The Emotional Roller Coaster of Responding to Requirements Changes in Software Engineering

Kashumi Madampe, *Graduate Student Member, IEEE,* Rashina Hoda, *Member, IEEE,* and John Grundy, *Senior Member, IEEE*

Abstract—**Background:** A preliminary study we conducted showed that software practitioners respond to requirements changes (RCs) with different emotions, and that their emotions vary at stages of the RC handling life cycle, such as *receiving, developing,* and *delivering* RCs. Furthermore, such developer emotions have direct linkages to cognition, productivity, and decision making. Therefore, it is important to gain a comprehensive understanding the role of emotions in a critical scenarios like handling RCs. **Objective:** We wanted to study how practitioners *emotionally* respond to RCs. **Method:** We conducted a world-wide survey with the participation of 201 software practitioners. In our survey, we used the Job-related Affective Well-being Scale (JAWS) and open-ended questions to capture participants emotions when handling RCs in their work and query about the different circumstances when they feel these emotions. We used a combined approach of statistical analysis, JAWS, and Socio-Technical Grounded Theory (STGT) *for Data Analysis* to analyse our survey data. **Findings:** We identified (1) emotional responses to RCs, i.e., the most common *emotions* felt by practitioners when handling RCs; (2) different *stimuli* – such as the RC, the practitioner, team, manager, customer – that trigger these emotions through their own different characteristics; (3) *emotion dynamics*, i.e., the changes in emotions during the RC handling life cycle; (4) *RC stages* where particular emotions are triggered; and (5) *time related aspects* that regulate the emotion dynamics. **Conclusion:** Practitioners are not pleased with receiving RCs all the time. Last minute RCs introduced closer to a deadline especially violate emotional well-being of practitioners. We present some practical recommendations for practitioners to follow, including a dual-purpose emotion-centric decision guide to help decide when to introduce or accept an RC, and some future key research directions.

Index Terms—emotions, affects, requirements, changes, human aspects, mixed-methods, software engineering, software teams, socio-technical grounded theory, job–related affective well–being scale, well–being, workplace awareness

1 Introduction

Requirements changes (RCs) in software development include such actions as additions, modifications and deletions of functional or non-functional requirements [1]. Such RCs naturally impact the cost, quality, and schedule of the project [2]. Hence, they are crucial to understand during software development. In traditional software development contexts, introducing an RC can be considered an intervention – sometimes unexpected – during the software development process. In agile contexts, RCs are welcomed "*even late in development*" [3]. Either way, an RC acts as a stimulus (trigger) in the development environment with the potential to elicit responses from the practitioners handling the RC. Humans respond to stimuli in various ways. One such way is through *emotions*. Emotions are defined as *"a sequence of interrelated, synchronised changes in the states of all the five organismic subsystems (information processing, support, executive, action, and monitoring) in response to the evaluation of an external or internal stimulus event as relevant to central concerns of the organism"* [4], for example *excitement, satisfaction, anxiety,* and *fatigue* [5], and reaction to a given stimulus through emotions is called *emotional response* [6]. Emotions act as behavioural motivators [7], and have direct linkages to cognition [7], productivity [7], [8], [9], and decision–making [10].

Through an interview-based preliminary study using Grounded Theory, we found that software development teams show emotional responses to RCs at stages of *receiving*, *developing*, and *delivering* the RC [11]. The findings of this preliminary study highlighted *the need to better consider developer emotional responses to improve the RC handling experience of software development teams*. Therefore, we were motivated to conduct a more in-depth study on how developer emotions vary over the RC handling life cycle. On the other hand, with agile methods becoming the default standard way to develop software, handling RCs is no longer limited to a single phase in software development. As RCs can have significant impacts on software products [1], developers need to handle RCs carefully. To do this, better cognition, productivity, and decision-making are necessary. As developers emotions can have direct linkages to these needs, developers need to be aware of their emotions when handling RCs. Being aware of one's own emotions while working has been shown to increase developer efficiency and effectiveness in practice [12]. Therefore, having a more comprehensive understanding of developer emotional responses to RCs should allow us to find ways to better handle RCs during software development.

Before moving forward with conducting an in-depth empirical study, we explored the existing literature on the topic to establish the importance of studying this topic further.

• *K. Madampe, R. Hoda, and J. Grundy are with the HumaniSE Lab at Department of Software Systems and Cybersecurity, Faculty of Information Technology, Monash University, Wellington Road, Clayton, VIC 3800, Australia.*
E-mail: kashumi.madampe@monash.edu

Manuscript received September 08, 2021; revised Month Date, 2021.

We found that research on understanding emotions of developers during software development contexts is gaining momentum. Several studies have focused on emotions and productivity of individuals in the software development teams [13], [14], [15], [16], [17], [18]. However, only a few studies have focused on emotions of software development teams during requirements engineering (RE) [7], [19].

Therefore, the key reasons for conducting this research were (1) the pressing necessity of finding better ways to improve RC handling, and (2) lack of existing knowledge on emotional responses to RCs by developers.

Consider a software development team working *"enthusiastically"*. Suddenly, the customer decides to change an already implemented requirement. Kash, who is a developer in the team, gets *"angry"* because of this RC arriving, which distracts from current work, potentially wastes previous efforts, and goes against the requests the customer previously made. Kash thinks that the customer does not understand Kash's job and its demands. At the same time, this RC makes her peers *"angry"* and even *"depressed"* about flow-on consequences of the new RC. The team disagrees about certain things about handling the RC. Kash feels that her manager cannot appreciate her team's emotions. However, she has no other option than working on the RC with her team. Therefore, she starts to implement the requested RC. At the beginning, Kash begins to feel more *"energetic"* while updating the code, and once she is done with coding, she feels *"inspired"* by solving the customer's new challenge. Then Kash decides to test the code, and during delivery she is *"excited"*, and *"inspired"*. Once the RC is delivered, Kash is fully *"relaxed"*. Kash's reflections on this whole RC handling experience show that not only hers, but also her peers' cognition and decision–making capabilities, were impacted by what they felt. Is the reality for practitioners similar or very different to this fictitious situation? What emotions do practitioners actually experience when responding to different kinds of RCs from different people at different times? Is RC the only stimulus that triggers practitioners' emotions? How can software practitioners have a better RC handling experience?

To gain a more comprehensive understanding of emotional responses to RCs in software teams, our broad question of this exploratory study is as follows:

How do software practitioners respond emotionally to requirement changes?

To answer this research question, we conducted a worldwide survey[1] with the participation of 201 software practitioners. We utilised the Job-related Affective Well-being Scale (JAWS) [5] which assesses people's emotional reactions to their job during the past 30 days and a set of open-ended questions. We used a combined approach using descriptive statistical analysis, JAWS and socio-technical grounded theory (STGT) [20] to analyse the data. In this paper, we use the terms "emotional responses" and "feeling emotions/ respond with emotions/ experiencing emotions" interchangeably. In simple terms, "emotional response" is bringing "emotion" into play. Our analysis resulted in identifying the following key findings:

KF1. **Emotional responses to RCs:** We found the most common emotions felt by practitioners when handling RCs;

KF2. **Stimuli triggering emotions:** We found several stimuli, such as the RC, the individual practitioner, the team, manager and customer, can all lead to the triggering of different emotions;

KF3. **Emotion dynamics in requirements changes handling life cycle:** We found the phenomenon of emotion dynamics in software team contexts. i.e., the fluctuation of emotional responses across time during the RC handling life cycle;

KF4. **RC stages lead to emotion dynamics:** We found that practitioners' emotions are triggered at specific stages of the RC handling life cycle;

KF5. **Regulation of emotion dynamics:** We discovered that adjusting time related aspects makes it possible to regulate (manage) the emotional responses of practitioners at RC handling stages.

The key contributions of this research include:

- **Identification of a range of emotions and stimuli experienced by software practitioners in handling RCs.** We identified and categorised a range of emotions and factors likely causing emotional responses to RCs, including when emotion occurs and other time related aspects, factors related to the individual, manager, team and customer.
- **A set of practical recommendations for practitioners** to follow in RC abundant environments;
- **A dual-purpose emotion-centric decision guide for both carriers of RCs and practitioners who do not act as carriers of RCs** to use to decide when to introduce an RC to the team and when to accept an RC, respectively;
- **Knowledge including a model representing an emerging theory** to learn and conduct further research about emotion dynamics in software team contexts.

2 Motivation from Related Work

Agile methods are predominant in software engineering. Therefore, it is necessary to note that one of the reasons for introducing agile to the software industry is to allow embracing changes [3]. However, RC handling in agile is easier said than done. Software practitioners face countless challenges when handling RCs. For example, lack of requirements traceability [21], incorrect requirements prioritisation [21], minimal requirements documentation [1], [22], [21], [23], contractual issues [21], and customer agreement [21]. However, several practices such as face–to–face communication [24], [25], [26], [27] iterative requirements [28], [29], [27] [26], prototyping [28], [29], [30] review meetings [29], [26], and prioritisation [28], [29], [26] are in place to allow better handling of RCs. Not only these practices, but also the practices that we have shared in Table 3 are used in agile software development contexts with the aim of better

1. Approved by Monash Human Research Ethics Committee. Approval Number: 23578

delivering the software product. However, there are challenges associated with adopting these practices [31]. Given the challenges and the low pleasurable emotions felt due to the challenges, the emotional responses in different the stages of the RC handling life cycle may still vary. Through this work, we show that the emotional responses to RCs vary, and we present how they vary throughout their life cycle in a comprehensive manner, whereas our preliminary study [11] was limited to only a handful of emotions and three high level RC stages.

Several studies have been conducted in emotions in SE contexts highlighting the importance of emotions in SE contexts. Graziotin et al. [17] and Crawford et al. [32] also review these in their works. Kuutila et al.'s work [33] on affective states versus software metrics highlights the existence of negative relationships between hurry and number of commits, and social interaction and hindered work well–being. Emotions have a direct potential impact on developers work and productivity [10], [13]. Reasons such as localising relevant code, better understanding parts of the code, clear next steps, writing code, and having new ideas increases emotions/ progress and difficulty in understanding how parts of the code/ API work, difficulty in localising relevant code, not being sure about next steps, realising that hypothesis on how code works is wrong, and missing/ insufficient documentation decreases emotions/ progress according to Muller and Fritz [10].

Handling RCs is a crucial phenomenon. It requires a lot of effort, as RCs have both the abilities to make and break the software. RC handling is not limited to a single phase in software development. Software practitioners emotionally invest in RC handling throughout the software development process [11]. Being aware of their own emotions while working increases the developers' progress by allowing them to mitigate negative emotions [12]. Therefore, it is very important to study how software practitioners feel when handling RCs. On the other hand, it is also arduous to study how software practitioners feel in this complicated circumstance, given that no prior studies exist on developer emotional responses to RCs. Through this paper, we fill this gap by providing an overall understanding of how emotions are experienced when handling RCs so that both practitioners and researchers can benefit from our findings.

3 Study Design

3.1 Definitions

We use some concepts from Psychology, Grounded Theory, and a few of our own terms throughout this paper. Appendix A presents the definitions of these terms.

3.2 Approach

An overview of our research approach is given in Fig. 1 and described in detail below. The replication package, including the survey questionnaire, demographic and project information of the participants, is available online[2].

2. https://github.com/kashumi-m/ReplicationPackageEmotionalRollerCoaster

TABLE 1: Job-related Affective Well-being Scale Sub-Scales (Abv.: Abbreviation)

Sub Scale	Abv.	Emotion
High pleasurable–High arousal	$High^2$	Energetic, Excited, Ecstatic, Enthusiastic, Inspired
High pleasurable–Low arousal	$High^1$	At-ease, Calm, Content, Satisfied, Relaxed
Low pleasurable–High arousal	Low^1	Angry, Anxious, Disgusted, Frightened, Furious
Low pleasurable–Low arousal	Low^2	Bored, Depressed, Discouraged, Gloomy, Fatigued

3.2.1 [Step 1]: Emotion Scale Selection

Our preliminary study suggested people may struggle to express and name emotions in an open format, from scratch [11]. Therefore, we decided that, we needed a way to describe human emotional responses to RCs. We evaluated 20 well-established emotion scales (15 as in [34] and PANAS [35], SPANE [36], JES [37], DEQ [38], JAWS [5]) by comparing their listed emotions and their applicability to use to describe practitioners' emotional responses to RCs. By 'applicability' we mean relevance of emotions in software engineering contexts and the concise nature of emotions in the scale. From our analysis, we found 3 scales – Discrete Emotions Questionnaire (DEQ), Job Emotion Scale (JES), and Job-related Affective Well-being Scale (JAWS) – as suitable candidates. From our own industrial experience, we opted not to use DEQ as we found that some emotions were irrelevant for software development teams (e.g.: "terror" and "craving"). We had used JES in our previous work [37] which consists of 16 emotions. However, we wanted to gain a comprehensive understanding of emotional responses to RCs including a wide range of possible emotions. In the end, we decided to use JAWS which has more emotions than JES, and which has been used extensively to assess emotional reactions of people to their jobs over the past 30 days [39], [40], [41]. As our survey questionnaire requested participants to respond by considering the current or most recent project they worked on, we found JAWS likely to be the best emotion scale for our study.

