

**HOW SOFTWARE DEVELOPMENT METHODOLOGIES  
AFFECT DYNAMIC CAPABILITIES UNDER EXTREME  
CONTEXTS: A COVID-19 STUDY ON AGILE AND  
WATERFALL METHODOLOGIES**

**Jyväskylä University  
School of Business and Economics**

**Master's Thesis**

**2021**

**Author: Nikita Salnikov  
Subject: Strategic Management  
Supervisor: Muhammad Imran**



JYVÄSKYLÄN YLIOPISTO  
UNIVERSITY OF JYVÄSKYLÄ

**ABSTRACT**

Author Nikita Salnikov	
Title How Software Development Methodologies Affect Dynamic Capabilities Under Extreme Contexts: a COVID-19 Study on Agile and Waterfall Methodologies	
Subject Strategic Management	Type of work Master's Thesis
Date 31.05.2021	Number of pages 64
<p>Abstract</p> <p>Software development methodologies affect different aspects of software as a service companies. This research studies a particular effect – how a software development methodology affects dynamic capabilities of a company. Another aspect that this research studies is whether these effects would be different under different contexts – normal and extreme contexts. In this research, a real-life case company is studied, which survived through the COVID-19 pandemic. It is analysed how the software development methodology helped and evolved, and how the effects of the methodology on dynamic capabilities changed.</p> <p>The research will first include some theoretical background and propose a framework. Then, the empirical part will come in through a series of interviews. The coding structure of the interviews will be laid out and the study results will be analysed. Then discussions and conclusions will be made.</p> <p>Main conclusions of this research revolve around the notion that the software development methodologies do affect the dynamic capabilities differently and these effects do change over time, depending on the evolutionary stages of the company, team, product, and the contexts that the company and team operate in. How and why these changes may happen is discussed further.</p>	
Key words Software Development Methodologies, Agile Methodology, Waterfall Methodology, Dynamic Capabilities, Extreme Contexts	
Place of storage Jyväskylä University Library	

## CONTENTS

1	INTRODUCTION.....	6
2	THEORETICAL FRAMEWORK .....	9
2.1	Software Development Methodologies .....	9
2.1.1	Overview of Software Development Methodologies .....	9
2.1.2	Description of Agile Methodologies .....	13
2.1.3	Description of Waterfall Methodologies .....	15
2.2	Dynamic Capabilities.....	17
2.3	Extreme Contexts.....	22
2.3.1	COVID-19 as extreme event and context.....	25
2.4	Summary of the theoretical framework .....	25
3	DATA AND METHODOLOGY .....	27
3.1	Methodological framework of the study.....	27
3.2	Brella case study company description.....	28
3.2.1	Interviews' description and questions .....	31
4	STUDY RESULTS .....	33
4.1	Study Results .....	33
4.1.1	Software development methodology in Brella and how it evolved	36
4.1.2	The effect of the software development methodology in Brella on the dynamic capabilities before and during COVID-19. ....	40
5	DISCUSSIONS .....	50
5.1	Theoretical contributions .....	52
5.2	Managerial implications .....	52
5.3	Future research directions and limitations .....	53
6	CONCLUSIONS.....	54
	REFERENCES .....	55
	APPENDIX 1 Brella platform feature overview.....	59
	APPENDIX 2 Interview questions.....	63

## LIST OF TABLES

Table 1. Global Usage of Software Development Methodologies.....	12
Table 2. Waterfall versus Agile-Scrum methodologies.....	16
Table 3. Essential Dynamic Capabilities.....	19
Table 4. Summary of interviews.....	32
Table 5. Summary of the methodology's effect changes on the dynamic capabilities...	51

## **LIST OF FIGURES**

Figure 1. Illustration of software Waterfall development methodology. ....	16
Figure 2. Typology of extreme events. ....	23
Figure 3. The theoretical research framework. ....	26
Figure 4. The Data Structure. ....	34

# 1 INTRODUCTION

The beginning of spring 2020 has influenced multiple spheres of life of many people. This event can now be prescribed as “extreme context”, which can be identified as “where one or more events are occurring or are likely to occur that may exceed the organization’s capacity to prevent and result in an extensive and intolerable magnitude of physical, psychological, or material consequences to – or in close physical or psychological proximity to – organization members” (Hannah et al., 2009, p. 898). Such extreme contexts put people and companies at risk as they have vast impacts not only over companies’ capacities to prevent and affect certain consequences, but also because they threaten actual survival of firms across multiple industries, thus threatening to invoke economic recessions and other negative consequences (Wenzel et al., 2020).

Now, to explain the motivations behind such a research, I, as a researcher, must go into explaining my personal motivations and a bit of a discourse in my work life and how such “extreme context” as explained above have affected my life. In the early 2020 I was working in a company called Brella Oy (referred to as Brella) as Quality Assurance Specialist and Project Manager. Brella operates in the event technology industry. In a nutshell, Brella provides access to event applications to the event organizers that decided to purchase Brella’s services. The reason for event organizers to purchase Brella’s services is that after getting access to the applications, they can set up and offer the applications to their event attendees, which the event attendees are then able to use for multiple purposes. Brella’s core customers have been business conference and tradeshow organizers. At times before COVID-19 outbreak, such conferences and tradeshows have been mainly organized in-person – i.e., with a real venue, where attendees would travel to and attend. Consequently, in Brella, the product and the whole way of working has been affected quite drastically when pretty much all in-person gatherings have been banned (Hadden, & Casado, 2020). Brella’s clients had to move to the virtual landscape – and so did the technology (Stokel-Walker, 2020). Brella pivoted their product to become a fully online tool (Paananen, 2020). This pivot was a rather interesting thing that was done by the Product team in Brella. As an employee, it even was subjectively the most interesting thing we had done in Brella throughout my employment. At some point in the Spring 2020, Brella had zero new sales and there were zero clients using Brella’s product. After the pivot that was done, Brella’s sales rates and the amounts of clients started to grow at a faster pace than they had been growing even before COVID-19. These rates continue to grow, at least up to the moment of writing this thesis. As an employee, I have been thinking a lot about what we have done there and how we managed to succeed during these times. I have also been seeing companies failing and ceasing to exist at the same time. As a researcher, I have been wondering, why did some companies manage to pivot and other did not? Why did Brella’s management decide to pivot, and managed to do so successfully, whilst other companies did not? What have been the capabilities that allowed Brella to do a successful pivot? Could anything have been done better?

This is a very interesting topic for me, especially because I am working in the field of product management. Such research would potentially help me with understanding the best ways to establish and maintain a product team which is efficient, able to work well both in times when “things are running smoothly”, and when “things go south”. I want to understand what the software development methodology behind

such a team should be and what effects it can have on the dynamic capabilities of an organisation. Moreover, I want to have an overview and try to study not only the best software development methodologies, but also understand what the weaknesses and strengths of some of the most popular software development methodologies are. These are the main motivational factors for the researcher from a practical point of view.

Therefore, this research is aimed at studying the following three main concepts: software development methodologies, dynamic capabilities, and how the methodologies affect dynamic capabilities under extreme contexts. This research analyses the topic of the software development methodologies, studies, and describes some of them fully, to understand their main differences to better identify what is the methodology in the case study company of my research. Once the methodologies to study are identified, the researcher studies the next main concept – dynamic capabilities of a company, and the factors that affect the company’s differences between them. Moreover, the researcher will need to introduce and better describe the concept of extreme contexts – what it is and how it can affect companies’ operations and companies’ dynamic capabilities. After I have researched the software development methodologies, what are the most popular ones and what are the main characteristics of those, I will then study what dynamic capabilities can be affected by the software development methodology in theory. These theoretical assumptions will then be tested out on the real-life case study company. Moreover, I will also study how these effects of the methodology on the dynamic capabilities have changed (if they have changed) for the study company during the last year – the year of COVID-19 – when the study company of choice was under the influence of COVID-19 extreme context.

The research objectives of this thesis therefore include three main parts. They are presented below.

1. Explore the topics of software development methodologies, dynamic capabilities, extreme contexts.
2. Elaborate on how these concepts correlate with each other and whether software development methodologies affect dynamic capabilities in any way and if so, what are the effects in normal circumstances.
3. Study whether these effects change during the extreme contexts.

Having the three objectives set for the research makes it easier to also set the two main research questions for this paper.

1. What is the effect of the software development methodology on the company’s dynamic capabilities during the normal time? What characteristics of software development methodology have effects on dynamic capabilities?
2. What is the effect of the software development methodology on the company’s dynamic capabilities under extreme context? Does the effect change? If so, how, and why? What characteristics of the software development methodology influence any change?

This research can benefit the academic literature in the field of software development methodologies and dynamic capabilities, and extreme strategic contexts. Some of the main concepts studied include Agile software development, Waterfall software development, extreme contexts and critical events, dynamic capabilities. The

scientific motivation of this research lies in low presence of studies done in relation to this research topic – there are not many studies available for software development methodologies and relation of software development methodology to dynamic capabilities nor on how these two concepts interact in times of extreme contexts. There have been studies on extreme contexts, what effects they have on companies and how they can affect strategic decisions (e.g., Wenzel, Stanske, & Lieberman, 2020 and Hällgren, Rouleau, & De Rond, 2018). There also have been studies done on software development methodologies (e.g., Wenxiao et al., 2017, Geambaşu et al., 2011) and dynamic capabilities (Schilke et al., 2018). The abundance of studies of how different software development methodologies can affect dynamic capabilities in times of extreme contexts and whether these effects can change is the main motivational factor for the researcher from the scientific point of view.

Besides lacking scientific knowledge on this subject, the timely manner of this research also brings additional value – while the global COVID-19 pandemic is still ongoing for the time when writing this research (Myers, 2021), we are most probably yet to experience and study all the long-lasting effects of this extreme event. Therefore, this research could also potentially benefit the future studies and understanding of how the COVID-19 pandemic affected the world we all live in.



## 2 THEORETICAL FRAMEWORK

The literature review and theoretical framework description are carried out in a classical way of exploring publications on the topics related to this research.

The first part of the theoretical framework used in this paper includes studying different software development methodologies. After introducing the methodologies concept, two most relevant methodologies are chosen for further research. The second part consists of studying the dynamic capabilities concepts including, what are its main mechanics and patterns, how it can be affected by the software development methodologies that software companies have. Then, the research will study the last piece of theoretical framework – extreme contexts including, what it is and how extreme contexts influence the effects a software development methodology has on dynamic capabilities of a company. Finally, the theoretical framework explaining the theorized nature of effects of these concepts will be presented. This theoretical framework will then be “tested out” in the empirical part of the study.

### 2.1 Software Development Methodologies

#### 2.1.1 Overview of Software Development Methodologies

Software development methodology, according to Avison & Fitzgerald (1995, p. 261), “is a collection of procedures, techniques, tools and documentation aids which will help the systems developers in their efforts to implement a new information system”. To understand different software development methodologies better, it’s important to acknowledge what are basically the things that need to be done in any information system (software) project. According to Jones (2018), there are at least twenty essential activities that pretty much each software project must do and pretty much any software development methodology included in it. These tasks are presented below (Jones, 2018, p. 1-2).

1. The project must be estimated in terms of costs – such as financial, time and people resources.
2. The project should be analysed from the sides of project quality and project risks.
3. The project normally has certain requirements of what it should be in the end so that it would fulfil the user’s requirements as to what job this software should do for them. Therefore, the project should have such requirements studied and documented.
4. Certain legal mandates should be considered (for example, as studied by Orpana (2019, p. 44), in IT projects legislative issues such developments of GDPR legislation in Europe often is perceived as an uncertainty and lack of information factors for the people working on the software project).
5. If the project is of a larger size, then oftentimes some system architecture design & development work may be needed.

6. There should be some level of design and visual interpretation of the to-be features and interfaces.
7. There should be some code created or some reusable code should be acquired.
8. The code developed for this software should be integrated to the other code and controlled.
9. Pre-tests should be run to avoid early-stage defects; such pre-tests include code inspections and static analysis.
10. Test cases should be designed and developed to test the working software later.
11. Testing should be executed; normally testing is done either manually or with automated testing tools.
12. Defects (referred to as bugs in software development), should be identified and reported and/or repaired.
13. Oftentimes software development methodologies prescribe controlling the infrastructure and code configuration for all the changes that have been made to the code.
14. Oftentimes software development methodologies also prescribe to certain user documents and educational materials to be prepared (such as help centre articles and/or technical support materials).
15. If the software being developed is for commercial use, then certain marketing materials for promotion of the software are normally a part of the software development process.
16. Many methodologies also imply that progress and cost information available for this software project should be collected and analysed.
17. In software projects, the user requirements described in point 3 of this list might change, therefore any methodology also dictates on how these changes should be communicated, controlled, and implemented.
18. If the software application consists of multiple pieces, then it should also be put together for the users to be able to use it.
19. The quality of the software being delivered should also be analysed from different perspectives to ensure that the key deliverables have been fulfilled.
20. The management team must approve delivery being made.

As further discussed by Jones, the generic activities described above are present in most of the methodologies known to date this way or another (Jones, 2018, p. 2). The differences between methodologies appear in multiple factors.

The first and potentially main reason for differences between methodologies constitute, as Abrahamsson et al. (2002) mention in their research, for differences between how the requirements of the software project are handled in mainly two different types of the software development methodologies. They mention that there's a difference between the more traditional methods (which are focused on the plan and step-by-step execution; thus also called plan-driven), which have a philosophy of locking down the requirements (the user requirements described above) completely before the design and development parts kick in, to the more agile methods, concentrating more on flexibility, adaptability with ability to make changes later in the development as well. There are more particular differences between more agile and more traditional methods, which are described below, but this is just one of the criteria how the methodologies can be categorized – by the way they handle the software project requirements. (Abrahamsson et al., 2002, p. 8-13).

Geambaşu et al. (2011) also found in their paper that some of the main factors that companies use when deciding on which methodology to use include the following.

1. Clarity of initial requirements,
2. Accurate initial estimation of costs and development time,
3. Incorporation of requirements changes during the development process,
4. Obtaining functional versions of the system during the development process,
5. Software criticality,
6. Development costs,
7. Length of the delivery time of the final system,
8. System complexity,
9. Communication between customers and developers,
10. Size of the development team.

Another major and interesting differentiator between methodologies is how they treat the software project – i.e., as being a totally new software project, or whether they account for a more maintenance and enhancement of legacy software project, supporting and configuring protection from cyber-attacks, commercial off-the-shelf packages, enterprise resource planning and open-source project modifications. Most of the methodologies known today assume the “new development” as the basis of any software project. That’s a rather interesting observation because as studied by Jones, the mixes of software projects during the last fifty years in a one Fortune 500 company has changed 360 degrees – fifty years ago most of the software projects were the new developments, while now of the project of this company are in maintenance and legacy improvements. At the same time, as mentioned by Jones, most of the business processes are automated in today’s world. Therefore, companies should account for legacy and enhancement efforts more than before. Jones argues that today the #1 cost driver of the entire software industry is the cost of finding and fixing bugs. Therefore, the orientation on the quality software development should be one of the main goals of today’s software development methodologies, and many methodologies don’t account for that. This may be an important factor to consider when studying the software development methodologies, even though such differences between software projects that a company handles probably differ from case to case. (Jones, 2018, p. 3).

Another important aspect to consider when analysing software development methodologies, as stated by Jones, is that for a long time the methodologies haven’t really been measured or evaluated anyhow – the software productivity and quality has always been a subject of debates (Jones, 2018, p. 14). As stated by Forsgren et al., there have been many attempts to measure the performance of the software teams where most of these measurements have been focusing on productivity, which in turn inflected two major drawbacks – focusing on outputs rather than outcomes and focusing on individual or local measurements rather than team or more global measurements (Forsgren et al., 2018, p. 45). The problem with measuring successfulness and usefulness was the main reason so many different techniques and methodologies of how to develop a software appeared – Jones mentions there are more than sixty methodologies nowadays (Jones, 2018, p. 13).

