




Designing a Helpdesk Process Model for Post-Release Software Maintenance Using Soft Systems Methodology



Arfive Gandhi^{*}, Dana Sulistiyo Kusumo^{*}, Nungki Selviandro^{*}

School of Computing, Telkom University, Bandung 40257, Indonesia

Corresponding Author Email: arfivegandhi@telkomuniversity.ac.id

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.310217>

ABSTRACT

Received: 28 September 2025

Revised: 10 December 2025

Accepted: 18 February 2026

Available online: 28 February 2026

Keywords:

helpdesk process model, Soft Systems Methodology, post-release software maintenance, complaint handling, stakeholder analysis

The New Membership Application (NMA) software, developed by XYZ Corp. for ABC Corp., faced significant bugs post-release due to unstable business processes, misinterpreted requirements, and frequent changes. As a result, establishing a robust helpdesk process model is essential for ensuring the continuity of the software's maintenance post-release. This study designs a comprehensive helpdesk model for the NMA software maintenance process using a qualitative approach grounded in Soft Systems Methodology (SSM). The research follows the seven steps of SSM, including artifact generation, system modeling, and improvement strategies. Key stages involved root cause analysis, CATWOE, and PQR analyses, which guided the development of the initial helpdesk process model. The model was refined through iterative validation, with 10 tactical plans developed for its implementation. The resulting model addresses critical business process issues, outlines the roles of stakeholders, and includes necessary artifacts and documentation. This framework provides a structured approach for minimizing failure risks and enhancing the efficiency of post-release software maintenance. By combining SSM's flexibility with practical strategies, this research offers an actionable and validated model for managing software maintenance in complex environments. It contributes to the field by providing a reliable solution for helpdesk management, ultimately improving stakeholder communication and overall project success.

1. INTRODUCTION

Software quality is essential for project management in software engineering context. Producing the quality product (in this context, “software”) has become a prominent factor to succeed in the current competitive world [1]. From many different standards, Colakuglu et al. [1] always mentioned these criteria: maintainability, freedom of risk, security, reliability, and functional suitability. Ideally, software development teams should fulfil those criteria, but more complex software could threaten their fulfilment and increase the opportunities of bugs existence.

The New Membership Application (NMA) is software owned by ABC Corp., which XYZ Corp. developed in 2022. This NMA accommodates the registration process for ABC Corp.'s new members, including the signup, membership scheme, and payment processes. However, the NMA repeatedly got bugs that triggered complaints from ABC Corp, especially from software operators and applicants. Bugs appeared due to several factors, including unstable business processes, misinterpretation of requirement gathering, and numerous changes made after the software release. Business process instability made the source code and application structure prone to changes and triggered new bugs. Moreover, some bugs were not adequately anticipated, so they appear in the NMA software, even as repetitive cases. It indicated the

importance of an agile and qualified helpdesk process model. Sporadic complaint bug handling with a lack of communication made the software developer in XYZ Corp. miss the bug-fixing processes. Those bugs caused complaints on ABC Corp., so the NMA' and XYZ Corp.'s credibility were threatened. Furthermore, those complex situations pushed XYZ Corp. to face new requests (out of the requirement scope as defined in the software design) since its credibility was politically threatened. Therefore, NMA software that has entered the maintenance phase should undergo massive repairs and redevelopment. Van Herck and Vangehuchten underpinned the capability of complaint handling as a key differentiator for corporate in the competitive market [2]. They emphasized the corporate should track each received complaints to ensure the issue solving although they had massive and complex content. In another literature, Aboalghanam et al. [3] hinted the better service quality and customer retention as the impacts against the corporate's ability to response any complaints. More, Binboga and Gumussoy found the customer factors as strongest predictor of process efficiency, sustainable software product quality and stakeholder satisfaction [4].

The existence of a mature helpdesk process model is an urgent requirement to maintain the continuity of the NMA software maintenance process. The helpdesk process model allows incoming complaints to be recorded, analyzed,

distributed, and followed up effectively and efficiently. The agility of XYZ Corp.'s software development teams in operating the helpdesk process model should enable them to withstand the maintenance phase's rhythm, allowing them to focus on improving the application rather than developing new features in response to new needs. In addition, the helpdesk process model serves as a professional user interface for XYZ Corp., handling NMA users' complaints efficiently and maintaining their satisfaction. It leads to customer satisfaction through service quality plays a pivotal role as recalled by Balinado et al. in 2021 [5]. Therefore, this study formulated two research questions:

- [RQ1] What helpdesk process model can be designated for NMA software in complaint handling of NMA software project?
- [RQ2] What strategies to implement helpdesk process model in complain handling of NMA software project?