JAWS has two forms: one with 30 emotions (long form) and another with 20 emotions (short form). We used the short form which the authors of JAWS claim as the scale that is most commonly used [42]. The 20 emotions in JAWS are categorised into 4 sub-scales along the dimensions pleasure and arousal (intensity). The sub-scales are namely, High pleasurable-High arousal ($High^2$), High pleasurable-Low Arousal ($High^1$), Low pleasurable-High Arousal (Low^1), and Low pleasurable-Low Arousal (Low^2). We abbreviated the sub-scales as above by making the abbreviation central to the pleasure. i.e., for example, when both pleasure and arousal are high, we abbreviated it as $High^2$; otherwise $High^1$. The emotions under each sub-scale are given in Table 1. The scale enables the participants to select one of the following five choices per emotion experienced: *never, rarely, sometimes, quite often*, and *extremely often*.

3.2.2 [Step 2]: Survey Questionnaire Development

After we chose the emotion scale for our study, we developed the survey questionnaire by following the guidelines

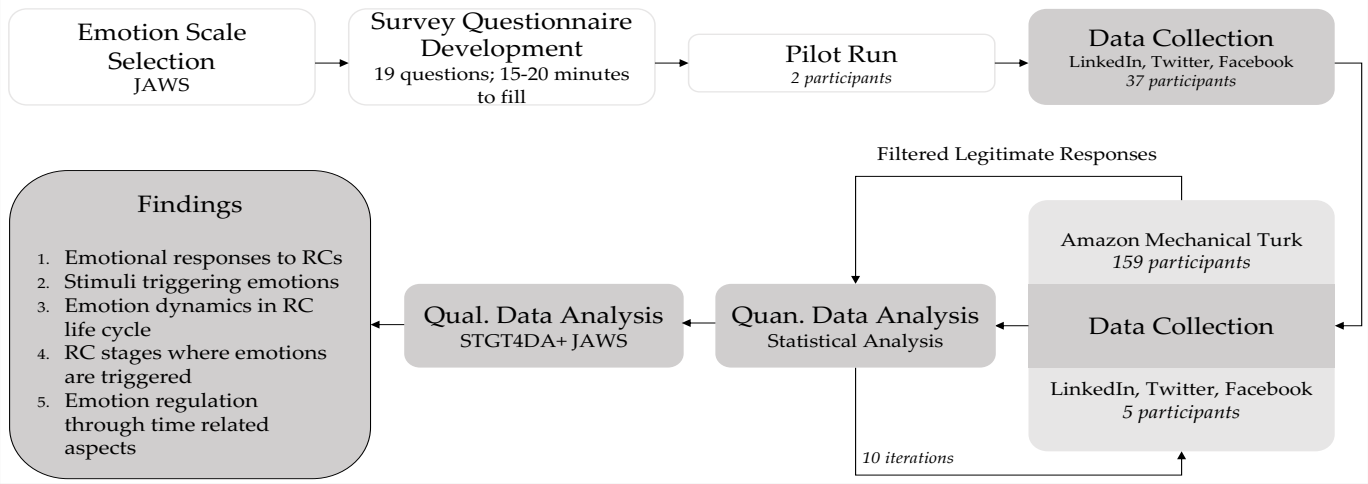

Fig. 1: Study Approach (Quan: Quantitative; Qual: Qualitative; JAWS: Job-related Affective Well-being Scale [5]; STGT4DA: Socio–Technical Grounded Theory for Data Analysis [20])

from Kitchenham et al.'s [43], [44], and Punter et al.'s [45]. The survey consisted of four sections (demographics information, project information, team information, emotional responses to RCs). Fifteen closed-ended questions were distributed among these sections, and four open-ended questions belonged to the emotional responses to RCs section which used the JAWS scale. The four open-ended questions represented each sub-scale of JAWS. The open-ended questions were customised and prompted for participants based on the answers they gave to the closed-ended question on emotions felt when handling RCs. If the participant chose *sometimes*, *quite often*, or *extremely often* for a particular emotion, the respective open-ended question was shown after answering the closed-ended question. This is illustrated in Fig. 2, also showing samples of closed-ended and open-ended questions. In the example given in Fig. 2, the choice "quite often" selected for the emotion *angry* prompted the respective open-ended question.

We used *Qualtrics*[3] as the survey platform and distributed the survey online following the Smith et al.'s study [46] to improve the survey distribution. We did not collect any personal information from participants, except from the participants who voluntarily provided their details for future interviews.

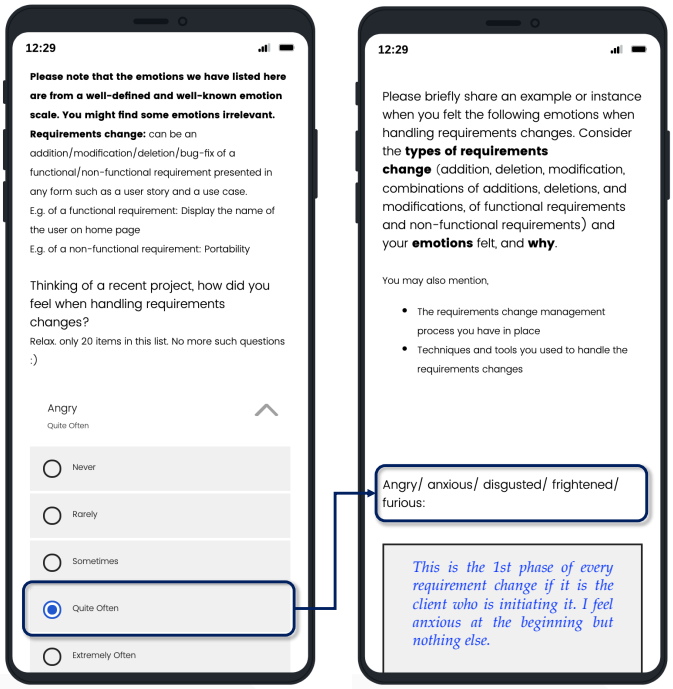

Fig. 2: Prompting of Open-ended Questions based on Answers Given to the JAWS based Closed-ended Question An answer given by a participant is shown in the blue text)

3.2.3 [Step 3]: Data Collection

Once the survey questionnaire was finalised, we sent the survey to two Ph.D. students who had recent industrial experience. They provided feedback about the survey in terms of time for completion. Before distributing the survey to the software development community, we calculated the sample size (random sampling) required for our study to represent the software development population via:

$$SampleSize = \frac{\frac{z^2 \times p(1-p)}{e^2}}{1 + (\frac{z^2 \times p(1-p)}{e^2 N})}; \quad (1)$$

where z=z-score, p=standard deviation, e=margin of error, N=population size. The tools we used to do this are available online[4].

According to Global Developer Population Report [47], the active developer population by 2019 was 18.9 M. We calculated the sample size required to generalise our findings to this population of 18.9 M. The sample size resulted in was N=385, by setting the confidence level to 95% (Z-score=1.96), standard deviation to 0.5, and margin of error to 0.05.

We distributed the survey by posting its link on social media such as *LinkedIn*, *Twitter*, and *Facebook*. The survey

3. https://www.qualtrics.com/

4. https://www.surveymonkey.com/mp/sample-size-calculator/

TABLE 2: Demographic Information of Participants (Dev: Developer; AC/SM: Agile Coach/Scrum Master; BA: Business Analyst; Other: $\leq$ 5 participants; XT: Total Software Development Experience; XTA: Total Agile Experience)

Location	# of Participants	Role	# of Participants
North America	96	Dev	75
Asia	40	Manager	21
Europe	24	BA	19
Australasia	22	Dev, Tester	14
South America	17	Tester	10
Africa	2	Dev, Manager	9
Gender	**# of Participants**	AC/SM	8
Male	115	AC/SM, Dev	7
Female	85	Other	38
Gender diverse	1		
XT	**# of Years**	**XTA**	**# of Years**
Minimum	1	Minimum	0
Maximum	35	Maximum	20
Mean	7.84	Mean	5.12

was available to the public for a period of one month and 20 days. The number of responses we received for the survey through this approach was limited. As we reached a plateau of 37 responses, we decided to recruit the rest of the participants via *Amazon Mechanical Turk* (AMT)[5]. The principal qualification criteria we used at AMT were: (1) *Employment Industry – Software and IT Services*; and (2) *Job Function Information Technology*. However, since we recruited the participants iteratively, we added additional criteria to different batches to have a uniform geographical and gender distribution as much as possible (purposive sampling). At the point of recruiting participants through AMT, we made a few changes to the survey questionnaire. One such change was making the open-ended questions mandatory. Later this became an advantage to us as we were able to filter the genuine responses by looking at the answers given to the open-ended questions. 49 responses were found as deceptive since unrelated answers were given to the open-ended questions. We rejected these responses and re-recruited participants until all responses appeared to be legitimate. We initially targeted recruiting the calculated sample of 385 participants. However, as we analysed the quantitative data after each round of data collection, we stopped collecting data when we were able to see clear results. i.e., when the number of responses for each emotion in the emotion scale became steady. Thus, we ended up recruiting only 201 participants. The survey was not limited to collecting qualitative data, but also collected quantitative data. A guide to recruiting participants using AMT is given in Appendix B. We would also like to highlight that the survey questionnaire filling is a task that appears on AMT for the participants that allow self–selection. Therefore, there could be a possible bias here, resulting in a threat to the validity of the findings. We have elaborated this further in Section 7.

The summary of demographic data of the participants is given in Table 2. The majority of participants represented North America (N=96; 47.78%) and the majority were developers (N=75; 37.31%). The participants had a mean total experience of 7.8 years (min(total experience)=1 year; max(total experience)=35 years). The summary of participants' most recent/current project is given in Table 3. The answers they gave to the survey questionnaire were based on these projects. The majority of the participants' projects were new developments (N=115; 57.21%), and the majority used agile methods in their projects (N=176; 87.56%) which is in line with reported agile use in the industry [48].

3.2.4 [Step 4]: Data Analysis

Table 4 summarises how the data collected were analysed using a mixed-methods approach which included a quantitative analysis and a combined qualitative analysis. Our mixed-methods approach included using the same instrument (JAWS) for data collection and analysis, besides analysing the open-ended questions. JAWS served as a tool for collecting the emotional responses of the participants (resulting in KF1, as listed in Section 1) through a closed-ended matrix question and to analyse qualitative data (resulting in KF3) in a combined manner along with STGT (resulting in KF2, KF4, KF5). We describe the analysis process in detail below.

Quantitative Analysis: The data collected through JAWS were descriptively analysed using *Python*. This resulted in emotional responses to RCs and Table 6 presents the analysis results.

Combined Qualitative Analysis: The collected qualitative data was stored and analysed in *MAXQDA*[6]. Additionally, *Microsoft Excel* was used when necessary. Categories of stimuli, emotions, emotion dynamics, RC stages, and time related aspects were identified from a combined approach of JAWS and *STGT for data analysis* [20]. We decided to use *STGT for data analysis* due to its ability to apply the basic data analysis techniques, such as open coding, constant comparison, and memoing (using diagrams in our case), within mixed-methods studies, in a limited capacity (as opposed to full theory development) and its suitability to analyse data in socio-technical research contexts. Our previous experience with applying open coding and constant comparison using the Strauss-Corbinian GT [49] also helped.

The combined qualitative analysis occurred in two steps: Combined Qualitative Data Analysis Step 1 (DA1) and Combined Qualitative Data Analysis Step 2 (DA2), as shown in Fig. 3. The analysis of raw data in DA1 yielded concepts such as *RC stages* and *time related aspects* along with their respective triggered *emotions*, and emerging *stimuli* that triggered them. Further analysis of emerging stimuli found in DA1, allowed us to group them under *RC, practitioner, team, manager,* and *customer*, in DA2. DA1 and DA2 are explained in detail below.

