Tell Me What You See: On a Proposal for a Computer Vision and Sentiment Analysis Tool to Support User Feedback Collection in Software Development

Regis Soares[1], **Leonardo Messa**[1], **Enzo Kleinpaul**[1], **Rafael Parizi**[1]

[1]Instituto Federal Farroupilha
Campus São Borja

{enzo.2021310072, leonardo.2021007357}@aluno.iffar.edu.br
alexandre.2021307059@aluno.iffar.edu.br, rafael.parizi@iffar.edu.br

Abstract. *Agile software development and User-Centered Design emphasize aligning with user needs and prioritizing effective feedback collection. Traditional feedback methods, like interviews and questionnaires, often fail to capture user reactions. To address this, we introduce CV4FeC, a tool that integrates OpenCV for analyzing facial expressions and sentiments, offering a comprehensive approach to understanding user communication. CV4FeC combines computer vision and sentiment analysis to provide valuable insights into user reactions. A preliminary evaluation demonstrated its effectiveness in enhancing feedback collection in software development.*

Resumo. *O desenvolvimento ágil de software e o Design Centrado no Usuário enfatizam o alinhamento com as necessidades dos usuários e a priorização da coleta eficaz de feedback. Métodos tradicionais de feedback, como entrevistas e questionários, muitas vezes não capturam as reações dos usuários. Para resolver isso, introduzimos o CV4FeC, uma ferramenta que integra o OpenCV para analisar expressões faciais e sentimentos, oferecendo uma abordagem abrangente para entender a comunicação dos usuários. O CV4FeC combina visão computacional e análise de sentimentos para fornecer insights valiosos sobre as reações dos usuários. Uma avaliação preliminar demonstrou sua eficácia em aprimorar a coleta de feedback no desenvolvimento de software.*

1. Introduction

Agile software development is an iterative and adaptable process that initiate by rapidly transforming ideas into viable minimum requirements [Mehta and Sood 2023]. This development cycle includes short sprints that produce functional product increments, during which software teams continually gather user feedback. Agile methods emphasize frequent reviews and retrospectives to ensure the software aligns with user needs, thereby facilitating continuous enhancements and reinforcing collaboration [Ågren et al. 2022].

In this context, User-Centered Design (UCD) approaches emphasize the prioritization of user needs and expectations [Shania et al. 2023, Parizi et al. 2022], placing greater importance on collecting effective user *feedback* from software prototypes [Hehn et al. 2020]. For example, [Li et al. 2023] highlights the increasing significance of user feedback in improving software products. They note that, while organizations have incorporated multiple sources for feedback collection, such as social networks, many still depend on traditional methods like interviews, emails, and questionnaires.

However, traditional methods for feedback collection have demonstrated limitations in capturing user reactions and perceptions, potentially leading to development gaps, such as inaccuracies in requirement specifications [Obaidi and Klünder 2021]. To address this issue, we introduce CV4FeC (Computer Vision for Feedback Collection), a tool that integrates OpenCV to analyze facial expressions and sentiments. Our goal is to propose a more holistic approach to interpreting users' verbal and non-verbal communications, thereby enhancing the effectiveness of feedback collection from prototypes. This approach contributes to richer and more detailed feedback activities and strengthens the application of UCD techniques in software development.

The remainder of this paper is organized as follows: Section 2 introduces Computer Vision, Sentiment Analysis, and the applications of both concepts in Software Engineering. Next, Section 3 presents the proposed CV4FeC tool itself. Section 4 shows the results of the early evaluation study conducted, while Section 5 discusses the final considerations, highlighting the limitations of the work and future research directions.

2. Background

Computer Vision (CV) involves both the capture of raw data and the interpretation of information extracted from images [Bekhit and Bekhit 2022]. It integrates concepts from image processing, pattern recognition, AI, and computer graphics [Wiley and Lucas 2018].

