JOURNAL OF APPLIED INTELLIGENT SYSTEMS & INFORMATION SCIENCES

VOL. 4, ISSUE 2, PP. 45-51, DECEMBER 2023.
Available at: **https://journal.research.fanap.com/**
DOI: https://doi.org/10.22034/JAISIS.2023.428549.1076

TACKLING WORKLOAD SURGES IN SOFTWARE DEVELOPMENT PROJECTS

Neda Fathi [1,*]

[1] *Department of Computer Engineering, Islamic Azad University of Central Tehran Branch, Tehran, Iran.*

ABSTRACT

Owing to extreme interconnections among individuals, messaging systems have become vital tools for communication and collaboration within organizations. This paper explores a crisis that occurred during the development of a customized messaging system facing an unexpected surge in communication demands. Initially designed to accommodate a gradual increase in user activities, the sudden decision to centralize all organizational communications through the messaging system led to a crisis. This crisis manifested in a 300% increase in users, causing a flood of improvement requests and performance malfunction reports. The paper delves into the lessons learned from this crisis, highlighting both technical and mental aspects of the challenges faced. Short-term technical solutions included restricting request channels, while long-term actions involved team relocations and staffing adjustments. The paper underscores the importance of addressing not only technical deficiencies but also the mental well-being of the development team. Despite the inevitability of future crises, the study provides insights into effective crisis management strategies, ensuring a more rapid team recovery and productivity enhancement.

Keywords: Software development, Messaging System, Analytical problem-solving approach, Software incident management, Soft skills

1. INTRODUCTION

Today, given the interconnected relations within the working environment, messaging systems have become an integral part of communication and interaction management tools for workgroups. The extensive volume of exchanges is such that these tools are even used for exchanging critical business information among relevant stakeholders. In such conditions, ensuring the security of data and exchanges on one hand, and utilizing chat-based extension capabilities in other systems on the other hand, has led some information technology service providers to develop messaging platforms.

The topics of developing messaging platforms and software crisis/incident management are related to the challenges and opportunities of software engineering in the digital age. Messaging platforms are essential software applications that enable online communication and provide various features and services for users and providers. Developing messaging platforms requires meeting the diverse and evolving needs and preferences of users, as well as competing in the dynamic and competitive market of online communication (Yan & Feng,

[*] *Corresponding Author, Email: ndfathi@gmail.com*

RECEIVED: 03 DECEMBER 2023; ACCEPTED: 22 DECEMBER 2023; PUBLISHED ONLINE: 24 DECEMBER 2023

2023). Software crisis/incident management is a process that aims to prevent, detect, resolve, and learn from software failures that can cause serious problems and losses for software developers, clients, users, and stakeholders (Bannerman, 2008). Software crisis/incident management involves identifying the root causes, impacts, and solutions of software incidents, as well as implementing corrective and preventive actions, communicating with the relevant parties, and documenting the lessons learned. Software crisis/incident management is important for ensuring the reliability, security, and performance of software systems, such as messaging platforms, to improve the underlying software development process and practices.

In this context, Dotin Company, a large-scale software solution provider, has placed the development of a customized messaging system, called Talk, on its agenda. It has been progressively advancing according to a well-defined plan. However, a sudden decision to centralize all communications and messages of the organization's employees through that customized system has put the entire development process under scrutiny. The messaging system, which until recently did not bear such a heavy load, needed to suddenly respond to multiple demands. This disruption created a crisis in the development of the messaging system, and reflections on the lessons learned from this situation lie ahead.

The present research focuses on an unexpected surge in the demand rate of using the messaging system. This situation leads to a 300% increase in users and an influx of improvement requests and performance malfunction reports. The study delves into the lessons learned from this crisis, emphasizing the challenges encountered in both technical and mental aspects. This unexpected scenario highlights the importance of not only addressing technical deficiencies but also considering the mental well-being of the development team. Hence, the research is to provide a comprehensive exploration of the contextual factors, challenges, and crisis management strategies associated with this unique experience, offering valuable insights for similar scenarios in the future.

