Affect, Affective Contagion and Decisions in Agile Development

A Thesis
Submitted to the Faculty of Graduate Studies and Research
In Partial Fulfilment of the Requirements
For the Degree of

Doctor of Philosophy
In
Software Systems Engineering
University of Regina

By
Abdulaziz Abdurabuh Alhubaishy
Regina, Saskatchewan
September, 2018

Copyright 2018: A. Alhubaishy
Abdulaziz Abdurabuh Alhubaishy, candidate for the degree of Doctor of Philosophy in Software Systems Engineering, has presented a thesis titled, *Affect, Affective Contagion and Decisions in Agile Development*, in an oral examination held on August 31, 2018. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner:  
*Dr. Ebrahim Bagheri, Ryerson University*

Supervisor:  
*Dr. Luigi Benedicenti, Adjunct*

Committee Member:  
Dr. Craig Gelowitz, Software Systems Engineering

Committee Member:  
**Dr. Mohammed El Darieby, Software Systems Engineering**

Committee Member:  
Dr. Malek Mouhoub, Department of Computer Science

Chair of Defense:  
Dr. Andrei Volodin, Department of Mathematics & Statistics

*Via ZOOM Conferencing  
**Not present at defense*
ABSTRACT

During the past decade, research on how affects—including emotions, moods, and feelings—influence software developers’ performance has increased. In software development, this influence has been considered and investigated mostly in terms of individual developers. Agile development, which requires collaboration and communication, can be more prone to the influence of affects on both individuals’ and teams’ decisions. There is a lack of a tested model in the literature that explicitly reflects the influence of developers’ affects when they make decisions in Agile. Furthermore, there is no explicit evidence of the influence of positive or negative affective contagion on software teams’ cooperation or level of conflict. Therefore, this thesis develops an improved understanding and theorizes the role of affects and their contagion in Agile decision-making processes. This thesis contributes by answering the main research question, which is: how are affects and their contagion related to decision making in Agile environments? To answer this question, the thesis adopted the explanatory sequential design method, which is considered a mixed-methods approach. Within sequential design, we applied survey research as quantitative design to test the role of affects and affective contagion, and to generalize the findings, and constructive grounded theory as qualitative design to collect and analyze data, and to understand the contributing factors for the quantitative model.

A total of 249 valid survey responses were collected and examined. These responses were analyzed to understand the current awareness of the impact of affects
and affective contagion on Agile decisions, cooperativeness, and conflicts. Both expert and novice Agile members confirmed the influence of affects and affective contagion on Agile team decisions, cooperativeness, and conflicts; however, novices’ acknowledgement of the role tended to be significantly less than that of expert Agile members. Furthermore, the PANAS scale was provided to Agile team members to extract the most influential affective states that play a role in Agile decisions. The results showed that being determined, inspired, active, and attentive were the four positive affective states that most influence Agile individuals and teams. On the other hand, being upset and distressed were the two most influential negative affective states. Then, parts of the survey data were used to test the measurement and structural models of affect and affective contagion; and the I-PANAS-SF scale was adopted to measure participants’ affective states. The measurement models were built and tested against the collected data. After reaching an acceptable model fit with the measurement model, the structural models were built and tested against the collected data. All research hypotheses were supported by testing for the significance of direct and indirect paths. We also examined the mediators in an aggregated model and confirmed the mediation of cooperativeness and conflicts between affects and affective contagion and decision quality.

A total of 16 in-depth interviews were collected and analyzed, and the constructivist grounded theory approach was adopted. As a result, a general framework, which we named “The Framework of Affects and Affective Contagion as Influential Factors on Agile Decision-Making Processes,” was introduced and discussed. The framework articulates seven concepts; namely, conceptualizing positive and negative affects, conceptualizing positive and negative affective contagion, linking affects and affective contagion, recognizing situations in which affects influence individuals, recognizing situations in which affective contagion influence teams, acting on affects to influence affective contagion, and work- and non-work-related factors.
Acknowledgement

First of all, I would like to express my special thanks of gratitude to my supervisor, Dr. Luigi Benedicenti, for his guidance during completing this thesis. Without his help, cooperation, and encouragement, this thesis would not be completed. One of the best things that happened in my life is conducting my research under his supervision. Expressing the amount of respect I carry for him cannot be explained with only these few words.

I would also like to thank the Saudi Cultural Bureau in Canada and Saudi Electronic University for their financial support during all these years.

Special thanks to my committee members, Dr. Malek Mouhoub, Dr. Mohamed El-darieby, and Dr. Craig Gelowitz, for their effort, time, support, and valuable comments.

Special thanks and gratitude to all FGSR staff; especially the Dean, Dr. Kathy McNutt, for their help to overcome many challenges I encountered during my study. The moments of their help are unforgettable.

Lastly, I would like to thank all of my friends for their encouragement and valuable discussions. Special thanks to my friends, Abdulhadi Alqarni and Ahmad Alzahrani, for every day fruitful discussion and meeting. I have benefited a lot from them to overcome many challenges I encountered during this research journey. Also, special thanks to what we call “The Apartment Friends”, who have been part of my success by reducing the stress after long days of study and research.
Dedication

I would like to dedicate this research to my mother, Sayeeda, and my father, Abdurabuh, for their lifelong support of me. Even though they have passed away before I started my PhD journey, their support, wisdom, and advices were guiding me all times. May Allah grant them with mercy and gather us in Jannah.

I would also like to dedicate this research to my wife, Amal, and my children, Kinan and Kiyan. Their existence in my life have provided me with the courage and ambitious to get this research done.

Finally, I would like to dedicate this work to my brothers and sisters; especially, my lovely sister, Dalal.
TABLE OF CONTENTS

ABSTRACT i

ACKNOWLEDGEMENT iii

DEDICATION iv

TABLE OF CONTENTS v

LIST OF FIGURES xiii

LIST OF TABLES xv

LIST OF PUBLICATIONS xvi

ABBREVIATIONS xviii

1 INTRODUCTION 1

1.1 Research Problem ........................................ 1

1.2 Motivation .............................................. 3

1.3 Research Objectives .................................... 4

1.4 Methodology and Scope .................................. 6

1.5 Original Contributions ................................ 8

1.6 Organization of Thesis ................................ 9

1.7 Chapter Summary ...................................... 10
2 Literature Review

2.1 Agile Methodology

2.1.1 Decision Making in Agile

2.1.2 Cooperativeness in Agile

2.1.3 Conflicts in Agile

2.2 Affects (Moods and Emotions)

2.2.1 Measuring Affects

2.3 Affects, Affective Contagion and Making Decisions

2.4 Affects, Affective Contagion and Software Process development

2.4.1 Developer’s Affects

2.4.2 Stakeholder’s Affects

2.5 Affects, Affective Contagion and Making Decisions in Agile Environments

2.6 Measures and Guidelines for Studying peoples’ affects and its contagion in Software Engineering

2.7 Chapter Summary

3 Research Framework

3.1 Research Questions

3.2 Model Formation and Hypotheses Development

3.2.1 The Relationship Between Affects, their contagion and Quality of Decisions

3.2.2 The Relationship Between Affects, their contagion and Cooperativeness

3.2.3 The Relationship Between Affects, their Contagion and Conflicts

3.3 Conceptual Research Model

3.3.1 The Model of Affect Influences on Cooperativeness, Conflicts, and Decision Quality
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.2</td>
<td>The Model of Affective Contagion Influences on Cooperativeness, Conflicts, and Decision Quality</td>
<td>66</td>
</tr>
<tr>
<td>3.3.3</td>
<td>The Aggregated Model of Affect and its Contagion Influences on Cooperativeness, Conflicts, and Decision Quality</td>
<td>70</td>
</tr>
<tr>
<td>3.4</td>
<td>Control Variables</td>
<td>72</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Experience</td>
<td>73</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Agile Method</td>
<td>74</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Role of Team Member</td>
<td>74</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Gender</td>
<td>75</td>
</tr>
<tr>
<td>3.5</td>
<td>Chapter Summary</td>
<td>76</td>
</tr>
<tr>
<td>4</td>
<td>Research Methodology</td>
<td>78</td>
</tr>
<tr>
<td>4.1</td>
<td>Research Designs</td>
<td>78</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Quantitative vs. Qualitative Designs</td>
<td>78</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Mixed Methods Design</td>
<td>81</td>
</tr>
<tr>
<td>4.2</td>
<td>Research Methods</td>
<td>83</td>
</tr>
<tr>
<td>4.3</td>
<td>Research Approach of this Thesis</td>
<td>84</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Research Approach</td>
<td>84</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Research Plan</td>
<td>91</td>
</tr>
<tr>
<td>4.4</td>
<td>The Adopted Procedure for Quantitative Data Collection</td>
<td>93</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Questionnaire Development</td>
<td>93</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Sample Size</td>
<td>100</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Questionnaire Instrument Validation</td>
<td>102</td>
</tr>
<tr>
<td>4.4.4</td>
<td>A Pilot Study</td>
<td>103</td>
</tr>
<tr>
<td>4.5</td>
<td>The Adopted Procedure for Quantitative Data Analysis</td>
<td>106</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Data Screening</td>
<td>107</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Testing Validity for a Measurement Model</td>
<td>108</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Testing Validity for a Structural Model</td>
<td>109</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Affects theory of impact on programming performance [1]</td>
<td>34</td>
</tr>
<tr>
<td>3.1</td>
<td>Affects Impact on Agile Team’s Cooperativeness, Conflicts, and Decisions</td>
<td>62</td>
</tr>
<tr>
<td>3.2</td>
<td>Affective Contagion Impact on Agile Team’s Cooperativeness, Conflicts, and Decisions</td>
<td>67</td>
</tr>
<tr>
<td>3.3</td>
<td>Conceptual Model of Affects and Affective Contagion Impact on Agile Decisions</td>
<td>70</td>
</tr>
<tr>
<td>4.1</td>
<td>Quantitative and qualitative research, differences and similarities [2] [3]</td>
<td>79</td>
</tr>
<tr>
<td>4.2</td>
<td>Explanatory Sequential Design [4]</td>
<td>86</td>
</tr>
<tr>
<td>4.3</td>
<td>Surveys, advantages and disadvantages [2]</td>
<td>87</td>
</tr>
<tr>
<td>4.4</td>
<td>Components of the Thesis Approach</td>
<td>91</td>
</tr>
<tr>
<td>4.5</td>
<td>Research Plan for the Study of Affects and Affective Contagion Role in Agile Decisions</td>
<td>92</td>
</tr>
<tr>
<td>4.7</td>
<td>Steps for Conducting Structural Equation Modeling (SEM)</td>
<td>110</td>
</tr>
<tr>
<td>4.8</td>
<td>Grounded Theory Coding Process and Phases, Adapted from [6]</td>
<td>116</td>
</tr>
<tr>
<td>5.1</td>
<td>Demonstration of the participants’ opinions about the influence of independent variables on dependent variables</td>
<td>124</td>
</tr>
<tr>
<td>5.2</td>
<td>Experts Vs. Novices Opinions (Decision Quality)</td>
<td>125</td>
</tr>
</tbody>
</table>
5.3 Experts Vs. Novices Opinions (Cooperativeness) ........................................... 126
5.4 Experts Vs. Novices Opinions (Conflicts) ......................................................... 126
5.5 Demonstration of the participants' opinions about the Agile team member's Positive affects of PANAS Scale that influence individuals ............................... 128
5.6 Demonstration of the participants' opinions about the Agile team member's Negative affects of PANAS Scale that influence individuals ............................... 129
5.7 Demonstration of the participants' opinions about the Agile team member's Positive affects of PANAS Scale that are spread to members ............................... 130
5.8 Demonstration of the participants' opinions about the Agile team member's Negative affects of PANAS Scale that are spread to members ............................... 131
5.9 CFA, Adjusted Model ......................................................................................... 138
5.10 Structural Model of the Affect (significant paths only) ......................................... 143
5.11 Hypotheses Test Results for the Affect Model ..................................................... 144
5.12 Structural Model of the Affective Contagion (significant paths only) .................... 145
5.13 Hypotheses Test Results for the Affective Contagion Model ................................ 146
5.14 Structural Model of the Affect and Affective Contagion (significant paths only) ........ 148
5.15 Hypotheses Test Results for the Aggregated Model of Affect and Affective Contagion ................................................................. 150
6.1 The Framework of Affects and Affective Contagion as Influential Factors on Agile Decision-Making Processes ............................................................ 158
6.2 The Concept of Conceptualizing Affects and its properties .................................... 160
6.3 The Concept of Conceptualizing Affective Contagion and its properties .................... 165
6.4 The Concept of Linking Affects and Affective Contagion and its properties ............... 168
6.5 The Concept of Recognizing Situations When Affects Influence Individuals and its properties ................................................................. 171
6.6 The Concept of Recognizing Situations When Affective Contagion Influence Teams and its properties 174
6.7 The Concept of Acting on Affects to Influence Affective Contagion and its properties 178
6.8 The Concept of Work and Non Work-Related Factors and its properties 182
C.1 Certificate of Approval 258
D.1 Pooled-CFA for the Affect Model 259
D.2 Pooled-CFA for the Affective Contagion Model 260
E.1 Detailed Framework of Affects and Affective Contagion as Influential Factors on Agile Decision-Making Processes 261
LIST OF TABLES

3.1 Summary of Research Hypotheses ........................................ 72
4.1 Strength and Weakness of Mixed Methods Design [7] ............. 82
4.2 Surveying Participants’ opinions of the influence of independent variables on dependent variables ........................................ 95
4.3 Research Constructs, Items and their Sources ....................... 99
4.4 Reliability of the Constructs and their Items ......................... 105
5.1 Demographic Information for valid responses ....................... 121
5.2 Reliability Analysis for each independent variable .................. 123
5.3 Summary of participants’ opinions about the influence of affects and affective contagion on Agile decisions, cooperativeness, and conflicts . 124
5.4 Testing Results of Participants’ opinions of the influence of independent variables on dependent variables ......................... 127
5.5 Exploratory Factor Analysis .................................................. 132
5.6 Reliability of the EFA Factors and their Items ....................... 135
5.7 Correlation Matrix ............................................................ 136
5.8 Validity and Reliability of the Measurement Model .................. 140
5.9 Factor Correlation Matrix .................................................... 140
5.10 Fitness Indices for the measurement model ............................ 141
5.11 Direct, indirect, and total effect of factors on Decision Quality .... 151
List of Publications

Parts of the thesis have appeared previously in some publications, and other parts have been submitted for publication.

Published Papers:


Submitted and In Progress Papers:


# ABBREVIATIONS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSE</td>
<td>Behavioural Software Engineering</td>
</tr>
<tr>
<td>CFA</td>
<td>Confirmatory Factor Analysis</td>
</tr>
<tr>
<td>CFI</td>
<td>Comparative Fit Index</td>
</tr>
<tr>
<td>CONF</td>
<td>Conflicts</td>
</tr>
<tr>
<td>COO</td>
<td>Cooperativeness</td>
</tr>
<tr>
<td>DQ</td>
<td>Decision Quality</td>
</tr>
<tr>
<td>EFA</td>
<td>Exploratory Factor Analysis</td>
</tr>
<tr>
<td>FA</td>
<td>Factor Analysis</td>
</tr>
<tr>
<td>GFI</td>
<td>Goodness of Fit</td>
</tr>
<tr>
<td>GT</td>
<td>Grounded Theory</td>
</tr>
<tr>
<td>I-PANS-SF</td>
<td>International-Positive And Negative Schedule-Short Form</td>
</tr>
<tr>
<td>ISD</td>
<td>Information Software Development</td>
</tr>
<tr>
<td>PA</td>
<td>Positive Affect</td>
</tr>
<tr>
<td>PAC</td>
<td>Positive Affective Contagion</td>
</tr>
<tr>
<td>PANAS</td>
<td>Positive And Negative Affective Schedule</td>
</tr>
<tr>
<td>RE</td>
<td>Requirement Engineering</td>
</tr>
<tr>
<td>NA</td>
<td>Negative Affect</td>
</tr>
<tr>
<td>NAC</td>
<td>Negative Affective Contagion</td>
</tr>
<tr>
<td>SAM</td>
<td>Self-Assessment Manikin</td>
</tr>
<tr>
<td>RMSEA</td>
<td>Root Mean Square Error of Approximation</td>
</tr>
<tr>
<td>SEM</td>
<td>Structural Equation Modelling</td>
</tr>
<tr>
<td>SPANE</td>
<td>Scale of Positive and Negative Experience</td>
</tr>
<tr>
<td>STEM</td>
<td>State-Trait Emotion Measure</td>
</tr>
<tr>
<td>TLI</td>
<td>Tucker-Lewis Index</td>
</tr>
<tr>
<td>TMX</td>
<td>Team-Member Exchange</td>
</tr>
<tr>
<td>VAD</td>
<td>Valence, Arousal, and Dominance</td>
</tr>
<tr>
<td>XP</td>
<td>eXtreme Programming</td>
</tr>
</tbody>
</table>

---

xviii
Chapter 1

INTRODUCTION

This chapter introduces the main research problem and question. The chapter also highlights the motivation for conducting this research and summarizes the research objectives and the methodologies used to achieve these objectives. Finally, the chapter briefly mentions the accomplished contribution of this thesis.

1.1 Research Problem

Many psychological, economic, and engineering studies have investigated affect as an influential factor in making decisions. “Affect” is an umbrella term that covers both emotions and moods, and it has been adopted by many authors, such as Weiss et al. \cite{weiss2018}; therefore, we use the terms “emotions”, “moods”, and “affects” interchangeably. Some investigations have found that different types of affects influence the making of decisions, such as current and expected emotions \cite{hall2019}, while different factors influence the decision makers’ affects, such as work environment \cite{smith2015}, problem characteristics \cite{brown2016}, personality \cite{john2018}, and experience \cite{lane2017}. Among the different models, Lerner et al. investigated all types of affects as well as the factors that influence them \cite{lerner2018}.

There is a gap in our understanding concerning the role of affect in software development decision making. Most of the research that has been conducted on
the decision making process and its relation to affects has focused on managerial decisions. For example, Dingsøyr et al. highlighted five challenges related to the effectiveness of software teams [14]. One of the issues they pointed out is the lack of studies that investigate team cognitive tasks as collaborative cognition in current team effectiveness models.

Xu and Wang proposed a framework that includes the influence of not only expected and incidental emotions but also immediate emotion on intelligent behaviour [9]. They considered all types of emotions, as well as different triggers for the emotions, in order to enhance the decision-making process even under uncertain decisions. Most decisions in software development projects are made under uncertain and risky conditions in all of the categories listed by Colomo-Palacios et al.; however, it seems that the researchers only considered one type of emotion, expected emotions, in their investigation.

Barsade has explained the influence of affective contagion on group behaviour from a managerial decisions perspective [15]. That study concluded that “the positive emotional contagion group members experienced improved cooperation, decreased conflict, and increased perceived task performance” [15].

There is evidence of the role affects play in software decision making. For example, Colomo-Palacios explained how requirement prioritization is influenced by affects even when many requirements techniques are available [16]. His study concluded that the “negative emotions are present in all hard decision making, even if the sample is composed of highly-skilled professionals where emotions like anxiety or resignation are not common beforehand” [16].

However, although the research by Colomo-Palacios et al. is interesting and their results are valuable, some shortcomings are evident. First, their investigation is only based on experience rather than measuring the actual affect at the time of decision making. Second, categorizing decisions might allow the interviewed expert to think
only about one situation that falls under that category; hence, researchers might elicit inaccurate affects. Third, emotions that come from different sources were neglected; rather, the researchers investigated only those affects caused by decision making. A recent study included the influence of not only expected and incidental affects but also immediate affect on intelligent behaviour [9]. However, it seems that Colomo-Palacios et al. considered only one type of affect, which is expected affects.

The main problem is that, within the field of software engineering, there is no tested model that explicitly reflects the influence of developers’ affects when they make decisions. Furthermore, there is no explicit evidence of the influence of positive or negative affective contagion on software teams’ cooperation, conflict, or task performance. Therefore, we need to develop an improved understanding of and theorize the role of affects and their contagion in Agile decision-making processes. In response to these problems, this thesis makes a contribution by answering the main research question, which is, how are affect and its contagion related to decision making in Agile environments? This question is answered by investigating sets of hypotheses and research questions, as we will explain later in Chapter 3.

1.2 Motivation

A number of researchers have shown the impact of positive and negative affective states on workers’ performance and productivity. Promising results have been found, recognizing individual developers’ affective states and how they influence performance, creativity, problem solving, and more. Recently, Graziotin et al. examined all consequences of being happy or not as a developer while analyzing, designing, implementing, and producing software [17]. The authors have stated that “developers experience happiness more as benefiting themselves, and unhappiness more as being detrimental to others.” The study has constituted great motivation for authors to
study how happiness is recognized by and contagious to others. We share this motivation for studying the influence of affects and their contagion within teams, and more specifically, Agile teams. In line with Graziotin et al., we gained this motivation from earlier studies in organizational psychology and industry that were conducted on the role of affects and their influences in workplaces.

Team cooperation and communication have been also found to be influenced by affective states. Agile processes, as people-oriented processes, account for the role of human factors more than the traditional development processes, as they aim to enhance communication. As an example, Shore et al. highlighted the problem of people not communicating effectively as being a main contributor to poor software quality [18]. Therefore, relating the level of communication and collaboration to the role of affects in different industries has motivated us to conduct this research.

By characterizing these roles, Agile development processes can be enhanced by avoiding or minimizing the recognizable consequences of negative affects and affective trust, thus avoiding the consequences of conflicts and enhancing the cooperation between Agile team members. Furthermore, positive affect and affective trust can be characterized and maximized to decrease conflicts and enhance cooperation.

Furthermore, by characterizing these roles, we provide the building blocks for future researcher to develop systems in order to capture and recognize these moods, emotions, and feelings. Without providing real evidence of the impact of affective states in Agile processes, there will still be a lack of motivation to develop and implement systems that recognize these affects.

1.3 Research Objectives

The main objective of this research is to theorize the role played by affects, emotions, and moods along with their contagion in the software decision-making process and,
more specifically, in the Agile environment. This research seeks to investigate how affect and its contagion are related to decisions in the Agile environment. Understanding this relationship can be achieved by achieving various distinct objectives.

First, affects and their contagion are still neglected in many professional environments. Making Agile decisions, affects and affective contagion are not considered significant factors that influence individual and team decisions. On the other hand, the literature has shown that affects and affective contagion have their implications with regards to decisions made in different managerial and industrial workplaces. The problem exists when Agile practitioners are unaware of the roles of affects and affective contagion. Therefore, the first objective is to understand Agile team members’ opinions and thoughts about these roles.

Second, the research intends to test whether or not there is a positive or negative influence of the affects and their contagion on individuals and teams by collecting real data and finding real evidence from different projects. Therefore, this research develops a set of hypotheses regarding the influence of affects and affective contagion and tests them scientifically to avoid biases in terms of team members’ opinions about the influence.

Third, the literature shows a lack of adoption of psychological measurements in software development. Although the literature has highlighted the adoption of some psychological measurements by individual developers, such as in [19], Agile development still lacks the adoption of psychological measurements and their findings regarding Agile teams. Therefore, validating different psychological measurements is an objective of this research.

Fourth, one research objective is to build, test, and validate measurement models as building blocks to highlight the role of affects and affective contagion within agile development. This objective includes having different measurement models for affect and affective contagion, and also a general measurement model for considering the
relationship between affects and affective contagion.

Fifth, another research objective is to build, test, and validate a structural model regarding the results of the measurement models for investigating the paths of influence from affects and affective contagion to the quality of decisions. Similarly, this objective includes different structural models for affect and affective contagion, and a general structural model for considering the path of influence between affects and affective contagion.

Sixth, cooperativeness and conflicts have been highlighted in the literature as key influences in Agile decisions, where affects and affective contagion can have an effect on these influences. Therefore, highlighting the effect of affects and affective contagion on team cooperativeness and conflicts, along with relating them to agile decisions, is an objective of this research.

Seventh, a further research objective is to understand and characterize how each highlighted path of influence within the structural model works, starting from the initiation of the affects to the consequences of these affects for the Agile decision-making process. Achieving these objectives will enable the reader of this thesis to understand the current roles and influences of affects and affective contagion regarding Agile decision quality and related factors, such as cooperativeness and conflicts.

1.4 Methodology and Scope

This research was carried out through various phases and by using various methods, strategies of inquiry, and data collection tools. In general, we adopted the explanatory sequential design method, which is considered a mixed-methods approach. This required starting by collecting quantitative data, followed by qualitative data. Within the sequential design, we applied surveys as a quantitative method, to test the role of affects and affective contagion and generalize the findings, and interviews as a quali-
tative method to collect and analyze data and understand the contributing factors of the quantitative model.

The quantitative research strategy was used through the adoption of the survey method to collect data about the affects and their contagion that play a role in decision making based on developers’ experiences and opinions. Mainly, the survey collected three types of data. The first type includes data about team members’ opinions of the influences of affects and affective contagion on Agile team decisions, cooperativeness, and conflicts. The second type of data involved adopting different psychological scales to acquire all types of positive and negative affects that play a role in Agile team decisions, cooperativeness, and conflicts. The last type of data includes Agile team members’ assessments of their last project’s outcomes, such as decision quality, level of cooperativeness, and level of conflicts among them. The analysis of these data involved adopting Structural Equation Modelling (SEM) to enable us to build, test, and validate both measurement and structural models that ultimately provide the scientific means for testing the research hypotheses.

The qualitative research adopted the constructive grounded theory in order to help us understand, interpret, contrast, and explain the results of the qualitative study along with the preliminary quantitative results. This is an important step to derive better understanding and confirm the hypotheses regarding the role of affects and affective contagion as related to Agile decision making based on in-depth interviews. Furthermore, the objective of the qualitative study is to understand how affects and affective contagion influence cooperativeness and conflicts and, hence, the overall quality of decisions made by team members. Interviews as qualitative strategy allowed us to gather information in different formats than the quantitative methods. For example, expert managers might have their opinions on how affects influence team members’ decisions in Agile, and the quantitative approach does not offer a way to grab and analyze such observations. The result of adopting the grounded theory
approach is a framework that articulates the process of how Agile team members experience and act on their affects.

1.5 Original Contributions

The original contribution of this thesis is multifaceted. First, the thesis contributes an improved understanding of the current state of the awareness of the influence of affects and their contagion among Agile team members. This understanding supported the idea of investigating how the current level of awareness participates in maximizing the negative influences of negative affects, while not exploiting the positive influences of positive affects.

Second, the thesis provides confirmed results regarding the relationship between affect and affective contagion and different influential factors in terms of Agile decisions, based on the adoption of many psychological and managerial scales. As a result, the thesis provides two frameworks to test positive and negative affects and affective contagion between Agile teams in order to enhance Agile teams’ cooperativeness and performance, lower conflicts, and to make decisions more accurate. Furthermore, the thesis aggregates the aforementioned frameworks to articulate the relationship between affects and affective contagion and how they stimulate each other in Agile teams.

Finally, the thesis contributes by characterizing how Agile team members experience and behave in relation to positive and negative affects and their spread during the decision-making process. As a result of this characterization, how individuals conceptualize, recognize, and act on affects, along with how teams conceptualize, recognize, and act on affective contagion, have been articulated within a general framework.

The main contributions of the thesis are succinctly described in the following points:
• Confirming the role of affects and affective contagion by acquiring Agile team members’ opinions.

• Understanding the current awareness level of affects’ and affective contagion’s influences in Agile development and factors that hinder escalating this level of awareness.

• Adopting different psychological scales for measuring the affects within Agile development.

• Extracting the most influential positive and negative affective states in terms of Agile decisions, cooperativeness, and conflicts.

• Building, testing, and validating measurement models for affect, affective contagion, and their interactions within the Agile development based on different psychological and managerial scales.

• Building, testing, and validating a structural model for each measurement model that highlights the path of influence from affect to the Agile decision quality.

• Characterizing and articulating the process of how affects and affective contagion influence Agile teams’ decisions, cooperativeness, and conflicts.

1.6 Organization of Thesis

The rest of this research is organized as follows:

Chapter 2 presents most of the related work and empirical studies of affect and its relationship with software development, as well as empirical studies of the roles of cooperativeness and conflicts in Agile processes.

Chapter 3 introduces the proposed models, research questions, and development of each hypothesis related to each relationship between the two constructs.
In Chapter 4, we outline the research design, methods, and approaches used in this study, as well as justifying the adoption of the approaches used. Furthermore, the chapter details the procedures adopted for collecting and analyzing both quantitative and qualitative data.

Chapter 5 provides the results of the survey findings. The chapter illustrates the steps used in testing the quantitative data by constructing and enhancing both measurement and structural models.

Chapter 6 provides the results of adopting the constructive grounded theory approach. The chapter articulates the current process in terms of how Agile team members conceptualize, recognize, and act on affects and affective contagion during Agile decision-making processes.

Chapter 7 discusses the results of the two previous chapters and answers the research questions. The chapter discusses the results by combining, comparing, and explaining the results for each set of hypotheses along with its relevant research question that investigates the influence of affects and their contagion on Agile decision quality, cooperativeness, and conflicts.

Finally, Chapter 8 concludes the thesis and explains its theoretical and pragmatic implications. Furthermore, the chapter mentions the implications of this thesis for future research.

### 1.7 Chapter Summary

- From an engineering management perspective, decision making helps software developers, managers, and other team members to find optimal solutions by relying on one decision-making model to identify the problem, examine all possible alternatives, and determine the optimal solution.

- From a psychological perspective, decisions are influenced by affects, emotion,
and moods.

- The role of affects and the factors that influence them were investigated from a managerial decisions perspective.

- The need for a deep understanding of the role of affect, emotions, and moods on software development has attracted many researchers.

- No empirical studies were found to investigate the influence of affects and their contagion on Agile decision making.

- Objective: Understanding Agile team members’ opinions and thoughts about affects’ and affective contagion’s roles.

- Objective: Collecting data and finding real evidences of the roles played based on Agile team members’ assessments of their project outcomes.

- Objective: Building and validating measurement and structural models to prove the influence of affects and affective contagion on Agile team decisions, cooperativeness, and conflicts.

- Objective: Characterizing how each highlighted path of influence within the structural models works.

- The mixed-methods design, in the form of the sequential design method, was chosen for conducting the research activities.

- Various data collection tools were used, based on each method and the strategies for inquiry.
Chapter 2

Literature Review

2.1 Agile Methodology

2.1.1 Decision Making in Agile

In this section, we highlight the main studies conducted on decision-making processes regarding the role of the decision-making process, types of decisions, the effect of processes and practices on decisions in terms of group or individual decisions, and other factors influencing the decision-making process in the Agile environment.

First, software development requires processes for making decisions at various stages, such as when gathering requirements, designing, and testing. The decisions are also considered at different levels, including operational, tactic, and strategic levels. Accurate decisions enhance the outcomes of the process; hence, they result in better software quality and reduced cost and time for successful software development. This is true, as evidence is found in [20], where the need to understand and enhance the decision-making process for Requirement Engineering (RE) is emphasized. However, the roles played by decision making and decision support are different, based on their different schools of thoughts in engineering management [20]. The main difference between the two schools concerns whether there is an optimal solution for all problems.
or not. Other differences are summarized by [20], such as the model, the process, the alternatives, type of problems, and whether the problem can be structured or not.

The decision making school of thought helps software developers, managers, and other team members to find optimal solutions, while it depends on only one model to deal with the problem and suggest best solution. It also assumes the possibility of comparing all available alternatives to find an optimal solution. The decision making school of thought also ignores the type of problem, such as ranking or selection, regarding ranking alternatives; hence, a way must be found to compare all available alternatives. Finally, the problem with the decision making model is that it must be structured in all cases in order to find the optimal solution.

On the other hand, the decision support school of thought believes that there is no optimal solution for most problems. Instead, it seeks suitable solutions for the real problem. Decision support does not compare all alternatives if the comparison is difficult. It also considers the differences between different types of problems, such as ranking and selection. Finally, decision support works as an exploratory model to explore the solution; hence, the problem does not need to be structured like it does in decision making. Ruhe described the methodology and applications of decision support in software engineering [21]. He illustrated the main requirements for successfully applying decision support methods, which are to have valid and comprehensive models and data, the ability to collect and analyze data, technologies to develop and analyze satisfactory solutions, and powerful infrastructure.

Anthony’s proposed model [22] consists of three groups of decisions based on three abstraction levels at organizations, which are operational, tactical, and strategic decisions. Operational decisions are the daily decisions made by lower-level managers or workers. This group of decisions has less influence on the organization but significantly impacts the daily production process. Tactical decisions exist between the strategic and operational levels. These involve applying the decisions made at the
higher strategic level, to ensure that the process at the organization is going well. Strategic decisions are made at the higher level of the organization, and they influence the whole organization. Strategic decisions help to achieve the main goals of the organization. Obviously, uncertainty increases with higher risk at the strategic decisions level, while information at this level is rare; therefore, experienced managers are required. On the other hand, operational decisions are more certain and involve less risk when the information is available. Strategic decisions’ risks, certainty, and availability of information exist between the strategic and operations decision levels.

Among the different models, most decisions in the software industry are classified under Anthony’s hierarchical structure. Ruhe et al. illustrate decisions made at different levels [20]. Software developers participate in making operational decisions, for example, choosing the design module and requirement prioritization, and making release decisions. Planning decisions, such as deciding on resources and time, are made at the tactical level. In other words, decisions made during project planning in software production are at the tactical level. Finally, decisions concerning main tools, the software development process, and enhancements to a process are examples of decisions made at the strategic levels. Based on each Agile practice, Benedicenti provides a framework that highlights the decisions made at each level [23]. The framework identifies each insertion point that can accept decision-making methods in order to enhance decisions made at all three levels. A total of 14 decisions were collected from 6 practices in eXtreme Programming (XP). The framework provided by Benedicenti opens avenues to explore more insertion points at the various decision making levels. Drury et al. categorize decisions in software development into two types, which are tactical and strategic decisions [24]. Their research mainly focuses on finding the obstacles to decision making in Agile developments. They categorize daily and short-term decisions as tactical decisions, while long-term decisions that influence organizations are considered strategic.
Have researchers examined how decisions are organized and made in software development? Answering this question opens up the door for new and enhanced decision-making models that can ultimately enhance software development process outcomes. Saarelainen et al. investigated how decisions are organized in different organizations, such as public and private ones [25]. They interviewed 29 experienced developers and applied qualitative methods to gather and analyze the collected data. The results showed that group decisions are most often used by both public and private organizations. More interestingly, decision-making processes are structured during a preparation stage, followed by the actual decision-making stage. The decision-making stage includes only the selection process, while the preparation phase includes the development phase and part of the selection phase, which is the evaluation step. Results also highlight research areas that need to be investigated, such as the organizing and training of groups, as well as individual awareness of risks related to group decision making.

Other qualitative research has tried to understand the relationship between software process, practices, and software developers’ empowerment [13]. The authors refer to empowerment as the ability to participate in making decisions and, more specifically, how software developers actually participate in the decision-making process and when they participate. The authors interviewed 11 software developers and classified their answers based on various criteria, such as their level of participation (high or low), how informed they were about decisions, and the types of decisions they participated in (operational or strategic decisions). The authors divided organizations that the developers participated in into four, while they applied two dimensions and four hierarchical levels based on the literature. The degree dimension spans between not participating in decisions to control in participation. The hierarchy levels are operational decisions, which denote daily activities in programming, testing, and design; resource allocations and tasks management, which denote assigning developers
to tasks and prioritizing these tasks efficiently; and low and high-level strategic decisions, which denote strategic changes and business decisions, respectively. Results show that the experience of developers plays an important role in their empowerment, i.e. developers’ involvement in making decisions, regardless of the type of organization, such as whether it is an organization with an undefined software development process or an organization with a well-defined development process. When the authors controlled for the experience factor, they found that the most empowered developers were found in Agile teams, while small in-house teams were found to include the least empowered developers.

Zannier et al. investigated how design decisions are made [26]. Their investigation used semi-structured interviews within multi-case studies. The interview goal was to record changes in design made by the participating developers as well as to describe the changes. Then, based on participants’ descriptions of changes, the authors asked further questions to elicit more information. Participant responses were analyzed and classified based on the type of decision method used, which involved naturalistic or rational decisions. Results showed that there is a relation between structuring a problem and the type of decision-making method. That is, the use of the rational decision method increased as the design problem was more structured and vice versa. Moreover, results showed that whether naturalistic or rational methods are used is the choice of developer at the beginning of making a decision, then later the other method might interfere. Based on the results, the authors proposed a decision-making model with three components, which are structuring flow, mechanisms, and perspectives. The first component, structuring flow, concerns transforming unstructured design problems into structured problems. The second component, structuring mechanism, represents all data used by decision makers to make decisions or to structure a design problem, including experience, opinions, and other sources of data. The third component, structuring perspectives, represents the decision method that was used, such
as using the rational or naturalistic method during design.

In line with previous research, Zannier and Maurer investigated earlier design decisions made within Agile environments [27]. They conducted interviews with 25 expert developers who were divided into two groups, namely developers and mentors. Mentors were defined by the authors as developers who were interviewed about others’ designs. By comparing the results from both types of participants, based on the interviews from 12 case studies, it was found that the naturalistic design decision-making method was most often used in Agile environments, while the rational design decision-making method supports it. Unlike in [26], which suggests either the naturalistic or rational method is the dominant method and supported by the other, Zannier and Maurer state that the naturalistic method is dominant in Agile, while the rational method is the supported method [27]. Zannier and Maurer earlier identified five areas in design, as well as identifying where the decisions used are either naturalistic or rational. These areas are implementation, structural, interaction, architectural, and usability design. For a definition of each area, we refer you to [28].

Colomo-Palacios et al. investigated the effects of difficult decisions within software processes [16]. Their study was conducted in two steps. First, they explored all difficult decisions that developers face during the software development process. Second, they explored the emotions’ influence on these decisions in order to study their implications. The results of the second step are mentioned later in Section 2.4.2. Colomo-Palacios et al. explored the difficult decisions by conducting interviews with expert developers. The exploration used the Delphi method in order to gather and analyze data from the group of developers. The exploration was conducted through two phases. During the first phase, 25 experts were asked to list five difficult decisions they faced while developing a project. Then, the authors aggregated all responses, analyzed them, and classified the decisions into groups. In phase two, these decisions were introduced to all expert developers, who were asked to rank them based
on their difficulty. As a result, 96 decisions were listed, and 60% of these decisions were “project prioritization, requirements prioritization, development strategy election, partner-supplier election, and internal personnel selection,” which were listed by the majority of expert developers [16].

Decision-making models in RE are proposed to solve problems, such as requirement elicitation, prioritization, and negotiation. The importance of having models for decision making is based in the reality that RE requires taking into account additional issues besides the general theory of making decisions [29]. First, stakeholders in the software development process are responsible for explaining system requirements to developers. This is a communication issue that occurs most often at the group level. The decisions of stakeholders influence the efficiency and effectiveness of the software; hence, they influence the deadline, price, quality, ... etc. Second, the decisions influence the three mentioned levels of decisions at organizations, which is mentioned by Anthony [22] and summarized for RE by Regnell et al. [29] as follows: 1) operational decisions in RE are concerned with deciding if the quality of a requirement is good, the possible action for applying such a requirement, and the requirement value, 2) strategic and 3) tactical decisions are grouped into resource decisions, such as increased resources time for requirement elicitation, responsibility decisions, such as being responsible for implementing the requirement, and scope decisions to ensure requirements are consistent with the strategic software goal. Based on Regnell et al.’s research from 2001, operational and strategic decision levels as well as 8 other areas are suggested for further research.