OpenCV is one of the most widely recognized libraries in the field of Computer Vision[1]. It provides functionalities such as facial recognition, object detection, and motion tracking [Khan et al. 2019]. OpenCV is utilized in both commercial and academic applications, including areas like robotics and human-computer interactions. Its capabilities allow for the visual analysis of user reactions, offering insights into their emotions and supporting sentiment analysis efforts [Sharma et al. 2021].

Sentiment analysis plays a crucial role in decision-making processes. The task of identifying individuals' emotional states through machine learning is particularly relevant in fields like affective computing [Malviya et al. 2020]. Machine learning algorithms help determine whether an analyzed object conveys positive, neutral, or negative emotions [Zhang et al. 2018]. In this context, sentiment analysis contributes to understanding users' perceptions of a product or service and has been applied in software development to identify quality issues, improve user satisfaction, and manage user experience [Obaidi et al. 2022].

The integration of Computer Vision and sentiment analysis in Software Engineering (SE) is well-documented in the literature. [Obaidi and Klünder 2021] conducted a systematic review identifying 80 studies that apply sentiment analysis in SE, with the aim of improving communication among developers. Similarly, [Lin et al. 2018] highlights the adoption of sentiment analysis tools within the SE community, where they are used to assess the polarity of app reviews, identify negative opinions about APIs, and monitor the emotional well-being of teams.

[1] https://opencv.org/

In this context, we introduce an innovative tool that combines Computer Vision and sentiment analysis to enhance feedback collection from prototypes in software development processes.

3. CV4FeC: Computer Vision For Feedback Collection

This paper presents CV4FeC, a tool designed to collect and analyze feedback through users' facial reactions to software prototypes. Our objective is to integrate computer vision and sentiment analysis to thoroughly explore users' responses to proposed software solutions in the form of prototypes, thereby contributing to the development team's decision-making process.

In designing and conducting the initial evaluation of CV4FeC, we were guided by the Design Science Research (DSR) methodology [Runeson et al. 2020]. This approach emphasized the Design and Validation stages within the solution space[2]. As a result, this study presents the developed artifacts, which represent the constructed solution.

3.1. Solution Design

During the Solution Design stage, we conducted brainstorming sessions with three researchers in Software Engineering from the Lardev Research group and one professional from software industry. These sessions took place between October and December 2023. As a result, the participants proposed the concept of CV4FeC and outlined a solution design process, which included: 1. Identifying personas related to *feedback* collection from prototypes; 2. Creating low-fidelity prototypes; and 3. Proposing an architecture diagram for the CV4FeC tool, along with an initial implementation that incorporates the OpenCV library.

3.1.1. Personas Definition

During the brainstorming sessions, we identified personas to represent user interactions with the tool. We followed the approach outlined by [Karolita et al. 2023] for defining personas. Figure 1 illustrates the two personas we defined, named Emily and Mark. This duality reflects the main requirements of CV4FeC: capturing detailed user feedback and enabling developers to refine the software based on these data, thereby highlighting the interaction between user experience and software development within a feedback cycle.

3.1.2. Low-level fidelity prototypes construction

We constructed low-fidelity prototypes of the tool, drawing inspiration from the defined personas, Emily and Mark. In Figure 2-A, users can log in to the tool. In Figure 2-B, a DevTeam user can view the feedback analyses already conducted and create a new feedback collection using the "+" button in the top right corner. Figure 2-C presents the fields for creating a new feedback collection, including the project's name (identifier), date, prototypes to be evaluated (with upload capability), and the emails of users who will validate them. Finally, in Figure 2-D, the DevTeam can access a completed project (e.g., Proj

[2]The Problem Understanding stage was achieved through a non-systematic literature review.

1 from Figure 2-B) and review user reactions while analyzing the uploaded prototypes. On this screen, it is possible to view the sentiment graph for each screen and the percentage breakdown of each collected sentiment, which includes Angry, Disgusted, Fearful, Happy, Neutral, Sad, and Surprised[3]. At the bottom of Figure 2-D, a general evaluation of the prototypes in terms of the mentioned sentiments (*overall score*) is displayed.

