		
Individual team framework	Scrum,		
	Disciplined Agile (DA), Dynamic Systems		
Agile at Scale frameworks	Development Method (DSDM), Large Scale Scrum		
	(LeSS), Scaled Agile Framework (SAFe)		
	Scrumban, (A combination of Scrum and Kanban)		
Hybrid framework			
	Crystal, DevOps, Iterative Development, Lean		
Related frameworks	Software Development		

Table I: Commonly used agile frameworks [3]:

In recent years, there has been a rising interest in combining different agile development methodologies or frameworks. The successful implementation of such a harmonizing approach requires combining best practices among those agile methodologies. Scrumban is a hybrid methodology that combines Scrum and Kanban. Scrum and Kanban are two of the most frequently used agile methodologies according to an empirical investigation [4]. Scrum commonly is the most used methodology in software development [5], which has a rich set of guidelines, and where project management is comprehended in its practices [1]. Some experts also go further and

consider it as one of the best agile methods for software development, since it has more control over the total flow than any other methodology [1]. Kanban, on the other hand, Kanban was mainly developed to function in manufacturing environments. It is a Lean approach and as such uses Lean principles. While Scrum concept is to work on one iteration at a time, Kanban adopts working in a continuous workflow [2], but also other principles, for example, restraining project commitments that result in project failure encourages having policies like just-in-time planning (which translates to "just-in-time" inventory in manufacturing terms), as well as imposing limits to the amount of work in progress (WIP) [6].

Scrumban is expressed as a hybrid agile methodology designed to deal with changing customer requirements and frequent coding problems [4]. A framework is a straightforward foundation used to steer the user, rather than a prescriptive process [3]. Frameworks differ from each other by the activities and the special terms they use to describe their processes, nonetheless, frameworks are not mutually exclusive making it possible to be brought together [3]. Apart from the frameworks and methodologies used, any combination must be used in accordance with the manifesto and Agile principles [3].

Scrumban combines the best practices of Scrum, with a Kanban-style pull-driven mechanism and limiting work-in-progress (WIP). However, Scrumban still has problematic issues according to some software practitioners like the confusion that might emerge of such a merge. For instance, the confusion that may arise after replacing sprints with Kanban workflow, which may in turn leads to some waiting loops within the team, of not having sprints, and that some team members have to wait for another preceding process [4].

1.2. Problem Statement

Agile teams' selection for the appropriate framework is grounded on factors like the project requirements, but also on the culture and structure of the organization [3]. In the case of software development, the System Development Life Cycle (SDLC) can use various methods such as waterfall, iterative, rapid application development (RAD), agile, etc. [7].

In terms of adopting Scrumban, in most cases, Agile teams look for the appropriate practices from the Kanban framework in order to enhance the Scrum framework by omitting some of its practices and adopting appropriate ones from Kanban [8]. In this way, the aim is that the appropriate practices of both methods are adopted based on different situations to meet the organization's needs [8].

Despite the noticeable development in Scrumban as a framework, it lacks a conclusive guide or specific definition. Actually, many reliable sources, disagree about what Scrumban implies [6].

Inconsistencies in Scrumban literature will lead to a confused adopter, affecting the expected performance [9]. As an example, including but not limited to, the adoption of the Scrum "Role" in Scrumban seems to have multiple recommendations in the literature. Some saw that Scrumban is like Kanban and does not need roles for its success [10]. Other opinions see that "Roles" are inherited from Scrum so they should be applied [11] [12]. Likewise, other experts saw that it should be left up to the team to decide on using them [8], but did not provide a practical method for the team to use.

Similarly, although Scrumban method is generally accepted, there is some debate as to how this method is applied in practice. State Of Agile 15th report [13], the continuous annual series of reviews of agile techniques and practices, shows that the Scrum method is highlighted as the most popular agile approach with 66%, while Scrumban is used with 9%, as illustrated in Figure (1).



Figure 1: Agile frameworks adoption. Adapted from. Adapted from [13].

Tracking down the this indicator between 2014 and 2021 in the series of "State Of Agile" annual reports, it is found that an increase from 40% in 2014 to 66% in 2021 for Scrum method, while Scrumban is still in lower levels with the adoption indicator value is increasing from 7% in 2014 to 10% in 2020 then finally lower to 9% in 2021 (15th state of Agile report) [13]. Figure (2) 10

displays the maturity of Agile method adoption indicator values between 2014 and 2021 for Scrum and Kanban and Scrumban.

By combining the two frameworks' experts are expecting Scrumban to be a more flexible framework, especially since those two frameworks, Scrum and Kanban, mostly complement each other [2] [14]. That is why it is expected to take precedence over both Scrum and Kanban [2]. For example, Scrumban integrates the iterative planning of Scrum, but it is more open and adaptive to changes in user requirements [2]. It is clear that the percentage of adoption is not improving over the years. In other words, the adoption of Scrumban is still lacking behind and that's proven by the previously mentioned numbers in the "State of Agile" reports.



Figure 2: Agile method adoption indicator values between 2014 and 2021 in State of Agile" reports [13]

Since Scrumban adoption reveals a poor level of "indication of the maturity" among Agile methods adoption, and in light of the scarcity of the literature, these all motivate the need for deeper research in this regard. It is believed that, reflecting on how Scrumban is currently applied in practice and shining some light on how the gap between the theory and practice should be filled, will contribute to a better understanding of the framework and as such will contribute to a better understanding of the practices and artifacts of Scrumban.

1.3 Research gap

Literature shows a wide difference as far as Scrumban is considered. Although Scrumban is not merely about combining elements of the two frameworks, rather it highlights the application of cooperating Kanban systems within a Scrum context and finding the synchronization between both 11

methods to get the evolutionary change needed [6], most studies tend to combine particular practices of Scrum and Kanban and adjust them accordingly to fit the organizations' needs and requirements. Thus, in the literature, it is noticed that the resulting Scrumban frameworks were customized to fit that specific situation or the special sittings at hand. Therefore, the resulting hybrid frameworks were not generalizable enough to be adopted in other settings or different situations. This may also justify the inconsistencies found in the literature about the practices of this framework as well as the current level of Scrumban adoption in recent years. As a result, the literature lacks a practical general Scrumban guidance that specifies what are the constants and the variables that can determine the main features of Scrumban, and on which base should those variables or practices be decided.

Generally, it seems that there is a lot of vagueness in the way in which Scrumban is planned and implemented. This research aims to examine Scrumban from the viewpoint of practitioners in order to address answers to why is Scrumban considered, from theory viewpoint, a highly strong agile framework and in reality, it still has a relative stable poor adoption level, uncover and reveal the difference between the theory and practice of Scrumban aspects, and identifying solutions that will contribute to providing new guidelines that can help to improve Scrumban application.

1.4 The research question

In light of the scarcity of supporting material and the inconsistencies among resources about the applied Scrumban practices, the main question for this research is:

How is Scrumban being applied from the practitioner's point of view?

To answer this question adequately, it can be broken down into the following three questions:

- How is Scrumban being applied in practice by practitioners?
- What is the basis of the Scrumban combination?
- What are the perceived benefits of Scrumban for practitioners?

Answering the research question adequately requires setting appropriate research objectives and a plan.

1.5 Research Objectives

This study aims to shine a light on inconsistencies and the agreed-on points of Scrumban methodology found in the literature and compare these to the practitioners' views. Practitioners' views can be achieved by means of in-depth interviews, after that, the data from both Scrumban

literature and interviews will be used to spot the weaknesses and then to develop a framework to solve those issues and to provide guidance and a tool to help Scrumban practitioners welling to adopt Scrumban choose the right practices based on their specific needs. More precisely, the objectives of this research are:

- 1. Explore and investigate available theory sources and literature about Scrumban.
- 2. Understand how Scrumban practitioners are using Scrumban and the perceived benefits of it from their perspective.
- 3. Understand how practitioner users address commonly found issues.
- 4. Compare the practical application of Scrumban to those found in the literature.
- 5. Contribute to a better application and understanding of Scrumban and for the benefit of the Agile community.
- 6. Propose a practical method for choosing the suitable Scrum/Kanban practices and offering some adjustment, constraints and control against the broad flexibility available in Scrumban theory sources.

1.6 Research relevance

State Of Agile Reports, which are yearly reports of agile situation in the markets, show Scrum as the most commonly used agile approach with 66%. However, according to Scrum co-founder Ken Schwaber, "75% of the organizations using Scrum will not succeed in gaining the benefits they hope from it" [15].

As a result of that, Scrum "variants" - which refers to methodologies that are not necessarily Scrum but share lots of practices with it, have gained wide acceptance in the agile world. Scrumban is labelled as such a variant, and those arising approaches provide promise for overcoming or mitigating the risk of losing Scrum benefits. Till this point, Scrumban is not well understood in agile communities because of the poor but also conflicting literature about it. Thus. confusion and a set of inconsistent representations are the main features around Scrumban [6].

The current situation of Scrumban in theory and practically, has emphasized and inspired this research to make direct debate and meetings, and then making in-depth interviews with the experts in Lean and Agile circles, to assess and address in reality, how is Scrumban understood and what the core tasks and activities are applied. Then based on this and the literature study, propose a framework that may contribute to a better understanding of the methodology and as a result raise the adoption level, debate and disseminate findings to leverage awareness and support the literature about Scrumban.

Chapter 2 Research Methodology and Design

This chapter will describe what research methodology will be used and what research approach will be taken. This chapter explains why the chosen method suits the research question.



Figure 3: research methodology stages

2. The Research design and methodology

This research is exploratory in its nature. According to Saunders [16], exploratory research tends to disclose the current situation of the topic in question by asking "what is happening" on the way to uncover new insights, and that coincides with the nature of the research question: "How is Scrumban being applied from a practitioners' point of view?". This will be led by a grounded theory approach where the data collection and the analysis will determine the research findings [16]. Exploratory research's most known principles are searching the literature and interviewing experts in the area in question. Consequently, this research has followed the following stages:

2.2. Literature research study

In this stage, by examining the literature, the research question was defined and more information about the topic itself was learnt. This phase normally results in having a clear and focused research statement. In this case, this stage resulted in defining the basics and the key point the research will be about. During the literature study, more knowledge about the research topic is gained. The foundation of the literature study serves as input to set up the interview questions, define the scope of the research, identify the literature gap, and lastly outline the research question. In Addition, during this step, the key problematic areas in the literature related to the Scrumban methodology are identified. Also, an understanding of the current issues surrounding the Scrumban application in the literature review was formed, like setting the research objectives and identifying the gap that the research aims to fill. Tools, like Leiden University Catalogue, and Google Scholar, were used in this step. This paper has also looked at publications related to the topic in question for detailed review and analysis. The following keywords are employed for the review:

- Kanban.

- Scrum.

- Scrumban.
- Scrum and Kanban
- Scrum with Kanban.
- Agile.
- Lean.
- Software Development.

2.3 The primary research

With a robust understanding during the previous step. The next logical step will be the design of the research method. One of the main objectives of this research is to gain a deeper understanding of how Scrumban is applied in reality by reflecting on the practical experience of practitioners. To this end, a primary qualitative data collection method will be utilized using interviews. Especially, the research question is aiming to understand "how" Scrumban is being applied in practice, and for such questions, qualitative studies are more suited [17].

2.4 Interviews design

Subsequently, the list of interview questions is used to explore and investigate the practices of the Scrumban methodology based on practitioners' viewpoints. In this stage, the interview questions were created and divided into several dimensions. Those are:

- Opening question
- Scrumban Teams
- Scrumban Delivery Mechanism
- Product Backlog, Planning, and Estimation
- Meetings and Ceremonies
- Working Mechanism
- Metrics

In this research, In-Depth Interviews are adopted. "In-depth interviewing" is an open-ended, discovery-oriented method to obtain detailed information about topics that are sensitive in nature. It is an effective qualitative method to dive deep into participants' perceptions about a certain topic [18]. Interviews can be conducted anywhere from five to fifty in-depth interviews, depending on the research goals, the more interviews are conducted, the more complex and time-consuming. In another word, "in-depth interviews" may focus on quality rather than quantity, so a balanced approach should be achieved there taking into account conducting nothing less than five interviews. In-depth interviewing can provide several advantages such as uncovering valuable insights and providing high-quality data. In addition, it can help interviewes to be most likely to

open up to disclose the required data considering that the interview is conducted on a one-on-one basis [18]. The most apparent disadvantages are Analysis can be challenging, it is time-consuming, and lastly, interviewing requires a high level of training and skills.

The term "theoretical saturation" was first invented by Glaser and Strauss in the book "The Discovery of Grounded Theory" [19] to emphasize the idea of conducting a number of interviews just as needed, instead of conducting a high number of them. Moreover, theoretical saturation is presented as the point where any new data will add no to little value to the already existing information [20] [21].

Semi-structured interview style is utilized in this study. It is in the between the open and the structured interviews. This ensures that all fundamental question is adequately answered, by allowing the interviewee some freedom in detailing their answer or deviate a bit from the question. Which fits perfectly the intent of this study. As a result, and although the key matter itself might be directly asked, the answer could be inferred from the context of the discussion.

2.5 Sampling and Expert Selection

Sampling can be divided into different ways. The random sampling of a representative population and non-random sampling. In this approach, the subjects are chosen intentionally. For example, because of the certain knowledge they have, or because due to easy access [22]. Some of its drawbacks lie in the fact that errors and biases cannot be calculated as the probability is not part of the sampling strategy.

In this research non-random sampling is the best choice because answering the interview questions requires highly skilled and qualified practitioners in Scrumban method who have experienced Scrumban first-hand.

In this research, seven highly skilled and qualified agile experts were enough to reach the saturation point. Also, this choice may help to avoid complexity in data analysis and provide an opportunity to focus on the quality of the interviews. During the selection, the following criteria were considered a must regarding the participant's knowledge, background, and experience:

- The interviewee must have extensive experts in agile software methods.
- The interviewee must be a professional expert in the field of applying Scrum and Kanban.
- The interviewee must have recently applied Scrumban as a member of the team.
- The interviewee must have completed training or certifications regarding, Scrum. Kanban, or both.

Table II: participants of the interviews:

Participants	Role of the	Length of the	Relevant participant experience
number	interviewee	interview	
P1	Scrum master	About one hour	Scrum Master agile coach for 10 years,
			Kanban Master for two years
P2	Scrum master	About one hour	An agile practitioner for five years and two
			years as a Scrum Master
Р3	Scrum master	About one hour	Worked as a Scrum master for five years.
		-	
P4	Scrum master	About one hour	Scrum master for about nine years.
P5	Scrum master	About one hour	Scrum master for almost three years.
P6	Scrum master	About one hour	Scrum master for eight years
P7	Scrum master	About one hour	Scrum master for about seven years.

2.6. The Analysis:

Data analysis is the main stage in qualitative research. In this stage the data transformed from raw linguistic, and recordings forms to statements that can serve as results for the research by interpreting and classifying the data. Mostly, qualitative data analysis includes different techniques from a rough examination of the material (overviews, condensation, summaries) to approaches from a deep analysis (category elaboration, hermeneutic interpretations, or structure identification) [23].

In this stage, the inductive style was employed. This approach is mostly used in interpreting qualitative data. Such an approach aims to create a sense of the data collected to find patterns and associations in order to develop a theory. It provides a straightforward method for acquiring findings in the context of focused questions [24].

In this research, the "Data Display and Analysis" inductive approach will be utilized. Michael Huberman and Matthew B. Miles are considered the pioneers in this approach. They suggest that the "Data Display and Analysis" approach consists of three processes [25]:

- Data reduction
- Data display
- Drawing and verifying conclusions.