Alenljung and Persson conducted an interesting study to discover the nature of RE in large organizations [30]. They proposed a general framework to describe the decision-making situation in RE that includes various factors, such as problem characteristics, decision makers' characteristics, class of decision makers (individual or group), decision input, decision output, and decision-making activities. The authors
conducted interviews with 15 requirement engineers divided into three groups in order to validate their results by triangulation. The authors concluded the following: 1) decisions in RE are not considered decisions that are interpreted by requirement engineering as a regular activity or judgement and, in turn, it “can lead to getting stuck in a certain course of action,” 2) numerous aspects that can influence RE are overlooked, such as developers’ skills, development process, and culture, and 3) there is a need to construct a “body of knowledge” regarding decisions in RE [30].

Moe and Aurum used the three management levels while investigating the decision-making process used in two projects that used Agile as the underlying process [31]. Their main goal was to investigate the decision theory within Scrum process to understand work coordination and team communication, feedback, estimation, and planning. The authors investigated the negotiation between scrum-master and product owner in order to highlight the decision-making process at all management levels, which includes strategic, tactical, and operational decisions. Interviews with developers of two projects took place after introducing Scrum, and data analysis was conducted using Nvivo software. The authors concluded that there are three main obstacles to applying Scrum regarding decisions. First, strategic and management overlap in decisions when a misalignment problem prevents the successful use of Scrum. Second, experience plays a negative role, especially at the operational level, as well as in terms of connecting the three levels. The authors understand the importance of specialization; however, they find that specialization can be a barrier to alignment. Third, they note the importance of daily meetings in Scrum to avoid decision hijacking.

Factors that impact the decision-making process can be divided into general and specific factors. By general factors we mean the ones that can influence the decision-making process in any industry, such as culture and personality. The factors that can influence the decision-making process in software companies are the specific factors.
We mention this because software production mainly focuses on people’s knowledge, experience, and their ability to solve problems, i.e. people-oriented processes.

2.1.2 Cooperativeness in Agile

Communication between developers is an important value in Agile manifesto. The Agile value insists that interactions between Agile team members is more important that the process and tools [32]. Communication in Agile plays an important role that reduce Agile project failures. According to Warden and Shore, issues related to people communicating effectively are of the main challenges that leads to decrease the chances of building great software projects [18]. The authors have highlighted these issues to include “unpredictability of moods and motives, or figuring out how to harness people’s desire to do the right thing for the team and the project” [18].

For example, XP is an example of Agile methods that considers communication as a very important value. Beck , in his book, has highlighted the goal of communication in XP by stating that “XP aims to keep the right communications flowing by employing many practices that can’t be done without communicating. They are practices that make short-term sense, like unit testing, pair programming, and task estimation. The effect of testing, pairing, and estimating is that programmers and customers and managers have to communicate” [33].

Cooperation between team members cannot be easily identified, understood, and explained. For example, Robertson et al. state that “there is no mystery to why people cooperate. Cooperation occurs when each person believes that he or she will benefit more by cooperating than by acting in some other way” [34]. The author explains the main reason that groups cooperate is to gain mutual benefits such as profits or improving self interest. Huhns supports the notion of self interest to improve the cooperation between team members to achieve either unified common goal or individual goal [35].
Cooperation in Agile goes beyond communication which, according to Collierin, means having a “smooth transfer of work in progress, work products, and information from one member to another. Team has a shared commitment to a common outcome, and individuals coordinate their activities in ways that support other group members. In a cooperative team, members interact in an egoless manner and understand their individual roles as they relate to the group’s objectives” [36]. Yilmaz et al have found the cooperation between software team members has a major influence on developers’ productivity [37].

Factors that influence the cooperative communication between team members have attracted many researchers. For example, Balijepally et al. have used the five factor model of personality, which are “neuroticism, extraversion, openness to experience, agreeableness, and conscientiousness”, in order to assess their effects on Agile team cooperation [38]. The authors concluded that agreeableness and neuroticism have direct influence on the cooperative communication of Agile teams. The authors also concluded that “in Agile teams presence of agreeable team members who are not very different from each other on this trait would be highly desirable” [38]. Another study has investigated the role of personality in any organizational team performance which states that “teams with higher cognitive ability and conscientiousness were judged to be better performers than teams that were lower on general mental ability and conscientiousness” [39].

2.1.3 Conflicts in Agile

Based on different perspectives, different researchers have classified conflicts differently. In general, there are two types of conflicts called affective and substantive conflicts [40]. The authors differentiate the two conflicts by considering the affective as conflicts related to the interpersonal issues; while the substantive as conflicts related to team’s task issues. Conflicts between team members have been studied by
many researchers. Some researchers have tried to find the causes of conflicts in team. For example, an analysis of literature has been conducted by Dreu and Weingart to find out the relationship between conflicts and teams’ performance and satisfaction and tasks [41]. The authors concluded that “conflict had stronger negative relations with team performance in highly complex (decision making, project, mixed) than in less complex (production) tasks. Finally, task conflict was less negatively related to team performance when task conflict and relationship conflict were weakly, rather than strongly, correlated” [41].

Regarding software development teams, fewer number of investigations and studies have been conducted. Jiang et al. have studied the role of conflicts in the distributed software teams [42]. The researchers have studied the influence of conflicts on distributed teams regarding their performance and satisfaction. In this study, four types of conflicts; namely task, cognitive, emotional, and process, have been investigated based on highlighted literature about conflicts. The researchers have hypothesized that all types of conflicts have influence on employee’s satisfaction with respect to team interaction. The results showed negative influence of both process and emotional conflicts; and positive influence of both task and cognitive conflicts on satisfaction. The results of the study have confirmed that team interaction has influence on task and emotional conflicts; while these two types of conflicts have direct influence on distributed team members satisfaction.

Liu et al. have investigated the influence of conflicts on software development processes and their outcome [43]. The researchers have surveyed 153 developers in order to test whether or not conflicts influence the process, process, and management performance. The researchers have hypothesized that the conflicts have negative influence on these three performances despite the existence of good software development process. They have concluded that “Conflict negatively impacts all three and the process does not mediate the impact on the final product or project outcomes. From a process...
perspective, conflict will reduce the ability to conduct the development in orderly fashion even though good development practices do improve outcomes. The product will not be of as high a quality. The project is more likely to fail to meet goals of budget, schedule and scope” [43].

Earlier study has investigated the role and characteristics of interpersonal conflicts in Information Software Development (ISD) [44]. The authors’ main goal was to define and assess the personal conflicts, their relationship to conflict management strategies and project outcome. The study has surveyed 537 staff and users of IS in more than 160 projects in order to test a postulated model that correlates between interpersonal conflicts, their antecedents, and their outcome. As a result, the study has concluded that three properties of interpersonal conflicts were found influencing IS projects; namely disagreement, interference, and negative emotion. These three properties, in addition to interdependences properties, were already highlighted by the previous literature in different fields such as psychology, communication, and conflict management.

Another study has investigated the role of conflicts between teams in ISD in order to find the relationship between conflicts among team members, users interaction and project performance[45]. The authors hypothesized that conflicts will negatively influence the project performance, and user interaction. The study has surveyed 212 developers and they have concluded that there is a direct relation between conflicts among teams and the project performance; while they ended their discussion by stating that “A team that can maintain focus and direction better is a team that has a greater chance of delivering a successful project” [45].

In general, there are five classifications of management styles for dealing with conflicts; namely Dominating, Avoiding, Compromising, Integrating, Obliging [46]. Nair defines these styles as “1- Dominating: high concern for self and low concern for the other or task; 2- Obliging: low concern for self and high concern for the other,
3- Avoiding: low concern for self and low concern for the other, 4- Integrating: high concern for self and high concern for the other, and 5- Compromising: moderate concern for self and moderate concern for the other” [17].

2.2 Affects (Moods and Emotions)

In recent decades, psychological researchers have introduced many definitions of affect or affective states, with no definite agreement on how to define these terms. In order to understand what affective states or affects are, we compare the definitions provided by various researchers regarding the relation of affects to the mind, components of affect’s definition, inclusion or exclusion of components from other definitions, conflicts with other definitions’ components, the relation to other words commonly used similarly (such as “mood” and “emotion”), and types of affects (such as positive, negative, or mixed affects).

In 1980, Hilgard stated that although the tripartite classification of the human mind is constructed of three main components, which are affects, conation, and cognition, the affect component is still neglected by psychological researchers [48]. The connection between the three components of the human mind and how they influence each other was the subject of many empirical studies. Over the last two decades, more attention has been paid to affect and its relation to conation and cognition, as well as efforts to specify what constructs affects in psychology and social psychology. Sloman et al. defined affective states as including moods, desires, emotions, and attitudes [49]. “Emotions” and “moods” are commonly used in daily language instead of the scientific term “affects”.

Forgas et al., as well as Sloman et al., argue that affects and mood are used interchangeably and integrated into the understanding of behaviour [50]. The Oxford Dictionary of Psychology defines affect as “Emotion or subjectively experienced feel-
ing, such as happiness, sadness, fear, or anger” [51]. Another psychology dictionary defines affect as “the act of feeling a sentiment spanning from distress to extreme joy...” [52]. Currently, emotions and moods are referred to frequently in many studies. More precisely, emotions are defined by the Oxford Dictionary as short psychological states, a term that includes happiness, joy, sadness, or any other inner feeling, while a mood is considered to be a “temporary but relatively sustained and pervasive affective state” [52]. Clore highlights a difference between feelings or emotions and bodily states, such as feeling hungry [53].

From the definitions of emotion, affect or affective state, and mood, this research is grounded in an understanding of affect based on the similarities among the definitions. To avoid the debate about defining what affect is, this research considers affect according to the common definition from the dictionary and Sloman et al. for three reasons. First, we consider the general definition of affect that encompasses the main components of affect as well other terms, such as emotions. We are investigating how affects influence decision makers in Agile environments which requires studying their emotions, moods, or affects. Second, the general definition, as stated by [49], is accepted by many psychological researchers, while moods, emotions, and affects are used interchangeably, as in [54] and [55]. Third and most important, this research tends to work in a manner consistent with the work of Graziotin et al. [19], who founded a new discipline called psychoempirical software engineering. Therefore, along with [19], we use the terms emotions, moods, and affects interchangeably, to maintain consistency in terms of understanding affect and its implications for software engineering.

Positive and negative affects occur to the human mind based on positive or negative stimuli. Positive affect occurs as an internal feeling when an objective is accomplished, danger is avoided, or in general when a good feeling causes happiness [55]. On the other hand, negative affect occurs when an objective is not accomplished,
danger is not avoided or becomes more significant, or in general a bad feeling causes sadness [55].

### 2.2.1 Measuring Affects

The concept of core affect was proposed by Russell et. al in order to allow researchers to measure the affective state of humans in terms of good or bad feelings [56]. They considered the core affect to be the smallest but still accessible element in affects and emotions that can be measured. Measurements, scales, and theories for measuring affects have been the concern of psychological researchers since the invention of the tripartite classification of the human mind. Russell et al. proposed a continuous scale for measuring pleasure and displeasure as positive or negative, and a continuous scale for measuring arousal as high arousal or low arousal [56].

Watson and his colleagues developed the PANAS scale, which was one of the first reliable scales to measure affects [57]. PANAS stands for Positive And Negative Affective Schedule, and it works based on stability for two months. The authors analyzed more than 60 items and came up with 10 validated items for positive and 10 items for negative affects. Each item is a word that is used to express an affective state, such as “afraid” or “scared” for the negative affective scale and “alert” or “active” for the positive affective scale. Six years later, Clark et al. extended the PANAS scale to cover more than two dimensions [58]. They included, in addition to positive and negative scales, 18 items for a basic positive emotional scale, 23 items for a basic negative emotional scales, and 14 items for other affective states, such as shyness and fatigue.

Due to culture differences and limited time for utilizing the scale, Thompson shortened the PANAS scale to only 10 items, naming it the International - Positive And Negative Schedule - Short Form (I-PANS-SF), for both positive and negative affects [59]. Thompson validated the scale in terms of different cultures, proving the
Levine et. al. invented a scale called the State-Trait Emotion Measure (STEM) that uses items related to affects from many scales. A total of 10 items were selected to represent both positive and negative affects [60]. The scale aggregates discrete emotions into an index of positive and negative traits. It was mainly developed and tested to measure affects as related to performance in the workplace. STEM was validated in the United States, China, and Romania to ensure its consistency and cross-cultural validity. STEM is different because it provides a definition for each item as well as an example of that item. STEM also allows the participant to rate their feeling based on a 10-points scale, from 0 for the least intense feeling to 10 for the most intense feeling related to the term as experienced during the working day, as well as the general feeling when at the workplace.

The Self-Assessment Manikin (SAM) scale, developed in 1994, is one of the most popular scales [61]. SAM is a 9-point scale with three dimensions for measuring arousal, pleasure, and dominance. SAM uses pictorial measures instead of verbal scales. For each emotion, such as arousal, there are five manikin pictures that the participants choose from, or sometimes they choose between two pictures. SAM has been validated and is now used in many contexts, such as advertisements and the economy, to measure affective states created by various stimuli in those contexts. Recently, the Affective Slider has been used in an attempt to enhance how participants express their feelings of arousal and pleasure, using a digital slider [62]. The authors insisted on the need to enhance the manikin pictures that were developed in the 1980s, based on modern designs and the state-of-art user interface, in order to ensure that participants are accurately evaluated.
2.3 Affects, Affective Contagion and Making Decisions

The decision-making process has been a concern since 1960 when decisions were considered as cognitive behaviours; thus, behavioural decision theory emerged as researchers investigated cognitive mistakes that decision makers should consider to enhance decisions. Affects were neglected when studying decision makers’ behaviours; however, Clore investigated how not only direct affects, that are related to decisions to be made, can play an important role in making decisions, but how also indirect affects, that are not related to decision to be made, can also play a role [53]. Clore highlighted that the main components that influence feelings during decision making include emotions related to known or unknown actions, heuristics developed though previous experiences of making decisions, and other factors, such as distraction and salience.

According to Loewenstein and Lerner, in order to understand how emotion enters into decision making, two types of affective influences need to be addressed [63]. Regarding the first type, the decision maker predicts the emotion that occurs to him/her when choosing an alternative and making decisions in a way that maximizes positive predicted emotion and minimizes the negative predicted emotion. For example, when a person evaluates two options and thinks of the alternatives like “if I choose alternative A, I would regret that if the outcome is as good as I expected, comparing it to alternative B” [63]. The emotion in this type is not experienced at the time of making the decision; therefore, it is called an expected emotion.

Regarding the second type, the decision maker experiences the emotion when deciding on an alternative and making a judgement. This kind of emotion is called immediate emotion, which influences the decision-making process either directly, by influencing the decision maker’s behaviour, or indirectly, by influencing the expec-
tation regarding outcome probability. Han and Lerner classified the influences of immediate emotion into two major influences, namely anticipatory and incidental influence. The anticipatory or integral influence is the one that arises from thinking of the decision and its consequences. On the other hand, the incidental influence arises from factors other than the decision’s consequences. Incidental influences can be due to environmental or physical factors.

Both types of influences relate to other tripartite systems that are cognitive and behavioural. The effects change over time in various ways, as Gutnik et al. illustrated. They explained that emotions are changed by stability of perceptions, unexpected events, and behaviour consistency. Stability of perception is affected over time by three factors, while behaviour consistency is influenced by perceptions and emotions over time.

Loewenstein and Lerner classified the problems in making decisions, based on the literature, under two major problems, which are making decisions when facing risks and making decisions with increased consequences over time. Expected Utility (EU) models deal with making decisions when facing risks, with many enhancements that try to understand the effect of emotions on peoples’ decisions, see and . On the other hand, the Discounted Utility (DU) model works in parallel with EU, but for delayed decisions over time and how the choice of an alternative at this time influences the probability of choosing the alternative at another time. For example, DU assumes that decision makers generally give less weight to delayed consequences and experience pleasure and pain as gain and loss.

Regardless of the ongoing debate concerning emotions as possibly a separate factor that influences or does not influence cognition, Epstein and Seymour stated that two systems are used to understand the risks, which are analytic and experiential systems. In the experiential system, the decision maker makes the decision based on past experience and current emotion. The analytic system uses logical algorithms.
and procedures to govern the behaviour. Epstein and Seymour suggested a synthesis of the two systems in order to rationalize decisions [70]. For a rational process, the researchers excluded emotion as an influential factor impacting the decisions. Even when researchers included emotion, such as [71], they considered the behavioural emotion of a decision maker as a positive or negative effect, without considering the actual emotion [65]. This consideration limits our ability to see the bigger picture concerning the effects of emotions and how they enter into decision-making processes. Gutnik et al. proposed a cognitive neuroeconomic framework that shares similarities with Loewenstein and Lerner’s work regarding the types of affects, but with added components, which are the sociocultural and environmental factors [65]. These factors combine to influence the temporal focus, in terms of the perceived risks and their probabilities.

Valence-based frameworks used to understand affect and its relation to cognition or behaviour in terms of making decisions has been researched through many studies. In 1995, Frogas found that affect can be entered into four different processes; namely, heuristic, direct access, motivational, and substantive [72]. The affects in this case are considered as information, and it is known as the affect-as-information model. The study investigated how, in heuristics for example, the lack of information about the affect plays an important role if the affect is experienced to a related object under decision in hand. Many other studies, in addition to the one by Frogas, have investigated the positive and negative influence of valence-based approaches. More precisely, Lerner et al. developed a framework for studying which positive emotions influence decision making more than others [11]. In other words, they provide a specific emotion framework to study the effects of that specific emotion on decision making. For example, fear and anger are both negative affects; but one has more influence on the decision-making process than the other.

A study by Pfister and Bohm provides a different perspective on affects and their
relation to the decision-making process [73]. They proposed a multifaceted emotion model based on different functions, arguing that emotions are heterogeneous regarding their role in making judgements. Their model classifies the emotions based on four functions. The first function is to offer information that the decision maker needs to make the decision. Experiencing previous decisions that feed this function as affect-as-information can influence this function. The second function is called speed and it reflects one of the situational or constant factors during decision-making activities, such as a time limit for making decisions. The third function concerns the associated influence of the decision at hand; i.e. expected emotion. The fourth function is the function of the emotions related to the commitment to making a decision.

Suppose we are in a decision-making situation while some empirical studies find out that decision makers have some negative affects or unwanted emotions that might influence the decision outcome. The relation of negative emotions to the accuracy of the decision outcome allows researchers to propose techniques for mitigating these negative emotions. Lerner et al. categorized the techniques into two major forms that fall under reducing the negative emotion or isolating them from the decision making process [74]. An example of the first form is the time delay between making the decision and the time that the affective states returns back to normal. The notion of time delay techniques is taken from the nature of the emotion. Levenson says that the emotion is “short-lived psychological-physiological phenomena that represent efficient modes of adaptation to changing environmental demands”, which, in turn, suggests that negative affect can be avoided when time passes [75]. Suppression and reappraisal are other techniques that fall under the first form, reducing negative emotion. For details, see [74]. Techniques under the second form include 1) crowding out affect by exposing the decision maker to a similar situation, 2) motivating the decision maker’s cognition financially, and 3) increasing the knowledge of the decision maker concerning the decision-making process [74]. The authors, based on a review of the literature,
found that the first two techniques were not promising, while the third one seemed more efficient. In addition, the authors mentioned that reframing or rearranging the structure of choices that appear to the decision maker is a useful technique that offers a promising solution for avoiding negative emotions. Synthesizing the previous issues of capturing and avoiding negative affects, as well as encounters with various characteristics of affects, into a framework is the most recent work conducted within the field of emotion and decisions [74].

2.4 Affects, Affective Contagion and Software Process development

In this section, we begin by investigating the literature on affects among developers and stakeholders, as part of the decision-making process, regardless of the specific software development process. Then, we open the scope of this investigation to include studies that take software processes into consideration. Since part of our ultimate goal in this thesis is to focus not only on developers’ affects but also all team members’ affects, including those of managers, stakeholders, and customers regarding making decisions, we iterate all related characteristics of affects concerning all team members in Agile environments who make decisions under different levels of abstraction. We chose to divide the literature on affects into two categories, namely developers’ affects and stakeholders’ affects.

2.4.1 Developer’s Affects

In this category, the literature on affects as related to developers’ productivity and performance is considered. Affects occur among team members in software engineering. The literature criticizes the studies that shows the effect of Agile processes on team members’ affective states. Finally, a comprehensive review of the effects of
different factors on team members’ affects is conducted.

Calling back tripartite system that is consist of affects, conation, and cognition; the software engineering field, like other fields, has been focused on human aspects related to the behaviour and conation of developers. Until recently, few studies investigated the influence of affects on developers in software research. Perhaps Graziotin et al. are the first researchers who investigated the relation between affective states and developers’ skills, such as creativity and analytical problem-solving skills that influence performance and productivity [1],[76], and [77]. The authors founded psychoempirical software engineering, including its background, theory, and guidelines [19], allowing them to provide their investigation of misconceptions about the affective states of developers with correct measurements [78]. Crawford et al. insisted on the need to draw a line between emotions and developers’ productivity [79].

Research on how emotions influence software developers’ performance has increased during the past decade. Graziotin et al. developed a theory of how affects influence programming performance [1]. They based their theory on two major pieces of ideas. The first idea concerns how affect impacts workers’ behaviour, their decisions, creativity, turnover/absence, and even pro-social behaviour, such as helping other workers [80]. The authors in [80] note that workers’ happiness and its implications regarding performance outcome is no longer measured by traditional means, such as job satisfaction, but now by positive and negative affects. They show, using various studies, the relation of emotions to workers’ performance as individual and groups, considering these new measures of performance including affects. For the second idea, they note the need to theorize developers’ performance from a behavioural perspective. Lenberg et al. proposed Behavioural Software Engineering (BSE) as an umbrella term that is related to any behavioural studies on topics including emotions [81].

Graziotin et al. used qualitative methods to develop a hypothesis using open-
ended interviews, observations, and email exchanges [1]. The on-site observations are intended to observe the effects of different factors, such as developers’ experience and perception, without any restrictions on practices, tools, or methods the developers used. The goal of the study was not told to participants for the sake of avoiding threats to the study. To analyze data, grounded theory was applied, which followed four phases of coding data: initial, focused, theoretical, and axial coding.

![Figure 2.1](image.png)

Graziotin et al. theorized in Figure 2.1 the building blocks of how work and non-work-related events influence affect and, subsequently, programming performance. They concluded that positive emotions have a positive impact on developer perfor-
mance in programming tasks, while negative emotions have a negative impact on developer performance in programming tasks. However, the scope of Graziotin et al.’s work is such that they only investigate developers’ programming tasks, and the test unit targeted individuals while no other artifacts, such as design or requirements, were considered. They call for further research to investigate different units of analysis, as [81] classified the units of analysis into individual, team, or organization. Moreover, managers, stakeholders, customers, product owners, and other software team members are not within the scope of Graziotin et al.’s study. Finally, although negative and positive emotions were recorded and mentioned in the study, only positive and negative impacts were considered, regardless of which specific valence, such as anger or joy, impacted the performance more.

Investigations of the influence of affects on creativity and problem-solving skills as aspects of workers’ performance have been conducted in the industry. For example, Amabile et al. investigated 222 workers in various organizations through a longitudinal study to explore the relationship between affects and “organizational incidents, subjective experience, work environment, and performance” [10]. The results of the investigation helped explain how affects influence creativity and problem solving. Therefore, Graziotin et al. investigated creativity and problem-solving skills as aspects of developers’ performance [76]. Forty-one software developers participated in the study to understand the relationship between the affective states of developers and their skills in terms of analytically solving problems and associated creativity. The authors used different measure for each task. For the problem-solving task, the Scale of Positive and Negative Experience (SPANE) questionnaire was implemented along with “version of the Tower of London task implemented in the open source Psychology Experiment Building Language (PEBL)” [76]. For the creativity task, the same SPANE questionnaire with six photos that have ambiguous interpretations was applied to measure the developers’ creativity. The results showed that there are
“indicators of higher analytical problem-solving skills,” while no evidence showed that creativity is related to affects. The authors called for more studies to be conducted on the correlation between emotions and developer creativity.

Earlier evidence showed that positive emotions influence programmers’ performance regarding the debugging task [82]. The authors conducted two experiments to investigate the relation. In the first experiment, 72 participants were engaged while emails and programming forums were used as mean to reach them. The experiment used a questionnaire associated with instructions on how to conduct the study. In the second experiment, a physical exercise intervention was used to increase arousal while showing a clip to increase the arousal. After each intervention, the participants applied the debugging task. The results of the first experiment showed that the performance of programmers was affected by arousal, while the second experiment confirmed the relation by changing the mood through different intervention methods.

Kosti et al. took a first step in studying the applicability of designing tools while considering developers’ emotions and taking advantage of them [83]. They insisted that if people cannot infer developers’ emotion and profit by enhancing project outcomes, then designing tools is not applicable. For that reason, Murgia et al. investigated reports from 117 projects from the “Apache Software Foundation issue tracking system” [83]. Sadness, joy, and love are three emotions highlighted, suggesting the visibility of inferring and mining emotions in software artifacts. However, researchers admit that further and more detailed studies are needed, as well as paying attention to various data sources and systems.

Following suggestions made by [79], Graziotin et al. correlated between programmers’ self-assessed productivity and their emotions [77]. They based their investigation on a similar study conducted by [84], which investigated the correlation between real-time emotions and productivity in some workplaces, such as office workers and daycare workers. Considering the differences in software development, Graziotin et al.
investigated the correlation in the software workplace. Their hypothesis, based on the literature, was that there is a positive correlation between programmers’ self-assessed productivity and their valence, arousal, and dominance. The SAM questionnaire was used every 10 minutes during a 90-minute programming task. Eight participants, including students and professionals, were engaged in the empirical study while 72 measurements were made. In addition to the SAM questionnaire, two interviews, pre-task and after-task, were conducted with each participant. Pre-task interviews collected data about participants’ experience in programming language, experience in assigned tasks, and whether the participant was professional or a student. Post-tasks interviews confirmed observations with participants to clarify any irregularities during any time interval. As a result of this study, arousal and dominance dimensions in affective states were proven to positively influence self-assessed productivity when they are high. However, no evidence was found to support the hypothesized effect of arousal on productivity. The authors suggest this may have been because participants misunderstood the questionnaire instructions.

Which emotion influences developer productivity, the frequency of that emotion occurring, the way that the emotion influences productivity, and the influence of risky emotions on productivity were investigated by Wrobel [85]. Surveys were used to collect quantitative data by means of questionnaire and interview. A total of 12 selected emotional states were introduced to participants, while 49 participants from different cultures completed the survey. The author found that 90% of participants experienced only anger and frustration as negative affects with many positive states. Pleasure, contentment, enthusiasm, and optimism were found to be the most common emotions among developers. Depressed and disgusted were the two negative affects that influenced productivity, while enthusiasm, happiness, and pleasure were the most positive affects to influence productivity. These result support Graziotin et al.’s empirical research on happiness [76] and [77]; however, other affective states,
such as feeling enthusiastic, were not empirically proven or disproven regarding the software industry. Finally, frustration was found to be the most risky affective state in relation to developers’ productivity and their programming frequency. Disgusted, disappointed, angry, and depressed were also found to be risky affective states, coming after being frustrated.

In line with previous studies on emotions and developers’ productivity, Müller and Fritz tested the correlation by applying biometric sensors [86]. They classified developers’ emotions by acquiring them through biometric sensors while changing the developers’ programming tasks. Their research was intended to find out about three main topics: 1) how developers’ emotions change during different tasks, 2) what changes developers’ emotions, and 3) if it is applicable to use sensors to measure developers’ emotions and progress. A total of 17 developers participated in this study, where each participant changed between two tasks and each task lasted for 30 minutes. The data gathered from sensors were entered into a classifier to distinguish between different affective states. A total of 213 emotions were identified as well as two emotions per participant, by showing a set of pictures to induce positive and negative emotions. They found similar results to the findings of Graziotin et al. regarding the correlation between emotions and productivity, in addition to the relevance of understanding code to introducing or changing positive or negative affective states. Some of the results agree with Wrobel’s investigation [85], which found that negative emotions sometimes work as motivators to solve complex problems. Results concerning the relation between biometrics and cognition led Müller and Fritz to study how they can predict and enhance the quality of codes in order to reduce costs and increase code understandability[87].

Ford et al. investigated the influence of frustration as an emotion on developers regarding their experience in learning programming [88]. The study surveyed 45 developers as a first step to framing their intervention related to the influence
of emotions on learning programming. Around 30 developers considered frustration as a major problem. A total of 11 categories were found by researchers to cause frustration, including mapping behaviour to its cause, programming tools, goal size, programming experience, and lack of resources. The authors call for more comprehensive investigations of all 11 categories in order to enhance experiential learning and avoid negative emotions. They also emphasize the need to consider programmers’ behaviours as a factor that, when understood, can prevent developer frustration.

Up to this point, we have only mentioned studies that focus on individual developers and tasks; however, a few studies have taken into consideration the developers as a group, while others have considered the effect that underlying software development processes can have on team behaviours.

The relation between the software development process and developers’ emotions has been examined in both directions. First, the effects of the process practices, procedures, organization, and environment on developers’ emotion have been studied. Using a second type of study, researchers have investigated the effects of developers’ emotions on the success of software process practices and procedures. Notice that the relationship is bidirectional, where emotions might influence a development practice or procedure, while the rules and regulation concerning such a practice can change developers’ affective states.

Under the first type, Omar et al. provide empirical evidence concerning whether Agile processes influence developers’ cognition, including emotions [89]. A total of 67 students were randomly divided into 16 groups. Half of the groups were assigned to a formal process, to carry out the project as the control group in the experiment, while the other half were assigned to follow the Agile process. The experiment was designed to measure differences in the “work-related well-being level” between the formal process groups and the Agile process groups [89]. The researchers hypothesized that teams engaged in the Agile process would experience a higher level of enthu-
siasm, contentment, and well-being as affective states, while they would experience lower levels of feeling depressed and anxiety as affective states. Furthermore, the experiments took into account the relation between applied Agile practices and their correlation with previously mentioned affective states. No significant differences were found between the two groups of teams; hence, the first hypothesis was rejected. The other hypothesis suggested that number of Agile practices would be positively correlated with enthusiasm and well-being; while they would be negatively correlated with anxiety and depression. No evidence was found to support a correlation between the number of Agile practices and positive contentment as an affective state. Motivation and leadership were two variables that the author proposed to investigate in terms of this hypothesis and to define the correlation between developers’ psychological aspects and performance. We found that the findings of Omar et al. were in line with earlier research conducted by Syed-Abdullah et al. [90], in which similar hypotheses and propositions were considered; therefore, it seems that repeated empirical studies have proven the influence of XP on emotions.

A longitudinal investigation was conducted by Syed-Abdullah et al. to highlight the effects of the XP methodology on developers’ moods [91]. The authors hypothesized that the temperament of developers would be higher when they apply XP practices as compared to the temperament of others who do not. The experiments’ participants were students who were divided into two groups to develop six projects. PANAS was used to capture teams’ temperaments during the development cycle. Data analyzed by T-test proved, in line with [89], that the XP methodology positively influenced software teams’ temperaments. A more interesting finding was that as the number of used XP practices increased, the software teams temperament increased also. However, no specific definition of temperament is found in this study; therefore, the study supports the general notion of the positive effects of XP on developers’ emotions. A similar study used the PANAS questionnaire during two different
intervals to measure the effects of the Agile method, XP, on increasing developers’ positive affective state \[92\]. Four teams of developers at a university computer science center participated in projects lasting eight months, where each team consisted of four programmers and two system analysts. Results showed that the level of positive states experienced was higher in the second interval than during the first interval, although not all XP practices were completely applied by the teams. The results suggest applying the flexible Agile method to cope with the organization, environment, and culture to increase developers’ emotions and, subsequently, improve software quality.

Under the second type, Islam and Zibran investigated whether emotions were different when software developers executed different development tasks, such as bug-fixing and refactoring \[93\]. The authors wanted to determine if the emotions influence software artifacts, in order to measure and avoid negative emotions and their implications for the associated tasks. They analyzed commit comments, produced by 30 developers, and used a sentiment analysis tool to investigate the problem. They found negative emotions were higher in new task features, while positive emotions were higher in refactoring and bug-fix, and positive and negative emotions were equal in “energy-aware tasks” \[93\]. In addition, researchers investigated whether groups of developers could be distinguished according to their affective state. Developers with higher levels of positive emotion can be assigned to particular tasks to increase team effectiveness. This point supports Dingsøyr et al. and their statement concerning the need for more studies to support team effectiveness models \[11\]. Bug-fix is the task where a group of developers were classified based on their experience of either positive or negative affects. Moreover, \[93\] objective was to examine changes in developers’ emotions during different days of the week and at different times of day. No evidence of significant variations in emotions for different days and times was found.

More than 1300 forum posts from five teams of students through three sprints in SCRUM were analyzed and studied by Marshall et al. \[94\]. The main goal was
to find a relationship between performance and emotions regarding software artifacts produced during the development process, such as forum posts. The authors used SEREBRO, a management tool for writing posts in daily scrum meetings, as well as some metrics for measuring developer productivity. Each team in the experiment was required to post three types of information so that their performance could be measured, which were 1) what has been done since the last post, 2) what is going to be done in the next post, and 3) what obstacles were faced between the last post and the current post. Emotions were extracted manually using different emotion analysis techniques. Generally, posts were classified as belonging to one of four categories: positive emotions, negative emotions, unemotional, or neutral emotions. Results showed that unemotional posts were more common than emotional posts.

Regarding the application of the sentiment analysis tools to software engineering, Robinson et al. investigated the correlation between developers’ sentiments and behaviours including, more precisely, routine changes [95]. The study built its investigation on 124 projects extracted from the GitHub website. Tracing developers’ behaviours, the authors monitored how frequently developers engaged in a specific sequence of actions. Jurado and Rodriguez suggested the possibility of applying sentiment analysis techniques to developers’ artifacts, such as ticket issues [96]. They investigated more than 10,000 issues from the GitHub open source projects repository. They concluded that sentiment analysis techniques can be used to effectively monitor software processes.

Guzman et al., in line with [93], [95], and [96], analyzed GitHub open source commit comments provided by project developers using lexical analysis to find out whether or not there is a relationship between emotions and programming language, time of day, day of week, geographical locations, and project approval [97]. More than 60,000 commit comments from 90 projects were lexically analyzed. Results showed that negative emotions were related to projects developed using the Java
programming language. The comparison of negative emotions took place regarding six other programming languages, while the Java language received the highest number of negative emotions. Regarding day and time, Mondays tended to be associated with more negative emotions than other weekdays, while afternoons tended to be associated with more positive emotions than evenings. No correlation was found between emotions and different locations. However, the analysis found that there is a relation between positive emotions and the number of countries that consists the distributed teams in the form that as a project includes team members from more countries, the positive emotions increase.

Guzman et al. analyzed collaboration artifacts in order to identify emotions and study their effects on software development teams [98]. The researchers proposed a technique for gathering information from various artifacts, such as code, emails, issues, and comments, then analyzed emotions using lexical data analysis techniques after grouping the emotions into topics, in order to provide a summarized representation during software development. Three teams participated in the study, which took three months to develop. The authors argue that the model helps in various ways, such as enhancing decision making. To strengthen the authors’ viewpoint concerning the necessity of their model, they conducted interviews with team leaders. Results showed that 66.6% of team leaders could find the correlation between positive and negative emotions in artifacts.

Guzman used the previous results in [98] to visualize them and allow teams to cope with monitoring emotions [99]. The visualization depicts the awareness of developers regarding their emotions during the development process. The visualization tool was validated through a case study that lasted for 3 months, and 3 teams were included. A total of 354 emails were exchanged by developers, which were analyzed and visualized. The results show that positive emotions were much higher in the beginning of project phases, such as when gathering requirements and analysis, while negative emotions
were much lower. When a deadline approached, Guzman found that not only did the number of emails increase, but also the diversity of emotions. Based on this, Guzman hypothesized that emotional diversity increases when deadlines approach, visualizations helps in remembering past experience from artifacts, and increasing the accuracy of computing emotions helps in applying trusted results shown through visualization tools. All three hypotheses are now under investigation by Guzman.

Mäntylä et al. searched through mined artifacts to investigate developers’ Valence, Arousal, and Dominance (VAD) [100]. The authors analyzed a data set containing 700,000 issue reports from a thousand projects that are open sourced. The authors investigated VAD’s involvement and relationship to report issues, as well as the possibility of VAD being able to measure the length of time that bug-fixing requires. The results of this investigation showed that arousal level was lower in relation to trivial report issues, while its increased with critical issues. Valence was found to be at its lowest level in bug fixing. The authors suggested that a short time for resolving issues was related to high dominance. Unexpectedly, dominance was found to be high with a longer resolution time for issues. Generally, during the issue lifecycle, valence and dominance were found to be at an ascending level while arousal was found to be at a slightly decreasing level.

2.4.2 Stakeholder’s Affects

Regarding this category, few studies have focused on the role played by affects in relation to stakeholders and customers during their involvement in the Agile process. Requirement Engineering, as one of the most important phases in software, requires interaction with stakeholders. Colomo-Palacios and colleagues investigated tasks related to knowledge, such as requirement engineering and its relation to developers’ and stakeholders’ emotions, which created an opportunity to study the effect of emotions in software engineering [101]. The authors characterized requirements
within the process and over different times. For each time, stakeholders’ emotions were reordered based on requirements to come up with a pattern of their emotion for all times. The authors hypothesized that higher pleasure and lower arousal among the stakeholders would be correlated with higher requirements scores. A total of 137 user requirements from different versions were collected from two different and real projects that lasted 6 and 7 months. Participants were asked to answer the following question: “What’s your emotion regarding this requirement definition?” \[101\] using the Affective Grid tool \[102\]. The researchers concluded that requirement stability is highly related to the emotions of both stakeholders and developers. To understand how customers’ experiences and emotions are affected in software engineering, we highlight the findings of Forlizzi et al. who investigated customers’ emotions in relation to new design products \[103\]. They proposed a framework that considers short and long-term experiences of emotions regarding new products. This experience of emotion is influenced by the environment that defines the interaction between the product and the customer, with the customer’s goal in mind. Later in this thesis, we highlight our finding of how stakeholders and customers experience emotions and how this influenced changing requirements in software products.

2.5 Affects, Affective Contagion and Making Decisions in Agile Environments

Sharp et al. investigated motivation among developers and managers in software process as linked to emotions \[104\]. The investigation sought to highlight the difference between what motivates developers and what motivates project managers. The authors collected data from four developers and six managers by questioning them in a workshop. The results showed that problem-solving, creativity, learning, and people are the highest motivators for developers. On the other hand, problem-solving,
people, and influence were found to be the only motivators for project managers. An interesting finding in the study was that managers’ behaviours to motivate developers are based on their perception of the best motivators (problem-solving, people, and influence), while this is not exactly the case on the developers’ side, where creativity and learning would be more effective motivators. However, motivation, job satisfactions and other terms are excluded by Graziotin et al. [178], although these are, along with other characteristics, explained more in 2.6.

Investigating decisions made by teams in programs, projects, and portfolios was the main purpose of a study conducted by Cunha et al. [105]. During the decision-making process, decision makers, according to [105], are not willing to enhance their decisions. They found “cognitive biases, tools and techniques, emotions and organizational factors” were the main contributors to enhance the decision-making process and, as a result, product outcomes. However, the authors did not mean to investigate how emotions play a role in enhancing the decisions, as they focused on cognitive factors. Nonetheless, we believe emotions play an important role, based on the general study of Barsade, which states that positive emotions not only influence group behaviour, but also “the positive emotional contagion group members experienced improved cooperation, decreased conflict, and increased perceived task performance” [15]. No evidence has been found of such implications of contagion on software teams in terms of cooperation, conflict, or performance, which is one of this thesis’ objectives.