Precisely, the paper contributes to the existing studies by identifying and thoroughly analyzing a crisis scenario in the development of a customized messaging system. It outlines the technical challenges faced during the crisis, such as the system's inability to handle the sudden increase in demand while devising applied and workable remedies. It recognizes the impact of the crisis not only on the technical aspects but also on the stress and resilience of the team members. This comprehensive approach contributes valuable insights into effective crisis response in the context of software development projects.

In what follows, Section 2 deals with a literature review while Section 3 introduces the problem's specification. Next, Section 4 provides an analytical approach and applied solutions for managing the underlying crisis. Eventually, Section 5 discusses the conclusion of the paper and recommends future directions for upcoming studies.

2. Literature Review

The integration of messaging systems into organizational communication strategies has become crucial in the recent landscape of the working environment. The challenges faced by the company in the development of its messaging system provide a unique case for understanding crisis management in this context. To contextualize and enrich the understanding of the present study, a review of relevant literature on messaging systems, crisis management, and software development is presented below.

2.1. Messaging Systems in Organizational Communication

In terms of Role and Importance, the literature emphasizes the pivotal role of messaging systems in facilitating communication within organizations. These platforms are not only tools for exchanging information but also serve as integral components for collaborative work among teams (Anders, 2016; Luqman et al., 2023). Furthermore, security concerns shed light on another significant perspective on messaging systems in organizational communications. As organizations increasingly rely on messaging platforms for critical business communication, the literature highlights the importance of ensuring the security and confidentiality of data exchanges (Yang et al., 2023). This experience aligns with the broader trend of organizations centralizing communications, bringing forth security challenges (Niemimaa, 2023).

2.2. Software Scaling Challenges

Based on the scaling systems, the previous studies on software development underscore the challenges associated with scaling systems to accommodate increased user activities. Sudden surges in user demands can lead to performance issues and necessitate optimization strategies (Harman et al., 2014; Wadleigh & Crawford, 2000). The present experience of a 300% increase in users, which will be revealed later, echoes the scalability challenges discussed in these studies. Best practices in software engineering stress the importance of periodic testing and user familiarization efforts. This implies the key position of testing and user familiarization. The corresponding activities are essential for identifying potential errors, ensuring optimal system performance, and enhancing user experience (Hertzum, 2022; Neri, 2023).

2.3. Crisis Management in Software Development

To address technical deficiencies, crisis management literature focuses on the need for swift responses to technical deficiencies. For a given crisis, the decision-making process is described by Nowroozi et al. (2020). Additionally, long-term technical actions should be considered in certain crisis cases to guarantee business continuity for the upcoming planning horizon. This aligns with crisis management strategies outlined in the literature (Stettina et al., 2023). Balancing workload and human resources, and minimizing error rates are crucial aspects of effective crisis response in business-oriented projects (Nicoletti & Padovano, 2019).

2.4. Mental Well-being in Software Development

Addressing the mental well-being of development teams is a contemporary focus related to workplace stress (Bhui et al., 2016). Efforts made by the influencers in the given case study are typical practices to manage work-related stress through psychological counseling and implementing guidelines for rest and recovery. According to Hoge et al. (2005), following such practices is aligned with the growing recognition of mental health in the workplace. Moreover, recognizing and addressing motivational factors are crucial for sustaining team morale and performance (Franco et al., 2002). Indeed, neglecting motivational policies leads to reduced resilience among developers, as discussed in the literature (Kurian & Thomas, 2023).

Therefore, the literature review provides a basis for understanding the contextual factors and challenges associated with crisis management in messaging system development. Drawing from studies on messaging systems, software development, and crisis management, this review lays the foundation for a nuanced analysis of the present experiences and offers valuable insights into effective strategies for managing similar crises in the future.

