Using Motivational Models to Promote Emotional Goals Among Software Engineering Students

1st Leon Sterling
Centre for Design Innovation
Swinburne University of Technology
Melbourne, Australia

2nd Eduardo Araujo Oliveira
School of Computing and Information Systems
The University of Melbourne
Melbourne, Australia

***Abstract*—There has been increasing awareness in recent years of the importance of considering broader, social and personal goals in software engineering. One important aspect is the need to engender certain feelings in users when they interact with software. Such emotional goals are relatively new to software development. While there are established methods for teaching requirements elicitation for standard goals and evaluating whether they have been met, there is much less work on evaluating whether emotional goals have been met through software. For the past five years, emotional goals have been explicitly included while teaching software requirements with students at The University of Melbourne and Swinburne University of Technology, both in dedicated subjects and in project subjects. Students have been strongly encouraged to incorporate emotional goals in the design of projects' requirements through the use of motivational models. In this paper we discuss and reflect on our experience teaching motivational modelling that is aimed at supporting students in the creation and evaluation of emotional goals in the software requirements phase.**

I. Introduction

The rapid rise in personal use of smart devices over the last fifteen years has changed software development in fundamental ways. Previously, software was primarily built for individual organisations and used to solve business problems. Now increased focus is given to the needs and experiences of end-users, allowing technology to support individuals and communities in their everyday lives. In the consumer space, people rapidly make discretionary decisions to use or reject apps. People can quickly reject new technology if it doesn't appeal to their emotional needs. There is little room for error concerning the user experience.

Lopez-Lorca and colleagues [1] suggest that, in order to develop systems that fulfil users' needs, it is necessary to explore users' emotions (i.e. the feelings that characterise their state of mind) during requirements engineering phase. Emotional goals need to be treated differently from traditional quality goals as they are user-centred, not system-based.

This paper discusses findings on the use of Motivational Models (MM) by 128 students that have been taught motivational modelling and effectively built over 20 motivational models for industry partners between 2019 and 2022. Motivational models are diagrams that present goals of a system in a hierarchical way and at a high level of abstraction. These models support the development of shared project understanding between diverse stakeholders in a non-technical, user-centred manner. Our findings suggest that the use of MM helped students to identify and design emotional goals, to better communicate and validate emotional goals with industry partners, and to identify and write better user stories in requirements engineering phase.

These findings have important implications for the identification and design of users' emotions in software development process. Combined with personas and user stories artifacts, motivational models can support students to validate emotional goals beyond demonstration of prototypes (and prior to development phase) and the potential to uncover hidden requirements that may otherwise be ignored. As these models are non-technical, they can be remarkably helpful to improve communication between different stakeholders. Motivational models keep the focus of requirements engineering phase to the problem that need to be addressed through the development of a software, not in the solution.

II. Background

As software has become more embedded in society, a broader view of software development has been advocated [2], [3]. Software systems should be designed with an understanding of the broad, social environment in which they are used. In this background section, we discuss the socio-technical context of modern software systems, the need for multidisciplinary engagement in software development, and the need for addressing emotions in requirements.

A. Socio-Technical Systems and Socially-Oriented Requirements

Software has been incredibly successful. Software underpins most areas of society. Early software applications were used by government, military and business. It was a challenge to understand how best to develop reliable software that met business needs and was easy to use. The disciplines of software engineering and requirements engineering emerged to improve practice.

Early software engineering focussed on getting functional requirements correct. It was soon realized that concentrating on functionality was not enough. As argued by many, including [4]–[7], it was important to give attention to the non-functional requirements. Whether non-functional requirements were met had a huge impact on whether a particular software system

would be successful. Note we prefer to use the term quality requirements rather than non-functional requirements as they represent desired qualities [8] of the system.