Figure 2-E and Figure 2-F show the features of CV4FeC for the user evaluating the prototypes (Persona Emily). In Figure 2-E, the tool provides text explaining how the evaluation works, the prototype to be evaluated, start and stop buttons, and the user's image that is being collected via OpenCV. Additionally, a bar indicating the remaining time for each prototype is displayed to assist the user.

In the initial version of CV4FeC, prototype evaluations are conducted manually, requiring users to independently initiate the reaction collection process. A "stop" button is available to terminate the evaluation at any time. The tool displays all prototypes registered by the Development Team, and upon completion of the evaluations, it concludes the evaluation cycle as shown in Figure 2-F. Users have the option to review the results for each prototype in section (F), where they can either agree with the findings or choose to restart the evaluation. If the user decides to finalize the evaluation, a "close" button is provided for this purpose.

3.1.3. Architectural Diagram Proposal and Initial Implementation

After completing the prototyping phase, we developed an architectural diagram of the tool (Figure 3). We used the Structurizr tool[4], adhering to the C4 model standard [Vázquez-Ingelmo et al. 2020]. The diagram illustrates the modules of CV4FeC and their relationships with the involved actors (User - Persona Emily and DevTeam - Persona Mark).

In CV4FeC, DevTeam users can create projects and upload prototypes (as images) to be evaluated by users through the FaceExpression Collection module. Once the prototypes are uploaded, users can access the tool, which collects reactions to the prototypes using the OpenCV library (OpenCV face analysis module). The tool gathers this data via the user's video camera. After collecting the reactions, CV4FeC sends the feedback

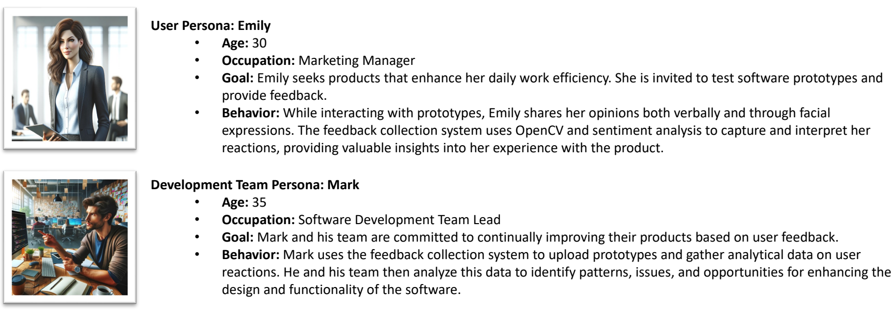

Figura 1. Personas Emily and Mark

[3]These reactions are listed based on the dataset used in the initial development of the tool.
[4]https://structurizr.com/dsl