3. Problem Specification

According to Fig. 1, the corresponding messaging platform consists of five working teams operating cohesively within the framework of Software Development Kit (SDK), Online Streaming, Async, Chat Server, and Talk. Notably, Async in a messaging system is a communication method that allows the sender and the receiver to exchange messages without waiting for each other's availability or response. The output of this collaboration has resulted in the formation of a customized messaging system. In this platform, users can only see the Chat Client section. The chat client is the software component that lets the user exchange messages with others, and that connects with the chat server and the operating system. However, other sections such as the messaging server with large databases, software development kit, Async section, and online streaming are present at the backend. In this regard, the evolutionary trend of the platform's development was advancing according to a well-defined schedule, and through periodic testing and user familiarization efforts, all system components were being tested under load, and potential errors were being tracked. Based on the test results, necessary actions such as optimizing the chat server, improving the user interface, and increasing bandwidth and database were focused and planned.

At that time, and with the precautions taken, the system was ready to accept more than seven hundred and fifty online users; a capacity that was insufficient considering the high volume of company employees. Therefore, a gradual improvement plan for the performance in the long term to accommodate a several-fold increase in simultaneous user activities was on the agenda. During development, occasional small crises occurred, such as the decision of the company to send all deposit/withdrawal text messages through the messaging platform, which slightly disrupted the natural development process but was quickly controlled and addressed.

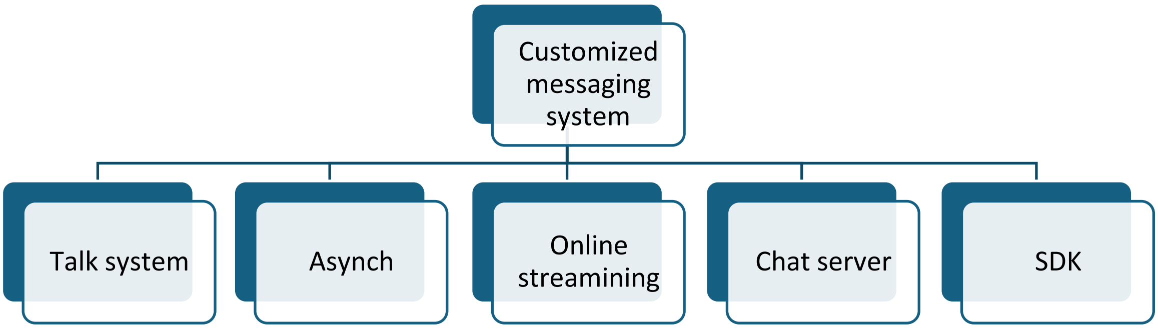

Fig. 1. Structure of customized messaging system

However, a decision to conduct internal organizational transactions through the customized messaging platform suddenly increased the utilization rate of this system among work teams in an escalating manner. A system that was supposed to be ready to accept a large number of company employees over a long-term plan was suddenly faced with a huge influx of requests. Requests that aimed to cover a large volume of the company's business communication space, including sending monitoring alerts, launching the web bank, using the organization's proprietary project management system through the messaging system, in addition to the daily use by employees for membership, group formation, video calls, and the transfer of necessary text and voice messages. Rapid changes and developments plunged the messaging platform development project into crisis. A crisis that, in addition to the expected technical deficiencies in a fledgling software system, burdened the employees involved in the development with excessive work stress and heavy responsibilities.

More precisely, a 300% increase in the number of users within a very short period led to a flood of improvement requests, performance malfunction reports, and error alerts from a multitude of company employees. On the other hand, since, according to the new conditions, internal organizational communications could only be done through the customized messaging channel, the immediate attention and resolution of requests from the perspective of work teams became essential. If the problems were not addressed, the possibility of communication between members of working teams would be disrupted, and as a result, project progress would be delayed. Therefore, company employees tried in every possible way, even informally, such as sending text messages or verbal recommendations to senior management and the system's team members, to convey and ultimately fulfill their requests. Also, the difference in how system issues were transferred to relevant parties compared to the past imposed excessive work stress on the team members.

Users who previously used the corresponding customized messaging system as an auxiliary tool now had a highly critical view of it and expected better performance. In addition to this, from a technical perspective, the sudden increase in the rate of the system's usage revealed infrastructure shortcomings and structural weaknesses in testing approaches. The high volume of work and the reduced resilience of the developers, due to reasons such as an imbalance between the number of requests and the small number of messaging team members, as well as the neglect of motivational policies, were among the problems that, in a chain-like manner, led to the chaotic crisis in the development.

