

When Work Relations Matter: A Vision of Forgiveness in Software Testing

Mary Sánchez-Gordón¹  

Østfold University of Applied Sciences, Halden, Norway

Ricardo Colomo-Palacios  

Universidad Politécnica de Madrid, Spain

Rahul Mohanani  

University of Oulu, Finland

Abstract

The software testing process is inherently adversarial, creating conditions for frequent and pervasive developer–tester conflicts. Although forgiveness has been studied as a tool for resolving conflicts and as a coping mechanism in the workplace context that promotes well-being and productivity, its role in software testing remains unexplored. This paper posits that cultivating forgiveness can help practitioners manage their emotional responses to conflict and therefore deserves further investigation in the context of software testing.

2012 ACM Subject Classification Social and professional topics → Management of computing and information systems

Keywords and phrases Forgiveness, Emotional regulation, Conflicts, Software testing

Digital Object Identifier 10.4230/LIPIcs.ESEM.2026.73

Category Emerging Results, Vision & Reflection Track Paper

1 Introduction

The nature of testing differs from other software development activities, such as design and building. Although the latter activities focus on the construction of software, testing is often perceived as destructive, as testers attempt to “break” the software built by developers [19, 12, 23]. Apart from the different work mindset (build vs break), previous research has identified other dimensions [23]: thinking focus (theory vs practice), project goal (efficiency vs effectiveness) and job knowledge (depth vs breadth). Testers and developers perceive the software development process differently [4]. From the testing perspective, those human beings who build the software make errors (mistakes), which produce defects (faults, bugs), which in turn may result in failures [19].

An interview study on the characteristics of good software found that testers believe that they often give bad news (defects or problems) and that it must be communicated in a way that does not blame anyone for it [5]. The idea of testers as “bearers of bad news” is widely spread and acknowledged by the International Software Testing Qualifications Board (ISTQB) [19]. Moreover, ISTQB point out that a common human tendency is to blame the bearer of bad news, and that test result communication is frequently perceived as criticism of both the software and its author [19]. Indeed, the testers interviewed in a study on conflict in software testing [4] reported the developer’s reluctance to accept the existence of errors, as they tend to view their code as an extension of themselves and thus took it personally when defects and failures were communicated.

¹ corresponding author



© Mary Sánchez-Gordón, Ricardo Colomo-Palacios, and Rahul Mohanani;
licensed under Creative Commons License CC-BY 4.0

20th International Symposium on Empirical Software Engineering and Measurement (ESEM 2026).

Editors: Robert Feldt, Maria Paasivaara, Daniel Mendez, Stefan Wagner, and Marvin Muñoz Barón;
Article No. 73; pp. 73:1–73:6



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

As illustrated, the software testing process is inherently adversarial, setting the stage for developer-tester conflict [4]. Previous research supports the notion that conflict between software testers and developers is inevitable [4, 18, 24]. In this sense, a study has proposed three basic conflict layers in software testing that include organization, process, and people [4], while another identified process, people, and communication [24]. The impact of interpersonal conflict has been perceived to be negative, regardless of how it was managed or resolved [2]. Their effect can be minimized through communication, mutual respect, and even social interaction [4]. However, communication during conflicts can be difficult and the way testers and developers feel about their work and workplace relationships matters. A more recent study [25] also supports that conflict in software testing arises from people.

Given that conflict involves emotions, perceptions, and behaviors [18], this vision paper argues that software testing involves actions that developers and testers perceive as disrespect and exploitation, violating values, rules, or norms in their relationship. In psychological research, such perceived violations are referred to as *interpersonal transgressions* [22]. In simple words, an act that goes against what is considered acceptable or proper. Recognizing this, it becomes necessary to bring emotionality to the fore. Although developers and testers could choose how to act in response to their emotions, they cannot choose the emotions themselves – whether anger, sadness, or happiness. From this perspective, forgiveness emerges as a recognized conflict management strategy within organizations that cannot only diminish the detrimental impact of interpersonal conflict, but can also promote well-being and productivity [3]. However, its role in software testing remains unexplored.