In summary, there has clearly been an abundance of software development methodologies appearing during the whole time when software development has been a topic for humankind. In a nutshell, all software development methodologies serve the

same purpose and must include similar activities and procedures one way or another. Most of the methodologies are mainly aimed at new software projects rather than maintenance and legacy improvement projects (Jones, 2018, p. 13). Therefore, it seems safe to assume that not all methodologies fit all projects and all companies. The lack of clear and working measurement frameworks and techniques has led to increasing numbers of methodologies, tools, and practices of how to develop a software.

One of the interesting classifications that Jones (2018, p. 37-38) introduced in his study, was the global method usage, which he derived from data of what methodologies the clients of his company used. Due to the lack of other data sources available on the global methodology usage and because the case company studied in the empirical part of this research use the methodologies highlighted in Jones' research, the use of Jones' classification seems to be reasonable.

Below is the table of global usage of thirty development methodologies, as adapted from Jones (2018, p. 37-38).

Table 1. Global Usage of Software Development Methodologies.

	<i>Methodologies</i>	<i>Approximate Method Start Year</i>	<i>Global Method Usage 2016</i>
1	Git development	2005	2,200,000
2	Legacy repair development	1960	775,000
3	COTS Modifications	1969	490,000
4	Agile/Scrum	2001	435,000
5	Waterfall development	1960	385,000
6	Prototypes: disposable	1959	275,000
7	Container development (65% reuse)	2012	76,500
8	Microsoft solutions	1999	73,000
9	Structured development	1973	65,000
10	Mashup development	2006	63,000
11	Legacy renovation	1995	61,000
11	ERP modification development	1996	60,000
12	Object-oriented (OO) development	1985	57,000
13	RUP from IBM	1996	48,000
14	Legacy replacement development	1989	47,000
15	Lean development	2003	46,500
16	DevOps development	2010	45,990
17	Iterative development	1990	43,000
18	Reengineering	1999	42,000
19	Spiral development	1983	36,000
20	CMMI development	1985	35,000
21	Prototypes: evolutionary	1965	34,000
22	TDD	2005	30,000
23	Micro service development	2014	23,400
24	Kaizen development	1955	23,000
25	Model-driven development	2009	18,000
26	Evolutionary development (EVO)	1993	17,885

27	Anti-patterns	1955	17,000
28	Cowboy development	1955	16,863
29	Feature driven (FDD)	2007	16,500
30	IE	1980	15,000

Source: Adapted from Jones (2018, p. 37-38).

The way Jones calculated the usage of the methodologies was through extrapolating the amounts of software projects of his clients that use a certain methodology (Jones, 2018, p. 30).

The interesting notion that can be derived from this table is that the 3 most widely used methodologies are the methodologies that are mostly suited for maintenance and legacy updates of the previously built software projects (Jones, 2018, p. 30). Therefore, they are not fully suitable for this specific study as it's aimed more at the methodologies that could combine both legacy improvement updates and new developments. Thus, the two methodologies that are more interesting for the study are the fourth and the fifth most widely used methodologies – namely the Agile/Scrum and Waterfall (highlighted with bold in the table above). The empirical study company present in this research also uses Agile/Scrum and Waterfall methodologies. It is also wise to limit the research to only these two methodologies to make it more feasible and viable. The comparison between these two methodologies will be the main question this study tries to answer from the software development methodology perspective.

### 2.1.2 Description of Agile Methodologies

As stated by Jones (2018, p. 49-50), Agile with Scrum is currently the most popular software development methodology in the world for new projects. Agile is both an evolutionary method based on iterative development and a new approach that has been popularized by the famous “Agile Manifesto” published by a group of software practitioners and consultants in 2001 (Jones, 2018, p. 49-50; Abrahamsson et al., 2001, p. 13-14; Beck et al., 2001). The main principles of Agile Manifesto are as follows:

- Individuals and interactions over processes and tools,
- Working software over comprehensive documentation,
- Customer collaboration over contract negotiation,
- Responding to change is better than following a rigid plan.

Therefore, as Abrahamsson et al. (2001) mention, these central values that the Agile Manifesto brings also adhere to the following main concepts of the agile methodologies.

First, the agile movement emphasises the importance of human interactions and communication between the developers, designers and any other stakeholders in a software project over following rigid processes. Agile methodologies therefore implies that there are to be set several activities that imply boosting team spirit, close interaction between the stakeholders, and the human aspect of the work is considered with a higher importance. (Abrahamsson et al., 2001, p. 13-15).

Secondly, the next main objective is to concentrate on activities that help in continuously releasing tested and working software. Oftentimes, Agile methodology

refers to frequent intervals between releases (from hourly to monthly). The developers are urged to keep the code simple, straightforward, yet advanced, which should also help in making it easier to keep the burden of updating complicated documentation to lower levels. (Abrahamsson et al., 2001, p. 13-15).

Thirdly, the emphasis is put on delivering business value of the project from the very beginning through constant interaction between the stakeholders. Thus, changing of the requirements and changes to contracts is possible, and strict following of contract conditions and requirements is of a lesser importance than bringing the actual business value through the software. (Abrahamsson et al., 2001, p. 13-15).

Finally, all stakeholders are prepared for the changing requirements and environments and are prepared to use the tools and activities necessary to achieve the business objectives set by the project. The Agile methodologies therefore imply using tools and activities that can facilitate such potential changes and developments. (Abrahamsson et al., 2001, p. 13-15).

Cockburn (2002, p. xxii) mentions that Agile methodologies often imply using “light-but-sufficient” rules for executing projects, where the emphasis should be put on human- and communication-oriented activities. Cockburn (2002, p. xxii) also mentions that sufficient rules will help the project to get executed, whereas the “lightness” characteristic of an Agile project would help to stay manoeuvrable and adaptive to the changing environment, thus being able to stay focused on delivery of the software and the business values. Cockburn (2002) also argues that ultimately, the following activities are necessary for an Agile project to be executed successfully.

- Two to eight people in one room, which would help with achieving better communication and community feelings.
- Onsite usage experts, which improve the feedback cycles from the stakeholders and users.
- Short incremental cycles, which help to address testing and repairing the software faster.
- Test-oriented approach in building software. Automated regression, unit and functional tests would stabilize the code and allow for continuous improvement.
- Experienced developers, who would help to speed up the development times for software projects.

Abrahamsson et al. (2001) then conclude that all Agile software development methodologies have four main characteristics.

1. Agile software development methodologies must be **incremental**, where small software releases with quick cycles help to deliver the feedback and results faster.
2. Agile software development methodologies must be **cooperative**, where all stakeholders work constantly together to ensure close and effective communication.
3. Agile software development methodologies tend to be **straightforward**, where the methods are not requiring rigid tools and activities, but instead are rather easy and intuitive to implement and set up.

4. Agile software development methodologies tend to be **adaptive**, where the changing environments, requirements, and conditions are perceived as something normal, and the tools and activities facilitate making changes easily.

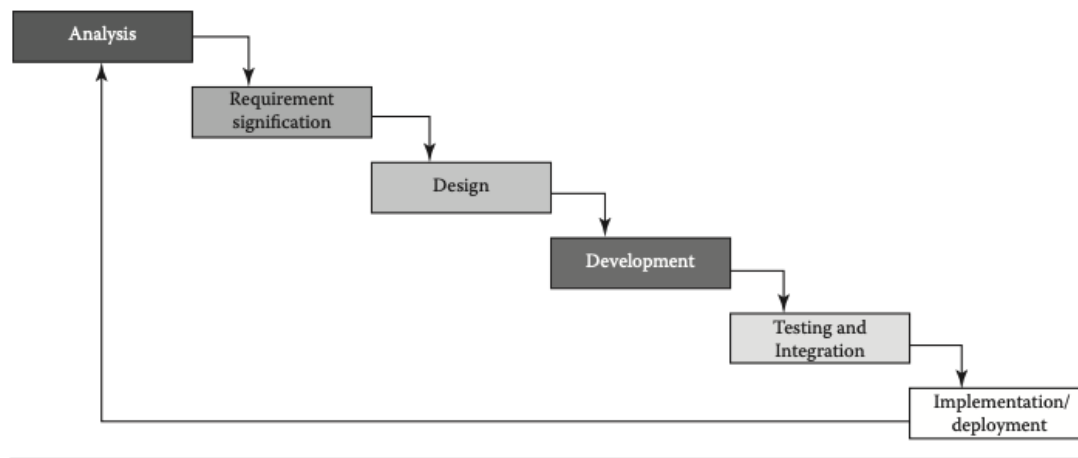
Agile includes a number of variations and methodologies that use similar approaches in building the software. Jones (2018, p. 50-51) mentions that similar methodologies include: Agile Lite, Agile Unified Process (AUP), Disciplined Agile Delivery (DAD), Extreme Programming, Scrum, Crystal development, Test-driven development, Agile with CMMI, Agile with DevOps, Agile with TSP, and Agile with Waterfall. Abrahamsson et al. (2001) also mention that more methodologies could also be referred to as Agile: Feature Driven Development, Rational Unified Process (RUP), Dynamic Systems Development Model, Adaptive Software Development, Open-Source Software Development, Agile Modelling, and Pragmatic Programming.

Having the abundance of hybrid methodologies that use different principles of Agile makes clear that most companies use different methodologies and different mixes of these methodologies. Therefore, it should be safe to say that there are companies that lean more towards Agile methodologies as well as companies that lean more towards Waterfall methodologies, and that there is not really a case where companies have pure Agile or pure Waterfall methodology. Because of this, the researcher will leave the description of Agile methodologies as it is and then describe the particular methodological situation in the case company and use the Agile methodology description described above to identify whether the methodology the case company has is more Agile or more Waterfall. Therefore, the question now is, what is the Waterfall methodology and what are the main principles and characteristics of the Waterfall methodologies?

### **2.1.3 Description of Waterfall Methodologies**

As mentioned by Jones (2018, p. 523), Waterfall development is the second oldest method after the Cowboy development, when in the 1960s the software teams and projects started to grow, and they started to require more organized and structured practices. Jones also mentions that Waterfall is still a very widely used methodology (Jones, 2018, p. 523). Waterfall methodologies is a rather simple concept and, one can say, a simpler one to implement than the Agile methodologies. The name Waterfall comes from how the visual representation of the development process under this methodology resembles a stream flowing over a series of waterfalls as shown in the Figure 1 below.

Figure 1. Illustration of software Waterfall development methodology.



Source: Adapted from Jones, 2018, p. 524.

This development method implies that the new stages do not start until the preceding stages are finished, hence comes the sequential nature of the Waterfall methodology (Castello, 2016, p. 200). The Waterfall methodology is implied to have very distinct analysis and design, build and test phases with big efforts put on requirements gathering, specification and signification (Castello, 2016, p. 200). At the same time, Jones mentions (2018, p. 523) that it is rather rare in real life that phases need to be completed before the next phases begin – normally requirements are usually only about 50% complete when the design starts; design is only about 60% complete when coding starts; coding is only about 35% complete when the testing starts. In this sense, the Waterfall methodology may seem more like a controversial topic as to how it actually normally works in practice. We can assume that again, there are different tendencies in different companies to lean towards a more documentation- and requirements- driven approach (more Waterfall like) or otherwise less documentation and more communication, reaction to feedback (more Agile like).

One more interesting aspect that Jones mentions (2018, p. 523) is that in Waterfall the design phase attempts to design the full system in the very beginning, while in Agile approach that is often considered differently, where in the beginning, they only may have the rough approximation of the design of the full system in the end.

Castello (2016, p. 200) also provides an interesting comparison between the Agile approach (in this case, he uses Agile-Scrum rather than Agile in general) and the Waterfall approach. Even though this research may not necessarily focus on the Agile-Scrum exploration, but rather simply Agile in general, this kind of comparison still gives the overall idea of where the Waterfall and Agile methodologies head the development towards. The comparison is presented in the Table 2 below.

Table 2. Waterfall versus Agile-Scrum methodologies.

Waterfall methodology	Agile-Scrum
Has very distinct analysis and design, build, test phases	These phases exist, but not necessarily in sequential order as they run within short, repetitive periods



Requires detailed documentation of design to be done	Requires minimal documentation, in extreme cases, the code is the documentation
End-users are consulted for their requirements during the analysis phase	End-user representatives are an integral part of the project and are involved during the whole duration of the project and regularly consulted
After the design is signed-off by the user, the build phase is conducted which will adhere to the signed-off specifications	Build is an inherent and iterative process, users are presented with the result, from which they critique, and modifications or new features are added
Scope is clearly defined at the beginning of the project	Scope is iteratively decided upon as the project progresses
Project teams can be large, composed of many different skillsets of people	Project teams are composed of 6-12 people maximum
Heterogeneous team with distinct, specialized skills per team member	Homogenous team, tasks can be accomplished by any team member

Source: Adapted from Castello, 2016, p. 200.

As can be seen from the comparison, Waterfall methodologies concentrate on rigid following of the structure, with no surprises and little alterations to the initial plan. These methodologies could assumingly work well for companies that require this kind of processes and rigid attitude towards development of the product.

As Jones also mentioned (2018, p. 525), Waterfall methodologies has come to be also rather successful in being parts of hybrid methodologies, combining Agile and Waterfall, combining Waterfall and CMMI.

With this information, it is possible to assume that there are certain distinct features of companies that use more Agile approach and more Waterfall approach in their software development.

The next thing that would be interesting to study is to try and understand what value a certain methodology with its characteristics brings to a company. Does it make a company stronger or weaker? How do you evaluate the company's strengths? How does the way the company builds its software relate to company's other strengths or weaknesses? That is where the concept of dynamic capabilities could come into play and could be interesting to study next.

## 2.2 Dynamic Capabilities

According to Eisenhardt et al. (2000), the dynamic capabilities in the firms can be built using the theoretical framework of Resource-Based View of the firm (RBV), which helps to understand how competitive advantage within firms can be achieved and how this advantage can be sustained over time (Barney, 1991). RBV framework conceptualizes firms' resources as bundles and that those resources can be heterogeneously distributed across firms and that this heterogenous nature persists over time. Researchers then generally assume that companies that have resources that are valuable, rare, inimitable and "nonsubstitutable", can achieve bigger competitive

advantages with those resources by utilizing them in a way that is not easily replicable by creating a unique and value-creating strategy (Barney, 1991).

Since the resources are at the heart of the RBV view, they become the actual assets that bring the actual essence of the value-creating strategies. The resources in this sense can include multiple things: physical (geographic location, specialized equipment, etc.), human (expertise in chemistry, etc.), organizational (superior sales force, etc.), and other local abilities or “competences” that bring the competitive advantages of firms in their respective industries (Barney, 1991). In volatile and fast changing markets, realities shift fast, and resources become less effective, whereas dynamic capabilities become more relevant (Eisenhardt et al., 2000).

Eisenhardt et al. (2000, p. 1107) therefore define the dynamic capabilities as:

*“The firm’s processes that use resources – specifically the processes to integrate, reconfigure, gain and release resources – to match and even create market change. Dynamic capabilities thus are the organizational and strategic routines by which firms achieve new resource configurations as markets emerge, collide, split evolve, and die.”*

Teece (2007, p. 1319), in turn, mentions that:

*“...dynamic capabilities can be disaggregated into the capacity (1) to sense and shape opportunities and threats, (2) to seize opportunities, and (3) to maintain competitiveness through enhancing, combining, protecting, and, when necessary, reconfiguring the business enterprise’s intangible and tangible assets.”*

Among the different definitions of dynamic capabilities concept that are popular nowadays, Schilke et al. (2018, p. 395-400), who studied more than three hundred academic articles related to the concept in order to understand better what definitions are the most popular among the scholars, mentioned that around one-third of all articles relate to Teece’s definition of the dynamic capabilities, which is, however, a decade older than the one presented above (see the original definition below). Schilke et al. (2018, p. 395-400) relate to the following definition as being the most popular among the scholars, although it is worth mentioning that the other popular definitions did not fail too much – the second most popular definition was the one presented by Eisenhardt et al. in 2000 and other definitions were stated in 16% of the articles or less.