4. Analytical Approach and Problem Solving

Undoubtedly, overcoming the crisis resulting from relying on a one-dimensional approach was not feasible. Considering the analysis and root cause identification, addressing, and managing the chaotic crisis in the development of the messaging system required measures on both technical and mental issues (see Fig. 2). From a technical perspective, solutions were categorized into short-term and long-term. One of the most essential short-term technical solutions adopted was to restrict the way requests were transferred and presented by the employees. The multitude of requests had led to confusion among the members of the messaging team, as the existence of various communication paths for error notification practically prevented the aggregation and prioritization of issues. Therefore, a specific electronic form called the Problem Declaration was designed and defined to be the only means of communication between the employees and the messaging system development team.

After addressing the request aggregation issue, a specific procedure was considered for prioritizing responses to these requests. It was decided that the priority of addressing requests should be classified into five categories: management opinions, critical, business-critical, essential, and normal. Criteria for approaching request resolutions were also defined. For instance, a quick solution with minimal complexity, creating a noticeable change in the messaging system, and focusing on improving common service quality, such as message history and the sequence of messages forming a conversation, were among these criteria. Additionally, a planned approach for the required changes in the chat server and SDK was simultaneously adopted in a short-term timeframe with a one-week interval. Similarly, for the next short-term period, adjustments to the system were made to edit changes within a minimum one-month timeframe.

Furthermore, other solutions were also considered, with a focus on long-term technical actions, given their time-consuming nature. These included relocating the team to the subsidiary office, hiring new personnel, and including testing personnel in the development team. These measures were taken to increase the number of members of the team, balance the workload and human resources, and reduce error rates in new development versions. Alongside technical precautions, crisis management in the development of the messaging system also depended on taking appropriate measures to overcome problems arising from work-related stresses imposed on the team members.

In this regard, efforts were made to address stress management through psychological counseling. Subsequently, by interacting with the human resources unit, guidelines such as providing incentive compensation for performance and mandatory one-day weekend leave for rest and recovery concentration were implemented. As a result of implementing such approaches and injecting both material and spiritual motivations into project contributors, the resilience of the messaging team members was strengthened. Despite the various measures taken to manage the sudden chaos in the development of the customized messaging system, there is no doubt that crises will manifest in different forms in the future. However, relying on the lessons learned in this specific timeframe and understanding how to manage order and stress under high work pressure, it can be ensured that the team recovery process and productivity enhancement will occur at a significantly faster pace.

It is also worthy to imply that the presented approaches to tackle the problem were drawn from a combination of empirical insights gained during the crisis and practical strategies implemented by the development team. While the paper did not explicitly reference specific previous works or external approaches, the presented methods were derived from a thorough examination of the unique challenges faced during the surge in communication demands. The strategies were not merely theoretical but were implemented and tested in response to the crisis scenario. The development team, in collaboration with relevant stakeholders, applied a problem-solving approach grounded in practical experience to address the sudden influx of improvement requests and performance malfunction reports. The effectiveness of these methods was gauged through real-time observations, iterative adjustments, and continuous feedback loops. By emphasizing the tangible outcomes and lessons learned from the crisis management strategies employed, the paper aims to convey the empirical nature of the presented approaches, showcasing their practical utility in navigating the challenges encountered during the development of the messaging system.