Data Reduction

During the first process, the data are summarized to focus on the important parts to condense them [16]. In this research, the data was organized into tables where all participants' answers to each question were summarized to acquire a focused response with brief reasoning behind the answer. Tables were chosen as they allow comparisons between all answers. the resulting data of this process will be prepared to offer a good insight into the participants' responses.

Data Display

Next, in the data display process, the focused data is organized into a visual representation structure, using matrixes or networks [16]. To conduct this step a framework called "Practice theory framework, Praxis, Practices and Practitioners" will be applied [26]. This framework is used in the strategy as practice (SAP) research approach in the field of strategic management. According to Whittington, who is a pioneer in SAP, not all elements of this framework must be



Figure 4: practice theory framework, praxis, practices, and practitioners. Adapted from [26]

used [26], and because of this flexibility, this framework was used in many other fields. The main three elements of this framework are naturally:

Practice, Praxis, and practitioners. Simply put and according to Whittington [26], Practices are regular behaviour including the traditions and the procedures for using things. Likewise, Praxis is refereeing to the actual activity done by practically by practitioners. Lastly, practitioners are those people who conduct the activities (Figure 4).

Making use of this flexibility of the framework, this research will be using the Practice (theory sources) and Praxis (practitioners) matrix to compare the two aspects. The proposed matrix framework will provide an excellent choice considering that it can display the key differences and similarities data for the theoretical references and the practical data extracted data from the expert interviews (Figure 5).



Figure 5: The matrix used Praxis, Practices matrix.

Drawing and verifying conclusions

Based on the displayed data, the results will be dependent on the compression and the key similarities and differences. It is believed that creating a visual displaying structure like matrixes will improve analytical thinking and the linkage between the components will evolve [16] [25]. After establishing the results, the next step is to discuss them and propose solutions.

Chapter 3. Theoretical Background and literature review

This chapter aims to examine how Scrumban framework is introduced and explained in the theoretical references, to provide the background theory and scientific basis for this research. But before going into Scrumban framework theory, it is necessary to provide an overview of the Software development process in general, with Scrum, and Kanban as the basic components of Scrumban framework.

3.1 Software development process evolution:

Several software models have been developed by software engineers, and everyone tries to prove that his model is the best [27]. At first, the waterfall approach is a famous traditional model that was discussed by Winston W. Royce in 1970 [28]. It was widely embraced by many industries like construction, IT, and software development because of its logical linear sequential steps. The waterfall model supposes that each step should be finished completely and thoroughly before stepping into the next step. The steps in this model are requirement, analysis, design, coding, testing, and operation, as illustrated in the following Figure (6) [29].



Figure 6: The waterfall model. Adapted from [29].

The attitude in the traditional models is that spending time in the initial phases during the software production can lessen expenses at late phases. This model depends heavily on documentation so that it is independent of the developers in case of leaving the project. Although this approach is suitable in cases where the requirements are known in advance, the possibility of change is low, and when there is enough time to do all steps sequentially [27], the waterfall model lacks the ability to deal with the ever-changing market needs, the unclear requirement, and sudden changes in customer's requirements [30]. This logically resulted in the appearance of agile. Agile software development is probably distinguished by emphasizing an iterative product development process, 20

which can be continuously assessed on its functionality, quality, and customer satisfaction. During these frequent iterations, the acquired information can help to determine the progress and enable the developers to react rapidly to customers' feedback, and consequently, the risk is minimized. Thanks to the hard rule of delivering a working functionality of the product, Agile can effectively help mitigate the schedule and budget risks [31].

Figure (7) is compression between the traditional waterfall method and Agile methodologies in terms of requirements gathering, product design, the development of the product, and testing. In contradiction to Waterfall, Agile frameworks can help create the most possible value for the customer by using frequent reviews and feedback from the client [31].



Figure 7: Comparison between Agile and Waterfall Approaches. Adapted from [3].

According to Rod Stephens, in software development we can recognize four types of approaches, assuming a software product with three features [27]:

1. Predictive (waterfall): Provides all three features at the same time with full fidelity.

2. **Iterative:** Initially provides all three features at a low (but usable) fidelity. Later iterations provide higher and higher fidelity until all the features are provided with full fidelity.

3. **Incremental:** Initially provides the fewest possible features for a usable application, but all the features presented are provided with full fidelity. Later versions add more features, always at full fidelity.

4. Adaptive (Agile): Initially provides the fewest possible features at low fidelity. Later versions improve the fidelity of existing features and add new features. Eventually, all the features are provided at full fidelity.

Agile is considered a strongly adaptive methodology [32]. By Agile it is meant all the different methods underneath it (e.g., Scrum, eXtreme Programming, ASD, etc...).

Agile methods are proven to be more successful and more efficient in contrast to structured methods. Based on [32] [33] [34], it can be stated that Agile is the most effective method for projects that start from zero and may be implemented at a continuous pace. Agile identifies four values in the Agile Manifesto [35] the values are:

- 1. Individuals and interactions over processes and tools.
- 2. Working software over comprehensive documentation.
- 3. Customer collaboration over contract negotiation.
- 4. Responding to change over following a plan.

There is no doubt that organizations in these times are struggling for more agility in their software development processes. Agile methodologies are proven to be effective in increasing productivity, especially since the agile success rate in most cases overcomes that of traditional project management [36]. According to a recent survey done by KPMG [37], the most common drivers for shifting towards agility are faster product delivery and a better response to changing customer needs, whereas traditional models showed less flexibility and a lot of complexity in dealing with.

On the other hand, agile methods may have major potential risks in the application. For example, most of the Agile practices assume small teams, with much larger teams, the policies of using white boards with sticky notes and face-to-face meetings start to collapse [38]. Agile methods are easy to misunderstand. Besides, it is well-known that there is no one-size-fits-all method for software and information systems development [39]. Therefore, companies, for example, that use waterfall as the main development strategy are reluctant to adopt Agile since such adoption on the organizational level might lead to a conflict with the conventional Waterfall method. As a result, a "Hybrid Agile" approach has emerged to help organizations transition to agile efficiently [40].

Combining two or more agile methods is not new in software development. Research studies on that are many and have gone in different directions. Thus, it is possible to add a fifth type to Rod Stephens's software development approaches, which is the "Mixed approach". By "Mixed" it is meant combining two established software development methodologies. One famous use of this approach is by organizations that are willing to implement agile methodologies after applying a specific methodology for a long time. In this case, they choose a hybrid approach by which they choose practices from various methodologies instead of rigorously applying a specific one [41].

How to combine these different methodologies has been always the question. In mixed software development environments, choosing practices and configurations of this environment is done on several bases to fit the specific situation at hand, in the following section a discussion of existing research on this topic is introduced. One approach emphasizes the discovery of the existing system and process and then uses the new framework to gradually introduce the new process as needed [6]. This kind of steady-paced introduction can be role-based and process-based. Karlsson [42] agrees with this statement by motivating the teams to adopt a specific methodology and incorporate each practice of it after assessing whether this particular practice fits with the goals or not, if it does not, then it can be replaced with another practice from another methodology. Another different approach taken by Boehm and Turner [43], is a risk-based approach for making methodologies in proportion to a project's needs based on five factors, namely: Size,

Criticality, Dynamism, Personnel, and Culture [43]. This means, "It is better to build your method up than to tailor it down" [43]. Similarly, another approach was taken by Austin, who developed a basic contingency framework to evaluate the practices based on cost/ benefit analysis and integrate them accordingly, with a special focus of this framework on when to use particular approaches and how they might be usefully combined. In practice, it is possible to differentiate two different ways of combining two software methodologies. The first one is by mixing two or more recognized agile methods. The second is the combination of one agile framework with a traditional method like a waterfall. This research will focus on Scrumban, methodology, thus, combining agile and traditional methodologies is out of the scope of this study.

Before discussing Scrumban, Scrum and Kanban methodologies will be examined as the main parts of Scrumban. Scrum and Kanban have extensive theory sources, so we will discuss the most relevant subjects of these methodologies.

3.2 Scrum methodology

The word "Scrum" originated from the rugby game. This terminology was first mentioned by Takeuchi and Nonaka [44]. According to the Scrum Guide, "Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems" [45]. Although Scrum is focused on software development projects, it has been used in many other fields like hardware, autonomous vehicles, schools, and almost everything we work on as individuals and as societies [45]. That might be mainly because Scrum tends to give more control over the total flow in software development and close the loopholes where other developments fail in doing so [1]. It is an iterative methodology that depends on delivering a working and potentially releasable product (Increment) at the end of each iteration (sprint). Scrum is founded on "Empiricism". Scrum is grounded on the premise that complex activities, such as software development, are impossible to accurately predict, and consequently, change is inevitable and must be accounted for in advance [45]. Scrum recognize three pins that sustain all experimental practices: visibility, inspection, and adaptation [45]. These three together constitute Scrum's pillars.

Breaking down units into smaller ones is core concept in Scrum [46]. Thus, the product should be divided into smaller prices during the development process. The organization is split into smaller cross-functional self-organizing teams, and lastly, the time is split into fixed-length iterations ranging from 1- to 4 weeks. Scrum adopts the idea that small teams that work independently are more effective [47]. Scrum aims to increase the speed of delivery of potentially shippable product functionality every period by facilitating regular stakeholders' feedback and thus a higher ability to respond to changing requirements and market conditions.

Scrum is also referred to as 3-5-3 method, which abbreviates the core component of Scrum:

- 1- Three artifacts:
 - a. Product backlog.
 - b. Sprint backlog.
 - c. Increment.

2- Five events:

- a. Sprint.
- b. Sprint planning
- c. Daily Scrum.
- d. Sprint review.
- e. Retrospective

3- Three roles:

- a. Product owner
- b. Scrum master
- c. Development Team/ Developers.

An iteration (sprint) of development activities, is the heart of Scrum [48] [45], it takes place one after another [49], each is time-boxed of 1- 4 weeks, and the key idea behind each sprint is to deliver valuable functionality at the end of it [50]. Before the start of each sprint, a planning session should take place in order to answer two main questions [45]: First, the features of the resulting increment of the next sprint, and secondly, the way the work needed to deliver this increment will be carried out. This meeting is time-boxed. The recommendation is to keep it to maximum of 8 hours when the team is utilizing a one-month Sprint. At the end of this meeting, the backlog is

updated, and team members sign up for their share of tasks [50]. During the sprint, the team has recurrent (usually daily) Scrum meetings, in which team members meet to inspect each other's activities [49], this is done by answering three questions mainly [45]:

- 1- What did I do yesterday?
- 2- What will I do today?
- 3- Do I see any impediment to my taken tasks?

It is recommended that this meeting is held within a time box of 15 minutes and at the same time and location every day. When the sprint ends, a Sprint review session takes place where the team is supposed to demonstrate the result of the sprint to the product owner and the stakeholders [45]. The output of this session plays a valid input for the sprint planning where the work for the coming sprint is decided by the whole scrum team and held for a maximum of 8 hours [45].

Between these two meetings, the sprint review and the sprint planning, a retrospective meeting is held by the Scrum Master. Here the team should go over the last sprint and try to improve the current process [45].

A "Product Backlog" is where the team tracks tasks in the shape of a list. The first starting point for the team is to order this list according to the importance of each item, and this grouping and prioritization is frequently changing during the project -as should be expected, to reflect the change in the business necessities and the speed in which the team is altering the product backlog into functionality [48]. In Scrum, the sprint backlog is a place to group the items that are selected to be worked on next sprint plus the plan on how they will be delivered during the planning meeting. This backlog comprises the outcome of the prediction done by the development team about the features for the next increment [45]. Each product increment builds up on previous ones [50].

Scrum recognizes three roles to achieve its process. First, the Scrum master leads the different Scrum meetings, set the initial backlog, and measures experimentally the development in the direction of supplying the increment [50]. Also, this role is responsible for adapting the Scrum process to fit with the organization's culture, as well as for educating everyone involved in the project about the Scrum process and practices [48].

The second role is the product owner, who is seen as the person responsible for the product backlog, as well as maximizing the value of the resulting product [45]. Also, this role has the duty of creating the project's primary specifications, return on investment (ROI) goals, and releases, as it represents the interests of all stakeholders involved [48]. Finally, the Development Team has three main features, which are self-managing, self-organizing, and cross-functional. The team is the one responsible for deciding on how to convert the product backlog into a functional product [48] [45].

3.3 Kanban methodology

Kanban means a signboard in Japanese [11], defined as "It is a flow control mechanism for pulldriven Just-In-Time production in which the upstream processing activities are triggered by the downstream process demand signals" [51]. The fundamental concept behind Kanban is to implement Lean thinking in real-life scenarios [52]. So, Kanban is a lean approach that is seen as one of the most used frameworks in software development [52]. The fundamental quality of lean is to avoid all sorts of waste in the development procedure.

It is a relatively new concept in the field of software engineering that was originally applied in Lean manufacturing [2] [52]. It is mainly a framework for encouraging transformation and allowing continuous advancement [6]. In recent decades, a solid movement advocating using Kanban in software engineering has begun [52]. David Anderson saw that the main reason for the rapid adoption of numerous organizations was because they were struggling with Scrum [53]. The Kanban method in software development started in 2004, while David J. Anderson was helping an IT team at Microsoft that was not operating as it is supposed to and introduced Kanban there to enable visualization and by limiting the work in progress (WIP) [54], and lead to astonishing results. In contrast to Scrum, it is less prescriptive, in the sense that it does not recognize iterations such as sprints, does not outline roles, does not delineate meetings like stand-up, demos, or retrospectives, and it does not contain process artifacts, such as a backlog [55].

The Kanban Method is deceptively simple, and Scrumban incorporates its four principles and six practices, which according to David Anderson, a leader, promoter, and coach of the Kanban method [56] [57] [58] are:

- 1- Four Principles are:
 - Start with what you do now.
 - Respect the current process, roles, responsibilities, and titles.
 - Agree to pursue incremental, evolutionary change as opportunities are discovered.
 - Encourage acts of leadership at all levels of the organization.
- 2- Six practices are:
 - Visualizing
 - Limiting WIP.
 - Managing flow.
 - Making policies explicit.
 - Developing feedback mechanisms at the workflow, inter-workflow, and organizational levels.
 - Improve collaboratively using model-driven experiments.

Where the principles are self-evident, it is important to shed some light on Kanban practices. Firstly: To **visualize** the work Kanban uses the "Kanban board". Its design differs widely among the practitioners, related to the way it is used since Kanban itself does not constrain how to design them. The importance of the board is that it offers more visibility to the development procedure, by showing the appointed work of each team member, addressing priorities plainly, and underlining bottlenecks [52]. Those are cards that represent the flow-control tickets between the different workstations and processes. In this stage, Kanban encourages measuring the cycle time (i.e., the average time to complete one task) [51]. The main incentive for the visualization is to proactively detect the limitations and barriers of the process as well as promote focusing on a single item at a time [52]. Another important thing to visualize is the policies, for example, explicitly putting a description on the lines between the columns of what should be done when an item moves from one column to another.

Secondly, **limiting the work in progress** (WIP) which implies the work items the team has started working on it but has not yet finished [59]. In traditional software development methods, the general mechanism is that the work is pushed to the next stage in the process. Instead in Kanban, WIP replaces a "push" system with a "pull" one, in which new items are not started until the items in the previous stage are completely done [57]. In addition, it imposes developing only those items which are requested. This allows the team members to concentrate only on those items at a time [52]. As a result, this produces a continuous flow of issued items to the end user. As a result, optimizing and the right limiting the working amount will lead to shorter lead time and higher delivery rate, and better quality [57].