The original Teece et al. (1997, p. 516) definition of dynamic capabilities is as follows:

*“We define dynamic capabilities as the firm’s ability to integrate, build, and reconfigure internal and external competences to address rapidly changing environments. Dynamic capabilities thus reflect an organization’s ability to achieve new and innovative forms of competitive advantage given path dependencies and market positions...”*

Therefore, in this study we can follow the original Teece et al. (1997) definition as the main definition and framework for describing the dynamic capabilities and how it

ties to the software development methodology a company can use. Teece et al. in their (1997) paper define that the advantages that firms can build as their competitive core are embedded in organizational processes and in the content of these processes, and the opportunities they bring for developing competitive advantages are always shaped by the assets the firm possesses, as well as the evolutionary path the firm decides to follow on their journey. As mentioned by Teece et al. (1997), these 3 elements – the firms organizational processes, the assets it possesses, and the evolutionary path it has chosen, define the essence of the firm’s dynamic capabilities and its competitive advantage. The key points that Teece et al. (1997) bring to explain why these are the main elements are that the properties or internal organizations, such as organizational processes, cannot be as easily replicated by other competitors as, for example, entrepreneurial activity, and cannot just lead to setting up unique organizational skills and simply combining different pieces of organizational structures overnight. To replicate, Teece et al. (1997) continue, time is needed and moreover, simply replicating the best practices may not be as easy and straightforward. Teece et al. (1997) then mention that firm capabilities aren’t just assets on the balance sheets, but rather organizational structures and managerial processes that have taken time to set up and embed in the company’s identity. Managerial and organizational processes in this case are referred as the way things are done in the firm, its routines, patterns, best practices (Teece et al., 1997).

Teece et al. (1997) then list the following dynamic capabilities that they consider the core competitive advantages in this perspective of the theory (see Table 3 below). The way they could be used in this research is by linking a certain dynamic capability description that Teece et al. (1997) give to what software development methodology stands for as described in the previous sections.

Table 3. Essential Dynamic Capabilities.

Dynamic capability area	Dynamic capability	Definition
Processes	Coordination / Integration	Organizational and managerial structures and processes that help to integrate different processes and procedures and coordinate the overall work of the company.
	Learning	A process which allows for repetition and experimentation through which the processes can perform better and quicker.
	Reconfiguration and transformation	Ability to sense a need to reconfigure the firm’s asset structure and to accomplish the necessary internal and external transformation.
Positions	Technological assets	Knowing how to produce the main product and the technologies needed for it.
	Complementary assets	Complementary assets needed to ensure the successful implementation and usage of technological assets such as additional

		products that may be needed to sell together with the main products (e.g., floppy disk sales increasing with computer sales).
	Financial assets	Cash position and ability to leverage the financial position.
	Reputational assets	Such as wide information available about the company or famous reputation because of certain factors.
	Structural assets	Hierarchy and formal & informal company structure.
	Institutional assets	Complexity and quality of relationships with government institutions, laws, cultural specifics.
	Market (structure) assets	Product market positions and its potential to shift and change and/or keep the competitive advantage.
	Organizational boundaries	Internal lack of integration and coordination may expose to market vulnerability (e.g., lack of intellectual property protection leading to lack of trusting to the company).
Paths	Path dependencies	The resources and complexity needed to switch from one evolutionary path to another.
	Technological opportunities	The availability of resources needed to explore and different options in choosing different evolutionary paths beyond.
	Assessment	Ability to correctly assess and transform the assets to choose a different evolutionary path.

Source: Adapted from Teece et al., 1997, p. 518-524.

The dynamic capabilities that are highlighted in bold above are those that, according to their definitions, can be rather strongly affected by the software methodology that a company chooses to have. It is important to highlight that this notion is studied and explored in this research only in relation to companies whose main product is software. To further explain the potential effects and relationship a software development methodology can have on each dynamic capability area, we dive into each of these areas below.

Using previously stated descriptions of software development methodologies, it is possible to state that all process-related dynamic capabilities (Coordination and Integration, Learning, Reconfiguration and transformation) are affected a lot, especially from the organizational point of view. Software development, especially in the competitive markets, is all about being able to hear the market and be able to provide the solution that suits the users' needs best. Based on the previous discussions in this research, it may be argued that each stage of software development process is highly important to be able to react to the market's needs and users' desires. In software

companies, software development is responsible to a big extent for the company's success since the software is the main product which sales teams sell, marketing teams market, and customer facing teams help customers to use and succeed with. In such companies, being able to adapt to the changing market requirements, to design and develop the product solutions that are needed, and to test and ensure the quality level that helps the success of the product usage – become extremely important abilities. Being able to integrate multiple sources of feedback, ensure the fast and efficient communication between teams and organization units, make sure to integrate different organizational units to develop software of the needed quality and characteristics – these capabilities become of utmost importance – and that is what a software development methodology dictates, either helpfully or it becomes an obstacle. Different software development methodologies allow for different levels and procedures for learning, experimentation, and repetition. Depending on the software methodology a company uses, it is possible to either easily transform and reconfigure a company's assets, and hence the company's product in the face of need, or become an obstacle, which can result in wasted resources, lack of software quality or redundant and inefficient code.

Moreover, company's dynamic capabilities in form of positions can also be easily affected by the software development methodology. Company's ability to convert users' needs and market's trends into sellable and viable product may result in a strong competitive advantage and thus increase its capabilities to survive and grow via utilizing its Technological assets well. Moreover, interestingly, Reputational assets may also be especially relevant in software companies: having abundance of credible information and insights into how a company develops the product, what actions it takes to ensure quality, efficiency and usability of its product may affect its reputation and ensure its unique competitive advantage. Structural assets are another capability that may be strongly affected by the software development methodology. Company's structure may well be dictated by the software methodology and thus affect the ways managerial decisions and organizational processes are handled in an organization – being dependent on many levels of managerial approvals before something can be changed into a product versus being able to quickly test and fail, may yet again affect company's dynamic capabilities. Market (structure) assets through product positioning is also highly affected by the software development methodology. If the software development team is clear and focused on its competitive advantage versus if it's clueless and unclear about what they should be focusing on may very well in the long run result in additional market advantage, or instead the lack of it. A clear and proven way of handling projects and, say, intellectual property and technological sophistication versus the lack of knowledge on what competitive advantage company's technical implementation brings (e.g., how sophisticated an AI algorithm inside the product is) may also result in a missed or acquired competitive advantage.

Finally, being able to easily experiment, research and learn through software development may very well help a company to facilitate better evolutionary path independency, or instead help to uncover new paths. For example, highly documentation- and structure-tied software development team may fail to assess its current path's state, learn about the Technological opportunities available and understand its Path dependencies. Instead, a fast and agile team may always be empowered and facilitated to pursue different potential paths, be lean about new opportunities, and clearly assess a situation in case of need.

Therefore, it should now be rather clear that software development methodologies affect a big chunk of company's dynamic capabilities or in other words company's short- and long-term success. While describing dynamic capabilities, multiple times it was noticed that being able to react and listen to the feedback, the market's needs and situation is very important to ensure the company's competitive advantage and survival. As also mentioned earlier, this ability differs from company to company depending on its software development methodology as different methodologies dictate having different procedures and processes to react to the changing market environment.

As this research is especially aimed to study how company's dynamic capabilities can be affected by the software development methodology's ability to quickly grasp, analyze, and implement changes based on the changing market environments, it would now be rather interesting to research about the changing market environments events and how they can affect companies. Moreover, it would also be interesting to describe one such market environment changing event that affected the case company studied in this research – the COVID-19 pandemic.

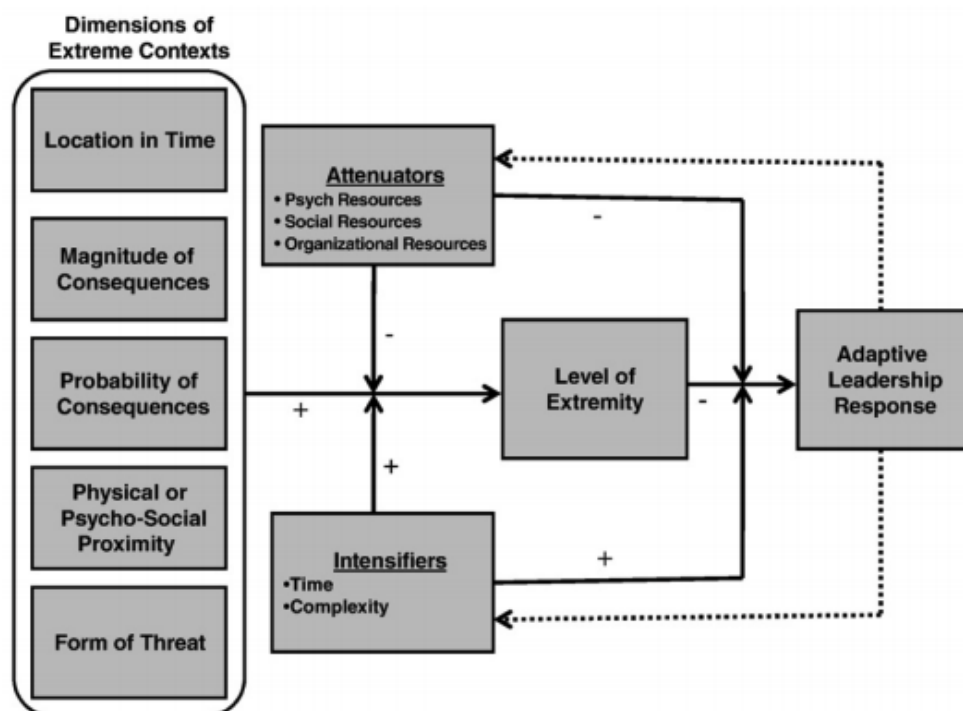
### 2.3 Extreme Contexts

Hannah et al. (2009, p. 898), define extreme events and extreme contexts as follows:

*“... we define an **extreme event** as a discrete episode or occurrence that may result in an extensive and intolerable magnitude of physical, psychological, or material consequences to – or in close physical or psycho-social proximity to – organization members... we define an **extreme context** as an environment where one or more extreme events are occurring or are likely to occur that may exceed the organization's capacity to prevent and result in an extensive and intolerable magnitude of physical, psychological, or material consequences to – or in close physical or psycho-social proximity to – organization members”.*

This definition gives some understanding of the nature and scope of extreme events and extreme contexts and what particular implications that may have on the companies and company members. Hannah et al. (2009, p. 899) then go into defining an actual model for a typology of extreme events, presented in the Figure 2 below and discussed further.

Figure 2. Typology of extreme events.



Source: Adapted from Hannah et al. 2009, p. 899.

First of all, Hannah et al. (2009) mention that the very important dimension for extreme contexts is Location in Time. Extreme contexts affect companies and leadership in different periods of time and leadership needs to be prepared, needs to be able to act during the time to mitigate the extreme context effects, and they have to be prepared post event to be able to transition from the extreme period to the post event period and stable operations (Hannah et al., 2009).

Magnitude and probability of consequences of an extreme context are also two dimensions that need to be considered when analysing extreme contexts – some extreme contexts have larger effects on multiple spheres and thus are harder to mitigate, whereas others may not be explicit and thus be easier to mitigate (Hannah et al., 2009).

Proximity of an extreme context include physical distance (such as whether the effect is direct and physical), psycho-social distance (such as having moral and psychological consequences) and psychological proximity effects on teams (some teams may be more vulnerable to extreme be harmed by contexts by the lack of leadership while others may in turn get more bonded together and thus improve the interactions and the strength of the team) (Hannah et al., 2009).

Form of threat is another aspect important to consider when studying extreme contexts (Hannah et al., 2009). Different threats may have different consequences and different implications of the extreme contexts, thus bringing different leadership qualities as a need (Hannah et al., 2009).

Hannah et al. (2009) also mention that different types of leadership may be needed for the organizations depending on the qualities of the extreme contexts studied – such as different forms of leadership may be needed before, during and after an extreme event. Moreover, the processes and procedures between the teams and leaders

are equally important in both directions as well – such as how the leader is perceived in the teams – whether the top-down and bottom-up approaches are prevalent in the companies or not (Hannah et al., 2009). Finally, it is also possible that teams' motivations and performance are affected by how leadership is being performed in an organization (Hannah et al., 2009).

Hannah et al. (2009) then also bring in the topics of “attenuators” or the things that can help mitigate the effects of an extreme context on the company. Such resources can be psychological resources (e.g., employees' readiness to be resilient and creative which may be caused by the culture inside the team and relationships therein); social resources (e.g., causing social group leadership effect once the problems occur); organizational resources (e.g., adaptability, financial, technical, human resources) (Hannah et al., 2009).

There are also “intensifiers” that may intensify the effects of extreme contexts on the company (Hannah et al., 2009). Such “intensifiers” include time (time being the key in case of extreme event – e.g., being able to react fast, resist for a long period of time or being able to persevere multiple times); level of complexity (when environments are highly interconnected and interdependent and the events collapse in an unexpected and unpredictable ways) (Hannah et al., 2009).

Having different levels of extremity (e.g., the extent and rareness of an event's effects), the leadership must adapt certain ways they can try to deal with the events (Hannah et al., 2009). In response to this, Wenzel et al. (2020) have studied different ways that leadership can be adaptive and responsive to extreme contexts. Wenzel et al. (2020) described the following leadership strategies to respond to extreme events: retrenchment (narrowing scope of activities by reduction of costs, assets, expenses, overhead), persevering (sustaining the level of previous activities and preserving the status quo at all costs); innovating (finding a way out of the current situation by finding new ways to perform activities before or altering company's dynamic capabilities); exit (discontinuation of a firm's business activities).

Extreme contexts and extreme events are therefore complex and multidimensional concepts where a lot is oftentimes at stake and different things can be set up and facilitated to mitigate the risks and consequences. Previous studies (e.g., Hällgren et al., 2018; Hannah et al., 2009; and Wenzel et al., 2020) mention that a lot of things in these environments depend on leadership and how they prepare, react, and work post-event in these cases. For software companies whose sole product is software, a lot depends on the software development teams in these cases. As was discussed before, software development methodologies dictate the ways the management of the product is handled – it dictates how the software development teams should process the decisions, requirements, changing environments and alter their development and their procedures based on that. In Agile teams, a lot of procedures and processes are focused on being able to learn and react fast, while in Waterfall teams the focus is rather on documentation and following the pre-made decisions. Companies that have more Agile methodologies in place seem to be able to have a better foundation to adapt and quickly mitigate the different effects during the extreme environments and have a bottom-up approach when altering the direction. In turn, Waterfall companies seem to require a bit more decision making and leadership in place and a more top-down decision making.

This is something that would now be rather interesting to learn and describe in detail from the case study company that has been affected by one such extreme context – the COVID-19 pandemic. The detailed descriptions of how the study company



handled and reacted to this extreme event will be presented and discussed later – in the results and discussions parts.

### **2.3.1 COVID-19 as extreme event and context**

COVID-19 pandemic is undoubtedly an extreme event and because of its long-term span it can also be considered an extreme context.

According to World Health Organization (2021), “Coronavirus (COVID-19) is an infectious disease caused by newly discovered coronavirus”. According to the Global Economic Prospects report done by World Bank (2020), the infection “has spread with astonishing speed to every part of the world and infected millions...with hundreds of thousands of deaths and many more suffering from diminished prospects and disrupted livelihoods”. As stated further in this report, various virus mitigation measures have been imposed in many countries (such as lockdowns, closure of schools and non-essential business, travel, and public gatherings restrictions), yet these measures have strongly affected consumption, investment, labour supply and production in many countries (World Bank, 2020). Consequently, major economies output projection has decreased dramatically during the first 3 months of the virus spread – from 8.4 points to 1.6; moreover, the pandemic will leave big and “long-lasting scars” on the economies of the world (World Bank, 2020).

Many industries and companies have been affected in this way or another – from having to turn to additional funding, online and remote work culture, layoffs, and closure – to flourishing in the new environments and grasping new opportunities. The detailed descriptions of how the case company, Brella, has been affected by this extreme context are provided in the case study description further. The study company has been affected by the rather large magnitude of this event in its own way. It had different physical and psycho-social proximity effects, with effects in different forms of threats and different consequences; it had different attenuators and intensifiers leading to different levels of extremity and different leadership responses. These will be also be discussed more in detail in the following sections.