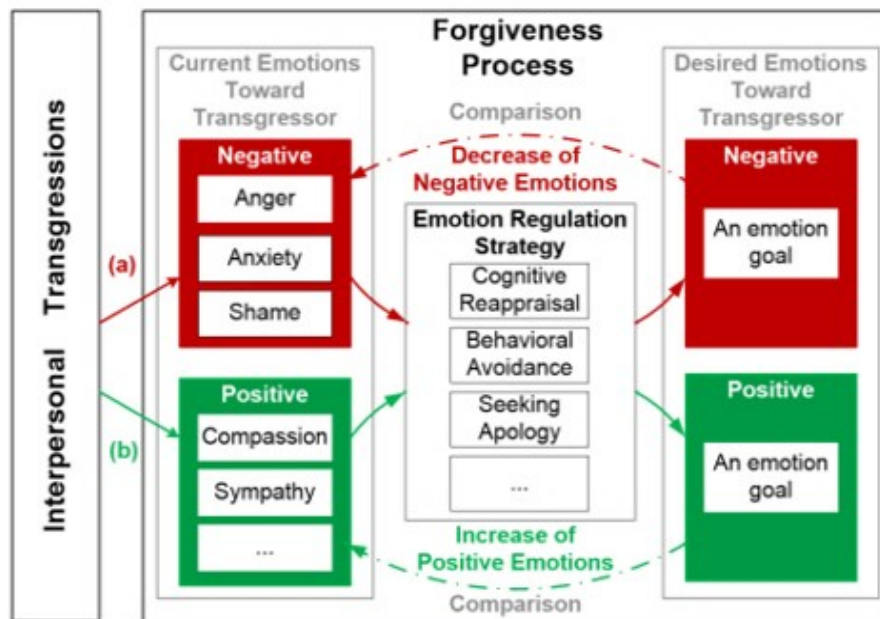
To address this gap, this paper presents a vision and research agenda for understanding forgiveness through the lens of emotion regulation. It conceptualizes forgiveness and outlines future research directions aimed at helping the software testing community foster healthier workplace relationships and better manage interpersonal conflict.

2 Forgiveness Conceptualization

The human aspects of software engineering have gained increasing attention due to their influence on interactions among software practitioners [10, 20]. One such aspect closely related to forgiveness is empathy, which has been investigated in the context of software requirements engineering [10]. However, forgiveness and empathy can remain independent: one may empathize with others without forgiving them, just as one may forgive without fully understanding the motives of the other persons. Social interactions also play a central role in the shaping of testing culture [20]. Thus, an affective dimension influences both the tester's experience (TX) [6, 20] and the developer's experience (DX) [8].

In this context, we need to choose the conceptualization of forgiveness that we use in the software testing context. To do this, we first review the forgiveness literature. Forgiveness has attracted scholarly attention across scientific disciplines, beginning with philosophy and psychology, e.g., [7]. More recently, forgiveness has gained attention within organizational and managerial science, e.g., [1]. Some reviews on measuring forgiveness highlight that forgiveness measures assess forgiveness differently [3, 9]. Although there is still no consensus on what fundamentally constitutes forgiveness or how it should be defined [3], the definitions of psychological research have focused on the intrapersonal aspects of forgiveness [13].

Some conceptual issues have emerged over time. To better understand the challenges for the study of forgiveness in organizational behavior, Brady et al. [3] conducted a critical review and analysis of the literature. Among the identified issues, they highlighted that many existing conceptualizations and operationalizations remain atheoretical, since they are



■ **Figure 1** Forgiveness process involving the decrease of negative emotions (a) and the increase of positive emotions (b) adapted from Brady et al. [3].

based on inductive research grounded in laypeople's understanding of forgiveness. This can lead to conceptual issues because this understanding is shaped by beliefs or ways of thinking about forgiveness, which in turn influence the experiences of laypeople [13]. Brady et al. [3] also proposed reconceptualizing forgiveness as a special case of emotion regulation, as shown in Figure 1.