Flow management aims to lessen the lead time and make the flow as predictable as possible with special importance on managing dependencies and bottlenecks. Improve the flow by making it as small as possible in order to avoid waste in the entire process [60]. It also includes managing the relationship with the service consumer [57]. The quality of the workflow is measured with four metrics: queue size, throughput rate, cycle time, and lead time [61]. In general, flow management is needed to make sure that the items are pulled into the workflow at the same rate they leave it, as well as responding quickly to the blocked or queued work items [59]. Knowing that incorporating such a methodology will not be challenge-free, Kanban Method's principles and practices emphasize pragmatism [58], thus experiment is the main part of this method.

Make policies explicit. Sometimes it is referred to as 'entry' and 'exit', as it signifies the criteria for pulling the work item pulled from one stage to another [61]. It also indicates establishing a common understanding with the stakeholder over the process, problems, and recommended enhancements [60]. Other examples are including defining when the work officially can be marked as done, and "Replenishment" to determine when a new work should be picked up when capacity allows [57]. The benefits of Making the policies explicit is that it enables the organizations to examine the "cause and effect" after adjustments to the process take place and to measure the throughput [61].

Developing feedback mechanisms, because creates a shared understanding of the state, coordinate the work, and introduce enhancements. In that sense, Anderson suggests seven specific feedback opportunities, or cadences [57], which are according to him:

- 1. Strategy alignment: This is for the selection of the services to be provided and sensing how the external environment is changing in order to provide direction
- 2. Operational coordination: To understand the balance between and across services.
- 3. Risk management: To understand and respond to the risks for effective delivery of services.
- 4. Service improvement: examine and improve the effectiveness of a service.
- 5. Replenishment: for moving items over the commitment point (and into the system).
- 6. The Kanban Meeting: This is the daily coordination, self-organization, and planning review for those collaborating to deliver the service. It often uses a "stand-up" format.
- 7. Delivery Planning Meeting: to monitor and plan deliveries to customers.

Although there is a consensus about the Replenishment and Kanban Meetings in almost all Kanban applications, it does not suggest that all the seven cadences have to be added as extra meetings to the organization's agenda.

Improve collaboratively using model-driven experiments, endlessly, and incrementally all parties involved should work on enhancing the process by having a shared understanding of the state [60]. Anderson sees that since perfection in an ever-changing environment is unattainable, improvement is a continuous process that has no endpoint. At times, using empirical methodologies is a suitable means to understand the ultimate appropriateness of objectives within an environment [57].

3.4 Scrumban

The term Scrumban was first invented by Ladas [14]. According to the latest agile state report, Scrumban is gaining more popularity than ever with more than 10% of teams indicating that they are hiring Scrumban as their main method for software development in 2014 [62]. In addition, it is not rare to find teams that use practices from another method without being aware of that, such as the case when a Kanban team agrees on meeting daily, which this a Scrum practice. Likewise, a Scrum team limits its work in progress per column and that is a practice of Kanban [46]. Since, and like any other tools, Scrum and Kanban are neither perfect nor complete [46]. Teams that apply Scrumban effectually enjoy the benefits of the product delivery of Scrum together with the process enhancement and workflow management of Kanban [63].

Scrum is recognized as a framework that has proven its effectiveness. However, in many cases, it lacks the guidance to get through specific events [63]. For example, in many cases, even highly skilled Scrum teams find themselves very behind in terms of delivering the wished functionalities 30

to stakeholders, although the team is using its full speed [63]. Leading to stakeholder frustration as the "pre-backlog" of ideas keeps growing [63]. Similarly, the foundation of Scrum is "commitment", this is the keyword that Scrum built upon to make the required changes, and this has led to executives having the mindset of treating the assigned product backlog items as nothing other than commitments [58]. That is again attributed to the fact that little guidance is offered by the framework to help the whole enterprise overcome pragmatic real-life situations like these, in this case, the executives' managerial level [58]. In this regard, Scrumban introduces a way of finding solutions for those problems that cause their teams to be overloaded [63].

Likewise, in many cases when Kanban is brought into the picture, mistakenly, it is treated as a system for managing the work, like a Scrum without sprints, leading to a method that does not have empirical process control and leading to worse situations than before [63]. In fact, Kanban is not a project management framework and that is the reason why it should always be layered with other software development frameworks [58].

3.4.1 Scrumban practices and applied principles in literature

1-Working mechanism:

The mechanism can be summarized in three Kanban principles:

1- Visualize. 2- Limit the WIP. 3- Pull.

One of the most important features that Scrumban takes from Kanban is visualization. Scrum board is not sufficient to clearly reveal the changes in Scrumban where sprints were replaced with Kanban features of a pull-driven coordination system and work-in-progress (WIP) limitations [4]. The aim of the visualization is that it should act like a visual control mechanism representing how the workflows are moving through the different stages of the development process, normally represented through sticky notes or e-cards on a wall system [64], as the project progresses those cards move through the different columns of the software development stages. As soon as a card is completed it moves to the next column allowing another from the previous one to replace it so that it keeps moving across the columns until it reaches the last column indicating that the 31

story/feature is done. Moreover, other visualizing features can be added to the board, like having the backlog before the first column to pull from including the planned stories [12]. Likewise, when visualizing each stage of the development process (like; coding, testing, etc...) each column can be divided into sub-columns like doing/ done indicating its current state, in order to give more transparency [65]. Another enhancement that differentiates Scrumban is that it can enable teams to separate the concept of the items assigning the work from prioritizing the work by adding a ready queue in between the backlog and work-in-progress column. This ready queue is where the items have just left the backlog, but then have more importance, and have not been assigned to any specific member. But then, the moment a member becomes available, an item from this queue is taken to work on instead of picking from the backlog immediately [14]. Similarly, visualizing the work helps to maximize the customer value, as well as waste, which will be reduced [11]. In addition, the board will help to measure the average time for completing one feature and hence, measure the whole project time [11], as well as allow transparency for every item and its current state [12].

Each column should have its own WIP limits written indicating the maximum number of cards that it can contain at the same time. These limits are based on the team's capacity. Limiting the WIP helps software teams improve software development productivity by simply not allowing team members to multi-task [4]. This is a fundamental difference from the Scrum board where there is no rule stopping the team from putting all items into a column at the same time [46]. In Scrum, the team works in its first few iterations to measure its velocity (the items done per one iteration) and build upon to limit the stories the team will take to work on in every next iteration. The velocity per sprint in Scrum is replaced with the WIP limit per workflow instead [46]. One benefit of limiting the WIP is that it considerably reduces lead time, which is a key measure of the team's throughput and productivity [64]. Once a certain stage reaches its limit, team members help each other to complete the tasks rather than starting a new one, which points to an important benefit of enhancing the collaboration between the team members and reducing the likelihood of bottleneck situations [2]. In [64]it is recommended that the WIP limit should be between 4-5 tasks in the development column at the same time. Here a challenge is presented which is to determine

the right values of limits to guarantee an optimal workflow [14] [4].

In this stage, after becoming familiar with how the process is visualized and how the limit on WIP is set, the team is already acquainted with the pull-based system. Here the team members pull tasks from one column to another and move them through the development stages. The general rule here is that as soon as there is capacity available a new work is "pulled" into the system to handle, rather than being "pushed" from the outside [64]. The harmony with limiting the WIP ensures that the pull system will not be overloaded, therefore maintaining a sustainable pace of development [64]. According to Reddy [58], this pull system is a way to make sure that a perfect match will be made eventually between the team's total capacity and a stable WIP level.

2- Scrum ceremonies in Scrumban (planning, Review, Retrospectives, and Daily Stand-ups)

Although it is not generally agreed on, Scrumban's ceremonies in the literature seem to be controversial. Some researchers indicate that only parts of Scrum's meetings are needed and/or the meetings, in general, take place only "as" and "when" needed [14] [66]. Whereas for others Scrumban treats those events as important ceremonies that are retained from Scrum [58] [12]. In most literature, those events are flow-based [59]. The review session is the occasion to have feedback from the stakeholders, as well as monitor the flow metrics which, as a result, will determine the progress toward the sprint/flow goal and the general productivity of the team [59]. Retrospective where the process is discussed and improvements are set. The daily stand-up is to highlight each member on who is working on which card and what are possible impediments. However, meetings in Scrumban are a bit different from those in Scrum. For example, the main goal of a sprint review in Scrum is to have the stakeholders' feedback whereas in Scrumban another important principle is integrated which is reviewing the overall service delivery borrowing that from Kanban [58]. Daily stand-ups might also be improved by taking more the shape of the focus group session by taking the approach of semi-structured interviews [4].

Planning, nonetheless, is conducted differently. Scrumban borrows JIT (just-in-time planning) from Kanban. In Scrumban the planning is done only for the short coming period of time [65] [60]

[12] [2], as having long planning meetings is not efficient in case the priorities are in continuous change, and especially the team pulls work into a ready queue before pulling it into work in progress which is a kind of planning. This also enforces limiting the WIP rule. Thereby as soon as a limit is reached team members should collaborate to finish the remaining tasks instead of starting a new one [2]. Similarly, at any time the amount of current tasks drops below an agreed-on level, planning should take place. According to C.Ladas [14], the main goal of this meeting should be filling the empty slots rather than determining the number of slots or filling the whole slots, which would consequently lead to less overhead and planning ceremonies.

3-Estimation and Metrics

In a traditional time-boxed Scrum process, velocity is what indicates the work handling speed of the team and is normally measured in story points and based on that the team members would be assigned the tasks in the backlog. However, Scrumban's backlog contains all similar-sized items and therefore the estimation does not have to be carried out before each iteration, gaining more flexibility to handle frequently changing requirements and programming defects [51]. Besides, and in compersion with Scrum where it is required to have this estimation session before every sprint, a fixed-size event-driven backlog would be more of an advantage in order to reduce the waste of creating and discussing too many user stories [4]. Thus, after having the stories divided into similar sizes, teams set a session to choose a fixed number of prioritized tasks from the backlog, this number is based on the team's completed items in previous iterations [14].

That is also a key motivation for why Scrumban uses cycle time and lead time over Scrum's velocity calculations [14]. The Lead time clock is the time from the moment of requesting a new feature till its delivery of it, whereas the Cycle time is when work starts on this feature up until when the item is ready for delivery [14]. Since estimations and used metrics are tightly related, Scrum's inaccurate estimates and time-boxes are reported to lead to longer lead times, leading to waste, and here Scrumban introduces another improvement over Scrum [67]. In fact, one of the main goals of using Kanban's JIT and limiting WIP technics is to reduce the lead time [12] [64]. The stated advantages of using these technics were that they helped to reduce the lead time to half,

and in some cases, even more, increase productivity, and improve quality [2] [46] [64]. Lead time is particularly important, measuring it allows for a comparison between the estimated against the actual lead times for the same delivery date [58]. Lead time can be calculated using Little's law:

Average lead time = average WIP / average throughput

Since this measurement is an indicator of business agility [58], Steve Tendon proposed a variety of important lead time calculations in the figure (8) [68].



Figure 8: Scrumban metrics. Adapted from [6].

Another metric outlined in the literature is the "**Throughput**". The amount of of work items completed per unit of time [58] [59]. The aim of taking this measurement is to constantly improve this rate until a rational level of optimization is reached [58]. Although they might seem related, throughput is different from Scrum's velocity. In the case of velocity, the time frame is limited (sprint), besides, it depends on team estimates of time and effort. Whereas throughput, measures completed delivered work, which is naturally more significant than story point (which is delivered in velocity measurement), another difference is that throughput is not bound to any specific system

and thus may be applied to multiple ones [58].

Some measurements can help in indicating the stability of the flow in advance, where lead time would be no help in such a case [58]. **The ageing of WIP** is the amount of time between the moment a work item has started and now, and is only true for items that are in progress [59]. This indicator is useful in the sense that it represents potential problems requiring attention.

3.4.2. Related Work

This part of the thesis will discuss the relevant previous research studies represented by published articles and some thesis in Scrumban domain, to assess and evaluate the available (literature/related works) that have been gathered and read surrounding Scrumban's previous research studies, and then discuss those.

Scrumban is fairly a newcomer to the field of software development and still needs time to fully mature. Scrumban is often misunderstood by practitioners in Lean and Agile circles [58]. Some treat Scrumban as if it is the usage of a Kanban system within Scrum, while another part sees it as a completely new method for software development that merges the best of Scrum and Kanban methods [58]. Scrumban is a combination of Scrum and Kanban that blends both software development frameworks and takes the agility from agile and the process management of Kanban.

Scrumban in the literature started in 2008 when Corey Ladas [14] invented the term "Scrumban" and introduced many essays in his book describing how best Scrum and Kanban can be merged and defended to loosen some practices as well as taking many useful others of both frameworks, the most characteristics identified his work were integrating a pull mechanism from Kanban including limiting the work in progress (WIP) with other Scrum-based practices. Therefore, he proved that flow-based development represents a great track to real Agility. This was followed by many pieces of research done to amend Corey's ideas in a way that fits the special settings at hand. In 2012, a case study was conducted in one software development company [55] where they described the transition process from Scrum to Scrumban and concluded that the two most 36
important elements in any transition are to create instruments for continuous process improvement and to have well-trained and dedicated personnel. In the same year [60] Lukasz D. Sienkiewicz claimed that Scrum is not enough for "Network Organization" so he took a Scrum-based model with some fitting practices from Kanban as well as removed/added new features to Scrumban as needed to sever the special settings of such organizations. In 2014 Viljan [64] researched the introduction of a more advanced Kanban board for teams willing to adapt Scrumban, supported by a real-life example, and for this purpose, he researched the most appropriate practices from both Scrum and Kanban and concluded that both frameworks can be merged in a way that enables a more productive software development. In 2016, a book by Ajay Reddy was published [58]. In his book, Reddy agreed with Corey on some Scrumban practices and disagreed with him on many more. Reddy took a more Scrum perspective for Scrumban, and it is noticed that he mostly took the approach of giving teams the freedom of deciding on their own what is the best practices out of many possibilities. In 2016, a case study was conducted by Yilmaz and Corner [4], where they carried out a wide survey to see the views of 30 practitioners about the adoption of the Scrumban in an SME enterprise, and then introduced a gamification approach and integrated it with this hybrid framework to increase the motivation of the software practitioners. The results were supporting the idea that using Scrumban solved some of the major issues during Scrum application, was also found that it enabled teams to solve problems related to estimations and prioritizations, and Scrumban-gamification integrated approach provides a systematic performance improvement. Next, in 2017, Banijamali et al [2] noticed that Scrumban impact is really poorly understood till now. In this study, six major software challenges were used to see how Scrumban can tackle them during a distributed project at two Software Factories in two universities in Finland and Italy. The study concluded that Scrumban has positively solved the majority of these challenges. In 2018, Patil and Neve [65] investigated the impact of Scrumban on productivity and found that it helped reduce 20-70% of defects, 30-70% faster time to market, 10-20 happier employees, and finally, 20 - 45% increase productivity. In the same year, Albarqi and Qureshi integrated Scrum, Kanban, and Lean creating a methodology called L-Scrumban in order to improve the effectiveness of the software development process. The resulting methodology was tested by a questionnaire against five goals. It was distributed among eight software development companies, and the results confirmed the efficiency of the proposed methodology in the software development field and its competence to overcome the weaknesses of the previous frameworks. Robinson and Beecham introduced their experience report [56] from Sony Interactive Entertainment (SIE) for a team involved in open-source software (OSS) project who ran into several difficulties when using Scrum as the main software development and those were tackled using Scrumban. Scrum.org, in 2019, issued its guide [59] for Scrum teams wishing to incorporate some practices of Kanban in their work process. Recently in 2020, Bhavsar et al raised the question of how Scrumban can solve issues related to the application of Scrum and pointed out some of the limitations of Scrumban in solving these specific challenges.