## **2.4 Summary of the theoretical framework**

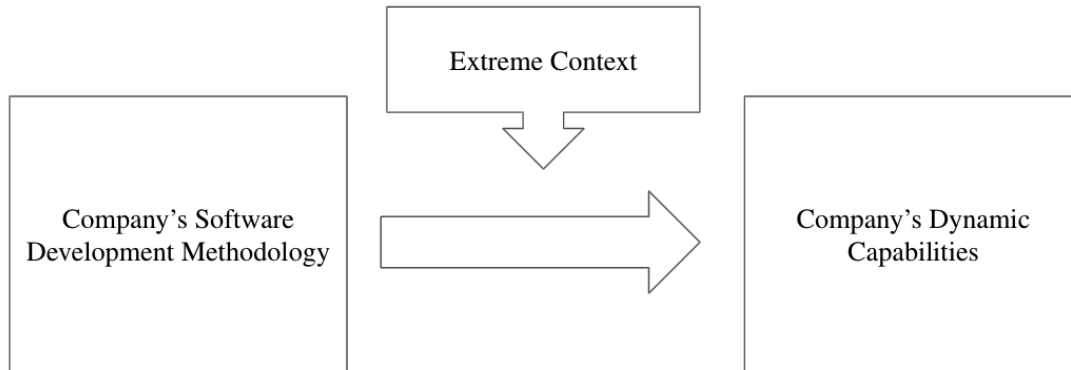
The theoretical research has shown different aspects of software development methodologies, what characteristics they bring into the development process and how they differ. Moreover, the research has shown and described the two main types of methodologies that this research will study in the empirical part – the Agile and Waterfall methodologies.

The theoretical research has also analyzed the topic of dynamic capabilities and how they can be affected by the software development methodology that a software company chooses to use.

The theoretical research has also studied the extreme contexts – its dimensions, characteristics and introduced the COVID-19 case as an extreme context into the research. The research has also suggested the ways software development methodologies affect dynamic capabilities of companies under extreme contexts.

Therefore, based on all this previous research done, the following theoretical framework can be created (see Figure 3 below).

Figure 3. The theoretical research framework.



The overall notion that the theoretical research was pursuing was to show that the company's software development methodology affects company's dynamic capabilities in certain ways and that these effects are different under the extreme context circumstances.

### 3 DATA AND METHODOLOGY

#### 3.1 Methodological framework of the study

The methodology used in this study is based on the pragmatism research philosophy because, firstly, external view is chosen to answer the research questions (Saunders et al., 2009, p. 108-148). Secondly, focus of the research is on practical applied study with different perspectives to help interpret the data (Saunders et al., 2009, p. 108-148). Thirdly, this research adopts both subjective and objective points of view (Saunders et al., 2009, p. 108-148). Finally, it uses mixed research methods (Saunders et al., 2009, p. 108-148). The research uses deductive approach because there are research questions derived in the beginning of this research, which are researched in the literature review and tested in the real-life case scenario in the case study (Saunders et al., 2009, p. 108-148). Single case study method is used because it seems to be the most convenient and efficient way to prove the theory in practice and because of the case study company's availability for the interviews with the researcher. The research will use case study method in a cross-sectional time horizon with the mix of archival data analysis; therefore, it will mainly be a mixed-method study (Saunders et al., 2009, p. 108-148).

The case studies are approached with a use of semi-structured interviews and analysis of the available archival data about how the case company performed during the case study period. Interviews are conducted using video conference software; the archival data is derived from the documents and information that the case study company can give and available publicly. The data is analyzed using content analysis where content of interviews, archival data, online sources, industry reports were analyzed using key terms in the research questions such as software development methodology, dynamic capabilities, and COVID-19.

These methods are the most convenient to use in this research. Moreover, they seem to be the most valid, as they answer the research questions well. They are also the easiest ones to do for this relatively short, light, and inexpensive research. The qualitative research design was chosen as it allows to study the research questions in more details by collecting and analyzing the data related to opinions, decisions, and behaviors of the interviewees instead of relying on numbers.

The researcher will be making sure to not show any sensitive data in the research and the interviewees and the case study company are informed on the contents and results of the study to ensure that the research is ethically correct.

Brella was chosen as a case study company because of the researcher's access to the company's employees and information about how the company has changed during the last year. The researcher, as for the moment of writing this research is a full-time employee in the company, therefore it is also in his own professional interest to study Brella as a main case study. At the same time, the researcher may be biased in certain opinions as he is an actor that was involved in the case study per se. Thus, the interviewees may be biased when telling the story to the researcher as they may have personal attitudes towards the researcher that occurred during the working experience together. Nevertheless, it should still be possible to preserve the objective nature of the research as the researcher would be trying to take the outsider perspective. Moreover,

some of the interviews were done with newly joined team members, thus, they may have a better outsider perspective. In the end, the coding results will be presented, so the readers can see for themselves the analysis process.

This research is also supposed to have objective and unbiased study process, where the results and conclusions generated are not specifically chosen to highlight some presumed point of view. The results and process of analyzing, i.e., results coding, will be presented in the study results section. The researcher does not have any presumptions or hidden aims other than declared here.

Validity and reliability of this paper shall be ensured by the wide range of sources used.

This study has several limitations, specifically at the case study stage. There is only one case study company being studied therefore the characteristics may not be representative enough to say that the findings of this study can be used universally in every company in every industry. The nature of the research was in software methodologies; therefore, it may only best apply to software companies. Moreover, because it's a case study of the company that has been affected by the one extreme context only – the COVID-19 pandemic – it may also not be universally applicable for the other extreme contexts. Finally, because the software methodology topic is rather illusive and so many companies use the methodologies so differently, the findings may again not be applicable that easily. Yet, despite all these limitations the study can still give a rather interesting perspective into how Agile and Waterfall software methodologies can affect the companies' dynamic capabilities in extreme contexts.

### **3.2 Brella case study company description**

Brella is a software company founded in 2016 in the city of Jyväskylä, Finland (Finder.fi, 2021). Brella operates in the industry of information technology consulting and information technology services, with 1-2 million EUR turnover and 20-49 people working in the company (Finder.fi, 2021). Brella's board members now include Markus Mikael Kauppinen (President and CEO, Member of the Board), Sakari Pihlava (Chairman of the Board), Janne Alekski Puustinen (Member of the Board), Hans-Peter Siefen, (Member of the Board), James Charles Wiandt (Member of the Board), Mikko Johannes Matikka (Deputy Member of the Board) (Finder.fi, 2021).

The event industry that the company is working in was valued to be over \$1,100 billion in 2019 (Events Industry by Type Report, 2021). As learned in the interviews, the Brella's focus inside this industry is on the business-to-business professional conferences, corporate conferences, and business tradeshows. However, the COVID-19 has had a major impact on the industry because of the traveling restrictions and conference closures (Events Industry by Type Report, 2021). According to Bizzabo's Evolution of Events Report (2020, 5), 75% of event and marketing professionals have pivoted to virtual events, and about 40-50% had to cancel or postpone their events. And that is where Brella managed to provide help by pivoting to a virtual event offering. As mentioned by Costa in her article (2020), "virtual events startups like Brella have been on the rise as they are uniquely positioned to capitalise on the digital transformation unfolding within the sector and to work with traditional events companies on adapting as the trend continues". Brella managed to increase requests for demos in 2020 by more

than 300% and helped to facilitate more than a million meetings in 2020 (Williams & Hill, 2020).

The information below is provided based on the company's official website. According to it, the company is "the world's leading virtual event platform". The company provides event (i.e., large conferences organised for attendees that they pay to get access to) organisers with the platform to host the events on and power their communities on.

Brella brings multiple values which can be considered as their competitive advantages, which form the core of their offerings. These offerings can be divided by the different types of values that Brella brings, from the perspectives of three main types of stakeholders that Brella serves: value for the event attendees, value for the event organizers, and value for the event partners/sponsors. However, it is also possible to say that if one value works for one type of stakeholders, then it also works for all other types. For example, the possibility to chat before an end user's meeting in the Brella software is mainly value for the attendee stakeholder, though surely it also indirectly brings value to the event organiser since the event becomes more efficient for attendees, thus the event itself becomes more competitive and unique. Below are the direct values that Brella brings to a specific type of its stakeholders, not the indirect ones.

Brella's website says "sometimes, finding the right connection is like finding a needle in a haystack". Brella matches the right people, it helps them make right connections at events, it allows them to save their time, and focus their efforts. In the end, Brella is also helping businesses to grow by helping them to make more valuable connections. Bigger and more efficient businesses usually also result in growth of either current industries or growth of emerging ones. Eventually this ends up in better economic growth for a country and the world. This might be considered as an overall vision of how Brella helps the world become a better place.

Basically, the values that Brella brings reflect the main features that the Brella product has. They are listed below with additional comments and descriptions.

### *Values for attendees*

Make right connections easily. The Brella tool helps to find a right connection for any type of business event attendee – from a young startup representative to a mature senior manager. People go to business events to potentially meet other people that may be of their interest. For example, a startup is looking for investors and an investor is looking for a startup to invest in. According to Brella's materials, often attendees make mistakes about what kind of people they think they want to meet and what kind of people they need to meet. So that's where the Brella's intelligent matchmaking algorithm comes in place.

Brella's artificial intelligence-based matchmaking algorithm helps meeting the right people. Helping attendees to meet the people that, from the first glance, may not seem like right connections – is also where a big direct value of Brella for attendees lies. So, they now can rely more on the software, have less stress, and have more energy for the actual communication, which is, in turn, a very important process supported by the Brella product.

Brella's product makes meetings more structured with a help of predefined meeting areas and time slots. This also helps to make meetings more serious, focused, and efficient. Chats bring better understanding and a mutual relationship in advance.

Individual schedules allow people to see and organize their schedule during the event, which also gives more freedom and energy. Thus, the whole networking process and experience becomes very simple and easy-going.

This simplicity is also ensured by easy to navigate and understandable software available on all major platforms.

Brella now also provides virtual calls and provides a virtual venue for watching live streams. It also allows attendees to visit virtual booths, which are places where sponsors can present themselves and display materials.

### *Values for organizers*

For organizers, Brella solves the problem of networking by allowing attendees to match and schedule meetings together with each other. According to “The Event App Bible 2021” (2021), many conference/event-organisers now do care about having some kind of a software for their attendees to get around or book meetings easier. Although a well working software is not an easy nor a cheap thing to do. Therefore, solving all these problems related to networking – i.e., software development, meeting slots creation, other networking opportunities provision (like marketing of networking importance or constant support) – by just laying it all down on some specialized company – that could bring a big direct value for the event organizers.

At the same time, one special value of Brella, could be that it allows business event organisers to demand customisation. Organisers often feel like it’s better to build their own software in their own branding style, and they think that it will be the reason for not working with such companies like Brella. With the branding possibility this is not a reason anymore. The Brella software is now also flexible and changeable – thus, some event-specific features may be added where some event-specific actions can be done in it, and organizers value that.

Another value is the possibility to have a full statistical coverage and dashboards with full data on how your event is doing, what your attendees are doing, and how they feel. Complete understanding of what your attendees did, with whom they met, what kind of aspirations they had while attending your event is a very valuable aspect of using Brella products.

Apart from software, there is also a great value in terms of offline help. In particular, the possibility of constant support by Customer Success Managers and Customer Support teams of Brella is a great value. This creates a situation, when there is not just general communication between the client and Brella, but the whole collaboration for the shared goal between them – it allows Brella to stand out from just a list of event software companies and become an event organising “partner”.

### *Values for event partners/sponsors*

Allowing sponsors to make their communication with the attendees in advance, facilitating different marketing opportunities through the software and predefined meeting process – that all creates a great monetisation opportunity for all involved in the event. It really might be a great advertisement opportunity for any sponsor/partner, who may be able to directly advertise to its target group. Moreover, with the release of the virtual features, the sponsors can now create their own “virtual sponsor booths” which

allows them to replicate the normal in-person conference concept of booths, share different marketing materials, and advertise themselves in the event using the platform.

Appendix 1 includes the list of all features that Brella currently has with additional comments about each.

According to the internal information derived from the interviews, the Brella company has experienced a significant turmoil during the last year. When the COVID-19 pandemic hit, the traditional business structure of Brella's offerings – the in-person event application – did not work anymore as all travelling and all public gatherings had been banned. In March and April 2020, the Brella company had been undergoing tough times when at some point there were no events going on – meaning no business for Brella. During these months, Brella had to impose layoffs and undergo a significant product strategy pivot. Moreover, as learned in the interviews, the whole industry was undergoing tough times as well. The event organisers did not know what to do – all of Brella's clients were handling the events in-person and they had to somehow come up with a new way of conducting their events. Therefore, both Brella and the industry was faced with a situation where they had to come up with a new way of doing business. That's how the choice went to being a virtual event offering company. The company then had to come up with ways to become one – by building features that would allow for business events to take place online in Brella. The company managed to successfully deploy these virtual features, and according to the interview results, triple its revenue during the year. This turned out to be the most successful year in Brella's history, and, as of the time of writing this research, it keeps on growing.

And that is where the interesting part comes in. Why was Brella so successful? Did the software development methodology of Brella facilitate such a successful pivot? Did the methodology affect any of the dynamic capabilities studied in the previous sections? And did these effects change during the year of being under such an extreme context as COVID-19 pandemic?

### **3.2.1 Interviews' description and questions**

The interviews were handled using the video-conferencing software Zoom. The interview questions were forwarded to the interviewees prior to the interview so that they could prepare and know what the discussion will be about. So, all interviews followed the same structure, yet sometimes the discussions were going into related subjects depending on where the conversation would bring the interviewer and the interviewee. The video-conference calls have been recorded by the interviewer.

Since the research question and the nature of the research lie deeply in the software development and studies the effects software development has on dynamic capabilities, the interviews were only conducted with the Brella's Product Development team. The customer facing teams in Brella could have also provided an interesting insight from their perspective, yet their insight wouldn't include anything regarding the software development in Brella, which is the main objective of the research. Besides, during the interviews, the researcher had an opportunity to talk to the people in the Product Development team, who because of the nature of their work, also must deal a lot with the business side of things – in particular, the Product Designers and the Chief Product Officer who I interviewed, – therefore, to some extent the business side is also represented in the study.

There were two rounds of interviews. The first round included interviews with people that have been in Brella before the COVID-19 pandemic and during it. The first round of interviews included interviews with Brella's Chief Product Officer, Brella's Frontend Lead, Brella's Lead Designer, and Brella's Backend Lead. The second round of interviews included only the people that joined the Product Development team during the pandemic: Brella's Product Designer, Quality Assurance Specialist, Chief Technical Officer, and Senior Frontend Lead. This way, the researcher could hear the perspectives of both people that have been employed at Brella pre-extreme context and during the extreme context, to better compare what changed and how they perceived the changes, while also considering the people that began employment later and who could only see the aftereffects of those changes. Below is the table summarising all the interviews made.

Table 4. Summary of interviews.

Interview Round	Interview Nr	Interviewee's Title	Interview Length
Round 1	Interview 1	Chief Product Officer (CPO)	1 hour 18 minutes
	Interview 2	Frontend (FE) Lead	1 hour
	Interview 3	Design Lead	1 hour 6 minutes
	Interview 4	Backend (BE) Lead	1 hour 18 minutes
Round 2	Interview 5	Product Designer	1 hour 26 minutes
	Interview 6	Quality Assurance Specialist (QA)	29 minutes
	Interview 7	Chief Technical Officer (CTO)	41 minutes
	Interview 8	Senior (Sr) Frontend (FE) Developer	53 minutes

Appendix 2 includes the list of the questions that were used in the interviews. Because all the interviewees were working in the software development fields or related to it, most of the terminology was clear. However, in case of a need, the necessary clarifications were made by the interviewer.

Based on the answers, the analysis was made, and the coding work has been done using the knowledge perceived from each of the interviewees. The case study results with the coding and analysis, as well as discussions on the case study results are presented in the next chapters.