Two relevant studies conducted by Colomo-Palacios et al. investigated difficult decisions in software projects and the role of affects in these decisions [16]. In the first study, the researchers aimed to collect from expert developers and managers the most difficult decisions they made during the software development process. The Delphi method was used to collect, aggregate, and analyze data to determine the top five difficult decisions. A total of 25 experts, with average experience of 17 years, parti-
pated in the study. The results of this investigation listed 96 difficult decisions which were classified into 11 classes of difficult decisions. Only the first five difficult decisions were entered into the second study, in order to investigate the influence of emotions on the decision-making processes. The five difficult decisions were “project prioritization, requirements prioritization, development strategy election, partner-supplier election and internal personnel selection” [16]. The authors aimed to extract the exact emotion related to each ranked category and determine which categories were more prone to emotional influence. For that reason, 30 developers with experience were given the five ranked categories and asked which emotion was more influential in each category and the category most influenced by emotions. This was done by conducting semi-structured interviews with the developers. Participants in the second study were given only four basic emotions to choose from, which were happiness, anger, fear, and sadness. The collected data were analyzed using NVIVO software. Results showed that negative emotions, such as frustration and resignation, were the emotions listed as highest in all categories by participants. In addition, requirement prioritization was influenced by emotions even considering the availability of many requirements techniques. Finally, the authors highlighted the necessity of having trained managers to consider, manage, and deal with developers’ emotions.

However, although the work of Colomo-Palacios et al. is interesting and their results are valuable, some shortcomings are evident. First, their investigation was based only on experience, rather than measuring the actual emotion during the real time of the decision-making process. Second, categorizing decisions might allow the interviewed expert to think only about one situation that falls under that category, and as a result researchers might elicit inaccurate emotions. Third, the emotions that come from various sources are neglected; rather, the researchers investigated only the emotions that are caused by making the decision. Xu and Wang proposed a framework that includes the influence of not only expected and incidental emotions
but also immediate emotion on the intelligent behaviour [9]. They considered all types of emotions as well as various triggers for the emotions in order to enhance the decision-making process even under uncertain decisions. Most of the decisions in software development projects are made under uncertain and risky conditions in all categories listed by Colomo-Palacios et al.; however, it seems that the researchers only considered one type of emotion, expected emotions, in their investigation.

2.6 Measures and Guidelines for Studying peoples’ affects and its contagion in Software Engineering

This section discusses the need to clarify concepts, measures used, and guidelines for conducting empirical research regarding affects in software engineering. Although the goal of this research is to investigate the correlation between Agile teams’ affects and their decisions made during software development, conducting experiments in this regard intersects with Graziotin et al. regarding clarifying misconceptions and applying correct psychological measures.

To clarify what falls under the umbrella of affects regarding previous studies, Graziotin et al. insist that job satisfaction and commitment, well-being, motivation, and even happiness are misunderstood in software engineering [78]. They are related to affects, but they are not considered to be affects. They built their argument by importing various definitions and experiments from psychology, behaviour, engineering and other fields.

In software engineering, based on the literature, emotions can be extracted in three ways: 1) from software artifacts, such as the experiments conducted by [82], [83], [93], [94], [98], and [100]; 2) from developers’ experiences of their past emotions during project development, such as the experiments conducted by [85] and [88]; or 3) direct measuring of developers’ emotions through real-time projects such as was
done by [1], [77], [77], [91], and [101].

For extracting the emotions from artifacts, various sentiment analysis tools can be used, which led Jongeling et al. to investigate which sentiment tools are more valuable in software engineering [106]. More precisely, the authors tested the suitability of sentiment tools for extracting developers’ emotions and investigated the possibility that using different sentiment analysis could provide consistent or contradict results. Four sentiment analysis tools were compared against each other, including SentiStrength, Natural Language Processing (NLP) Stanford sentiment analyzer, Alchemy and Natural Language ToolKit (NLTK). Results showed that using such a tool can lead software engineering researchers to obtain different results than if they use another tool. They advise the use of more accurate tools, based on training data, in order to allow researchers to obtain trusted results. One of their suggestions is the use of the Support Victor Machine (SVM) environments, such as Weka data mining [107].

For the direct measuring of developers’ emotions through real-time projects, Graziotin et al. recommend the use of either SPANE (and its extensions, such as SPANE-B) or SAM questionnaires in software engineering. SPANE can be used to measure developers’ affects regardless of which stimuli causes the extracted affects to appear, increase, or decrease. On the other hand, SAM is suggested for use when the researcher wants to measure developers’ affects while also paying attention to the stimuli. They suggest the use of core affects to measure software developers’ affective level.

Since there are no guidelines for conducting this type of research, including how to choose the right psychometrics, Graziotin et al. provided researchers in this fields with general guidelines for conducting “Psychoempirical Software Engineering” [19]. First, it is suggested that researchers must pay attention when framing their research and specify what exactly they want to measure. For example, researchers must specify if they want to measure affects as well as their stimuli, dimensions, time of occurrence, etc. Second, choosing the right instrument for measurement is important.
This means that researchers must choose validated instruments from psychology to conduct software engineering experiments. Graziotin et al. insisted that changing instruments, as Colomo-Palacios et al. did in their experiment [16], can influence the results, because changing instruments requires new psychological studies to validate the switched instruments [19]. Third, instructions for applying instruments must be followed correctly and efforts should be made to ensure that developers’ responses during the experiment are not influenced. Finally, a strong analysis must be conducted with a strong understanding of the collected data, to apply the right test and acquire correct results.

Finally, considering emotions under the concept of behavioural software engineering allows us to include the results of Lenberg et. al. regarding “units of analysis” [81]. They analyzed, through a systematic review, studies that focussed on human aspects in software engineering. They concluded that individuals, groups, and organizations are the three types of units of analysis. Based on the literature, most studies of affects fall under the individual unit of analysis, while only a few studies focus on group emotions (such as in the Agile team) or organization emotions (such as how cultures impact emotions differently) in software engineering. In consideration of this gap, this thesis considers the study of affects in terms of the group unit of analysis.

2.7 Chapter Summary

- The chapter discussed most of the empirical studies conducted regarding the Agile team’s cooperativeness, conflicts, decision making and how decisions are made, factors that influence decision making, and studies focusing on affect within the industry.

- First, the chapter highlighted studies that tried to understand the cooperativeness in Agile environments, positive and negative factors that influence cooper-
ativeness between team members, and the influence of cooperativeness on Agile process outcome.

• Second, the chapter highlighted studies that tried to understand the conflicts in Agile environments, positive and negative factors that influence conflicts between team members, and the influence of conflicts on Agile process outcome.

• Third, the chapter highlighted studies that tried to understand the process of making decisions in Agile environments.

• In section two, the chapter defined affects, emotions, and moods, and their categorization as affective states, as part of the tripartite classification of the human mind.

• The third section discussed studies that considered affect and its contagion as an influential factors related to decision making.

• The fourth section highlighted most of the empirical studies that investigated the role of affect and its contagion within software processes.

• The chapter finalized the discussion of affects by highlighting the correct psychological measures and rules to use in order to successfully conduct empirical research within software engineering.
Chapter 3

Research Framework

The main goals of this chapter are to highlight the research questions, develop the research hypotheses, and form conceptual models to be tested throughout the research activities. Two models, for the affects and affective contagion, are described, as well as a third model, which is an aggregated model that highlights the correlations between affects and affective contagion. The chapter ends with a description of the control variables that can influence the results when testing these models.

3.1 Research Questions

The main research objective is to investigate the following research question:

RQ: How are affects and their contagion related to decision making in Agile environments?

The main question of the thesis has emerged from the intensive reading of the literature. During the intensive process of highlighting the research gap, I have highlighted the main elements from the current literature that can help me answer the main research question. As a result, I was able to develop sets of hypotheses and conceptual models based on investigating different psychological, economic, and engineering studies. These hypotheses mainly investigate confirming the role of affects
and affective contagion in Agile decisions and related factors such as cooperativeness and conflicts among Agile team members. Different sets of hypotheses, along with their conceptual model, are developed, as explained in the next section 3.2, to provide us with the building blocks for answering the main research question.

The main research question, then, can be answered by breaking it down into the following questions:

RQ1: How do Agile decision makers experience affects?

Experiencing affect within the Agile decision-making process is an important issue to know. The first step to understand the influences and roles of affects on Agile decision makers is to understand how they conceptualize the affects and under which conditions they conceptualize them. The second step for answering the question is understanding how the conceptualized affects are currently dealt with in Agile teams. The affective contagion is also part of conceptualizing the affects while this thesis seeks to know how the spread of affects among team members are currently experienced and dealt with within the Agile teams. We consider both types of decisions; namely, individual and team decisions to highlight the ways that Agile team members conceptualize and respond to affects and their contagion during decision making.

RQ2: How do affects and their contagion influence the accuracy of decisions in Agile?

Direct and indirect affects and their contagion influences on the individual and team decisions are assessed within this question. The direct affects can be related more to individual decisions while indirect influences can be seen through team decisions. The question also tests the contagion of affects and their impact on the overall level of decisions quality. Further, the questions test the influences of affects on decisions that have been highlighted in different managerial and industrial literature.

RQ3: How do affects and their contagion influence cooperativeness between Agile team members?
This question seeks for understanding both direct and indirect influences of positive and negative individual’s affect on the level of cooperativeness. Further, the research question investigates whether or not individual’s affect are contagious and influence the overall level of cooperativeness of Agile teams. The question is related to the main research question because the literature has highlighted a relationship between the level of cooperativeness and the accuracy of decisions. Based on the literature, this thesis hypothesizes that affects and their contagion influence the level of cooperativeness; while this research question seeks beyond highlighting the influence by investigating in-depth how that influence is being conceptualized, experienced, and dealt with within the agile team.

RQ4: How do affects and their contagion influence conflicts between Agile teams?

Similar to RQ3, this question seeks to understand both direct and indirect influences of positive and negative individual’s affect on the level of conflicts among agile team members. Further, the research question investigates whether or not individual’s affect are contagious and influence the overall level of conflicts among Agile team members. The question is related to the main research question because the literature has highlighted a relationship between the level of conflicts and the accuracy of decisions. Based on the literature, this thesis hypothesizes that affects and their contagion influence the level of conflicts; while this research question seeks beyond highlighting the influence by investigating in-depth how that influence is being conceptualized, experienced, and dealt with within the Agile team.
3.2 Model Formation and Hypotheses Development

3.2.1 The Relationship Between Affects, their contagion and Quality of Decisions

Briggs and Little studied the cultural and personality traits that exist in software companies when decisions are made [12]. Their advice was to pay attention to managers’ skills, organizational character and culture, and peoples’ behaviours. Although organizational culture and other factors are mentioned in other studies, these authors uniquely highlight the need to be aware of people’s psychology as well. They insist that, “the significant differences in type between members of the decision team manifest themselves in clear behaviours that the technical manager can identify and guide the team through. Depth of analysis and the tendency to extrapolate ahead to the likely outcome are common tensions here. The leader can identify these tendencies for the team and develop compromises that retain buy-in while containing time and energy expenditures” [12].

Being unsatisfied with management, a lack of involvement and influence, insufficient relationships with team members, and boredom are some of negative factors highlighted by Deak et al., who interviewed 34 testers from 12 software companies [108]. The other factors that Deak et al. highlighted include time pressure, environment issues, and technical issues faced during the testing phase [108]. We infer from the two previous studies that behaviours or affects can play important roles in various phases, such as in analysis, design, and testing. However, these studies did not investigate factors and obstacles in the real environment, such as by observation or other means. Whether or not the underlying environment itself influences decision making was the focus of an investigation conducted by Zannier and Maurer [109]. They compared Agile and non-Agile teams in three companies. They concluded that Agile processes enhance the decision-making process more than other processes. They
insisted that the reason for this is because Agile teams communicate more, while other process teams tend to communicate less.

In line with Briggs and Little [12], Cunha and Moura also investigated organizational factors in terms of making decisions, as well as other factors, including tools and techniques, emotions, and cognitive biases [105]. These factors were found to be the dominant factors affecting decision making at all governance levels. Cognitive biases, tools, and techniques influence portfolio and strategic decisions. Program and tactical decisions were found to be more heavily influenced by organizational factors, but less so by cognitive biases, tools, and techniques. Finally, project and operational decisions are influenced by emotions more than tools and techniques.

Drurya et al. investigated obstacles to making decisions in Agile environments [110]. The authors first extracted all of the decisions that developers made during four different phases, using focus groups involving 43 developers and managers. Decisions from all four phases of development were categorized into tactical and strategic decisions. Six main obstacles preventing effective decision making were highlighted as a result of the study, which are: “1) people are unwilling to commit to a decision, 2) conflicting priorities, 3) inconsistent resource availability during sprint, 4) decisions are not implemented, 5) lack of ownership, and 6) lack of empowerment” [110]. These results certainly suggest possibilities for dealing with the obstacles that Agile teams face when making decisions. However, the study failed to investigate obstacles among actual team members. The authors noted that such a study, where the decision-making process is directly observed, could provide a better understanding of all obstacles faced by Agile team members.

An investigation conducted by Drury-Grogan and O’dwyer used interviews and observation to explore the factors that influence decision making in Agile [111]. The authors considered Scrum as the main process in order to investigate daily meetings as well as sprint planning meetings. A total of 34 developers in four teams were
interviewed, while 19 observations were captured and analyzed accordingly. The researchers found that the three main factors that influence Scrum decisions are experience, sprint duration, and resource availability. The investigation helped us to develop the assumption that some factors in Agile processes can influence developers’ affective states, which, in turn, influences the Agile decision quality.

Based on the aforementioned evidence, the general hypotheses for considering the role of affects and their contagion on Agile decisions, along with the correlation between affects and affective contagion, are represented as follows:

H1: The affects of Agile teams have a significant influence on the accuracy achieved in the Agile decision-making process.

H2: The affective contagion between Agile teams have a significant influence on the accuracy achieved in the Agile decision-making process.

H3: There is a correlation between positive affect and positive affective contagion in Agile decision making insofar as an individual’s affect positively influences the affective contagion between team members.

H4: There is a correlation between negative affect and negative affective contagion in Agile decision making insofar as an individual’s affect negatively influences the affective contagion between team members.

3.2.2 The Relationship Between Affects, their contagion and Cooperativeness

Shore et al. have highlighted the problem of people communicating ineffectively as being a main contributor to the development of poor software [18]. Moods, motivations, and desires are three factors that the authors suggest focusing on in order to overcome poor communication. Ensuring these factors are adequately supported can lead to the enhancement of the Agile process through improved communication, as this enhances the cooperation between developers. Furthermore, cooperation, along
with other individual and social factors, influences the development process [37]. The researchers stated that cooperation between team members can be maximized by maximizing the social connections among individuals, which, in turn, increases “the enthusiasm of teams and individuals to cooperate in a voluntary manner” [37]. Despite making this point, the researchers do not highlight the link between cooperation and personal factors.

A study conducted in 1998 found that emotional stability is a significant factor related to team (but not individual) cooperation, coordination, stability, and the overall social interaction of the team [39]. The authors stated that, “Although agreeableness and emotional stability have not been found to consistently relate to performance for individuals, our results suggest that they (along with extraversion) can be important team-level constructs that predict team performance and team viability” [39].

A series of four studies was recently conducted by Motro et al. that successfully found correlations between emotions—more specifically, anger—and team cooperativeness [112]. The authors hypothesized that the anger of team members would decrease the level of cooperation between team members insofar as, if two partners were angry, the possibility of cooperation between these two members would decrease. The anger discussed in these four studies represents both integral anger (caused by negotiations between the two team members) and incidental anger (caused by factors not related to work). These studies have shown that, “in a single-round cooperation task, angry individuals paired with other angry individuals cooperate less than do other pairings of angry and neutral individuals” [112].

Considering the fact that Robertson et al. in [34] and Huhns in [35] discuss how cooperation can be enhanced when each team member believes that he or she benefits and/or their self interest is increased; this belief can be considered a construct that is related to different factors, one of them being his or her affective state. For example, when a team member benefits from cooperation in Agile but he or she does not feel
that way, feeling disappointed, distressed, sad, etc. can work as barriers preventing
the formation of that belief. Therefore, although the benefits exist for the Agile team
member, these barriers can reduce the level of cooperation in Agile.

Based on the aforementioned evidence, the specific hypotheses for considering
the role of affects and their contagion regarding Agile teams cooperativeness, along
with those hypotheses related to considering cooperativeness in Agile as a mediator
between affects and their contagion and decision quality, are represented as follows:

H1.1: The positive affects of Agile teams have a direct positive significant influence
on team cooperativeness.

H1.2: The negative affects of Agile teams have a direct negative significant influ-
ence on team cooperativeness.

H2.1: The positive affective contagion between Agile teams have a direct positive
significant influence on team cooperativeness.

H2.2: The negative affective contagion between Agile teams have a direct negative
significant influence on team cooperativeness.

H5: Cooperativeness between team members mediates the influence of positive
and negative affects on Agile decision quality.

H6: Cooperativeness between team members mediates the influence of positive
and negative affective contagion on Agile decision quality.

3.2.3 The Relationship Between Affects, their Contagion and
Conflicts

A recent study conducted by Jiang et al. in [42] showed that emotional conflicts
in distributed software development teams have a strong negative influence on team
members’ satisfaction and their work efficiency. The researchers highlighted that the
reason for this is the lack of communication between distributed teams. The results of
this recent study highlight the need to further study the role that emotional conflicts
play even when team members communicate more often. As mentioned earlier, Agile requires communication, and, according to Jiang et al., increased communication between team members will eliminate the negative effect of emotional conflicts on developers’ satisfaction.

However, we think that the lack of communication itself cannot be the main cause of emotional conflicts; for example, the affective states and their contagion that exist between teams during communication and decision-making processes can have a strong negative influence on emotional conflicts as well as on developer’s satisfaction. Therefore, communication itself can be either a positive or a negative factor in relation to all types of team conflicts.

Considering the negative emotion as a construct and integral component of developers’ and users’ “perceptions of interpersonal conflicts in software development” as stated by [44], we need to highlight these emotions and the specific types that increase the effect on interpersonal conflicts and, hence, the project outcome. The authors in [44] concluded, in contrast to the literature on conflicts, that interpersonal conflicts can negatively influence the project outcome even with the application of conflict management. Although this study was well done and offers a thoughtful discussion, the authors considered emotion in terms of only two emotional states—namely, frustration and anger—and they did not use any reliable measures, such as PANAS or SPANE, to measure the negative emotion. Including a reliable psychological metric could have positively influenced the measured model and highlighted the correlation between interpersonal conflicts and project outcome in a more reliable way.

Based on the aforementioned evidence, the specific hypotheses for considering the role of affects and their contagion on Agile teams conflicts, along with considering the conflicts in Agile as a mediator between affects and their contagion and decision quality, are represented as follows:

H1.3: The positive affects of Agile teams have a direct positive significant influence
on team conflicts.

H1.4: The negative affects of Agile teams have a direct negative significant influence on team conflicts.

H2.3: The positive affective contagion between Agile teams has a direct positive significant influence on team conflicts.

H2.4: The negative affective contagion between Agile teams has a direct negative significant influence on team conflicts.

H7: Conflicts between team members mediate the influence of positive and negative affects on Agile decision quality.

H8: Conflicts between team members mediate the influence of positive and negative affective contagion on Agile decision quality.

3.3 Conceptual Research Model

3.3.1 The Model of Affect Influences on Cooperativeness, Conflicts, and Decision Quality

Based on these research hypotheses, see the model in Figure 3.1 created to illustrate the potential influence of an individual’s positive and negative affect on an Agile team’s cooperativeness, conflicts, and decisions. Each ellipse represents exogenous variable, as explained in SEM [5], that is influenced by factors outside the model we intend to test, such as work related factors. Rectangles represent endogenous variables that are influenced by endogenous or exogenous variables. Each arrow represents either positive or negative influence as hypothesized in this thesis.
This thesis hypothesizes that an individual’s positive affect can play a role with regards to cooperativeness in Agile teams, as well as with regards to the quality of the decisions that the individual participates in decisions. Although good communication and team cooperation are very important and considered important factors for the success of Agile methods, an individual’s affects can trigger the development of such good communication and cooperation between Agile team members. Therefore, we highlight that an individual’s affect is the central motivator for increasing the level of cooperativeness in Agile.

An example of the direct influence of positive affect can be seen in the pair programming practice in XP, where two programmers work on one machine [33]. Many researchers have highlighted the factors that can lead to better or worse cooperation between the pair. Recently, Alsheri et al. provided a methodology to decide which factors should be considered when matching the pair in XP [113]. The authors
concluded from the literature that factors that must be considered when matching experts and novice programmers in XP are: “Speed: pairs with the highest chance to accelerate the coding practice, Sharing Knowledge: pairs with the highest chance to exchange knowledge, Code Quality: pairs with the highest chance to improve code quality more, Learning: pairs with the highest chance to foster a training and learning environment” [113]. Of these factors, speed, sharing knowledge, and learning are exposed to the influence of affects. Thomas et al. conducted a study that concluded that programmers with high confidence and skills in programming like pair programming the least, while programmers with less confidence and skills like pair programming the most [114]. This evidence supports the idea that some affects play a role in people-centric processes such as Agile methods.

An individual’s positive affect can also play a role in decreasing the level of conflict between team members. An individual’s positive affect can partially participate in the increasing of conflicts, especially with a small number of team members. Suppose that in a team with three developers, two team members’ affects are positive while the third developer’s affect is negative. The possibility of conflicts with regards to the affects is different than, for example, if there were a team with six developers who had negative affects and only one developer with positive affect. In this model, however, we are not discussing the spread and contagion of the negative affect to those with positive affect or the opposite as such; instead, we are discussing the role played by the positive and negative affect of individuals in increasing or decreasing the level of conflict.

Similar to cooperativeness, the quality of decisions can be positively influenced by an individual’s positive affect. Most of the decisions in Agile are still made by either managers or expert developers, while the novice developer’s opinion might be neglected. Many authors have investigated how to incorporate all team members into decision making. For now, we are only focusing on the decision maker’s affect,
regardless of how many team members are involved in that decision. The positive affects of individuals can occur when a developer chooses an alternative for a decision at hand and expects the emotion. Another possible positive affect can occur when the developer is making the decision and experiences the affect. These two kinds of affects in decision making processes—namely, expected and immediate emotions—are also expected to occur in Agile processes, and have been highlighted by many authors, such as [63] and [64]. However, what is different in Agile is that not every developer makes the decision and expects or experiences the affective state; rather, Agile teams and individuals make decisions constantly and the overlap of continuous good communication, information sharing, and cooperation play a role in these types of individual affects, in each development software release and spanning until the final product is finished. Positive affect can occur in Agile when an individual makes a good decision and has a good feeling about the expected results, along with the contentious process of maintaining the positive affects of individuals to keep the overall expected and experienced affective state positive.

On the other hand, negative affects can reduce the level of cooperativeness between Agile teams. The cooperation of Agile team members can be influenced by a developer’s affective state, such as anger, fear, etc. Although the evidence supporting a relation between team cooperativeness and negative affects is already mentioned earlier in [112], the need to emphasize this relation in Agile arises for many reasons. First, communication does not enforce cooperation; instead, good communication is required to have a cooperative team. What makes teams cooperative in Agile is the flexibility of the process and the considering of people over the process. This means that negative affects can be a barrier to achieving cooperativeness. For example, in SCRUM process, when the scrum master’s affective state is negative, cooperation with developers could be reduced and the level of motivation and other positive factors lowered. Another example is when the number of Agile team members is quite
low, such as three, for example, as the negative affective states of one or more member could lower the level of cooperation between team members.

The negative affects can also play an important role in decreasing the degree of decision accuracy in all phases. First, some decisions made in the early stages of software development can lead to more sophisticated solutions, delay in completion times, and increase the associated cost of these decisions. This is because in software development, each software release is built on earlier releases until the final product is reached, which leads to what is called software debt [115]. Second, no psychological techniques are in use when the developers are in case of making a decision within Agile process; instead, the process does not consider the possibility of affect’s role. Finally, although decision makers are using current techniques, not all of them are applicable to Agile processes. For example, a technique is used called time delay between decisions [75], which is not always applicable in Agile because of the need for fast decisions to reconcile with the cycle period, while a technique aiming to enhance some characteristics related to the decision maker, such as his/her knowledge, motivation, and crowding out affect by exposing the decision maker to a similar situation [74] will not always work with regards to the current decision at hand.

The last factor that is a concern in this model is the relation between negative affects and team conflicts. Based on the literature, it is hypothesized that the negative affect of individuals can have an important negative influence on the level of conflict between team members. The influence can vary, depending on which negative affective state is experienced or expected by the developers. Given the results of the study in [44] that investigated the influence of two emotional states—namely, frustration and anger—on conflicts in software development, the influence of these two states along with others can play a more important role with regards to the level of conflict in Agile because of the characteristics of Agile, such as the fact that Agile is a people-centred process that focuses more on people and communication than the process itself.
Finally, work- and non work-related are the two factors that influence positive and negative affects. Both factors can change the affective state of individuals. Although this thesis mentions these factors throughout its discussion, studying the relationship between the work and non-work related influences and affects in Agile decision making is outside the scope of this thesis. Instead, the mention of these factors will only highlight which work and non-work factors developers and teams think influence their affective states.

3.3.2 The Model of Affective Contagion Influences on Cooperativeness, Conflicts, and Decision Quality

Given the fact that Agile requires good communication and a high level of cooperation between team members, the model in Figure 3.2 was created to illustrate the potential influences of positive and negative affective contagion on an Agile team’s cooperativeness, conflicts, and decisions. The difference in this model is that we introduce and define positive and negative affective contagion as the positive and negative affect of the Agile team instead of individual. Therefore, positive affective contagion is the spread of positive affect of individual among the team; and negative affective contagion is the spread of negative affect of individual among the team. Similar to the model of affect, each ellipse represents exogenous variable, as explained in SEM [5], that is influenced by factors outside the model we intend to test, such as work related factors. Rectangles represent endogenous variables that are influenced by endogenous or exogenous variables. Each arrow represents either positive or negative influence as hypothesized in this thesis.
Figure 3.2: Affective Contagion Impact on Agile Team’s Cooperativeness, Conflicts, and Decisions

The first relationship that the model of affective contagion investigates is the role of positive affective contagion on the level of cooperativeness between Agile team members. This level is hypothesized to be positively influenced when the sum of an individual’s affective states is spread and becomes higher as compared to the negative affective states between the team members. One might argue that as long as there are positive affects in individuals, the positive affective contagion will be high, and vice-versa. Thus, to increase the level of cooperativeness, the affective contagion does not only depend on the positive affect of individuals; however, the spread of an individual’s affective state itself is the controlling factor in this situation. In other words, spreading the affects is what makes either a higher or a lower level of cooperativeness. For example, the project manager’s positive affect might be more important than those of other team members. This positive affect can lead the manager to make a greater effort to cooperate with team members, hence improving communication.
Further evidence can be considered from [38], who have stated that “neuroticism, extraversion, openness to experience, agreeableness, and conscientiousness” are the key factors that influence cooperativeness. Some of these factors can be maximized when team members positively influence the affective states of others. Positive affective contagion is also hypothesized to be an influential factor that improves decision quality. Operational and tactical decisions are made frequently in Agile, and that leads the quality of the team’s decisions to be influenced more by the positive affective contagion. Furthermore, positive affective contagion between the Agile team members can overcome the negative factors that are related to decisions, such as the lack of empowerment [110], insufficient relationships with team members [108], and emotion [12].

Positive affective contagion is hypothesized to reduce the level of conflict between team members. The spread of positive affects can raise the level of agreeableness and aid in the avoidance of unsolicited conflicts. Most researchers in the literature on conflicts agree that conflict is something that cannot be avoided in the decision making process. This thesis does not focus on overcoming the conflict itself; rather, it seeks to remove the unsolicited conflict that is related to the affects. By highlighting this relationship, Agile teams can notice how the contagion of positive affects can reduce the level of conflict.

On the other hand, the contagion of negative affects are hypothesized to reduce the level of cooperativeness between team members. The literature has found a correlation between the role of negative affective states and contagion and cooperativeness between team members. Considering the finding in [112], that anger is an influential factor that decreases cooperativeness, anger and many other negative affective states can influence the cooperativeness of developers when they need to communicate constantly and effectively. Furthermore, almost all factors that are related to poor communication in Agile, such as moods and motivation [18], can be initiated or
influenced by negative affective contagion.

Similar to the level of cooperativeness, the negative affective contagion is hypothesized to be an influential factor in terms of decreasing decision quality. The main obstacle in Agile decision making mentioned in [110] is that team members are not willing to commit to their decisions; however, this can be seen from two different angles related to negative affective contagion. The first is that there might be a high level of negative affective contagion between the team members, which, along with other factors such as lack of motivation, can be a reason for not being willing to commit to the decisions. The second is that team members might not be satisfied with the decision they have made as a team, due to different factors, such as the existence of a high level of negative affective contagion that has led team members to avoid expressing their thoughts about the decision made.

The negative affective contagion are considered in the literature to influence conflicts. Therefore, we believe that a high level of negative affective contagion between team members significantly increases conflict in the Agile environment. Evidence of the role of emotions on interpersonal conflicts and project outcome has already been mentioned [44]. However, the relationship between negative affects and conflicts in Agile can be stronger than in traditional software processes because of the high emphasis on communication and cooperation in Agile.

Finally, similar to the model in Figure 3.1, work- and non-work related factors that influence the affective contagion will not be investigated in this thesis; instead, we illustrate these factors as influencing only an individual’s affects. What is different in this model is how it considers an individual’s affect as an influence on affective contagion, which will be explained in the next section when the aggregated model is illustrated.
3.3.3 The Aggregated Model of Affect and its Contagion Influences on Cooperativeness, Conflicts, and Decision Quality

For a more comprehensive test of how affects and their contagion influence Agile decisions, the model in Figure 3.3 aggregates and tests both affect and affective contagion models. The reality of Agile decisions requires us to combine and cover all possible influences that can occur during the decision-making process.

![Conceptual Model of Affects and Affective Contagion Impact on Agile Decisions](image)

**Figure 3.3:** Conceptual Model of Affects and Affective Contagion Impact on Agile Decisions

Decisions that are made by managers at tactical or strategic levels are less likely
to be shared and discussed with all team members; therefore, it can be subject to
manager’s affect with no role played by other teams members’ affective states (i.e.
affective contagion). For an example of this, a responsibility of the scrum master is
to manage the product backlog by deciding which methods are suitable to support
the product owner. On the other hand, an individual’s affect can lead to the spread
of that affect to become the team’s affect in decisions that are carried out by the
whole team. For example, the development team in scrum is responsible for making
decisions collectively when analyzing, developing, testing, and releasing software.

The hypotheses H3 and H4 were added to correlate between the role of affect and
the role of affective contagion in Agile decisions. Testing this aggregated model will
shed light on which construct (affect or affective contagion) plays a more important
role in the Agile environment. Later, in testing these hypotheses for building a struc-
tural (path) model, we will demonstrate which construct influences cooperativeness,
conflict, and decision quality more than the others. Understanding this will open the
way to derive and propose new techniques to avoid Agile decisions being influenced
by the most influential construct. Therefore, affective contagion will be tested in
the model as a mediator between affect and cooperativeness, conflict, and decision
quality.

Finally, Table 3.1 summarizes all constructs and the associated hypotheses that
will be tested during the research activities outlined in this thesis.
Construct | Hypothesis Number | Research Hypothesis
--- | --- | ---
Affect | H1 | The affects of Agile teams have a significant influence on the accuracy achieved in the Agile decision-making process
Positive Affects | H1.1 | The positive affects of Agile teams have a direct positive significant influence on team cooperativeness
 | H1.3 | The positive affects of Agile teams have a direct positive significant influence on team conflicts
Negative Affects | H1.2 | The negative affects of Agile teams have a direct negative significant influence on team cooperativeness
 | H1.4 | The negative affects of Agile teams have a direct negative significant influence on team conflicts
Affective Contagion | H2 | The affective contagion between Agile teams has a significant influence on the accuracy achieved in the Agile decision-making process
Positive Affective Contagion | H2.1 | The positive affective contagion between Agile teams has a direct positive significant influence on team cooperativeness
 | H2.3 | The positive affective contagion between Agile teams has a direct positive significant influence on team conflicts
Negative Affective Contagion | H2.2 | The negative affective contagion between Agile teams has a direct negative significant influence on team cooperativeness
 | H2.4 | The negative affective contagion between Agile teams has a direct negative significant influence on team conflicts
Affect & affective Contagion | H3 | There is a correlation between positive affect and positive affective contagion in Agile decision making insofar as an individual’s affect positively influences the affective contagion between team members
 | H4 | There is a correlation between negative affect and negative affective contagion in Agile decision making insofar as an individual’s affect negatively influences the affective contagion between team members
Cooperativeness | H5 | Cooperativeness between team members mediates the influence of positive and negative affects on Agile decision quality
 | H6 | Cooperativeness between team members mediates the influence of positive and negative affective contagion on Agile decision quality
Conflicts | H7 | Conflicts between team members mediate the influence of positive and negative affects on Agile decision quality
 | H8 | Conflicts between team members mediate the influence of positive and negative affective contagion on Agile decision quality

| Table 3.1: Summary of Research Hypotheses |

3.4 Control Variables

Control factors (or variables) are the ones that could possibly influence the research. Many authors have studied the possible effects of demographic variables on affects. For example, Schaubroeck and Jones highlighted the roles played by age, gender, and education level in emotional labour [116], while Pooja and Kumar considered age, gender, experience, education level, and management level [117]. In this research, the
number of factors might influence the results of this study; therefore, we include them in the study to measure any possible influence on the results of this research. The selected control factors are experience in Agile development, the Agile method, team member’s role in the software development, and gender. The following subsections briefly demonstrate the roles played by these variables in software development.

### 3.4.1 Experience

The experience of developers is considered to perhaps be the most influential factor in software development. Domino et al. insist that experience is positively and strongly related to performance [118]. The authors state that highly experienced team members have a higher level of satisfaction in code practices, such as pair programming in XP, and other practices that require face-to-face communication and collaboration, as compared to less experienced team members.

Software development phases, such as the design phase, requires developers to have knowledge in order to formulate and find the necessary information to solve a specific problem [119]. The problem in structuring design is highlighted by Robillard, who states that “software design is generally a mixture of ill- and well-defined problems. The specification and the design of the algorithms or the system architecture often constitute an ill-defined problem type; translation of the detailed design into programming code is more of a well-defined problem type ” [120]. The role of experience in these matters is important given that, “a novice may find a problem ill-defined, while an experienced designer considers it well-defined, because a well-defined goal for reaching the solution is available” [120]. Therefore, the experience of team members in software development is an important factor that must be taken under consideration when carrying out this research. Furthermore, the research will not only consider the experience of the team members, but also their experience in terms of the Agile development processes.
3.4.2 Agile Method

A number of studies have been conducted to investigate the influence of software processes, methods, and practices on developers and managers. For example, an empirical study concerning whether Agile processes influence developer cognition was conducted by Omar et al. in [89]. The positive influence of some Agile practices has been highlighted by some studies that state that these practices increase developers’ enthusiasm levels. Syed-Abdullah et al. also concluded that Agile has a positive influence on developers’ affective states [90]. The authors also highlighted the effects of the XP methodology on developers’ moods [91]. Furthermore, the authors found that when the number of XP practices used increased, the software team’s temperament increased also. The aforementioned studies have suggested applying the flexible Agile method to cope with the organization, environment, and culture in order to increase developers’ emotions and, subsequently, improve software quality.

Therefore, we will consider the variable of Agile method as a factor that can influence the results of this research. The research will consider whether team members are applying Agile or non-Agile methods. The focus in this research will be on Agile methods and practices, to highlight the role of affects and their contagion; therefore, the specific Agile method will be investigated to test whether the specific Agile method, such as XP or scrum, influences the results of this study.

3.4.3 Role of Team Member

The role of the team member in software development varies, especially in Agile environments. The differences are due to the reality of Agile, which considers the communication of all team members as a main value; hence, developers, testers, managers, and other team members are required to accomplish their roles and communicate effectively. Boehm and Turner have classified the levels of developers, where the lowest level represents one who is “unwilling or unable to work in a collaborative
environment” [121]. This classification of developers highlights their role in Agile development, which opens the way to consider the levels of other Agile team members and their possible influence, in terms of their affects, on the process. We consider the role of team member to be a control factor because managers’ and leaders’ affects influence the process of making decisions more than other team members’ affects regardless of the level of involvement in the Agile project. Therefore, the role of the team member, for example developer or manager, will be monitored during the research activities.

3.4.4 Gender

Gender is considered an important factor that can influence the overall performance in organizations [122]. Many studies have discussed the role of diversity and the presence of women in engineering. For example, Fonselius et al., in their thesis, discussed the role of women in the field of software development [123]. The authors also mentioned the fact that, “Ideologically, engineering and science working climates are systematized “masculine”, and most of the engineering fields are disciplined hard, objective, rational, and logical” [123]. On the other hand, the authors summarized and synthesized studies that insist on the need for diversity in organizations even though women are more emotional than men.

Considering that gender might influence this research, it is seen from a psychological point of view, taking into account the possible differences that gender might create in terms of experiencing affects and their contagion. Within the pair programming practice, for example, Choi investigated the different matching of pairs, based on his summary of the literature, stating that, “Women are found to be more concerned about others and tend to conform more to societal norms; in contrast, men are driven more by individual attitudes and objectives. The general consensus is that women prefer feeling-based judgement and manage well with ambiguity, whereas men
prefer facts, an analytic-based judgement, and a logical and rational approach” [124]. Therefore, in this thesis, we include the gender factor to see if it has any influence on the research results regarding the influence of affects and their associated spread.

### 3.5 Chapter Summary

- The chapter provided the research questions in this thesis, which are taken from the main question: How are affects and their contagion related to decision making in Agile environments?

- The correlation between the decision quality and affects and their contagion are highlighted within software development along with other fields.

- The correlation between the cooperativeness and affects and their contagion are highlighted within software development along with other fields.

- The correlation between the conflicts and affects and their contagion are highlighted within software development along with other fields.

- The chapter summarized the main theories and models that this research considers as its theoretical background.

- Based on the correlations between constructs, the chapter speculated the research hypotheses and models.

- The model of affects demonstrates how they positively influence decision quality and cooperativeness, while negatively influencing conflict.

- The model of affective contagion demonstrated how they positively influence decision quality and cooperativeness, while negatively influencing conflict.
• The aggregated model demonstrated the correlation between affects and affective contagion, as well as considering cooperativeness and conflict as mediators between affects, affective contagion, and decision quality.

• Finally, the chapter briefly discussed experience, the Agile method, the team member role, and gender as control variables for this research.
Chapter 4

Research Methodology

The main goal of this chapter is to outline the research design, methods, and approaches used in this study, as well as justifying the adoption of the approaches used. Furthermore, the chapter details the procedures adopted for collecting and analyzing both quantitative and qualitative data.

4.1 Research Designs

4.1.1 Quantitative vs. Qualitative Designs

In general, there are two main research approaches to empirical studies in software engineering; namely, they are exploratory (qualitative) and explanatory (quantitative) research, as Wohlin et al. have explained [125]. The use of both approaches in the same study has yielded an approach called mixed methods [1]. Within software engineering, Wohlin et al. have explained that exploratory research, or the studying of a current situation in its natural state, is used to extract observations about that situation. Therefore, the data collected from this kind of research are qualitative data. The authors also stated that data might be collected using inductive research, which infers data from people’s explanations concerning a situation in its natural state. On
the other hand, the authors explained that explanatory research is intended to find the causes and effects of a situation by, for example, conducting a controlled empirical study. Therefore, data collected from this kind of research are quantitative data; for example, statistical data comes from testing such activity in a process. Similarities and difference are explained by [2] and [3] in Figure 4.1.