It can be said that there have been only a few pieces of research done in a purely industrial reallife setting [56]; Instances are seen either to be done in universities' coding labs, or on cases on how Scrumban, when merged with user's specific methods, can replace Scrum in order to decrease the workload [56]. In addition, since Scrumban was invented by Corey Ladas [14] till now there has not been any concrete definition as to what features of Scrum or Kanban should get into a team's implementation. That is also demonstrated by looking at the literature. Since it is a relative newcomer, pieces of research on this framework are limited, compared with other frameworks like Scrum, Kanban, XP, Waterfall, etc... Thus, the literature is not able to provide answers for teams that are willing to adopt the framework on how Scrumban is being applied in a general environment regardless of the special settings of the organizations.

Lastly, it is seen that none of the discussed studies tried to highlight the low adoption level of Scrumban and how it can provide an appropriate tool to help agile teams at all levels apply Scrumban successfully.

Table III: Related work overview:

Reference	Study title	Study object	Key findings	
Mashal	An Empirical Study	Aims to show how	- Different factors - such as roles	
Alqudah et al,	of Scrumban	Scrumban method is	adoption time, team size, etc., assist	
2018	Formation based on	formed based on a	Agile team members in the formation of	
	the Selection	combination of	Scrumban	
	of Scrum and	Kanban and Scrum	- Scrumban was found to be more	
	Kanban Practices	methods.	appropriate than Scrum or Kanban in	
			saving time, improving quality, and	
			minimizing waste	
Natalja	From Scrum to	Reports on	It describes the main adjustments made	
Nikitina	Scrumban: A Case	a transition from	to the development process besides the	
et al 2012	Study of a Process	Scrum to Scrumban is	transition and the improvements	
IEEE	Transition	a software	achieved and the lesson learned from it.	
		development		
		corporation.		
Lucija	Comparison of agile	Aims to find the	It provides a comparative study between	
<u>Brezočnik</u>	methods: Scrum,	answer to an	the three most prevalent - Scrum,	
et al 2016	Kanban, and	important question:	Kanban, and Scrumban to support the	
	Scrumban "Which agile met		selection between the three methods	
		is optimal for our	- They proposed to choose the one agile	
		company?".	method that best meets the requirements	
			and wishes of the company.	

	with Waterfall in	with Waterfall	of software development and	
	Software Engineering	'Scrumbanfall'	management practices like direct	
		positioning Kanban in	involvement of external stakeholders in	
		the <u>centre</u> of	project requirements and analysis	
		integration within the	documentation and tracker. and an	
		Scrum guidelines.	accurate estimation of the complete	
			project;	
			- highlights limitations of Scrumbanfall	
			like team members and their roles and	
			capabilities for the large-scaled project	
			having team members in a distributed	
			environment,	
Krunal	Scrumban: An Agile	This study used	Concludes that Scrumban is an Agile	
Bhavsar et al	Integration of Scrum	literature reviews, a	Business Process Reengineering (ABPR)	
2020	and Kanban in	case study, and	approach for the Scrum and Kanban	
	Software Engineering	surveys to investigate	practitioners as it can resolve some of the	
		how Scrumban can be	challenges of Scrum like work item flow	
		combined as a hybrid	management and visualization of its	
		framework with a	progress, reduction of lead time with the	
		focus on Scrum	help of JIT concept that helps in CICD	
		drawbacks.	(Continuous Integration and Continuous	
			Delivery)	
Zahoor	Scrumban - Adaptive	Improving the used	A new process model was suggested and	
Ahmad Khan	Agile Development	agile development	tested. This resulted in providing the	
2014	Process	method in the case	flexibility necessary to the team, and the	
	Using Scrumban to	company to overcome	workflow, and enhanced team	
	improve the software	the various problems	collaboration.	
	development process	faced during the		
		product development		
		based on Scrum.		

Fernando	Modelling and	Use modelling and The results showed that both factor		
Sambinelli	Performance	discrete-event	total project duration, and product	
2018 IEEE	Analysis	simulation to examine complexity, affect the productivit		
	of Scrumban with	the impact of Test-	development team that adopts TDD	
	Test-Driven	Driven Development	practice, and the most advantageous	
	Development	(TDD) on team	scenarios were identified. Based on these	
	using Discrete Event	productivity. 18	two variables a Fuzzy Logical System	
	and Fuzzy Logic	scenarios were	was also implemented that recommends	
		simulated for teams	TDD based on these two factors.	
		that are using		
		Scrumban.		
Łukasz D.	Scrumban – the	Extended the Scrum-	Propose a Scrumban software	
Sienkiewicz	Kanban as an	based model and the	development method that combines the	
2012	addition to Scrum	Kanban approach to	Scrum-based model and Kanban	
	software	make the most of both	principles The proposed approach is	
	development method	and find a more	dedicated to an Agile environment with	
	in a network	sufficient approach	Third Party Services introduced in a	
	organization	for managing	Network Organization.	
		software development	It highlighted the common parts of	
		in a strongly	Kanban and Scrum-based approaches	
		distributed Agile		
		environment (i.e.,		
		Network		
		Organization).		
Marian	Analyzing Agile	Aim to identify the agile methodologies tend to occup		
STOICA et al	Development – from	elements that impact	largest part of the software development	
2016	Waterfall Style to	agility in developing market. Also, it is translated		
	Scrumban	software products, in	more to other sectors.	
		a gradual approach,	-there is no "absolute best" agile	
		from the waterfall	development methodology	
		model towards		

		approaches like - Scrumban hybrid, maybe a		
		Scrumban.	"compromise" solution for the	
		Additionally, will	supporters and adversaries of the parent	
		understand the social	methodologies.	
		dimension of using		
		agile methodologies		
		in software		
		development projects		
		and the main barriers		
		to adopting such		
		methodologies.		
Ahmad	An Empirical Study	Describes an	Shows that aspects of Scrumban, such as	
<mark>Banijamali</mark> et	on the Impact of	An empirical study	iterative development, enhanced	
al 2016	Scrumban on	was conducted within	visibility, and limited work-in-progress,	
	Geographically	two Software Factory	would alleviate the challenges of	
	Distributed Software	settings in Finland	leveraging resources, synchronization	
	Development	and Italy to	between distributed sites,	
		investigate how	communication, and culture. It also	
		Scrumban can impact	explains that technical and security	
		coordination in	issues in the coordination of	
		geographically	distributed projects may demand	
		distributed software	solutions other than Scrumban.	
		development.		
		- provides the results		
		from a project case		
		that applied Scrumban		
		to develop a web-		
		based application for		
		Time-banking		

Sessione di	Using Scrumban	The work aims to	The applied Case Study, by showing	
I auroa dal at	Method	promote Scrumban how the procedure has been applied.		
Laurea der et	for Dradaetian	interdention in the	now the procedure has been applied,	
al /2015	for Production	introduction in the	shows a "Model of Application"	
	Planning and Control	manufacturing	demonstrating the compatibility of this	
	in Manufacturing	industry, identifying	instrument in the manufacturing	
	Companies	points of contact and	industry.	
		differences with the		
		software field and		
		develop a standard		
		Procedure that could		
		guide its		
		implementation in		
		different companies in		
		the manufacturing		
		industry.		
	Agile scrumban	Explore how IT	The proposed Model shows that Agile	
ALEX	project management	Audits can be made	Scrumban significantly improves the	
SUDEORA	in its audit	more efficient. and	Key Performance Indicators (KPIs) of	
2017		assess the	Price, Time, and Quality in a Service	
		improvement of IT	Organization Control (SOC) Audit,	
		Audits completed	compared to the Waterfall project	
		with an Agile	management methodology. Agile	
		Scrumban project	Scrumban can significantly reduce	
		management	variance from a SOC Audit's budget,	
		methodology versus	decrease the amount of time a project	
		the currently used	takes, and reduce the number of mistakes	
		Waterfall	made during an audit engagement.	
		methodology.		
	А	Presents an empirical	Scrumban was found more compatible	
Murat Yilmaz	Scrumbanintegrated	case study for with the workflow of the software		
et al /2016	gamification	adopting a hybrid	development organization. task	

approach to guide	Scrumban	prioritization and proper estimation
software process	methodology with an	problems became acute and visible. they
improvement: a	integrated	suggest that such issues may emerge
Turkish case study	gamification	because of problematic user stories or
	approach, which was	due to software programming defects.
	conducted in the	This study shows its actual
	context of a small-	implementation with initial results from
	medium enterprise	the field.
	(SME).	they found evidence that using
		Scrumban, and gamification provides a
		systematic performance improvement,
		and suggest that a Scrumban integrated
		gamification framework improves the
		software development process
		and supports the conceptual premise that
		software development organizations
		benefit from a Scrumban integrated
		gamification approach.

Chapter 4 The Results

In this chapter the three research sub-questions will be answered Based on the data acquired form the interviews with Seven agile experts.

4.1- Scrumban Application

In this section the aim is to answer the sub question "*How is Scrumban being applied in practice by practitioners*?". To do that, the results will be categorized and presented based on the seven dimensions. The dimension "estimation, planning, and changes to WIP/ Sprint Backlog" is discussed and merged to delivery mechanism. Right after, the data will be displayed using "*praxis, practices and practitioners*" matrix in order to compare it with the data from the literature (the rest of the matrixes will be found in Appendix B). Then at the end of this section a conclusion is drawn and discussed further in the next chapter "Discussion".

4.1.1. Teams' compositions

1-1. Roles

Although Kanban does not require any roles, the participants' responses show that roles are being used in practice. Those are either inherited as-is from Scrum (the three main roles: master, product owner, and development team), or it takes other names like Flow master instead of Scrum master, Kanban master. Those different names than those used in Scrum do not reflect major changes in the responsibilities of the role itself. Also, in practice, if the team does not use the roles already (for example if they were using Kanban before converting to Scrumban) practitioners will establish them.

"We still have a product owner. My teams are four to five people, including me and the product owner. Yeah. And I think I changed my name a little bit from Scrum master to Flow master" P1.

According to Scrumban literature resources, in most cases, Scrumban doesn't require any specific roles, one of the considered references [6], indicates that Scrumban respects existing roles and responsibilities. While Praxis data (practitioners) represented by real data extracted from the interviewees show a necessity for at least Scrum roles.

1-2. Cross functionality vs specialized teams

Cross-functionality is the most preferred way of composing the team. The goal of this is to make the team capable of solving all kinds of problems they may face. Though, in fewer cases, practitioners showed flexibility in this regard, so they may go for specialized teams in some cases. The type of the team is dependent on many other factors like the product type, whether it is support-45 based service or product based. This will be described in more detail in the discussion chapter.

"I try to come to a point that we're actually going to be cross-functional for everyone. And that's also why I use Kanban. Because, by introducing WIP limits, at some point, they're only working on two or three stories and those stories are not finished, but the development work has already been done, then because of that WIP limit, basically, the rest of the team is forced to help out the other people." P4.

"Yes. So actually, the three teams I worked with we use Scrumban....For one team the data scientist team, you need the specialities, you can't really say they all need to know data engineering and all need to know machine learning. The other two teams actually they were cross-functional... If it is up to me, I will always try to focus on the cross-functionality" P2.

1-3. Team size

In practice, the ideal team size is 7 members with a maximum of 9 people. The main concern in deciding on that is the communication complexity and the project size.

"In my opinion, the Scrum guidance is good to follow in this case. So, nine plus two minus two, even limit minus three, I would say. It's good, because it's, again about communication, and collaboration. You will have stronger impediment removal and faster reactions." P5.

In the literature, there is no specific number that indicates the size of the team. In most cases, it ranges from 6 up to 12 members. However, deciding on this aspect is depended on the situation at hand.

Team					
		Roles	Cross functionality	Team size	
Practices based three references	1 (Ajay Reddy, 2016)	It doesn't prescribe specific roles evolving roles represent environmental risk. Scrumban respect existing roles and responsibilities	Limited cross-functionality (complex technology systems required pockets of specialization)	12 or fewer people, its members are more likely to care about one another	
	2 (Zahoor Ahmad Khan, 2014)	Team plus as needed roles	Can be either specialized or cross- functional	The used team size is six members.	
	3 (Evaldas Bieliūnas, 2014)	Scrumban does not have clearly defined roles for team members so the roles may wary and it is up to the team whether any roles should be used at all.	Scrum-ban is a methodology where the team can be either specialized or cross- functional. Team member's cross- functionality in Scrum and their specialization in Kanban	No specific limitation of the number of team members	
Praxis Based on <u>Practitioners</u> interviewees	P1	We have a product owner and Scrum master	It's cross functional, but also specialists.	I support seven minus plus two (as in Scrum guidelines)	
	P2	The three scrum roles are still needed	We use our cross functional teams. But not all them. And we use specialized team as needed.	It's maximum nine people. For me, from practice, I will prefer the seven.	
	Р3	The roles really work with Scrumban as in Scrum	I would say it was cross functional. because with cross functionality comes flexibility.	I would say seven max. is, ideal team size	
	P4	Keep those roles as in Scrum	we have cross functionality in the people in the team. And certain people are still specialized in something	Team size of five to six.	
	Р5	In our cases, we always stick to the (Scrum) defined roles.	In my teams, right now we are not cross functional . we already started this discussion about cross functionality	Nine plus two minus two	
	P6	I would imagine that you always would come back to these three roles.	I go for the cross functional in my teams.	I would make it three to seven. I think that would be my ideal size.	
	P7	I actually use roles whether we are using Scrum or Kanban or scrumban	For me, it doesn't really matter if they are cross functional, or they are just specialized.	There were seven.	
Identify the key findings based on contrast Between Practices and Praxis		Scrumban theory almost doesn't require any roles, while Praxis data (reality application) show a necessity for at least scrum roles.	Scrumban theory indicates that teams can be either specialized or cross-functional. At the same time, Praxis data (reality application) focuses on the work objects and results and uses what the whole process demands and the product type, whether it is cross-functionality or specialization.	Scrumban theory doesn't require any specific size of team members. At the same time, Praxis data gives different team size that may reflect the type of the practitioner job, relates the team size with project size and complexity, and indicates that the smaller the team the better it is, mainly for ease of communication.	

Figure 9: practice, praxis teams' compositions matrix

4.1.2. Delivery mechanism

2-1. The optimal way of delivery in Scrumban

Practitioners try to use a combination of both sprints and workflow as delivery mechanisms. Some practitioners are combining the two strategies. In this case, the PBIs go handled with a flow-based process which lasts for the duration of a sprint instead of an open-ended workflow like in Kanban. In practice, the product type and the maturity of the team were also considered deciding factors in the delivery mechanism.

"We still had Sprint, but the flow-based working was really inside the sprint. So, making sure as I said, looking at the board every morning from left to right" P3.

2-2. Scrumban backlog

Kanban does not describe any type of product backlog, there is a consensus that the backlog is the same as in Scrum. Mostly it is visualized before on the board before the first column, resulting in a combination of the backlog screen from Scrum and the Kanban board into one agile board function like a Scrum board backlog.