## **4 STUDY RESULTS**

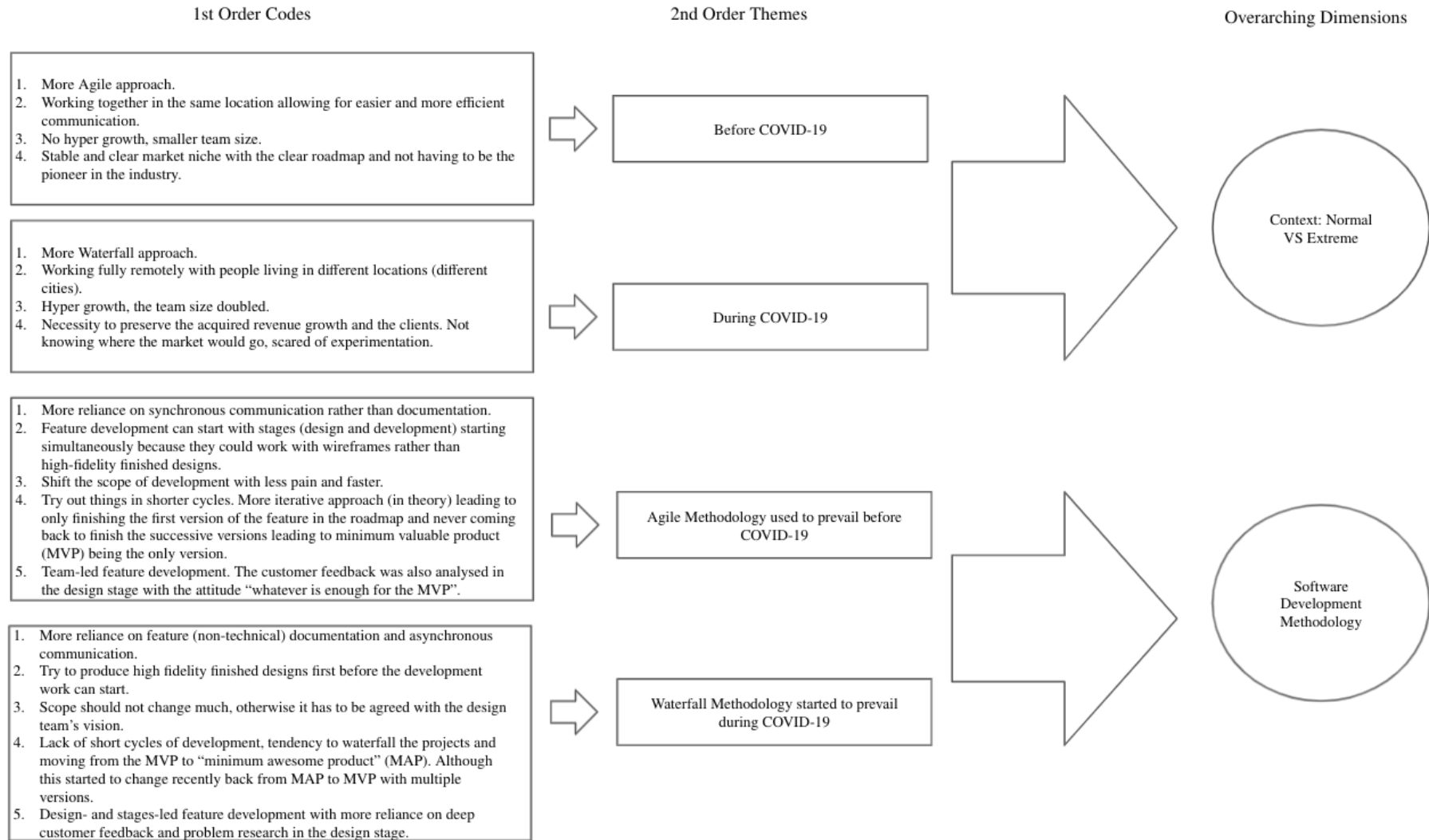
The study results were analysed using the coding technique. The codes within the data structure are presented below. Moreover, part of the interviewees' answers is cited in each question and are presented below.

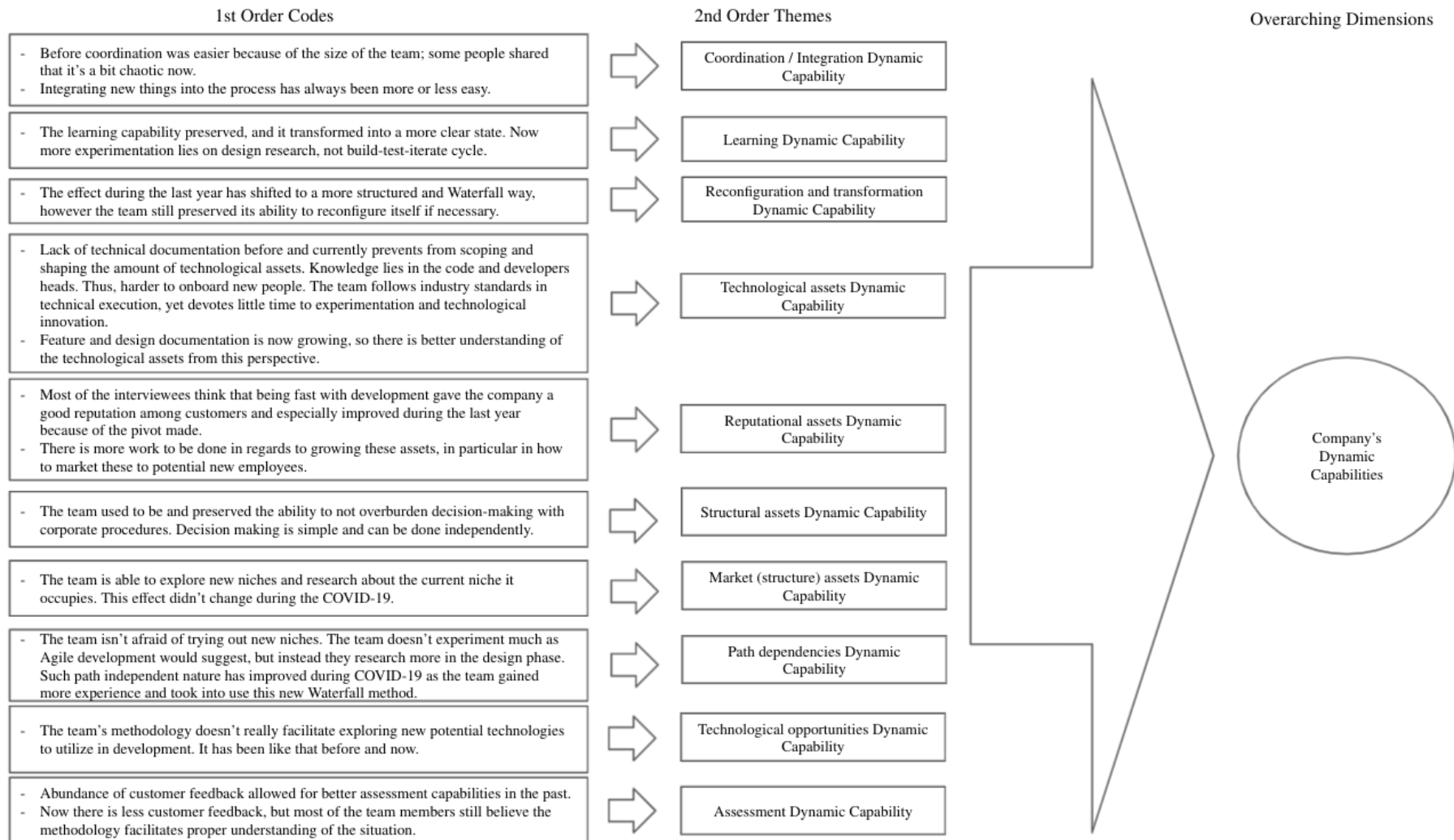
After presenting the study results, they will be discussed in the discussions section.

### **4.1 Study Results**

First, the coding data structure will be presented with 1<sup>st</sup> Order Codes, 2<sup>nd</sup> Order Themes and Overarching Dimensions in Figure 4, and then each 2<sup>nd</sup> Order Theme will be discussed more in detail with general paraphrasing of interviewees' answers which will also be supported by select citations deemed supportive from interviewees' answers.

Figure 4. The Data Structure.





#### 4.1.1 Software development methodology in Brella and how it evolved

The four people that I interviewed who could tell about the methodology of how the Brella product has been developed in the past, namely the CPO, the Design Lead, the FE Lead, and the BE Lead – all mentioned that the methodology was more leaned towards Agile. They used to rely more on synchronous communication and less on documentation, they used to have more software projects (i.e., feature development projects) done in simultaneously ongoing stages (i.e., design and development) as they used to rely less on high-fidelity finished designs but rather work on with wireframes. They used to be able to shift the scope of the project faster and with less pain, and they used to also try out things in faster cycles.

In the past, the team used to be smaller and most of the team members used to be in the same office, therefore for most of the projects, synchronous in-person communication used to be a norm. As mentioned in the theoretical review section, Cockburn (2002) was pointing out that for the Agile product development to succeed, the small teams need to ideally be in the same location. The team never really paid much attention on the documentation – neither feature related documentation (i.e., how certain features work, and what are the planned updates for these in the next iterations), nor the technical documentation (i.e., how the code works and what technical challenges are associated with it). According to the interviewees, it was not because the team was reluctant or following some specific goal, but rather there was never really time dedicated to documenting and saving things for later.

More projects used to go in simultaneous stages, where the development and architecture building would be possible to start with just certain wireframes available rather than the high-fidelity designs and fully thought through design.

Changing the scope also used to be simpler because of the ease of synchronous communication, fast development work, lack of documentation that required to be changed, and less people involved in the projects.

At the same time, the team never normally used to follow the full incremental cycles Agile way, where they would try to develop the projects in sprints and then get customer feedback and then repeat. The team used to focus on development of the features and on getting customer feedback to an extent where it would be enough for the MVP. The team didn't fully utilize the concept of sprints or incremental cycles – instead these cycles were spanned over for months or years rather than weeks. So, what turned out to be for most of the cases was that the team would develop the MVP version as version Nr 1 for a particular feature, but then never really come back (or come back rarely) to the features and leave the MVP version as the only version in the platform.

Because of ease and abundance of impromptu synchronous communication, less concentration on high-fidelity design and deep design research, the team used to have a more team-driven approach where all team members would feel that they are building a new feature together, rather than just following a staged procedure, where the design research comes first, comes up with a solution, gives the solution to the developers and then the developers just have to implement whatever they were told to do.

Below you can see a citation from the interview with Brella's BE Lead about how they used to do things in the past:

*“I personally think that that is what we used to do in more of an agile way when we were a much smaller team. When everybody was in the same room, it was very easy to do these micro-communications that you can ask something and there is going to be like, ‘ok’. But now that there’s bigger teams, there are people remote, I think that somehow got lost and people aren’t that active anymore on those. And I am not sure why that happened. Maybe it’s just that people couldn’t see how we used to work and then they can’t relate to it, or they don’t know how it’s done. Or maybe it is that we didn’t actually have enough documentation on how we basically develop our stuff or those features”.*  
(Interview 4)

Below you can see a citation from the interview with Brella’s FE Lead about how Agile turned out to be during the COVID-19:

*“...especially during the COVID... I felt like the Agile method is really hard for these remote teams to work with. It works when you are in the same office and you have the same information, information pool is at hand for you all the time, so you can go and ask and have these small chats. But during the remote work, I think the teams are so separated and the communication is so scheduled – maybe one meeting a week – so you are not actually having this Agile method of building software. If you are not documenting, it’s very likely that at some point somebody needs that documentation, and it should have been done in the first place... At least for the tools that we use, I am kind of curious are there tools for remote teams that we are not actually using that could solve these problems that I am seeing: missing information and lack of understanding of where we are. I think the tools we use, at least are not good enough to be as efficient as we could be...”.*  
(Interview 2)

All the four first round interviewees agreed that the company used to develop the product in a more Agile way in the past. The CPO mentioned that the team is still being rather Agile in the way it develops the product, yet the FE Lead, the BE Lead and the Design Lead mentioned that the team moved away from being that Agile towards being more like Waterfall. What is also interesting is that FE Lead, BE Lead and the Design Lead share the same reasons for why the things had to go more Waterfall way during the COVID-19, yet only the FE and the BE Leads share some strong feelings that the things should ideally move back to a more Agile way.

All eight interviewees described the same process of how they see the product being developed in Brella. The conclusion about the development methodology in Brella can be made that it is more Waterfall, with Agile elements. The interviewees mentioned that first, there is a prioritization and high-level roadmap creation stage. The managers of the company decide what needs to be built using various sources of feedback, and then they collaborate with the design team to come up with the solutions. Normally, the design team already gets some ideas how things should be solved, so by the time an OKR (Objective and Key Results – i.e., a feature or set of features that needs to be developed) is created and the team is assigned, the designers can already start some design and research work. The second stage is where the teams get assigned to each OKR and then they meet regularly to kick-off the development, check on the

design and development progress, and handoff the design to the developers. The third stage is the development and that is where things become a bit more Agile, since the developers may often end up doing incremental cycles where they would develop, then test and so on. Once the development is finished, the testing begins. Once the quality standards are passed, the feature gets released. The final stage of the software development in Brella is verifying that the feature achieved its goal and ensuring that the feature release was successful as well as having a retrospective of how the feature development went for this project.

Six out of eight people that I interviewed mentioned that the methodology of software product development in Brella was ultimately leaning towards Waterfall during the COVID-19 time. The CPO said that the methodology of developing the product continued to be more of an Agile fashion. Then, the Product Designer mentioned that we cannot really say whether Brella's methodology leans more towards Waterfall or Agile as he believes there is no other way, in general, of developing the product other than with Waterfall. The six people that said the methodology distinctly leans towards Waterfall than Agile were the FE Lead, the BE Lead, the Design Lead, the CTO, the QA Specialist, and the Sr FE Lead. What is also interesting to point out is that this opinion was shared not only by the people that were employed before the COVID-19 and thus could easier identify the changes, but also the newer employees – this additionally proves that it did indeed start to move more towards Waterfall. It must be mentioned though that pretty much all eight people also pointed out that it very much differs from project to project – for some projects, the Agile way can prevail, for others the Waterfall way. However, the six people who said that the methodology started to be more Waterfall were pointing out that most of the projects are done using Waterfall. That is also why the whole methodology can be considered as more Waterfall. Another important aspect to mention was that the FE Lead and the CTO shared the same opinion that the team is still more Agile in the development stage, but the overall process and the design stage are Waterfall.

With the COVID-19 outbreak, not only did the teams have to start working completely remotely (because of the group gathering restrictions imposed by the governments), leading to less impromptu synchronous communication, but also the team itself grew nearly twice as much – from eight people to nearly twenty people. Therefore, during the last year the team had to come up with a more standardised way of communication through scheduled videoconferencing for synchronous communication through tools like Zoom and rely more on asynchronous communication through tools like Slack, GitHub, or Clubhouse. Moreover, because of the remote work and growing design team size, the design team had to start moving more towards documenting the feature descriptions and proper high-fidelity designs before the development work could start. This was additionally forced by the business requirements as well. The company itself during the last year grew nearly twice as much, therefore new teams and new processes were created – thus, the old processes of sharing the product information (such as roadmaps, feature descriptions, and vision description) did not work anymore. The company required more proper documentation and information sharing to the business teams.

The team started to lean more towards trying to produce high-fidelity finished designs with design taking a very big chunk of the feature research, and not involving the developers in it. Only after the properly executed design would developers start the execution where it is mere execution – just implement what the designers told you to.

In most of the projects, scope creeping and major scope changes were not welcomed. Such changes would have to be discussed separately with the design team and then potentially booked for future iterations. However, confirming the scope changes would not normally be too hard of a task – just having a meeting or chat discussion between the project team members would normally be enough.

Because of the team's tendency in the past to never really come back to the first MVP versions created for a feature, the team would instead lean towards MAP (referred to as “minimum awesome product” by the interviewees). This means increasing the project cycle and time needed for each stage – design, development, testing. However, pretty much every interviewee mentioned that this has also started to change back to smaller road-mapped versions of a feature for some projects.