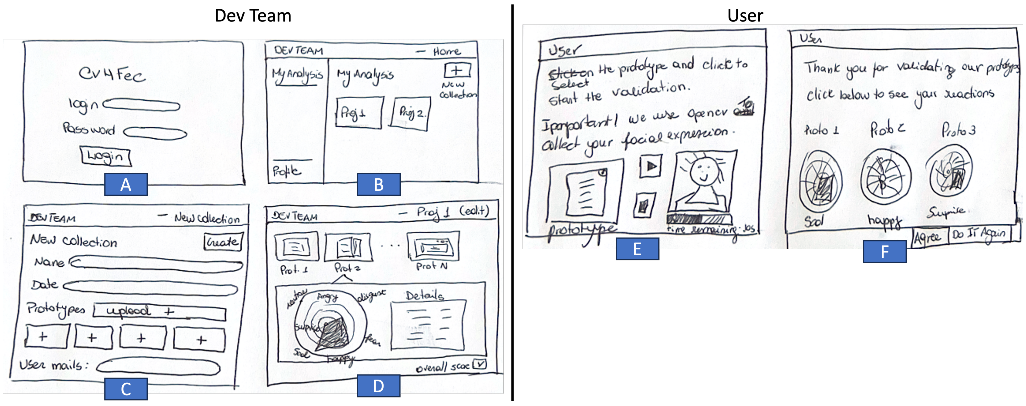

Figura 2. Low level fidelity prototypes CV4FeC for Personas Emily and Mark

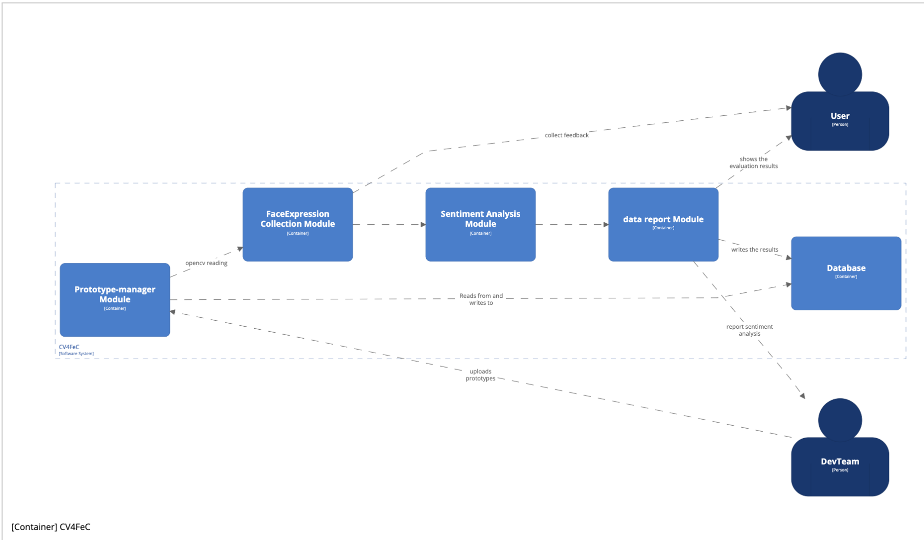

Figura 3. Architectural Diagram based on C4 model

to the Sentiment Analysis module, which generates an analysis graph and forwards it to the development team through the Data Report module. Additionally, the CV4FeC tool integrates with a database for data storage (user reactions, projects, etc.)[5].

We developed CV4FeC using Python, a programming language that supports the implementation of OpenCV libraries, image datasets, and training based on facial recognition. For instance, CV4FeC uses Haar Cascade for facial expression detection, a well-known method for object detection in images, particularly in facial recognition [Minu et al. 2020]. The FERC-2013 dataset [Sambare 2023] is employed for training facial recognition models [Sang et al. 2017]. The training set contains 28,709 examples, while the public test set includes 3,589 examples. The objective is to classify each face based on the emotion expressed in the facial expression into one of seven categories: 0 = Anger, 1 = Disgust, 2 = Fear, 3 = Happy, 4 = Sad, 5 = Surprise, and 6 = Neutral.

Figure 4 shows the initial screen of the CV4FeC tool, which displays the projects

[5]CV4FeC does not store images of users, in compliance with the General Data Protection Law.

and data collected from users who evaluate the prototypes[6]. When clicking on the "Quick Results" button in Project 1, the tool presents the screen shown in Figure 5. In Figure 5, three key pieces of project information are displayed in columns. The left column contains the prototypes viewed by the user during evaluation. The central column shows the collection of user reactions to the prototypes, and the right column displays the graph of the feedback results collected from the user. In this example, the "Happy" emotion was the most prominent among the others[7].