DA1 (STGT and JAWS): DA1 consisted of a combination of STGT basic data analysis and JAWS. STGT analysis resulted in identifying RC stages, and emerging stimuli and JAWS helped in identifying emotions. They were applied independently and complimented each other.

STGT data analysis: We followed a micro-analysis (line-by-line) approach in identifying the fragments of data in textual open-ended answers by the participants, since survey

5. https://www.mturk.com/: a crowd–sourcing marketplace that allows recruiting participants including for survey studies

6. https://www.maxqda.com/

TABLE 3: Information of Current/Most Recent Project of the Participants (XP: Extreme Programming; Other: ≤ 5 participants)

Project Domain	**# of Participants**	**Project Category**	**# of Participants**	**Development Method Used**	**# of Participants**
IT	122	New development	115	Scrum	57
Finance & Banking	30	Software as a Service	47	Dynamic System Development	44
Manufacturing	10	Maintenance	22	Feature Driven Development	25
Transport	10	Migration	17	Waterfall	17
Telecom	7			Kanban	14
Healthcare	6			Extreme Programming	14
Other	16			Crystal	11
Team Size	**# of People**	**Iteration Length**	**# of Weeks**	None	8
Minimum	1	Minimum	1	Other	11
Maximum	100	Maximum	10		
Mean	18.61	Mean	5.67		
Standard Deviation	18.8	Standard Deviation	2.11		
Practices Followed (Order of the Bars in Each Graph Below: Never → Sometimes → About half the time → Most of the time → Always)					
Collective Estimation		Product Backlog		Scrum/Kanban Board	
Customer Demos		Short Iterations/Sprints		Self-assignment	
Daily Standup/team meeting		Release Planning		Sprint Backlog	
Definition of Done		Retrospectives		User Stories	
Iteration Planning		Review Meetings		Use Cases	
Pair Programming					

TABLE 4: Mixed-methods Approach (DC: Data Collection; DA: Data Analysis; JAWS: Job Affective Well-being Scale [6]; STGT4DA: Socio-Technical Grounded Theory for Data Analysis [20]; DA1: Combined Qualitative Data Analysis Step 1; DA2: Combined Qualitative Data Analysis Step 2)

Method	**DC Technique**	**DA Technique**	**Key Finding**
Quantitative	Closed-ended matrix : JAWS Scale	Descriptive statistical analysis	KF1
Combined qualitative	Open-ended questions	STGT4DA+JAWS (DA1)	KF2 (emerging), KF3, KF4, KF5
		STGT4DA (DA2)	KF2 (emerged)

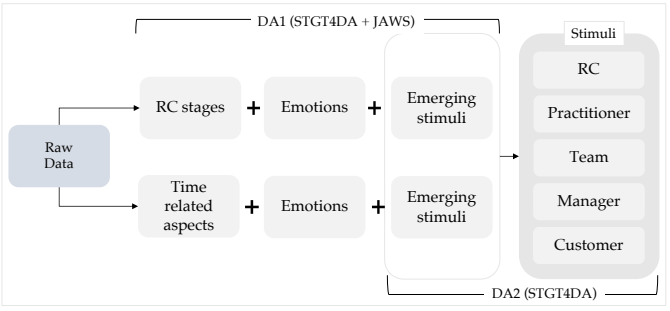

Fig. 3: Combined Qualitative Data Analysis Approach (STGT4DA: Socio-Technical Grounded Theory for Data Analysis [20]; JAWS: Job-related Affective Well-being Scale [5])

responses were 1–3 lines each. These identified data fragments were then meaningfully labelled (as codes). Through constant comparison, we grouped similar codes to form concepts, and then grouped similar concepts to form sub-categories, which are explainable through their characteristics (properties and dimensions). After the sub-category development, we further applied constant comparison to form categories.

JAWS data analysis: Since the open-ended questions listed the emotions in JAWS, most participants used the exact terms of emotions as in JAWS (see Fig. 2). When the answers in the open-text did not list any specific emotion, we considered the participant felt the majority of emotions in the emotion sub-scale of the question. This allowed us to extract the emotions from participants answers to the open-ended questions. Identifying emotions using JAWS was done independently and complimented the STGT analysis.

The extreme left-hand side of Fig. 4 (block A) shows the extraction of emotions and emerging stimuli at code level from the raw data (block B). Blocks to the right of the raw data show the emergence of the category "Stages of RC handling life cycle" in the order of analysis of raw data (block B) → codes (C1) → concepts (C2) → category (C3). Below, we elaborate this further by taking a single data point from Fig. 4. This data point represents the raw data that consists of (a) the distinct event at granular level, i.e., the code – "first phase of RC", which led to the concept – "receiving RC" with other similar codes as in Fig. 4, and that later resulted in the category – "stages of RC handling life cycle", (b) emerging stimulus which is at code level – "RC introduced by customer, and (c) the emotion – "anxious".

Raw data: *"This is the 1st phase of every requirement change if it is the client who is initiating it. I feel anxious at the beginning but nothing else." – P22 [Business Analyst]*

[JAWS] Emotion: Anxious

[STGT] Code: First phase of RC
[STGT] Concept: Receiving RC
[STGT] Category: Stages of RC handling life cycle

[STGT] Code (leading to the emergence of "Customer as a stimulus" in DA2): RC introduced by customer

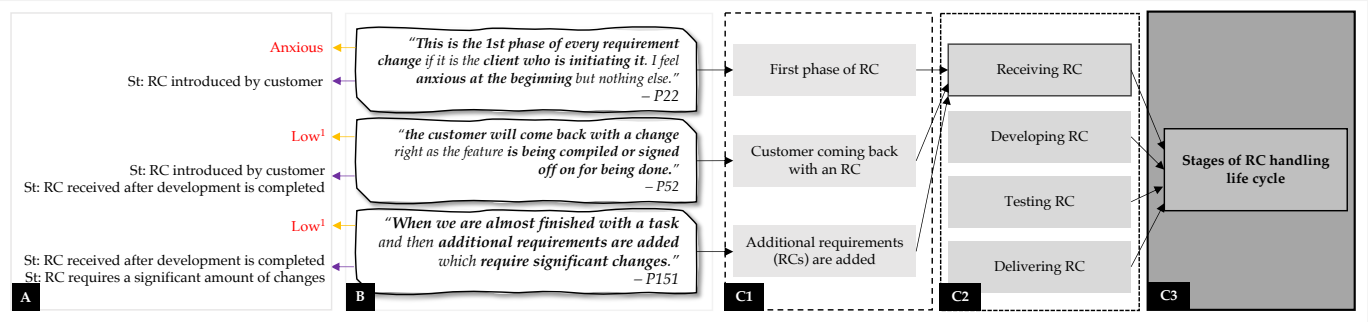

Fig. 4: DA1 Example: Emergence of the Category "Stages of RC handling life cycle"; Capturing of Emotions, and Stimuli (St: Stimulus), using JAWS (Block A) and STGT for Data Analysis (right-hand side of Block B)

DA2 (STGT): We further analysed the emerging stimuli that we found during DA1. We identified that these are certain aspects of *RC, practitioners, team, manager,* and *customer*. Grouping these together led to the category "stimuli".

During DA2, we also identified that a set of codes belong to certain well-established principles in Psychology (see Table 7). Therefore, we used those principles to conceptualise, and name the concepts "individual conation", "individual cognition", "social cognition", "team dynamics", and "emotional intelligence". In the example below, mapping to these Psychological concepts, we show the further analysis of emerging stimuli that we found during DA1.

[STGT] Code (captured during DA1): Practitioner's sustained attention
[STGT] Property [<dimension>]: Self cognitive skill [+]
[STGT] Concept: Individual cognition
[STGT] Sub-category: Practitioner as a stimulus
[STGT] Category: Stimuli

Further examples of DA2 which applied STGT for data analysis is given in Table 5.

At the end of combined qualitative data analysis: The strongest categories identified through our combined qualitative data analysis are as below.

Categories: Stimuli, Emotions, Emotion dynamics, Stages of RC handling life cycle, Time related aspects

Of these, "Emotions" were identified using JAWS, while the others emerged through the application of STGT for data analysis. We present these categories in detail in Section 4. We further found some emerging relationships as expected by the end of the STGT basic analysis, and we share these in Section 5.3 in the form of an emerging theoretical model.

Holistic views of categories stimuli, stages of RC handling life cycle, and time related aspects are given in Appendix C. The emotion categories we used were developed using the emotions in JAWS (Table 1). The emotion dynamics category, that presents the phenomena of emotion fluctuation over time, is presented in Fig. 5.

4 Findings

In this section, we present the key categories: Emotions or emotional responses to RCs (Section 4.1), Stimuli (Section 4.2), Stages of RC Handling Life Cycle (Section 4.3), Emotion Dynamics (Section 4.3), and Time Related Aspects (Section 4.4).

When both high and low pleasurable emotions are found at a given event, we mention it by the term *Mix*. In mix cases, we indicate the dominating emotion sub-scale within brackets. That is the sub-scale for which we found the most number of emotions.

4.1 Emotional Responses to Requirements Changes

The results from quantitative analysis provide a high level view of the emotional responses to RCs as a whole. We found that all high pleasurable emotions are more commonly felt by the software practitioners than low pleasurable emotions when handling RCs. Among the 10 low pleasurable emotions, only three low pleasurable emotions were found as commonly felt (*anger, anxiety,* and *fatigue*) when handling RCs. Table 6 summarises the feeling of a particular emotion within their current/most recent project when handling RCs as reported by the participants. We considered a certain emotion was felt commonly if the highest number of responses were found for "sometimes/quite often/extremely often" options.

4.2 Stimuli Triggering Practitioners' Emotions at RC Stages

While *RCs* act as the central stimulus in our study, through our STGT analysis we found several stimuli across RC stages, and some time related aspects. By careful observation, we noticed that these are not stand-alone stimuli, but properties of a set of associated stimuli that lead to the triggering of emotions of practitioners. These associated stimuli are stakeholders, including *practitioners* themselves, their *team*, their *manager*, and their *customer(s)*. Below we list these stimuli found in the specific stages of RC handling life cycle.

Central stimulus: RC. The properties of an RC that enable the triggering of emotions include RC stability, its point of introduction, frequency of introduction, impact, definition, type, changing extent, status, and challenging nature. Among these, *point of introduction* is the leading

TABLE 5: DA2 Example (Emergence of Sub-Category: "Practitioner as a Stimulus" from Stimuli found during Data Analysis Step 1; [+]: Positive dimension of the property; [-]: Negative dimension of the property)

Code (Emerging stimuli after DA1)	Property [<dimension>]	Concept	Sub-category (Stimuli after DA2)
• Practitioner motivated to put much effort • Practitioner motivated since not coded in a while	Self motivation [+]	Individual conation	Practitioner as a stimulus
• Practitioner feeling that hard work is wasted	Self motivation [-]	Individual conation	Practitioner as a stimulus
• Practitioner knowing the possibility of successful delivery	Self-efficacy [+]	Individual cognition	Practitioner as a stimulus
• Practitioner perceiving themselves as a perfectionist	Personality perception [self]	Individual cognition	Practitioner as a stimulus
• Practitioner's sustained attention • Practitioner's predictive ability • Practitioner's ability to work timely	Self cognitive skill [+]	Individual cognition	Practitioner as a stimulus
• Practitioner perceiving team lacks cohesion • Practitioner perceiving team lacks skills • Practitioner perceiving team has skills • Practitioner perceiving the lack of emotional intelligence (social awareness:empathy) of the manager • Practitioner perceiving the lack of emotional intelligence (social awareness: empathy) of the customer	Social cognition [+]	Social cognition	Practitioner as a stimulus

TABLE 6: Emotional Responses to Requirements Changes

Emotion	Never	Rarely	Sometimes	Quite Often	Extremely Often
High Pleasurable High Arousal Emotions					
Ecstatic	22.39%	26.87%	29.85%	14.93%	5.97%
Energetic	5.47%	15.92%	27.86%	32.84%	17.91%
Enthusiastic	7.46%	11.94%	28.86%	34.33%	17.41%
Excited	8.46%	16.42%	29.85%	24.88%	20.40%
Inspired	9.45%	15.42%	23.88%	30.35%	20.90%
High Pleasurable Low Arousal Emotions					
At-ease	2.99%	23.88%	39.30%	26.37%	7.46%
Calm	2.49%	17.91%	29.35%	40.30%	9.95%
Content	6.97%	21.89%	31.34%	29.85%	9.95%
Relaxed	10.95%	15.92%	33.33%	30.85%	8.96%
Satisfied	6.47%	14.93%	27.36%	28.36%	22.89%
Low Pleasurable High Arousal Emotions					
Angry	16.42%	31.84%	40.80%	9.45%	1.49%
Anxious	10.95%	24.38%	34.83%	24.38%	5.47%
Disgusted	44.28%	26.37%	15.92%	10.95%	2.49%
Frightened	27.36%	31.84%	25.87%	10.95%	3.98%
Furious	30.85%	29.35%	21.39%	13.43%	4.98%
Low Pleasurable Low Arousal Emotions					
Bored	31.34%	26.37%	24.88%	14.93%	2.49%
Depressed	29.35%	32.84%	24.88%	10.45%	2.49%
Discouraged	24.38%	31.34%	28.86%	12.44%	2.99%
Fatigued	9.45%	28.86%	34.33%	20.40%	6.97%
Gloomy	24.88%	32.34%	22.39%	15.92%	4.48%

property that trigger the emotions of practitioners. The definitions of these properties are available in Appendix C.