Figure 4.1: Quantitative and qualitative research, differences and similarities

Quantitative research is simply explained by many authors as dealing with numbers. Creswell defines the quantitative approach as an “approach for testing objective
theories by examining the relationship among variables. These variables, in turn, can be measured, typically on instruments, so that numbered data can be analyzed using statistical procedures” [4]. Quantitative research is considered research that can generalize its results by controlling the samples and variations between participants [126]. Quantitative research seeks to test a theory deductively. This is done by developing hypotheses regarding relationships between variables, and then confirming the predicted outcomes, which ultimately helps construct the theory [127].

The advantages of quantitative research are that the study can be replicable, which leads to the generalizing of results, the isolation of the researcher in terms of his/her involvement in the study, and it normally takes less time and costs less to conduct the study [128]. To have an overview of the strengths and weakness of both quantitative and qualitative research methods, we encourage the reader to read [7].

Qualitative research is simply explained by many authors as dealing with texts and other forms of data other than numbers. Creswell defines that qualitative approach as an “approach for exploring and understanding the meaning individuals or groups ascribe to a social or human problem. The process of research involves emerging questions and procedures, data typically collected in the participant’s sitting, data analysis inductively building from particular to general themes, and the researcher making interpretations of the meaning of the data” [4]. Qualitative research, unlike quantitative, seeks to collect data first, without having hypotheses. From the data, the researchers develops a theory based on the phenomena in their natural settings, as stated by Worthen and Sanders in [129]. The authors stated that the qualitative approach “is generally conducted in natural settings, utilizing the researcher as the chief “instrument” in both data gathering and analysis. The benefits of qualitative inquiry are embedded in its emphasis on thick description, i.e. obtaining real, rich, deep data which illuminates everyday patterns of action and meaning from the perspective of those being studied” [129].
Some advantages of qualitative research are the role of researcher in the study and the richness of the data gathered; it has been stated that qualitative approaches “generate rich, detailed data that leave the participants’ perspectives intact and provide multiple contexts for understanding the phenomenon under study” [130]. However, many researchers have explained how the role of the researcher in qualitative research significantly increases the time and cost needed to collect the data.

Finally, researchers have highlighted that quantitative and qualitative are not as rigidly separate as they are often expected to be. Instead, the investigation in a study can be more quantitative than qualitative and vice-versa [4].

### 4.1.2 Mixed Methods Design

Mixed methods research involves the simultaneous use of both quantitative and qualitative methods during a single study, as stated in [7]. Creswell highlights the need to use both qualitative and quantitative methods for the same investigation, which can be conducted through one or more studies [4]. There are many advantages to conducting mixed methods research, as summarized by [7] in Table 4.1, including acquiring the advantages of both qualitative and quantitative methods.

Creswell clarifies three main types of mixed methods, or how quantitative and qualitative methods can be used jointly within a single study. These mixed methods are convergent parallel, explanatory sequential, and exploratory sequential. The convergent parallel mixed method requires that the researcher collect both types of data, qualitative and quantitative, analyze them separately, and compare the results of each type. The exploratory mixed method requires that the researcher collect only qualitative data and analyze them. Based on the results of the qualitative method, the researcher collects quantitative data and analyzes them in order to arrive at the final results. As opposed to exploratory, the explanatory mixed method requires the researcher to collect only quantitative data and analyze them; based on the results,
<table>
<thead>
<tr>
<th>Strength</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Words, pictures, and narrative can be used to add meaning to numbers.</td>
<td>• Can be difficult for a single researcher to carry out both qualitative and quantitative research, especially if two or more approaches are expected to be used concurrently; it may require a research team.</td>
</tr>
<tr>
<td>• Numbers can be used to add precision to words, pictures, and narrative.</td>
<td>• Researcher has to learn about multiple methods and approaches and understand how to mix them appropriately.</td>
</tr>
<tr>
<td>• Can provide quantitative and qualitative research strengths.</td>
<td>• Methodological purists contend that one should always work within either a qualitative or a quantitative paradigm.</td>
</tr>
<tr>
<td>• Researcher can generate and test a grounded theory.</td>
<td>• More expensive.</td>
</tr>
<tr>
<td>• Can answer a broader and more complete range of research questions because the researcher is not confined to a single method or approach.</td>
<td>• More time consuming.</td>
</tr>
<tr>
<td>• A researcher can use the strengths of an additional method to overcome the weaknesses in another method by using both in a research study.</td>
<td>• Some of the details of mixed research remain to be worked out fully by research methodologists (e.g., how to interpret conflicting results).</td>
</tr>
<tr>
<td>• Can provide stronger evidence for a conclusion through convergence and corroboration of findings.</td>
<td></td>
</tr>
<tr>
<td>• Can add insights and understanding that might be missed when only a single method is used.</td>
<td></td>
</tr>
<tr>
<td>• Can be used to increase the generalizability of the results.</td>
<td></td>
</tr>
<tr>
<td>• Qualitative and quantitative research used together produce more complete knowledge necessary to inform theory and practice.</td>
<td></td>
</tr>
</tbody>
</table>

**Table 4.1: Strength and Weakness of Mixed Methods Design** [7]
the researcher collects qualitative data and analyzes them in order to arrive at the final results. The last two types of mixed methods require two different phases in order to conduct the study.

4.2 Research Methods

Generally, Wohlin et al. outline four major strategies for conducting empirical study in software engineering, which are survey, case study, experiment, and quasi-experiment [125]. Creswell classifies quantitative designs as experimental or non-experimental (Survey research), while qualitative design are narrative or phenomenological designs [4].

The survey strategy concerns collecting information from people by different means, such as interviews and questionnaires, as stated by Fink in [131]. The author explained that the information might be about people’s knowledge, behaviour, practices, or attitude; which is later aggregated, compared, and analyzed in order to come up with results that can be generalized to the community that the information is collected from. He highlighted seven activities that have to be conducted in order to successfully apply this type of research. These activities are “setting objectives for information collection, designing the study, preparing a reliable and valid survey instrument, administering the survey, managing and analyzing survey data, and reporting the results” [131]. Since survey research is one of the strategies that this thesis uses, the first four activities for this survey research are mentioned in Section 4.3.1, while managing and analyzing data, as well as reporting the results, are discussed in Chapter 5.

The case study strategy concerns studying a case or situation in its reality [125]. Runeson et al. state that case study method collects data and provides results from a real project or activities, whereas no other method, such as a survey, can provide the
same results [132]. However, the disadvantage of the case study is that researchers have less control over a case study and cannot change inputs; therefore, the case study is considered an observational study [125]. Both Wohlin et al. and Runeson et al., along with other researchers, provided step-by-step activities for conducting case study research within the field of software engineering. The main five steps taken to conduct a case study in software engineering are summarized in [125].

Experiments, as Wohlin et al. state, are the most difficult strategy, as they require accurate procedures to draw strong conclusions about the studied situation [125]. The authors explain that an experiment is the most controlled strategy, as the researcher must have a lot of control over the factors and variables during the experiment. They state that the researcher must have an idea or hypothesis about the relationship between the cause construct and the effect construct. This relationship is tested through observation of the treatment of the cause construct by applying independent values in order to reach the outcome based on dependent values. The researcher must not overlook any factor that has an effect on the dependent variable. Like a case study, an experiment must follow specific steps; namely, scoping, planning, operation, analysis and interpretation, presentation and packaging, and reporting, which are also summarized in [125].

4.3 Research Approach of this Thesis

4.3.1 Research Approach

During this research, we applied mixed methods to broaden our understanding regarding the role played by affects and affective contagion as related to developers’ decisions. Mixed methods enabled us to investigate the role of affects and affective contagion based on team members’ opinions and experiences, along with measuring their actual affects and the possibility of them being contagious to other team mem-
Furthermore, mixed methods enabled us to investigate the role of affects, their work-related sources, how they are contagious during the making of Agile decisions, and if the affective states mentioned in the investigation are the most influential or if there are other affective states that have been overlooked.

Choosing the mixed methods design enables us to exploit the strengths of both quantitative and qualitative methods, as mentioned in Table ??, in addition to the strengths of mixed methods, as mentioned in Table 4.1. Furthermore, some of the disadvantages to mixed methods research are not fully considered as drawbacks in this thesis for various reasons. First, being the single author of these research activities is minimized because of the guidance of my thesis supervisor. Second, as a PhD student, learning both methods enables me to conduct more diverse studies after graduation, along with broadening my knowledge about these methods, instead of strictly sticking to either quantitative or qualitative methods. Finally, the time-consuming problem was considered as soon as it was decided to apply the mixed methods approach in this thesis. Therefore, a specific timeline for conducting the thesis activities was set and strictly followed during my study.

For this study, the explanatory sequential design strategy was used, which means that we started with quantitative research followed by qualitative research, as depicted by [4] in Figure 4.2. Cresswell states that the explanatory sequential design is suitable for providing “A more in-depth understanding of the quantitative results (often cultural relevance)” [4]. He continues by explaining that mixed methods does not mean that the data collected through the two methods has to be actually merged in all cases; instead, it can be connected, as in the explanatory sequential design strategy. Connected here means that quantitative data is gathered and analyzed, and then qualitative data is gathered based on the analysis of the quantitative responses. This is suitable in our case because we can find surprising results concerning the influence of affects and their contagion in the Agile environment, which might re-
quire in-depth inquiry by adopting qualitative research methods. Finally, as Creswell states, explanatory sequential design “is better suited to explaining and interpreting relationships” [1]. First, we intend to gather information from experts and novices concerning what they think about the role of affects and affective contagion as related to their decisions, and the actual roles of their affects during Agile project activities, where possible activities can be related to the environment, daily meetings, and problem characteristics. Then, this data will help to determine the direction that we will take to investigate in greater depth the roles of affects and affective contagion in terms of how they influence decisions.

The quantitative research strategy is used through the adoption of the survey method to collect data about the affects and their contagion that play a role in decision making based on developers’ experiences and opinions. A survey is defined as “a system for collecting information from or about people to describe, compare or explain their knowledge, attitudes and behaviour” [125]. The survey collects information from developers, managers, stakeholders, and other team members in order to describe the potential sources that influence their affects, their contagion, and their expectations regarding the role of affects as related to their decisions. The survey also collects data on the experiences of developers, as we believe experience is a factor that can influence the results. Unlike the study conducted by Colomo-Palacios et al. [16], in which they examined affects related to decision characteristics, we examine all potential sources, including decision characteristics, using this survey. The sample for this quantitative survey will allow us to enhance our understanding of the influence of affects and
their contagion on the population and the sources of affects before examining this further during the next phase. Therefore, the survey is more focused on exploring the issue, which is called an explorative survey in [125]. Wohlin explains that these types of surveys “are used as pre-study to a more thorough investigation to assure that important issues are not foreseen” and that is exactly our idea for this phase [125].

The objective of this survey is to collect data from a larger pool of Agile developers, managers, customers, stakeholders, and any other members involved during software development. The objective in terms of how the collected data will be used is to test the conceptual models depicted in the previous Chapter 3.3. It was decided to adopt the survey method to test the models because of the advantages that survey research can provide. First, Blaxter et al. state that survey research provides the researcher, along with other advantages noted in Figure 4.3 with the ability to gen-
eralize his/her results [2], which is suitable for testing our conceptual models in order to highlight the roles played by affects and affective contagion within Agile software development. Furthermore, in response to the need for a large number of participants to test both measurement and structural models, as we explain later in Section 4.5.2 and Section 4.5.3, the survey allows us to reach a larger pool of Agile team members. Blaxter et al. state that “with a good response rate, surveys can provide a lot of data relatively quickly” [2]. Most of the disadvantages of the survey are related to its being quantitative research, such as Blaxter et al. stating that “the survey relies on breadth rather that depth for its validity” [2], which is avoided in this thesis by also engaging in more in-depth qualitative study.

Questionnaires used to collect data were sent to developers through both e-mail and Qualtrics survey software. The University of Regina has the licence for this software, which allowed me to design the survey, disseminate it, and collect the data through it. At the beginning of designing the survey, we made sure to include the necessary instructions for how to complete the survey, as well as clarifying all questions, definitions, and terms used in the questionnaires. The differences between terms such as emotion, moods, and affects, along with what is considered to be an emotion and what is not, is clarified as well. Unfortunately, one of the drawbacks of the survey method, as a form of quantitative research, is that we are unable to conduct interview surveys and obtain the associated benefits, such as reducing the number of unanswered questions and other benefits noted by Wohlin [123], due to the small number of expert developers and the lack of an interactive and open line of communication. However, different techniques were used to deal with missing data, outlier responses, and other cleaning techniques were also used for the data gathered in this survey, as is explained later in this chapter and in Chapter 5.

We then adopted the grounded theory (GT) strategy for the qualitative research, in order to help us understand, interpret, contrast, and explain the results of the
qualitative study as well as the preliminary quantitative results. This is an important step to take in order to derive a better understanding and confirm the hypotheses regarding the role of affects and affective contagion as related to Agile decision making based on in-depth interviews.

The objective of the qualitative study is to understand how affects and affective contagion influence cooperativeness and conflicts and, hence, the quality of decisions made by team members. The investigation in this qualitative study includes highlighting how the constructs influence each other, how the constructs influence the decisions, and all the possible work-related influences on affects and affective contagion in Agile. The interview, as a qualitative strategy, allows us to gather information in different formats than when a quantitative strategy was used. For example, expert managers might have opinions on how affects influence the team members’ decisions in Agile, which could be shared in an interview, while the quantitative method does not offer a way to collect and analyze these observations.

GT is a qualitative methodology that enables researchers to achieve their goal of deriving a theory. Unlike testing or validating an existing theory deductively, GT seeks to generate the theory inductively from the data, through complicated but systematic procedures [6]. Glaser and Strauss [133] are the founders of GT, and it has been applied in several fields, such as psychology and social studies. There are three versions of GT used by researchers, with some differences existing between them. Recently, Stol et al. reviewed the use of GT within the field of software engineering [134]. They found that several studies did not explicitly explain the procedures used when applying GT, nor did they explain which version of GT was adopted. The authors explained the differences between the three versions, which include the Glaserian version developed by Glaser and Strauss [135], the Straussian version developed by Strauss and Corbin [136], and the Constructivist version developed by Charmaz [6]. The main difference between the three versions is the process used for
coding the data. Although both the Glaserian and Straussian versions agree on the definition of the coding process, which involves analyzing data in GT, they conduct GT differently. Glaser divides the coding process into two main phases: substantive and theoretical coding. The substantive phase is further divided into two sub-phases: open and selective coding. On the other hand, Strauss divides the coding process into three main phases: open, axial, and selective coding. Other differences are related to the literature review’s role, when research questions are defined, the types of questions asked during the interviews, and evaluation criteria. Charmaz, on the other hand, based her version of GT on social constructionism, which considers how individuals and groups construct their reality. For this research, the constructivist GT is the most appropriate version for many reasons. First, the research questions and light literature review are important for collecting secondary data during the process of collecting data. The classic version, for example, suggests delaying the literature review, while the research questions should not be defined before conducting the study. Second, the constructivist version enables us to investigate, during the study, how affects and affective contagion can be influenced during communications between Agile teams, as well as considering their thoughts and feelings during their collective actions. The in-depth interview is the strategy used to collect the qualitative data, which are not only used to test the measurement models, as they are tested quantitatively, but they are also used to confirm each relation or influence of a construct on other constructs in order to test the aggregated research model. When a construct is found to significantly influence another construct in both quantitative and qualitative studies, the results become confirmed quantitatively and qualitatively. On the other hand, when the results between quantitative and qualitative methods conflict, then they are explained, compared, and contrasted during the discussion of the overall results in Chapter 7. Finally, based on the aforementioned pillars of any research, as suggested by authors such as [4], Figure 4.4 illustrates the approach adopted in this
thesis along with the philosophical paradigm, research design, and research methods for each design, including data collection and analysis tools.

### 4.3.2 Research Plan

The research plan in this thesis (see Figure 4.5) starts by highlighting the gap in the existing knowledge and justifying the conducting of this research. Then a comprehensive literature review regarding the three main components in this thesis—namely, affects, affective contagion, and decisions in Agile—along with other relevant components is included. This results in the conceptual models of affects and affective contagion, and the roles they play in Agile decision making. The models are tested using the approach proposed earlier in 4.3.1. During phase two, quantitative research in the form of a survey was conducted. In this thesis, we decided to put the weight of testing the models more on the quantitative side; hence, we gathered a large pool of developers, managers, and other team members, and analyzed the data intensively using complicated statistical tests and models, such as SEM. The reason for this decision was the need for more generalized results to confirm the models regardless of
the setting of the investigation. The second reason for putting more weight on the quantitative side was the lack of empirical studies concerning the role of affects and affective contagion in Agile decision making (see [2.4]).

Figure 4.5: Research Plan for the Study of Affects and Affective Contagion Role in Agile Decisions.

Phase three considers the results of phase two in order to open the way for designing supporting questions and investigations. Therefore, the qualitative method in the form of interviews was developed based on survey findings, in order to confirm these findings and highlight any further findings. The last phase is involves combining the findings and revising the models accordingly, along with providing a discussion of the
findings and conclusions as part of confirming the proposed models.

4.4 The Adopted Procedure for Quantitative Data Collection

4.4.1 Questionnaire Development

We developed a questionnaire in order to collect quantitative data that would test the models and related hypotheses mentioned in Chapter 3. The survey sought to measure developers’ affects and acquire their opinions regarding the level of cooperation achieved, the level of conflict that occurred, and the level of decision quality achieved. This means that our questionnaire focuses on a type of data called subjective data, while objective data is gathered for the control variables. Furthermore, a few open-ended questions were designed to gather data in order to specify internal and external (or work- and non-work-related) factors that cause affects and affective contagion to influence Agile decisions. As explained by Kitchenham and Pfleeger, who investigated survey research within software engineering, “subjective data (concerning individual’s opinions, attitudes and preferences) and objective data such as demographic information for example a subject’s age” [137].

The questionnaire targets three types of data; data about control variables, data about team members’ opinions, and data about the team members’ affective states, cooperativeness, conflicts, and decisions made during their actual involvement in an Agile project (Appendix A).

Starting with demographic variables first, participants are asked about their gender, general experience, experience in Agile development, roles played within a team, and the Agile method used during their last project. These are the control variables in the study that might influence the results, as explained in 3.4. Furthermore, an
important question asks if the participant was involved during the last two months in a project. This question is important in order to collect valid data when we measure the participants’ affect using the psychological measurements that use a two-month period to capture the affects. However, participants are allowed to complete the survey even if their answers are excluded when testing the models. The excluded participants’ answers are acquired nonetheless in order to consider their opinions about affect’s role in Agile decision making.

Second, regarding the role of affect and affective contagion in Agile decisions, participants are asked about their opinions on whether or not they think positive and negative affects and affective contagion influence Agile decisions (see Table 4.2). Furthermore, the team members are asked, if they believe affects and their contagion play a role, to select what positive and negative affective states they think have the most influence on Agile decisions. A list of all affective states, adopted from the original PANAS scale \[57\], is displayed for the participants to choose from. Although these data are analyzed quantitatively, they will not be used to test the models, because we want to test the models based on real measurements of genuine involvement in Agile projects, not based on opinions. That is, the opinions of the team members will only highlight their awareness of the role played by affects in Agile decisions, and determine which affective states the team members consider to be most influential. Highlighting their awareness is especially important when the results of the actual measuring in the next step prove the influence of affects, while opinions do not agree with those results. Furthermore, developers’ opinions can support and justify making some possible modifications to the model later on when testing measurement and structural models. One possible justification could be the need to delete one item in the measurement model in order to enhance the results, as explained in \[4.5.2\]. Furthermore, participants are also asked two optional open-ended questions that revolve around what work- and non-work-related factors they think most cause or change
<table>
<thead>
<tr>
<th>Independent Variable</th>
<th>Dependent Variable</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Affect (PA)</td>
<td>Decision quality</td>
<td>Positive affects (emotions and moods) enhance the quality of Agile decisions</td>
</tr>
<tr>
<td></td>
<td>Cooperativeness</td>
<td>Positive affects increase the level of cooperativeness in Agile teams</td>
</tr>
<tr>
<td></td>
<td>Conflicts</td>
<td>Positive affects decrease the level of conflicts in Agile teams</td>
</tr>
<tr>
<td>Negative Affect (NA)</td>
<td>Decision quality</td>
<td>Negative affects (emotions and moods) decrease the quality of Agile decisions</td>
</tr>
<tr>
<td></td>
<td>Cooperativeness</td>
<td>Negative affects decrease the level of cooperativeness in Agile teams</td>
</tr>
<tr>
<td></td>
<td>Conflicts</td>
<td>Negative affects increase the level of conflicts in Agile teams</td>
</tr>
<tr>
<td>Positive Affective Contagion (PAC)</td>
<td>Decision quality</td>
<td>Positive affective contagion between team members enhances the level of Agile decisions quality</td>
</tr>
<tr>
<td></td>
<td>Cooperativeness</td>
<td>Positive affective contagion between Agile team members increases the level of cooperativeness</td>
</tr>
<tr>
<td></td>
<td>Conflicts</td>
<td>Positive affective contagion between team members decreases the level of conflicts</td>
</tr>
<tr>
<td>Negative Affective Contagion (NAC)</td>
<td>Decision quality</td>
<td>Negative affective contagion between team members decreases the level of Agile decisions quality</td>
</tr>
<tr>
<td></td>
<td>Cooperativeness</td>
<td>Negative affective contagion between Agile team members decreases the level of cooperativeness</td>
</tr>
<tr>
<td></td>
<td>Conflicts</td>
<td>Negative affective contagion between team members increases the level of conflicts</td>
</tr>
</tbody>
</table>

Table 4.2: Surveying Participants’ opinions of the influence of independent variables on dependent variables

Positive/negative affects/affective contagion to influence decisions in Agile projects.

Third, to test the models depicted in Section 3, the survey adopts various instruments for measuring Agile team members’ affects and their contagion, while combining others to measure levels of cooperativeness, conflict, and decision quality within Agile decision making processes, in order to have a validated and reliable instrument for this research. The different constructs and the items and sources of these items are summarized in Section 4.3.

Among the different scales, the affect scale is adopted from the International - Positive And Negative Schedule - Short Form (I-PANS-SF) scale, which was developed.
by Thompson in [59]. We chose this scale for two reasons. First, the scale was validated in terms of different cultures, proving the scale’s usefulness for cross-cultural experiments. This survey collects data from different Agile team members regardless of their locations, as long as the target team members follow Agile processes or practices. Therefore, understanding the affective states of developers from different countries, especially those whose first language is not English, is an important issue to consider in this research in order to have valid results. Second, I-PANAS-SF is a scale that was shortened from the original 20 items of the PANAS scale [57], and it includes only 10 items for measuring positive and negative affects. This is suitable for this research, which seeks to investigate whether or not affective states influence Agile decisions. The original PANAS scale in [57] can be adopted in other cases, such as when researchers seek a more comprehensive understanding of all core affective states and their influence in a specific culture.

Five items are included in the I-PANAS-SF scale to measure positive affects; namely, they are active, attentive, determined, inspired, and alert. The other five items of the scale are included for the negative affects; namely, they are upset, hostile, ashamed, nervous, and afraid. The main questions in the two scales were adopted from the original PANAS scale [57] with slight adjustments in order to fit the Agile iteration or phase. Both PANAS and I-PANAS-SF scales were tested and validated to measure affects for time periods of up to two months. Agile requires iterations with short periods of time; a typical time is two weeks per iteration [138]. Therefore, all Agile team members who participate in this study fall within the period that the I-PANAS-SF scale was designed for. Furthermore, to avoid the Agile teams misunderstanding, we adjusted the main question of the scale to include the word “iteration” and added a direct question in the survey to ask each participant whether or not the last iteration he/she was involved in occurred within the last two months. A 5-point Likert scale was adopted (ranging from 1= very slightly or not at all to 5
The affective contagion scale was adopted from a scale developed by Klapwijk and Van Lange [139] to measure affects during a certain task. Although the affective contagion between team members is not clearly mentioned by the authors, we slightly changed this scale to measure team member’s affective changes during the iteration and during decision making within the iteration. Therefore, we consider changes to team members’ affects because of other team members to be contagion, or susceptibility to a higher level of changing affects because of other team members. The scale contains eight items, including four items for the positive affective contagion and four items for the negative affective contagion. The positive items are contented, happy, proud, and enthusiastic, while the negative items are frustrated, disappointed, indignant, and angry. We are aware that these items are different than the items in the positive and negative affects scale. However, we adopted the aforementioned items because we want to build our instruments on reliable scales tested in the fields of psychology and social psychology. Furthermore, the items in this scale are related to the I-PANAS-SF scale; for example, when a manager’s affective state is measured by the I-PANAS-SF scale as showing a high level of being upset, he/she might behave in a way that make developers feel angry, which can be measured by the affective contagion scale. To maintain consistency with other measures, a 5-point Likert scale was adopted (ranging from 1 = not at all to 5 = very much).

A cooperativeness scale was adopted from Dierdorff et al.’s scale [140] that contains four items that measure the quality of team cooperation. The scale was adopted from the original 14-item scale called team-member exchange (TMX), which was developed by Seers [141] to measure “team member self-perception of the willingness to help others, to share ideas and feedback with other team members, and team member perception of how readily help, information, and recognition are received from others” [140]. The four adopted items were adjusted slightly in order to in-
crease understandability within Agile software development. A 5-point Likert scale was adopted (ranging from 1 = strongly disagree to 5 = strongly agree) to acquire participants’ answers.

The conflict scale, which measures the presence of different types of conflicts, was adopted from Jehn in [142]. The author includes a set of items for measuring task conflict, which is defined as “disagreements among group members about the content of the tasks being performed, including differences in viewpoints, ideas, and opinions” [142]. The author also includes another set of items for measuring relationship conflict, which is defined as “interpersonal incompatibilities among group members, which typically includes tension, animosity, and annoyance among members within a group” [142]. We considered and selected items that measure only task conflicts because we investigated affects and affective contagion in Agile environments that contain different characteristics in terms of tasks and decisions. On the other hand, although relationship conflict can exist in any group, it shares similar characteristics in Agile and other environments. Furthermore, highlighting the different types of conflicts and their characteristics is beyond the scope of this research. As a result, four items related to task conflicts were selected and slightly modified in order to increase understandability within the Agile software development. A 5-point Likert scale was adopted (ranging from 1 = none to 5 = a lot) to acquire participants’ answers.

The decision quality scale was adopted from Janssen and Veenstra in [143]. The authors proposed a measurement model to measure the effectiveness of team decisions, and they provided three items to measure decision quality. We adopted these three items and combined them with two more indicator items, adopted from [144], which ask the Agile team members whether or not the overall decision quality is high and about the best decisions made during the iteration. A 5-point Likert scale was adopted (ranging from 1 = strongly disagree to 5 = strongly agree) to acquire participants’ answers.
<table>
<thead>
<tr>
<th>Construct</th>
<th>Coding Number</th>
<th>Item</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Affect (PA)</td>
<td></td>
<td>PA1 Indicate the extent you have felt this way over the past week/weeks of your project’s iteration: Active</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PA2 Indicate the extent you have felt this way over the past week/weeks of your project’s iteration: Attentive</td>
<td>[53], [57]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PA3 Indicate the extent you have felt this way over the past week/weeks of your project’s iteration: Determined</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PA4 Indicate the extent you have felt this way over the past week/weeks of your project’s iteration: Inspired</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PA5 Indicate the extent you have felt this way over the past week/weeks of your project’s iteration: Alert</td>
<td></td>
</tr>
<tr>
<td>Negative Affect (NA)</td>
<td></td>
<td>NA1 Indicate the extent you have felt this way over the past week/weeks of your project’s iteration: Upset</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NA2 Indicate the extent you have felt this way over the past week/weeks of your project’s iteration: Hostile</td>
<td>[53], [57]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NA3 Indicate the extent you have felt this way over the past week/weeks of your project’s iteration: Ashamed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NA4 Indicate the extent you have felt this way over the past week/weeks of your project’s iteration: Nervous</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NA5 Indicate the extent you have felt this way over the past week/weeks of your project’s iteration: Afraid</td>
<td></td>
</tr>
<tr>
<td>Positive Affective</td>
<td>PAC1</td>
<td>The behaviour related to the other’s affective state made me feel happy</td>
<td>[139]</td>
</tr>
<tr>
<td>Contagion (PAC)</td>
<td>PAC2</td>
<td>The behaviour related to the other’s affective state made me feel contented</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PAC3</td>
<td>The behaviour related to the other’s affective state made me feel contented</td>
<td>[139]</td>
</tr>
<tr>
<td></td>
<td>PAC4</td>
<td>The behaviour related to the other’s affective state made me feel proud</td>
<td></td>
</tr>
<tr>
<td>Negative Affective</td>
<td>NAC1</td>
<td>The behaviour related to the other’s affective state made me feel disappointed</td>
<td>[139]</td>
</tr>
<tr>
<td>Contagion (NAC)</td>
<td>NAC2</td>
<td>The behaviour related to the other’s affective state made me feel frustrated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NAC3</td>
<td>The behaviour related to the other’s affective state made me feel indignant</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NAC4</td>
<td>The behaviour related to the other’s affective state made me feel angry</td>
<td></td>
</tr>
<tr>
<td>Decision Quality (DQ)</td>
<td>DQ1</td>
<td>Final team decisions is of much higher quality than the initial proposals of the individual members</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DQ2</td>
<td>Final team decisions reflects the best that could be extracted from the team</td>
<td>[143], [144]</td>
</tr>
<tr>
<td></td>
<td>DQ3</td>
<td>Final team decisions usually extended the quality of team member’s individual input</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DQ4</td>
<td>I am confident that the final decision we came up with is the best decision</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DQ5</td>
<td>Overall, it is my opinion that our final decision is of high quality</td>
<td></td>
</tr>
<tr>
<td>Cooperativeness (COO)</td>
<td>Coo1</td>
<td>Other team members usually let me know what they expected from me</td>
<td>[139]</td>
</tr>
<tr>
<td></td>
<td>Coo2</td>
<td>I often made suggestions to other team members about better work methods</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coo3</td>
<td>When I was busy, other team members volunteered to help me out</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coo4</td>
<td>was willing to help finish work that had been given to other team members</td>
<td></td>
</tr>
<tr>
<td>Conflicts (CONF)</td>
<td>Conf1</td>
<td>How often do your team members disagree about opinions regarding the work being done</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conf2</td>
<td>How frequently are there conflicts about ideas in your team</td>
<td>[142]</td>
</tr>
<tr>
<td></td>
<td>Conf3</td>
<td>How much conflict about the work you do is there in your team</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conf4</td>
<td>To what extent are there differences of opinion in your team</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3: Research Constructs, Items and their Sources
Despite the fact that all of the adopted scales were tested for validity and reliability by their authors, none of these scales were intended to measure the associated variables within software development. Furthermore, some of the scales were slightly changed, such as the conflict and cooperativeness scales, while other scales were shortened and combined with other scales, such as the affective contagion and decision quality. However, coming up in Section 4.4.4, we provide the results of a pilot study involving the proposed instrument that we conducted before disseminating the survey. We also provide, later in Section 5.4.2, a detailed description of the reliability of the instrument and its items based on the results of reliability tests such as Cronbach’s alpha.

4.4.2 Sample Size

The sample size required for this thesis differs based on the phase, i.e. quantitative or qualitative. Determining the sample size is important to generalize the results of the study based on the represented sample that includes diverse parts of the population. In determining the sample size for the quantitative phase, two issues arise. First, there is the need to determine the sample size based on a well-defined targeted population in software development along with considering the properties of this research that focus on psychological and social psychological factors. Second, there is the need to consider the sample size required for testing the research hypotheses using SEM, which requires a larger number of subjects as compared to other statistical models.

First, since the objective of the survey research is to highlight the role of affects and affective contagion in Agile environments regardless of the specific location or population, the targeted population in this research is unknown. The survey is disseminated to any team member who was involved in an Agile project regardless of his/her company or location in the world. However, some researchers have provided different ways to estimate the sample size based on the number of independent vari-
ables that the researcher intends to examine. For example, Tabachnick and Fidell state that the sample size must be greater than $50 + 8m$ (where $m$ is the number of independent variables in a study) \[145\]. Another researcher states that, in social research, the sample size must not be less than $15 \times m$ (where $m$ is the number of independent variables in a study) \[146\]. Considering these equations, note that this research contains four main independent variables; namely, they are positive and negative affects and positive and negative affective contagion. Each independent variable is tested against three dependent variables; namely, they are cooperativeness, conflict, and decision quality. Therefore, by calculating the sample size based on the two aforementioned formulas, we need between 146 and 180 team members to answer this survey. The Internet is the main means for disseminating the survey and getting the responses needed from different countries and companies.

However, since we use SEM for testing the research models, the number of participants needed for this study is different. Considering the fact that there is no rule of thumb for estimating the number of participants for SEM, different researchers suggest different ranges of participants needed when using SEM. For example, Kline recommends that the typical number of participants must be 200 or above \[147\], while Wolf et al. have surveyed many studies in psychology that use SEM in their testing and found the range of participants to span between 30 and 450 participants \[148\].

To conduct our research with a suitable sample size, we increased the threshold mentioned by the aforementioned authors regarding the number of participants to include no less than 250 responses from team members. To achieve that, we considered the low response rate of Internet surveys as compared to other means. Nulty investigated many studies and compared the response rate of online surveys, finding that 33% is the average response rate for these surveys \[149\]. Considering the average response rate (33%) while considering the required number of responses (>250), we followed a procedure of sending the survey out to three different batches of potential
participants, with each batch composed of approximately 300 potential participants.
A total of 900 Internet surveys were disseminated, and our average response rate
was 31%, making the total number of responses 257. Eight responses were deleted
during pre-processing and screening of the data, which left a total of 249 responses
to be processed and used to examine developers’ opinions about the role played by
affect and affective contagion in Agile decisions. However, 233 of the total responses
were used for building and testing the measurement and structural models because
16 participants stated that they were not involved in any Agile project during its last
two months; therefore, their responses were excluded.

4.4.3 Questionnaire Instrument Validation

Validating a survey questionnaire involves testing its content, construct, and reli-
ability [150]. Fortunately, almost all items were adopted from previous studies in
different fields that provided the validation of their instruments. Moreover, we chose
to measure the affects using the I-PANAS-SF scale in order to ensure understandabil-
ity from all participants and avoid national bias. However, because we deleted and
modified some items from some of the adopted scales, we included two steps in or-
der to validate the instrument before the final dissemination of the survey questions.
In the first step, I reviewed the questionnaire with two PhD candidates in software
engineering department and one PhD candidate in the computer science department
at the University of Regina. The candidates are either knowledgeable or interested
in the field of behavioural software engineering and affect’s role in software develop-
ment. The process of reviewing the feedback and adjusting the questions to ensure
content validity within software engineering took place through three different meet-
ings. Second, we conducted a pilot study before the actual process of disseminating
the survey. A total of 12 participants were involved in the pilot study. I conducted
the survey either face-to-face or over the telephone in order to observe participants’
understanding and build the construct items and reliability validity. The results of the pilot study are presented in the next Section 4.4.4.

### 4.4.4 A Pilot Study

This pilot study was conducted to assure the understandability and clarity of the questionnaire, and to test the internal reliability of each group of items for each construct. In the presence of the researcher, or by telephone, 12 Agile team members were asked to answer the survey questions. The researcher’s objective was to make sure the questions were well understood by considering participants’ feedback and comments along with the researcher’s observations. These comments and observations were considered in the development of the final version of the survey questions.

Three main groups of questions were designed and tested in the pilot study. The first group concerns gathering the participant’s demographic data, which are gender, general experience, experience in Agile development, team size, and Agile methods used by the participant. All of these questions, except the questions concerning experience, are multiple-choice questions, in order to make the survey easier for the participant and to reduce the time needed to complete the survey questions.

The second group of questions, concerned with Agile team members’ opinions, consists of various questions types. A matrix table was designed to ask each team member to provide his/her opinion about whether or not affects and affective contagion influence team decisions, cooperativeness, and conflicts, as in Table 4.2 in the form of 5-point Likert scale.

Furthermore, a set of multiple-choice questions was given to the participant in order for them to select all affective states that he/she thinks influence the level of quality of decisions, cooperativeness, and conflicts. The list of all affective states is adopted from the original PANAS scale. Finally, two open-ended questions are presented to the participants in order to obtain their opinions regarding the sources
of work- and non-work-related factors that influence affects and affective contagion in Agile decisions.

The third group of questions concerns measuring the real role of affects and affective contagion in Agile decisions, cooperativeness, and conflicts, are also consists of a matrix table of questions. Two matrix tables were designed, one to measure affect using the I-PANAS-SF scale and the affective contagion scale in the form of a 5-point Likert scale. Another three matrix tables were designed to measure decision quality, cooperativeness, and conflicts in the last project’s iteration that the team member was involved in.

To test the construct validity of the internal items in each construct, we conducted a Cronbach’s Alpha test. Cronbach’s Alpha mainly tests the consistency in a measurement scale to prove that all items under a certain construct are consistent [151]. Table 4.4 shows the results of Cronbach’s Alpha test. The results of the test reside between 0 and 1, while the closer the number is to 1 suggests higher reliability [152]. How to interpret these numbers is explained by George and Mallery, who state that the score > 0.9 is excellent, > 0.8 is good, > 0.7 is acceptable, > 0.6 is questionable, > 0.5 is poor, and < 0.5 is unacceptable [153]. Our pilot study results showed that the construct with the lowest Cronbach’s alpha got 0.77, which falls within the acceptable range.
<table>
<thead>
<tr>
<th>Construct</th>
<th>Number of Items</th>
<th>Cronbach’s alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Affect (PA)</td>
<td>5</td>
<td>0.80</td>
</tr>
<tr>
<td>Negative Affect (NA)</td>
<td>5</td>
<td>0.78</td>
</tr>
<tr>
<td>Positive Affective Contagion (PAC)</td>
<td>4</td>
<td>0.84</td>
</tr>
<tr>
<td>Negative Affective Contagion (NAC)</td>
<td>4</td>
<td>0.83</td>
</tr>
<tr>
<td>Decision Quality (DQ)</td>
<td>5</td>
<td>0.77</td>
</tr>
<tr>
<td>Cooperativeness (COO)</td>
<td>4</td>
<td>0.81</td>
</tr>
<tr>
<td>Conflicts (CONF)</td>
<td>4</td>
<td>0.79</td>
</tr>
</tbody>
</table>

**Table 4.4:** Reliability of the Constructs and their Items

Based on the pilot study, we came up with some observations and feedback. First, an unexpected and surprising observation is that six developers did not clearly understand the word “affect” because they thought of the verb tense meaning which means “impact”; therefore, I included in the survey the meaning of affect by adding the words “emotions”, and “moods” along with providing a clear example of affective states, such as “happiness” and “sadness.’ In addition, for the opinions of Agile team members, a list of almost all core affective states was provided for the participants to choose from, as well as allowing them to write down any other affective state that was not among those listed.

Another observation is that I found that the sequence of questions distracted the developers. The questions were originally organized so that all questions about affects and affective contagion were answered first; then questions about the real measurement of affects and affective contagion were asked. When we changed this, we grouped the questions related to affect (participants’ opinions and actual affective states during the project’s iteration), grouped questions related to affective contagion (participants’ opinions and actual affective contagion states during the project’s
iteration), and then grouped the questions about measuring decision quality, cooperativeness, and conflicts at the end of the survey. We tested the newly organized questionnaire with the last three participants and it enhanced their attention and ability to finish the survey.