Awareness has increased that software engineering is a social activity [3]. It is not obvious how to take the 'social' of socio-technical systems seriously, which has led to a consideration of socially oriented requirements [2], [9] as a topic to be considered in its own right. An important aspect of social oriented requirements are emotions that people may or may not feel when interacting with software. If it is accepted that socially oriented requirements are important, it must be asked whether current requirements elicitation methods suffice to elicit them. It is debatable whether people will articulate more abstract requirements with standard elicitation methods. Asking people what specific function a piece of software should perform is different to asking people how they want to feel when interacting with software. People need to feel comfortable when asking and talking about feelings.

B. Emotional Requirements for Consumer-Oriented Systems

A typical first step in discussions of design thinking is 'empathise with your users' [10]. A key way to empathise is to appreciate the emotions a user may feel when interacting with a system. Such an empathetic step is not typically discussed in software engineering where expectations are that the user will articulate their needs in functional terms. The role of the software engineer is to translate needs into program requirements and designs. It is important in our experience to represent the emotions that the system is intended to engender in the stakeholders as they interact with the software components of the socio-technical system, and see if any new functional requirements are needed.

Many different types of emotions are important to consider when designing technology. For example, basic emotions such as joy [11], [12] can be a core goal of many software systems. A major motivation of computer games or software systems that utilise gamification is to create a fun and delightful experience. Other emotional frameworks contain different types of emotions that are more reflective, such as shame and resentment [13]–[15]. Another type of emotional requirement involves idiomatic expressions, phrases or words that reflect the way a user or stakeholder would describe how they wish to feel. Examples can be found in healthcare technologies that allow older patients to continue living in their own homes and feel independent as per the earlier anecdote. Capturing such expressions is important in requirements engineering if technology is well received. We need to be able to capture information about emotions in ways that stakeholders can naturally and comfortably express.

We note that experts disagree on the theoretical underpinnings of how emotions form. Some believe in a hierarchy of emotions, building from basic emotions as fear, anger or joy. Others believe that emotions are constructed concepts developed through life experience [16]. Regardless, having emotions as explicit goals for the whole software team is of value, regardless of how they are exactly formed in the mind.

Exactly defining emotions is not necessary in our experience. Emotions are not clear-cut and hence emotional requirements are ambiguous. The ambiguity may lead them to be sidelined. We believe that emotions are important to address and need to be integrated with the elicitation of functional and other quality requirements.

III. Teaching Motivational Modelling

Over two thousand students from undergrad and master's courses at University of Melbourne and at Swinburne University of Technology have been taught motivational modelling between 2019 and 2022. There are two main stages involved in teaching motivational modelling to higher education students. The first stage is a controlled brainstorming session using the do/be/feel method [17]. The second stage is transforming the lists produced by the do/be/feel method into a motivational model - a single hierarchical diagram encapsulating the project on a single page. As the project progresses and understanding increases, the model should be appropriately modified. The models need to be explained to the client who is expected to 'sign off' on the model. The model is also assessed by the project supervisor. In our teaching, modifications to the initial model came about through feedback from the teaching staff and clients either through formal assessment or direct feedback on the clarity of the model.

In assessing how effectively students have been able to use emotional goals, we need to explain how the material has been taught. The amount of time taken to teach a topic and the support material given to students will affect how easy the material has been to learn. Because motivational modelling is often only a small part of a project subject, usually there was a single week of content available. After the initial teaching, it was discovered it was useful to split the content into two halves - one to discuss do/be/feel method to elicit project goals, and the other to transform the lists produced in the elicitation session into a single page diagram.

Having two sessions was more effective than having a single session. The effectiveness was adjudged on two criteria - the quality of the models produced and the need to explain the material. Note that an 18 page handout explaining motivational modelling and giving an example of its use was provided to the students to read at their leisure. In 2018, a team of year-long software engineering students from The University of Melbourne built a Motivational Model Editor Tool (MM Tool). After the development of MM Tool, students were provided access to it for drawing the models. Overall, students received good support for their learning of motivational modelling.