Associated stimuli: Stakeholders. *Practitioner as a stimulus:* The practitioner's individual conation (the proactive part of motivation. See Appendix A for full definition), individual and social cognition make the practitioner themselves a stimulus. As practitioners perceive team dynamics, and emotional intelligence (EI) of their managers and customers, their social cognition is also associated with other human stimuli. *Team as a stimulus:* Team dynamics, such as collective skills and cohesion, results in *team* being a stimulus of emotional response to RCs. *Manager as a stimulus:* Managers of the practitioners act as stimuli because of their emotional intelligence and being the source and carriers of some RCs. *Customer as a stimulus:* Similar to manager, customer's EI and being the source and carriers of RCs make them a stimulus for the emotional responses to RCs of practitioners. Similar to the central stimulus, the definitions of the terms under associated stimuli are available in Appendix C.

4.3 Emotion Dynamics of Practitioners in Requirements Change Handling Life Cycle

We discovered additional stages of the RC handling life cycle to the stages we found in our preliminary study [11]. Our initial research found the *receiving*, *developing*, and *delivering* stages where practitioners respond emotionally. In addition to that, through this study, we found the *testing* stage, sub-stages of the developing stage (*beginning, writing code, coding completed, troubleshooting*), sub–stages of delivering stage (*almost completed, completed, released/delivered*). Fig 5 depicts a comprehensive illustration of these stages, corresponding emotions triggered at the stages, and found stimuli that trigger the emotions. For example, in the "delivering" stage, a sub–stage called "almost completed" exists, and the "relaxed" feeling is triggered by the stimuli "practitioner" as shown in Fig. 5. Further, we found that practitioners feel a combination of high^2 and high^1 emotions *throughout development* in general.

4.3.1 Emotion Dynamics Pattern across Requirements Change Stages

Fig. 6 shows the emotion dynamics pattern of the RC handling life cycle. The emotions fluctuate in a **Mix (Low^1)** $\rightarrow$ **Mix (Low^1)** $\rightarrow$ **Mix (High^2, High^1, Low^1)** $\rightarrow$ **High^2** $\rightarrow$ **High^2** $\rightarrow$ **High^1** $\rightarrow$ **High^1** $\rightarrow$ **High^2** $\rightarrow$ **High^2** way. Only 2 high^2 emotions (*energetic, excited*) were found at the *receiving* stage, and the rest were low^1 emotions (e.g.: *angry*), and low^2 emotions (e.g.: *depressed*). At the *developing* stage, a mix of emotions was found where the domination of high^2, high^1 (e.g.: *calm*), and low^2(e.g.: *bored*) while *writing code* was seen.

As indicated by the black dotted box in Fig. 6, the transition of emotions happens when writing code in the development stage.

However, when the *coding was completed* and during *troubleshooting*, high^2 emotions (e.g.: *inspired*) were prominent. Then, by the *testing* and *delivering* stages, high^1 emotions (e.g.: *relaxed*), and high^2 emotions (e.g.: *excited*) were seen.

4.3.2 Stimuli at Requirements Change Stages

In this sub-section, we present the stimulus property contributing to emotion triggering next to the RC stage where it was found as well as some supporting quotes from the participants. The dominating stimulus at each RC stage is shown in Fig. 6.

Receiving stage [*RC* – point of introduction, impact of RC on other requirements, RC definition, RC type, changing extent; *Customer* – carrier of RC; *Practitioner* – individual conation]: The majority of the stimuli found at the receiving stage of an RC were related to the low^1 and low^2 emotions. *RCs point of introduction* – such as receiving after development is significantly/fully completed, during testing and in the middle of the project make practitioners feel low pleasurable emotions. e.g.: *"I would think we were done and then more changes would be asked as others reviewed. It left me depressed and fatigued at times." – P191 [Business Analyst].*

Impact of RC on other requirements, where a significant amount of inter-dependencies exist, is another stimuli which triggers the emotions of practitioners. Similarly, *definition of RC*, i.e., its initial learning where the practitioners learn the clear meaning of RC, and *type of RC* such as improvements for better user experience, also trigger practitioners' emotions at this stage. Also, *source* and *carrier of RC* such as manager, and customers also trigger practitioners' emotions such as *anxiety*. e.g.: *"This is the 1st phase of every requirement change if it is the client who is initiating it. I feel anxious at the beginning but nothing else." – P22 [Business Analyst], practitioners' individual conation* – where the practitioner is motivated to put much effort to work on the RC – makes the practitioners *energetic* when they receive the RC. e.g.: *"Usually I feel more energetic when the work is assigned and there's a lot of work to do" – P148 [Developer, Tester].*

Development stage [*RC* – code status; *Practitioner* – individual conation; *Team* – Team dynamics]: Practitioners are *energetic* in cases where they are motivated. For instance, when the practitioners have not coded in a while, they feel energetic. e.g.: *"I felt energetic at the start of the development as I haven't coded for a while." – P9 [Developer].*

High^2 emotions occur when the team is rapidly solving issues i.e., where positive team dynamics occur. For example, working as a team which is ready to embrace change, overcome difficulties, and be able to solve issues rapidly makes practitioners feel *ecstatic* and *inspired*. e.g.: *"As a crack team, bracing for any sort of change, especially changes in requirements, was like second nature to us. Whether planned or abrupt, we implemented any and all changes without needing to break much of a sweat. Sure, there were moments - quite rare, honestly - when we ran into some difficulties, but getting to troubleshoot and then rapidly resolve them left us all ecstatic and - if nothing else - greatly inspired." – P101 [Manager].*

High^1 emotions, such as *satisfaction* and *contentment*, are prominent where attention is sustained for a certain period of time, a cognitive skill e.g.: *"After a few solid hours of coding I did feel some satisfaction and content." – P9 [Developer].*

Negative team cohesion – for example having disagreements among team members, a team dynamic, results in low^1 emotions. e.g.: *"I felt this emotion at the beginning because there were some obstacles with disagreements between the developers" – P181 [Developer, Manager, Product Owner, Tester].*

On the other hand, when the code written by the practitioner works well, the practitioners feel high^2 emotions. e.g.: *"I felt all [high^2 emotions] this when the written codes work well." – P65 [Tester].*

Testing stage [*Practitioner* – individual cognitive skills]: Timely fixing of bugs and tests passing as expected

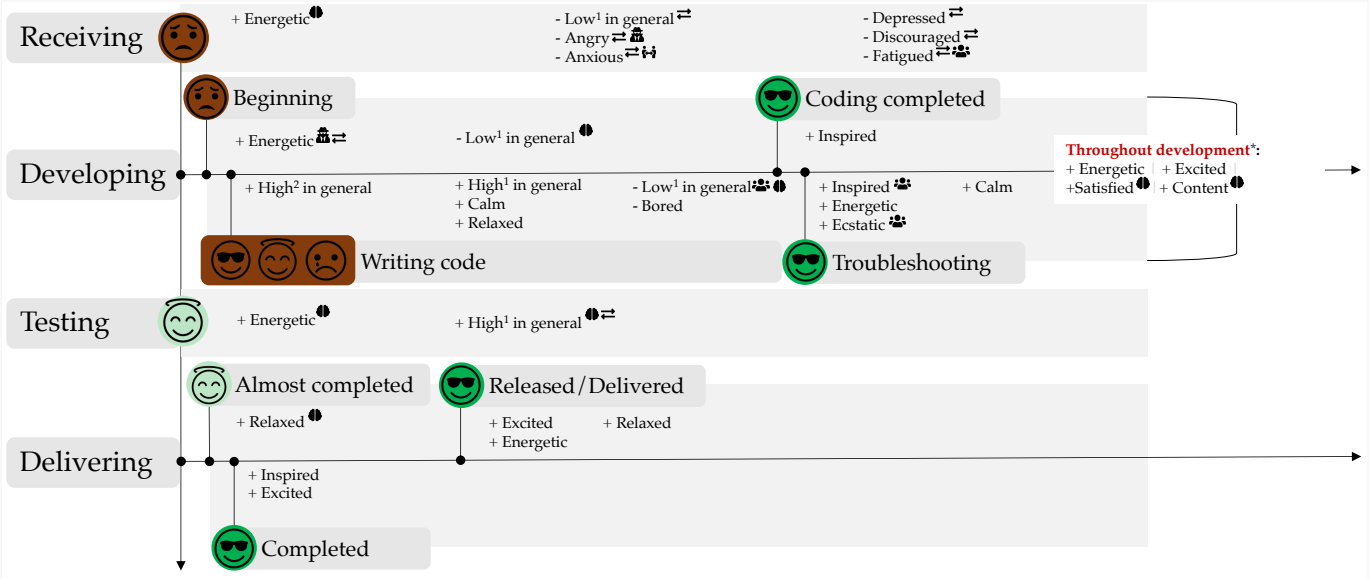

Fig. 5: Emotion Dynamics of Practitioners in Requirements Change Handling Life Cycle (*in general while specific emotions are triggered at specific stages; Emoji: Dominating emotion sub-scale; : High^2; : High^1; : Low^1; : Low^2; : both high/low emotions exist; Stimuli: : RC, : Practitioner, : Team, : Manager, : Customer)

trigger high[1] emotions of the practitioners. e.g.: *"energetic at every time that the testing was occurring as expected." – P97 [Tester].*

Delivering stage [*Practitioner* – self–efficacy]: We found that when practitioners' self-efficacy reported by *P89*, where practitioners know the possibility of successfully delivering the RC, trigger a more relaxed frame of mind in them. e.g.: *"When I was closer to the end and all the most dangerous and boring parts had passed, I ended up relaxing because I knew I could deliver." – P89 [Developer].*

4.4 Emotion Dynamics Regulation by Time Related Aspects

We found that adjusting time related aspects can regulate practitioners' emotions. Fig. 7 shows the key time related aspects and related emotional responses of the practitioners in our survey, and Fig. 8 explained in Section 5 shows how this adjustment could be made possible practically.