Finally, four team members commented on the time they needed to think about the open-ended questions; therefore, we chose to make these optional questions and to investigate them more during the qualitative phase.

4.5 The Adopted Procedure for Quantitative Data Analysis

This section elaborates on how the collected data was processed and analyzed. We started by data screening (pre-processing) to have a valid and ready-to-use data set. The data, then, was analyzed using SEM, which is defined as “a collection of statistical techniques that allow a set of relationships between one or more independent variables (IVs), either continuous or discrete, and one or more dependent variables (DVs), either continuous or discrete, to be examined” [154]. A SEM is composed of a measurement model that links each construct (called latent variables) with its items (called observed variables) and a structural model that links the latent variables to each other in order to test direct and indirect effects [155]. Figure 4.6 demonstrates the general structure of a SEM, showing three measurement models while the link between them is the structural model [5]. Each ellipse represents a latent variable that is connected to many rectangles that represent the observed variables (items) of that latent variable. For example, each ellipse in this research is either an independent or a dependent variable (such as positive affect, negative affective contagion, and cooperativeness), while each of these variables is connected to its items as shown in the Table 4.3. The arrow from A to B and C, and the arrow from B to C represent
the structural model. Therefore, the structural model in this research is represented by both affect and affective contagion constructs (or latent variables), which influence cooperativeness, conflicts, and quality of decisions constructs (or latent variables) in the Agile environment, while cooperativeness and conflicts influence the quality of decisions.

Figure 4.6: Structural Equation Modeling (SEM), Measurement and Structural Models

4.5.1 Data Screening

Cleaning data is an important step to take before testing the model, especially when testing for measurement and structural models, as is the case in this research. In this thesis, I follow a well-defined procedure of checking for missing data, resolving the problem of these missing data, removing outlier responses, and dealing with data normality and linearity. The followed procedure for cleaning data was adopted from Gaskination [156] who explain the need to provide good data to be tested in SEM
and to avoid problems when testing for Factor Analysis (FA) and other tests needed to examine the research hypotheses and models.

### 4.5.2 Testing Validity for a Measurement Model

Exploratory Factor Analysis (EFA) is a variable reduction method that falls under the FA technique. The main goal of EFA is to determine the number of latent variables (called factors or constructs) based on a set of variables [157]. In our quantitative analysis, we conduct EFA on the 31 items mentioned in Table 4.3 in order to extract the latent variables (factors) as depicted in the table. Each set of items must load on its respected latent variable; for instance, positive affect, decision quality, and conflicts. A procedure of deleting the items that threaten the reliability of the scales, i.e. the items that load on more than one latent variable, was conducted. The EFA was carried out using IBM software called Statistical Package for the Social Science (SPSS).

Confirmatory Factor Analysis (CFA) is a part of SEM that tests the measurement model. The main goal of CFA is to evaluate the overall measurement model and construct its validity [158]. Under construct validity, there are some validity tests such as convergent and discriminant validity that have to be considered when conducting CFA, which this research carries out to test the models in Chapter 5. Many authors, such as [158], have suggested that EFA be used as a first step to construct the factors and then CFA be applied to confirm the identified factor structure.

One key difference between EFA and CFA is that CFA needs a predefined model with each construct allocated with its observed variables (items) to test the correlation between the constructs and to provide an evaluation of the overall model fit. The results of the EFA are fed to the Amos software, a module added to SPSS software, in order to build the model and conduct a CFA of the model. Finally, similar to EFA, some items can be deleted or correlated with other items when conducting the CFA.
in order to enhance the measurement model based the overall model fit.

### 4.5.3 Testing Validity for a Structural Model

After testing the measurement model, SEM was used to test the structural model. In other words, it was used to test the hypotheses regarding direct and indirect influences of one construct on other construct, to test the influence of mediations, and to test the influence of mediators (control variables) after testing the items, constructs, and correlations between constructs when EFA and CFA were applied. The path of influence between each independent variable and dependent variables is depicted earlier in Figure 3.3.

To conduct a SEM, five steps have to be followed; namely, they are model specification, model identification, model estimation, model testing, and model modification, as explained by [159]. Model specification is the process of building the theoretical model based on the existing theories and literature. We surveyed the literature in Chapter 2 and highlighted the theoretical models and the relationships between all constructs in Chapter 3. Model identification is the process of finding one solution for each parameter in the specified model. The model can be in one of three states; namely, it can be under-identified, identified, or over-identified. Problems attend the under-identified model, in that there is more than one solution for each parameter. Estimation is the process of estimating the parameters in the model in order to reach as close as possible to the perfect model. The testing model involves the process of examining the analyzed data in terms of their support and fit with the theoretical model. Finally, model modification is the recursive process of modifying the theoretical model when it is not strong enough and evaluating accordingly. Most of these steps are conducted using a SEM software program called Amos, a module added to SPSS software. The steps for conducting the SEM in this research are depicted in Figure 4.7, while the detailed process for conducting these steps is explained in
Figure 4.7: Steps for Conducting Structural Equation Modeling (SEM)

Chapter 5

Finally, when we mention the word “path” in this thesis, we do not mean the path analysis (PA) technique, although it is considered a special type of SEM. However, PA does not consider the error with the measured variable, nor does it consider latent variables. Instead, PA considers observed variables, with one measured indicator for each variable [160]. More importantly, errors can be correlated in SEM while they cannot be correlated in PA. During the reading in SEM and PA, it has been found in the literature that misunderstandings exist between the two models; therefore, we
pay attention to that and clearly note that we apply SEM, not PA, in order to test the research hypotheses and models. However, we mention the words “Path Diagram” to denote the structural model in SEM.

4.6 The Adopted Procedure for Qualitative Data Collection

4.6.1 Sampling Strategy

First, eligible participants for the interviews are Agile team members that include individuals such as the software developer, tester, manager, scrum master, product owner, customer or any other player in Agile teams. However, the eligibility of participants is constrained to Agile team members who have had no less than 5 years of experience. Unlike the quantitative study, we sought to not only confirm the role of affects and affective contagion, but also to understand how their roles are understood and dealt with by Agile team members, whom we expected could provide us with the rich data that we needed to apply the GT methodology and answer the research questions.

To achieve the required number of participants for this study, we decided to conduct either face-to-face or on-line interviews with participants based on their location, with a preference for face-to-face interviews. The required number mentioned in most of the literature, such as by Cresswell [4], ranges between 20 and 30 participants. Therefore, we sought a certificate of approval from the Ethical Board at the University of Regina for conducting either face-to-face or on-line interviews.

The recruitment process started with asking survey participants if they were willing to participate in an in-depth interview. Unfortunately, of the large pool of participants, only five participants showed their willingness to participate and provided
us with their e-mail addresses. We sent e-mail messages to the five participants to arrange the interview, and only three responded, so we conducted the interviews with them (on-line interviews).

The next step for recruiting participants was to post the recruitment poster to the Agile Regina group, which is a group of developers who deliver periodic lectures and presentations to share their knowledge about applying Agile practices in their workplaces. Only six Agile experts were willing to participate. We organized a suitable time for them and we conducted the interviews.

The last step for recruiting interviewees was to post the recruitment poster to various social media platforms, such as Twitter and Facebook, along with spreading it through blogs, colleagues, and friends to reach the required number of interviewees. As a result, we were able to conduct seven more on-line interviews. The total number of interviews was then 16 (six face-to-face interviews and 10 on-line interviews).

4.6.2 Interview Design

The main collection method used in the qualitative research was interviews. We conducted an in-depth interview with each participant. The semi-structured interview was adopted, as it is considered the most-used method in qualitative studies [16]. It is suitable for this research’s objective because of the need to understand how affects and affective contagion influence developers’ and managers’ decisions, cooperativeness, and conflicts.

The interview questions are divided into three main categories (Appendix B). The first category contains demographic information, such as the experience, gender, and role of the Agile team member, to examine whether or not the demographic information influences the results of the study.

The second category of the interview contains seven open-ended questions that are mainly related to the role of affects and their influence on individuals. Within this
category, we would like to understand if the affect of individuals influences his/her decisions or participation in making decisions, along with the influence on his/her cooperativeness and conflicts. Furthermore, a few questions concern how managers deal with the issue of individuals’ negative affects.

Different from the second category, the third category contains questions related to the role of affective contagion and their influence on the whole team instead of individuals. We wanted to understand how the propagation of negative individuals’ affects influences team decisions, hampers cooperativeness, and increases conflicts. The issue of managers’ behaviours toward affective contagion is also investigated within this category.

The last two questions of the interview allow the interviewee to share an incident or story to support his/her arguments, as well as allowing them to add any comments or feedback regarding the role of affect and affective contagion on Agile decisions. Designing the interview questions and interview guides, including revisions based on the pilot study, occurred over a period of six months.

4.6.3 A Pilot Study

We conducted a pilot study to test the interview questions, the interview guidelines, and to refine them accordingly. Two interviewees participated in the pilot study during the winter of 2018. The first interviewee was a university professor in the software engineering department and the second interviewee was a PhD candidate in the computer science department. Both interviewees were aware of and interested in Agile processes, and the main interest of the first interviewee revolves around enhancing tools and techniques for Agile decisions.

After conducting the pilot study, we conducted a main revision that reflected the need to group questions about affects and questions about affective contagion in different categories. The reason for this was because interviewees were confused
by the two terms, “affects” and “affective contagion” so they answered the questions similarly. For example, we asked the interviewee, “what does affect (emotions and moods) means to you when participation in Agile decisions” and we followed this with the question, “what does affective contagion means to you when participation in Agile decisions?”; we found that interviewees answered both questions similarly without considering the meaning of individual affect versus spreading this affect to others. Therefore, we grouped the questions as described in the previous section.

4.7 The Adopted Procedure for Qualitative Data Analysis

For analyzing the interview data, Charmaz suggested using the unified coding process for GT, which involves four phases: initial, focused, axial, and theoretical coding. Further, considering the use of GT, researcher’s observations and notes are very important matter. We briefly describe, below, the four coding phases from [6] and how they were applied during this research phase.

4.7.1 Initial Coding

The coding process generates the main parts that later constitute the theory through theoretical integration. Initial coding is the first step that the researchers take to analyze the data. The initial coding requires researchers to stay open and analyze the data while asking, “what could this data be used to study?”, “what does the data suggest?”, “from whose point of view?”, and “what theoretical category does this specific datum indicate?” [6]. Charmaz indicates that researchers often rely on previous concepts and work on them before they actually start coding. We are aware that previous concepts should not be invoked; therefore, we insist that the survey data will only prove the concept of affect’s influence on Agile decision making.
and help us to decide where to start gathering rich data. However, proven models
of influential factors whereby emotions influence decision-making processes will not
be neglected, as they will help us with theorizing these and other possible factors
within the Agile decision-making area. Points of view can include those of Agile
developers, customers, stakeholders, or others within the Agile environments. The
analytic procedure used during this coding phase goes through intensive word-by-
word, line-by-line, and incident-by-incident comparison and analysis.

4.7.2 Focused Coding

The focused phase takes place to extract significant codes, frequent codes, or both.
Therefore, decisions made during this phase highlight the analytic sense. Charmaz
insists that the process of coding is not linear; instead, the researchers must continue
collecting data, if necessary, until they turn implicit statements into explicit ones. In
this sense, the result of focused coding is to define the best categorizations that make
analytic sense. Interview data are collected from Agile software developers regarding
their decisions, and observations will continue until we make every statement explicit.
Furthermore, factors related to the affects influence on Agile decisions are grouped
under their categories and subcategories.

4.7.3 Axial Coding

The axial coding phase enables researchers to connect categories and subcategories,
and define the dimensions and properties for each category. The axial coding goal,
which was mainly proposed by Strauss and Corbin [136], is to make data coherent
by synthesizing and organizing them after the initial and focused phases. Therefore,
categories and subcategories of affects’ influences on Agile decisions are synthesized,
organized, and sorted.
4.7.4 Theoretical coding

Specifying the possible relationships between all defined categories is the main and most complicated goal in theoretical coding. Charmaz explains that focused codes “may help you tell an analytic story that has coherence. Hence, these codes not only conceptualize how your substantive codes are related, but also move your analytic story in a theoretical direction” [6]. Factors that influence Agile decisions are conceptualized, while all possible relationships between factors are addressed. Figure 4.8 summarizes the coding steps for collecting and analyzing the interviews data.

Figure 4.8: Grounded Theory Coding Process and Phases, Adapted from [6]
4.8 Ethical Consideration

I started to collect quantitative data after I received the approval certificate from the Research Ethics Board (REB) at the University of Regina on April 19, 2018 (Appendix C). One application was submitted to the REB for both the survey and interviews, while different supporting documents for each type of study were provided, such as the poster for recruitment, consent form, and initial contact. When designing and disseminating the survey, we followed the process as accepted by the ERB, and we maintained participant confidentiality and privacy by avoiding acquiring any personal information in the survey. Participants were informed that whenever they submitted their responses, they could not retrieve their data.

We also followed the same procedure explained in the REB application for conducting the interviews. Interviewees who participated in face-to-face interviews signed consent forms. Interviewees who participated in on-line interviews were read consent forms, and their acknowledgement and consent to participate were recorded at the beginning of each interview. Interviewees’ data, confidentiality, and privacy were secured to the maximum level, as instructed by the REB. The interviewees were given up to two weeks after the interview to withdraw their participation, as stated in the consent forms. Fortunately, no interviewee contacted me to withdraw his/her data.

All of the collected data, including audio recordings and the notes made during the interviews, were stored in a locked cabinet accessible only to the researcher. The transcribed textual data were stored, accessed, and processed on a secure (and password protected) computer accessible only to the researcher at the University of Regina. After analysis was completed, the paper materials were shredded and destroyed. The digital data will be kept in a secure location for up to 10 years in order to work on possible publications, and then the data will be erased.
4.9 Chapter Summary

- This chapter provided the reader with the research design, strategies of inquiry, and data collection tools.

- For this research, a mixed method called explanatory sequential design is adopted, which requires the collection of both quantitative and qualitative data.

- The explanatory sequential design requires that the quantitative research must be done first before qualitative research is begun, in order to adequately explain the studied situation.

- The quantitative research is conducted using the survey method that uses questionnaires as data collection tools.

- EFA and CFA were suggested for building and examining the measurement models, while SEM was suggested for building and examining the structural models.

- The qualitative research is conducted using Charmaz’s version of GT.

- Considering the use of GT, the researcher’s observations and notes, along with semi-structured interviews, were proposed as data collection tools in addition to the data collected through the questionnaires.
Chapter 5

Quantitative Analysis

This chapter provides the results of the survey findings. The chapter begins by highlighting the descriptive statistics of the survey. The chapter then explains the results of the survey in four sections. The first section elaborates on the participants’ opinions concerning the role of affects and affective contagion with regards to Agile decisions, cooperativeness, and conflicts. Then, the chapter provides the results of adopting the PANAS scale item and shows which positive and negative affective states have the largest influence on Agile decisions. The third section takes the valid data of the participants’ responses and builds and tests the measurement models by applying the EFA and CFA. These results are given for the three well-examined measurement models, which are focused on the affects, affective contagion, and the affects and affective contagion as the aggregated models. Finally, the chapter builds and tests the structural models using SEM, based on the measurement models. The results for these robust structural models enabled us to examine the proposed hypotheses, which are highlighted in this section as well.
5.1 Descriptive Statistics

Through a comprehensive effort to collect a large number of survey responses, a total of 257 responses were received over the course of three months. Eight responses were excluded from the dataset because they had a high standard deviation, and when examined manually it was found that most of the survey questions were answered similarly. As a result, a total of 249 responses were examined. However, not all of the participants were involved in a project during its last two months. A total of 16 participants were not involved in any Agile development activity during the last two months of a project; therefore, their responses were only considered when examining developers’ opinions about the roles of affect and affective contagion, but they were excluded when building and testing the measurement and structural models. Therefore, 249 responses were analyzed for sections 5.2 and 5.3, while 233 of these responses were analyzed for Section 5.4. Table 5.1 summarizes the demographic information of all participants in this study.

By conducting the Pooled-CFA and adjusting the measurement model, we overcame invalidity and reliability issues, and achieved the best results in terms of the fit indices. Then, we were able to proceed and test our theoretical model and hypotheses.

The demographic variables show diversity in the responses. For gender, around two-thirds of the respondents were male and one-third were female. Experience in Agile development ranged between 0 to 15 years. We divided the experience factor into two classes: expert and novice. We considered any team member who worked less than five years in Agile development as a novice, while five years and more classified them as an expert. There is no cut-off for this classification; however, Yuan et al. defined experts, based on the literature, as people who have over three years of experience, while novices are those who have less than two years [162]. The authors considered people with between two to three years of experience to be novice-experts. However, we only considered the two classes; experts are team members who have
<table>
<thead>
<tr>
<th>Demographic Variable</th>
<th>Category</th>
<th># of Participants</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td>Male</td>
<td>183</td>
<td>73.5 %</td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>66</td>
<td>26.5 %</td>
</tr>
<tr>
<td>Agile Experience</td>
<td>0 - 4</td>
<td>157</td>
<td>63 %</td>
</tr>
<tr>
<td></td>
<td>&gt;= 5</td>
<td>92</td>
<td>37 %</td>
</tr>
<tr>
<td>Agile Method</td>
<td>SCRUM</td>
<td>57</td>
<td>22.9 %</td>
</tr>
<tr>
<td></td>
<td>eXtreme Programming (XP)</td>
<td>125</td>
<td>50.2 %</td>
</tr>
<tr>
<td></td>
<td>Kanban</td>
<td>19</td>
<td>7.6 %</td>
</tr>
<tr>
<td></td>
<td>Lean Software development (LSDP)</td>
<td>14</td>
<td>5.6 %</td>
</tr>
<tr>
<td></td>
<td>Crystal</td>
<td>12</td>
<td>4.8 %</td>
</tr>
<tr>
<td></td>
<td>Feature-Driven Development (FDD)</td>
<td>11</td>
<td>4.4 %</td>
</tr>
<tr>
<td></td>
<td>Dynamic Systems Development Method (DSDM)</td>
<td>10</td>
<td>4 %</td>
</tr>
<tr>
<td></td>
<td>Other</td>
<td>1</td>
<td>0.4 %</td>
</tr>
<tr>
<td>Role</td>
<td>Developer</td>
<td>152</td>
<td>61 %</td>
</tr>
<tr>
<td></td>
<td>Tester</td>
<td>29</td>
<td>11.6 %</td>
</tr>
<tr>
<td></td>
<td>Manager</td>
<td>37</td>
<td>14.9 %</td>
</tr>
<tr>
<td></td>
<td>Customer</td>
<td>15</td>
<td>6 %</td>
</tr>
<tr>
<td></td>
<td>Product Owner</td>
<td>3</td>
<td>1.5 %</td>
</tr>
<tr>
<td></td>
<td>Scrum Master</td>
<td>12</td>
<td>4.8 %</td>
</tr>
<tr>
<td></td>
<td>Other</td>
<td>1</td>
<td>0.4 %</td>
</tr>
<tr>
<td>Team Size</td>
<td>2 - 5 Members</td>
<td>74</td>
<td>29.7 %</td>
</tr>
<tr>
<td></td>
<td>6 - 10 Members</td>
<td>108</td>
<td>43.54 %</td>
</tr>
<tr>
<td></td>
<td>11 - 15 Members</td>
<td>27</td>
<td>10.8 %</td>
</tr>
<tr>
<td></td>
<td>&gt; 15 Members</td>
<td>40</td>
<td>16.1 %</td>
</tr>
</tbody>
</table>

Table 5.1: Demographic Information for valid responses

at least five years of experience and novices are team members who have less than five years. By adopting this classification, we intended to draw a clear line between experts and novices in Agile software development; therefore, unlike in Yuan et al., people who work between two to three years were considered novices. As a result, 157 responses in this survey fell under the novice category, while 92 responses fell under the expert category. For the Agile method factor, around 70% of the participants were from either XP or SCRUM teams. For the position of the team member, developers’ participants constituted more than 60% of the responses, while testers and managers constituted around 25%. Finally, the team size factor showed that around two-thirds of the responses indicated that their team size was less than 10 members, which is the
ideal team size for most Agile processes. For testing the normal distribution, both skewness and kurtosis tests were conducted on all demographic variables. Both test’s results were within the range (-1 to 1) to give a great distribution of the data.

5.2 Results of Team Members’ Opinions of the Role of Affects and Affective Contagion

The 12 questions mentioned in Table 4.2, which were used to acquire the participants’ opinions about whether or not affects and affective contagion play roles in Agile decisions, cooperativeness, and conflicts, are analyzed in this section. The mean for all 12 questions ranges between 3.14 to 3.30. First, we observed the normal distribution of all responses and conducted the skewness and kurtosis tests. No agreed-upon rule of thumb exists regarding an accepted threshold in skewness and kurtosis. In line with [163], we chose a stricter range (between -2 to 2) to test the distribution of the skewness of the data. Both the skewness and kurtosis of the responses to the 12 questions are within the range (-.36 to -.20 for skewness and -1.51 to -1.29 for kurtosis), which indicates good distribution results. However, all skewness and kurtosis results are negative. For the skewness, this means that most of the responses were from the “strongly agree” and “agree” parts of the scale, with a long tail toward the “disagree” and “strongly disagree” parts of the scale [164]. Negative kurtosis means that most of the responses are shaped in a broad distribution around the mean [164].

To test the reliability of the 12 questions and the inter-correlation between each of the three questions related to each construct’s influence on the three dependent variables, we conducted a reliability analysis by testing the inter-item correlation for each construct, as illustrated in Table 5.2. Cronbach’s alpha shows excellent reliability and the inter-item correlation for each construct. We calculated the mean of inter-correlation items for each construct, and all items ranged between 0.63 and
<table>
<thead>
<tr>
<th>Construct</th>
<th>Independent -&gt; dependent</th>
<th>Cronbach’s Alpha</th>
<th>Inter-Item Correlation (Mean)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Affect (PA)</td>
<td>PA -&gt; Decision Quality (DQ)</td>
<td>0.9</td>
<td>0.77</td>
</tr>
<tr>
<td></td>
<td>PA -&gt; Cooperativeness (COO)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PA -&gt; Conflicts (CONF)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative Affect (NA)</td>
<td>NA -&gt; Decision Quality (DQ)</td>
<td>0.9</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td>NA -&gt; Cooperativeness (COO)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NA -&gt; Conflicts (CONF)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positive Affective Contagion (PAC)</td>
<td>PAC -&gt; Decision Quality (DQ)</td>
<td>0.87</td>
<td>0.70</td>
</tr>
<tr>
<td></td>
<td>PAC -&gt; Cooperativeness (COO)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PAC -&gt; Conflicts (CONF)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative Affective Contagion (NAC)</td>
<td>NAC -&gt; Decision Quality (DQ)</td>
<td>0.89</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>NAC -&gt; Cooperativeness (COO)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NAC -&gt; Conflicts (CONF)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: Reliability Analysis for each independent variable

0.80.

Table 5.3 summarizes the results of the participants’ opinions. The Table shows that both “agree” and “strongly agree” scored higher than “disagree” and “strongly disagree.” This agreeable result is concordant for all 12 questions that are related to each relation between a dependent variable and an independent variable. On average, around 53% of the total responses agree about the positive influence of positive affects and affective contagion on decision quality, cooperativeness, and conflicts, and the negative influence of negative affects and affective contagion on decision quality, cooperativeness, and conflicts in Agile environments. Figure 5.1 demonstrates the average answers of all participants in terms of their opinions about the role of affects and affective contagion.
Figure 5.1: Demonstration of the participants’ opinions about the influence of independent variables on dependent variables

Table 5.3: Summary of participants’ opinions about the influence of affects and affective contagion on Agile decisions, cooperativeness, and conflicts
When investigating the data, we paid attention to all control variables and how they might be related to the participants’ opinions. The only variable that showed an influence on the results is the experience of the participant. Of the expert team members, an average of 62.3% agree on the role played by affect and affective contagion, while only 32.5% disagree about the existence of the influence of affects and affective contagion in Agile environments. On the other hand, of the novice team members, an average of 48% agrees about the role played by affects and affective contagion, while 41.9% disagrees about the existence of any influence. When comparing these results to the aforementioned results concerning all participants’ opinions in Table 5.3, we found that novices’ answers (48%) dropped the overall average for agreement to 53%, as compared to the experts’ answers at 62.3%. Figures 5.2, 5.3, and 5.4 demonstrate the variance between experts’ and novices’ opinions of the role played by affects and affective contagion with regards to Agile decisions, team cooperativeness, and conflicts between team members.

Figure 5.2: Experts Vs. Novices Opinions (Decision Quality)
Table 5.4 summarizes the results of the analysis of participants’ opinions regarding the influences of positive and negative affects and affective contagion. All defined items are supported regarding the average of all responses, which underlines the importance of engaging in more in-depth investigations by applying psychological measures and characterizing the influence of affect and affective contagion on Agile decisions.
<table>
<thead>
<tr>
<th>Independent Variable</th>
<th>Dependent Variable</th>
<th>Item</th>
<th>Testing Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Affect (PA)</td>
<td>Decision quality</td>
<td>Positive affects (emotions and moods) enhance the quality of Agile</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>decisions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cooperativeness</td>
<td>Positive affects increase the level of cooperativeness in Agile teams</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Conflicts</td>
<td>Positive affects decrease the level of conflicts in Agile teams</td>
<td>Supported</td>
</tr>
<tr>
<td>Negative Affect (NA)</td>
<td>Decision quality</td>
<td>Negative affects (emotions and moods) decrease the quality of Agile</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>decisions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cooperativeness</td>
<td>Negative affects decrease the level of cooperativeness in Agile teams</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Conflicts</td>
<td>Negative affects increase the level of conflicts in Agile teams</td>
<td>Supported</td>
</tr>
<tr>
<td>Positive Affective</td>
<td>Decision quality</td>
<td>Positive affective contagion between team members enhances the level</td>
<td>Supported</td>
</tr>
<tr>
<td>Contagion (PAC)</td>
<td></td>
<td>of Agile decisions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cooperativeness</td>
<td>Positive affective contagion between Agile team members increases</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the level of cooperativeness</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conflicts</td>
<td>Positive affective contagion between team members decreases the level</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>of conflicts</td>
<td></td>
</tr>
<tr>
<td>Negative Affective</td>
<td>Decision quality</td>
<td>Negative affective contagion between team members decreases the</td>
<td>Supported</td>
</tr>
<tr>
<td>Contagion (NAC)</td>
<td></td>
<td>level of Agile decisions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cooperativeness</td>
<td>Negative affective contagion between Agile team members decreases</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the level of cooperativeness</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conflicts</td>
<td>Negative affective contagion between team members increases the</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>level of conflicts</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.4: Testing Results of Participants’ opinions of the influence of independent variables on dependent variables

5.3 Results of Affective states that Play Role in Agile Development

To understand which affective states influence individuals and teams during the Agile development process, we adopted the original PANAS scale [57]. Participants who acknowledged the existence of affects’ influence on Agile were asked to select which positive and negative affects are the most influential affective states. All positive PANAS scale items were given to the participants: interested, inspired, excited, determined, strong, attentive, enthusiastic, active, proud, and alert. In addition to these items, the item ‘happiness’ was added, which is often used by Agile teams to describe an emotionally positive environment. Participants were allowed to choose more than one item from the list. Figure 5.5 shows the results of participants’ selections of the most influential items. The results show that being determined scored the highest per-
Figure 5.5: Demonstration of the participants’ opinions about the Agile team member’s Positive affects of PANAS Scale that influence individuals

centage (17.65%), followed by being active (12.94%), and being interested (11.76%). Participants selected these three items 217 times. Attentive and inspired were each scored 10.59%, while participants selected them 109 times. The least-selected were happiness of (4.31%), being excited (5.69%), and being proud (5.69%). Participants selected these three items together only 80 times.

Similar to extracting the positive affective states, all negative PANAS scale items were similarly given to the participants: distressed, ashamed, upset, nervous, guilty, jittery, scared, afraid, hostile, and irritable. In addition to these items, the item ‘sadness’ was added, as it is is often used by Agile teams to describe an emotionally negative environment. Participants were allowed to choose more than one item from the list. Figure 5.6 shows the results of participants’ selections of the most influential items. The results show that being upset scored the highest percentage (28.02%), followed by being distressed (25.15%). More than 50% of the influence of negative affects was from these two affective states, and participants selected them 260 times. Hostile and afraid were selected as the third and fourth items (7.16% each), and participants selected them 70 times. Ashamed was the least-selected item (3.07%),
Figure 5.6: Demonstration of the participants’ opinions about the Agile team member’s Negative affects of PANAS Scale that influence individuals

followed by being sad (3.48%), and participants selected them only 32 times.

The 10 positive PANAS scale items, as well as the item happiness, were shown to the participants again, but this time while asking them about the most positive affects of a team member that influence (or are contagious to) other team members. The results in Figure 5.7 show that being determined scored the highest percentage (21.31%), followed by being inspired (12.75%), being attentive (12.58%), and being active (12.45%). Participants selected these four items 352 times. Compared to the results regarding the positive affects that influence individuals, the determined item scored the highest for both questions. Furthermore, being active, attentive, and interested were also within the five highest-selected items. An interesting finding is that while being inspired was the fifth item selected as influential on individuals, it was the second highest item selected for its influence on other team members. The least-selected item was happiness (2.68%), followed by being proud (2.85%) and being excited and alert (5.70% each). Participants selected these four items together only 101 times.

Similarly, the 10 negative PANAS scale items, as well as the item sadness, were
Figure 5.7: Demonstration of the participants’ opinions about the Agile team member’s Positive affects of PANAS Scale that are spread to members shown to the participants again, but this time while asking them about the most negative affects of a team member that influence (or are contagious to) other team members. The results in Figure 5.8 show that being upset scored the highest (28.63%), followed by being distressed (26%). Both upset and distressed items accounted more than 54% of all negative affects, and they were selected by participants 179 times. All other negative items shared the remaining 46%, and the lowest items were being ashamed (1.96%) and being sad (3.33%), which participants selected only 27 times.

5.4 Results of Building and Testing Measurement Models

5.4.1 EFA for Constructing Validity

A total of 233 responses were investigated and analyzed for this and subsequent sections. As mentioned, we excluded participants who were not involved in any Agile development activity during the last two months of a project in order to provide
Figure 5.8: Demonstration of the participants’ opinions about the Agile team member’s Negative affects of PANAS Scale that are spread to members

accurate results via the analysis.

The data were factor analyzed to reduce the dimensions of the responses and to make sure that each set of items appropriately measured the related factor. A total of 31 items were examined for factorability. We adopt the EFA using the principal component analysis as the extraction method and Promax with Kaiser Normalization as the rotation method. Promax with Kaiser Normalization rotation method was selected because the factors were found correlated with absolute values $> 0.32$. When the correlation between factors are high, the literature, for example [165], suggests adopting an oblique rotation method such as Promax. Based on eigenvalues $> 1$, the results suggested seven factors to be the optimal extracted factors, which is equal to the number of constructs in our theoretical model that is introduced in Chapter 3. The extracted factors, as illustrated in the Table 4.3, are PA, NA, PAC, NAC, DQ, COO, and CONF. However, four items were loaded on more than one factor (cross-loading). The researcher decided to remove these items in order to avoid any subsequent threat to the validity when building the measurement and structural models. The removed items were PA5, NA2, DQ2, and CONF4. The remaining 27
items were correlated, and the lowest item scored 0.59. Table 5.5 provides the factor analysis results for the extracted factors. The Kaiser-Meyer-Olkin measure of sampling adequacy is recommended to be above 0.6, and we achieved 0.79. Bartlett’s test of sphericity was significant ($\chi^2 = 2816, df = 351, p < .05$). The seven factors respectively explained 21%, 10%, 9%, 7%, 6%, 5% of the variance, while the cumulative loading of the factors explains 66%.

![Table 5.5: Exploratory Factor Analysis, Pattern Matrix](image)

**Table 5.5:** Exploratory Factor Analysis, Pattern Matrix

Discriminant validity can be achieved by ensuring that no item is highly cross-loaded on more than one factor. To achieve this type of validity in EFA, we
removed the four aforementioned items from the dataset. Two of the eliminated items, PA5 and NA2, are from the I-PANAS-SF scale for measuring team members’ affective states. PA5 was removed because of its cross-loading on two factors, PA and NAC, with loading of 0.5 and 0.4, respectively; meanwhile, NA2 was removed because of its cross-loading on two factors, NA and CONF, with loading of 0.5 and -0.35, respectively. By taking a closer look of these two items, PA5 and NA2 represent the measurement of alert and hostile affective states, respectively. These two affective states were not among the highest priority items when we asked the participants to select the most influential affective states that play a role in Agile environments.

DQ2 was removed because of its cross-loading on two factors, DQ and COO, with loading of 0.6 and 0.3, respectively. Finally, Conf4 was removed because of its cross-loading on two factors, CONF and COO, with loadings of 0.67 and 0.49, respectively. To avoid any threat to further analysis in the subsequent steps and to ensure the availability of enough items for each factor, the researcher decided to drop these items from the dataset.

Almost all items have high loadings, which is a good indicator in terms of ensuring convergent validity. Coo3 scored the lowest loading with 0.59, which is acceptable according to many researchers, such as [166], who highlighted a threshold of 0.50 for each item and an average of 0.70 for each factor. Fortunately, all 27 remaining items are above 0.50 and the mean of lowest factor (COO) is 0.71; therefore, convergent validity is achieved. Eliminating the item Coo3 would raise the mean of the lowest factor (COO) to 0.75; however, we decided to keep this item and monitor it during the building of the measurement model.

5.4.2 Reliability of the Scales

Each set of items, after the specified items were eliminated, was tested to measure the internal consistency and ensure the reliability of each factor. As we did in the pilot
study, we adopted Cronbach’s Alpha to measure the internal consistency and make sure that the data acquired by each set of items involved the right information [167]. Table 5.6 shows the Cronbach’s Alpha for each construct along with its remaining items.

PA and NA Cronbach’s Alpha are 0.82 and 0.84, respectively. This is considered to show a high reliability according to many researchers, such as [152] and [153], who suggested a threshold of 0.70 or above as an indicator of an acceptable reliability. This reliability of PA and NA within software development teams is as good as the reliability of the I-PANAS-SF scale tested by the scale founder, which was 0.84 for PA and the same for NA items. In line with the original founders of the PAC and NAC scales, we obtained a reliability of 0.82 and 0.83 (was 0.93 and 0.91 in the original scale), which provided a high reliability for PAC and NAC, respectively. DQ and CONF provided acceptable levels of reliability, at 0.77 and 0.79, respectively. The only concern is COO, which provided the lowest level of reliability, at 0.70. We re-tested the items to enhance the reliability and found that the reliability of COO would increase to 0.76 if the Coo3 item were removed from the scale. For now, we decided to keep this item and accept this low but acceptable reliability for later investigations in the subsequent sections.

5.4.3 Testing for Correlation (Multicollinearity)

Testing for multicollinearity is important in order to discover whether or not independent variables are highly correlated. When variables are highly correlated, it means that the variables have a multicollinearity issue; i.e., variables can be predicted from others [168]. Of the many techniques one can use to investigate multicollinearity, we adopted bivariate analysis and extracted the Pearson Correlation test values for each pair of variables. The correlation values ranged between (-1 to 1), while an absolute value close to 1 exhibits a multicollinearity issue. Hair et al. suggests that if any two
<table>
<thead>
<tr>
<th>Construct</th>
<th>Coding Number</th>
<th>Item</th>
<th># of Remaining Items</th>
<th>Cronbach's alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA</td>
<td>PA1</td>
<td>Indicate the extent you have felt this way over the past week/weeks of your project’s iteration: Active</td>
<td>4</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td>PA2</td>
<td>Attentive</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PA3</td>
<td>Determined</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PA4</td>
<td>Inspired</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NA</td>
<td>NA1</td>
<td>Indicate the extent you have felt this way over the past week/weeks of your project’s iteration: Upset</td>
<td>4</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td>NA3</td>
<td>Ashamed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NA4</td>
<td>Nervous</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NA5</td>
<td>Afraid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAC</td>
<td>PAC1</td>
<td>The behaviour of the other made me feel: Happy</td>
<td>4</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td>PAC2</td>
<td>Contented</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PAC3</td>
<td>Proud</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PAC4</td>
<td>Enthusiastic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAC</td>
<td>NAC1</td>
<td>The behaviour of the other made me feel: Disappointed</td>
<td>4</td>
<td>0.83</td>
</tr>
<tr>
<td></td>
<td>NAC2</td>
<td>Frustrated</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NAC3</td>
<td>Indignant</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NAC4</td>
<td>Angry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ</td>
<td>DQ1</td>
<td>Final team decisions is of much higher quality than the initial proposals of the individual members</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DQ3</td>
<td>Final team decisions usually extended the quality of team member's individual input</td>
<td>4</td>
<td>0.77</td>
</tr>
<tr>
<td></td>
<td>DQ4</td>
<td>I am confident that the final decision we came up with is the best decision</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DQ5</td>
<td>Overall, it is my opinion that our final decision is of high quality</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COO</td>
<td>Coo1</td>
<td>Other team members usually let me know what they expected from me</td>
<td>4</td>
<td>0.70</td>
</tr>
<tr>
<td></td>
<td>Coo2</td>
<td>I often made suggestions to other team members about better work methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coo3</td>
<td>When I was busy, other team members volunteered to help me out</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coo4</td>
<td>I was willing to help finish work that had been given to other team members</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONF</td>
<td>Conf1</td>
<td>How often do your team members disagree about opinions regarding the work being done</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conf2</td>
<td>How frequently are there conflicts about ideas in your team</td>
<td>3</td>
<td>0.79</td>
</tr>
<tr>
<td></td>
<td>Conf3</td>
<td>How much conflict about the work you do is there in your team</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.6: Reliability of the EFA Factors and their Items

135
variables have a correlation value $\geq 0.90$, they share a multicollinearity issue [169].

Table 5.7 shows the results of testing the correlations between all variables in this research. All correlation values are within the range of -0.34 to 0.352, indicating that no variables in this research have any multicollinearity issues.

![Table 5.7: Correlation coefficients between variables](image)

5.4.4 CFA for Constructing, Testing, and Enhancing the Measurement Model

We started building the measurement model by conducting pooled-CFA using SEM. The steps for this process were followed as in [170] to develop a reliable and validated model with the best data at hand. To make the pooled-CFA model, we combined all latent variables (constructs) in one measurement model using Amos software. The
first test showed a few issues related to unidimensionality, validity, reliability, and the fitness of the model. Many of the issues were related to one or more items having low factor loadings.

Unidimensionality can be achieved by eliminating items that have low loading, one at a time. Some authors, such as [170], have highlighted that items with less than 0.60 should be eliminated. We calculated the factor loading for each item in the model (a total of 27 items). Two items, Coo3 and DQ4, showed low factor loadings of 0.33 and 0.49, respectively. We eliminated Coo3 and DQ4, one at a time, and re-specified the model after each elimination. After the second elimination, all remaining items (25 items) showed a factor loading of no lower than 0.60; therefore, we achieved unidimensionality.

Up to this point, six (19.4%) out of 31 items have been eliminated through the conducting of both EFA and CFA. As advised by [171], the number of eliminated items should not exceed 20% of the total number of items; otherwise, the constructs become invalid. We achieved less than 20% in terms of the eliminated items; therefore, our constructs seem valid and there are no issues with eliminating those six items. However, no more items can be removed from the model, in order to keep the percentage of eliminated items below 20%. Figure 5.9 shows the pooled-CFA conducted for all research constructs and their remaining items, along with the standardized estimation of relationships between all of these constructs.
Discriminant, convergent, and construct validity, along with reliability, were four requirements that we needed to achieve before proceeding to the evaluation of the model fit and the building of the structural model.