A. Do/be/feel process

It is natural to run an inclusive brainstorming session with technical and non-technical stakeholders. Both are involved in the decision making process of what should appear in the model. Students are expected to run a do/be/feel session with their clients and come up with a list of goals. Emotional goals were expected to be included in the list. The goals, after validated, are then transformed into a list of user stories.

In recent years, students were presented with a method of ensuring consistency between motivational models and user stories [18].

As a running example we demonstrate the use of motivational modelling in an IT project subject. A team of five undergraduate students developed an application for a client who was a Masters student at the same university. The project subject had a theme of developing item registers. For this group, the master's student suggested that the undergraduate team build a Web app for entering and reading reviews of restaurants.

The first stage of motivational modelling develops four lists. The first list is of the stakeholders. The second list is of the functions the software should provide. The third list is of the qualities the system should have. The fourth list is of the emotions that stakeholders should feel when interacting with the system. The lists are respectively the who list, do list, be list, and feel list.

For the running (simple) example, the four lists are:

- Who list: User, Guests
- Do list: Restaurant Review Web App, View Reviews, Post reviews, Edit Reviews, Search
- Be List: Reliable, Enriched, Easy-to-use, Efficient, Responsive
- Feel list: Appealing, Engaging, Pleasant

We acknowledge the ambiguity in the terms. Two people are unlikely to agree exactly on what is appealing and engaging to users, for example. However ambiguity allowing conversation and clarification throughout a project can be viewed as a strength not a weakness as argued in [19]. The findings discuss whether the terms in the lists were well understood, especially the emotional goals.

User experience does need to take emotional elements into consideration. However it is our view that emotions need to be considered by the whole team. A positive emotional experience cannot always be easily added if it has not been included in the design. From requirements methods, user stories can address the emotional needs as can personas, but they will not be considered unless they are explicitly goals of the project. Everyone needs to be on the same page and part of ensuring that the emotional goals are met.

Discussing emotions is new for most students and industry partners. Including emotional goals is typically not covered in software engineering subjects. During the lecture on motivational modelling, there was conversation about differences between emotions and other qualities. Students can struggle to appreciate the difference. We expand this point in the Findings section.

B. Building the model

The lectures present a seven stage process for converting the four lists produced in a do/be/feel process into the motivational model [18]. The handout to the students contains a worked example. The first and last stages of the process involve checking with the client. There is also a clustering step of the functional requirements. The four remaining steps are building the 'do' hierarchy, then adding 'who' elements, adding 'be' elements and 'feel' elements to the appropriate level of the hierarchy. The students are given access to MM Tool to help with drawing the diagram. Some students opt to use the tool, others use tools such as draw.io instead.

We continue with the Web app example for restaurant reviews. A simplified motivational model is given in Figure 1. It was built by the team of five undergraduate students. The model could be improved. It is chosen here for ease of explanation, and to illustrate the findings in the next section.