"The product backlog for me, it doesn't really matter if it's Scrum or Kanban. I mean, both product backlogs have the same goal which is first to make sure that you know what is coming. And if it gets to the unmanageable size that means as a product owner, you're not doing a good job by saying no" P7.

2-3 Product Backlog content

In practice, the content of a Scrumban product backlog is the same as in Scrum. Therefore, the product backlog items can be features, and stories that are required for the product, prioritized by value and managed by the product owner. Moreover, the size of the PBI mostly does not have to be the same among all PBIs, as this is desired but hard to achieve, especially within teams that are dynamic and might not have the time to refine the size before starting with the PBI (Support teams as an example), but PBI should be as small as possible.

The product backlog items (PBIs) represent the input for the Scrumban board. It is seen that in case it gets to an unmanageable size it is the product owner's responsibility to solve it.

"And in my opinion, a feature team will be always difficult to have fixed-size items. That's really not possible, I didn't see it even happening, because it's not realistic" P2.

2-4 Planning

In terms of planning the practitioners' responses show a wide variety of options. For some, it was a regular planned session:

"In this case, it's (the planning) every sprint. So just every two weeks because we use the sprint still." P3.

"While for others, it was a triggered planning session, "In the Scrumban team, we do not do planning like in Scrum And if we look at the ready for Dev column, if it is drying up, we need to refine stories to add to it. That is basically by just looking at that column to see if it is filled enough or not. We hold the meeting to add stories." P6.

"For my teams, we do not have a planning session, because we treat our daily standards that we 48

use as also a potential planning session. Especially when it's needed, we do extend the time box of the session" P5.

It is worth noting that almost all Scrumban practitioners indicate having another sort of planning on a higher scale, like a "Roadmap planning meeting". In such a session, the long-term plan is communicated either to align within the team itself, with other teams, or on the organizational level.

"A bit of wider Look, the bigger image, it's not like limited to two weeks, three weeks or four weeks, but it's actually a quarter maybe. So just to have the alignment....Like the product owner talking: Hey, guys, girls, this is what can be expected from us. This is much what maybe we need to do in the longer run. And that will give the team some also ideas about what is expected from them." P2.

2-5 Estimation

The opinions about estimation were very divided between providing it or not. People who do not support it see Scrumban metrics as a sufficient replacement for this event.

"And I don't think I actually encourage the teams that are using workflow like Kanban or Scrumban as a workflow to use estimations. It does not add anything. Because estimation is replaced by metrics." P2.

On the other hand, those who see that it is crucial

"The main goal for me of estimation is to do it is to become confident as a developer, that it actually worked as you are able to commit within the sprint, and also, therefore, can commit that we're going to release that. And the second, more important for business, so my second objective is also to have some kind of estimation that you can use for product owners and stakeholders to show off

that you track record we're normally doing about this many points" P4.

"Totally agree that story point estimation can be wasteful. What I think the value of it (estimation) is making sure that you have the conversation of the developers acknowledging that there are

aligned about what the item is or not." P6.

2-6 Changes during the work in progress

According to the participants, for one part, the work backlog, or in Scrum terminology the sprint backlog, should be dynamic and changes to it should be allowed, providing that there is a good reason to justify it, and also on condition that it does not affect the short term plan/ goal.

"In Scrumban as a workflow or Scrumban as a sprint, you need to keep that possible, because it's all about Inspect and Adapt. And all these frameworks are talking about Inspect and Adapting. So while you are working, you could discover something you can be wrong about your forecasting, and you still need to be able to change something." P2.

For the other part, they are strict about that, so changes are not allowed to the worked-on items, as this may have undesired consequences.

"But changes to the sprint backlog or actually, if we translate it to the Kanban method is work which is already in the flow in a process, then I don't think that's a good idea to add new items to it or to remove it or whatever. Only if it becomes absolute. So you don't need it anymore. Then you can remove it. But don't change it." P7.



Figure 10: practice, praxis delivery mechanism (1) matrix

4.1.3. Meetings and ceremonies

3.1 Scrum ceremonies in Scrumban

Since Kanban does not describe specific meetings, all practitioners see that Scrum ceremonies are beneficial for the Scrumban teams, and they removed the need for other meetings and ceremonies since all of those are incorporated in Scrumban.

"I think it is needed. It still depends on the maturity of the team I would say." P4.

"So inspecting and adapting your product with the reviews and inspecting and adapting your process with the retrospective. So yeah, I would say those are thought-out ceremonies and useful ceremonies." P6.

3.2 Other regular events/ meetings

In general, there was no fundamental change in terms of ceremonies to those done in Scrum. Participants tend to add some meetings that seem to be useful depending on the situation of the team/ organization like a Roadmap Meeting, or a mini retro in case of long sprints.

"And what I also do is every two weeks, it's a mini retro, especially just like 30 minutes. Do we need to change the policies? Do we need to change the WIP limits? Do we need to change the swimming lanes? Do we need to focus on the process? So really Process wise" P1.

Also, the need for other regular events should be brought to a minimum, since this is the idea of the Scrum inherited events, and those should be incorporated already within them.

"And most of the time, all the other meetings that are around should be able to be incorporated in those moments." P4.

3.3 non-regular meetings

The need for such events is also tied to the team itself or the organization. Some participants suggested some non-periodic meetings like problem-solving sessions, and postmortems, which may take a place after a large-scale failure.

"So, if there's a real big or a real big error or failure or anything, and then at least we'll have a specific meeting focused on that specific topic or on that specific event that happened to also reflect on that. So more an additional retrospective or additional brief" P4.

4.1.4. Working mechanism:

4.1 Workflow visualization

Visualization in general can be different between one team and another. There is no way for doing it. The focus here is on the general theme that Scrumban teams use. The most important aspect of the visualization is that it is the part where all principles within the team come to life, for example, the "Pull mechanism", WIP limits, the policy toward blocked items, etc... This part is mainly taken from Kanban as Scrum does not describe any specific way for visualization.

From the participants' responses, it is noticed that most Scrumban teams are using three main stages that can be broken down into sub-columns. The stages are:

4.1.1- The Backlog

Where all known required features are visualized next to the board.

4.1.2- Ready

In this lane, all refined stories that are ready to be picked up are found. coming from the Backlog Lane. In the case of Scrumban with sprints, this lane may represent the sprint backlog for the team. It is worth noting that some teams may implement a separate "Refining" lane.

"We have in our workflow "New" lane, "Refining". And those that most of the time the product owner is working on them. Then we have "Ready for development". and when the whole team understands what needs to be done, or what we need to achieve, then it is picked up for development and the cycle time starts." P1.

4.1.3- Work-in-progress section

In this part, all work stages are visualized as much as possible, incorporating Scrumban's "Transparency" principle.

Under this section, many almost all the team's principles should take effect in the visualization part. Stages can use buffers like dividing the lane into two sections: "Doing" and "Done". This will help implement the "Pull Mechanism" so that the owner of the next stage can know whether the item is ready to be pulled or is still under work. So, each Colum is a backlog for the next process fostering the (Pull system).

Also dealing with the blocked items policy has a major effect on this part. According to the respondents' answers, there are two approaches to dealing with the blocked item. The first one is by keeping it in its current lane while giving it another label "Blocked" label. This policy was the most preferred one by the practitioners. The benefit of this policy is that it hinders the WIP limit, as a result, the team should try to solve this issue as soon as possible in order for them to be able to carry out the rest of the work as normal, fostering the Pull mechanism, the collaboration within the team, and helping to reveal the bottlenecks in the process.

On the other hand, the second way in dealing with the blocked items is to move them to a special lane (in the board underneath) in order not to hinder the WIP limit for that lane. WIP limits policy also has to be seen in the visualization.

"In dealing with the blocked items, after in progress, we have "on hold" for the blockers to actually separate it, and we also track how many days it's blocked." P5.

"They get a tag called blocked and then it turns red. It stays in the column where it is and it blocks your work and your WIP limit." P6

4.2 The WIP limit

Although some practitioners have pointed out the known equation for determining the WIP limit in the process: **Efficiency = VA / (VA + NVA)**

and then applying **the Total Tasks the team can work on at a time = Team size / Efficiency** Where (VA): are value-adding activities, and (NVA) non-value-adding activities.

The Scrum principle of "empiricism" plays a major role in deciding on the WIP limits. Almost all interviewees indicated using the number of people in the team as a starting point for the WIP limit, or the number of people plus one keeping some room for flexibility at the beginning. Then Scrum principles "inspect and adapt" are implemented by inspecting how it is going and then adapting this number according to the situation and bottle nicks faced.

"I think it's four now, and the reason is that just in the implementation, they wanted to make sure that all developers could start to work on new work" P6.

"And normally I gave them like good practices where I say you sometimes can use the amount of developers a plus one. And why plus one is just to get keep some flexibility. And then from there, we start to inspect and adapt" P2.

4.3 Documentation

All interviewees agree that the documentation in Scrumban should not be an exception to the Agile manifesto recommendation. Thus, to document to degree that it is not creating any waste.

"In general, it's nice, as lean, to not have too much documentation. But the only thing is that people use these kinds of things, also to say, oh, well, it's agile, we don't have to do documentation. And I'm not agreeing with that, because you do need some, of course, but I often linked that more to what the people work on and the specific product that they own" P3.

4.4 Other Scrum/ Kanban principles

All interviewees did not indicate using any other principles other than those pointed out in the previous two points 4.1 and 4.2.

4.1.5. Metrics

The metric used in practice and indicated by practitioners were:

54

Cycle time: is considered one of the most important metrics and was almost mentioned by all practitioners.

"Yeah, and cycle time is the most important one, in my opinion, which gives you a lot of information about your productivity, about your delivery, about your predictability as well" P2.

The ageing metrics like the age of the items in the backlog, the number of days an item is blocked, and the number of days an item is in progress (WIP age).

"And the ageing time is something you can learn from it. Where are we wasting some time? Where are we may be doing it faster as we can." P2.

Other metrics like the throughput are also considered one of the most important metrics as it helps identify whether the process is productive or not. Also, the cycle time scatterplot and cumulative flow diagram.

It is worth noting that Scrum velocity is still being used in some teams as well. "So the velocity, how many points and how many items we can deliver in the sprint? Yeah. And based on that, you can see a little bit, but again, yeah, it's sometimes hard to because, over time, you get better estimation, you might change the currency in which you're estimating." P4.

4.1.6. Conclusion on Scrumban Application

A glance at the Matrix model, the following outcomes in the Scrumban methodology can be deduced:

- There are disagreements about Scrumban sub-dimensions in the considered theory sources.
- There are some dissimilarities about some Scrumban basic sub-dimensions in the participant's practitioners' opinions.
- There are disagreements about sub-dimensions in the participant's practitioners' opinions and the considered theory sources.

Based on the previously mentioned points and considering the Praxis and Practice data, it can be concluded that applying Scrumban in the current state is not a straightforward process, as there is a variety of choices of how and why to apply a specific Scrumban practice. For example: Keeping Scrum roles or not, using cross-functional, specialized teams, or both, and what is the optimal team size. These examples may be considered an obstacle for Scrumban adopters.

Consequently, there is a need for a framework or a tool that can guide Scrumban new adopters to find out the most suitable way of applying Scrumban practices for their situation and the rationale behind it. Also, it can be beneficial for experts by sharing and comparing their knowledge and optimizing the way they are applying Scrumban.

In the next chapter, selector frameworks are proposed and discussed as tools to provide guidance to support a successful selection method from Kanban/Scrum practices.

4.2 Scrumban components representation model

Based on the interview results, with the contribution of the literature review and the interviewer's notes, the following model, illustrated in Figure (11), is introduced. In this model, the key components of Scrumban are located in the rectangular area, which is defined as Scrumban, and those components are shifted to Scrum rectangular area or Kanban rectangular area according to the proportion in which those are applied in practice. Although the model shows Scrumban components and provides a general overview of Scrumban. However, it is not intended to give an idea about the relationship between those components.

A glance at the model can show that Scrumban methodology relies on Scrum rather than Kanban. It shows in the rectangular area of Scrumban, that the adopted activities from Scrum are more than those from Kanban. The introduced model gives an answer to the sub-question of *what is the basis of Scrumban*? Thus, whether Scrumban is being applied on Scrum or Kanban bases. Based on the results in *4.1* the model shows obviously that Scrumban is a Scrum integrated by Kanban but not vice versa.

This can be for many reasons. Firstly, almost all practitioners agree on the fact that Kanban lacks a descriptive guide compared to Scrum which has clear rules.

"The main problem with Kanban is it does not have a guide. So they pick up some principles, and core principles, and people use those with different interpretations. With the Scrum, they're strict to certain rules" P5.

Secondly, most Scrumban practitioners have a Scrum background, leading to using Scrum as the main framework and Kanban as a strategy on top of it to get to the flow-based process. Especially with the widespread of Scrum, most teams might already be applying Scrum, so they take it as the basis.

"It is more like a strategy to get flow management into the system while Scrum is being the framework. I see it like a good edition to Scrum instead of replacing Scrum." P6. 56

"What I think is that most people seem that they come from Scrum. And they think, like, maybe. So that's their basis. So they start from what they're already what already is there. Scrum is, in the end, the most used way of working. So it's really practical" P3.

Another important reason to highlight and related to the second one, is the commercial success of Scrum, that leaded to companies being introduced widely to Scrum more than any other method.

"I have the feeling that a part of it is more than commercial success with the commercial story of Scrum. Scrum is commercially outed. There's a lot of there. So there are Scrum Alliance and Scrum .org at least as the main two, but also a lot of companies around them" P4.

A further aspect is the type of the product. Most practitioners see that Scrum is fitting for more products than Kanban which mainly fits with the operational team.

"It also depends on the product that the team is delivering. Is it a feature team again? Or is it a component team? And yes, in the feature team, of course, I would say yeah, Scrum is in lead. If you use Scrumban, why because you always start from Scrum." P2.



Figure 11: The allocation model of Scrumban key components between Scrum and Kanban.

4.3 Scrumban perceived benefits

It is seen that Scrumban can help Scrum teams that are having a hard time delivering at in steady pace, by combining visualization and workflow management from Kanban with the good structure of Scrum, leading to a better value delivery routine.

"If applied correctly, I would choose Scrum with Kanban practices over using them separately. The ultimate goal is to deliver value at a sustainable pace. Kanban practices can help visualize and manage the flow and Scrum helps to keep focus and work as a self-managing team to deliver value." P7.

Also, it is believed that Scrumban practices like applying the WIP limit, can bring the team's focus, enhance collaboration and ultimately a better value delivery.

"In that case, the WIP limits can help the team to create better focus and sometimes even force them to work together and get things finished. More focus within the sprint and therefore a more steady pace of delivering value (can be seen in a more steady burn down)" P3.

"So I would almost advise you to always try to use it whenever you've got the chance. Because it will greatly increase flow within your sprints and stimulate collaboration and focus on the team." P4.

"For teams using Scrumban over Scrum, it made them focus on delivering more and smaller items. Delivering faster and avoiding multitasking." P5.

In some situations, teams need a transition from Scrum to Kanban. Applying Scrumban is seen as a more appropriate way to facilitate this transition. Moreover, it is seen as a good way to experiment if this transition would work.

"Scrum ban gives a bit more structure than Kanban. If you manage Scrumban I think you are a few steps away from Kanban and maybe eliminate de sprint planning or sprint goal." P1.

"A more easy transition from one way of working to another, imagine a team that wants to switch from Scrum to Kanban. To see if it works for them you can implement Kanban elements into their Scrum way of working (so work Scrumban) and let them ease into it and test if it works." P3.