Forgiveness as a form of emotion regulation occurs after an interpersonal transgression [3]. It involves an intraindividual change where individuals aim to improve their emotional response toward a transgressor by reducing negative emotions and/or increasing positive ones. This process is guided by a discrepancy between two emotional states: the current emotional state (how the person feels now) and the desired emotional state (how they want to feel). The current state reflects the emotional aspect of forgiveness, while the desired state represents a cognitive goal. When individuals recognize this emotional gap, they can consciously or unconsciously engage in forgiveness by using emotion regulation strategies – such as cognitive reappraisal – to align their current emotions with their emotional goals over time. For example, reinterpreting a transgression as unintentional can help decrease negative emotions and move the individual closer to emotional resolution. Common emotion regulation strategies identified in the psychological literature [16] include acceptance, avoidance, cognitive reappraisal, distraction, mindfulness, rumination, and suppression. These strategies are highly context-dependent and can take intraindividual, interpersonal, cognitive, or behavioral forms. Consequently, selecting an effective strategy can be complex and challenging, and people often struggle with emotion misregulation [3].

Finally, although the Eastern view of forgiveness as “compassion in action” [17] provides an example of a discrepancy, this concept is not examined in detail. In Eastern philosophy, pain is inevitable, but suffering arises from attachment to it. Consequently, forgiveness can be understood as the deliberate practice of employing emotion regulation strategies to release such attachment, transform emotional pain, and reduce unnecessary suffering that widens the emotional gap.

3 Forgiveness in Relation to Adjacent Constructs

This vision paper acknowledges that developer–tester relationships vary across organizational cultures and development approaches and argues that forgiveness should complement rather than replace adjacent constructs such as accountability, managerial responsibility, psychological safety, conflict resolution, or conflict management.

Unlike psychological safety and a blameless culture, which are team-level characteristics shaped by organizational culture, forgiveness is an intraindividual process of repairing emotional responses to perceived transgressions. Psychological safety does not mean the absence of blame. Blame may be appropriate when people intentionally cause or risk harm. Similarly, forgiveness is not appropriate in all situations, such as abuse, repeated misconduct, persistent disrespect, or unhealthy team cultures. These situations, as well as other conflicts rooted in structural issues such as process failures or power imbalances, can also require appropriate intervention rather than forgiveness alone.

In addition, psychological safety is improved by making explicit what types of behavior are acceptable and unacceptable within the organization [14] and by promoting practices such as blameless postmortems [21, 15] that focus on systems, processes and organizational learning rather than individual blame. However, neither psychological safety nor blameless postmortems eliminate conflict. Team members inevitably bring diverse experiences, perspectives, and past traumas into their interactions, and unresolved conflicts can persist. Forgiveness addresses this gap by helping individuals regulate emotions and gradually adopt more prosocial attitudes toward a transgressor. In this way, forgiveness can also complement conflict management and conflict resolution, which focus on reaching agreement between involved parties.

4 Research Agenda for Forgiveness in Software Testing

The research agenda is structured around three key themes: (i) Identifying conflicts and forgiveness in software testing, (ii) Understanding the role of forgiveness as a response to conflicts in testing practice, and (iii) Cultivating forgiveness within the testing culture.

i) Identifying conflicts and forgiveness in software testing. To the best of our knowledge, there is currently no empirical evidence on forgiveness in software testing. However, conflicts vary in severity and can arise from a wide range of testing-related issues. Future research should aim to:

- Identify the conflicts, including ethical and non-ethical workplace transgressions, that feel like interpersonal transgressions and trigger the forgiveness process.
- Map the (current and desired) emotions toward transgressors that are triggered by conflicts.
- Determine what emotion regulation strategies lead to a decrease or increase in emotions and how they affect working relationships over time.

Conceptual research approaches, such as thematic analysis and grounded theory, can help establish this theoretical foundation.

ii) Understanding the role of forgiveness as a response to conflicts in testing practice. Emotion regulation offers a path for navigating conflicts but the willingness to forgive is closely related to situational factors [11] that is a nonlinear and evolving journey. Thus,

researchers must investigate what motivates software practitioners involved in testing to demonstrate forgiveness. Potential contributions include:

- Identify the situational factors that shape forgiveness throughout the Software Testing LifeCycle (STLC).
- Identify perceived forgiveness-related challenges (e.g., cognitive distortion like black-and-white thinking) and benefits (e.g., restore motivation).
- Examine how practitioners employ emotion regulation strategies to repair damaged relationships and overcome debilitating thoughts and emotions.