When two indicators show a lack of discriminant validity, that means redundant items are present in the model. The first indicator occurs when specifying the model; for example, in Amos, the modification indices (MI) measure the existence
of redundancy between two items by providing a high value ( > 15 as stated by [171]) between these items. The second indicator is that no correlation between two constructs (exogenous ones) should have a high value (<= 0.85 as advised by [171]).

In our model, we tested for discriminant validity, with no sign of the first indicator, where all modification indices values are less than 15. This means that there are no redundant items in the model. Furthermore, for the correlation between all constructs, including exogenous ones, values are between - 0.06 to 0.44 (< 0.85). Therefore, we achieved discriminant validity by making sure that the model has no redundant items or multicollinearity issues between constructs. Moreover, we calculated the maximum shared variance (MSV) of constructs, which is also indicator of discriminant validity. It has been highlighted in the literature that the MSV must be less than the average variance extracted (AVE) for all constructs, which is achieved in this model as shown in Table 5.8.

Average Variance Extracted computes the average variance between latent variables in the measurement model and is the indicator of the convergent validity as well as the reliability of the model. Fornell and Larcker suggest an AVE value of less than 0.50 for each construct, which means that the error is captured more by the measurement model than the construct; hence, the construct and its items are not questionable [172]. We computed the AVE for our seven constructs using the equation provided in [172]. All seven AVE values for the seven constructs are within the range of 0.51 to 0.59, with no construct less than 0.50; therefore, convergent validity was achieved.

In addition to AVE, two other types of reliability needed to be achieved: internal reliability and Composite Reliability (CR). We already achieved internal reliability by providing Cronbach’s Alpha values for each construct when conducting the EFA. The CR can be achieved by measuring the consistency of each construct using the formula found in the literature, such as in [171], which highlights a value of > 70 as
the lowest value acceptable to achieve CR. We calculated the constructs using the CR formula and found that all constructs in our model fall between 0.76 and 0.58, which indicates that we achieved CR. Table 5.8 illustrates the results in terms of the reliability and validity in our measurement model, while Table 5.9 shows the square root of each AVE on the diagonal, and the shared variance between the factors.

<table>
<thead>
<tr>
<th></th>
<th>CR</th>
<th>AVE</th>
<th>MSV</th>
<th>MaxR(H)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COO</td>
<td>0.764</td>
<td>0.519</td>
<td>0.190</td>
<td>0.764</td>
</tr>
<tr>
<td>PA</td>
<td>0.819</td>
<td>0.533</td>
<td>0.171</td>
<td>0.890</td>
</tr>
<tr>
<td>NA</td>
<td>0.850</td>
<td>0.591</td>
<td>0.096</td>
<td>0.941</td>
</tr>
<tr>
<td>PAC</td>
<td>0.825</td>
<td>0.543</td>
<td>0.188</td>
<td>0.955</td>
</tr>
<tr>
<td>NAC</td>
<td>0.830</td>
<td>0.550</td>
<td>0.182</td>
<td>0.963</td>
</tr>
<tr>
<td>DQ</td>
<td>0.784</td>
<td>0.553</td>
<td>0.162</td>
<td>0.969</td>
</tr>
<tr>
<td>CONF</td>
<td>0.795</td>
<td>0.568</td>
<td>0.190</td>
<td>0.973</td>
</tr>
</tbody>
</table>

Table 5.8: Validity and Reliability of the Measurement Model

<table>
<thead>
<tr>
<th></th>
<th>COO</th>
<th>PA</th>
<th>NA</th>
<th>PAC</th>
<th>NAC</th>
<th>DQ</th>
<th>CONF</th>
</tr>
</thead>
<tbody>
<tr>
<td>COO</td>
<td>0.721</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PA</td>
<td>0.343</td>
<td>0.730</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NA</td>
<td>-0.178</td>
<td>-0.064</td>
<td>0.769</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAC</td>
<td>0.434</td>
<td>0.148</td>
<td>-0.152</td>
<td>0.737</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAC</td>
<td>-0.366</td>
<td>-0.414</td>
<td>0.211</td>
<td>-0.171</td>
<td>0.742</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ</td>
<td>0.402</td>
<td>-0.007</td>
<td>-0.203</td>
<td>0.229</td>
<td>-0.163</td>
<td>0.744</td>
<td></td>
</tr>
<tr>
<td>CONF</td>
<td>-0.436</td>
<td>-0.334</td>
<td>0.310</td>
<td>-0.344</td>
<td>0.427</td>
<td>-0.365</td>
<td>0.754</td>
</tr>
</tbody>
</table>

Table 5.9: Factor Correlation Matrix: Diagonal shows the square root of average variance extracted. Off-diagonal shows the shared variance

The final step in testing for validity was to test for construct validity, which can be achieved by having good model fit, i.e. ensuring that the collected data fits in the measurement model. The literature specifies three categories of indicators of model
fitness; each category contains fit indexes. Various authors have specified that at least
one fit index from each category must be achieved in order to have a good model;
otherwise, the model has a poor fit. These categories are absolute fit, incremental fit,
and parsimonious fit. From the absolute fit category, we chose the Root Mean Square
Error of Approximation (RMSEA) and Goodness of Fit (GFI) indexes. From the
incremental fit category, we selected the Comparative Fit Index (CFI) and Tucker-
Lewis Index (TLI). From the parsimonious fit category, we chose the ratio of Chi-
square divided by degree of freedom (χ²/df). We summarized our evaluation of the
indexes that show the goodness of our model along with the accepted level for each
index and its associated literature in Table 5.10.

<table>
<thead>
<tr>
<th>Fit Index</th>
<th>Measurement Model Value</th>
<th>Accepted Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>χ²/df</td>
<td>1.67</td>
<td>&lt; 3.0</td>
</tr>
<tr>
<td>GFI</td>
<td>0.88</td>
<td>&gt; 0.80</td>
</tr>
<tr>
<td>CFI</td>
<td>0.93</td>
<td>&gt; 0.90</td>
</tr>
<tr>
<td>TLI</td>
<td>0.91</td>
<td>&gt; 0.90</td>
</tr>
<tr>
<td>RMSEA</td>
<td>0.052</td>
<td>&lt; 0.08</td>
</tr>
</tbody>
</table>

Table 5.10: Fitness Indices for the measurement model

The required level of acceptance for each fit index differs based on the literature.
We tried to achieve the best fit we could based on the most commonly followed
threshold for each fit index. The literature, as in [173], specifies that the χ²/df should
be less than three. We achieved the stricter threshold of 1.67 for the χ²/df. Notably,
GFI, CFI, and TLI are highlighted by the literature, for example in [174], to be > 0.90
or close to 0.95, as stated in [175]. We achieved the required levels of the thresholds
of 0.93 and 0.91 for CFI and TLI, respectively. Although the GFI value (0.88) did
not reach the stricter required level, we already provided the result of another index
from the same category, which is RMSEA. The literature, as in [176], suggests that
RMSEA should be < 0.08 to acquire a good fit, while a stricter threshold indicates that the index should be < 0.05 to achieve a better fit. We achieved the required level of 0.052 for RMSEA.

Finally, we conducted separate pooled-CFA measurement models for affect and affective contagion models, and tested their validity, reliability, and goodness of fit. We include these in the appendix (Appendix D). Despite it being suggested in the literature that conducting pooled-CFA for all constructs would be enough, we needed the results for each measurement model to compare them with the respective structural model in order to strengthen our findings.

5.5 Results of Building and Testing Structural Models

Achieving validity, reliability, and acceptable goodness of fit in terms of the measurement model enabled us to start building the structural models. We started testing each theoretical model mentioned in Section 3.3 along with its associated hypotheses. The aggregated model will allow for the re-testing of the hypotheses regarding both the affect and affective contagion models, in addition to testing the mediation hypotheses. Furthermore, re-testing the hypotheses using the aggregated model enabled us to understand the correlation between affect and affective contagion, and how they together heighten or lower levels of cooperativeness and conflicts, which, in turn, influence decision quality.

5.5.1 SEM Model of Positive and Negative Affects

We conducted the SEM test for the affect model using Amos software. Figure 5.10 shows the significant path of the theoretical model along with its fit indices using SEM. The first test of the proposed affect model resulted in a great model fit ($\chi^2/df$
= 1.57, GFI = 0.92, CFI = 0.95, TLI = 0.95, and RMSEA = 0.048). However, since we conducted the SEM test using Amos, there was a chance to possibly enhance the model by checking the modification indices suggestions. It has been indicated in the literature, for example in [171], that two factors with a high number (>15) must be correlated; otherwise, not setting the covariance between them as free parameters causes the model to increase the discrepancy. However, we cannot change the model based on all suggested modifications since no single modification can be made without supporting that modification theoretically. Fortunately, modification indices indicated a high number only between COO and CONF, of approximately 15. Since both factors are considered endogenous variables, we only correlated the residuals (error terms) of these variables. The new model resulted in a better model fit ($\chi^2/df = 1.45$, GFI = 0.92, CFI = 0.96, TLI = 0.96, and RMSEA = 0.043).

![Figure 5.10: Structural Model of the Affect (significant paths only)](image)

The casual path effects are explained in Figure 5.11 using both standardized path coefficient estimates and t-value. All endogenous and exogenous variables have an
absolute value of t-value > 1.96. The PA has a significant effect on both COO ($\beta = 0.32$, t-value = 3.85, and $P < 0.001$) and CONF ($\beta = -0.31$, t-value = -3.95, and $P < 0.001$). As expected, PA has a positive significant effect on COO and a negative significant effect on CONF; therefore, hypotheses H1.1 and H1.3 are supported. Similarly, NA has a significant effect on both COO ($\beta = -0.16$, t-value = -2.13, and $P < 0.05$) and CONF ($\beta = 0.30$, t-value = 4.08, and $P < 0.001$). As opposed to PA, NA has a negative significant effect on COO and a positive significant effect on CONF; therefore, hypotheses H1.2 and H1.4 are supported. Furthermore, both COO and CONF were found to be significant factors that influence DQ. COO has a positive effect on DQ ($\gamma = 0.28$, t-value = 3.16, and $P < 0.01$), while CONF has a negative effect on DQ ($\gamma = -0.24$, t-value = -2.84, and $P < 0.01$). Although the influence of COO and CONF is not considered in our hypotheses, it is important to find their significance in order to confirm hypotheses H5 and H7. We chose to investigate these two hypotheses using the aggregated model, in order to highlight whether COO and CONF work as full or partial mediators between PA and NA and the DQ.

![Diagram](image.png)

**Figure 5.11:** Hypotheses Test Results for the Affect Model
5.5.2 SEM Model of Positive and Negative Affective Contagion

Similarly to the affect model, the affective contagion model and relevant hypotheses were tested using SEM. The results in Figure 5.12 show the significant paths of the theoretical model along with its fit indices. Both PAC and NAC are considered exogenous variables in this model. The first test of the proposed affective contagion model resulted in an acceptable model fit ($\chi^2/df = 2.0$, GFI = 0.90, CFI = 0.92, TLI = 0.91, and RMSEA = 0.064). We checked the modification indices and, for consistency, we correlated between COO and CONF residuals errors. This correlation slightly enhanced the model fit ($\chi^2/df = 1.97$, GFI = 0.90, CFI = 0.93, TLI = 0.91, and RMSEA = 0.063).

![Figure 5.12: Structural Model of the Affective Contagion (significant paths only)](image)

The casual path effects are illustrated in Figure 5.13 using both standardized path coefficient estimates and t-value. All endogenous and exogenous variables have absolute an value of t-value $> 1.96$. It was found that PAC has a significant effect on both COO ($\beta = 0.39$, t-value = 4.83, and $P < 0.001$) and CONF ($\beta = -0.28$, ...
t-value = -3.77, and P < 0.001). As expected, PAC has a positive significant effect on COO and a negative significant effect on CONF; therefore, hypotheses H2.1 and H2.3 are supported. Similarly, NAC has a significant effect on both COO ($\beta = -0.29$, t-value = -3.78, and P < 0.001) and CONF ($\beta = 0.38$, t-value = 4.78, and P < 0.001). As opposed to PAC, NAC has a negative significant effect on COO and a positive significant effect on CONF; therefore, hypotheses H2.2 and H2.4 are supported. Furthermore, both COO and CONF are found to be significant factors that influence DQ. COO has a positive effect on DQ ($\gamma = 0.30$, t-value = 3.35, and P < 0.001), while CONF has a negative effect on DQ ($\gamma = -0.23$, t-value = -2.73, and p < 0.01). Although the influence of COO and CONF is not considered in our hypotheses, it is important to find their significance in order to confirm hypotheses H6 and H8. We chose to investigate these two hypotheses using the aggregated model in order to highlight whether COO and CONF work as full or partial mediators between PAC and NAC and the DQ.

**Figure 5.13:** Hypotheses Test Results for the Affective Contagion Model
5.5.3 SEM Model of Affects and its Contagion

The aggregated model for the role of affects and affective contagion was tested using the SEM. Both PAC and NAC were tested as endogenous variables in this model, while they were tested as exogenous in the affective contagion model. We considered a correlation between PA and PAC, and between NA and NAC, as stated in H3 and H4. Therefore, the aggregated model tested these two hypotheses and re-tested the hypotheses that we tested using the previous models. Furthermore, we re-tested hypotheses H5, H6, H7, and H8 to discover whether COO and CONF are mediators (fully or partially) between PA, NA, PAC, and NAC, and the perceived DQ.

Figure 5.14 shows the significant paths of the aggregated model. The results of the first test of the proposed aggregated model showed acceptable model fit ($\chi^2/df = 1.8$, GFI = 0.87, CFI = 0.91, TLI = 0.90, and RMSEA = 0.057). Modification indices proposed a direct effect to be established between PA and NAC. We drew a direct arrow and found the influence to be negatively significant ($\beta = -0.40$, t-value = -5.0, and $P < 0.001$), while the overall model fit was enhanced ($\chi^2/df = 1.69$, GFI = 0.87, CFI = 0.92, TLI = 0.91, and RMSEA = 0.053). On the other hand, no direct effect was found between NA and COO ($\beta = -0.65$, t-value = -1.12, and $P > 0.05$); therefore, the direct link was eliminated. In order to keep our models consistent and to test the role of mediations later on, we decided to keep the model with its acceptable level as in Figure 5.14.
Figure 5.14: Structural Model of the Affect and Affective Contagion (significant paths only)

Figure 5.15 shows the casual path effects using both standardized path coefficient estimates and t-value. The exogenous variables in this model are PA and NA, while PAC, NAC, COO, CONF, and DQ are the endogenous variables. It was found that PA has a positive significant effect on PAC ($\beta = 0.17$, t-value = 1.96, and $P < 0.05$); therefore, this influence of PA on PAC supports H3. Similarly, NA has a positive significant effect on NAC ($\beta = 0.21$, t-value = 2.88, and $P < 0.01$); therefore, this influence of NA on NAC supports H4.

Re-testing the hypotheses regarding the influences of PA and NA on COO and CONF confirmed that all hypotheses are supported except one. It was found that PA has a significant effect on both COO ($\beta = 0.19$, t-value = 2.55, and $P < 0.05$) and CONF ($\beta = -0.18$, t-value = -2.52, and $P < 0.05$). As expected, PA has a positive significant effect on COO and a negative significant effect on CONF; therefore, hypotheses H1.1 and H1.3 are supported again within the aggregated model. It was also
found that NA has a positive significant effect on CONF ($\beta = 0.21$, t-value = 2.97, and $P < 0.01$); therefore, hypothesis H1.4 is supported again within the aggregated model. However, the effect of NA on COO was not significant ($\beta = -0.08$, t-value = -1.07, and $P > 0.1$); therefore, H1.2 is not supported as a direct influence in this model. Since the direct effect was found in the affect model, we will investigate this hypothesis again when testing for mediation, to confirm if NA has an indirect effect on COO through the NAC. To say it in a simpler way, as stated in Baron and Kenny’s method [177], if the coefficient between NA and NAC is significant and the coefficient between NAC and COO is significant as well, there is evidence of an indirect effect between NA and COO. The significant effect of NA on NAC is confirmed, as well as the significant effect of NAC on COO; therefore, there is evidence. However, we chose to apply a more efficient method to test the mediations, called the bootstrapping method, in order to find the significance of the indirect effects between NA and COO, based on a specified confidence level.

It was found that PAC has a significant effect on both COO ($\gamma = 0.38$, t-value = 4.69, and $P < 0.001$) and CONF ($\gamma = -0.26$, t-value = -3.51, and $P < 0.001$). As expected, PAC has a positive significant effect on COO and a negative significant effect on CONF; therefore, hypotheses H2.1 and H2.3 are supported again within the aggregated model. Similarly, NAC has a significant effect on both COO ($\gamma = -0.24$, t-value = -3.22, and $P < 0.001$) and CONF ($\gamma = 0.29$, t-value = 3.85, and $P < 0.001$). As opposed to PAC, NAC has a negative significant effect on COO and a positive significant effect on CONF; therefore, hypotheses H2.2 and H2.4 are supported again within the aggregated model.

COO has a positive effect on DQ ($\gamma = 0.28$, t-value = 3.21, and $P < 0.001$); therefore, the first step of confirming hypotheses H5 and H6 is completed. Similarly, CONF has a negative effect on DQ ($\gamma = -0.23$, t-value = -2.78, and $P < 0.01$); therefore, the first step of confirming hypotheses H7 and H8 is completed. The second
step of confirming H5, H6, H7, and H8 requires investigating COO and CONF as mediators (whether fully or partially) between PA, NA, PAC, NAC, and the perceived DQ.

Figure 5.15: Hypotheses Test Results for the Aggregated Model of Affect and Affective Contagion

5.5.4 Testing for Mediations

To test mediations, we tested each direct and indirect effect in the aggregated model using the bootstrapping method. Table 5.11 summarizes the direct, indirect, and total effects of each variable on the others. The total effect of PA, NA, PAC, and NAC shows a significant influence on DQ ($P < 0.001$), and the total effect of COO and CONF also shows a significant influence on DQ ($P < 0.01$).

To test each specific path effect, Table 5.12 explains each path along with its standardized estimation and p-value. It was found that PA has a strong effect on DQ through COO ($\beta = 0.080$ and $P < 0.05$) and CONF ($\beta = 0.072$ and $P < 0.05$). Similarly, NA has a strong effect on DQ through COO ($\beta = -0.054$ and $P < 0.05$), while a stronger effect was found to exist through CONF ($\beta = -0.082$ and $P < 0.001$).
The significance of these paths confirms the influence of positive and negative affects on decisions; therefore, hypothesis H1 is supported along with its sub-hypotheses.

It was found that PAC has a strong effect on DQ through COO ($\gamma = 0.11$ and $P < 0.05$), while a stronger effect was found to exist through CONF ($\gamma = 0.064$ and $P < 0.001$). It was also found that NAC has a strong effect on DQ through COO ($\gamma = -0.067$ and $P < 0.05$), while a stronger effect was found to exist through CONF ($\gamma = -0.070$ and $P < 0.001$). The significance of these paths confirms the influence of positive and negative affective contagion on decisions; therefore, hypothesis H2 is supported along with its sub-hypotheses.

Although the earlier results showed PA to have a positive significant effect on PAC ($\beta = 0.17$ and $P < 0.05$), while PAC has a significant effect on both COO ($\gamma = 0.38$ and $P < 0.001$) and CONF ($\gamma = -0.26$ and $P < 0.001$), the indirect paths of PA on both COO and CONF show a weak influence through PAC ($P < 0.1$). Considering the indirect effect using Baron and Kenny’s method [177], it is notable that the coefficient between PA and PAC is significant and that the coefficients between PAC and both COO and CONF are significant as well. Therefore, there is evidence of indirect effects between PA and both COO and CONF. Furthermore, the direct and total effects, as shown in Table 5.11, highlight the significant effect of PA on PAC; therefore, H3 is supported whether or not indirect significant paths are found.

On the other hand, a strong indirect path was found between NA and both COO
(\(\gamma = -0.054\) and \(P < 0.001\)) and CONF \((\gamma = 0.063\) and \(P < 0.001\)) through NAC. Therefore, NA’s influence on NAC is found to be significant, along with NAC being a mediator between NA and both COO and CONF; therefore, H4 is strongly supported. Finally, hypothesis H1.2 was not supported in the first test (i.e. the direct effect), while the link between NA and COO is found to be significant through NAC; therefore, H1.2 is supported when considering the indirect effect only.

<table>
<thead>
<tr>
<th>Indirect Path</th>
<th>Standardized Estimated</th>
<th>P-Value</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA -&gt; PAC -&gt; COO</td>
<td>0.059</td>
<td>0.075</td>
<td>Weak evidence of proving the path significance ((P &lt; 0.1))</td>
</tr>
<tr>
<td>PA -&gt; PAC -&gt; CONF</td>
<td>-0.040</td>
<td>0.070</td>
<td>Weak evidence of proving the path significance ((P &lt; 0.1))</td>
</tr>
<tr>
<td>PA -&gt; COO -&gt; DQ</td>
<td>0.080</td>
<td>0.075</td>
<td>Strong evidence of proving the path significance ((P &lt; 0.05))</td>
</tr>
<tr>
<td>PA -&gt; CONF -&gt; DQ</td>
<td>0.072</td>
<td>0.070</td>
<td>Strong evidence of proving the path significance ((P &lt; 0.05))</td>
</tr>
<tr>
<td>NA -&gt; NAC -&gt; COO</td>
<td>-0.054</td>
<td>0.001</td>
<td>Very strong evidence of proving the path significance ((P &lt; 0.001))</td>
</tr>
<tr>
<td>NA -&gt; NAC -&gt; CONF</td>
<td>0.063</td>
<td>0.001</td>
<td>Very strong evidence of proving the path significance ((P &lt; 0.001))</td>
</tr>
<tr>
<td>NA -&gt; COO -&gt; DQ</td>
<td>-0.054</td>
<td>0.018</td>
<td>Strong evidence of proving the path significance ((P &lt; 0.05))</td>
</tr>
<tr>
<td>NA -&gt; CONF -&gt; DQ</td>
<td>-0.082</td>
<td>0.001</td>
<td>Very strong evidence of proving the path significance ((P &lt; 0.001))</td>
</tr>
<tr>
<td>PAC -&gt; COO -&gt; DQ</td>
<td>0.11</td>
<td>0.002</td>
<td>Strong evidence of proving the path significance ((P &lt; 0.05))</td>
</tr>
<tr>
<td>PAC -&gt; CONF -&gt; DQ</td>
<td>0.064</td>
<td>0.001</td>
<td>Very strong evidence of proving the path significance ((P &lt; 0.001))</td>
</tr>
<tr>
<td>NAC -&gt; COO -&gt; DQ</td>
<td>-0.067</td>
<td>0.002</td>
<td>Strong evidence of proving the path significance ((P &lt; 0.05))</td>
</tr>
<tr>
<td>NAC -&gt; CONF -&gt; DQ</td>
<td>-0.070</td>
<td>0.001</td>
<td>Very strong evidence of proving the path significance ((P &lt; 0.001))</td>
</tr>
</tbody>
</table>

Table 5.12: Indirect Paths

Table 5.13 summarizes the results of testing the research hypotheses using the aggregated model. All hypotheses are supported through the analysis of the quantitative data using the CFA and SEM models.
<table>
<thead>
<tr>
<th>Construct</th>
<th>Hypothesis Number</th>
<th>Research Hypothesis</th>
<th>Testing Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affect</td>
<td>H1</td>
<td>The affects of Agile teams have a significant influence on the accuracy achieved in the Agile decision-making process</td>
<td>Supported</td>
</tr>
<tr>
<td>Positive Affects</td>
<td>H1.1</td>
<td>The positive affects of Agile teams have a direct positive significant influence on team cooperativeness</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>H1.3</td>
<td>The positive affects of Agile teams have a direct positive significant influence on team conflicts</td>
<td>Supported</td>
</tr>
<tr>
<td>Negative Affects</td>
<td>H1.2</td>
<td>The negative affects of Agile teams have a direct negative significant influence on team cooperativeness</td>
<td>Supported (indirect paths only)</td>
</tr>
<tr>
<td></td>
<td>H1.4</td>
<td>The negative affects of Agile teams have a direct negative significant influence on team conflicts</td>
<td>Supported</td>
</tr>
<tr>
<td>Affective Contagion</td>
<td>H2</td>
<td>The affective contagion between Agile teams has a significant influence on the accuracy achieved in the Agile decision-making process</td>
<td>Supported</td>
</tr>
<tr>
<td>Positive Affective Contagion</td>
<td>H2.1</td>
<td>The positive affective contagion between Agile teams has a direct positive significant influence on team cooperativeness</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>H2.3</td>
<td>The positive affective contagion between Agile teams has a direct positive significant influence on team conflicts</td>
<td>Supported</td>
</tr>
<tr>
<td>Negative Affective Contagion</td>
<td>H2.2</td>
<td>The negative affective contagion between Agile teams has a direct negative significant influence on team cooperativeness</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>H2.4</td>
<td>The negative affective contagion between Agile teams has a direct negative significant influence on team conflicts</td>
<td>Supported</td>
</tr>
<tr>
<td>Affect &amp; affective Contagion</td>
<td>H3</td>
<td>There is a correlation between positive affect and positive affective contagion in Agile decision making insofar as an individual’s affect positively influences the affective contagion between team members</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>H4</td>
<td>There is a correlation between negative affect and negative affective contagion in Agile decision making insofar as an individual’s affect negatively influences the affective contagion between team members</td>
<td>Supported</td>
</tr>
<tr>
<td>Cooperativeness</td>
<td>H5</td>
<td>Cooperativeness between team members mediates the influence of positive and negative affects on Agile decision quality</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>H6</td>
<td>Cooperativeness between team members mediates the influence of positive and negative affective contagion on Agile decision quality</td>
<td>Supported</td>
</tr>
<tr>
<td>Conflicts</td>
<td>H7</td>
<td>Conflicts between team members mediate the influence of positive and negative affects on Agile decision quality</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>H8</td>
<td>Conflicts between team members mediate the influence of positive and negative affective contagion on Agile decision quality</td>
<td>Supported</td>
</tr>
</tbody>
</table>

Table 5.13: Quantitative Analysis Results of the Research Hypotheses

### 5.6 Chapter Summary

- The chapter highlights the findings regarding the analyzed quantitative data.
- The quantitative data was collected through survey research.
- Two types of data were collected from Agile team members, regarding their opinions and actual involvement in their last project.
• The opinions of Agile team members concerned whether on not affect and affective contagion play a role with regards to Agile decisions, cooperativeness, and conflicts.

• Expert Agile team members confirmed the influence of affects and affective contagion on Agile team decisions, cooperativeness, and conflicts.

• Less-expert Agile team members also confirmed the role of affects and affective contagion; however, their acknowledgement of the role tends to be significantly less than that of expert Agile team members.

• Furthermore, the 20 affective states of the PANAS scale were provided to Agile team members, for them to extract the most influential affective states that play a role in Agile decisions.

• Being determined, inspired, active, and attentive were the four positive affective states that were most influential on Agile individuals and teams.

• Being upset and distressed were the two main negative affective states that were most influential on Agile individuals and teams.

• The second type of data, actual data, were used to test the measurement and structural models of affect and affective contagion using pooled-CFA and SEM.

• The I-PANAS-SF scale was adopted to measure participants’ affective states.

• The measurement models were built and tested against the collected data using pooled-CFA.

• After reaching an acceptable model fit with the CFA, the structural models were built and tested against the collected data using SEM.

• All research hypotheses were supported by testing for the significance of direct and indirect paths.
• The chapter ends by examining the mediators in the aggregated model using the bootstrapping method.
Chapter 6

Qualitative Analysis

This chapter, as indicated during the research design, seeks to confirm the existence of the influence of affects and affective contagion on Agile team’s decisions, cooperativeness, and conflicts. The chapter goes beyond that by extracting and understanding the current process regarding how Agile team members conceptualize, recognize, and act on affects and affective contagion during Agile decision-making processes.

6.1 Descriptive Statistics

A total of 16 Agile team members participated in in-depth interviews. The demographic variables show diversity in terms of their gender, experience in Agile, Agile methods, position of the team member, and team size. Regarding gender, 14 were male and two were female.

Experience in Agile development ranges from five to 20 years. The experience factor in this study is important because we decided that only Agile experts were eligible to participate in this study. We classified any team member who worked five years or more as an expert. There is no cut-off for this classification; however, Yuan et al. defined experts, based on the literature, as people who have over three years of experience. We considered experts to be team members who have at least five years of
experience. By adopting this classification, we intended to draw a clear line between experts and novices in Agile software development; therefore, unlike in Yuan et al., people who worked less than five years were not considered experts. The average of all participants’ years of experience was nine years.

For the Agile method factor, most of the participants were from either SCRUM or Kanban teams. For the position of the team member, scrum masters’ and managers’ participants constituted around 45% of the responses, while testers and developers constituted around 60%. Notice that some participants indicated that they played more than one role and worked on more than one method during their experience. The team size factor showed that around 70% of the participants indicated that their team size was less than 10 members, which is the ideal team size for most Agile processes. Finally, the total number of face-to-face interviews was six, and the total number of on-line interviews was ten.

6.2 The Framework of Affects and Affective Contagion

Analyzing the collected data, we also integrated the researcher memos while adopting the grounded theory, resulting in a general framework that we named “The Framework of Affects and Affective Contagion as Influential Factors on Agile Decision-Making Processes,” as illustrated in Figure 6.1. The framework explains the main concepts/categories that emerged from the data. The framework represents main concepts along with their associated properties, and specifies the links between these concepts. An elaborated framework that illustrates the codes that emerged and their interactions with concepts can be found in the appendix (see Appendix E).
Figure 6.1: The Framework of Affects and Affective Contagion as Influential Factors on Agile Decision-Making Processes

Within the explanation of the framework, its concepts, and properties, we provide confirmation of the research hypotheses that mainly concern about confirming the roles of affects and affective contagion with regards to Agile decisions, cooperativeness, and conflicts. Furthermore, the process of analyzing data enabled us to deepen our understanding of how affects and affective contagion are specifically participating in the processes of making decisions, enhancing cooperation, and managing conflicts. The framework is broken down into its main concepts, to explain each concept along with its links to other concepts and in order to highlight the interactions between
these concepts. We provide quotes from different participants for each concept and its properties to enrich the findings and provide examples from Agile practitioners. Participants’ names are not disclosed; instead, we use P1, P2, etc. to denote the participant whom we quote.

Six main concepts emerged from the data; namely, they are 1. Conceptualizing Affects, 2. Conceptualizing Affective Contagion, 3. Linking Affects to Affective Contagion, 4. Recognizing Situations When Affects Influence Individuals, 5. Recognizing Situations When Affective Contagion Influence Teams, and 6. Acting on Affects to Influence Affective Contagion. Further, one more concept that emerged from the data, which we lightly focused on, is 7. Work and Non Work-Related Factors. Each subsequent section elaborates upon a concept along with its properties and interactions with other concepts within the framework.

6.2.1 Concept One: Conceptualizing Affects

The first concept that emerged from the data is conceptualizing positive and negative affects. Conceptualizing the affects within Agile came from understanding how Agile team members actually understand affects, internalize them, and how they influence their thinking and intentions towards participation in making Agile decisions.

During the iterative process of investigating the concept by aggregating memos, investigating initial and focused codes, and theoretical sampling, we were able to highlight this concept along with all related properties. Both positive and negative affects play roles regarding the concept properties, providing either positive or negative attitude while conceptualizing the affects.

Conceptualizing the affects was found to be mainly related to four properties that emerged from the data; namely, they are willingness to cooperate, willingness to communicate well, making individual decisions, and calling on personal experience. Figure 6.2 illustrates the concept and its properties.
First, the individual’s affects can either increase or hinder an Agile team member’s willingness to cooperate. Willingness to cooperate is defined as the tendency to help others with their tasks, such as sharing knowledge or solving problems. For example, P5 stated that “If one person is not emotionally in the mood to work, or they’re not focused or in the zone, then they might want to work alone for more time than they usually do. Or they might not be focusing as much and they might not be sharing as much of the work professionally—as much as they should otherwise or in perfect conditions.” Similarly, P6 shared about their personal experience and attitude when having negative moods by stating that, “it does influence my cooperation because I would like to avoid people when I’m not feeling up to it. I would like to not talk to anyone, and I’d like to think about my own problems and my own worries instead of talking about various issues with my team members.” Mainly, there are two issues to consider from the emerged data. First, the team member’s desire to work alone increases when experiencing a negative mood or emotion. Second, even when this team member cooperates, he/she tends to not share as much knowledge.
and experiences as when in his/her positive affective states.

Willingness to communicate well means that developers, who tend to communicate in a positive and effective way, are conceptualizing positive affect, while developers who tend to communicate in negative and ineffective ways are conceptualizing negative affects. We have highlighted a series of codes that mention how communication is influenced by requirement dependency; while it creates tensions and other negative affective states. For example, P4 states that, “You have task A, developed or finished by developer A, and another task B depends on task A. If the A is not active, it’s not doing job, it affect the B. The B will be not happy because he’s waiting for the A. And A is not fast, is not finishing the job at the right time. It affects the other developers.” The problem exists when developer A is either “lazy” or “not having the required skills,” as stated by P4. Mentioning happy, as P4 does, suggests more investigation is needed, and P5 illustrated the problem of describing the Agile environment as “happy” or “sad” by noting that, “happiness or sadness are used more as blanket terms to cover for many detailed emotional effects”; therefore, we find that a set of affective states is conceptualized during the problem of dependency.

An interesting finding that needs to be addressed is many participants have mentioned the highest impact of negative affects is related to RE activities. Ultimately, RE is a decision-making process; and the adoption of agile practices in RE activities has changed the nature of this decision-making process into an iterative process. The iterative process in agile practices requires increasing the level and quality of communication between all members in agile processes, including the stakeholders. Participants have highlighted that negative affects hinder their ability to effectively elicit, analyze, specify, validate, and manage the requirements. More interestingly, participants have mentioned that during the iterative process of RE activities, the amount of negative affects play an important role on their trust and behaviour. Therefore, understanding the specific role of affects and affective trust in Agile RE activities
requires more studies and in-depth investigations. We have proposed a model and hypotheses along with the appropriate research methodology in order to investigate these roles which can be found in [178].

By seeking out these affective states, P5 highlighted that the developer who is either lazy or who does not have the required skills for the tasks will carry negative affects, such as being afraid or defensive, while the waiting developers becomes distressed and angry. P4 explained that even when the issue of dependency required a delay in accomplishing tasks not because of being lazy or not having skills, the negative affects of waiting developers continue to develop.

Looking back to the dependency issue, P4 provides a scenario that happens frequently, stating, “Imagine sometimes, some people they choose the easy tasks. And if you don’t speak up, or if you are shy, maybe you don’t choose anything, you end up doing something you don’t know.” In other words, the participant is referring to unfair assigning of the tasks, where developers who have fewer skills and who do not speak up end up taking on complicated tasks that he/she cannot finish or that require more time to complete. Another scenario was given by P14 concerning when developers evaluate user stories and assign the days needed to complete tasks. The participant states that when unfair evaluation was given to a task, the manager either re-evaluated the task or assigned it to other team member who participated to accomplish it. However, many participants related that when a developer is shy to speak out, he/she ends up by conceptualizing negative affects as well as those of developers whose tasks depend on his/her task.

Individual decisions encompass both decisions made by a responsible team member for the group, such as a manager, or participation in group decision making. Two issues are related to individual decisions that have been highlighted from the emerged data; namely, decisions related to continuous learning and decisions related to teaching.
First, the lack of continuous learning, such as learning about new technologies, has been found to be an influential factor. The process of conceptualizing the negative affects comes when a developer resists deciding to apply any new solution that he/she is not aware of. The developer might have good reasons for not deciding on the new solution, or he/she might not want to learn about the new solution and stick to old solutions. In both cases, the resistance of that developer leads them to argue as a team member for not deciding to adopt the new solution or technology. Whenever the team decides for the new solution, the developer’s resistance somehow changes to anger and complaints during the adoption of that new solution. For example, P16, who is expert with almost 10 years experience, said “Fear of change... they don’t want to change. They don’t want to learn. They want to be retired with doing the same thing.”

Second, avoiding teaching new developers is a problematic issue related to many Agile teams. When new members join the team during a project, old team members need time to introduce the project to the new developers. With a high proportion of turnover in the Agile team, both old and new team members conceptualize negative affects. P5 describes one of the consequences of the turnover problem that happened during a project he was involved in by saying that, “We started to avoid more ambitious visual changes in the project. We started to stick with a more traditional things and more simpler things for our game. Some of the ideas of the designers and some of the traditions of the developers even, were refused or ignored. They were refused because of the very reason that we knew that it’s going to take so much more time to train new resources for our project.” As a consequence, the quality of decisions was influenced by the stress created from the time needed to train new members.

Personal experience was found to also emerge as a property. Many participants have mentioned that Agile team members’ points of view are not considered. Some participants explained that many expert developers, who have more experience than
other members while they are all at the same position level, conceptualize negative affective states from the ignorance of their ideas and participation in making decisions. Therefore, the conceptualization of affects by the expert leads him/her to keep complaining throughout the project that the other’s proposed ideas, solutions, or decisions are not accurate, simply because he/she is more expert than them.

6.2.2 Concept Two: Conceptualizing Affective Contagion

Conceptualizing positive and negative affective contagion is another concept that emerged from the data. Conceptualizing the affective contagion within Agile helps us understand how Agile team members actually conceptualize the influence caused by other team member’s affective state. Both positive and negative affective states played roles in the concept properties, providing either positive or negative attitudes while conceptualizing the affective contagion.

The level of team decision accuracy, level of good communication, level of cooperativeness, and level of conflicts are the four properties that characterize conceptualizing the affective contagion. Figure 6.3 illustrates the concept and its properties.
A highly related property has been identified within this framework, which is the accuracy of team decisions. Participants highlighted that the influence of affective states is conceptualized when a decision has been made. In Agile decisions, two issues have been found to be related to conceptualizing the affective states. First, the uncertainty of decisions leads team members to expect that the decision might not be accurate enough, which results in conceptualizing fear and other negative affective states when making that decision. This situation has been highlighted by [74] as the characteristics of the problem at hand that influence the decision maker’s current affective state.

The second issue is related to the affective states that accompany the team member when recalling the results of previous decisions, i.e. the cumulative accuracy of earlier decisions during the project iterations. Most of the participants were managers who experienced team pressure and other negative affective states conceptualized and internalized because of the outcomes of previous decisions. Also, this conceptualized
type of affective state have been defined by [74] as the expected outcome (including expected affective state) that influences the decision maker’s current affective state.

The level of good communication is the property concerning affective states spread among team members that eventually hinder good communication during the project iterations. Participants mentioned different situations regarding the spread of member’s positive or negative affective states to others. First of all, the positive affective state of a team member can open a communication channel with others to discuss, decide, and cooperate. On the other hand, negative affective states lead team members to communicate less and sometimes communicate in an inefficient way. P6 described a simple scenario that usually happens in a positive Agile environment by saying that happy developers always “opens up a communication channel, and that communication channel spreads towards all the team members. And this is really interesting... that means you can come to better decisions.” On the other hand, some negative affective states, such as being angry or shy, lead to a reduced number of communication channels as the negative states are conceptualized from one to another.