Further in this regard, the power of Scrumban comes from the fact that it combines two frameworks that can complement each other but do not conflict. Like Scrum structure and Kanban flow flexibility. By applying one of both the team will be limited to this framework and as a result, miss out on a chance to improve their performance.

"The advantage of Scrumban is that you apply both concepts, and thus benefit from both!" P4.

"I think if you do the combination, you get the best of both worlds. A team benefits from the structures that Scrum provides with its roles, events and artifacts. And using the metrics and WIP limits to manage the flow from Kanban helps keep the feedback loop going on a daily basis and not only on sprint level." P6.

"Sometimes the team is stuck or struggles with the framework itself, because either they want to do it by the book, and then you have people who are resistance about how certain things should happen, so they will be too much limited to the framework or its principles." P2.

Chapter 5 Discussion

According to the available Scrumban theory sources, Scrumban is frequently defined as a hybrid Agile development methodology that combines the structure of Scrum with the flexibility and visualization of Kanban [12] [8]. The flexibility of the resulting combination allows Agile teams to make pragmatic choices regarding the process that needs to be implemented based on their special situation.

Although Scrumban is well-known for its high flexibility and the wide variety of combinations of options, this flexibility can be worrying for teams that are new to Agile or lack an internal coach [69]. Especially since Scrumban is expressed as a new and untested methodology [70]. Furthermore, there is a lack of established best practices in the literature. Thus, teams usually tend to incorporate concepts based on their own experience [70]. Scrumban is also built to expand as the Agile team grows in maturity [69].

In this section, the research aims to address the above-mentioned problem, by proposing a general method that can help practitioners willing to adopt Scrumban based on their own needs and specific settings. The method will use the data gathered from the practitioners' viewpoints to portray justifications for each practice in a diagram which can be used to select the suitable practices based on their expected outcome. These diagrams are named selectors.

5.1. Selector's definition and work mechanism

In this chapter, several proposed selectors are designed to cover most of the Scrumban areas. The proposed selectors are depicted in consistent figures that contain the Scrum/Kanban practices corresponding to the Scrumban under consideration aspect. In addition, each selector has a pointer arm that moves toward the most preferred choice by the practitioners' responses. The selector is created to select one position between a few suggested positions. Each of these represents a Scrum or Kanban practice or in a few cases a merged case between Scrum and Kanban practices. The Scrumban team can choose the most appropriate choice according to their needs and situations. It is worth noting that the accuracy and the successful selection can be influenced by some factors that can support a specific choice, including, but not limited to team maturity, product nature and organizational structure and culture.

The following paragraph presents some examples to illustrate how the proposed selectors can guide for selecting the appropriate Scrumban practices.

5.1.1 Scrumban Team Roles selector

This selector provides guidelines regarding the available choices for the team roles in Scrumban, which are: "Kanban No Roles", and "Keeping Scrum Roles" according to practitioners' interviews. Using analyzing and extracting the relevant data, it is observed that most of the practitioners supported the position "Keeping Scrum Roles", so the pointer arm is pointing to position 2 "Keeping Scrum Roles". Each position is presented with related data that provide answers for why to select and what the attributes of this particular position are if selected. Starting with the most selected choice, position 2 will be described. Figure (12).

Position 2 " Keeping Scrum Roles"

The mentioned position is entitled "Keeping Scrum Roles". This position has some attributes associated with it.

1- Keeping Scrum roles helps focus on bringing value to the organization.

This is done by trying not to change the process that is already working, as much as possible, which the team is used to. For example: in case the team was using Scrum, then introducing new roles or removing them might introduce new obstacles that may distract the team from focusing on the process. This also agrees with the Kanban principle "Respect the current process, roles, and responsibilities"

"Especially the Clarity and the borderlines of roles and where they overlap are defined in Scrum. They make it a little easier for teams to focus on what is bringing value to the organization instead of bringing new ways of work to focus on" P5.

2- Smoother Communication

Also, roles can support smoother communication, where the role can provide regular communication lines between the product owner, Scrumban master and other team members.

"Communication also begins with roles, who is the communication line? Who are the ones working on the items, who to contact, who to debate?" P5.

3- Knowing the expectations of each role

When roles are in place, it is easier for each team member to know the exact expectations of him/her, and it draws the line about who should be doing what.

"Imagine you are responsible for getting things done (developers), responsible for managing stakeholders (product owner), and for ordering things (Scrum master). That's very difficult" P7.

"But I think the idea behind those roles that are in Scrum, is to separate, the process, the people that do the work, and the guy that's owning the product and can challenge" P3.

4- Accountability and responsibility:

Another Justification is accountability. Implementing the roles will help the team dived the work in a way that everyone is responsible for implementing a piece of work.

"And what I also like very much is the accountability. Differences of Scrum, as a Scrum team, you are accountable for certain stuff. So, it's not individual. But as a team, we need to get things done." P7.

5- Good guidance from Scrum

Scrum is mostly labelled as a descriptive framework. Moreover, in the literature, it is a well-studied framework, and as such is the Scrum roles. So, this makes it easier to apply and creates more clarity for the team on what are the duties, commitments, and boundaries.

"With Scrumban I think roles are meant to stay and they give good guidance. Scrum has a good guideline for roles and what they do." P5.

Position 1 "Kanban No Roles"

In practice, no teams were using special Kanban roles or no roles at all. The same applies to literature where roles in Kanban are hardly found. In general Kanban, references indicate that Kanban does not prescribe any roles since it is designed to adapt to any process in place, but those roles can be added according to the need.

Kanban literature recognizes the following justifications:

1- In a small project, unnecessary roles could lead to waste (or sub-optimization & micromanagement) [46].

2 - When adding roles, the team should make sure that the additional roles add value and do not conflict with other elements of the process [46].

3- Some resources describe two roles like Service Request Manager and Service Delivery Manager in addition to a Project Manager role in a big project. [71] [46].



Figure 12: Scrumban role selector

5.1.2 Scrumban team specialized or cross-functional selector

This selector provides some guidelines and information about what to select between the available choices related to adopting the teamwork type, and they are: "Specialized", or "Cross-Functional". The mentioned justifications will be discussed according to the sequence in the figure (13).

Position 2 "Cross-Functional Teams"

Cross-functional teams can be interpreted in two ways, Cross-functionality at the team level and the team member level.

"The term "cross-functional" can mean two things, we have cross-functionality in the people in the team. So basically, all the people in the team can do everything. And we have that certain people are still specialized in something. And secondly, the most common case and also it is more usual, is the cross-functionality between teams. So, when some people are experienced in the front end and others are more experienced in the back end. But we try to come to a point that we're actually going to be cross-functional for everyone." P4.

In both ways it has the following features:

1- Product independence (make the product independent of the people).

In agile philosophy, dependencies should be kept to a minimum, as it hinders the team's progress. Having a cross-functional team plays a major role in business continuity. That is because it helps reduce the dependency within the team and avoids the single point of failure problem, that might be a result of having knowledge silos within the team itself.

"T-shaped cross-functional teams people almost always give a value, because it has to do with scalability, and also dependency, which is a very good term in software engineering". P5.

"We try to make products independent of the personas of the team member. We think it makes more sense in any case." P5.

2- Team independence

Having cross-functional teams also helps reduce the team's dependency on other parties, when facing impediments. Commonly, teams aim to lower dependency on external support, by having all the knowledge needed within the team.

"We always try to focus on the cross-functionality, because we are still a product-oriented and not a services-oriented company. And in that case, we still need a team that can at least do almost everything by themselves, and not have any dependency on other teams" P2.

3- Improving the team skills to deliver a full product

Since it reduces dependency on other parties, cross-functionality encourages the team to grow new skills needed to deliver a full product without any external support.

"But we want to have all the skills needed within the boundaries of the team. So, for example, the People app team requires good knowledge of front-end development, react, and CI/CD testing. And so, the team needs to own those capabilities. Yes, that's what we meant with cross-functioning" P1.

4- A better collaboration and it integrates well with applying the WIP limit

Cross-functionality will create better collaboration. Having all the knowledge needed, the team will be more able to own the PBI's from A to Z.

"Because if you say okay, we are a cross-functional team and we are doing Scrumban, that means within the complete team, we focus on one item getting from start to end." P7.

"So the mission is that the team owns all the knowledge and skills that are needed to maintain the product and develop the product, right? That doesn't mean that everybody has the same knowledge level, covering the whole spectrum And when we see that a specific area needs more capabilities or more capacity, then somebody will start learning it. And we'll take his time." P1.

Also, cross-functionality integrates well with applying the WIP limit as it enables the cross-functional team to help each other with removing the blockers.

"By introducing WIP limits, at some point, they're only you're working on two or three stories and those stories are not busy not finished, but the development work has already been done. So do we need to test or do we need to do something else? And yeah, because of that WIP limit, basically, the rest of the team is forced to help out the other people." P4.

"That's due to the fact that because the blocked items actually limit the working progress, of course, so the team do their best, they swarm on it." P5.

5- Flexibility and enhancing the cycle time:

Cross functionality can have a positive effect on the delivery, as it produces flexibility in task assignment leading to faster delivery and a better cycle time:

"Because with cross-functionality comes flexibility. And it's way easier to really work on the actual priority. And we really think as a team, we'll get more focus, we get more really working together and having the same goal working on the same stuff. we think it makes us faster, it makes us more flexible. And we do think that it's more fun, and like, more enjoyable." P3.

Position1 "Specialization"

Specialized teams also have an additional value in Scrumban. This structure might be necessary for some situations.

1- Works better with component product type

specialized teams are necessary in case the knowledge needed to finish the work cannot be distributed equally within the whole team, instead each plays a role in the total process of delivery: *" so for a data team. And that means we have a component team, and we have the feature teams.*

So, in the data team, a specialized team is needed. Because we have the data scientist, we have the data engineer, we have the machine learning engineer, and we have the software developer. Because all these roles are needed to deliver the mission of that team, which is the component, which is the data algorithms, or maybe big data, or maybe some optimization for the data. So we need the specialties, we can't really say they all need to know data engineering and all need to know machine learning". P2.

2- Setting up an accurate WIP limit will become more important

The idea of imposing a WIP limit is that the team will be able to help each other to overcome the blocker issue as the knowledge is spread among the team members. This would be challenging in component product teams since the individuals are specialists in specific areas: *"If you have a specialized team then WIP limits will be more and more important"* P7.

3- Might lead to silos creation.

As it does not enforce knowledge distribution, specialized teams might lead to silos creation:

"And not having cross-functional teams creates silos that lead to lack of sharing information". P5.



Figure 13: Scrumban specialized / Cross functional team selector.

Chapter 6 Conclusion

6.1- Conclusion

This section summarizes the research process from the initial stage of setting objectives and designing the appropriate plan, then to the theoretical study within the available sources and literature review. In addition, the research methodologies used in this research are described, followed by the key findings and the conclusions and future work.

6.2 Summary

The main objective of this thesis study was to examine and explore Scrumban from the practitioners' viewpoints. This orientation created opportunities to assess the current situation of Scrumban performance in some teams that try to adopt it, in order to overcome disadvantages in other agile software development methods and to help those organizations address the problematics areas of Scrumban. Then, an extensive theory study and literature review were implemented to examine the available Scrumban resources and previous work, in order to evaluate the current Scrumban situation and to address contradictions and problematic areas that may hinder Scrumban's performance and its adoption level.

Relying on the findings from the theoretical study and literature review, seven dimensions of Scrumban were concentrated to cover the scope of this research. Those dimensions are Teams, Delivery Mechanisms, Planning, Estimation, Meetings & Ceremonies, Working Mechanisms, and Metrics. An analysis is implemented based on a strategy as practice approach that depends on collecting practice data about the addressed Scrumban dimensions and comparing those to the praxis data collected from the practitioners during the recorded interview discussions. The adopted approach has led to a framework (the selectors) that aims to help Scrumban adopters deal with the wide variety of options and the contradictions among resources, but also the differences found between the practical application of Scrumban and the literature.

6.3 Future work

There are possibilities to do further studies within the scope of this research on two axes:

6.3.1 Applying quantitative studies and analysis

In this research, a qualitative data collection method is adopted and played an important role in providing information to understand the processes behind the practitioners' viewpoint about Scrumban. Also, in-depth interviews were adopted to fit the qualitative data collection technique and were used to implement a comprehensive study. Quantitative analysis as another shape of analysis may light on the numerical side of this study such as quantifying the percentage of

practitioners' opinions about each dimension of Scrumban such as team roles, cross-functionality, team size and so on. Usual quantitative data-collecting approaches involve managing questionnaire with closed-ended questions, where the researcher asks a standard set of questions and nothing more, or questionnaires often make use of checklists and rating scales. These tools help simplify and quantify people's behaviours and attitudes [72]. This could verify the results of this research and validate them.

6.3.2 Scrumban (influencing) success factors studies

In this research, we have seen that the selection choice for the positions in the selectors can be influenced by outer factors. Some of those were discussed as they were mentioned by the interviewees, like the product type, but those were not deepened in, as they are out of the scoop of this research. Moreover, several studies have been done on agile success factors that influence software development project success, Because when organizations recognize those factors it will lead to enhanced project management strategies, which in turn results in improved efficacy, cost-effectiveness, and a better throughput, consequently all organization levels can benefit from that [73]. Therefore, it is suggested to apply similar studies on Scrumban to address Scrumban's success factors.

6.4 Outcome Vs Objective

The main objective of this thesis was to is to investigate theoretically and practically the issues surrounding Scrumban and provide solutions in order to improve Scrumban adoption. The study proposed a framework model that can be used to select Scrumban practices based on the desired outcomes. Overall, the thesis was successful based on the provided main points. First, spot some light on the differences between how Scrumban is being applied compared to the literature. Also, investigating the basis of Scrumban practices, and finally introducing the proposed framework model to help teams to decide on the appropriate Scrum/Kanban practices.

6.5 Research contributions

- Through the literature study, this research underlined the low Scrumban adoption. Additionally, it looked at the available literature about Scrumban and highlighted the issues around it, like the scarcity of resources and the wide differences among the literature.
- Using strategy as a practice framework to compare the practice and the praxis data, produced the following advantages:

1- Display the differences for some cases between the considered theory sources about the Scrumban subject under consideration.

2- Display the differences for some cases between the considered practitioners about the Scrumban subject under consideration.

3- Display the differences for some cases between the considered theory sources and practitioners about the Scrumban subject under consideration.

4- Summarize and focus on the key Scrumban practices and issues in depicted form, to provide important information about Scrumban application from theory.

5- Support the existing theory sources with newly proposed models that can light on the problematic areas in Scrumban for future works.

- Investigate the combination basis, explore the reasons behind it, and illustrate this visually.
- The proposed framework (selectors), has presented the following advantages:

1- A practical method for selecting the appropriate Scrum/Kanban practices and providing some adjustment, constraints and control against the broad flexibility and disagreements in the available Scrumban theory sources.

2- Categorize the practices and their advantages (the rationale behind each) for each option (position) of the selectors that cover the most important areas in Scrumban.

- This research drew attention to the current state of Scrumban, and how it is applied from practitioners' point of view.
- This research has presented ideas that can offer opportunities to develop and support the Scrumban method in future research.

6.6 Limitations

This research is influenced by various shortcomings. In this section, these are reviewed.