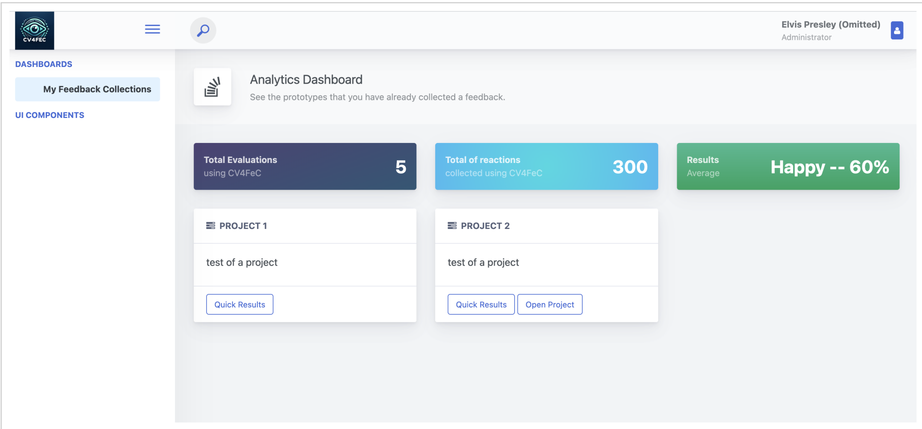

Figura 4. Initial Implementation - Projects screen

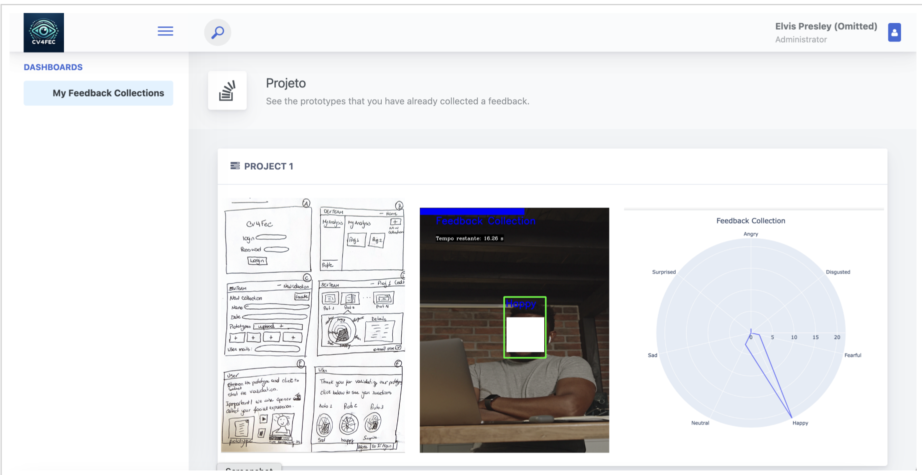

Figura 5. Initial implementation - feedback collection and report screen

3.2. Early Tool Evaluation and Data Analysis

To conduct a preliminary evaluation of CV4FeC, we utilized its initial implementation to collect feedback on the tool's own developed prototypes. Five researchers from our research group, who were not involved in the development process, participated in this initial assessment.

We collected feedback using the Technology Acceptance Model (TAM) and the System Usability Scale (SUS) [Davis 1989, Brooke 1996], as shown in Table 1 and Table

[6]The data presented is fictitious, as it is part of a tool proposal.
[7]For the purposes of reviewing the article, the image of the user using the tool is fictitious.

2, respectively. For the quantitative data analysis, we applied statistical methods to evaluate the collected feedback. This analysis enabled us to quantify user perceptions and usability, offering valuable insights into the effectiveness of CV4FeC in capturing and analyzing user feedback through facial reactions.

Tabela 1. TAM Statements for CV4FeC

Perceived Usefulness	
PU 1	Using CV4FeC increases the effectiveness of collecting feedback from software prototypes.
PU 2	CV4FeC improves the quality of feedback analysis through facial reactions.
PU 3	Using CV4FeC allows the development team to make more informed decisions about the prototypes.
PU 4	CV4FeC speeds up the process of evaluating software prototypes.
Perceived Ease of Use	
PEU 1	Interacting with CV4FeC is easy and understandable.
PEU 2	CV4FeC is intuitive and does not require much effort to use.
PEU 3	Learning to use CV4FeC was easy for me.
PEU 4	Navigating through the features of CV4FeC is straightforward.
Intention to Use	
IU 1	I intend to continue using CV4FeC to collect feedback on prototypes in the future.
IU 2	I would use CV4FeC in other software development projects for feedback collection.
IU 3	I would recommend using CV4FeC to colleagues in similar projects.
IU 4	I see value in integrating CV4FeC into future software development processes.