Ethnographic fieldwork, team-based case studies, and situated interviews can uncover how forgiveness manifests in software testing practice. These studies are critical to building models of shared accountability and to inform governance policies.

iii) Cultivating Forgiveness within the Testing Culture. A key area of the agenda is the development of training programs to cultivate forgiveness. This includes:

- Identify skills to facilitate the forgiveness process and support a forgiveness testing culture within software projects.
- Understand how to build a common sense of the value of forgiveness in software culture.
- Define a set of actions that can be followed to increase awareness of forgiveness as a potential response to conflicts.
- Determine how to provide responsive managerial support during conflicts to improve practitioner well-being and strengthen software testing practices.

Collectively, these research directions define a path towards informing a set of actionable practices that ensure forgiveness is recognized, supported, and integrated into the software testing culture, allowing practitioners to continue working together in healthy and constructive ways despite interpersonal conflicts.

We acknowledge that conflict in software testing is inevitable and hope that this research agenda will bring attention to social interactions in software testing, especially the cultivation of forgiveness as a promising research direction.

References

- 1 Karl Aquino, Steven L. Grover, Barry Goldman, and Robert Folger. When push doesn't come to shove: Interpersonal forgiveness in workplace relationships. *J. Manag. Inq.*, 12(3):63–80, 2003. doi:10.1177/1056492603256337.
- 2 Henri Barki and Jon Hartwick. Interpersonal conflict and its management in information system development. *MIS Q.*, 25(2):195–228, 2001. URL: <http://misq.org/interpersonal-conflict-and-its-management-in-information-system-development.html>.
- 3 Daniel L. Brady, Maria Francisca Saldanha, and Laurie J. Barclay. Conceptualizing forgiveness: A review and path forward. *J. Organ. Behav.*, 44(2):261–296, 2023. doi:10.1002/job.2632.
- 4 Cynthia F. Cohen, Stanley J. Birkin, Monica J. Garfield, and Harold W. Webb. Managing conflict in software testing. *Commun. ACM*, 47(1):76–81, 2004. doi:10.1145/962081.962083.
- 5 Anca Deak. What characterizes a good software tester? - A survey in four norwegian companies. In *Testing Software and Systems - 26th IFIP WG 6.1 International Conference, ICTSS 2014, Madrid, Spain, September 23-25, 2014. Proceedings*, Lecture Notes in Computer Science, pages 161–172. Springer, 2014. doi:10.1007/978-3-662-44857-1_11.
- 6 Oswald Mesumbe Ekwoje, Awdren de Lima Fontão, and Arilo Claudio Dias-Neto. Tester experience: Concept, issues and definition. In *41st IEEE Annual Computer Software and Applications Conference, COMPSAC 2017, Turin, Italy, July 4-8, 2017. Volume 1*, pages 208–213. IEEE Computer Society, 2017. doi:10.1109/COMPSAC.2017.232.