Moreover, all the changes above are reinforced by the reason that the company started to be the industry leader and the industry pioneer during the last year. Also, the competition intensified as the competitors were getting investments and trying to get a hold on the same market and the same customer profiles. This meant that there was more at stake and experimenting with features by leaning more to incremental cycles could not really be a choice, as believed by the Lead Designer for example. Because of that reason, the tendency to spend more time in the design research, UX (user experience) and UI (user interface) design started to prevail.

Below you can see a citation from the interview with Brella's BE Lead about the team's tendency to waterfall:

*“Why we are good at waterfaling and why we aren't Agile is that you can identify these blocks of things that keep happening: after something is done, then something happens. And usually that means that when something is done by somebody, somebody gets released out of the OKR. So that means that the whole team isn't actually developing a certain feature, it means that one person is developing or doing something for the feature at some point of the whole cycle. So, in my opinion that's kind of Waterfall-ish... We are really good in waterfaling, but we aren't really good in Agile. Or I wouldn't say we aren't good at Agile, but certain things are done in a waterfaling way”.*  
(Interview 4)

Interesting finding from the interviews could also be that the design team does not really believe in this traditional Agile way, whereas the developers, the CTO and the QA do believe in a more Agile way and do think that the team should move back to it.

Below you can see a citation from the interview with Brella's Product Designer about how to define the methodology in Brella:

*“It's always a mix. There is no way to invent some methodology that won't have some features or characteristics that are in the Waterfall”.*  
(Interview 5)

Below you can see a citation from the interview with Brella's Design Lead about how to define the methodology in Brella:

*“I think it will depend on project to project. It's completely like that. There are certain projects where we need to have these step-by-step things and in certain*

*projects, we are really Agile and really working together, breaking down things in versions, MVPs... In a waterfall model... you always design everything from end-to-end. There are pros and cons to that. Manually designing things from end-to-end, you have a better clear vision, you have a better control on the finesse of the product, but then the project tends to be longer. Then you either need more manpower to shorten the project time or you need more time to finish the project... In terms of Agile... Particularly, I am not a very big fan of MVP mentality if we don't work on the second versions of it... Since we are a startup, a lot of times what happens is we decide to work on something, we work on the first version of it because we are agile, we want to release it as fast as possible, we want to break it down so that we don't work on that project forever. So, we will work on the version 1 and then because business priorities change or because something changes... we will never come back to the version 1.1. And as a designer, that frustrates me quite a lot because you have a clear vision, you have a mentality of giving the best user experience, giving the best product experience. And I think that suffers a little bit if we don't work on the next versions of it".*  
(Interview 3)

Below you can see a citation from the interview with Brella's CTO about how to define the methodology in Brella:

*"Well, I definitely would say that we are not a Waterfall company because with traditional Waterfall software companies it's usually 1 or 2 releases a year. So, in that sense we are really Agile, we can deploy immediately when the feature is ready, which is super cool, it's not so common in many SaaS companies even today. So, in that sense we are Agile and really Agile I would say. But getting started with doing small improvements, getting customers' feedback, doing smaller iterations – that's some area we need to improve. So definitely more on the Agile side compared to the Waterfall. Maybe, starting the new feature – that's a bit heavy – there it's a bit Waterfall problem we can see, but once we enter the actual development, then we are Agile, the way we push to production, it's Agile. But the whole process could be more Agile and could be much faster, having smaller releases more often, getting customer feedback – that's something that Brella should focus on in the coming months, I would say".*  
(Interview 7)

#### **4.1.2 The effect of the software development methodology in Brella on the dynamic capabilities before and during COVID-19.**

##### **1. The effect the software development methodology in Brella had on the Coordination / Integration dynamic capability before and during COVID-19.**

Before COVID-19, the Product Development team size used to be smaller. Majority of the team used to be in just one office. Such a small team used to also to do more software development projects in the Agile fashion. For example, a small team would be developing the feature in smaller development cycles and try things out faster



to get feedback faster. Having a smaller team was easier to coordinate the whole development. Even though it was in a less structured and more communication-oriented way, it felt more coordinated to the interviewees.

The integration used to also not be a big problem, where the methodology that was used would be adaptive and perceptive for new processes, workflows, and tools. Moreover, it can be said that the whole development process was more integrated where the business teams would get faster feedback and knowledge with faster cycles.

During COVID-19, the methodology shifted to being more Waterfall and more structured. At the same time though, more structured ways appeared in only certain parts of the process such as design research and feature communication to the business teams. However, holistically, because of the team growth, poor documentation and difficulties around remote communication, the ability to coordinate suffered. Most of the interviewees mentioned that it feels more chaotic now, but pretty much everyone also explained that as being the outcome of team growth, poor documentation of the past work methodology and difficulty of remote work.

The integration part did not change much during COVID-19. The team is now establishing new processes for working in this extreme context, therefore, perception of new ways of working is on par. The whole development workflow also still felt rather integrated for some and less integrated for others.

Below you can see a citation from the interview with Brella's QA Specialist about this:

*"I don't think we have a very controlled and clear way. Some things are very well documented, but some other things are not... We are growing, used to be a very small team doing the whole development process – then you don't need so much because it's a small team, the information goes to everybody. And now when we are growing, we are facing issues because of that".*

*(Interview 6)*

Below you can see a citation from the interview with Brella's Sr FE Developer about this:

*"We have this specific structure that design goes first and then there still might be time to pick out the right people to work on the new OKRs or features, while the design is ongoing. And then... the problem comes when you rely more on design handoffs... and then there might be chaos later on... You have to coordinate that these people work on this feature, and they should do this kind of things, and based on these designs it should go like this, but then when we handoff the designs and when we use more waterfall-based approach things may not be as they seem in the designs... We might run into an issue that this will be done, but we found some major problems with the designs that we couldn't anticipate because we [developers] aren't included in the design phase, we can't ask questions that much, we have to ask questions when we are doing the work, which is good, and we get answers from the designers, but some of the things could be tackled early on".*

*(Interview 8)*

All in all, the effect of the methodology on coordination capabilities suffered the worse during the COVID-19 pandemic, while the effect on integration capabilities stayed mainly untouched. Both, however, transformed to some extent.

## **2. The effect the software development methodology in Brella had on the Learning dynamic capability before and during COVID-19.**

The learning capability did not really suffer any bad impacts during the COVID-19 pandemic. In fact, the interviewees that were the actors during the product pivot mentioned that they were extremely Agile when executing the pivot (the CPO, the FE Lead, and the BE Lead). They mentioned that they basically just gathered in the room, discussed the things necessary to change and started the execution. The CPO was doing extensive communication with the business teams and the clients trying to get the information from there and bring it to the table for the developers. They would then draw wireframes on whiteboards, think together on how to solve the problem and then the developers would start trying to get some MVP or proof-of-concept to see if their idea worked. The new features would be tested with an MVP mentality and shipped fast to get feedback as fast as possible. The second iterations of the same features would then follow. According to the description, it was true Agile, and the team was able to move fast.

The learning capability preserved, and it transformed into a clearer state. Now more experimentation lies on design research, and not on a build-test-iterate cycle. During the COVID-19, the approach shifted a bit to being more conservative and trying to learn much more during the research and design phase. It is controversial based on the interviews which approach is better (MVP-get feedback-iterate vs deep research and develop) and the interviewees share rather different opinions.

Below you can see a citation from Brella's Design Lead about this:

*“Before we were quite not sure exactly. It has to do a lot with designers’ involvement in the process and designers’ research. Personally, I would like the designers to finish quite a lot rather than developers building something and failing after that... I don’t think we have had that before COVID at Brella”.*  
(Interview 3)

Below you can see a citation from Brella's FE Lead about how the Agile methodology during the product pivot helped them to try out new things faster, and not being afraid to experiment:

*“I think it was natural that everybody would expect us to make a little bit more mistakes because we were trying so much new things and building that faster...”.*  
(Interview 2)

It is possible to assume that the methodology's change just transformed the way it is affecting the learning dynamic capability but did not really worsen it. According to the design team, it was improved.

### **3. The effect the software development methodology in Brella had on the Reconfiguration and transformation dynamic capability before and during COVID-19.**

As mentioned above in the description to the Learning dynamic capability, the Agile methodology allowed the team to reconfigure their resources extremely efficiently and easily and thus transform the product and the company offerings to the market. Pretty much all interviewees mentioned that both the past and the current way they develop the product allows for easy and very fast shifts of the priorities to where they can start working on something totally new, basically, in an instant. However here comes the difference between how it has been before and how it is now. In the past, the team would naturally go in an Agile way for most of the projects, but now it depends on how fast a change is needed – in majority of the projects now, the development would go into the Waterfall where the design would do deep research and experimentation. It is now up to the managers and communication to explain that a project does not need to go into the Waterfall way, but instead should go into an Agile way.

Below you can see a citation from the interview with Brella’s CTO about that:

*“If it’s a bug, we can be super-fast, we can fix it and deploy it... If it’s a small improvement, which is visible for the customer then we are too slow because then we can start to see that there are many steps: design, FE, BE – too many people are involved. Every time you bring new people to the process, it slows things down – so there we are not Agile”.*

*(Interview 7)*

Below you can see a citation from the interview with Brella’s CPO about that:

*“...theoretical answer to this question is ‘yes’, and we practice that to some extent. However, I also believe that there can be quite big biases that we have been building ourselves during all these years we have been working that sometimes it might be hard to change whether or not we should do this type of reallocation”.*

*(Interview 1)*

To summarize, it seems that the team could do the necessary transformation in case of a need and in case there is a clear understanding among the team members that this change is necessary. However, it also seems that in most of the cases right now the team would approach the reconfiguration of objectives in a rather Waterfall way unless there is a clear communication from the managers team that there is a need to solve it faster. Therefore, the effect on this dynamic capability of the methodologies during the last year has shifted to a more structured and Waterfall way. However, the team still preserved its ability to reconfigure itself if necessary.

### **4. The effect the software development methodology in Brella had on the Technological assets dynamic capability before and during COVID-19.**

In software business, the Technological assets dynamic capability can be considered from two perspectives – the way the knowledge and the know-how are

stored (i.e., documentation) and whether the current way of developing the product facilitates trying new technologies and thus growing and improving your technological assets.

Most of the team members mentioned that neither the old way, nor the current way implied doing extensive technical documentation. Of course, developers can look at the code and code tests and learn about the product from there, but as also mentioned by the newest team members, more proper and structured documentation could have helped them onboard faster and easier.

From the design point of view, the situation with documentation changed. As the team grew, became more remote, and moved towards the Waterfall way of doing things, the design and functional documentation started to improve drastically. Therefore, in this sense the technological assets are being preserved better.

What comes to trying new technologies and improving the technical part of things, the interviewees mentioned that they use industry standards and do not really try to innovate in this sense. They grow their technological assets by hiring new people and opening new roles (such as automated testing or continuous integration and delivery/deployment).

Below you can see a citation from the interview with Brella's BE Lead about that:

*"I think, we didn't document that much of stuff. I think we still don't document that much of stuff... But I think that isn't really COVID-related. Well, you could say it is as we work remotely, we need to share this knowledge to different places. But I think, it's more related to the growth of the company because everybody cannot know what happens with every line of code – that just doesn't work. You need to have this help of explaining something to the next engineer to explain 'hey, this is how this feature is meant to work these things work like this'".*

*(Interview 4)*

Below you can see a citation from the interview with Brella's Sr FE Developer about that:

*"From the technology perspective, we are following industry standards and that's okay. I don't think we need to experiment or improvise that much from the technical perspective and in the code because following industry standards is definitely the right way to go as long as we choose the right tools".*

*(Interview 8)*

In summary, it seems that the design side of the documentation has improved, which means better preserving the technological assets. The technical documentation is not really in place and has never been. The team is not really innovating in terms of technical implementation but trying to follow the industry standards and has always been. Therefore, the effect of Agile methodology on this dynamic capability can be seen in lack of documentation and thus lack of preserving of the technological assets, as they may potentially be at risk. Waterfall methodology in turn started to help the team to preserve and grow its tangible knowledge and know-how. Therefore, the Waterfall methodology and the things that happened during COVID-19 had a positive impact

partially because of the extreme context, which caused people to work remotely and document more, because of the lack of communication.

### **5. The effect the software development methodology in Brella had on the Reputational assets dynamic capability before and during COVID-19.**

The methodology's effects on this dynamic capability can be considered from two perspectives – how much the reputation of the way the team builds a product affects the clients' decisions to buy the product and how much the same reputation affects potential new recruits' decision to get into the company.

The main opinion that was shared among the interviewees regarding how much the customers want to buy Brella's services because of its approach to developing the product was that the customers and the market do not really care about the methodology per se, but rather how fast and agile the development is – i.e., how fast it can release new features and how fast it can fix bugs. The general notion among the interviewees (especially those that were here before and after COVID-19) was that such reputation did improve, especially during the product pivot when the clients merely did not know how to continue organizing events as all in-person gatherings were banned. Brella's fast response to proposing virtual way of organizing events in some cases even saved the clients' businesses, thus improving the product reputation among them. At the same time, the interviewees mentioned that such reputation does not really spread outside of their current client base and even the clients that were impressed with it initially, gradually started to lose their trust in it as the competitors' growth intensified, thus making their development output better sounding than Brella's.

The interviewees also mentioned that the methodology never really had a strong effect on the reputation among the potential new employees. However, the interviewees also mentioned that this is something worth working on and starting to advertise as the company grows. Some interviewees mentioned that potential reason for lack of such an effect on reputation was that the Product team has been growing immensely lately so its methodology was reshaping quite a lot and thus there was no clear understanding of what to advertise in the human resources field.

Below you can see a citation from the interview with Brella's QA Specialist about that:

*“I think we should go a little bit more to the Agile side to actually be able to say that ‘we are doing things like this, this is so cool, come and work Agile with us’. Now we are still on the way, we want to get there, but we are not there yet. So, I don't think we can really brag about that”.*

*(Interview 6)*

Below you can see a citation from the interview with Brella's Product Designer about that:

*“I would say it will come to us along the way. That's something that will be formed also with the overall processes that we are forming in the way. The way how we communicate for example within the team, how we communicate with other teams...”.*

*(Interview 5)*

In summary, it seems that for the clients, the methodology per se does not really have an impact on the reputation, what matters is how many quality features and improvements you make. The reputation did increase a lot during the pivot, but then it started to gradually diminish. For the employees, there is no reputation really in place yet and has never really been.

#### **6. The effect the software development methodology in Brella had on the Structural assets dynamic capability before and during COVID-19.**

Most of the interviewees shared the same opinion that the structure in the Product team used to be and remained rather flat. Since the teams work in smaller independent units when developing a particular feature or when working on a particular project, they remain quite independent and do not have to consult with too many stakeholders or managers before being able to make a decision. Many interviewees consider that being an advantage of the way the Product team works and appreciate it being this way. Because of this reliance on smaller independent teams, the shift towards Waterfall way of doing things did not bring any negative effects on this dynamic capability. The team considers this flat structure with abilities for independent decision making a positive capability of the team and the company.

Below you can see a citation from the interview with Brella's Design Lead about that:

*"I think nowadays we are more Agile in that. [name of designer] is the designer, who will work on something, and I think the only time when he comes to me for approval is right before we submit the final designs".*  
(Interview 3)

Below you can see a citation from the interview with Brella's CTO about that:

*"Of course it depends on the feature, but let's say that we have some medium-sized feature. I see that the team can decide, so personally I am not interfering much there and same for CPO. If it's a bigger item, then we might have some input... but even with bigger items, the team can decide. It depends on the feature".*  
(Interview 7)

Overall, the effect of the methodologies on this dynamic capability did not change – it remained rather positive even under the extreme context conditions.

#### **7. The effect the software development methodology in Brella had on the Market (structure) assets dynamic capability before and during COVID-19.**

Because of the team's deep knowledge appearing from either fast cycles and close collaboration with customers (like it was in the past) or deep design research including prototype experimentation and showcasing prototypes to clients, the team was able and is still able to preserve this dynamic capability. The effect of the methodologies is positive, and it did not really change under the COVID-19 extreme

context. The team is still able to understand its market very well and thus preserve and improve its market assets.

Moreover, the team has proven during COVID-19 that it's also able to shift in case of a need to acquire and get established into newer niches. The team understands their company's competitive advantage and utilizes it as much as it can.