As reported by our participants, a widespread incident they face in their projects are "last minute" RCs. Such a "last minute" could be closer to a set deadline. i.e., could be by the end of an iteration where a feature is anticipated to be released, or by the end of the project. We found that they feel low[2] emotions due to such last minute RCs. e.g.: *"A client wanted several significant changes made shortly before a deadline. I was very angry because they should've mentioned this earlier in development and having to make broad sweeping change so close to the deadline was very frustrating." - P138 [Developer].* We also found that they feel low[1] emotions, such as *anxiety*, when they do not have enough time to work on the RC. e.g.: *"I felt anxious because I was worried about whether or not I had enough time to make the change and complete the project on time" - P150 [Developer, Tester].* However, when they are able to meet the RC deadline, they feel high[2] emotions. e.g.: *"When they added a few usability enhancements due for the next day and I managed to set them in time for the deadline." - P180 [Tester]*

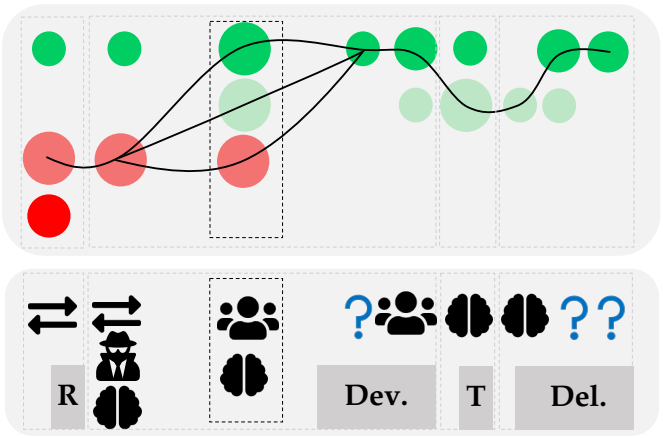

Fig. 6: Emotion Dynamics Patterns (1st row: Dominating emotions; 2nd row: Dominating stimuli; ●: high[2]; ●: high[1]; ●: low[1]; ●: low[2]; Size of the circle: more the number of emotions, larger the circle; ⇄: RC; 🧠: Practitioner; 👥: Team; 🕴: Manager; 🤝: Customer; ?: Unknown; line: only to emphasise the fluctuation of emotions; reading order: In line with stages in Fig. 5, each column is a stage/ sub–stage: R: Receiving, Dev.: Developing; T: Testing; Del.: Delivering)

4.4.1 Emotion Regulation of RC Handling Life Cycle

If an RC fits the project timeline, practitioners feel high[2] and high[1] emotions, thus making it the way to direct emotion regulation towards high pleasure. e.g.: *"When useful user-facing features are being added, when there is time to do so." – P10 [Developer], "When there's plenty of time for implementing new requirements (rarely) because there's no reason to be up" – P180 [Tester], "If there is enough time, and the change is not a big modification in which I have to undone my work." – P125 [Developer], "This happens only when the changes are easy to implement and are not time-consuming. Some examples are the wordings on screens, minor layout changes, additional variations, or changes that can be implemented without major changes to the existing application design or program flow. Such changes are not a cause for concern and are usually easy to accommodate within the original estimates." – P128 [Developer]*

4.4.2 Stimuli in Time Related Aspects

The majority of stimuli in time related aspects we found were about the RC, especially if its *impact on scope* is minor. e.g.: *"The project added a minor and easy to develop requirement which did not add significantly to the timeline or scope." - P183 [Business Analyst]*, more significant changes in scope. e.g.: *"When the project scope changes and extended the project." - P170 [Business Analyst]*, on other requirements. e.g.: *"When clients made last minute adjustments to planned feature" - P59 [Developer]*, and on design/program flow.

We found that *RC's challenging nature* is also a significant stimulus in time related aspects, especially when they are easy to implement, and not consuming time. e.g.: *"This happens only when the changes are easy to implement and are not time consuming. Some examples are the wordings on screens, minor layout changes, additional variations, or changes that can be implemented without major changes to the existing application design or program flow. Such changes are not a cause for concern and are usually easy to accommodate within the original estimates." - P128 [Developer]*, and a stimulus related to **RC's delivery** was found. e.g.: *"When they added a few usability enhancements due for the next day and I managed to set them in time for the deadline." – P180 [Tester]*

We also found stimuli related to *lack of EI of the customer*, and *social cognition of the practitioner*. The practitioners are able to perceive that the customer lacks EI (social awareness: empathy). e.g.: *"Late in the dev phase of a new work project, my client changed their requirements; specifically they wanted to change how their data is loaded into the application, changing it from an XLS file to a database table. I had to rebuild the file loader completely. It made me feel angry because I felt they do not understand my position." – P150 [Developer, Tester].*

There is also *individual conation*, where practitioners are demotivated due to lack of EI of customer and when they act as the source, and carrier of RC. e.g.: *"Well we had almost completed the given task suddenly client called and changed their idea to an whole new thing. That moment I felt that my hard work was just wasted I felt very anxious while deleting and creating a new one crystal orange." – P142 [Developer]*

Practitioners are impacted by perceiving their own personality. e.g.: *P150* mentioned that she is careful about completing

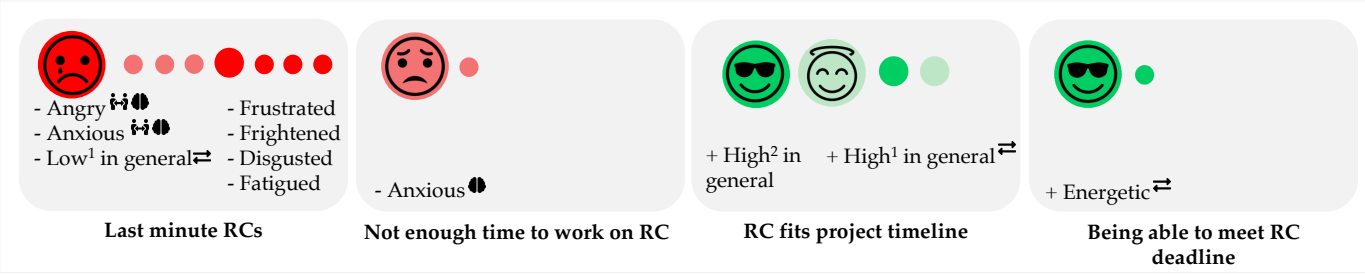

Fig. 7: Time Related Aspects which Regulate Emotional Responses (Emoji: Dominating emotion sub-scale; : High[2]; : High[1]; : Low[1]; : Low[2]; Stimuli: : RC, : Practitioner, : Customer)

a RC within the given time frame, as she does not want to have a bad impression about herself because she perceives herself as a perfectionist. e.g.: *"I felt anxious because I was worried about whether or not I had enough time to make the change and complete the project on time, otherwise it would just look bad on me regardless of the requirements being changed, and I am a perfectionist." – P150 [Developer, Tester].*

Practitioners' *social cognition*, where they are perceiving their *manager lacks EI*, impacts their emotions. When a manager is not empathetic enough to feel how the team feels, the practitioners feel a loss of energy. In this case, when the practitioners require more time to work on the RC, as new issues appear and they need to be solved – and yet their manager still requires the work to be delivered on time – low pleasurable emotions are triggered in the practitioners. This situation not only triggers low pleasurable emotions, but also results in the loss of the flow of work for the practitioners. e.g.: *"I worked more and more time each day to try to successfully make the job, and new issues appeared every day that needed more and more time. We were all at same feelings but team leader didn't hear his team feeling. He requested us to perform this project on at 'just in time'. I lost energy and I worked without any positive feedback. I lost my flow." – P173 [Developer]*

5 Discussion

5.1 Comparing to Related Work

In Colomo–Palacious et al.'s comparison work [50] on emotions felt while presenting and coding, they found that *anxiety* and *nervousness* are felt when presenting and *satisfaction* and *enjoyment* are felt when coding. This emphasises the human element in triggering emotions. This is further confirmed through our work where we identified that "humans" in the software context trigger the emotions of the software practitioners when handling RCs, thus making an impact on their emotional well–being. As we found collaboration and self–perception trigger the emotions of practitioners when handling RCs, Giradi et al. [13] also emphasises that the same trigger in SE contexts. In this paper, we have explained how different characteristics of the RC trigger emotions of the software practitioners. Given that when the RC is not stable, i.e., when the RC continues changing, low pleasurable emotions are felt by the practitioners. In relation to this, Colomo–Palacious et al. [19] found that pleasure felt when the requirements are final is high, i.e., when the requirement is no longer changing and stable.

As Wrobel mentions in their study [15], *frustration* is commonly felt in SE contexts, and lowers productivity, whereas *anger* and *enthusiasm* increases the productivity. Wrobel also states that emotions transit from *frustration* → *anger* → *contentment* → *enthusiasm*. This is similar to emotions in the RC handling life cycle we found, where low pleasurable emotions dominate the first stages and high pleasurable emotions dominate the last stages.

5.2 Implications for Practitioners

Emotion-centric Decision Guide. We present an emotion-centric decision guide that we created using the above implications in Fig. 8 and this is available online to download[7]. This dual-purpose guide caters for the carriers of RCs by helping them when to decide when to introduce an RC, and for practitioners to decide when to accept an RC by being considerate about their emotions. The checkpoints [A], [B], and [C] require the carriers of RCs and practitioners to work collectively. The colors of the circles indicate the commonly felt emotions. Green indicates that high pleasurable emotions are commonly felt, and red indicates low pleasurable emotions are commonly felt at that particular point. Following are a couple of possible use–cases of using this guide in practice.

Use–case 1: Emotions of the team are a high priority and RC may be delayed – Customer (carrier) has an RC to be introduced, but the customer and team manager are very much concerned about how the team feel when handling the RC, especially if it will be a challenging one. Therefore, both customer and manager decide to use the emotion–centric guide to help them make a decision as to whether to introduce the RC now. The customer and manager use the emotion–centric decision guide to see where the team emotionally are at. Together with the team, the customer and manager decide whether or not to introduce the RC.

Use–case 2: Emotions of the team are important, but the RC is mandatory – The customer has an urgent RC to introduce which cannot be delayed or avoided. The customer however, uses the emotion–centric decision guide to see where the team is in right now to get an understanding of the team's emotions. The customer introduces the RC but, empathising with the team, does so in a way that is cognisant of the team's current emotional state. However, the customer and manager realise this was an exception

7. https://kashumim.com/tools/isnowgood/

and do not apply it frequently to avoid severe negative emotional impacts. The team appreciates the urgency of the RC and its careful introduction, and decides to implement the RC. Technical concerns such as interdependencies will be checked by the team before initiating the implementation.

Recommendation 1. Practitioners monitoring of their own emotions and managers monitoring of their team's emotions is necessary. *Anxiety* is common among practitioners. Monitoring emotions helps improve their EI through self-awareness, and eventually helps take actions (self-management) to maintain positive emotional well-being, thus providing the support the practitioners need in terms of emotions. Emotion tracking and monitoring tools exist. For instance, *Emotimonitor* [51], the tool we developed for agile teams, could be used to track and monitor self and team emotions.

Recommendation 2. Customers and managers should improve their emotional intelligence. Customers and managers lacking EI impact the emotional well-being of practitioners. As practitioners have EI, they do know when their managers, and customers lack EI. To have a better relationship with the practitioners, customers and managers are required to improve their EI. Since we found a lack of empathy identified by the developers, customers and managers, empathy is specifically needed to be improved and prioritised. For example, Bregman [52], explains how prioritising empathy can make a conversation more productive.

Recommendation 3. Being considerate of when to introduce/accept RCs is necessary. In relation to the previous recommendation, the point of RC introduction plays a significant role in the emotional responses of practitioners. Avoiding or reducing last minute RCs may mitigate or reduce the feeling of low^1 and low^2 emotions of the practitioners.

Recommendation 4. When possible, extending the project timeline where necessary, and add new RC to the next iteration instead of the current one may help maintain high^2 and high^1 emotions of the practitioners. For example, when there is not enough time for the practitioners to work on the RC, they are *anxious*. However, when the RC fits the project timeline, practitioners feel high^2 and high^1 emotions. Therefore, allocating enough time, which may result in an extension of the project timeline, may awaken the practitioners' high^2 and high^1 emotions. However, this recommendation is applicable if and only if the deadline extension is possible with all other business factors considered.

Recommendation 5. Do not abuse agile values and principles. Finally, we urge the carriers of RCs and practitioners not to abuse the agile value "*responding to change over following a plan*" by entirely avoiding to follow a plan, and the principle "*welcoming changing requirements even late in development*" by regularly introducing late and out of sprint plan changes, placing teams under extreme time pressure. Following guides (for example, the dual–purpose guide we have presented in this paper) which provide lightweight RC guidance plans may ultimately help maintain better practitioner emotional well-being.

5.3 Implications for Researchers

Specific stimuli dominate the emotion dynamics at RC stages. In the RC handling life cycle, the **practitioner as a stimulus** dominates the majority of the stages and then the RC. This hints that even though RC acts as the central stimulus, the practitioner's individual conation (the proactive part of motivation. See Appendix A for full definition), individual cognition, and social cognition play a larger part in triggering their own emotions. We anticipate to see research on how practitioners explore, utilise, and improve these aspects.