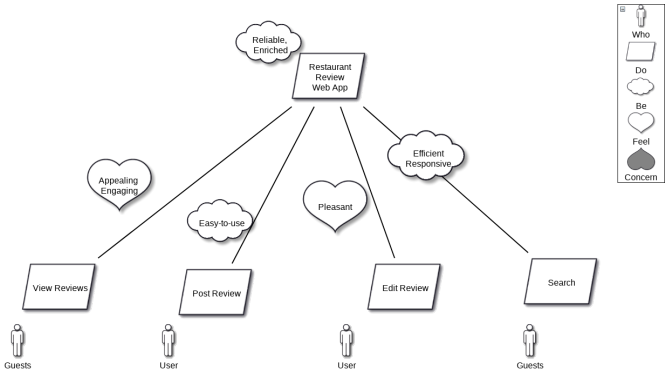

Fig. 1. A simple motivational model developed by students

As can be seen, the model contains several shapes, specifically parallelograms, hearts, people, and clouds as per Figure 1. There is value in having a model on a single page. If a model is too complicated, it does not work as well. In our experience it is better to segment a model when there are too many nodes. In our given example, there are four main functional goals of the Restaurant Review Web App, expressed in the four parallelograms under the central parallelogram in the top of Figure 1. They are to 'View Reviews,' 'Post Review,' 'Edit Review,' and 'Search' The stakeholders are users and guests indicated in the people at the bottom of the figure. The desired qualities of the Restaurant Review Web App are expressed in the three clouds in the figure. Overall the Web App needs to be reliable and enriched, posting a review must be easy-to-use or easy to do, and the search must be efficient and responsive. The desired emotions of users and guests interacting with the Restaurant Review Web App are expressed in the hearts. The users should feel that the app is appealing and engaging, and the editing should feel pleasant. It is beyond the scope of the paper to critique the model, but it shows the students have noted emotional requirements.

IV. METHODS

Participants

Participants were informed about this research investigation via e-mail or announcements posted on Canvas Learning Management System (LMS) and provided informed consent

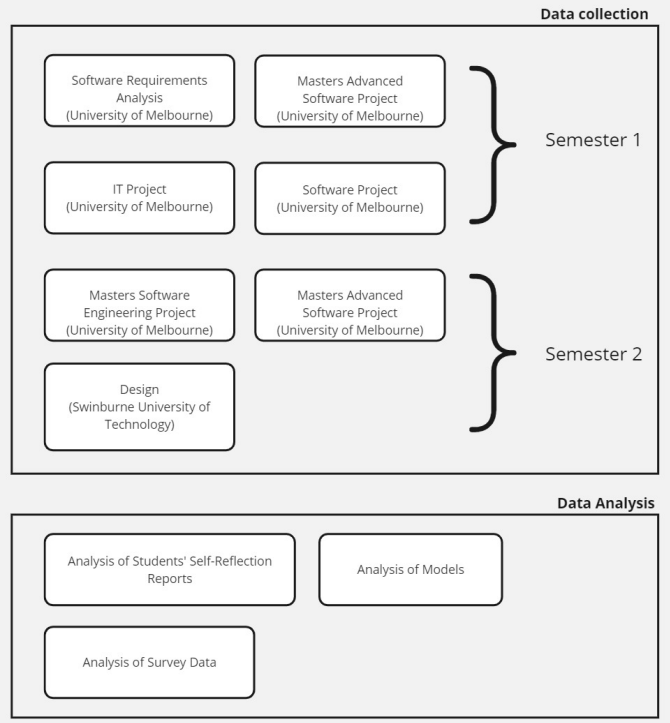

Fig. 2. Research Procedure

(Ethics approval #24272). Even though over two thousand students from The University of Melbourne and Swinburne University of Technology have used motivational models between 2019 and 2022, our sample in this study included a total of 128 participant students (those who signed our consent form and agreed to participate in this research). The participants in this study generated over 20 models during those years.

Data Collection: Models developed by participant students were submitted to Canvas LMS or Confluence, and downloaded by subject coordinators at the end of the teaching semester. All participant students were undertaking one of the five subjects from the master of software engineering and master of information and technology at University of Melbourne or a design subject from the bachelor of design at Swinburne University of Technology (Figure 2).

For three of the subjects, students were required to build a motivational model. The other three subjects students were encouraged to use motivational modelling to express requirements, but motivational modelling was not mandated. Most students chose to use it.

Data Analysis: Our evaluation of the effectiveness of teaching students the use of emotional goals through motivational modelling came through two separate analyses: (i) analysis of models, (ii) analysis of students' self-reflection reports.

1) Analysis of Models: An assessment rubric with criteria below was used to guide our analysis of models:

- Granularity: The goal model has an adequate level of granularity that helps understand and explain the context of the project, its main problem and motivation. It shows well the goals of the system to be developed to a level of detail similar to user stories. The level of granularity allows for a correct association of quality and emotional goals to particular functional goals.
- Hierarchy: structure The hierarchy of functional goals makes perfectly sense for somebody not familiar with the problem domain and it can be followed to explain and motivate the problem to an external party.
- Use of elements: The notation has been followed and the elements have been used correctly.
- Complete: The motivational model completely covers the information elicited during the session and other interactions with industry partner/stakeholders involved in this process.
- Do/be/feel list and Do/be/feel hierarchy: both are presented together with the model.
- Consistency: The motivational model is consistent with the industry partner understanding of the problem and with the Do/be/feel list and Do/be/feel hierarchy.
- Presentation: The motivational model has an elegant and neat layout, including clarity (labels and spelling).