Below you can see a citation from the interview with Brella's FE Lead about that:

*“When the COVID hit, we changed a little bit how we work, and we started cutting more corners. We were fluent enough as a company to be able to change the way we work and then that allowed us to take advantage of this emerging markets that we saw because ours was dying and next to it there was market that was booming”.*  
(Interview 2)

Below you can see a citation from the interview with Brella's CPO about that:

*“The day we decided with the leadership that everything is going down, we sat together and thought about how can we approach this problem. Moving to virtual was something that we understood was needed because the event industry is not just going to die away [and we could not] just wait and do nothing. So, we sat together and drew some ideas ‘ok, what can we do to change the product from in-person logic to virtual logic’”.*  
(Interview 1)

All in all, both in cases of urgent need and in cases of normal long-lasting projects, the team can adjust and fit the market needs to preserve and acquire market assets.

## **8. The effect the software development methodology in Brella had on the Path dependencies dynamic capability before and during COVID-19.**

The team is not afraid of trying out new niches. The team does not experiment as much as Agile development would suggest, but instead they research more in the design phase. Such path-independent nature has improved during the COVID-19 extreme context as the team gained more experience and took into use this new Waterfall method. If need be, the team will transform and allocate some of the team members to work on potential new paths, but currently a lot of these things are being discussed only in the design stage or among the leadership.

As proven during the COVID-19 time, the team does not really mind exploring new markets and is able to do so in case of need. Currently though, the team does not really spend much time on experimenting and trying new things by default. Currently the team spends most of the time trying to develop in the chosen paths. However, the team is not scared of trying new paths and is able to do so rather easily.

Below you can see a citation from the interview with Brella's Lead Designer about this:

*“I think Agile methodology allows us to change much easier. I think we had a different version of how we want to be in next 3 years. As a designer, you need to*

*align with these kinds of visions. And I think if we were not Agile, if we were not in this kind of mindset, I think, it would have been much harder for us to change our paths... [and now] I think we are even more Agile in certain things”.*  
(Interview 3)

Below you can see a citation from the interview with Brella’s BE Lead about this:

*“Why I mean that we are really good at waterfalling is that somebody can move away from a project for a while and then everybody will continue their work on the Waterfall normally, and then somebody else will do something else for a while and come back”.*  
(Interview 4)

In summary, the team is rather flexible and can shift priorities rather easily, allowing room for trying out new things and not being dependent on the path. The effect is positive, and it even improved to some extent during the last year.

#### **9. The effect the software development methodology in Brella had on the Technological opportunities dynamic capability before and during COVID-19.**

What comes to technological opportunities, as mentioned before, the team’s methodology does not really benefit or has any positive effect here. Also, it did not really improve or worsen during the last year. The team follows industry standards and does not experiment much to try new technological opportunities.

Below you can see a citation from the interview with Brella’s Sr FE Developer about this:

*“We have downtime after each OKR, but that downtime might be one day, one week or one month. It just varies a lot, and I don’t think we talk too much that we should try and experiment stuff. It’s just when you are out of an OKR, you have no work to do, you go to issue board, try to find some bugs to fix – that’s it. We try to improve our current platforms, but we don’t experiment. I don’t think that has even been talked about that we could use a time to experiment features and stuff”.*  
(Interview 8)

Below you can see a citation from the interview with Brella’s CTO about this:

*“... [technological opportunities exploration] needs to somehow be related to the events industry or our product. We don’t have that luxury to start building something completely different. I like that approach: we would have that kind of hackathon-type of day or whatever we choose it will be called so that people could try out new things. I think, we should try that out”.*  
(Interview 7)

In summary, there is no real effect of the methodology on this. It could be positive if the methodology was to facilitate new technological experiments, however,



currently it does not. The only option to get some new technological opportunities discovered is through someone from the team or the leadership to deliberately say to try something out, but nothing is coming from the methodology per se.

#### **10. The effect the software development methodology in Brella had on the Assessment dynamic capability before and during COVID-19.**

In the past, the team used to be more Agile, and had more frequent feedback loops, which allowed them to keep a closer look on the market situation inside the development cycles. Nowadays, this shifted to having deeper design analysis, which also allows for high quality assessment from the Product team as mentioned by most of the interviewees. Deeper and more sophisticated design research helps the team to understand the situation in the industry and the overall trends from more global perspectives such as SaaS (software as a service) trends and UX/UI trends.

Below you can see a citation from the interview with Brella's QA Specialist about this:

*“I think one of the biggest strengths of Brella is the open communication. And I think that's a big part of it. You can see so much of communication between the different departments – so you can get an idea also of the business side even if you don't work there and you don't know anything outside of your own 'bubble'”.*  
(Interview 6)

Below you can see a citation from the interview with Brella's Product Designer about this:

*“We have this practice of directly contacting with the users of our product, with the people from the industry. This way we see the insights, their thoughts right away and that could be one source of these 'informational markers of the future changes'. And that's definitely an advantage of the project management approach we use”.*  
(Interview 5)

In summary, the effect of the methodology is strong and positive. From theory research that was done before in this thesis, it may sound that by going from more Agile to more Waterfall way, the assessment capabilities may worsen. However, most of the interviewees mentioned that the opposite is true. With more emphasis put on the design research, the team considers that it holds a very good grip on the market trends' pulse.

## 5 DISCUSSIONS

Overall, Brella's Product Development team's methodology moved from being primarily Agile to being primarily Waterfall. The move did not happen entirely for all the development processes that the team works with – only general elements became more Waterfall, whereas some elements also remained Agile. The design stage became much more powerful and much more important. A lot of things that were a part of the cyclical Agile nature of the team moved to design stage and the development was majorly left out for pure execution. The design documentation reliance increased.

The move happened not only because of the COVID-19 pandemic extreme context – it was only one of the reasons. The COVID-19 pandemic directly led to more difficult communication and shift towards remote work with asynchronous communication, higher reliance on documentation, tools, processes, and more structured work. Waterfall may seem like an obvious or even natural move in these kind of conditions as Agile only works best when there are small co-located teams.

Another equally important reason for such a shift was the Product Development team's double growth in headcount. The team was not prepared to onboard such a big amount of people while preserving its old ways of working – the natural shift was to tighten the reliance on processes and more structured way of working. However, the shift itself caused problems, such as new people starting to work in one way and using a new set of tools. However, the old employees would still use the older set of tools. This led to more difficult coordination of work and less trust in the old ways of working. Everyone was trying to build new ways to make it work because the old ways did not seem to work, and everyone was just trying to help. Another important reason to identify here is that potentially insufficient onboarding happened because the previous ways of working did not work anymore for the bigger team. However, it remains a question whether the old ways did not work for the bigger teams or whether they could have worked but were just not explained well enough. Whatever was the reason, the new people did not start adopting the old methods, but instead influenced the shift towards being more Waterfall.

The last but not the least reason was the market's instability where spending time on experimentation and development cycles on all fronts was not possible. The team tried to survive and cope with the growth and market changing needs, while also having troubles with communication and insufficient onboarding of new employees. The push towards Waterfall and more experiments and research happening in the design stage (which was perceived to be cheaper and less risky than the actual development) may again seem like the most obvious and natural move.

During the interviews, it felt that all interviewees were just saying what occurred to happen during the last year, not that they were deliberately trying to do a more Waterfall way. Therefore, it may sound like there was absence or lack of management. At the same time, the team does not consider itself being in a bad state. In turn, to most of interviewees it seems like the best way to work in current conditions.

What is even more interesting is that overall, the effect of the team's methodology on the dynamic capabilities mainly improved during the last year. See the table below for the summary of what happened for each dynamic capability during the last year.

Table 5. Summary of the methodology's effect changes on the dynamic capabilities.

Dynamic capability	The methodology's effect change	Reasons for changes
Coordination / Integration	<b><u>Coordination became harder;</u></b> Integration is not difficult and did not change	Difficulties in communication; Bad onboarding; Difficult market situation
Learning	<b><u>This transformed and, according to some, improved</u></b>	Change to being more Waterfall with more emphasis on design research
Reconfiguration and transformation	This transformed, but did not change	Change to being more Waterfall with more emphasis on design research
Technological assets	<b><u>This improved largely in the design team;</u></b> Otherwise mainly untouched	Focus on more documentation in design; Change to being more Waterfall
Reputational assets	<b><u>Reputation among clients and globally increased (temporarily);</u></b> Methodology reputation among potential recruits did not change	Outputs of the product team and the product pivot
Structural assets	Did not change	Did not change
Market (structure) assets	Did not change	Did not change
Path dependencies	<b><u>This transformed and, according to some, improved</u></b>	Change to being more Waterfall with more emphasis on design research; Experience with the product pivot under extreme context
Technological opportunities	Did not change	Did not change
Assessment	<b><u>This transformed and, according to some, improved</u></b>	Change to being more Waterfall with more emphasis on design research; Experience with the product pivot under extreme context

Waterfall methodology therefore may not necessarily sound like something that will ultimately kill the positive effect of the methodology on the company's dynamic

capabilities. In fact, the shift that the Product team made, improved four out of ten dynamic capabilities, and worsened only one. The shift did not change four and one was improved because of the outputs (i.e., feature and the product pivot) the team has made. The only thing that was directly influenced by the COVID-19 extreme context was the Coordination dynamic capability, where it had more difficult communication and more unclear market situation, which in turn led to more difficult coordination. All other effect changes were only indirectly caused by the extreme event, with other reasons prevailing.

The methodology's effects on dynamic capabilities changed partially because of the extreme context. Therefore, it may be safe to conclude that the extreme context may or may not lead to changes in the methodology's effects on dynamic capabilities. Furthermore, it may or may not even lead to changes in the software development methodology in the whole company.

## **5.1 Theoretical contributions**

There are two main contributions of this thesis. With the help of theoretical research and empirical study analysis, it helped to better state the relationship between the software development methodology and how it can affect certain dynamic capabilities. Even though, this research may be rather case-dependent, and its findings may not necessarily be applicable universally, it still managed to get some explanation on what dynamic capabilities can be theoretically affected by the software development methodology of a company, and what its relation may be in theory (Schilke et al., 2018). Moreover, this research then studied these theories in practice to see how it worked for one case example in real life (Schilke et al., 2018). The second main contribution of this research is related to the context in which the empirical case study has been done and the effects that the context has had on the relation of the software development methodology to the dynamic capabilities. It explained that the extreme context does bring additional factors that come into play when studying the methodology's effects on dynamic capabilities (Hällgren et al., 2018).

Apart from the dynamic capabilities and extreme contexts field, this research may also be interesting for software methodology theories. It gave some interesting implications on how the Agile and Waterfall methodologies would interact and replace each other in a growth company going through turbulent times (Geambaşu et al., 2011).

## **5.2 Managerial implications**

The findings of this research could also be especially interesting for managers and executives. It can be interesting and beneficial for both general managers, no matter what their field is and especially for the managers working in the product and software development fields.

For managers in general, it could be rather interesting to see how companies go through such extreme contexts like the COVID-19 pandemic – what effects it has on communication, remote work practices, project, and change management. It may also

show some findings on how difficult it may be to work in hyper growth environment and where the team could head towards given certain conditions and contexts, by considering which areas should be especially taken care of and which areas become less important.

Additionally, product development team members and managers, apart from the interesting implications on communication, growth, and other difficulties of surviving during the extreme context, can also learn quite a lot about the particulars of working in a product team. These include some challenges, benefits, and characteristics different approaches give – be it Agile or Waterfall. It could also give some rather interesting understanding of where the team would direct to given certain conditions and what elements of methodology it would be majorly using, given certain conditions. It is also a very interesting example of how the case company's team tried to cope with the hyper growth, and what drawbacks and benefits of the previous and current approaches it had.

Finally, it can also be interesting to see some implications from the dynamic capabilities' perspective. It turned out that product team's reliance on Agile approaches (or wrong implementation of them) may bring harder coordination, worse documentation, and worse onboarding – meaning that the team would be less ready to scale and grow in turbulent times. At the same time, Agile methodology benefits largely Assessment and Path Dependencies dynamic capabilities. With bigger presence of Waterfall however, Technological assets can be majorly improved with more structured and documented processes. However, for the managers it should be obvious that these are very much case dependent. The relationships and nature of methodology's effects on dynamic capabilities can vary largely based on the way certain elements of methodology are implemented, team structure, company, product and industry evolutionary stage, and many other reasons.

### **5.3 Future research directions and limitations**

More research would be needed to identify whether these kinds of effects would also happen in other software companies. This research does not give universal suggestions, but rather very case-specific descriptions of what and why certain things happened and how the theory worked in practice. Therefore, the amount of case studies is a very important limitation of this research, more case studies (three to six) would be needed among comparable companies in comparable research contexts.

Another interesting question to research would be whether different methodologies lead to different changes in the methodology's effects on dynamic capabilities under normal and extreme contexts. This research only studies Agile and Waterfall methodologies – some other methodologies could be especially interesting to study as well. Naturally, other components would also be very interesting to study such as whether the company, industry product and team evolutionary stages or contexts' nature affect the dynamic capabilities differently. This study is only limited to the COVID-19 pandemic extreme context which itself may be a rather big limitation as well.

## 6 CONCLUSIONS

The research has shown some interesting findings about how the methodology affects the dynamic capabilities of a company, and whether and how these effects change under the extreme contexts. It has studied some theoretical sources, first to give some background to the problem and summarize the theoretical knowledge in a framework, and then used the real-life case study to see whether the theory would work in practice.

Agile and Waterfall methodologies have been studied in this research as they were the present ones in the case study company. Also, they were identified as the two most popular methodologies for creating new features and new software projects. They were studied and analysed in the theoretical perspective and then the case company's implementation of these methodologies was also researched.

It was also identified that the software development methodologies can affect certain dynamic capabilities. It was proposed that Coordination / Integration, Learning, Reconfiguration and transformation, Technological assets, Reputational assets, Structural assets, Market (structure) assets, Path dependencies, Technological opportunities, Assessment can be affected by software development methodologies.

The extreme context – the COVID-19 pandemic – has influenced the way the software development methodologies affected these dynamic capabilities. However, the extreme context per se was not the main and only reason for changes in the effects, there were also other reasons. Agile methodology has established certain preconditions of how the product team could react to the changing environments. When the Waterfall methodology came in, the effects of the methodology on the dynamic capabilities started to change.