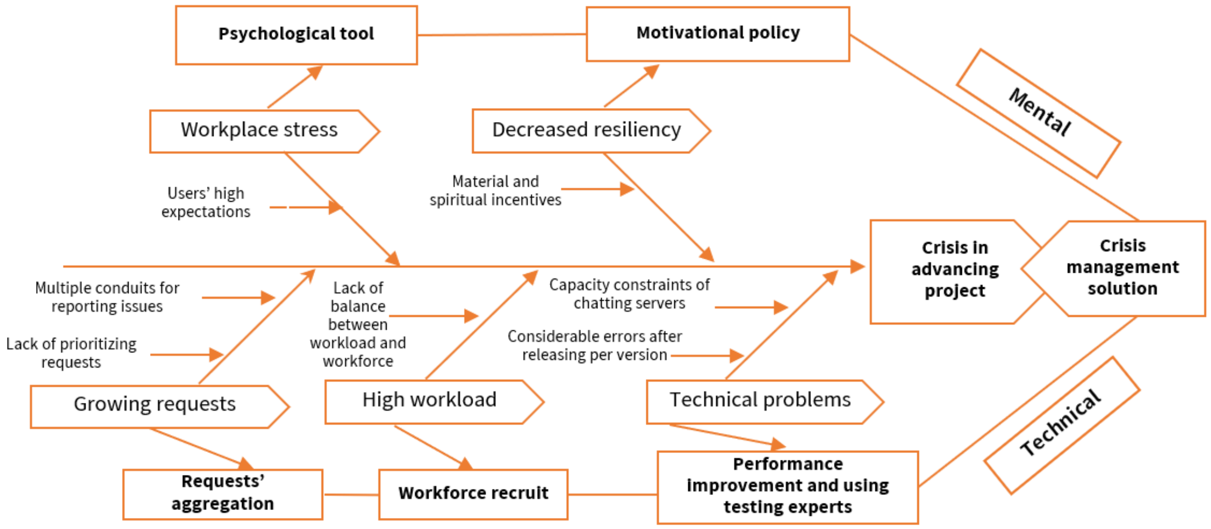
Psychological tool
Motivational policy
Mental
Workplace stress
Decreased resiliency
Users' high expectations
Material and spiritual incentives
Crisis in advancing project
Crisis management solution
Multiple conduits for reporting issues
Lack of prioritizing requests
Lack of balance between workload and workforce
Capacity constraints of chatting servers
Considerable errors after releasing per version
Growing requests
High workload
Technical problems
Technical
Requests' aggregation
Workforce recruit
Performance improvement and using testing experts

Fig. 2. Causes and solutions

5. Conclusion

This paper sheds light on a critical crisis encountered during the development of a customized messaging system, emphasizing the unanticipated surge of users and the subsequent challenges faced by the development team. The study effectively analyzes both the technical and mental aspects of the crisis, providing practical short-term solutions like restricting request channels and proposing long-term actions such as team relocations and staffing adjustments. Furthermore, the paper highlights the crucial need to address the mental well-being of the development team, recognizing the impact of the crisis on stress levels and resilience. The comprehensive approach to crisis management, considering technical deficiencies and team dynamics, contributes valuable remedies for similar scenarios in software development projects.

The advantages of the approach lie in their adaptability to the unique context of the crisis, particularly in the short-term technical measures such as restricting request channels and introducing a Problem Declaration form, which efficiently aggregated and prioritized requests. The categorization of response priorities and planned approaches for system adjustments demonstrated a structured response to the crisis. The long-term technical actions, including team relocations and staffing adjustments, are advantageous for ensuring business continuity and addressing scalability issues. Additionally, the emphasis on mental well-being, with measures like psychological counseling and guidelines for rest and recovery, contributes to team resilience.

However, the proposed solutions may have certain disadvantages that can be discussed in future works. The paper does not extensively discuss the potential drawbacks or challenges that may arise from team relocations, such as disruptions in collaboration or knowledge transfer. The long-term technical actions, while crucial, might require significant time and resources, potentially delaying the overall project timeline. Furthermore, the effectiveness of the Problem Declaration form and the prioritization criteria may depend on the responsiveness of the development team, and future studies could benefit from discussing the potential challenges in implementing these measures in different organizational contexts. Overall, the advantages lie in the nuanced and tailored response to the crisis, while potential disadvantages could arise from the practicalities of implementing long-term technical actions and the adaptability of the proposed short-term solutions in diverse organizational settings.