RC stages act as stimuli. We define RC stages as the specific points in the RC handling life cycle at which the stimuli trigger the emotions of the practitioners. However, at some sub–stages in RC stages – when the coding is completed (in the developing stage), RC is completed (in the delivering stage), and RC is released/ delivered (in delivering stage), we were not able to find the stimuli which triggered the emotion, but the emotion itself. This could be because the distinct event itself triggered the emotion and nothing else, or there are stimuli triggering emotions where participants did not mention. In the former case, it can be said that the RC stages also fall under stimuli. This suggests that in some stages of the RC handling life cycle, where central/ associated stimuli trigger the emotions of the practitioners, both the stage and the respective stimulus trigger the emotions of the practitioners collectively. We encourage researchers to study this further and improve the big picture of RC handling life cycle.

Using JAWS to assess the emotional responses when handling RCs and in SE research. To use an emotion assessment tool in any discipline, including SE research, it is necessary to experiment with it in several scenarios. For example, in their work [53], Cummaudo et al. found a common emotion tagging framework used in SE is not fit for purpose and needs tailoring to specific domains. We have only used JAWS in our research that focuses on emotional responses to RCs. Our experience with it so far is positive as we were able to achieve our objective where we were able to gain a reasonable understanding of emotional responses to RCs. However, more studies using JAWS would help the researchers to establish a greater degree of accuracy and usefulness on this matter.

6 Future Work

6.1 Future Work on Theory Development

The emerging theoretical model. At the end of STGT basic data analysis stage, we found emerging relationships among the categories. We noted these down and used them to aid further discovery and strengthening of the connections. The first author drew the emerging model on paper iteratively, and shared and discussed it with the other two authors until the current model (as presented in Figure 9) was formed. The emerging relationships identified among categories are given in bold italic text below.

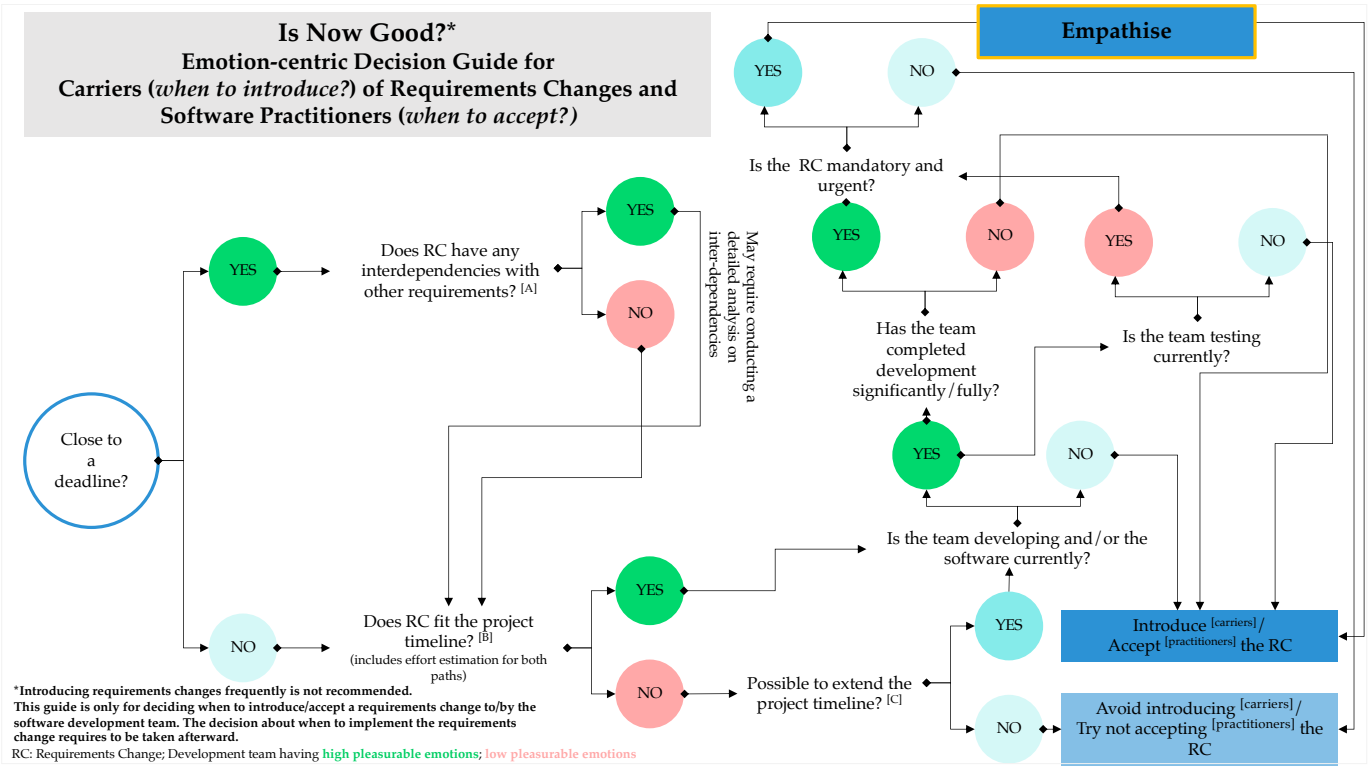

Fig. 8: Dual-purpose Guide: Emotion-centric Decision Guide for Carriers of Requirements Changes and Practitioners

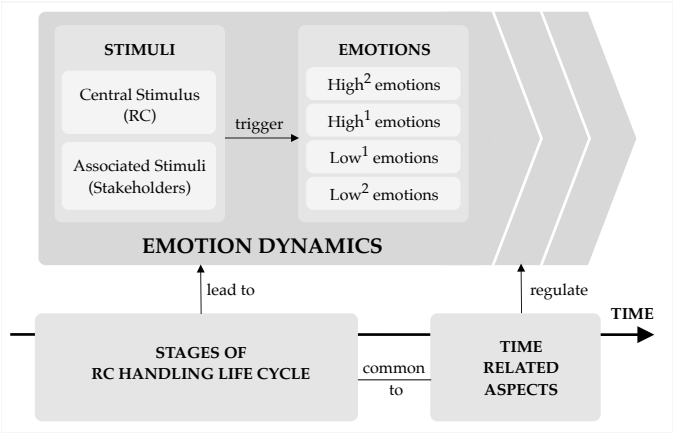

Fig. 9: Emerging Theoretical Model: Emotion Dynamics of Software Team Contexts

Emerging relationships among categories:
Emotion dynamics ***is the fluctuation*** of emotions over time; Stimuli ***trigger*** emotions; Stages of RC handling life cycle lead to emotion dynamics; Time related aspects ***common to*** stages of RC handling life cycle ***regulate*** emotion dynamics

We suggest future work on this, where researchers could either choose the emergent or structured mode of STGT to develop a full theory. For instance, if the researchers choose the structured mode, the next steps will include structured data collection, structured data analysis, advanced memoing, and finally theoretical integration [20].

The model representing the emerging relationships is given in Fig. 9. The core category at this stage is:

Emerging core category (central phenomenon):
Emotion dynamics in software team contexts

6.2 Insights for Future Research

Our findings and our thoughts on the findings enabled us to derive some key insights about agile (as a majority of our survey participants practice agile methods), emotional intelligence, and manager vs non–manager emotional responses to RCs. These insights could be used as starting points for future research.

Agile values welcome changes but agile practitioners are not always *pleased* with requirements changes. The Agile Manifesto values responding to change over following a plan [3]. Practitioners are seen to respond to RCs but the nature of their response varies. For example, our findings from in-depth qualitative analysis show that practitioners are not always pleased with RCs and show a mix of emotions throughout the RC stages. We encourage researchers to explore how the nature of responses vary when responding to RCs and also the changes beyond RCs.

Welcoming late RCs on a regular basis can be detrimental to the practitioners' emotional well–being. Even though the agile manifesto encourages welcoming changing requirements, even late in development [3], it is notably clear from our findings that last minute RCs trigger low pleasurable emotions of practitioners. By the end of the

project, or even by the end of an iteration, if closer to a deadline, the above-mentioned emotions are felt when an RC is introduced. Therefore, welcoming late RCs on a regular basis can be detrimental to the practitioners' emotional well–being. This would be a fruitful area for future research.

There could be differences in emotional responses to RCs in agile and traditional software projects. In a typical agile software development project, when an RC is received in the middle of an iteration, it is meant to be implemented in an upcoming sprint. This is after following several practices, such as breaking the RC down into manageable tasks, estimating effort, assigning work, and executing the tasks. How this plays out in practice, and how it compares to the impact on the emotional well–being of practitioners working on traditional software development projects remains to be seen. A future study on comparing the emotional responses to RCs in traditional software development could provide a better understanding.

Emotional responses to agile practices. As we found that the emotional responses in handling RCs vary, and as many of our participants practise agile in their software development work, researchers may look into the variation of emotional responses when practising specific agile practices. Colomo–Palacious et al.'s work [50] also confirm that emotional responses vary when coding and presenting. Therefore, this could be a fruitful area to study in the future.

Non–agile practices in agile contexts. Our findings indicate that in practical agile contexts, introducing RCs in the middle of the iterations take place, managers exist in agile contexts (also found by Shastri et al. in their work [54]) and by being a carrier of RC, they contribute in acting as a stimulus in triggering practitioners' emotions. These can be said as non–agile practices in agile contexts. Therefore, we suggest researchers to study in the future, why such practices exist in practical agile contexts and what can be done to improve the agility in the context.

Practitioners have emotional intelligence. But the practitioners perceive that their managers, and customers lack it. EI has four aspects, namely, self-awareness (aware of own emotions), self-management (manage own emotions), social-awareness (aware of others emotions), relationship management (build relationships with the use of other three aspects). Our study suggests some preliminary findings to show that practitioners possess self-awareness of emotions and tend toward self–management too. When it comes to social-awareness, some responses suggest practitioners find their managers lacking in emotional awareness. This also could have been seen different from the perspective of the managers and customers. Therefore, it is essential to collect data from managers and customers to compare and contrast how EI vary according to the role. Future work can explore these aspects of EI in detail.

7 Threats to Validity and Evaluation

External Validity: Equal geographic distribution of participants was not achieved. Almost half of the participants of our study were from North America, and therefore, generalizability of our findings is limited. Similarly, participant counts across the genders was not equal. Majority of participants were male, rest were female except a single gender-diverse participant. As emotions are bounded to the biological nature of humans, including the gender, we see this as a threat to validity. This also could be potential issue for future work to explore. However, in both of the above-mentioned cases, we tried our best to recruit representative participants. While there is a large imbalance of males vs females in the software industry, we did our best to try and recruit a gender balanced group of practitioner participants. This study was conducted during Covid-19 global pandemic. Mental health was a persistently talked and researched topic during this time. As emotions heavily impact the mental well-being, the results of this study may have been threatened by the pandemic situation. A study on work of software engineers during Covid-19 also found the impact of it on the mental well-being [55]. However, this situation was an uncontrollable one, therefore stays as a threat to the validity of the findings. Hence, the findings may not be equally applicable to software contexts in a post–Covid world.

Internal Validity: The first author went through the emotion scales carefully and assessed the applicability of the scale in this study. Then all authors discussed the findings to come to a conclusion on the best emotion scale to use in the study. This measure mitigated internal validity issues that could have been caused by the emotion scale. The particular kinds of RCs the participants talked about could have been different. For example, minor vs major RCs, and different types of RCs. Similarly, the quality of change management process maybe different from one participant to another. However, we did not ask the participants about the RC explicitly or their quality of change management process. Therefore, this exists as a threat to the validity of the findings. However, it is not possible to ask the participants to elaborate on the RC or their change management process more as we used a survey to collect the data. This threat could have been mitigated if semi–structured interviews were used to collect data. The first author analysed all data and the emerging findings were presented to the second and third authors during fortnight meetings where the findings were discussed. For a few codes, the authors disagreed. Such codes were carefully checked and mutually agreed upon through discussions. We also consulted a psychology expert to ensure the correctness of terminology use, and our analysis of data.

Construct Validity: Given the Covid-19 pandemic situation, we were not able to conduct any experience sampling or observations to collect emotional responses shown at the exact moment, as we planned. Therefore, we decided to carry out an online survey allowing participants to self-report their emotions by thinking about their current/recent experiences. Our piloting experience, for this and other surveys studies, suggests that survey respondents tend to skim read or skip instructional text and focus more on answering the questions. Even with the most comprehensive instructional text, a survey can hardly compete with the contextual consistency that, say for example, an interview can provide. Having said that, for similar surveys in the future, we will try to make it clearer that participants need to consider the same context for answering all the questions.