- 7 Robert D Enright, Ching ru Wu, Leanne Olson, Jeffrey Chandler, Elizabeth Gassin, Suzanne Freedman, Sandra Golden, and Issidoros Sarinopoulos. Piaget on the moral development of forgiveness: Identity or reciprocity. *Hum. Dev.*, 37:63–80, 1994. doi:10.1159/000278239.
- 8 Fabian Fagerholm and Jürgen Münch. Developer experience: Concept and definition. In *International Conference on Software and System Process, ICSSP 2012*, pages 73–77. IEEE Computer Society, 2012. doi:10.1109/ICSSP.2012.6225984.
- 9 Feng Gao, Yuanwei Li, and Xuejun Bai. Forgiveness and subjective well-being: A meta-analysis review. *Pers. Individ. Dif.*, 186, 2022. doi:10.1016/j.paid.2021.111350.
- 10 Hashini Gunatilake, John C. Grundy, Rashina Hoda, and Ingo Mueller. Empathy guidelines for improving practitioner well-being & software engineering practices. *IEEE Softw.*, 43(3), 2026. doi:10.1109/MS.2026.3654969.
- 11 Inbal Holtzman and Galit Nimrod. Forgiveness in human-machine interaction. *Frontiers Comput. Sci.*, 7, 2025. doi:10.3389/fcomp.2025.1617471.
- 12 Tanjila Kani, Robert G. Merkel, and John C. Grundy. An empirical investigation of personality traits of software testers. In *8th IEEE/ACM International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2015*, pages 1–7. IEEE Computer Society, 2015. doi:10.1109/CHASE.2015.7.
- 13 Kathleen A. Lawler-Row, Cynthia A. Scott, Rachel L. Raines, Meirav Edlis-Matityahou, and Erin W. Moore. The varieties of forgiveness experience: Working toward a comprehensive definition of forgiveness. *J. Relig. Health*, 46:233–248, 2007. doi:10.1007/s10943-006-9077-y.
- 14 Per Lenberg and Robert Feldt. Psychological safety and norm clarity in software engineering teams. In *11th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2018*, pages 79–86. ACM, 2018. doi:10.1145/3195836.3195847.
- 15 John Lunney and Sue Lueder. Postmortem culture: Learning from failure. In Betsy Beyer, Chris Jones, Jennifer Petoff, and Niall Richard Murphy, editors, *Site Reliability Engineering: How Google runs production systems*. O'Reilly Media, Inc., 2016.
- 16 Kristin Naragon-Gainey, Tierney P McMahon, and Thomas P Chacko. The structure of common emotion regulation strategies: A meta-analytic examination. *Psychol. Bull.*, 143(4):384–427, 2017. doi:10.1037/bul0000093.
- 17 Victor Parachin. *Eastern Wisdom for Western Minds*. Orbis Books, 2015.
- 18 Steve Sawyer. Effects of intra-group conflict on packaged software development team performance. *Inf. Syst. J.*, 11(2):155, 2001. doi:10.1046/j.1365-2575.2001.00100.x.
- 19 Lucjan Stapp, Adam Roman, and Michaël Pilaeten. *ISTQB® Certified Tester Foundation Level - A Self-Study Guide Syllabus v4.0*. Springer, 2024. doi:10.1007/978-3-031-42767-1.
- 20 Mark Swillus, Rashina Hoda, and Andy Zaidman. On the emergence of testing strategies: A socio-technical grounded theory. *Empir. Softw. Eng.*, 31(147), 2026. doi:10.1007/s10664-026-10828-0.
- 21 Bianca Trinkenreich, Marco Aurelio Gerosa, Anita Sarma, and Igor Steinmacher. Guidelines for cultivating a sense of belonging to reduce developer burnout. *IEEE Softw.*, 42(2):84–91, 2025. doi:10.1109/MS.2024.3404361.
- 22 Lydia Woodyatt, Michael Wenzel, Tyler G. Okimoto, and Michael Thai. Interpersonal transgressions and psychological loss: Understanding moral repair as dyadic, reciprocal, and interactionist. *Curr. Opin. Psychol.*, 44:7–11, 2022. doi:10.1016/j.copsyc.2021.08.018.
- 23 Xihui Zhang, David W. Nickels, Robin S. Poston, and Jasbir Singh Dhaliwal. One world, two realities: Perception differences between software developers and testers. *J. Comput. Inf. Syst.*, 58(4):385–394, 2018. doi:10.1080/08874417.2017.1289355.
- 24 Xihui Zhang, Thomas F. Stafford, Jasbir Singh Dhaliwal, Mark Lee Gillenson, and Gertrude Moeller. Sources of conflict between developers and testers in software development. *Inf. Manag.*, 51(1):13–26, 2014. doi:10.1016/j.im.2013.09.006.
- 25 Xihui Zhang, Thomas F. Stafford, Tao Hu, and Hua Dai. Measuring task conflict and person conflict in software testing. *ACM Trans. Softw. Eng. Methodol.*, 29(4):25:1–25:19, 2020. doi:10.1145/3395029.