This rubric has three ratings (inadequate, good, excellent) for each criterion and was available to students on Canvas LMS, prior to their assignments. The assessment criteria were explained in details in the lectures that focused on teaching do/be/feel and MM to students. Subject coordinators assessed students' submissions.

2) Analysis of Student's Self-Reflection Reports: Students' self-reflection reports were analysed through qualitative content analysis in which authors categorised and tagged themes on reports.

Students in three of the subjects were required to provide a 500-word subject reflection as part of their assessment. The suggested items for the reflective report included what the students learned, and overall impressions. The students were not asked, nor were they expected to comment on motivational modelling or emotional goals.

All references to motivational modelling or emotional goals were recorded. The number of times either of the two terms were mentioned was small, amounting to less that two percent of submissions. Nonetheless some of the quotes were memorable. They are recorded to support the findings listed below.

V. Findings and Discussions

We present findings that emerged from participants in the past five years of teaching motivational modelling and emotion goals to software engineering and design students. The findings section is organised and presented in two main sections: (i) analysis of models, (ii) analysis of students' self-reflection reports.

A. Analysis of Models

Our findings, from reviewing over 20 models, is that this artifact is a teachable skill, as evidenced by all the student projects' deliverables:

- criteria students did mostly well:

  - consistency: all MM were consistent with other requirements artifacts (i.e., personas and user stories). Great use of elements;
  - complete: all MM were validated with industry partners and improved through several iterations. They fully captured the main motivation behind the development of future systems and stakeholders involved in the design of that solution;
  - hierarchy and use of elements: MM could be created in meaningful ways by all participants (easy to use);
- observed challenges:
  - system-based or user-centred: some descriptions in the 'do' list were too technical and described features of systems;
  - emotional goals vs quality goals: it has been hard for students to differentiate between them;
  - clarity: some descriptions used in the motivational model are ambiguous, vague and/or unclear;
  - granularity: some motivational models have too many sub-goals or levels;

Overall, students produced great models together with the industry partners. 27 models were assessed in this study using the same rubric. The average mark for models submitted by participants was 7.13/10, which puts them between good and excellent ratings. Motivational modelling fits in easily with agile development processes. The students are able to link the goals from the motivational model with other software engineering requirements artefacts such as user stories, personas and customer journeys. Models were perceived as lightweight and effective for agile development. Two aspects of being lightweight are being easy to document and to construct.

In this process, we observed students found the evaluation of emotional goals to be fundamentally subjective. Deciding whether an emotional goal has been met was (and still is) challenging for students. Our analysis noted that evaluation of feelings is done for consumer products, but in a different way to standard software projects that require more objective measures. Realistically, it is beyond the scope of a student project to seriously evaluate goals. In our teaching process, we did suggest a guideline of deciding between quality goals and emotional goals by whether the testing criteria would be objective or subjective. Not all students followed that perspective.

Where to place emotions and qualities relative to the functional goals remains a challenge. In any case, we observed a significant increase in the number of emotional goals identified, discussed, and validated during requirements engineering phase when compared to previous years in which students didn't use motivational models to communicate with industry partners. Motivational modelling has potential to improve user-centred design of requirements at early development phases.

B. Analysis of Students' Self-Reflection Reports

Several students provided qualitative written comments in their subjects' self-reflection reports, which provided an opportunity to reflect on whether the use of do/be/feel process and MMeditor tool by the teaching team had a positive impact on students' teaching and learning experience, and identification of emotional goals in requirements engineering phase.