Our findings rely on JAWS. JAWS is used to assess emotional reactions of people over the past 30 days. However,

in our case, it is impossible to assume that project and RC handling life cycles were limited to a period of 30 days. Therefore, this remains as a threat to validity. Additionally, the number of emotions vary across the different emotion scales. Therefore, the analysis may differ if other emotion scales are used. In addition, we assumed that emotions listed in the scale are understandable and interpreted in the same way by the participants, as none of our pilot study participants provided any negative feedback on understandability of the survey questions. Moreover, as participants may be reluctant to report their negative emotions [15], this is a possible threat. On the other hand, there could be a possibility that our participants self–selected the survey based on their emotional maturity and having experienced issues in their software development processes. This could have happened especially with AMT participants, as they have been given the opportunity to choose the task they would like to work on – in our case, filling out our survey questionnaire. However, we see this as a common threat in all survey studies where questionnaire filling is voluntary. The recommendations we presented assume that high pleasurable emotions impact handling of RCs positively and low pleasurable emotions impact handling of RCs negatively in general. However, certain emotions such as *anger* might sometimes make software teams more productive [15].

JAWS was used preliminary as a mechanism to encourage sharing of emotions, since our previous study [11] suggests this can be challenging for some people. Using JAWS also enabled consistency in terminology, making it easier to identify emotions in the open text responses. Using JAWS may be seen as influence of literature on open coding. However, this is not the case because STGT for data analysis was not used for Emotions identification. We used the JAWS classification for that. Open coding led to the emergence of other categories, as described in Section 3.2.4.

STGT Evaluation: STGT outlines criteria for evaluating the application of the method. As this paper does not propose a mature theory, we evaluated our emerging model (Fig. 9) against the criteria for emerging theories: credibility and rigor. **Credibility:** We have provided details in Section 3.2.3 on how participants were recruited (social media and AMT), the applied an initial sampling method (random sampling followed by purposive sampling), how iterative and interleaved data collection and analysis occurred, and that memos written and used (diagrams). **Rigour:** In Section 3.2.4, we have provided examples of our basic coding (how raw data was analysed to produce codes, subcategories, and categories), embedded sanitised evidence (quotes from the participants throughout out the paper). Similar to method application evaluation, we evaluated our outcome as well against the criteria – originality, relevance, and density, provided in STGT. **Originality:** We have: (a) discussed our findings in relation to existing literature in Section 2 and have made evident how our work fills the gap in the literature, and (b) discussed the potential biases and their impact in Section 7. **Relevance:** According to STGT, relevance is achieved through feedback from the participants, other practitioners, and independent reviewers that serve to validate the findings. We shared the findings with two senior industry practitioners, who were not part of our study and who deal with RCs regularly, and we were able to improve the manuscript based on the feedback received from the anonymous reviewers. **Density:** Density, refers to the depth/richness of the categories and is achieved in our paper as our categories are rich with multiple underlying concepts and evidence from underlying raw data (quotes) given in Section 4.

8 Conclusion

In this paper, we present emotional responses to requirements changes, and *emotion dynamics*: how emotions of software practitioners fluctuate over time in their project when they are handling requirements changes. We found specific stages in the requirements change handling life cycle where practitioners emotions are triggered. We found that these emotions are not only tied to the requirements change stages, but also to the *stimuli* of requirements change, such as practitioner, team, manager, and customer. In addition, we discovered that regulation of emotions is possible through *time related aspects* such as by not introducing last minute RCs, and making RC fit project timeline. We conclude that practitioners are not always pleased with requirements changes, and that positively welcoming changing requirements in late development is largely impractical, often violating the emotional well–being of the software practitioners. We propose a dual-purpose emotion-centric decision guide for the carriers of the requirements changes (customers and internal stakeholders who liaise with customers such as managers) and practitioners to decide when to introduce/accept a requirements change to/by the team. We also provide recommendations for practitioners to follow and directions for researchers to explore this area further in the future.

Acknowledgments

This work is supported by a Monash Faculty of IT scholarship. Grundy is supported by ARC Laureate Fellowship FL190100035. Also, our sincere gratitude goes to Dr William Bingley for providing his invaluable feedback on this work, everyone who read the pre–print and provided feedback, anonymous reviewers who reviewed this manuscript, and all the participants who took part in this study.

References

[1] K. Madampe, R. Hoda, and J. Grundy, "A Faceted Taxonomy of Requirements Changes in Agile Contexts," *IEEE Transactions on Software Engineering*, 2021.

[2] S. McGee and D. Greer, "A software requirements change source taxonomy," in *4th International Conference on Software Engineering Advances, ICSEA 2009, Includes SEDES 2009: Simposio para Estudantes de Doutoramento em Engenharia de Software*, 2009.

[3] K. Beck, M. Beedle, A. V. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "Manifesto for Agile Software Development," 2001.

[4] K. R. Scherer, "Toward a dynamic theory of emotion," *Geneva Studies in Emotion*, 1987.

[5] P. T. Van Katwyk, S. Fox, P. E. Spector, and E. K. Kelloway, "Using the Job-Related Affective Well-Being Scale (JAWS) to investigate affective responses to work stressors.," *Journal of occupational health psychology*, vol. 5, no. 2, pp. 219–230, 2000.

[6] G. R. VandenBos, *APA Dictionary of Psychology*. American Psychological Association, 2007.

[7] R. Colomo-Palacios, A. Hernández-López, . García-Crespo, and P. Soto-Acosta, "A study of emotions in requirements engineering," in *Communications in Computer and Information Science*, 2010.

[8] D. Graziotin, X. Wang, and P. Abrahamsson, "Do feelings matter? On the correlation of affects and the self-assessed productivity in software engineering," *Journal of Software: Evolution and Process*, 2015.

[9] A. Kolakowska, A. Landowska, M. Szwoch, W. Szwoch, and M. R. Wrobel, "Emotion recognition and its application in software engineering," in *2013 6th International Conference on Human System Interactions, HSI 2013*, 2013.

[10] S. C. Müller and T. Fritz, "Stuck and frustrated or in flow and happy: Sensing developers' emotions and progress," in *Proceedings - International Conference on Software Engineering*, 2015.

[11] K. Madampe, R. Hoda, and P. Singh, "Towards Understanding Emotional Response to Requirements Changes in Agile Teams," in *IEEE/ACM 42nd International Conference on Software Engineering New Ideas and Emerging Results (ICSE-NIER20)*, (Seoul, Republic of Korea), p. 4, ACM, New York, NY, USA, 2020.

[12] A. Fountaine and B. Sharif, "Emotional awareness in software development: Theory and measurement," in *Proceedings - 2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering, SEmotion 2017*, pp. 28–31, Institute of Electrical and Electronics Engineers Inc., 6 2017.

[13] D. Girardi, F. Lanubile, N. Novielli, and A. Serebrenik, "Emotions and Perceived Productivity of Software Developers at the Workplace," *IEEE Transactions on Software Engineering*, pp. 1–1, 6 2021.

[14] M. R. Wrobel, "Towards the participant observation of emotions in software development teams," in *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016*, 2016.

[15] M. R. Wrobel, "Emotions in the software development process," in *2013 6th International Conference on Human System Interactions, HSI 2013*, 2013.

[16] B. Crawford, R. Soto, C. L. de la Barra, K. Crawford, and E. Olguín, "The Influence of Emotions on Productivity in Software Engineering," in *Communications in Computer and Information Science*, 2014.

[17] D. Graziotin, X. Wang, and P. Abrahamsson, "Happy software developers solve problems better: Psychological measurements in empirical software engineering," *PeerJ*, 2014.

[18] D. Graziotin, X. Wang, and P. Abrahamsson, "Software developers, moods, emotions, and performance," *IEEE Software*, 2014.

[19] R. Colomo-Palacios, C. Casado-Lumbreras, P. Soto-Acosta, and . García-Crespo, "Using the affect grid to measure emotions in software requirements engineering," *Journal of Universal Computer Science*, vol. 17, no. 9, pp. 1281–1298, 2011.

[20] R. Hoda, "Socio-Technical Grounded Theory for Software Engineering," *IEEE Transactions on Software Engineering*, pp. 1–1, 8 2021.

[21] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Computers in Human Behavior*, vol. 51, pp. 915–929, 10 2015.

[22] E. Bjarnason, K. Wnuk, and B. Regnell, "A case study on benefits and side-effects of agile practices in large-scale requirements engineering," in *Proceedings of the 1st Agile Requirements Engineering Workshop, AREW'11 - In Conjunction with ECOOP'11*, 2011.

[23] M. Käpyaho and M. Kauppinen, "Agile requirements engineering with prototyping: A case study," *2015 IEEE 23rd International Requirements Engineering Conference, RE 2015 - Proceedings*, pp. 334–343, 11 2015.

[24] Z. Bakalova, M. Daneva, A. Herrmann, and R. Wieringa, "Agile Requirements Prioritization: What Happens in Practice and What Is Described in Literature," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6606 LNCS, pp. 181–195, 2011.

[25] N. Baruah, "Requirement management in agile software environment," in *Procedia Computer Science*, vol. 62, pp. 81–83, Elsevier B.V., 2015.

[26] F. G. De Oliveira Neto, J. Horkoff, E. Knauss, R. Kasauli, and G. Liebel, "Challenges of aligning requirements engineering and system testing in large-scale agile: A multiple case study," *Proceedings - 2017 IEEE 25th International Requirements Engineering Conference Workshops, REW 2017*, pp. 315–322, 9 2017.

[27] L. Jun, W. Qiuzhen, and G. Lin, "Application of agile requirement engineering in modest-sized information systems development," *Proceedings - 2010 2nd WRI World Congress on Software Engineering, WCSE 2010*, vol. 2, pp. 207–210, 2010.

[28] D. Albuquerque, E. Guimaraes, M. Perkusich, A. Costa, E. Dantas, F. Ramos, and H. Almeida, "Defining agile requirements change management: A mapping study," in *Proceedings of the ACM Symposium on Applied Computing*, pp. 1421–1424, Association for Computing Machinery, 3 2020.

[29] B. Ramesh, L. Cao, and R. Baskerville, "Agile requirements engineering practices and challenges: an empirical study," *Information Systems Journal*, vol. 20, pp. 449–480, 9 2010.

[30] S. Wagner, D. Méndez Fernández, M. Kalinowski, and M. Felderer, "Agile Requirements Engineering in Practice: Status Quo and Critical Problems," vol. 21, no. 1, 2018.

[31] A. R. Asghar, A. Tabassum, S. N. Bhatti, and A. M. Jadi, "Impact and challenges of requirements elicitation & prioritization in quality to agile process: Scrum as a case scenario," *International Conference on Communication Technologies, ComTech 2017*, pp. 50–55, 10 2017.

[32] B. Crawford, R. Soto, C. L. de la Barra, K. Crawford, and E. Olguín, "The Influence of Emotions on Productivity in Software Engineering," in *Communications in Computer and Information Science*, vol. 434 PART I, pp. 307–310, Springer Verlag, 2014.

[33] M. Kuutila, M. V. Mäntylä, M. Claes, M. Elovainio, and B. Adams, "Using experience sampling to link software repositories with emotions and work well-being," in *International Symposium on Empirical Software Engineering and Measurement*, 2018.

[34] M. K. Curumsing, *Emotion-oriented requirements engineering*. PhD thesis, 2017.

[35] D. Watson, L. A. Clark, and A. Tellegen, "Development and validation of brief measures of positive and negative affect: The PANAS scales.," *Journal of Personality and Social Psychology*, vol. 54, no. 6, 1988.

[36] E. Diener, D. Wirtz, W. Tov, C. Kim-Prieto, D. w. Choi, S. Oishi, and R. Biswas-Diener, "New well-being measures: Short scales to assess flourishing and positive and negative feelings," *Social Indicators Research*, vol. 97, no. 2, 2010.

[37] C. D. Fisher, "Emotions at work: What do people feel and how should we measure it? Part of the Human Resources Management Commons," tech. rep., 1997.

[38] University of York. NHS Centre for Reviews & Dissemination, *Systematic Reviews: CRD's guidance for undertaking reviews in health care*. CRD, University of York, 2009.

[39] B. A. Basińska, E. Gruszczyńska, and W. B. Schaufeli, "Psychometric properties of the polish version of the Job-related Affective Well-being Scale," *International journal of occupational medicine and environmental health*, vol. 27, no. 6, pp. 993–1004, 2014.