Literature was the main starting point for this research and based on this Scrumban's seven dimensions were constituted. However, as pointed out earlier, Scrumban literature is scarce and includes wide inconsistencies resulting in inadequate theoretical data. Most of the literature research was limited to certain cases and special conditions at hand. Thus, this might have influenced the scope or/ and foundation of this research.

Further, small sample size may make it challenging to verify if a specific result is a valid result. But the small sample size may be justified by the saturation point that interviews have reached. However, clearly, the likelihood of capturing most of the opinions increases with the number of people interviewed.

Another limitation of this study is that all participants have the same role in their teams, namely, Scrum masters and agile specialists/ experts. Although those are the people supposed to determine the practices and the ones involved in the adoption decision, interviewing other teams' roles would have helped enrich the information inquired about Scrumban, and would have another perspective about this adoption.

Although almost all participants have done some certifications or training in Kanban, another limitation is the fact that the participants have been using Scrum for a long time before Scrumban came to exist, and as a result, they have a stronger Scrum background than Kanban. Although this situation faced the researcher because of the lack of Scrumban adopters, this may have led to a bias in the information shared about the application of Scrumban in practice.

Bibliography

- I. P. Saliya Sajith Samarawickrama, "Continuous Scrum: A Framework to Enhance Scrum with DevOps," in *International Conference on Advances in ICT for Emerging Regions* (*ICTer*): 019 - 025, Moratuwa, Sri Lanka, 2017.
- [2] D. R. A. M. S. J. O. M. L. K. Banijamali A., "Empirical Investigation of Scrumban in Global Software Development," *Communications in Computer and Information Science*, vol 692. Springer, Cham, pp. 1-22, 14 September 2017.
- [3] a. C. H. Timothy M. Persons, "GAO Agile Assessment Guide: Best Practices for Agile Adoption and Implementation," United States Government Accountability Office, September 2020.
- [4] R. V. O. Murat Yilmaz, "A Scrumban Integrated Gamification Approach To Guide Software Process Improvement: A Turkish Case Study," *Tehnicki Vjesnik*, vol. 23(1), pp. 237--245, February 2016.
- [5] K. S. a. M.Beedle, Agile Software Development with SCRUM, Prentice-Hall, 2002.
- [6] A. Reddy, The Scrumban [R]Evolution: Getting the Most Out of Agile, Scrum, and Lean Kanban, Addison-Wesley Professional, 2016.
- [7] M. A. B.-z. a. I. jaluta, "Criteria for selecting software development models," in *GlobalSummit on Computer & Information Technology (GSCIT)*, Sousse, 2014.
- [8] M. a. R. R. Alqudah, "An Empirical Study of Scrumban Formation based on the Selection of Scrum and Kanban Practices," *International Journal on Advanced Science, Engineering and Information Technology*, pp. 2315-2322, 2018.
- [9] M. Hodgson, "Business Agility, Product Development. Avoiding the Scrumban trap," zenexmachina, 14 May 2021. [Online]. Available: https://zenexmachina.com/avoiding-thescrumban-trap/. [Accessed 07 October 2022].
- [10] L. B. a. Č. Majer, "Comparison of agile methods: Scrum, Kanban, and Scrumban," in *the 19th International Multiconference Information Society*, Slovenia, 2016.
- [11] a. R. Q. Aysha Abdullah Albarqi, "The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile software development," *International Journal of Information Engineering and Electronic Business*, p. 12, 2018.
- [12] K. Bhavsar, "Scrumban: An Agile Integration of Scrum and Kanban in Software Engineering," *International Journal of Innovative Technology and Exploring Engineering* (*IJITEE*), p. 9, ISSN: 2278-3075, Volume-9 Issue-4, February 2020.
- [13] Digital.ai, "5th State of Agile Report: Agile adoption accelerates across the enterprise," 2021.
- [14] C. Ladas, Scrumban-Lean Software Engineering-Essays on the Continuous Delivery of High Quality Information Systems, 2008.
- [15] Z. Masood, "Real World Scrum A Grounded Theory of Variations in," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING,, p. 13, SEPTEMBER 2020.
- [16] P. L. a. A. T. Mark Saunders, Research Methods for Business Students, Mateu Cromo, Artes Graficas, Spain: Pearson Education Limited, 2007.
- [17] M. Marshall, "Sampling for Qualitative Research," *Family Practice*, pp. vol 13, 522-525, 01 December 1996.
- [18] C. a. N. P. Boyce, Conducting In-Depth Interview: A Guide for Designing and Conducting In-Depth Interviews for Evaluation Input, Pathfinder International Tool Series, Monitoring and Evaluation-2, 2006.
- [19] S. A. Glaser B, The Discovery of Grounded Theory: Strategies for Qualitative Research., New York: Aldine, 1967.
- [20] E. N., C. Greg Guest, "A simple method to assess and report thematic saturation in qualitative research," *ploS one*, vol. 15, no. https://doi.org/10.1371/journal.pone.0232076, 2020.
- [21] G. B. A. &. J. L. Guest, "How many interviews are enough? an experiment with data saturation and variability.," *Field methods*, vol. 18, pp. 59-82., 2006.
- [22] H. P. Nayeem Showkat, "Non-Probability and Probability Sampling," in *Media and Communications Study*, e-PG Pathshala, July 2017.
- [23] U. Flick, The SAGE Handbook of Qualitative Data Analysis, SAGE Publications Ltd, 2014.
- [24] D. R. Thomas, "A General Inductive Approach for Analyzing Qualitative Evaluation Data," *American Journal of Evaluation*, vol. 27(2), p. 237–246, 2006.

- [25] M. B. &. H. A. M. Miles, Qualitative data analysis: An expanded sourcebook., sage, 1994.
- [26] R. Whittington, "Completing the Practice Turn in Strategy Research. Organization Studies," *SAGE journals*, vol. 27(5), p. 613–634, 2006.
- [27] R. Stephens, Begining of software engineering, indianapolis: John willy &sons, 2015.
- [28] M. A. L., M., C. I. O. M. Steven Palmquist, Parallel Worlds: Agile and Waterfall Differences and Similarities, Carnegie Mellon : Software Solutions Division, October 2013.
- [29] D. Bell, Software Engineering for Students, Edinburgh Gate: Pearson Education Limited, 2005.
- [30] T. B. S. S. Huub J. M. Ruël, "The Waterfall approach and requirement Uncertainty:an In-Depth Case Study of anEnterprise Systems Implementationat a Major airline Company," *International Journal of Information Technology Project Management*, 1(2), , pp. 43-60, , April-June 2010.
- [31] T. M. P. a. A. Team, AGILE ASSESSMENT GUIDE, Washington, DC 20548: U.S. Government Accountability Office (GAO), 2020.
- [32] Ł. D. Sienkiewicz, "SCRUMBAN THE KANBAN AS AN ADDITION," INFORMATYKA EKONOMICZNA BUSINESS INFORMATICS, pp. 75-81, 24 2 2012.
- [33] A. Cockburn, "Selecting a project's methodology," IEEE Software, 2000.
- [34] A. J. Nandhakumar J., "The fiction of methodological development. A field study of information system development," *Information Technology and People*, 1999.
- [35] M. J. E. Eva-Maria Schön1, "Agile values and their implementation in practice," *International Journal of Interactive Multimedia and Artificial Intelligence* -, pp. 61-66, january 2015.
- [36] Dr. Ayelt Komus Et al, "Status Quo (Scaled) Agile 2020," Koblenz University, Germany, 2020.
- [37] T. d. William Koot, "Agile Transformation. From Agile experiments to operating model transformation: How do you compare to others?," KPMG, 2019.
- [38] B. Leusink, "Agile software development process improvement in large organizations," Netherlands, 2012 August 20,.

- [39] K. S. A. M. T. B. M. Joey F. George, "The amorphous nature of agile: no one size fits all," *Journal of Systems and Information Technology*, 22 May 2018.
- [40] R. Wankhede, "Hybrid Agile Approach: Efficiently Blending," Harrisburg University of Science and Technology, 2016.
- [41] w. M. G. T. G. P. a. A. A. Dave West, "Water-Scrum-Fall Is The Reality," July 26, 2011.
- [42] F. P. Å. Karlsson, "Exploring agile values in method configuration," *European Journal of Information Systems*, 2009.
- [43] B. B. T. Richard, "Using Risk to Balance Agile and Plan Driven Methods," *IEEE Computer*, 01 June 2003.
- [44] G. L. j. P. Alexander menzinsky, Scrum master Scrum Manager: Core Subject Area I, Spanish: Iubaris Info 4 Media SL, January 2019.
- [45] K. S. a. J. Sutherland, The Definitive Guide to Scrum: The Rules of the Game, Scrum.org, 2017.
- [46] H. K. a. M. Skarin, Kanban and Scrum making the best of both, infoq, 2010.
- [47] J. N. Rashina Hoda, Self-Organizing Roles on Agile Software Development Teams, May 2014.
- [48] K. Schwaber, Agile Project Management With Scrum, Washington: Microsoft Press, 2004.
- [49] V. M. a. S. Drnovscek, "Agile Software Project Management with Scrum," 2005.
- [50] N. S. J. Linda Rising, "The Scrum Software Development Process for Small Teams," software IEEE, pp. 26-32, July/August 2000.
- [51] M. Ikonen, Lean Thinking in Software Development Impacts of Kanban on Projects, Helsinki Unigrafia, 2011,.
- [52] J. M. M. O. Muhammad Ovais Ahmad, "Kanban in software development: A systematic literature review," in (SEAA), 2013 39th EUROMICRO, Conference on (pp. 9-16). IEEE, 2013.
- [53] M. A. R. J. M. B. a. S. B. John Noll1, "A Study of the Scrum Master's Role," *Lero, the Irish Software Research Centre, University of Limerick, Ireland,* pp. 1-17, 2017.

- [54] D. J. Anderson, Kanban: Successful Evolutionary Change for Your Technology Business, Blue Hole Press, 2010.
- [55] M. K.-M. a. M. Strale, "From Scrum to Scrumban: a case study of a process transition," *IEEE*, pp. 140-149, June 2012.
- [56] P. T. R. a. S. Beecham, "TWINS This Workflow Is Not Scrum: Agile process adaptation for Open Source Software projects," in 2019 IEEE/ACM International Conference on software and system processes (ICSSP), 2019.
- [57] D. J. A. a. A. Carmichael, Essential Kanban Condensed, Seattle, Washington: Lean Kanban University press, 2016.
- [58] A. Reddy, The Scrumban [R]Evolution: Getting the Most Out of Agile, Scrum, and Lean Kanban, Addison-Wesley Professional, 2016.
- [59] a. Y. Y. Daniel Vacanti, "The Kanban Guide For Scrum Teams," Scrum.org, 2021.
- [60] Ł. Sienkiewicz, "Scrumban-the Kanban as an addition to Scrum software development method in a Network Organization.," in *Business Informatics*, Publishing House of Wroclaw University of Economics, 2012, pp. pp.73-81.
- [61] M. O. D. D. C. K. &. O. M. Ahmad, "Kanban in Software Engineering: A Systematic Mapping Study," *Journal of Systems and Software*, vol. Vol137, pp. 96-113, November 2017.
- [62] CollabNet, "The 14 th Annual State Of Agile Report," Digital.ai, 2020.
- [63] A. Stellman, What Is Scrumban?, O'Reilly Media, Inc., 2019.
- [64] V. MAHNIC, "Improving Software Development through Combination of Scrum and Kanban," in *Recent Advances in Computer Engineering, Communications and Information Technology*, springer, 2014, p. 8.
- [65] S. P. P. a. D. J. R. Neve, "Productivity Improvement of Software Development Process through Scrumban: A Practitioner's Approach," in 2018 International Conference On Advances in Communication and Computing Technology (ICACCT), Sangamner, Ahmednagar, India., 2018.
- [66] W. Patrick, "Implementing Scrumban- A short guide to implementing Scrumban at your organization," in Switching to Scrum: How to Implement Scrum in your Software Development Organization, CreateSpace Independent Publishing Platform, 2012.

- [67] A. J. a. J. S. Dag I.K. Sjøberg, "Quantifying the effect of using kanban versus scrum: A case study," *IEEE Softwatre*, 2012.
- [68] S. T. a. W. Müller, Hyper-Productive Knowledge-Work Management- the tame flow appraoch and its application to scrum and kanban, Leanpub, 2013.
- [69] Workfront, "Scrumban," 2021. [Online]. Available: https://www.workfront.com/projectmanagement/methodologies/scrum/scrumban. [Accessed 21 2 2022].
- [70] "The Definitive Guide to Scrumban Methodology," 6, September 2021. [Online]. Available: https://kissflow.com/project/agile/guide-to-scrumban-methodology/. [Accessed 20 2 2022].
- [71] M. Hammarberg, Kanban in Action, 2014.
- [72] S. M. S. Kabir, "METHODS OF DATA COLLECTION," METHODS OF DATA COLLECTION, p. 77, 25 June 2018..
- [73] D. S. Nguyen, "Success Factors That Influence Agile Software," *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, p. 51, 2016.

Appendix A: Interview questions

Opening question:

1- Do you agree if I record the interview in order to have a better analysis of the interview later on?

2- Can you tell me who are you and your work experience?

3- Can you give me a short description of the company/companies you are working for, and what is your role/service you do for?

4- For how long have you been using Scrumban? Almost how many teams of yours used/are using Scrumban?

Teams:

1- In my study, I found that Scrumban gives the flexibility to either use/keep roles or not to do. What do you think about using it? Do you use them?

2- Teams in Scrumban can be specialized or cross-functional. What are your teams? On which basis do you decide?

3- Unlike scrum, Scrumban is more flexible in terms of team size. What do you think about the ideal team's size? Could you elaborate on why?

Delivery mechanism, Product Backlog, Planning, and Estimation:

1- In the literature, it is found that Scrumban should incorporate the Kanban method of workflow based instead of sprints. What do you think is the optimal way of delivery in Scrumban?

2- We know that, for example, in Scrum, the product backlog is a collection of all known features

and requirements of the desired product, whereas in Kanban it is not necessarily so, and even in some Scrumban literature, it is found that some teams would define only the features that will be worked on in the next sprint/workflow but not more not less. This is mainly to avoid having a single prioritized list where the backlog grows to an unmanageable size. How would you define the product backlog in Scrumban in your experience?

3- In the literature, I found that Scrumban's product backlog can be different than in Scrum. In In another word, Scrumban takes the principle "**just in time planning**" from Kanban so that, advocates defining a high-level plan, strategy, or roadmap. Those are less likely to change in the short term and they help the team makes decisions when needed. As a result the backlog contains only items that will be worked on next.

So, In your teams, what are the contents of the product backlog, do you use fixed-size stories, or is it treated like in scrum (pool of features and requirements)?

4- How often do you do planning? Is it regular or as needed? Why?

5- In estimation, some Scrumban practitioners think that estimation should not be provided. So, instead, a fixed-size event-driven backlog may be more beneficial for reducing waste, i.e. creation

and discussion of too many user stories. In other words, smaller fixed-sized ones can do the purpose, others keep scrum estimation. What do you think about it and why?

6- What do you think about allowing changes to the product backlog/sprint backlog (in Scrum terminology) during the iteration/ workflow? And why?

Meetings and ceremonies:

1- What do you think about incorporating scrum ceremonies in Scrumban?

2- What other regular events/ meetings do you have in Scrumban? Could you describe the aim of each of which?

3- Other than those regular meetings, are there any non-periodic meetings you carry on?