Tabela 2. SUS Statements for CV4FeC

Question	**Statement**
Question 1	I think that I would like to use CV4FeC frequently.
Question 2	I found CV4FeC unnecessarily complex
Question 3	I thought CV4FeC was easy to use
Question 4	I would need the support of a technical person to use CV4FeC
Question 5	I found the various functions in CV4FeC were well integrated
Question 6	I thought there was too much inconsistency in CV4FeC
Question 7	I would imagine that most people would learn to use CV4FeC very quickly
Question 8	I found CV4FeC very confusing to use
Question 9	I felt very confident using CV4FeC
Question 10	I needed to learn a lot of things before I could get going with CV4FeC.

4. Preliminary Results

This section presents the results of a preliminary evaluation conducted on the CV4FeC tool. At this stage, we collected feedback through simulations using videos sourced from public databases on the internet, featuring people expressing various reactions.

4.1. Technology Acceptance Model Evaluation

Figure 6 presents the results of the TAM evaluation for CV4FeC. The analysis shows that participants overwhelmingly responded positively across all categories: Perceived Usefulness, Perceived Ease of Use, and Intention to Use. The majority of respondents either "Totally Agree" or "Strongly Agree" with the statements provided. This indicates that users found the CV4FeC tool effective in improving the feedback collection process for software prototypes, easy to use, and something they intend to continue using in future projects. The consistently high ratings across all TAM statements reflect the tool's strong usability and perceived value among its users.

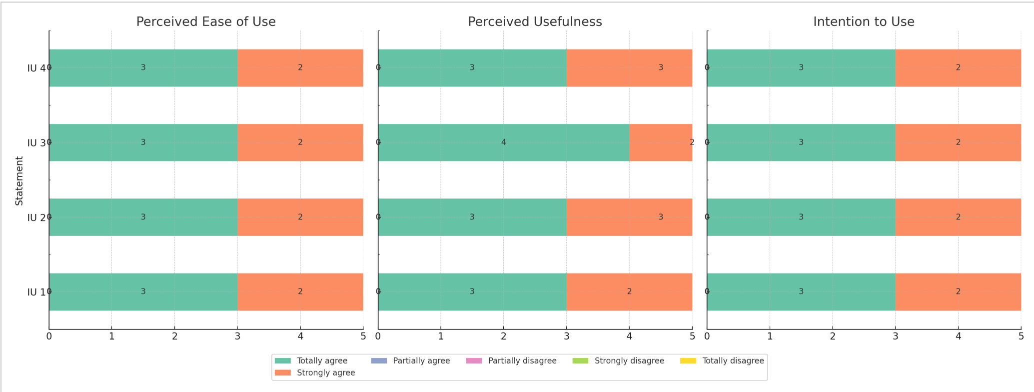

Figura 6. TAM Final

4.2. System Usability Scale Evaluation

Blattgerste *et al.* (2022) [Blattgerste et al. 2022] introduced an online analysis tool specifically designed for the System Usability Scale (SUS). This tool simplifies the calculation of the SUS score, median values, response distribution, and other key metrics. Figure 7 shows the SUS scale as applied to CV4FeC. The overall SUS score was 97.5 (Percentile Value), placing it in the "Best Imaginable" category.