[40] Z. M. Ünal, "Influence of leaders humor styles on the employees job related affective well-being," *International Journal of Academic Research in Accounting, Finance and Management Sciences*, vol. 4, no. 1, pp. 201–211, 2014.

[41] L. Morrissy, P. Boman, and A. Mergler, "Nursing a case of the blues: An examination of the role of depression in predicting job-related affective well-being in nurses," *Issues in mental health nursing*, vol. 34, no. 3, pp. 158–168, 2013.

[42] "Job-related Affective Well-being Scale JAWS - Paul Spector."

[43] B. A. Kitchenham and S. L. Pfleeger, "Personal opinion surveys," in *Guide to Advanced Empirical Software Engineering*, pp. 63–92, Springer London, 2008.

[44] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. El Emam, and J. Rosenberg, "Preliminary guidelines for empirical research in software engineering," *IEEE Transactions on Software Engineering*, vol. 28, no. 8, pp. 721–734, 2002.

[45] T. Punter, M. Ciolkowski, B. Freimut, and I. John, "Conducting on-line surveys in software engineering," in *Proceedings - 2003 International Symposium on Empirical Software Engineering, ISESE 2003*, pp. 80–88, Institute of Electrical and Electronics Engineers Inc., 2003.

[46] E. Smith, R. Loftin, E. Murphy-Hill, C. Bird, and T. Zimmermann, "Improving developer participation rates in surveys," in *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2013 - Proceedings*, pp. 89–92, 2013.

[47] SlashData Ltd., "The Global Developer Population Report 2019," tech. rep., 2019.

[48] "14th State of Agile Report — State of Agile," tech. rep., https://stateofagile.com/, 2020.

[49] A. Strauss and J. Corbin, *Basics of qualitative research techniques.* 1998.

[50] R. Colomo-Palacios, T. Samuelsen, and C. Casado-Lumbreras, "Emotions in software practice: Presentation vs. coding," in *Pro-*

ceedings - 2019 IEEE/ACM 4th International Workshop on Emotion Awareness in Software Engineering, SEmotion 2019, pp. 23–28, Institute of Electrical and Electronics Engineers Inc., 5 2019.
[51] M.-A. A. El-Migid, D. Cai, T. Niven, J. Vo, K. Madampe, J. Grundy, and R. Hoda, "Emotimonitor: A Trello Power-Up to Capture Emotions of Agile Teams," 5 2021.
[52] P. Bregman, "Productive Conversations Take Real Empathy,"
[53] A. Cummaudo, U. M. Graetsch, M. K. Curumsing, R. Vasa, S. Barnett, and J. Grundy, "Emotions in Computer Vision Service Q amp;A," in *2021 IEEE/ACM Sixth International Workshop on Emotion Awareness in Software Engineering (SEmotion)*, pp. 13–18, 2021.
[54] Y. Shastri, R. Hoda, and R. Amor, "The role of the project manager in agile software development projects," *Journal of Systems and Software*, vol. 173, p. 110871, 3 2021.
[55] P. Ralph, S. Baltes, G. Adisaputri, R. Torkar, V. Kovalenko, M. Kalinowski, N. Novielli, S. Yoo, X. Devroey, X. Tan, M. Zhou, B. Turhan, R. Hoda, H. Hata, G. Robles, A. M. Fard, and R. Alkadhi, "Pandemic Programming: How COVID-19 affects software developers and how their organizations can help," 5 2020.
[56] P. Koval, E. A. Butler, T. Hollenstein, D. Lanteigne, and P. Kuppens, "Emotion regulation and the temporal dynamics of emotions: Effects of cognitive reappraisal and expressive suppression on emotional inertia," *Cognition and Emotion*, vol. 29, pp. 831–851, 7 2015.
[57] J. J. Gross and R. Thompson, "Emotion regulation: Conceptual foundations. In J.J. Gross (Ed.), Handbook of emotion regulation. New York: Guilford Press.," in *Handbook of Emotion Regulation*, vol. 3, 2007.
[58] S. Koole, "The psychology of emotion regulation: An integrative review," 2009.
[59] I. Cristofori and H. S. Levin, "Traumatic brain injury and cognition," in *Handbook of Clinical Neurology*, vol. 128, pp. 579–611, Elsevier B.V., 1 2015.
[60] Project Management Institute, *PMBOK GUIDE*. 2008.

APPENDIX A
DEFINITIONS OF KEY TERMS USED

Table 7 presents the key terms that we used in this paper and their definitions. The cited definitions are directly from their sources and not paraphrased. We use the term "individual cognition" instead of the original term "cognition" to avoid the confusions with "social cognition". Similarly, "team dynamics", and "team cohesion" are used instead of "group dynamics" and "group cohesion".

TABLE 7: Definition of Key Terms Used

Term	Definition
Individual cognition	All forms of knowing and awareness, such as perceiving, conceiving, remembering, reasoning, judging, imagining, and problem solving [6]
Cognitive skills	The skills involved in performing the tasks associated with perception, learning, memory, understanding, awareness, reasoning, judgement, intuition, and language [6]
Conation	The proactive (as opposed to habitual) part of motivation that connects knowledge, affect, drives, desires, and instincts to behavior [6]
Emotion	A sequence of interrelated, synchronised changes in the states of all the five organismic subsystems (information processing, support, executive, action, and monitoring) in response to the evaluation of an external or internal stimulus event as relevant to central concerns of the organism [4]
Emotion dynamics	The patterns and regularities with which emotions fluctuate over time [56]
Emotional intelligence	Type of intelligence that involves the ability to process emotional information and use it in reasoning and other cognitive activities [6]
Emotion regulation	Any process that decreases, maintains, or increases emotional intensity over time, thereby modifying the spontaneous flow of emotions [56], [57], [58]
Emotional response	An emotional reaction, such as happiness, fear, or sadness, to give a stimulus [6]
Empathy	Understanding a person from his or her frame of reference rather than ones own, or vicariously experiencing that persons feelings, perceptions, and thoughts [6]
Motivation	A persons willingness to exert physical or mental effort in pursuit of a goal or outcome [6]
Perception	The process or result of becoming aware of objects, relationships, and events by means of the senses, which includes such activities as recognising, observing, and discriminating [6]
Self-efficacy	An individuals subjective perception of his or her capability to perform in a given setting or to attain desired results [6]
Social cognition	Cognition in which people perceive, think about, interpret, categorise, and judge their own social behaviors and those of others [6]
Stimulus	Any agent, event, or situationinternal or externalthat elicits a response from an organism [6]
Sustained attention	The ability to sustain attention over time in specific goal-directed behaviors [59]
Team cohesion	The unity or solidarity of a group, including the integration of the group for both social and task-related purposes [6]
Team dynamics	The processes, operations, and changes that occur within social groups, which affect patterns of affiliation, communication, conflict, conformity, decision making, influence, leadership, norm formation, and power [6]
Dimensions	The range along which general properties of a category vary, giving specification to a category and variation to a theory [49]
Properties	Characteristics of a category, the delineation of which defines and gives it meaning [49]
Project	A temporary endeavor undertaken to create a unique project service or result [60]
Project life cycle	The execution flow across time of any software project that allows the software team to meet the project goal
Requirements Change (RC)	Additions/modifications/deletions of functional /non-functional requirements in a software project
RC handling life cycle	The execution flow across time of an RC that allows the software team to deliver it. RC handling life cycle begins when the RC is introduced to the team and lies within the project life cycle
Team contexts	The socio-technical environment where software practitioners work collectively as a team to achieve a common goal
time Related Aspects	Time related concerns that have impacts on project and RC handling life cycles

APPENDIX B
GUIDE: RECRUITING PARTICIPANTS VIA AMAZON MECHANICAL TURK

Please do the needed at Qualtrics end before publishing the survey at the AMT end.

At Qualtrics End: A unique ID is needed to be created at Qualtrics end so that the workers can fill that at AMT end.

Follow the steps given at https://blog.mturk.com/getting-great-survey-results-from-mturk-and-qualtrics-be1704ff9786

For general information: https://blog.mturk.com/getting-started-with-surveys-on-mturk-e2eea524c73

At AMT End:

1) Sign up as a requester at https://www.mturk.com/
2) Click on **Create** tab (https://requester.mturk.com/create/projects)

3) Click on **New Project** (Top left corner https://requester.mturk.com/create/projects/new)

For surveys,

1) Click on **Survey Link** on left pane and click on **Create Project** (bottom right corner)
2) Fill the properties
   We paid 6.40 USD per participant (Survey had 4 open-ended questions and 10 closed-ended questions. Average time for completion is 15–20 minutes). This is entirely up to you. We suggest you pay a fair amount to the participants for their time spent.
   Keep a fair allotted time for worker (We kept 30 minutes)
   In **Specify additional qualifications**, select the geographical locations. and other qualifications such as "Employment Industry - Software IT Services" and "Job Function - Information Technology".
   We changed the number of workers and qualifications in different batches to balance the demographics of the survey respondents.
3) After filling the properties, click on **Design layout** (bottom right corner)
4) Edit the design layout (edit the text)
5) Insert the link of the survey in the **Survey Link** box
6) Click on Preview
7) Credit card information will be asked to fill in if you are doing this for the first time. AMT charges 20% more as their service fee

APPENDIX C
CATEGORIES

Category	Definition	Sub Category	Concept	Definition	Code	Definition
			Stability	The requirement is changing/not changing (stable/unstable). This could be resulted by the changing extent.	Unstable	The RC is changing
					After design and implemention	The RC introducted <as in Code>
					During testing	The RC introducted <as in Code>
					After development is fully completed	The RC introducted <as in Code>
					After development is significantly completed	The RC introducted <as in Code>
					Middle of the project	The RC introducted <as in Code>
					During implementation	The RC introducted <as in Code>
					After peers reviewing the completed requirements	The RC introducted <as in Code>
			Point of introduction	The point in the project where RC is introduced (when)	After demo	The RC introducted <as in Code>
					On other requirements	The RC impacts <as in Code>
			Impact	The impact the RC makes on the other components of the project	On scope	The RC impacts <as in Code>
			Definition	How understable is the RC	Initial learning	The definition of the RC is learnt for the time
			Type	The type of RC	Improvements	The RC is an improvement
			Changing extent	How often does the RC change (frequent/not frequent)	Frequent	The RC changes frequently
			Code status	The code status of the RC - functioning/not functioning	Functioning	The code status of RC is <as in Code>
		RC	Challenging nature	The challenging nature of the RC - easy/difficult to implement	Easy to implement	The RC is <as in Code>
					Self motivation [+]	The practitioner is self motivated
			Individual conation	The aspects related to individual conation (See Table 7 for definition)	Self motivation [-]	The practitioner lacks self motivation
					Self efficacy [+]	The practitioner has self efficacy (see Table 7 for definition)
					Personality perception [self]	The practitioner perceives the own personality
			Individual cognition	The aspects related to cognition (See Table 7 for definition)	Self cognitive skills [+]	The practitioner has self cognitive skills (see Table 7 for definition)
		Practitioner	Social cognition	The aspects related to social cognition (See Table 7 for definition)	Social cognition [+]	The practitioner has social cognition (See Table 7 for definition)
					Team cohesion [-]	The team lacks cohesion (See Table 7 for definition)
					Team skills [+]	The team has team skills (See Table 7 for definition)
			Team dynamics	The aspects related to team dynamics (See Table 7 for definition)	Team skills [-]	The team lacks team skills (See Table 7 for definition)
		Team	Source of RC	Originates RC		
			Emotional intelligence	The aspects related to emotional intelligence (See Table 7 for definition)	Emotional intelligence [-]	The manager lacks emotional intelligence
			Source of RC	Originates RC		
		Manager	Carrier of RC	Carries RC		
Stimuli	What triggers the emotions	Customer	Emotional intelligence	The aspects related to emotional intelligence (See Table 7 for definition)	Emotional intelligence [-]	The customer lacks emotional intelligence
		Receiving				
			Beginning	Coding of RC starts		
			Writing code	While writing the code of the RC		
			Coding completed	Coding of the RC is completed		
		Developing	Troubleshooting	Troubleshooting		
		Testing				
	When does the stimuli trigger the emotions in the		Almost completed	RC implementation is almost completed and closer to delivery		
RC Stages	RC handling life cycle	Delivering	Released/delivered	RC is released/delivered to the production environment		
		Last minute RCs				
		Not enough time to work on RC				
		RC fits the project timeline				
Time related aspects	Time related aspects	Being able to meet RC deadline				

Fig. 10: Categorisation