References

Anders, A. (2016). Team Communication Platforms and Emergent Social Collaboration Practices. *International Journal of Business Communication*, *53*(2), 224-261. https://doi.org/10.1177/2329488415627273

Bannerman, P. L. (2008, 26-28 March 2008). Toward An Integrated Framework of Software Project Threats. 19th Australian Conference on Software Engineering (ASWEC 2008),

Bhui, K., Dinos, S., Galant-Miecznikowska, M., de Jongh, B., & Stansfeld, S. (2016). Perceptions of work stress causes and effective interventions in employees working in public, private and non-governmental organisations: a qualitative study. BJPsych bulletin, 40(6), 318-325.

Franco, L. M., Bennett, S., & Kanfer, R. (2002). Health sector reform and public sector health worker motivation: a conceptual framework. *Social Science & Medicine*, *54*(8), 1255-1266. https://doi.org/https://doi.org/10.1016/S0277-9536(01)00094-6

Harman, M., Jia, Y., Krinke, J., Langdon, W. B., Petke, J., & Zhang, Y. (2014). *Search-based software engineering for software product line engineering: a survey and directions for future work* Proceedings of the 18th International Software Product Line Conference - Volume 1, Florence, Italy. https://doi.org/10.1145/2648511.2648513

Hertzum, M. (2022). *Usability Testing: A Practitioner's Guide to Evaluating the User Experience*. Springer International Publishing. https://books.google.com/books?id=fYlyEAAAQBAJ

Hoge, M. A., Tondora, J., & Marrelli, A. F. (2005). The fundamentals of workforce competency: implications for behavioral health. *Administration and Policy in Mental Health and Mental Health Services Research*, *32*(5), 509-531. https://doi.org/10.1007/s10488-005-3263-1

Kurian, R. M., & Thomas, S. (2023). Importance of positive emotions in software developers' performance: a narrative review. *Theoretical Issues in Ergonomics Science*, *24*(6), 631-645. https://doi.org/10.1080/1463922X.2022.2134483

Luqman, A., Zhang, Q., & Hina, M. (2023). Employees' proactiveness on enterprise social media and social consequences: An integrated perspective of social network and social exchange theories. *Information & Management*, *60*(6), 103843. https://doi.org/https://doi.org/10.1016/j.im.2023.103843

Neri, G. R. (2023). The Use of Exploratory Software Testing in SCRUM. *SIGSOFT Softw. Eng. Notes*, *48*(1), 59–62. https://doi.org/10.1145/3573074.3573089

Nicoletti, L., & Padovano, A. (2019). Human factors in occupational health and safety 4.0: a cross-sectional correlation study of workload, stress, and outcomes of an industrial emergency response. *International Journal of Simulation and Process Modelling*, *14*(2), 178-195. https://doi.org/10.1504/IJSPM.2019.099912

Niemimaa, M. (2023). Evaluating compliance for organizational information security and business continuity: three strata of ventriloqual agency. *Information Technology & People*, *ahead-of-print*(ahead-of-print). https://doi.org/10.1108/ITP-03-2022-0156

Nowroozi, A., Teymoori, P., Ramezanifarkhani, T., Besharati, M. R., & Izadi, M. (2020). A Crisis Situations Decision-Making Systems Software Development Process With Rescue Experiences. *IEEE Access*, *8*, 59599-59617. https://doi.org/10.1109/ACCESS.2020.2981789

Stettina, C. J., Garbajosa, J., & Kruchten, P. (2023). *Agile Processes in Software Engineering and Extreme Programming: 24th International Conference on Agile Software Development, XP 2023, Amsterdam, The Netherlands, June 13–16, 2023, Proceedings*. Springer Nature Switzerland. https://books.google.com/books?id=pcu_EAAAQBAJ

Wadleigh, K. R., & Crawford, I. L. (2000). *Software Optimization for High-performance Computing*. Prentice Hall PTR. https://books.google.com/books?id=IRN0IEXJzKEC

Yan, X., & Feng, Y. (2023). Mandating interoperability of interpersonal communications services in an ecosystem competition context. *Computer Law & Security Review*, *50*, 105850. https://doi.org/https://doi.org/10.1016/j.clsr.2023.105850

Yang, J., Chen, Y.-L., Por, L. Y., & Ku, C. S. (2023). A Systematic Literature Review of Information Security in Chatbots. *Applied Sciences*, *13*(11).