As part of this analysis of students' self-reflection reports, the paper authors performed independent content analysis, categorising, tagging and performing thematic analysis on participants' reports. Initially, ten main themes were identified by authors: (i) learning curve to adopt MM in teaching; (ii) MM is contextual; (iii) usability of MM tool; (iv) emotional goals and overall understanding of a project; (v) MM process; (vi) challenges in the use of MM; (vii) quality and emotional goals; (viii) MM compatibility with agile methodology; (ix) focus on problem instead of solution; (x) communication among stakeholders. Authors organised a two meetings to discuss and agree on identified themes together. The themes were then combined into four main findings:

Finding 1: Motivational modelling supports user-centred and user-friendly processes

Students and teaching staff follow the do/be/feel process with little instruction. This has contributed to motivational modelling becoming standard in teaching how to design user-centred solutions at our universities. The models are mostly well-structured, neatly clustered and organised. Given the models are directly assessed, it is perhaps not surprising. For most of the subjects, requirements elicitation is only a small component. So to succeed at scale, the teaching could not rely on specialist skills. We have developed useful material, including an 18 page handout, and videos on applying the method to build a small model.

A major benefit of running do/be/feel sessions, which has been anecdotally observed, is that everyone is kept engaged and a positive vibe is created. Sessions stress the principle that 'no idea is a bad idea'. The positive engagement of people during the modelling activities facilitates the co-design principles, and increases buy-in after the process. Checking back with participants as models evolve is key to ensuring people accept and adopt the models.

Finding 2: Context is important

Some domains require emotions more importantly than others. That was seen in the models produced and discussed with students. For Figure 1 emotions are not important. For an items register which was an overarching theme for the projects, emotions are of minor importance. For the project from two years earlier to build an ePortfolio system for highlighting masters student work, emotions were more important. It was frequently expressed to say that masters students wanted to feel pride in the display of their work. The experience level of clients had a significant impact on the quality of models. Clients with familiarity with models expressed emotional needs and the models were clearer. The skill level and interests of students also had an impact.

Finding 3: The difference between qualities and emotions is important but not entirely intuitive

We started our investigation of emotional requirements with the expectation that emotions were just another kind of quality requirement. We separated emotional requirements

from quality requirements for several reasons. A primary reason came from experience with eliciting the requirements. Asking what properties a system should have, or how it should be, elicited different responses from asking how users should feel when interacting with the system. So the process to generate requirements needs to differentiate between qualities and emotions.

There is a difference between evaluating quality goals and evaluating emotional goals. Evaluating qualities can be done by testing against objective measures. Evaluation of emotional goals on the other hand is fundamentally subjective. Deciding whether an emotional goal has been met is challenging. However, we note that evaluation of feelings is done for consumer products, but not in a way that is usual in standard software projects which want more objective measures. Realistically it is beyond the scope of a student project to seriously evaluate emotional goals. We do suggest a guideline of deciding between quality goals and emotional goals by whether the testing criteria will be objective or subjective.

Finding 4: Motivational Modelling is lightweight

Motivational modelling fits in easily with agile development processes. Students are able to link the goals from the motivational model with other requirements artifacts such as user stories, personas and customer journeys. Models need to be lightweight to be effective for agile development. Two aspects of being lightweight are being easy to document and being quick to construct. Moreover, its constructive process should create a positive feeling around team members. The combined process of running a do/be/feel workshop and building a motivational model can comfortably be completed in an hour for moderate sized groups with ten or so goals in each of the lists. This contrasts to many brainstorming sessions which take much longer.

Another aspect of the motivational models is that they can be inspected at a time after when they were generated. It is a straightforward to have new participants add goals of all flavours to existing models, and to critique the existing models. This has been done with several of our applications. Indeed there is scope in using motivational models in orientation sessions for people new to a project or organisation. We continue to find new uses for motivational models. This semester it will be used to help guide performance evaluation of supervisors. In general we evolve our processes and much of our experience is anecdotal. This is in fact a limitation of our work from a research perspective, that the observations are not from a controlled study since projects change and guidelines evolve.