Working mechanism:

1- I found that there are many ways to visualize the workflow in Scrumban. For example, visualizing the backlog before the board to pull from, dividing each column into doing/done, etc. How do you visualize the workflow of Scrumban? Could you tell me more about how the cards move through?

2- Scrumban inherits "limiting the work in progress" from Kanban. On which basis do you limit your WIP?

3- Do you incorporate other principles from Scrum or Kanban?

4- What do you think about documentation in Scrumban?

Metrics:

1- What metrics do you use to track productivity during and after the delivery in Scrumban? Could you elaborate more on the importance of each of these?

2- Some practitioners see that limiting the work in progress (WIP) can act like the velocity in scrum by not allowing the team member from taking on too many tasks to work on. What do you think about that and which one is used in your teams?

Scrumban perceived benefits:

1- In what situations or cases do you see that Scrumban is most suitable instead of scrum or Kanban separately?

2- What do you think are the advantages of applying Scrumban over one of Kanban or Scrum? Or in other words, what benefits result / have resulted from choosing to apply Scrumban in your teams over Scrum/Kanban?

Appendix B: The practice, praxis matrixes

Delivery mechanism 2

Delivery mechanisms (2)						
Planning Estimation Allowing the changes to the "WIP/ Sprint Backlog"						
Practices based three <u>references</u>	1 (Ajay Reddy,2016)	Planning takes place on two primary levels—release planning and sprint planning.	The importance of estimation is reduced under Scrumban, also factors that allow for estimation support higher confidence in the probabilistic forecasting methods.	Scrumban provides no rules that forbid changes in work plan at any given time Tasks in To Do state can be easily replaced with new ones, also tasks that are already in process can be taken back to To Do and more important tasks can be pulled in. (Lucija Brezočnik, et al 2016)		
	2 (Evaldas Bieliūnas, 2014)	Scrum-ban as well as Kanban does not prescribe a precise planning routine. Therefore, the teams can choose to plan when they run out of backlog items.	Estimation of the item duration is optional. Some teams still choose to carry out the estimation in order to have more predictability. Alternative approach is to make sure that each of the items is of the same size, and therefore can be completed in the same amount of time. (Evaldas Bieliūnas ,2014).	Scrum-ban allows adding new items whenever there is enough capacity in the queue. Therefore, Scrum-ban is very useful for functions with continuous flow of new items.		
	3 (Zahoor Ahmad Khan, 2014)	Scrumban has shorter planning meetings in order to update the backlog queue as and when required.	Therefore rather than estimating each and every story within an iteration, why not divide the stories into similar size items. And accordingly team can decide on selecting the fixed number of prioritized tasks	Unlike a sprint backlog (Scrum), Scrumban backlog may be updated more than once within a sprint.		
Praxis Based on <u>Practitioners</u> interviewees	P1	It's useful to have time slots. So every Tuesday or Thursday, we have refinements, and we use a time slot	We try to do estimations just to facilitate the discussion, and it helps more for inspection and adoption regularly	I don't support the change to the product backlog, but that depended on the necessity of that change.		
	P2	It's still the sprint planning, every two weeks, ,We call it the "daily planning" of the week. We see what actions needed	Although it's very helpful for forecasting. But the problem is that , the stakeholders or product owners may perceive it, as a promise, I think estimations is replaced by metrics	it depends how you would change. But the change should be always possible.		
	P3	It's every sprint. Every two weeks because we use the sprint. Mini planning during a standup, we make a roadmap for the next quarter.	Estimations is really to have some sort of predictability, But because of this hybrid kind we struggle with it, so we actually left it.	it was actually a challenge. We had support work and staff interrupting what we planed to do, and you want to keep it as a minimum,		
	P4	We do it at least every two weeks for formal planning, and some kind of official refinement planning.	I did estimation, and it is also to have some estimation that you can use for product owners and stakeholders	You sometimes you don't have a choice if it had happened once, well, there might be a good reason for it. But if it happens regularly, then there's another problem		
	P5	We don't have a regular planning session. (When needed)	We use the throughput metric as an estimation tool and based on the principle the good size. But the story point estimation doesn't really work, in my opinion.	means we have a sprint goal. If the goal is still there, and the changes will lead to meet that goal then do it.		
	P6	We don't do planning like in scrum. So we have the review, and we have a one-hour meeting on our calendar.	Story point estimation can be wasteful. The value of it is that you have the conversation of the developers and that they are aligned about what the work means or not.	I don't allow for the things have been started and it needs to be stopped because of something else. So I want him/her to focus on finishing things first.		
	P7	We have it on a regular basis in regards of backlog items. by the end of the week, for example	I use estimation. But to trigger discussion, create workable items. It's also easier to say, on average, we have four items a week. but need to calculate the average.	I don't think that's a good idea to add new items to it or to remove it or . Only if it becomes absolute. you don't need it .		
Identify the key findings based on contrast Between Practices and Praxis		Theory sources are not definitive about planning but are inclined to use flexible planning meetings. Also, the interviewees' data is divided and tends to have it as regular as in Scrum.	Some of Scrumban theory references show that estimation is optional, practitioners' viewpoints are to use estimation as a tool to reinforce discussion and conversation in teams.	Scrumban theory references, give freedom to make changes during iterations, while generally, practitioners are more strict in allowing changes only if they support sprint goals.		

Figure 14: practice, praxis delivery mechanism matrix.

Meeting and ceremonies					
		Regular meetings	Other meetings		
Practices based three <u>references</u>	1 (Ajay Reddy, 2016)	Daily meetings are as much a ceremony with Scrumban as they are with Scrum. Scrumban's mechanics improve the effectiveness while reducing the time and effort associated with conducting ceremonies.	Scrum ceremonies (and their timing) don't have to change under Scrumban, but the framework opens up a variety of alternatives that require less overhead, such as adopting specific continuous improvement.		
	2 (Evaldas Bieliūnas, 2014)	In Scrumban meetings are optional. They can be avoided entirely, or agreed upon on a regular or on demand basis.	The most common option of Scrum-ban meeting is short Kaizen event. It can be done from time to time in order to focus		
	3 (Gionata Carmignani et al , 2015)	Scrumban like scrum in: having ceremonies such as daily meeting and retrospective meeting, (Aysha Albarqi ,2018)	Kaizen is important and to improve the work, review and retrospective meetings from scrum are used. This ensures that incremental value is added after every iteration and that the team practices are efficient for future projects.		
Praxis Based on <u>Practitioners</u> interviewees	P1	Scrumban use The retrospective is every four weeks, do every two weeks, it's a mini retro like 30 minutes, and refinements session	like problem-solving session when there's a problem, you sit down with a team, that you solve the problem		
	P2	I say,scrumban use retrospective as a workflow, we do it monthly, stand ups It's the 15 minutes. It's still the daily Scrum	The refinement is to clarify what is this item is about, What is the goal of it.the roadmap planning it's a not regular quarter maybe.		
	P3	We did refinements in a meeting, every week or sprint, there were some meetings in the roadmap planning process we did every quarter.	When the roadmap process was going, we might, need more refinements, in a stand up we need something to refine this		
	P4	Scrumban meeting such as quarterly meetings session or bi quarterly even in some case?and this is used to align at the organizational scale	When a specific event is happened an additional retrospective, as an already planned meetings I call it a post-mortem or a pre-mortem		
	P5	We have retrospectives, stands ups, The review sessions are also very important, (Scrumban) sprint planning will not be necessary	We have the roadmap discussion at the unit level to get the people more involved		
	P6	The reviews and retrospective. are ceremonies Including daily scrum without planning, Tuesday /Thursday, are optional one hour meetings to refine stories.	The roadmap session, is held with developers, the stakeholders and one senior dev stays with the product owner to discuss the roadmap.		
	P7	I think it's very good. To incorporating Scrum ceremonies in Scrumban framework.	We didn't do any other meetings		
Identify the key findings based on contrast Between Practices and Praxis		Most of the literature references, introduce meetings and ceremonies as an option and recommend "Daily Scrum" and "Stand- ups". While practitioners recommend keeping Scrum meetings, others might adapt those such as using mini-retrospective meetings, and refinements sessions, and adding some others on top of these, like problem-solving sessions when there's a problem, and a roadmap discussion.			

Figure 15: practice, praxis meetings matrix.

Working mechanism

		😑 Work Visualization (Limiting WIP	Documentation
Practices based three <u>references</u>	1 (Ajay Reddy, 2016)	Visualizing the relationships the current state of each relationship, and the steps to a solution. It helps to resolve dissatisfaction.	Teams should limit the amount of WIP at any given time using explicit limits across the board, within each column Limiting WIP improves flow efficiency.	Formal Systems may include the documentation of processes, organizational charts, etc. Working software over comprehensive documentation is a core principle related to Manifesto. Extra detailed documentation increase wait time and deliver little value to customer
	2 (Zahoor Ahmad Khan, 2014)	Visualization is representing workflow stages. It is a visual control mechanism, indicating how work flows though different stages of the software development process.	It is one of the important aspects once a limit is reached within a particular stage, rather than starting working on something new, it is time to help someone else within the team.	No or minimum documentation. (Krunal Bhavsar et al, 2020)
	3 (Evaldas Bieliūnas, 2014)	While Scrum task board is defined and reset each sprint in Scrum-ban as well as in Kanban, board remains persistent and is not reset, as there are no pre-set periods for backlog item completion	Each team member should be working on no more than one task at a time. To reinforce this rule WIP limits from kanban are used, limiting the number of tasks in the progress columns. This reinforces team collaboration and resolve any bottlenecks.	Scrumban solves the Scrum limitation with lack of documentation policy. (Ioannis K. Kirpitsas et al, 2022)
Praxis Based on <u>Practitioners</u> interviewees	P1	Stages are: backlog, "Refining", then "Ready", to "Developing", "Done developing", "Review", "Done Reviewing" "Testing" and "Re-work"."	I prevent people are working on multiple BPI's, I count all the WIP limits, then I divide it by the team members, it should not be more than two	I think if the code is readable, and apply good comments, And a wiki page. you have this API generators that helps generate documentation
	P2	So sprint backlog, Product Backlog sprint backlog and implementation, "Code Review" "Testing", "Code Releasing", and "Done".	I gave them like good practices where I say you sometimes can use the amount of developers a plus one. The less WIP limit, the more we deliver	maybe only that decisions ,try to not make it as a goal but make it as a help material.
	P3	Product backlog to the sprint backlog. to do is contains the items that will be working on to sprint backlog team check, And the PO check	We use it as small as the number of team members, when we had five people, the limit in progress was four, that could create more urgency	I would say documentation is a little less simple and small as in Scrumit's nice to have as lean to not have too much documentation
	P4	Visually we have three columns, to do, doing done, work front end work, test work, and then the ops work, everything is waiting for QA.	depends on the number of people in the team. I like start with three in development and doing QA at some points	I think you need documentation, a lot of it can be in code And in comments. But also in the naming of variables or functions, or of components, writing unit tests, the unit test.
	P5	In non-active part, a "Refinement" column, Then "Ready" or "to do", The active part starts with the "In progress". Then "Team check". Then "Done". Then "on hold" for the blockers	For five people, the WIP limit for "In progress" plus "Team check" plus whatever, shouldn't be more than the number of the team members plus two max.	I think It is no need to have too much documentation
	P6	It is ready for Dev comes from the backlog. then implementing the development and the automated testing and code review, and the testing column. then it's ready to be released.	We have four for implementation, two for code review, and three for testing. This way, they make sure that all developers could start to work on new work.	I think just like the agile manifesto only when necessary and your product should be your documentation.
	P7	It is design, design has "in progress" and "done", then development in progress divided. So if it's done in development, then test	We should limit work in progress of the other teams so they can get things done quicker.	I think what really should be documented is policy, like when something should be done, so everybody can see what the criteria we have.
Identify the key findings based on contrast Between Practices and Praxis		Both theory sources and practitioners agree on the importance of visualization, each for their own reasons. However, the exact way to do it seems to be different on both sides.	The literature and the practitioners agree on limiting the WIP, because of the many benefits it beings. However, the way of deciding on the exact number seems to differ between the two sides.	The literature indicates that in general, the process may include documentation. Still, working software over comprehensive documentation is a key principle related to Manifesto. So, spending additional time on detailed documentation leads to additional wait time and deliver small value to the stakeholders. Commonly, most practitioners agree on a little documentation.

Figure 16: practice, praxis working mechanism matrix

Metrics

Practices based three <u>references</u>	1 (Ajay Reddy,2016)	The Scrumban provides a variety of metrics, it is used to monitor outcomes and gauge the level of performance. Productivity Metrics describe what the teams s can use to monitor overall performance. Quality metrics can use to monitor overall performance. Risk Metrics consist of both uncertainty and cost. It enables tracking a wide range of metrics to integrate it from other frameworks and models and visualize a wealth of information at the right times and places.
	2 (Evaldas Bieliūnas ,2014)	Scrum-ban as well as Kanban uses average lead and cycle time as its key metrics. The performance is measured via the cumulative flow diagrams and by estimating lead and cycle time. Cumulative flow diagram shows the total items that are in progress, as well as the time it takes to complete them. Average lead and cycle time estimations shows how much time, on average, does it take to manufacture and deliver one item.
	3 (Zahoor AhmadKhan, 2014)	Metrics are abused by managers and business stakeholders who want to unnaturally simplify a complex process into a one-dimensional number. Velocity, the amount of story points a Scrum team completes in a single Sprint, is such a metric that incentivizes lower quality at the end of a Sprint as a team scrambles to finish every last story they committed to.
Praxis Based on <u>Practitioners</u> interviewees	P1	The average cycle time of a story, we have the percentage of PBI's is that being finished within time box. how many It take more time it is scatter table. The lead time .we don't use WIP limit as velocity, because we do sprint planning.
	P2	In teams that are using a Scrumban in Sprint's we don't use metrics, we make some, adjustments, But for teams using Scrumban as a workflow, we focus on cycle time and ageing time. We don't use velocity, I agree on that (not needed) with limiting the work in progress.
	P3	We started and struggled with velocity, then experiment the cycle time, lead time, These metrics are difficult. because you have to track, I think work in progress limit is better at limiting the amount of tasks to act like the velocity.
	P4	The velocity, how many items we can deliver in sprint, cycle time and lead time those are powerful. I haven't been using with limiting the work in progress. As velocity but I see the advantage in that. And that is something that I need to start myself at some point.
	P5	We check mostly Cycle Time. Then the conversations about good size, and the "Block time" show bottlenecks, We're not using velocity. Limiting the Work in Progress is purposeful. So one thing at a time, work in progress is also meant for avoiding multitasking.
	P6	The cycle time, I was looking at the lead time also, and the age of the items the work item age, and blocked items it blocks your work. WIP limits are in place to improve flow through the system
	P7	Throughputs, is important. I prefer the whole picture how long it take from A to B. we track the time of overall, that impact on the throughput, and the age it spent in every stage of ours process. We definitely use (limiting the WIP),
Identify the key findings based on contrast Between Practices and Praxis		The literature references indicate a wide variety of options regarding metrics, and mostly show that Velocity is not recommended. This is also applicable for the practitioners, however, their metrics usage is focused mostly on fewer options like Cycle Time and lead time.

Figure 17: practice, praxis metrics matrix.

Appendix C: The selectors

Team size adjustor



Figure 18: Team size adjustor

Sprints/ Workflow selector

Scrumban Sprints/ Workflow Selector



Figure 19: Sprint/ Workflow selector.





Figure 20: Estimation selector.

Meetings selector



Figure 21: Meetings selector.

Metrics selector



Figure 22: Metrics selector.