Second, the level of good communication is also prone to be shaken during meetings and group discussions when managers’ moods, feelings, and emotions are negative. The negative state of a manager is more conceptualized and contagious to the team members. Although the negative state of any team member can be conceptualized during meetings and discussions by others, team manager’s negative affective states create problems and maximize the possibility of the negative affective states spreading to all other team members, hence creating a negative environment for the whole project. We include a quote from P5 who confirms this by stating that when there is “two or more developers in the same team, and one of them being in a little more crucial role, or a leading role as compared to others, then of course the contagion effect of emotions, or moods between these two people is going to have an effect in their communication with the rest of the team members.”
Level of cooperativeness has been coded as an important property as well. Both the sender and the receiver of the affective state influence cooperation among team members. We have highlighted the situation where the willingness of a team member is influenced by his/her affective state. Furthermore, many participants provided evidence of negative moods and emotions of a team member that not only prevented him/her from cooperating with others, but also prevented others from cooperating with him/her during the subsequent days. For example, P7 explained that negative affects of a team member create a negative impression for the whole team, which influences the level of cooperativeness within the team.

Similar to the level of cooperativeness, the level of conflicts has been characterized as an important property. Conflicts among team members can happen as a reaction instead of being an action. Many participants have highlighted that when a team member with a negative affective state influences other members, most of them conceptualize the negative affects and tend to conflict with that member, which, in turn, increases the level of conflicts. On the other hand, the positive affective state of a team member enables other members to conceptualize that affective state, hence tending to lower the level of conflicts to an acceptable level when discussing and deciding about solutions.

6.2.3 Concept Three: Linking Affects and Affective Contagion

Both concepts, conceptualizing affects and conceptualizing affective contagion, were found to be linked and related to each other. The linkage concept between the two concepts defines the changes that occur when affects or affective contagion are conceptualized. The direction of change can span from conceptualizing affects to conceptualizing affective contagion and vice versa. When an individual or team is influenced by positive or negative affects, there are many implications for conceptualizing these affects. The main properties that have been discovered from the data, which link
affects and affective contagion, are changes in behaviour, changes in members’ involvement in decisions, and changes in willingness. Figure 6.4 illustrates the concept and its properties.

Many authors have discussed the behaviours of Agile team members. However, our property that emerged from the data—namely, changes in behaviour—is defined as changes in behaviour as a consequence of being exposed to an affective state or as a reaction to another's affective state.

Participants acknowledged that their thinking and reactions differ based on both his/her affective state and other team members’ affective states. P8 highlighted that negative affective states can change a member’s behaviour by making his/her decisions more biased. This claim was investigated and two issues were found that can be related to being biased when making decisions; namely, they are the affects of the decision maker and/or changes to negative affects when making a decision because of other members’ affective states. The participants, along with others, acknowledged

Figure 6.4: The Concept of Linking Affects and Affective Contagion and its properties

Many authors have discussed the behaviours of Agile team members. However, our property that emerged from the data—namely, changes in behaviour—is defined as changes in behaviour as a consequence of being exposed to an affective state or as a reaction to another's affective state.

Participants acknowledged that their thinking and reactions differ based on both his/her affective state and other team members’ affective states. P8 highlighted that negative affective states can change a member’s behaviour by making his/her decisions more biased. This claim was investigated and two issues were found that can be related to being biased when making decisions; namely, they are the affects of the decision maker and/or changes to negative affects when making a decision because of other members’ affective states. The participants, along with others, acknowledged
that the bias could be avoided if the decision makers have clear thoughts while they are negatively influenced by negative affects.

Actions and reactions were also found to be influenced by affects and affective contagion. Many participants mentioned that Agile team members act irrationally and react based on their negative affects. Although the majority of participants insisted that affects cannot be completely mitigated, some actions and reactions of team members are considered to be highly influenced by affects that can be mitigated.

Changes in members’ involvement in decisions is defined as the process of considering or neglecting team members’ opinions based on the individual or collective influence of affective states. The individual influence of affective states is mainly related to managers or other responsible positions of the team. Participants concentrated on cases where managers ignored team members’ opinions and stuck to their decisions because of their negative moods and emotions. For example, P6 shared a story about a manager who was experiencing a negative mood by stating that, “because he was in a bad mood, he just decided to scratch two whole weeks of work that we had done. I mean the points he had raised could have been fixed within like a couple of days at maximum.” As a consequence of the manager’s decision, P5 said that the team spent almost three weeks re-building the scratched tasks. The literature has investigated the problem of neglecting team members’ opinions when making decisions and expert and managers using the advantage of their experiences and responsibility to make decisions instead of involving all team members. Thus, this property defines the role of the affective state as an influential factor that hinders the involvement of all Agile team members.

The influence of collective affective states means that when an issue happens within the team, the consequences, along with their associated affects, are spread among members, which could have a collective influence. Many participants identified this type of influence, for instance, in terms of blaming each other, or complaining
about others’ productivity or quality of work. The problem with negative collective states is that they mostly hinder good communication among teams and develop into cumulative negative affects, which can span to other sprints/phases of the project. Managers can play an important role in mitigating the cumulative affective states of their team; this role will be addressed in concept 6.

The change in willingness property is defined as the shift in intention to cooperate and/or communicate. The willingness property has been defined when conceptualizing the affects, and this conceptualized willingness is prone to change when opposite affects are conceptualized within the team. For example, when a team member is conceptualizing positive affects, his/her willingness to cooperate and communicate well rises. However, this willingness can be changed because of conceptualized negative affects within the team, as many participants highlighted.

6.2.4 Concept Four: Recognizing Situations When Affects Influence Individuals

Recognizing situations in which affects influence individuals is an important and concrete concept that is defined as identifying main situations where an individual’s affect arises or changes. Concept one has shed light on how affects are conceptualized, while this concept sheds light on when and where these conceptualized affects occur during the project. The concept encompasses highlighting main situations related to how individual’s affective states arise or change, such as an inability to accomplish tasks, lack of guidance, and neglecting member’s opinion. Figure 6.5 illustrates the concept and its properties.
The skills of team members are considered key to the success of any project. From the emerged data, different issues related to skills are found to constitute a major property of this concept, termed inability to accomplish tasks. Mainly, a lack of required skills and inefficient tasks assignation are the two issues related to this property. First, the lack of required skills can cause many problems to the project, such as time delay. From the individual’s viewpoint, the lack of required skills problem can propagate to internalize the negative affects of fear and shyness. P4 stated that one of the things that he likes best about adopting Agile methods is that the Agile exposes every team member’s work frequently, instead of letting them “work in the dark.” Therefore, an Agile developer’s lack of skills can make them more prone to different negative affective states that lead to decreased levels of confidence, decreased willingness to cooperate, and decreased willingness to communicate well.

Another situation related to skills revolves around the inefficiency of assigning tasks. Every team has different levels of skills and expertise. In some situations,
assigning tasks can be done in such a way that each team member’s skills and expertise are not considered. It has been highlighted that, in many cases, experts and more skilled team members are assigned less complicated tasks, while the less skilled members are assigned more complicated tasks. Different types of negative affects appear during these cases. For example, some participants stated that some developers are shy to speak out while they are stuck accomplishing assigned tasks within the specified time until the last minute. In the end, developers explain that the assigned tasks were more complicated than expected and that they need more time. In fact, many expert participants stated that developers are aware of the obstacles they face from the beginning; however, because they are internalizing their fear and shyness, they prefer to continue while hiding their negative affects.

The lack of guidance for individuals is a property that characterizes when an individual’s affective state turns negative. The lack of guidance means that managers are more concerned about accomplishing tasks within a determined timeline, avoiding interference in team members’ jobs to have the tasks accomplished on time. The lack of guidance has emerged from the data in a similar way as the inefficiently assigned tasks property. When team members find themselves lost somewhere when accomplishing their tasks, hiding the problem from meeting to another, their affects turn negative, such as becoming frustrated and afraid, as the deadline for the tasks to be completed approaches. Despite the fact that Agile methods provide greater transparency than traditional methods, managers need time to find the problems that delay the accomplishment of tasks. Considering the individual’s affective state, manager’s affects can turn to negative, such as anger and stress, whenever he/she discovers that the developer has been hiding the problem for a certain time and did not tell them from the beginning.

Finally, an important situation occurs when managers neglect team members’ opinions when making decisions. Some participants mentioned that managers tend
to consider some team members’ opinions and hear from them while neglecting other members’ opinions. Similarly, when issue arises and the manager hears from the different parties, some developers feel that their explanations and arguments are ignored and disregarded. The consequence of not considering all members’ voices is that it creates negative affective states, such as being upset and irritable.

6.2.5 Concept Five: Recognizing Situations When Affective Contagion Influence Teams

Recognizing situations in which affects spread and influence the whole team is an important and concrete concept that is defined as identifying the main situations where a team’s affects arise or change. Concept two has shed light on how affective contagion are conceptualized, while this concept sheds light on when and where these conceptualized team’s affects occur during the project. The concept encompasses highlighting the main situations related to the arising or changing of teams’ affective states because of individuals’ affective states; namely, these are participating in team decisions, changes in managers’ affective states, and calling on past affects’ influences. Figure 6.6 illustrates the concept and its properties.
Figure 6.6: The Concept of Recognizing Situations When Affective Contagion Influence Teams and its properties

Team decisions are considered to be a main property in this concept, which concerns characterizing the situation when making the decision contributes to influencing team members. Two periods can be considered in this situation: during the making of the decision and after getting the results of that decision. The first situation happens when team members are discussing and deciding about proposed solutions. Of the many factors, the current affective state of a decision maker interferes with his/her choices, which are mostly related to the anticipated results of the made decision. When the decision maker experiences a negative affect regarding solutions at hand, the resistance toward those solutions increases and leads to complaining about achieving the results. The continuous complaining creates negative affects for the whole team and created either less trust regarding the results or less affective trust and anger toward the complaining developer.

The second period in the situation concerns after the results of the decision are clarified. When a team decides against an individual without discussing his/her opin-
ion and the results of that decision do not provide an acceptable solution, an individual’s negative affective states, such as being angry and hostile, are conceptualized and called upon for any subsequent decision. On the other hand, when the decision provides an acceptable solution, the stability of that developer’s affects is affected, and that leads to the conceptualization of fear and shyness for future participations in team decisions.

The past experiences of team members’ regarding behaviours and affects constitutes their overall willingness to initiate good communication and cooperation within the team. Participants discussed situations in which team members do not have the willingness to communicate well or cooperate with others because, sometimes, they recall past experiences of others’ bad behaviours related to negative affects. Recalling past affects when connecting with other team members can take different forms. First, the team member can communicate and discuss things with others in a perfect way, but he/she resists cooperating with others because of past experiences of negative affects. P2 described the chain of actions followed by many team members when, for example, member A seeks help and cooperation from team member B who refuses to cooperate because of member B’s current negative affects. Member A internalizes a bad experience and has less affective trust toward member B. In turn, whenever member B seeks cooperation, member A refuses to cooperate based on an unwillingness to cooperate related to that level of affective trust and bad experience. P4 simply summarized that team members’ actions and reactions towards each other are guided by revenge, whereas the trigger for that revenge can be the level of negative affects among them. The worst form of connecting with others was highlighted as being when negative affects influence a team member and keep him/her focused on the assigned tasks without concern for good communication and cooperation with others. This property creates a problematic issue for managers who cannot find the root causes of the negative affects.
On the other hand, the positive affects of a team member who provides good communication and cooperation to other team members enables him/her to find cooperation and help from others whenever he/she seeks it. Many participants acknowledged that when a team member’s positive affects are more than their negative affects, the tendency to cooperate and communicate increases. An interesting finding is that whenever that team member conceptualizes negative affects and seeks cooperation, other team members are highly likely to cooperate and communicate with him/her to help overcome the issue being faced. Participants also mentioned that the level of tension among team members increases as they hold negative attitudes toward others. It has been found that the conceptualized negative affect that occurs during early conflicts is recalled during the subsequent conflicts, which increases the level of conflicts among team members. On the other hand, the evidence shows that positive affect increases the level of consensus among team members.

Despite the fact that the responsibility of managers and leaders is to manage Agile teams and assure smooth progress, changes in their affects contribute to the spreading of either positive or negative affects to team members. A simple and common issue that changes managers’ affective states to either positive or negative is mainly related to the outcome of the iteration/sprint, such as achieving high quality in tasks while staying committed to the timeline. Managers have been identified as an influential factor that either positively or negatively changes the team’s affective state. Most participants acknowledged that managers are happy as long as the team is progressing, but whenever the team is stuck on certain issues, the manager’s mood changes and that influences the team’s moods, which creates additional obstacles.

It also has been highlighted that the chain of actions of team members can influence a manager’s affective state. A manager’s behaviour is a property in this concept that illustrates the situation when a manager’s affect influences the whole team. Different situations can occur as a chain of actions that ultimately influence the team’s
affects through the changing of manager’s affect. First, it has been highlighted from the data that some lazy or unskilled developers tend to deceive managers by faking or exaggerating the progress achieved. P14 emphasized that managers are usually not technical experts and that developers can deceive them; however, technical managers can identify that deception faster than non-technical managers. However, personality and culture play roles that either hide or reveal deception in front of the manager. When the manager discovers this deception, his affective state turns to negative toward the deceivers and, sometimes, toward the whole team.

6.2.6 Concept Six: Acting on Affects to Influence Affective Contagion

The last main concept that emerged from the data encompasses the current acts taken toward positive and negative affects within the Agile team. Current actions on negative affects include one of four types of actions: two of these types—namely, controlling and mitigating—decrease the sustainability of negative affects and their consequences, while the other two types—namely, ignoring and exaggerating—increase the sustainability of negative affects and their consequences. On the other hand, the current actions on positive affects can be one of two types—namely, optimizing, which leads to increased sustainability of the positive affects and their advantages, and ignoring, which decreases the sustainability of the positive affects and their advantages. Figure 6.7 illustrates the concept and its properties.
Controlling negative affects is an act taken by Agile managers and expert team members. Controlling negative affects’ influences requires that the manager or team member be able to recognize them in the early stages. The current technique for controlling negative affects’ influences is to assure that the decision made or the participation in team decision making is not influenced by negative affects. P3 emphasized the need for recognizing and being transparent about your negative affects. The participant said that, “Even though I am feeling angry, but I am going to put that aside, and we are going to this meeting, and try to come up with the best decision, even if I am feeling frustrated, I should recognize the fact and be transparent about it.” Being transparent about negative affects allow managers and team members to keep negative affects under control during meetings, discussions, and decision making.

Mitigating negative affects is an act taken by managers and team members so that these negative affects and their consequences are minimized. Negative affects appear in Agile due to different reasons, such as time pressure, complicated tasks,
and conflicts during interactions among team members. Expert participants reported that although the consequences of the negative affects cannot be completely isolated, they adopt various strategies to mitigate them.

Some participants insisted on the need for one-on-one discussion with any team member that the manager thinks shows negative affect. The discussion enables managers to discover the root factors that caused that negative affect and then mitigate them by handling these factors, especially if they are related to work, such as other team members' behaviours or affects. P7, who adopted individual meetings with his team members, suggested that Agile managers conduct feedback sessions in order to understand and recognize the current affective states of their team members, enabling the managers to mitigate the consequences of negative affects.

Negative affects' influences can also be minimized during meetings and when making decisions by ensuring that all opinions and ideas are heard and evaluated within the team. This is mostly the responsibility of team managers. This procedure ensures that the negative affects of team members are not maximized due to his/her opinions and ideas being ignored. For example, P3 illustrated this by giving an example of when a team wants to evaluate multiple solutions. The participant said that, “there are going to be negative emotions, anger, frustration, or helplessness, so this will then override, a more sensible approach of being transparent, like why you are promoting one decision over another decision. The person with the loudest voice can win, which might not be the best decision to be made.” The participant concluded that these negative affects play a role and can also influence members whose voices were not heard. Therefore, by allowing all team members to express their opinions and voices, their negative affects' consequences in terms of feeling neglected will diminish, and they will believe that their voices are as valuable as everybody else’s.

Relating dependency to negative affects can be avoided or minimized by accounting for at least one of the two solutions. First, managers need to assign tasks to
people who are willing to do them, such as assigning developers who are shy to speak in public to tasks that they like and are able to accomplish. This can create initially positive team affects, thus minimizing the dependency problem. Second, adopting the most accurate estimation techniques for the tasks to avoid giving tasks less time than they should be given. This can lead to the avoidance of the negative affects of waiting developers by acknowledging them with the most accurate time in which it the required tasks are expected to be done.

Another type of action is the tendency to ignore affects. Within this type, we have highlighted an important paradox that many participants shared with us. The paradox can be explained as follows: Agile team members believe that negative affects should not exist as influential factors in professional workplaces, such as the software industry with its professionals and experts. On the other hand, participants, most of them being experts in Agile, acknowledged that negative affects play roles in Agile environments. After investigating this paradox in depth, it has been found that although negative affects influence Agile teams, many developers and managers tend to ignore this influence because of their belief in their professional workplaces; therefore, they tend to ignore any negative affects. For example, P7 recognized the role of negative affects, but he noted that professional workplaces should not be influenced by factors such as emotions, moods, and feelings. At the end of the interview with P7, he said: “I will try to transfer this to my team and instill new energy in them, which will help them for making some better decisions and will achieve bigger results in our team.” The participant meant that after he was thinking about the paradox, he decided to consider affects as a potentially influential factor that needs to be addressed in his development team.

Exaggerating negative affects’ consequences is the action that leads to increased negative affects among individuals or teams along with their consequences. We have mentioned the scenario where a team member conceptualizes but hides negative affects
because of task difficulty, which leads to deception of his/her manager. The action of the manager toward the deception can lead to either the reduced influence of that member’s negative affects or their exaggeration. In addition, when the manager discovers the deception, he/she can recognize negative affects and spread them to influence the whole team members if no actions in terms of controlling or mitigating are considered in that moment.

Similarly, exaggerating negative affects’ consequences can occur within the team. For example, the negative affects of a team member are increased whenever he/she lacks guidance and help from others. Furthermore, the overall level of interaction and communication among team members lack involvement of that team member who recognized negative affects while seeking good cooperation in the past.

The last action in this concept revolves around the current actions for optimizing the influence of the positive affects. Many participants related positive affects to the acts of motivating and encouraging. Managers’ positive affects have been highlighted as a trigger that leads them to encourage, motivate, and enthuse team members. Similarly, expert managers tend to optimize the influence of a team’s positive affects by mentioning and appreciating their efforts in terms of productivity, communication, cooperation, and their interactions. Furthermore, these managers tend to memorize points in order to raise them again during future tasks. Finally, as P10 stated, “being a manager, you should understand the behavioural changes and the moods and emotions or pure team members first and once you understand those things, once you cater those things, I think it would definitely would have a lot of impact on their performance in a good way possible.”

6.2.7 Concept Seven: Work and Non Work-Related Factors

The work and non-work-related factors concept is the broad concept that encompasses all factors that influence Agile team members’ affective states. Notice that work-
related factors are the ones that influence the affective states of teams, but not the factors that influence them from inside the project or those related to it because these factors have been already identified and discussed within previous concepts. Figure 6.8 illustrates the concept and its properties.

![Figure 6.8: The Concept of Work and Non Work-Related Factors and its properties](image)

Work-related factors identified by the participants include organizational policies, management, and motivations. First, participants mentioned that organizational policies could influence the team’s affective states. An example of these policies is the dealing with the problematic issue of employee turnover during projects. Employee turnover creates a severe problem in software companies because of the need for skilled team members who can complete tasks in order to help accomplish the project within the required timeline. P15, for example, concluded that a company that faces the problem of turnover looks only at the skills of employees at the beginning, while neglecting other indicators of whether or not the employee will work a long time for the company. We asked one participant about these indicators, and he stated that,
for example, culture and personality differences with current team members are such indicators. Therefore, teams that have a high level of employee turnover are found to conceptualize and recognize more negative affective states than positive ones.

Second, many participants mentioned that the management process and time allowed for making tactical decisions related to projects can influence the team’s affects. Some participants complained about the lack of flexibility in making decisions, as many organizations either lack the experience of adopting Agile as they moving forward to it, or adopt only some practices in Agile. For example, P13, who is a scrum master, shared the steps that her company has taken for adopting the Agile process. However, the participant complained about the hierarchy of the decision-making process that still hinders the full adoption of the Agile process as they are moving toward it. The participant said that, “Moving toward full Agile process takes time; while projects accomplished through this time holds some drawbacks of not making decisions so fast.” During the discussion, the participant mentioned that affective bonds among team members and managers are influenced when the need to make fast decisions is restrained.

Salaries and bonuses that assure Agile members’ loyalty to a company are two factors that influence a team’s affects. Skilled team members expect a high salary that matches their skills and expertise. Whenever a team member is satisfied/dissatisfied with his/her salary, the team member’s thinking about their high/low salary and how much work and cooperation should he/she provide as compared to that salary influences his/her affective states and loyalty to the company. Furthermore, motivations in the form of bonuses play an important role in Agile teams, helping to increase the level of loyalty to the company while also enhancing the affective states of the team members.

Non-work-related factors that influence Agile team member’s affects are essential. Participants mentioned many factors, such as family issues, transportation issues,
and psychological issues such as anxiety and depressions. Although these factors are external factors that, mostly, cannot be quantified and dealt with within the workplace, their influences on affects can be avoided or mitigated. For example, some participants suggested excusing team members who are facing any type of negative affects from morning meetings sometimes, until the member’s affective state recovers and they become able to communicate and cooperate effectively.

6.3 Chapter Summary

- The chapter highlighted the findings of the analyzed qualitative data.

- The qualitative data was collected through semi-structured interviews.

- A total of 16 in-depth interviews were collected and analyzed, while Charmaz’s version of GT was adopted throughout the course of data collection and analysis.

- Seven concepts emerged from the data, along with their properties and interactions among them.

- Concept one conceptualizes positive and negative affects, which was characterized using four properties; namely, willingness to cooperate, willingness to communicate well, calling on personal experience, and making individual decisions.

- Concept two conceptualizes positive and negative affective contagion, which was characterized using four properties; namely, they are level of good communication, level of cooperativeness, level of conflicts, and level of team decision accuracy.

- Concept three links affects and affective contagion, and it was characterized
using three properties; namely, they are changes in behaviour, changes in members’ involvement in decisions, and changes in willingness.

- Concept four recognizes situations in which affects influence individuals, and it was characterized using three difficulties that individuals can face; namely, they are the inability to accomplish tasks, lack of guidance, and neglecting individual’s opinions.

- Concept five recognizes situations in which affective contagion influence teams, and it was characterized using three properties; namely, they are participating in team decisions, calling on past affects’ influences, and changes in managers’ affective states.

- Concept six acts on affects to influence affective contagion, and it was characterized using five properties; namely, they are optimizing, controlling, mitigating, ignoring, and exaggerating.

- Concept seven is work- and non-work-related factors, and it can be characterized using various properties for work-related factors, such as organization polices, management, and motivations, and various properties for non work-related factors, such as family issues, transportation issues, and psychological issues like anxiety and depression.
Chapter 7

Mixed Methods Interpretation and Discussion

The goal of this chapter is to discuss the results of both the quantitative and qualitative studies. The chapter discusses the results based on combining, comparing, and explaining the results for each set of hypotheses, along with their relevant research question that investigates the influence of affects and their contagion on Agile decision quality, cooperativeness, and conflicts. To provide a clear and effective discussion of the results, affects and affective contagion are discussed separately when characterizing their roles and influence on Agile team decisions, cooperativeness, and conflicts. The chapter ends by discussing the general hypotheses and main research question, which involves a discussion of the main results of the aggregated model for the quantitative study, as in Figure 5.15 and the general framework for the qualitative study, as in Figure 6.1.

7.1 The Role of Affects in Agile Decision Quality

The main hypothesis was proposed to test the existence of affects’ influence on Agile decisions quality:
H1: The effects of Agile teams have a significant influence on the accuracy achieved in the Agile decision-making process.

The process of testing this hypothesis began with acquiring the Agile team members’ opinions and thoughts regarding whether or not positive and negative affects enhance the quality of Agile decisions. Two statements were provided to the participants to choose if they agreed or disagreed with these statements. The statements stated that positive/negative affects (emotions and moods) enhance/decrease the quality of decisions. The statements were used to support the idea that the hypothesis is adequately relevant and worth testing within the Agile decision-making processes. Statistically, 53% of participants agreed to the role of positive and negative affects. Only 38% of participants disagreed that positive and negative affects play a role. About 9% of participants neither agreed nor disagreed with the statements.

It has been found that the experience factor played a role in participants’ responses regarding the influence of positive and negative affects on the quality of Agile decisions. Most of the 38% were novices in Agile development, while the majority of the 53% were experts in Agile development. This significant difference between novices’ and experts’ opinions opened the door to investigating the hypothesis in a more efficient way.

The hypothesis was tested through collecting and analyzing the responses by adopting the EFA, CFA, and SEM statistical tests. The results showed significant paths between positive affects and decision quality through two mediators—namely, cooperativeness and conflicts—as illustrated in affect model [5.11]. Similarly, significant paths between negative affects and decision quality were found through the same mediators; therefore, the main hypothesis H1 was supported.

The confirmed hypothesis along the significant finding of differences between experts and novices prompted us to proceed and investigate the following research question:
RQ2: How do affects and their contagion influence the accuracy of decisions made in Agile?

Note that we consider the discussion of the affects’ influence in this section, while we postpone the discussion of the affective contagion until the next section.

The findings in Chapter 6 answer this question by highlighting the concepts and their correlations that participants suggested either enhance or decrease the quality of decisions.

Affects influence the quality of individual decisions starting with the conceptualizing of affects, where the individuals’ willingness to provide a good level of cooperativeness and communication is either increased or decreased. The first decision when conceptualizing positive/negative affects is whether or not to cooperate and communicate with other team members. This individual decision can either enhance or decrease the quality of the decision, along with when and how to cooperate and communicate.

The individual’s decision to adopt new solutions is related to his/her willingness to engage in continuous learning, which has been identified to be influenced by positive and negative affects that are conceptualized by the individual. On the other hand, deciding on continuous teaching and helping other team members is related to the expert’s willingness to cooperate, where the conceptualized affects can become the dominant factor influencing that level of willingness.

The conceptualized negative affects can be recognized when the Agile team member is experiencing difficulty while accomplishing tasks. Different factors work together to hinder the achievement of a high decision quality, such as the lack of required skills and lack of guidance. Furthermore, the quality of decisions is affected by the overall rate at which the team member’s opinions and ideas are considered throughout the project life cycle. The individual’s decision quality is influenced by feelings about his/her productivity and involvement in the team. Interestingly, the
sustainability of negative or positive affects of an individual tend to influence his/her individual decisions, which was found to be mainly related to whether his/her opinions and ideas are considered or neglected during meetings and regular Agile discussions.

The individual or the manager can act on the individual’s affects. The individual acts toward the conceptualized and recognized positive affects by optimizing them. The optimization of the positive affects allows the individual to positively influence his/her level of willingness to learn, which, in turn, increases the willingness to acquire knowledge and understand more about decision choices and their possible results. Unfortunately, optimizing positive affects can be achieved by keeping the individual’s positive mood unchanged, which is a difficult task because of the many project issues, such as the dependency problem.

On the other hand, an individual’s decisions can be enhanced when individual control or mitigate the negative affects as much as possible. Unfortunately, many are unaware of ways and techniques for controlling or mitigating the negative affects of individuals, as managers are often more concerned about the results and outcomes of the individual’s decisions, which can lead to poor decisions being made by individuals. Managers can control and mitigate the negative affects of individuals by providing guidance and help for the individuals in a sustainable way. The manager also can play a role in the individual’s decision by ensuring that the individual is not influenced by any known negative affects.

Therefore, the role of positive and negative affects on the quality of individuals’ decisions was confirmed quantitatively by surveying participants’ opinions and collecting data on their real involvements in Agile projects, where these data were used to confirm significant paths between affects and quality of decisions. Furthermore, this role was qualitatively investigated and understood through three concepts, which are conceptualizing affects related to the individual’s willingness to learn and teach continuously; recognizing the situations where these affects occur, such as when there
is a lack of guidance and skills; and acting on the positive and negative affects to op-
timize or avoid their influences on individual’s decisions. The properties and concepts
connect and collaborate to participate in either enhancing or decreasing the quality
of an individual’s decisions.

7.2 The Role of Affective Contagion in Agile Decision Quality

The main hypothesis to test the existence of the influence of affective contagion on
Agile decision quality was as follow:

H2: The affective contagion between Agile teams has a significant influence on
the accuracy achieved in the Agile decision-making process.

Similar to the previous process used to test the hypothesis, two statements were
provided to the participants to choose if they agreed or disagreed with these state-
ments. The statements stated that positive/negative affective contagion between
team members enhance/decrease the level of Agile decisions quality. A total of 53%
of participants agreed to the existence of positive affective contagion’s influence on
the quality of Agile team decisions and 35% of participants disagreed. Similarly,
55% of participants agreed to the existence of negative affective contagion’s influence
and 36% disagreed. Furthermore, 12% and 9% of participants neither agreed nor
disagreed with the positive and negative affective contagion statements, respectively.
The results indicated that negative affective contagion constitute a problematic is-
ue that most participants are aware of. The experience factor played a role in the
responses regarding the influence of positive and negative affective contagion on the
quality of team decisions. Most of the 35% and 36% participants who disagreed to
the positive and negative affective contagion roles, respectively, were novices in Agile
development, while the majority of the 53% and 55% were experts in Agile develop-
This significant difference between novices and experts opened the door for a further investigation of the hypothesis in a more efficient way.

The results of adopting the SEM model to test participants’ data showed significant paths between positive affective contagion and the Agile team’s decisions quality through two mediators—namely, cooperativeness and conflicts—as illustrated in the affective contagion model. Similarly, significant paths were found between negative affective contagion and the Agile team’s decisions quality through the same mediators; therefore, the main hypothesis H2 was supported.

The results of testing the hypothesis and the significant difference between expert and novice responses were involved in investigating how the influence of affective contagion occurs, by investigating the following question qualitatively:

RQ2: How do affects and their contagion influence the accuracy of decisions made in Agile?

The level of team decision accuracy emerged as a property in conceptualizing affective contagion. This level is influenced by the uncertainty regarding the decision at hand, along with expected outcomes and the current affective states of team members. When making decisions, affective states are conceptualized and transferred from one team member to another. The spread of affective states is recognized deeply when team members anticipate the results of their ideas and participation to make decisions, along with recalling the experience of making previous decisions that the team members participated in and their outcomes.

Therefore, the sequence of affective contagion influencing the quality of team decisions starts by conceptualizing the spread and contagiousness of the affective states when teams make decisions under uncertainty, along with considering the current affective state of the team. Then, the anticipated results of individuals’ involvement in team decisions are recognized with their related affective states, which can be recalled by the team during future decisions.
The end of the sequence is the determinant of the influence of affective contagion on Agile team decisions, i.e. the acting on the team’s affects. Managers and team members can optimize or ignore the positive affective contagion’s influences, and control or mitigate negative affective contagion, or at worst exaggerate them. Enhanced visualization of the problem, highlighting all possible outcomes and what to expect when using an alternative approach over others, and having a comprehensive evaluation of all alternatives have been found to enhance the quality of team decisions as well as less negative affects can be conceptualized at the team level. The managers can also mitigate the current affective states when they notice any tensions or other negative affects at the team level. Managers who use urgent strategies and mitigation techniques as soon as they notice these negative affects are found to experience decisions with a higher level of quality.

Finally, the influence of positive and negative affective contagion on team decisions’ quality was confirmed quantitatively by surveying participants’ opinions and collecting data on their real involvements in Agile projects, where this data was used to examine significant paths between affective contagion and the quality of team decisions. Furthermore, this role was qualitatively investigated and understood through three concepts, which are conceptualizing affective contagion related to current team’s affects during decision process and expecting their results and uncertainty; recognizing the situations when these affects are conceptualized as contagious among team members, which is mainly related to past experiences of participation in team decisions; and acting on the positive and negative affective contagion. These properties and their associated concepts connect and collaborate to either enhance or decrease the quality of team decisions.
7.3 The Role of Affects in Agile Teams’ Cooperativeness

Two hypotheses were provided to test the existence of affects’ influence on Agile team cooperativeness. The two hypotheses were as follows:

H1.1: The positive affects of Agile teams have a direct positive significant influence on team cooperativeness.

H1.2: The negative affects of Agile teams have a direct negative significant influence on team cooperativeness.

Following the same testing process, the two hypotheses were examined against Agile team members’ opinions and thoughts. Two statements were provided to the participants to choose if they agreed or disagreed with these statements. The statements stated that positive/negative affects increase/decrease the level of cooperativeness in Agile teams. The statements were used to support the idea that the hypotheses are adequately relevant and worth testing within the Agile decision-making processes. Statistically, 56% of the participants agreed to the role of positive affects and 58% agreed to the role of negative affects. An average of 36.5% of participants’ disagreed about the role of positive and negative affects. Only around 6% of participants neither agreed nor disagreed with the statements. Experience was highlighted as a significant factor influencing their responses, which led us to test the hypotheses using more accurate statistical models such as SEM. Both hypotheses H1.1 and H1.2 were supported, while the results showed a direct significant path between positive affects and cooperativeness, as illustrated in the affect model [5.11]. Similarly, a direct significant path between negative affects and cooperativeness was also found. The confirmed hypotheses allowed us to seek more information in order to understand how these direct influences occur; therefore, the following research question was introduced and investigated qualitatively:
RQ3: How do affects and their contagion influence cooperativeness between Agile team members?

For the rest of this section, we discuss the affects’ influence, while we postpone the discussion of the affective contagion’s influence until the next section.

Cooperativeness has been identified as being influenced by individuals’ willingness. Positive affects have been conceptualized when this level of willingness to cooperate is high. On the other hand, when the individual has a low level of willingness to cooperate, negative affects are conceptualized. Conceptualized positive and negative affects are more easily recognized when an individual is unable to accomplish the required tasks. The problems related to his/her inability to accomplish tasks have been found to act as obstacles that not only decrease his/her performance and productivity, but also his/her willingness to cooperate with others and ask for cooperation from them.

Acting on these recognized situations either increases or decreases the influence of positive/negative affects on an individual’s willingness to cooperate. The identified actions that increase willingness to cooperate are the wise deal with requirements. Although requirements engineering in Agile requires no complete list of requirements but an iterative process to elicit, analyze, specify, validate, and manage the requirements, managers who pay attention to negative affects and control or mitigate them during the iterative process of RE have been found to have fewer individuals complaining about dependency problems. An individual also has his/her role to play in controlling negative affects and avoiding their influence in terms of hindering the willingness to cooperate.

Finally, the influences of positive and negative affects on individuals’ cooperativeness were confirmed quantitatively by surveying participants’ opinions, and through data collected on their real involvements in Agile projects, where this data was used for confirm the existence of direct significant paths between positive and negative affects and individual’s cooperativeness. Furthermore, this influence was qualitatively
investigated and understood through three properties related to their concepts, which are conceptualizing individual’s affective states when conceptualizing the willingness to cooperate, recognizing the situations when these affective states are conceptualized in collaboration with the individual’s ability to accomplish tasks, and acting on the positive and negative affects when dealing with the problems that influence the individual’s willingness to cooperate. These codes and their associated concepts connect and collaborate to either enhance or decrease the quality of team decisions.

7.4 The Role of Affective Contagion in Agile Teams’ Cooperativeness

Two hypotheses were proposed to relate cooperativeness at the Agile team level to affective contagion. The proposed hypotheses were as follows:

H2.1: The positive affective contagion between Agile teams has a direct positive significant influence on team cooperativeness.

H2.2: The negative affective contagion between Agile teams has a direct negative significant influence on team cooperativeness.

First, to acquire participants’ opinions about these proposed hypotheses, two statements were provided to the participants to choose if they agreed or disagreed. The statements stated that positive/negative affective contagion between Agile team members increase/decrease the level of cooperativeness. A total of 56% of participants agreed to the existence of the role of positive affective contagion on the level of cooperativeness and 36% of participants disagreed. Similarly, 53% of participants agreed to the existence of the role of negative affective contagion and 39% disagreed. Furthermore, 12% of participants neither agreed nor disagreed with the positive and negative affective contagion statements.

The experience factor played a role in the participants’ responses, where the ex-
experts were significantly more likely to agree with these statements than novices. The data on participants’ involvement in their projects were acquired and analyzed to test the two hypotheses. A significant path was highlighted between positive affective contagion and cooperativeness, as illustrated in the affective contagion model 5.13. Similarly, a significant path was highlighted between negative affective contagion and cooperativeness between team members; therefore, both hypotheses H2.1 and H2.2 were supported. Notice that the cooperativeness items in the survey were constructed to test both the cooperativeness of the individual and the cooperativeness of the team; therefore, the model of affective contagion considers the same cooperativeness scale as the affects model.

Proving these hypotheses shed light that encouraged us to investigate in greater depth how this influence occurs, so we carried out part of the qualitative study to investigate the following research question:

RQ3: How do affects and their contagion influence cooperativeness between Agile team members?

Answering this question began with highlighting when positive and negative affective contagion are conceptualized. The level of cooperativeness has been found to be a concrete property for conceptualizing affective contagion. The level of cooperativeness represents the conceptualized team’s affects through the series of actions and reactions that occur during project activities. For example, a series of bad actions have been highlighted as related to current team members’ affects, and a series of bad reactions have been highlighted from the other team members. At the end of these series of actions and reactions, negative or positive affects are spread and become conceptualized. This conceptualizing leads to either an increased or decreased level of cooperativeness at the team level. The team with plenty of negative affects and poor behaviours related to these negative affects is found to be the team with the minimal level of cooperativeness. On the other hand, positive affects enforce good
behaviour, which has been found to be a good indicator of the sustainability of a high level of cooperativeness among team members.

Recognizing positive or negative affects is influenced by the accumulation of past cooperation. The participants in the qualitative study, for example, identified that even if the team carries positive affects at some moments, the past level of cooperativeness controls the team’s willingness to increase their cooperation level.

The current actions to take on negative affects to avoid lowering the level of cooperativeness includes only ignoring them. As mentioned by the participants, managers consider the problem whenever the team faces obvious obstacles that hinder the success of projects. Unfortunately, a low level of team cooperation is not considered to be an obstacle by many managers. The problem exists when the cumulative cooperation level becomes harder to enhance at advanced stages of a project, especially when the team conceptualizes negative affects. Similarly, ignoring positive affects have been found to lead to a high level of cooperation becoming less sustainable.

Finally, the influence of affective contagion on the level of cooperativeness among team members was confirmed quantitatively by surveying participants’ opinions, and through collecting data on their real involvements in Agile projects, where this data enabled us to find a significant path between positive affective contagion and team cooperativeness. Furthermore, this influence was qualitatively investigated and understood through the interactions between properties related to three concepts, which are conceptualizing affective contagion related to current actions and reactions of current team affects, recognizing the situations in which these actions and reactions connect to past experiences of teams cooperativeness, and acting on the positive and negative affective contagion to obtain an acceptable level of cooperativeness. These properties and their associated concepts connect and collaborate to either enhance or decrease the overall level of team cooperativeness.
7.5 The Role of Affects in Agile Teams’ Conflicts

A team’s conflicts were identified in the literature as one of the barriers that can hinder the success of decision-making processes. The literature identifies relations between affects and conflicts in various industries. Throughout these research activities, we investigated the possibility of affects playing a role in Agile teams’ conflicts. Therefore, we hypothesized the following:

H1.3: The positive affects of Agile teams have a direct positive significant influence on team conflicts.

H1.4: The negative affects of Agile teams have a direct negative significant influence on team conflicts.