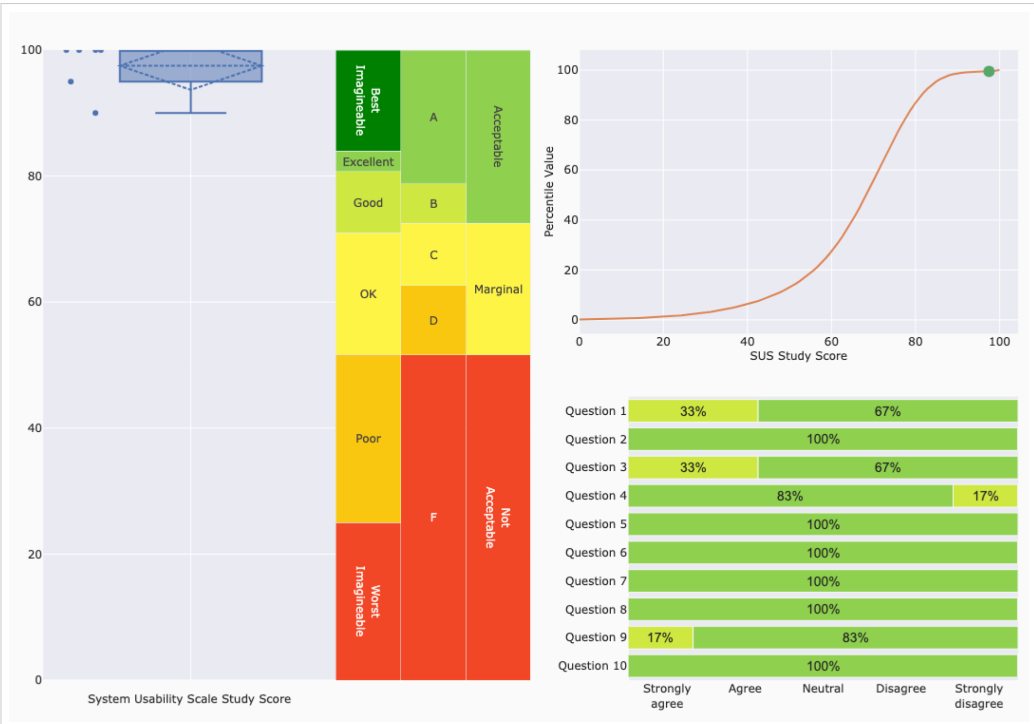

Figura 7. SUS results

5. Final Remarks

This paper introduced CV4FeC as an innovative tool for software development, enabling teams to collect and analyze feedback through the integration of facial expression and sentiment analysis. This approach not only enhances the accuracy of specifying functional requirements but also aligns software design with user expectations.

However, given its nature, our study faces several validity threats. The reliance on facial expression and sentiment analysis may introduce biases due to variations in user expressions, potentially affecting feedback accuracy. To address this, we utilized the FERC-2013 dataset and conducted diverse user testing, though the findings may not be fully generalizable. Additionally, the feedback reflects the subjective views of our participants, which may not represent the broader user population. We mitigated this by conducting multiple evaluation iterations using TAM and SUS methods. Moreover, the specific software prototypes tested could influence perceptions of the tool's usefulness and ease of use; we minimized this by selecting a variety of prototypes, but the results should be interpreted with this limitation in mind.

Despite these limitations, the study shows that CV4FeC is effective in a controlled environment and holds significant potential to enhance feedback collection in software development. Future work will focus on expanding the tool's application to a wider range of user groups and prototypes, as well as refining the analysis algorithms to further improve accuracy and generalizability.

Referências

Ågren, P., Knoph, E., and Berntsson Svensson, R. (2022). Agile software development one year into the covid-19 pandemic. *Empirical Software Engineering*, 27(6):121.

Bekhit, A. F. and Bekhit, A. F. (2022). Introduction to Computer Vision. *Computer Vision and Augmented Reality in iOS: OpenCV and ARKit Applications*, pages 1–20.

Blattgerste, J., Behrends, J., and Pfeiffer, T. (2022). A Web-Based Analysis Toolkit for the System Usability Scale. In *Proceedings of the International Conference on PErvasive Technologies Related to Assistive Environments*, pages 1–10, Corfu, Greece. ACM.

Brooke, J. (1996). SUS-A Quick and Dirty Usability Scale. *Usability Evaluation in Industry*, 189(194):4–7.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 13(3):319–340.