## REFERENCES

- Abrahamsson, P., Salo, O., Ronkainen, J., Warsta J. (2002). Agile Software Development Methods: Review and Analysis. *VTT Publications*, 478, 1-112. <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>
- Avison, D. & Fitzgerald, G. (1995). *Information Systems Development: Methodologies, Techniques & Tools*. 2<sup>nd</sup> ed. Berkshire: McGraw-Hill Education.
- Barney, J. B. (1991). Firm Resources and Sustained Competitive Advantage. *Journal of Management*, 17(1), 99-120.
- Brella.io official company website. Accessed on the 11<sup>th</sup> of April 2021. Retrieved from: <https://www.brella.io/>
- Beck, K., Beedle, M., Bennekum van, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D. (2001). *Manifesto for Agile Software Development*. Agile Manifesto (website). Accessed on the 25<sup>th</sup> of February 2021. Retrieved from: <http://AgileManifesto.org>
- Castillo, F. (2016). *Managing Information Technology*. Springer Link online library. Accessed on the 21<sup>st</sup> of March 2021. Retrieved from: <https://link-springer-com.ezproxy.jyu.fi/content/pdf/10.1007%2F978-3-319-38891-5.pdf>
- Cockburn, A. (2002). *Agile Software Development*. Boston: Addison-Wesley.
- Costa, M. (2020). *How events embraced digital transformation overnight*. Article on Raconteur. Accessed on the 7<sup>th</sup> of May. Retrieved from: <https://www.raconteur.net/corporate-events/events-industry-virtual/>
- Daily, L. *Git makes software development, well, easier*. Atlassian Agile Coach (website). Accessed on the 15<sup>th</sup> of February 2021. Retrieved from: <https://www.atlassian.com/agile/software-development/git>
- Eisenhardt, K. M., Martin, J. A. (2000). Dynamic Capabilities: What Are They? *Strategic Management Journal*, 21, 1105-1121.
- Finder.fi (2021). *Brella Oy*. Accessed on the 11<sup>th</sup> of April 2021. Retrieved from <https://www.finder.fi/IT-konsultointi+IT-palvelut/Brella+Oy/Jyv%C3%A4skyl%C3%A4/yhteystiedot/3128771>

- Forsgren, N., Humble, J., Kim, G. (2018). *The Science Behind DevOps: Accelerate: Building and Scaling High Performing Technology Organizations*. 1<sup>st</sup> ed. Portland: IT Revolution Press.
- Geambașu, C. V., Jianu, I., Jianu, I., Gavrilă, A. Influence Factors for the Choice of a Software Development Methodology. (2011). *Accounting and Management Information Systems*, 10(4), 479-494.
- Hadden, J., Casado, L. (2020). *Here are the latest major events that have been canceled or postponed because of the coronavirus outbreak, including the 2020 Tokyo Olympics, Burning Man, and the 74th Annual Tony Awards*. Business Insider (website). Accessed on the 9<sup>th</sup> of February 2021. Retrieved from: <https://www.businessinsider.com/major-events-cancelled-or-postponed-due-to-the-coronavirus-2020>
- Hällgren, M., Rouleau, L., De Rond, M. (2018). A Matter of Life or Death: How Extreme Context Research Matters for Management and Organization Studies. *Academy of Management Annals*, 12(1), 111-153. <https://doi.org/10.5465/annals.2016.0017>
- Hannah, S. T., Uhl-Bien, M., Avolio B. J., & Cavarretta, F. L. (2009). A framework for examining leadership in extreme contexts. *The Leadership Quarterly*, 20(6), 897-919. <https://doi.org/10.1016/j.leaqua.2009.09.006>
- Himanshu, V., Roshan, D. (2021). Events in Industry by Type Report. *Allied Market Research*. Accessed on the 7<sup>th</sup> of May. Retrieved from: <https://www.alliedmarketresearch.com/events-industry-market>
- Jones, C. (2018). *Software methodologies: a quantitative guide / Capers Jones*. Boca Raton: CRC Press.
- Orpana, L. (2019). *Exploring Uncertainty from Prioritization Perspective in Information Systems Development*. Information Systems Science, Master's Thesis. Accessed on the 11<sup>th</sup> of February 2021. Retrieved from: <https://jyx.jyu.fi/bitstream/handle/123456789/64891/URN%3aNBN%3afi%3ajyu-201906273494.pdf?sequence=1&isAllowed=y>
- Myers, J. (2021). *COVID-19: What you need to know about the coronavirus pandemic on 9 February*. World Economic Forum (website). Accessed on the 9<sup>th</sup> of February 2021. Retrieved from: <https://www.weforum.org/agenda/2021/02/covid-19-coronavirus-pandemic-9-february/>



- Paananen, J. (2020). *Brella announces turnkey solution for Virtual Events*. Brella (website). Accessed on the 9<sup>th</sup> of February 2021. Retrieved from: <https://www.brella.io/news/virtual-events-solution>
- Saunders, M., Lewis, P., Thornhill, A. (2009). *Research Methods for Business Students*. 5<sup>th</sup> ed. Rotolito Lombarda: Pearson Education Limited.
- Schilke, O., Songcui, H., Helfat, C.E. (2018). Quo Vadis, Dynamic Capabilities? A Content-Analytic Review of the Current State for Future Research. *Academy of Management Annals*, 12(1), 390-439. <https://doi.org/10.5465/annals.2016.0014>
- Stokel-Walker, C. (2020). *What the future of conferences could look like*. BBC (website). Accessed on the 9<sup>th</sup> of February 2021. Retrieved from: <https://www.bbc.com/worklife/article/20200826-what-the-future-of-conferences-could-look-like>
- Teece, D. J. (2007). Explicating Dynamic Capabilities: The Nature and Microfoundations of (Sustainable) Enterprise Performance. *Strategic Management Journal*, 28, 1319-1350. <https://doi.org/10.1002/smj.640>
- Teece, D. J., Pisano G., Shuen, A. (1997). Dynamic Capabilities and Strategic Management. *Strategic Management Journal*, 18(7), 509-533. [https://doi.org/10.1002/\(SICI\)1097-0266\(199708\)18:7<509::AID-SMJ882>3.0.CO;2-Z](https://doi.org/10.1002/(SICI)1097-0266(199708)18:7<509::AID-SMJ882>3.0.CO;2-Z)
- The Event App Bible 2021*. (2021). EventMB Studio Team. Accessed on the 11<sup>th</sup> of April 2021. Retrieved from: <https://www.eventmanagerblog.com/event-app-bible>
- Bizzabo Report. (2020). *The Evolution of Events Report*. Accessed on the 7<sup>th</sup> of May. Retrieved from: <https://welcome.bizzabo.com/reports/evolution-of-events-report>
- Wenxiao, B., Feng, Y., Yue, Y., Feng, L. (2017). Organizational Structure, Cross-functional Integration and Performance of New Product Development Team. *Procedia Engineering*, 174, 621-629. <https://doi.org/10.1016/j.proeng.2017.01.198>
- Wenzel, M., Stanske, S., Lieberman, M. B. (2020). Strategic responses to crisis. *Strategic Management Journal*, 41, 7-18. <https://doi.org/10.1002/smj.3161>
- What is Gitflow and how it Works?* NexusLink (website). Accessed on the 15<sup>th</sup> of February 2021. Retrieved from: <https://nexuslinkservices.com/git-workflow-in-software-development-life-cycle/>

- Williams, S., Hill, K. (2020). *Global Conference, Communications Week™, Taps Leading Event Platform Brella for Transformative Digital-First Experience*. Article on BusinessWire website. Accessed on the 7<sup>th</sup> of May. Retrieved from: <https://www.businesswire.com/news/home/20201008005891/en/Global-Conference-Communications-Week%E2%84%A2-Taps-Leading-Event-Platform-Brella-for-Transformative-Digital-First-Experience>
- World Bank. (2020). *The Global Economic Outlook during the COVID-19 Pandemic: a Changed World*. Accessed on the 5<sup>th</sup> of April 2021. Retrieved from: <https://www.worldbank.org/en/news/feature/2020/06/08/the-global-economic-outlook-during-the-covid-19-pandemic-a-changed-world>
- World Health Organization (2021). *Coronavirus, Overview*. Accessed on the 5<sup>th</sup> of April 2021. Retrieved from: [https://www.who.int/health-topics/coronavirus#tab=tab\\_1](https://www.who.int/health-topics/coronavirus#tab=tab_1)

## **APPENDIX 1     Brella platform feature overview**

### **Platform**

- Virtual, hybrid and in-person events – using Brella you can host all types of events.
- Livestream hosting – you can embed your livestream or webinar into the Brella platform for all attendees to see.
- Multi-livestream support – you can embed multiple livestreams or webinars into the Brella platform.
- Event home page – there is a special page that the attendee land on once they have joined your event inside the Brella platform, which gives an overview of everything they can do on the platform.
- Audience interaction (chat, polls, questions and answers) – you can have open chats, polls and questions and answers next to your livestream or webinar inside Brella which will allow for live interaction for your attendees.
- 1:1 video calls – once your attendees have booked a meeting with each other, they can meet using video conferencing features of the Brella platform.
- Virtual booths for exhibitors and sponsors – your sponsors can have their own booths to share advertisement materials and interact with the attendees.
- Breakout rooms and group meetings – attendees can have group meetings instead of just the one-on-ones and can join breakout rooms.
- Time zone support – the platform supports experience for attendees that are in different time zones.
- Available on all platforms – the platform is available for your attendees on iOS native app, Android native app, all web browsers, and all responsive web browsers.

### **Networking and matchmaking**

- Smart matchmaking algorithm powered by artificial intelligence – once entering your intents and interests at the conference, your attendees will be matched with the best people they should meet using the platform's artificial intelligence algorithm.
- 1-to-1 meeting scheduling – the attendees can schedule meetings with each other, be it in virtual, in-person or hybrid landscape.
- Dedicated meeting slots – attendees can schedule meetings for specific meeting slots so that they can plan their whole event experience.
- Automated notifications – attendees will be notified about the things they shouldn't miss, such as an important session starting, attendee's meeting starting and many others.
- In-app messaging – attendees can message each other in the app.
- Designated meeting area – in in-person events attendees can see where they are supposed to meet each other.
- Meeting reminders – the attendees will be reminded of the new meeting requests, and meetings that they have confirmed.

- Hosted 1:1 video conferencing – attendees can use the in-built video-conferencing tool to talk to each other in the meetings.
- Event Schedule – attendees can see the whole event schedule, including their meetings, the seminars and other content going on in the conference.
- What's live – attendees can easily see what seminars are live and join them instantly.
- Multitrack agenda – if there are sessions going simultaneously, attendees can see and choose the tracks that want to follow.
- Schedule management – attendees can bookmark the sessions that want to attend and thus manage their own schedule at the event.
- Dedicated speaker section (bios, links, etc.) – attendees can see the speakers presenting at an event and learn more about them.
- Session tags – attendees can easier orient in the conference schedule by knowing the tags associated with different sessions.
- Session locations – attendees can know where to go to for their sessions of choice.
- Schedule embedded widget – the event organizers can build the conference schedule in the Brella platform and then embed this schedule on their own conference website.
- Schedule API widget – if the event organizers have abilities to establish API connection between the Brella's schedule and their own schedule tools, then can also easily do so.
- Calendar sync – attendees can sync their calendars with their preferred calendars such as Outlook Calendar, iCal, or Google Calendar.
- Schedule import and export – event organizers can download and upload the schedule contents.

### **Sponsors, partners and exhibitors**

- Sponsor listing and grouping – sponsors can be listed on the platform.
- Sponsor bio – sponsor description information can be listed on the platform.
- Website, FB, Twitter & LinkedIn links – sponsor website and social media links can be listed on the platform.
- Sponsor dashboard – event organizers can see how their sponsors are doing – how much traffic they have, how many meetings they had and much more.
- Sponsor ads – you can set up advertisements of your sponsors on the Brella platform.
- Virtual booths - you can set up virtual replication of in-person sponsor booths on the platform where sponsors can list their materials and offerings and interact with the attendees.
- Shareable content (demo videos, brochures, etc.) – sponsors can list shareable materials inside their virtual booths.
- Live chat – sponsors can live chat with the attendees visiting their booths.

### **Event management**

- Hosted buyer program – it is possible to book meetings on behalf of your attendees as an event organizer.
- Custom branding – you can customize the looks and feels of your event using the customization features.
- Event floor map – in case it's an in-person event, it's possible to list the venue floor map for all attendees to orient at the event.
- Links to event website and event social media accounts – it is possible to list information about your event including social media accounts and other links as well as detailed event description.
- Custom push notifications – you can notify your attendees of any announcements in your event.
- Event join code and join link – you can share links for how attendees should join your event. If you choose, you can use custom invitation-only links so that only particular attendees can join your event.
- Event info pages – any information you have about your event can be listed in dedicated info pages.
- Event reminder emails – send automated emails to remind attendees about the event.
- Event engagement emails – send automated emails to spur engagement at your event.
- Meeting reminder emails – send automated emails to remind those that haven't answered to their meeting requests to facilitate higher engagement.

### **Data and Analytics**

- Networking and Event Engagement – learn how engaging and successful your event was data-wise.
- Number of meetings – see how many meetings happened.
- Who is meeting who – learn what meetings happened and who were the participants.
- Pre-engagement vs during event engagement - learn what was the difference in engagement before and after the event.
- Livestream analytics – learn the traffic analytics and how engaging your live streamed shows or webinars were.
- Market research/Behavioural Analytics – learn what were the things the attendees were looking for in your event – i.e., what were their intentions and interests and who they wanted to meet at the event.
- Supply and demand signals – learn what type of attendees was prevailing in your event and whether you need to improve in certain aspect.
- Attendee intents – learn what were your attendee intents from attending your conference.
- Understand market trends – see the market trends in the industries you are in by knowing what your attendees' intents and interests were.
- Sponsorship sales data – use data about how many meetings and how much traffic your sponsors had at the event to know better how to sell to them.

- Marketing analytics – get analytics on whether you need to market your conference in this way or another by knowing your attendees’ interests and intents.
- Needs for specific content – learn what type of content your attendees are mostly interested in by learning what were their intents and interests at the conference.
- Event CRM integrations – establish integrations with your CRM systems to easily transfer the data into the system you use.

### **Customer Success and Networking Consultation**

- Dedicated Customer Success Manager – use the services of our dedicated CSM to learn how to best utilize the platform and ensure the success of your event.
- Kickoff meeting – get to know the product and the different options you have for how to ensure the best event possible.
- Event Communications support – get to know the best practices for how to communicate about the event and the platform to all types of your stakeholders.
- Event Sponsorship Consultation – learn how to ensure that your sponsors get the most out of the event.
- Matchmaking Consultation – learn how to make the best matchmaking and networking experience possible for your attendees.
- Support materials – get support materials on various topic aimed at making your event the best.
- User and technical support – get support on any technical hiccups you’re your attendees or sponsors experience with the platform.
- Full end-user support – all your attendees and sponsors will be able to get instant support in case of any problems with the platform.
- Full technical support - any potential bugs or problems will be fixed in timely manner.

## APPENDIX 2 Interview questions

There were two rounds of interviews. The first one included only the employees that have been working in Brella before the COVID-19 pandemic and during it. The second round of interviews included only the employees that came in during the COVID-19 pandemic and so could only see the situation as it turned out to be.

The first round of interviews included the following questions.

1. Can you describe your process of developing the software product from start to finish?
2. Based on your description, it seems that it is more related to Agile or Waterfall methodology? What do you think? Why do you think this methodology suits your company best? Would you like to change it? Why?
3. How were you affected by the COVID-19 extreme event? What effects did it have on the product development process?
4. Do you think the way you develop the product affects any dynamic capabilities of your company? How did it change because of COVID-19?
  - a. What effects did the methodology have on Coordination / Integration dynamic capability in normal circumstances? How did it change because of COVID-19?
  - b. What effects did the methodology have on Learning dynamic capability in normal circumstances? How did it change because of COVID-19?
  - c. What effects did the methodology have on Reconfiguration and transformation capability in normal circumstances? How did it change because of COVID-19?
  - d. What effects did the methodology have on Technological assets dynamic capability in normal circumstances? How did it change because of COVID-19?
  - e. What effects did the methodology have on Reputational assets dynamic capability in normal circumstances? How did it change because of COVID-19?
  - f. What effects did the methodology have on Structural assets dynamic capability in normal circumstances? How did it change because of COVID-19?
  - g. What effects did the methodology have on Market (structure) assets dynamic capability in normal circumstances? How did it change because of COVID-19?
  - h. What effects did the methodology have on Path dependencies dynamic capability in normal circumstances? How did it change because of COVID-19?
  - i. What effects did the methodology have on Technological opportunities dynamic capability in normal circumstances? How did it change because of COVID-19?
  - j. What effects did the methodology have on Assessment dynamic capability in normal circumstances? How did it change because of COVID-19?

The second round of interviews included the following questions. As can be seen, the questions did not include the perspectives on how the things have changed – only they are now.

1. Can you describe your process of developing the software product from start to finish?
2. Based on your description, it seems that it is more related to Agile or Waterfall methodology? What do you think? Why do you think this methodology suits your company best? Would you like to change it? Why?
3. Do you think the way you develop the product affects any dynamic capabilities of your company?
  - a. What effects does the methodology have on Coordination / Integration dynamic capability?
  - b. What effects did the methodology have on Learning dynamic capability?
  - c. What effects does the methodology have on Reconfiguration and transformation capability?
  - d. What effects does the methodology have on Technological assets dynamic capability?
  - e. What effects does the methodology have on Reputational assets dynamic capability?
  - f. What effects does the methodology have on Structural assets dynamic capability?
  - g. What effects does the methodology have on Market (structure) assets dynamic capability?
  - h. What effects does the methodology have on Path dependencies dynamic capability?
  - i. What effects does the methodology have on Technological opportunities dynamic capability?
  - j. What effects does the methodology have on Assessment dynamic capability?