Two statements were provided to the participants in order to acquire their opinions. The statements stated that positive/negative affects decrease/increase the level of conflicts in Agile teams. A total of 51% of participants agreed to the existence of the role of positive affects in relation to Agile team conflicts and 39% of participants disagreed. Similarly, 54% of participants agreed to the existence of the role of negative affects and 37% disagreed. In addition, 9% of participants neither agreed nor disagreed with the positive and negative affect statements, respectively. We proceeded to investigate the hypotheses using the data on participants’ involvement in Agile projects. The results highlighted significant paths between positive affects and conflicts, as illustrated in the affect model [5.11]. Similarly, a significant path was found to exist between negative affects and conflicts; therefore, both hypotheses H1.3 and H1.4 were supported.

To understand how this influence occurs within the Agile teams, we proceeded to qualitatively investigate the following question:

RQ4: How do affects and their contagion influence conflicts between Agile team members?

Understanding how affects influence team conflicts starts with highlighting how
they influence individuals. Different conceptualized affects can occur throughout the individual’s willingness level to communicate. Sharing information and improving understanding among team members can help achieve good communication. The conceptualized affects that are related to the individual’s willingness to communicate are also indicators for increasing conflicts with other team members. Furthermore, recalling personal experiences of creating or avoiding continuous complaints creates the conceptualized affects. Therefore, the affects’ influence on conflicts arises when others neglect individuals’ opinions and participation. If the recognized affects are negative, the team member tends to develop conflicts continuously. On the other hand, the conceptualized positive affects drive the team member to lower their engagement in unsolicited conflicts.

Managers who encourage good communication and positive interactions, which is one of the agile values, are found more successful in eliminating the influence of negative affects on conflicts. They act to avoid or mitigate the influence of negative affects by encouraging the individual to increase his/her willingness to interact and communicate. This also leads them to consider the individual’s opinions throughout meetings and discussions while he/she is more encouraged to speak out; hence, the recognized negative feeling of being neglected is avoided. On the other hand, managers who avoid encouraging team member to communicate and interact frequently have been found to have team members with a low level of willingness to communicate, along with plenty of negative affects due to not having team members fully involved in team discussions.

Finally, the influence of positive and negative affects on conflicts was confirmed quantitatively by surveying participants’ opinions, and through data on their real involvement in Agile projects, where this data was used to confirm the existence of direct significant paths between positive and negative affects and conflicts in Agile. Furthermore, this influence was qualitatively investigated and understood through
some properties related to their concepts, which are conceptualizing the affective states when conceptualizing the willingness to communicate, recognizing the situations in which these affective states are conceptualized with the collaboration of the individual’s feelings of being neglectful, and acting on these affects by encouraging continuous communication and the interactions of each individual. These properties and their associated concepts connect and collaborate to either enhance or decrease the individual’s level of conflicts with others.

7.6 The Role of Affective Contagion in Agile Teams’ Conflicts

The final two hypotheses were proposed to test the existence of affective contagion’s influence on the level of team conflicts. The hypotheses were intended to investigate the overall level of conflicts based on individuals’ affects and their spread among team members. The proposed hypotheses were as follows:

H2.3: Positive affective contagion between Agile teams has a direct positive significant influence on team conflicts.

H2.4: Negative affective contagion between Agile teams has a direct negative significant influence on team conflicts.

Participants’ opinions about the aforementioned hypotheses were acquired using two statements that were provided to them. The statements stated that positive/negative affective contagion between Agile team members decreases/increases the level of conflicts. A total of 54% of participants agreed to the existence of positive affective contagion’s influence on the level of conflicts and 40% of participants disagreed. Similarly, 55% of participants agreed to the existence of negative affective contagion’s influence and 36% disagreed. In addition, 6% and 9% of participants neither agreed nor disagreed with the positive and negative affective contagion state-
ments, respectively. The experience of participants also played an important role, as the majority of participants who voted for “disagree” were novices, while the majority who voted for “agree” were experts.

After the promising evidence of the existence of affective contagion’s influence on conflicts, the participants’ data on their involvement in Agile projects were analyzed to test the hypotheses. A direct significant path was found between positive affective contagion and team conflicts, as illustrated in the affect model 5.13. Similarly, a direct significant path was found between negative affective contagion and team conflicts; therefore, both hypotheses H2.3 and H2.4 were supported. Confirming the hypotheses allows us to deepen our understanding by seeking out how this influence occurs; therefore, the following research question was qualitatively investigated:

RQ4: How do affects and their contagion influence conflicts between Agile team members?

Conceptualizing affects that influence conflicts occurs when members react to their affective states, where the negative affects have been found to be the dominant factor for increasing the level of conflicts. Therefore, a team member’s negative affect becomes the motivator to create conflicts with others. The other team members become influenced by the negative affects; therefore, they conceptualize the spread of that negative affect. For example, managers have identified that negative affects are conceptualized while the team members discuss and communicate with another who is stubborn, and it is difficult to change his/her mind because of his/her negative mood. Many studies have found that those personalities become less of an influential factor when they are in a positive mood; hence, the other team members conceptualize the positive mood.

One of the situations that leads individuals to recognize the influence of affects on conflicts is increasing the level of discussions while recalling the past conflict level. Recalling the past conflict level brings only the negative or positive affects, whereas
the conflicts can be related to past discussions or situations. Another situation that can be related is the team’s willingness to continuously and openly communicate to keep the consensus among team members sustainable. A more important situation is related to a manager’s changes in his/her affective states. An example of those changes can be seen as in direct relation with the project/iteration outcome. Therefore, a manager’s affects can be the motivator for increasing/decreasing the level of conflicts and unsolicited discussions that influence a team’s communications and interactions.

The last step to take in order to understand how affects influence conflicts should include the current actions for dealing with them. Team members with negative affects tend to close communications channels, which creates less consensus among the team members. The conflicts that arise in this case are difficult to deal with; however, some managers were found to be trying to avoid the low level of communication and to open up discussions frequently. Although this solution might increase the conflicts in the beginning, managers acknowledged that the level of conflicts decreased as the affective trust among team members increased. On the other hand, some managers create more problematic issues when they conceptualize negative affects and expose them while blaming the team. As a result, the negative affects of the manager become transferable to the team, which leads to an increase in the conflict level and decreases the positive communication among team members.

Finally, the influence of positive and negative affective contagion on the level of conflicts among team members was confirmed quantitatively by surveying participants’ opinions, and through analyzing the data on their real involvement in Agile projects, where analyzing this data enabled us to find significant paths between positive and negative affective contagion and team conflicts level. Furthermore, this influence was qualitatively investigated and understood through the interactions between properties related to three concepts, which are conceptualizing affective contagion related to the current team’s affects and communication channels, recognizing the
situations in which these affects are connected to past experiences of team’s conflicts along with manager’s affects changes, and acting on the positive and negative affective contagion to obtain an acceptable level of conflicts. These properties and their associated concepts connect and collaborate to either enhance or decrease the overall level of team conflicts.

7.7 The Research Models of Affect and its Contagion influences on Agile Decisions

The aggregated model in Figure 3.3 was provided to re-test all aforementioned hypotheses and to investigate the hypotheses that correlate between affects and affective contagion. Therefore, we needed to add two more hypotheses to the model to test the correlation between these constructs, which are as follows:

H3: There is a correlation between positive affect and positive affective contagion in Agile decision-making insofar as an individual’s affect positively influences the affective contagion between team members.

H4: There is a correlation between negative affect and negative affective contagion in Agile decision-making insofar as an individual’s affect negatively influences the affective contagion between team members.

The results of testing the aggregated model, as in 5.14, showed that there was direct significant influence of positive affects on positive affective contagion and a direct significant influence of negative affects on negative affective contagion; therefore, H3 and H4 were supported.

The aggregated model showed that when affects and affective contagion constructs correlate, the direct influence of affect on decision quality becomes insignificant. We then tested for the mediation and highlighted that positive affective contagion represented a partial mediation between positive affects and cooperativeness. Unlike
positive affects, negative affective contagion represented a full mediation between negative affects and cooperativeness, where the direct influence became insignificant. The last step to take in order to cover all influences on Agile decisions was to prove that cooperativeness and conflicts represent influential factors that impact decision quality, and to prove their representation as mediators. Therefore, four hypotheses were proposed; the two hypotheses used to test cooperativeness were as follows:

H5: Cooperativeness between team members mediates the influence of positive and negative affects on Agile decision quality.

H6: Cooperativeness between team members mediates the influence of positive and negative affective contagion on Agile decision quality.

Cooperativeness represented a full mediator between positive affects and decision quality. This mediation role was partial when we tested the affect model. The affect model results showed that cooperativeness represented a partial mediation between negative affects and decision quality. Unexpectedly, no direct influence was found to exist between negative affects and cooperativeness within the aggregated model; however, the indirect influence was found through the negative affective contagion. As a result, cooperativeness represented a second mediator, after the negative affective contagion, between negative affects and decision quality. Cooperativeness also represented a full mediator between both positive and negative affective contagion and decision quality. This mediation role was partial when we tested the affective contagion model; therefore, hypotheses H5 and H6 were supported. Similarly, the two hypotheses were proposed to test conflict as a mediator were as follows:

H7: Conflicts between team members mediate the influence of positive and negative affects on Agile decision quality.

H8: Conflicts between team members mediate the influence of positive and negative affective contagion on Agile decision quality.

Similar to cooperativeness, conflict represented a full mediator between positive
affects and decision quality. This mediation role was partial when we tested the affect model. Unlike cooperativeness, the aggregated model results showed that conflict and affective contagion were represented as the two partial mediators between affect and decision quality. Conflict also represented a full mediator between both positive and negative affective contagion and decision quality. This mediation role was partial when we tested the affective contagion model; therefore, hypotheses H7 and H8 were supported.

We conclude from the results that conflicts are highly influenced by individuals’ and team conflicts, which ultimately influence decision quality. Therefore, managers need to consider all types of affects that influence individuals’ motives to engage in conflict and their contagiousness to teams. The cooperativeness of the whole team is highly influenced and increased by the positive affects of individuals and their contagion. However, individuals’ negative affects do not highly influence cooperativeness unless they propagate and spread to a larger number of team members.

From all confirmed hypotheses within the aggregated model, we have come to understand the overall process of experiencing affects within Agile decisions; therefore, the following question was proposed and investigated qualitatively:

RQ1: How do Agile decision makers experience affects?

Answering this question involved highlighting individuals and teams experiencing both positive and negative affects. The concept linking affects and affective contagion connects experiencing individuals’ affects and their transferable and contagious states to the whole team. Within this concept, many properties and codes were highlighted. Mainly, changes in willingness, involvements, and the behaviours of individuals are the main motivators for conceptualizing affects. These motivators lead, when no actions occur, to the encouraging and linking of more team members to conceptualize the associated affects, until changes in a team’s willingness, involvements, and behaviours are reached.
Based on linking the two concepts, many situations appear to be experienced with the role of creating positive and negative affects for both individuals and teams. These recognized situations were described using two different concepts for characterizing situations, where both individuals and teams recognize affects. The consequences of these recognized affects were also explained within these concepts, and the different actions on these consequences were highlighted in different concepts. Therefore, the results of how Agile team members experience affects yielded the general framework that characterizes different aspects, from conceptualizing affects to acting on them.

Finally, both the results of the quantitative study that tested the aggregated model, as in Figure 5.14 along with the qualitative study that emerged as a framework of concepts and properties, as in 6.1 helped in answering the main research question for this thesis, which was:

RQ: How are affects and their contagion related to decision making in Agile environments?

The results of the aggregated model provided scientific proof of the existence of affects’ and affective contagion’s influence on Agile decisions and other factors, such as cooperativeness and conflicts, which influence these decisions while they are also highly influenced by affects and their contagion. All proposed hypotheses were supported, which allowed us to move forward and deepen our understanding by investigating how these influences occur and contribute to either enhancing or decreasing the overall quality of the Agile decision-making processes. Therefore, conceptualizing affects, recognizing their situations, linking affects and their contagion, and Agile individuals and teams acting on affects were addressed and characterized within a general framework to answer the main research question.
7.8 Chapter Summary

- The chapter discussed the results of both the quantitative and the qualitative studies.

- The discussion included grouping and explaining each set of hypotheses, along with the related research questions for each two-constructs relationship.

- The discussion started by highlighting the proven hypotheses which denote the existence of an influence between the two constructs; then, it explains how this influence occurs within the Agile processes.

- Affects were hypothesized and proven to be an influential factor with regards to decision quality, cooperativeness, and conflicts in Agile.

- The approved hypotheses regarding affects’ influence drove the researcher to investigate and characterize how this influence occurs within the Agile decision-making processes.

- Affective contagion were hypothesized and proven to be an influential factor on decision quality, cooperativeness, and conflicts in Agile.

- The approved hypotheses regarding affective contagion’s influence drove the researcher to investigate and characterize how this influence occurs within the Agile decision-making processes.

- The only inconsistent issue was that within the quantitative aggregated model, no direct influence was highlighted between negative affect and cooperativeness, while within the qualitative framework, it was highlighted that negative affect has a strong influence on an individual’s willingness to cooperate.

- Cooperativeness and conflicts were hypothesized and proven to be mediations
between affects and affective contagion and decision quality, while these medi-
atations were characterized and understood qualitatively.

- Finally, both the aggregated model of the quantitative study, as in Figure 5.15 and the emerged general framework of the qualitative study, as in Figure 6.1 articulate the answer to the main research question.
Chapter 8

Conclusion and Future Work

The goal of this chapter is multifaceted. First, the chapter highlights the research objectives and how they were addressed throughout the research activities. Second, the chapter provides both theoretical and pragmatic implications of this research. Finally, the chapter contributes to the implications and recommendations for future research.

8.1 Research Overview

During the past decade, research on how emotions influence software developer performance has increased. Graziotin et al. developed a theory of how affects influence programming performance [1]. They based their theory on two major ideas. The first idea considers how affect impacts workers’ behaviour, their decisions, creativity, turnover/absence, and even pro-social behaviour, such as helping other workers [80]. The authors in [80] note that worker happiness and implications regarding performance outcome is no longer measured by traditional means, such as job satisfaction, but now by positive and negative affects. They show, using various studies, the relation of emotions to worker performance as individuals and groups, considering these new measures of performance including affects. For the second idea, they note the
need to theorize developer performance from a behavioural perspective. Lenberg et al. proposed Behavioural Software Engineering (BSE) as an umbrella term that is related to any behavioural studies on topics including emotions [81].

The main problem is that there was no tested model that explicitly reflects the influence of developers’ affects when they make decisions. Further, there was no explicit evidence of the influence of positive or negative affective contagion on software teams’ cooperation, conflict, or task performance. Therefore, researchers need to develop an improved understanding of and theorize the role of affects and their contagion in Agile decision-making processes. Based on these problems, this thesis contributes to answering the main research question, which is, how are affect and its contagion related to decision making in Agile environments? This question is answered through investigating sets of hypotheses and set of research questions as we will explain later in Chapter 3.

Characterizing the impact of affects and affective contagion on Agile team decisions, cooperativeness, and conflicts, can enhance Agile development processes through avoiding or minimizing the recognizable consequences of negative affects and affective trust; hence, avoiding the consequences of conflicts and enhancing the cooperation between Agile team members. Further, positive affect and affective trust can be characterized and maximized to decrease the conflicts and enhance the cooperation. Further, characterizing these impacts allows us to provide the building blocks for future researchers to develop systems in order to capture and recognize these moods, emotions, and feelings. Without providing real evidences of the impact of affective states in Agile processes, there will still be a lack of motivation to develop and implement systems that recognize these affects.

The objectives of this thesis therefore include: 1. Understanding Agile team members’ opinions and thoughts about affects and affective contagion roles, 2. Collecting data and finding real evidences of the role based on Agile team members assessments.
of their project outcomes, 3. Building and validating measurement and structural models to prove the influence of affects and affective contagion on Agile team decisions, cooperativeness, and conflicts, and 4. Characterizing how each highlighted path of influence within the structural models occurs.

For achieving the aforementioned objectives, we adopted the explanatory sequential design method, which is considered a mixed method approach. We started by collecting quantitative data followed by qualitative data. Within the sequential design, we applied surveys as a quantitative method to test the role of affects and affective contagion and generalize the findings, and interviews as a qualitative method to collect and analyze data and understand the contributing factors of the quantitative model. A total of 249 valid survey responses were collected and examined. These responses were analyzed to understand the current awareness of the impact of affects and affective contagion on Agile decisions, cooperativeness, and conflicts, as in Section 5.2, and 233 of these responses were analyzed for testing measurement and structural models, as in Section 5.4.

The opinions of Agile team members concern whether or not affect and affective contagion play a role with regards to Agile decisions, cooperativeness, and conflicts. Expert Agile members confirmed the influence of affects and affective contagion on Agile team decisions, cooperativeness, and conflicts. Less-expert Agile members also confirmed the role of affects and affective contagion; however, their acknowledgement of the role tends to be significantly less than that of expert Agile members. Furthermore, the 20 affective states of the PANAS scale were provided to Agile team members to extract the most influential affective states that play a role in Agile decisions. The results showed that being determined, inspired, active, and attentive were the four positive affective states that were most influential on Agile individuals and teams. On the other hand, being upset and distressed were the two main negative affective states that were most influential on Agile individuals and teams.
The second type of data, actual data, was used to test the measurement and structural models of affect and affective contagion using pooled-CFA and SEM. The I-PANAS-SF scale was adopted to measure participants’ affective states. The measurement models were built and tested against the collected data using pooled-CFA. After reaching an acceptable model fit with the CFA, the structural models were built and tested against the collected data using SEM. All research hypotheses were supported by testing for the significance of direct and indirect paths. We also examined the mediators in the aggregated model using the bootstrapping method and confirmed the mediation of cooperativeness and conflicts between affects and affective contagion and decision quality.

A total of 16 in-depth interviews were collected and analyzed; while Charmaz’s version of GT was adopted throughout the course of collecting and analyzing the data. As a result, seven concepts have emerged from the data along with their properties and interactions among them. The first concept is conceptualizing positive and negative affects, which is characterized by four properties; willing to cooperate, willing to good communicate, calling personal experience, and deciding on individual decisions. The second concept is conceptualizing positive and negative affective contagion, characterized by four properties; level of good communication, level of cooperativeness, level of conflicts, and level of team decision accuracy. The third concept is linking affects and affective contagion, which is characterized by three properties; changes in behaviour, changes in member involvement in decisions, and changes in willingness. The fourth concept is recognizing situations when affects influence individuals, which is characterized by three difficulties that individuals can face; inability to accomplish tasks, lack of guidance, and neglecting individual’s opinion. The fifth concept is recognizing situations when affective contagion influence teams, which is characterized by three properties; participating in team decisions, calling past affects’ influences, change in managers’ affective states. The sixth concept is acting on affects.
to influence affective contagion which is characterized by five properties; optimizing, controlling, mitigating, ignoring, and exaggerating. The seventh concept is work- and non-work-related factors, which can be characterized throughout various properties for work-related factors such as organization polices, management, and motivations, and various properties for non-work-related factors such as family issues, transportation issues, and psychological issues such as anxiety and depression.

The results of both quantitative and qualitative data were discussed and explained. Affects and affective contagion were hypothesized and proved as influential factors on decision quality, cooperativeness, and conflicts in Agile. The approved hypotheses of affects influence drove the researcher to investigate and characterize how this influence occurs within the Agile decision-making processes. Only one inconsistent issue was highlighted when discussing the two results, which is the absence of direct influence between negative affect and cooperativeness within the quantitative aggregated model; while it has been highlighted that negative affect has strong influence on individual’s willingness to cooperate within the qualitative framework. Finally, cooperativeness and conflicts were hypothesized and proved as mediations between affects and affective contagion and decision quality; while this mediations were characterized and understood qualitatively.

To conclude, both the aggregated model of the quantitative study, as in Figure 5.15, and the emerged general framework of the qualitative study, as in Figure 6.1, articulate the answer for the main research question.

8.2 Theoretical Implications

This thesis expands the current theoretical lens of understanding the impact of affects and affective contagion on individuals. The thesis findings highlighted the consequences of negative affects on individual willingness to cooperative and communicate
along with the proclivity to create more conflicts with others. On the other hand, the thesis incorporated the advantages of positive affect influences on individual willingness to cooperate, communicate, and reduce unsolicited conflicts with others. In line with Graziotin et al., who have provided a theoretical background for understanding the affects of developers [19], we shed the light on the impact of these affects, but on Agile developers. Further, the provided framework enriches the understanding of different affective state roles on Agile developers along with theoretically connecting their affects to behaviour and individual decisions.

The thesis also creates a new theoretical lens by demonstrating what the role affects and affective contagion have and how they are incorporated into team collaboration and communication within Agile development. The study findings highlighted the consequences of negative affects on the overall levels of team cooperativeness and good communication along with the proclivity to generate more conflicts among team members. Positive affects influences were also incorporated with the fact of enhancing the team’s collaboration and communication along with hindering the unsolicited team conflicts.

The thesis findings demonstrated the influence of affects on Agile decision maker experience, behaviour, and thinking. The influence was demonstrated through different concepts related to recognizing and acting on affects. This demonstration provided theoretical framework of how different criteria in Agile decisions are correlated and incorporated with work and no-work factors. The explanations were expanded to cover past affects, current affects, expected affects, cumulative affects, problem characteristics, decision characteristics, along with current acts on these factors when incorporating with affect influences.
8.3 Pragmatic Implications

There are many practical implications of this research. First, the thesis provided a real assessment of the current level of awareness of the role affects play among Agile team members. The findings demonstrated that novices team members are considering the influence of affects and affective contagion less. These findings can advise both novices and their managers to mitigate, control, and avoid to exaggerate the negative consequences of negative affects; while utilizing and optimizing the influences of the positive affects. Further, the thesis provided the most influential positive and negative affective states based on adopting and validating some psychological scales. Agile practitioners can pay closer attention to the existence of these affects and consequences within their developmental environment.

Second, managers become aware of when and where the most situations that the team members can recognize positive and negative affects. Other than no work related factors, this research enables managers to be able to pay closer attention to different triggers and motivators of negative affects and act on them accordingly. Even though creating techniques and methods to act on affects are outside the scope of this research, the findings of this research encourages managers to consider and incorporate different techniques or systems to deal with affects and their influences.

Third, managers can consider the negative affects of individuals and teams in Agile environments to enhance the level of cooperativeness and good communication among the team members. According to the findings, individual willingness to communicate and cooperative is the motive for the overall team’s level of communication and cooperation. Managers can consider the willingness of individuals and monitor them to keep the overall team performance and productivity at its highest level. This can be done, for example, by considering all team members’ voices and encourage them frequently to avoid the initiation of negative affects during discussions and making decisions.
Finally, the findings lead managers to increase the awareness of negative affects influence regardless of considering professional workplaces. In line with industrial and managerial literature, this thesis finds a significant influence of affects and their contagion among the Agile environments regardless of their professionalism and technical acumen. Therefore, managers can base their considerations of these influences and act accordingly.

8.4 Limitations

Eventhough the effort was made by the researcher, time and resources related to conduct all research activities, including the adoption of explanatory sequential design method which requires more time, have hindered the researcher to conduct more methods, such as case studies, to confirm and characterize the impact of affects more efficiently. However, we were able to conduct both quantitative and qualitative studies which provided good results and findings that can help future researchers to build their assumptions when adopting case studies and other design methods.

Regarding the limitations of the qualitative study, there were two issues. The first limitation is the sample size. Eventhough the 16 in-depth interviews provided us with saturated data needed to build robust framework that characterizes all properties of how affects influence Agile team members, more samples are needed to confirm and build more reliable theories of affects in Agile environments.

Second, on-line interviews hinders the researcher observations and notices on participant responses. We tried to conduct as many face-to-face interviews as possible. Because of the need for Agile experts, we found only six participants while the others were on-line interviewees. Another problem with on-line is internet connection was not stable for some interviewees. We tried to resolve this problem with repeating answers and making sure that all participant data was clearly recorded. However, the
transcribing of participant responses showed some missing data. We put extra time to re-investigate recordings and extract more accurate data.

8.5 Threats to Validity

Internal validity is concerned with researchers measuring the causation correctly. In survey research, there is less control than in other research designs, for example, experiment research. Therefore, internal validity tends to be low in surveys. In this research, we adopted different psychological scales to measure participants’ affects, and different scales to measure their project outcome, based on their assessments, to highlight the influence of affects on the Agile team and project outcome. In this survey research, two issues arise regarding the internal validity.

First, measuring the influence of affects on a project outcome can be influenced by the number of participants from each team. We cannot determine how many team members of a project have participated in the survey. Therefore, for example, conducting a case study and controlling its settings by acquiring all team members’ affects, along with measuring project outcome other than the participants assessments, can lead to higher internal validity. The second issue related to the internal validity is the adoption of the I-PANAS-SF scale, which measures participants’ affects during the past two months. In fact, we were not able to measure participants’ affects at the time of making the decision, rather, the overall affects during an iteration/phase. It could be more accurate for adopting the I-PANAS-SF scale or any other psychological scale to measure the current affects of participants at the time of making the decisions, in order to examine the influence of positive and negative affects and their contagion on their decisions, cooperativeness, and decisions.

Credibility in qualitative research is an equivalent to internal validity in quantitative research [179]. During the adoption of the constructivist grounded theory,
we ensured the engagement of multiple sources of information, data from quantitative study, interviews, and notes, along with sharing the findings with participants to confirm them and make sure that the emerged data represents the reality of the phenomena we intended to measure, which rotates around understanding how the affects related to Agile team members.

External validity is concerned with assuring the researchers are measuring what happens in the real world. There were no main issues regarding the external validity because in quantitative research we adopt the survey research which we disseminate the questionnaire to participants and they were able to answer them in natural settings without the intervention of researchers.

Transferability in qualitative research is an equivalent to external validity in quantitative research [179]. Transferability was enhanced by interviewing participants with diversity regarding their demographic variables. Even though we have sought for only expert Agile team members to interview, the expert factor itself was enhanced by seeking for experts with a wide range span of five to 20 years. This diversity, along with explaining the research objective and describing it clearly, assured an enhanced transferability in this research.

8.6 Future Work

The thesis also creates implication for future research. First, researchers can build their assumptions on the thesis findings; i.e. researchers can build on the influence between affects and individual willingness to cooperate and communicate to investigate other team factors such as the correlation between affects, willingness, and team creativity. For example, having a creative team can lead to more positive affects and more cooperation among the team members or vice versa. Given the fact that there is an existence of influence of affects on cooperation, can be considered the building
blocks to the causality between team creativity and affects and cooperation.

Another implication of future research resides where researchers tend to design and implement systems for capturing the affects. Current systems consider face or voice recognition to understand the current affective states of individuals. The findings of this thesis shed light on different situations that researchers need to consider when trying to capture these affects. Therefore, in adopting these systems, one can consider some factors related to the situations when these affects are recognized. For example, researchers can target techniques or systems when recognizing negative affects during situations where the cooperation level is decreased or conflict is increased. Related triggers and motivators during these situations can be found and analyzed faster, hence, the negative affect can be detected.

Further experiments are needed to explore in-depth how the affects and affective contagion can be related to different factors in Agile development such as creativity and motivation. Experiments in the form of case studies can be helpful and provide insightful results. Further, the adoption of more psychological scales, for example SPANE, within the case study can provide more accurate results.

Considering the in-depth investigation of how specific affective state participate in Agile team cooperation and communication can be helpful. We have discussed the lack of interpretation and ignorance of the affective states; rather, practitioners tend to use ‘happy’ or ‘sad’ to describe individual or team affective state. Therefore, there is a need for in-depth analysis of the affective state such as upset and distress which scored the highest recognized negative affective states in the Agile decision making processes.

Finally, studying the role of affects and affective contagion in domains that have different characteristics and recently adopted Agile methods is an important and promising venue. For example, the adoption of Agile methods with mission critical systems, such as a defense sector, is increasing while some problems related with
team conflicts, team cooperation, team decisions and the cost related to inaccurate decisions in Agile methods are still not resolved. The risks related to human errors in these risky domains are different than other domains; while affects can play a crucial role in impacting human errors. Therefore, there is a need for theorizing the role of affect and its contagion influences, as undiscovered influential factors on Agile teams, when using Agile as underline process for producing mission critical systems.

8.7 Chapter Summary

- The chapter provided an overview of the research and its results and findings.

- The thesis expanded the current theoretical lens of understanding the impact of affects and affective contagion on individuals.

- The thesis also created a new theoretical lens by demonstrating what is the role of affects and affective contagion and how are they incorporated through team collaboration and communication within Agile development.

- Practical implications on Agile managers and their team members of this research were highlighted.

- The chapter provided some limitations related to time, resources, and samples.

- Finally, the chapter highlighted the need for future experiments and studies to enrich this research findings.
REFERENCES


[9] Ping Xu and Tao Wang. Emotion in decision making under uncertainty and
cognitive control: a computational framework. In Third International Confer-
ence on Natural Computation (ICNC 2007), volume 2, pages 149–154. IEEE,
2007.

Affect and creativity at work. Administrative science quarterly, 50(3):367–403,
2005.

emotion-specific influences on judgement and choice. Cognition & Emotion, 14

[12] Clark Briggs and Patrick Little. Impacts of organizational culture and person-
ality traits on decision-making in technical organizations. Systems Engineering,

[13] Bjørnar Tessem. An empirical study of decision making, participation, and em-
powerment in norwegian software development organisations. In International

[14] Torgeir Dingsøyr and Tore Dybå. Team effectiveness in software development:
Human and cooperative aspects in team effectiveness models and priorities for
future studies. In Proceedings of the 5th International Workshop on Co-operative


[16] Ricardo Colomo-Palacios, Cristina Casado-Lumbreras, Pedro Soto-Acosta, and
Ángel García-Crespo. Decisions in software development projects management.


[30] Beatrice Alenljung and Anne Persson. Portraying the practice of decision-
making in requirements engineering: a case of large scale bespoke development.


225


[84] Cynthia D Fisher and Christopher S Noble. A within-person examination of


[105] José Adson Oliveira Guedes da Cunha and Hermano Perrelli de Moura. Towards a substantive theory of project decisions in software development project-based organizations: A cross-case analysis of it organizations from brazil and portu-


[148] Erika J Wolf, Kelly M Harrington, Shaunna L Clark, and Mark W Miller. Sample size requirements for structural equation models: An evaluation of power,


Appendix A

Survey Questions

Project Title: Affect (emotions and moods) influences on agile decision making

Researcher(s): Abdulaziz Alhubaishy, Graduate Student, Software systems Engineering, University of Regina, phone: 3065803932, email: alhubaia@uregina.ca

Supervisor: Luigi Benedicenti, Software systems Engineering, email: luigi.benedicenti@uregina.ca

The objective of this part of research is to explore the role played by affects, emotions, and moods in the software decision-making process and other factors related to agile decisions such as team cooperativeness and conflicts. We intend to highlight the role of both positive and negative influences on agile decision makers. Further, this part of research seeks to test whether or not there is a positive or negative influence of affective contagions (i.e. the spread of a member’s affective state, such as happiness, to other members) on teams and their decisions.

The procedure of this project is by start collecting data from developers by questionnaire designed and disseminated by the Qualtrics software. Participants are asked to answer a series of questions regarding their opinions and actual involvement in a software process decision activities. The actual involvement requires answering the survey questions that aimed to measure their affects during last two months of their
involvement in decision activities.

The benefit of this study is to help us to identify how agile teams become more productive. No direct benefit for the participation in this study; however, as a developer you can become more aware by the role of affect in your activities during software production process.

Your participation takes around 15 to 20 minutes. All of the responses in the survey will be recorded anonymously. The data will not be exposed under any circumstances. It is for the purpose of understanding the role of affect in agile decisions. During the collection and analysis of data, the data will be stored in computers protected by the University of Regina.

Your participation is voluntary and you can withdraw from the research project for any reason, at any time without explanation or penalty of any sort. If you withdraw, your data will not be stored or used in this study. When you submit the questionnaire, there is no possible way to delete your answers because responses are recorded anonymously. To obtain results from the study, you will be asked to enter your e-mail address at the end of the survey. We will email you the results at the end of the analysis and writing results.

This project has been approved on ethical grounds by the UofR Research Ethics Board on (April 2018). Any questions regarding your rights as a participant may be addressed to the committee at (306-585-4775 or research.ethics@uregina.ca). Out of town participants may call collect.

If you have any questions regarding the survey of this research, please contact me at alhubaia@uregina.ca or my Supervisor Dr. Luigi Benedicenti, who can be contacted at luigi.benedicenti@uregina.ca

IMPLIED CONSENT FOR SURVEYS By completing and submitting the questionnaire, YOUR FREE AND INFORMED CONSENT IS IMPLIED and indicates that you understand the above conditions of participation in this study.
What is your Gender?

- Male
- Female
- Other

What is your experience in Software Development? (Years)

What is your experience in Agile Software Development? (Years)

In your last project, which of the following agile methods have you used?

- SCRUM
- eXtreme Programming (XP)
- Kanban
- Crystal
- Feature-Driven Development (FDD)
- Dynamic systems development method
In your last project, what was your role in the software team?

- Developer
- Tester
- Manager
- Customer
- Product Owner
- Scrum Master
- Other (please specify)

In your last project, what is the size of team you have worked with?

- 2 - 5 Members
- 6 - 10 Members
- 11 - 15 Members
- More Than 15 Members

Was your last involvement in agile project during the last two months?

- YES
- No
Within agile software development teams, how much do you agree or disagree with the following statements:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neither agree nor disagree</th>
<th>Agree</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive affects (emotions and moods) enhance the quality of agile decisions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positive affects increase the level of cooperativeness in agile teams</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positive affects decrease the level of conflicts in agile teams</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative affects (emotions and moods) decrease the quality of agile decisions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative affects decrease the level of cooperativeness in agile teams</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative affects increase the level of conflicts in agile teams</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Indicate the extent you have felt this way over the past week/weeks of your project's iteration:
If you think there are influence of affects on agile decisions, what are the most influential POSITIVE affects?

<table>
<thead>
<tr>
<th></th>
<th>Very Slightly or Not at All</th>
<th>A Little</th>
<th>Moderately</th>
<th>Quite a Bit</th>
<th>Extremely</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attentive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determined</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inspired</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alert</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Upset</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hostile</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ashamed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nervous</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Afraid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you think there are influence of affects on agile decisions, what are the most influential POSITIVE affects?

- Interested
- Excited
- Strong
- Inspired
- Determined
- Attentive
If you think there are influence of affects on agile decisions, what are the most influential NEGATIVE affects?

- Enthusiastic
- Proud
- Alert

- Active
- Happiness
- Other ... please Specify

If you think there are influence of affects on agile decisions, what are the most influential NEGATIVE affects?

- Distressed
- Upset
- Guilty
- Scared
- Hostile
- Irritable

- Ashamed
- Nervous
- Jittery
- Afraid
- Sadness
- Other ... Please Specify

Within agile software development teams, how much do you agree or disagree with the following statements:

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neither agree nor disagree</th>
<th>Agree</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
During agile decision making process, indicate the extent you were influenced by the other team member/s in your last project

<table>
<thead>
<tr>
<th>Positive affective contagions between team members enhance the level of agile decisions</th>
<th>Not at All</th>
<th>A Little</th>
<th>Somewhat</th>
<th>Quite a Bit</th>
<th>Very Much</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive affective contagions between agile team members increase the level of cooperativeness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positive affective contagions between team members decrease the level of conflicts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative affective contagions between team members decrease the level of agile decisions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative affective contagions between agile team members decrease the level of cooperativeness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative affective contagions between team members increase the level of conflicts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The behaviour related to the other’s affective state made me feel happy

The behaviour related to the other’s affective state made me feel contented
If you think there are influence of affects on agile decisions, what are the most POSITIVE affects of a team member that influence (or contagion to) other team members?

- Interested
- Excited
- Strong
- Enthusiastic
- Proud
- Alert
- Inspired
- Determined
- Attentive
- Active
- Happiness
- Other ... please Specify
If you think there are influence of affects on agile decisions, what are the most NEGATIVE affects of a team member that influence (or contagion to) other team members?

☐ Distressed  ☐ Ashamed
☐ Upset  ☐ Nervous
☐ Guilty  ☐ Jittery
☐ Scared  ☐ Afraid
☐ Hostile  ☐ Sadness
☐ Irritable  ☐ Other ... Please Specify

Within your last project, how would you assess the perceived decision quality regarding the following criteria:

- Final team decisions reflects the best that could be extracted from the team
- Final team decisions usually extended the
Within your last project, how would you assess the conflicts in the team regarding the following criteria:

<table>
<thead>
<tr>
<th>Quality of Team Member's Individual Input</th>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neither agree nor disagree</th>
<th>Agree</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>I am confident that the final decision we came up with is the best decision</td>
<td>○</td>
<td></td>
<td></td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Final team decisions is of much higher quality than the initial proposals of the individual members</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Overall, it is my opinion that our decisions are of high quality</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

Within your last project, how would you assess the conflicts in the team regarding the following criteria:

<table>
<thead>
<tr>
<th>Conflicts in the Team</th>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neither agree nor disagree</th>
<th>Agree</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>How often do your team members disagree about opinions regarding the work being done</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>How frequently are there conflicts about ideas in your team</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>How much conflict about the work you do is there in your team</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>
Within your last project, how would you assess the team’s cooperativeness regarding the following criteria:

To what extent are there differences of opinion in your team

Optional Question: In your Opinion, What are the most Work-related factors that cause or change positive/negative affect which influence decisions in agile projects?
Optional Question: In your Opinion, What are the most NON Work-related factors that cause or change positive/negative affect which influence decisions in agile projects?

If you would like to get the results of this study, please enter you e-mail address

Finally, we intend to conduct interviews related to this project. The interviews will take about 60 to 90 minutes. If you would like to participate in this study, please enter your e-mail address and we will contact you

Powered by Qualtrics
Appendix B

Interview Questions

A. Demographic Questions:
Same as the survey.

B. Affect Questions:
1. What does affect (emotions and moods) means to you when participation in agile decisions?
2. Does the affect influence your individual decisions or participation in your team decisions?
3. Is affect considered in your development team? Why or Why not?
4. Does the affect influence your cooperativeness with your agile team? If yes, How?
5. Does the affect influence the level of conflicts within your team? If yes, How?
6. As a team member, what do you do to mitigate or prevent the influence of affect and/or affective contagion on your decisions?
7. In your opinion, even though the majority of a survey participants acknowledged the influence of affect, why there is no such actions taken by managers to deal with it?

C. Affective Contagion:
1. What does Affective contagion means to you when participation in agile decisions?
2. Does affective contagion influence your participation in your team decisions?
3. Is affective contagion considered in your development team? Why or Why not?
4. Does the affective contagion influence your cooperativeness with your agile team? If yes, How?
5. Does the affective contagion influence the level of conflicts within your team? If yes, How?
6. In your opinion, eventhough the majority of survey answers acknowledged the influence of Affective contagion, why there is no actions taken by managers to deal with it?
7. In your opinion, are there different levels of affect and its contagion influences regarding the decision phase (for example requirements)? Why?

D. Closing Questions:
1. Can you share a story or an evidence when positive or negative affects or affective contagion had an influence on your or your team decision?
2. Do you have comments or feedback related to affect and its contagion in agile decisions?
Appendix C

Certificate of Approval

Figure C.1: Certificate of Approval
Appendix D

Pooled-CFA Models for Affect and Affective Contagion

Figure D.1: Pooled-CFA for the Affect Model
Figure D.2: Pooled-CFA for the Affective Contagion Model
Appendix E

Framework of Affect and Affective Contagion on Agile Decisions

Figure E.1: Detailed Framework of Affects and Affective Contagion as Influential Factors on Agile Decision-Making Processes