Hehn, J., Mendez, D., Uebernickel, F., Brenner, W., and Broy, M. (2020). On Integrating Design Thinking for Human-Centered Requirements Engineering. *IEEE Software*, 37(2):25–31.

Karolita, D., Grundy, J., Kanij, T., Obie, H., and McIntosh, J. (2023). What's in a persona? a preliminary taxonomy from persona use in requirements engineering. In *International Conference on Evaluation of Novel Approaches to Software Engineering 2023*, pages 39–51. Scitepress.

Khan, M., Chakraborty, S., Astya, R., and Khepra, S. (2019). Face Detection and Recognition using OpenCV. In *Proceedings of the International Conference on Computing, Communication, and Intelligent Systems*, pages 116–119. IEEE.

Li, Z. S., Arony, N. N., Devathasan, K., Sihag, M., Ernst, N., and Damian, D. (2023). Unveiling the life cycle of user feedback: Best practices from software practitioners.

Lin, B., Zampetti, F., Bavota, G., Di Penta, M., Lanza, M., and Oliveto, R. (2018). Sentiment Analysis for Software Engineering: How Far Can We Go? In *Proceedings of* *the International Conference on Software Engineering*, ICSE ’18, page 94–104, New York, NY, USA. ACM.

Malviya, S., Tiwari, A. K., Srivastava, R., and Tiwari, V. (2020). Machine learning techniques for sentiment analysis: A review. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 12(02):72–78.

Mehta, K. and Sood, V. M. (2023). Agile software development in the digital world–trends and challenges. *Agile Software Development: Trends, Challenges and Applications*, pages 1–22.

Minu, M., Arun, K., Tiwari, A., and Rampuria, P. (2020). Face recognition system based on haar cascade classifier. *International Journal of Advanced Science and Technology*, 29(5):3799–3805.

Obaidi, M. and Klünder, J. (2021). Development and Application of Sentiment Analysis Tools in Software Engineering: A Systematic Literature Review. In *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering*, EASE ’21, page 80–89, New York, NY, USA. ACM.

Obaidi, M., Nagel, L., Specht, A., and Klünder, J. (2022). Sentiment Analysis Tools in Software Engineering: A Systematic Mapping Study. *Information and Software Technology*, 151:107018.

Parizi, R., Prestes, M., Marczak, S., and Conte, T. (2022). How has Design Thinking being Used and Integrated into Software Development Activities? A Systematic Mapping. *Journal of Systems and Software*, 187:1–27.

Runeson, P., Engström, E., and Storey, M.-A. (2020). *The Design Science Paradigm as a Frame for Empirical Software Engineering*, pages 127–147. Springer.

Sambare, M. (2023). Fer2013 facial expression recognition dataset. Kaggle. Acessado em Dezembro de 2023.

Sang, D. V., Van Dat, N., et al. (2017). Facial expression recognition using deep convolutional neural networks. In *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*, pages 130–135. IEEE.

Shania, M., Raharjo, T., and Fitriani, A. N. (2023). Implementation user-centered design in agile software development: Systematic literature review. *Indonesian Journal of Multidisciplinary Science*, 2(7):2812–2831.

Sharma, A., Pathak, J., Prakash, M., and Singh, J. (2021). Object Detection using Opencv and Python. In *Proceedings of the International Conference on Advances in Computing, Communication Control and Networking*, pages 501–505. IEEE.

Vázquez-Ingelmo, A., García-Holgado, A., and García-Peñalvo, F. J. (2020). C4 model in a software engineering subject to ease the comprehension of uml and the software. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 919–924.

Wiley, V. and Lucas, T. (2018). Computer vision and image processing: a paper review. *International Journal of Artificial Intelligence Research*, 2(1):29–36.

Zhang, J., Li, Y., Tian, J., and Li, T. (2018). Lstm-cnn hybrid model for text classification. In *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 1675–1680. IEEE.