VI. Conclusions

We have discussed our experience with teaching motivational modelling to undergraduate software engineering students, with an emphasis of whether they appreciated emotional goals and would be likely to contribute to more human-centred software engineering. Motivational models are more abstract than typical requirements models and explicitly include emotional goals. Creating motivational models is an efficient process which fits comfortably into agile development methods. They have been successfully applied to many projects and have enhanced communication over the lifecycle of the projects. We have presented five findings from our experience. Overall, we strongly advocate using motivational modelling, but we need to be clear on the context of the domain, the supervisors and students. We especially need to be aware of the language background of the students building the models, and not presume that all terms are equally understood.

Acknowledgments

We acknowledge and thank students, clients, and supervisors from the subjects involved in this study.

References

[1] A. A. Lopez-Lorca, T. Miller, S. Pedell, L. Sterling, and M. K. Curumsing, "Modelling emotional requirements," 2014.

[2] G. Baxter and I. Sommerville, "Socio-technical systems: From design methods to systems engineering," *Interacting with computers*, vol. 23, no. 1, pp. 4–17, 2011.

[3] K.-J. Stol and B. Fitzgerald, "The abc of software engineering research," *ACM Transactions on Software Engineering Methodology*, vol. 27, no. 3, 2018.

[4] I. Ramos and D. M. Berry, "Is emotion relevant to requirements engineering?" *Requirements Engineering*, vol. 10, no. 3, pp. 238–242, 2005.

[5] I. Somerville, "Software engineering must be more social," 2010.

[6] S. Thew and A. Sutcliffe, "Investigating the role of'soft issues' in the re process," in *2008 16th IEEE International Requirements Engineering Conference*. IEEE, 2008, pp. 63–66.

[7] E. Yu, "Towards modeling and reasoning support for early-phase requirements engineering," in *RE '97 Proceedings of the 3rd IEEE International Symposium on Requirements Engineering*, 1997.

[8] L. Bass, R. Kazman, and P. Clements, *Software Architecture in Practice*. Pearson Education, 2012.

[9] S. Pedell, T. Miller, F. Vetere, L. Sterling, and S. Howard, "Socially-oriented requirements engineering: Software engineering meets ethnography," in *Perspectives on Culture and Agent-based Simulations*. Springer, 2014, pp. 191–210.

[10] D. H. Mortensen, "Stage 1 in the design thinking process: Empathise with your users." [Online]. Available: https://www.interaction-design.org/literature/article/stage-1-in-the-design-thinking-process-empathise-with-your-users

[11] P. Ekman, "An argument for basic emotions," *Cognition & emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[12] G. L. Clore and A. Ortony, "Appraisal theories: How cognition shapes affect into emotion." 2008.

[13] L. L. Martin and A. Tesser, "Some ruminative thoughts," *Advances in social cognition*, vol. 9, pp. 1–47, 1996.

[14] P. Desmet and P. Hekkert, "Framework of product experience," *International journal of design*, vol. 1, no. 1, 2007.

[15] R. Plutchik, *Emotions and life: Perspectives from psychology, biology, and evolution.* American Psychological Association, 2003.

[16] L. F. Barrett, "The theory of constructed emotion: an active inference account of interoception and categorization," *Social cognitive and affective neuroscience*, vol. 12, no. 1, pp. 1–23, 2017.

[17] E. A. Oliveira and L. Sterling, "Motivational models for validating agile requirements in software engineering subjects," *arXiv preprint arXiv:2306.06834*, 2023.

[18] E. Oliveira, V. Maram, and L. Sterling, "Transitioning from motivational goal models to user stories within user-centred software design." in *RESOSY@ APSEC*, 2021.

[19] J. Paay, S. Pedell, L. Sterling, F. Vetere, and S. Howard, "The benefit of ambiguity in understanding goals in requirements modelling," *International Journal of People-Oriented Programming (IJPOP)*, vol. 1, no. 2, pp. 24–49, 2011.