Given the complexity of the phenomenon and the stakeholders' idealistic expectations for a quick solution, this research employed the Soft System Methodology (SSM) as its primary approach. This selection argued with its capabilities to elaborate on the proposed idealistic solution and available realistic constraints. SSM enables model designing processes and delivers the necessary strategies for their implementation [6]. Also, SSM had various success stories in many areas [7-9], such as information system development, project management, performance management, and knowledge management. Nakakawa et al. [10] also recognized SSM as the most appropriate approach for enabling rational or systemic thinking. These areas were aligned with this research since the selected case study had a significant and complex problem related to project management of information systems development, especially software development, performance maintenance, and the project team's knowledge utilization.

This study aimed to design a helpdesk process model at PT XYZ, focusing on NMA software. From a practical standpoint, the helpdesk process model becomes the standard for helpdesk services in various other applications. XYZ Corp. can set it as a benchmark for organizational capabilities and a helpdesk for different software. This study employed the SSM to design solutions for complex situations, given the numerous unique and problematic issues encountered in this software project with limited resources.

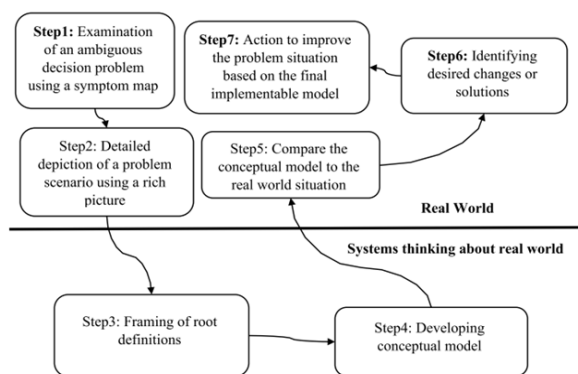


Figure 1. The steps in Soft System Methodology (SSM) [6]

This study offers several essential novelties. First, it expands the use of SSM in the software engineering context. Compared to prior work that predominantly uses the SSM for

requirements engineering and software modeling, this study extends its use to the maintenance phase, an underexplored area. According to the SWEBoK V4 [11], the maintenance phase involves corrective efforts for faults and latent defects. By adopting SSM, this study addresses a theoretical and practical gap by examining socio-technological issues more systematically. Second, this study offers novelty in project management, which has been dominated by planning and execution stages. Most project teams (including the software project team) lacked effective conflict and complaint management, especially during post-release stages. This study's findings will encourage more holistic software project management through sustainable governance. Third, this study targeted the output until strategies formulation, not only conceptual modelling. It motivates researchers to implement the SSM holistically in seven steps, engaging stakeholders' feedback rather than relying solely on the researchers' point of view.

2. RELATED WORKS

Developed by Checkland and Pouter [6], SSM has become a mature, rigorous, well-established methodology. Many researchers have applied it in various contexts, disciplines, and countries, relying on resilient principles [6]. The SSM processes launch the exploration of problematic situations through issue identification and dynamic factors analysis, such as the culture and power relations [6]. SSM actualizes a qualitative research technique by stressing the meaning, self-reflection, interpretation, human experience, learning, and engagement based on thinking systems [12]. It emphasizes the worthiness and feasibility of the proposed changes, particularly highlighting cultural feasibility as the most critical and dynamic factor [13]. More, Hindle [9] reminded us to realize that real-world projects will be open-ended, so that SSM adoption will require iteration between stages.

When implementing it, the SSM users relied on the modeling language of SSM due to its flexibility and powerful conceptual language to support and document the thinking process. Surely, the system modeling used should be in a common analytical approach within system thinking and management science. It also should be flexible and cover the relatively abstract concept of a "purposeful activity system" [9]. However, the modeling generation should involve the related stakeholders, rather than relying on the researchers' subjective ideas. It analyses unstructured problems using techniques including symptom maps, rich images, root cause analysis, root definition, final implementable models, and conceptual models, emphasizing systemic thinking and addressing underlying issues [14]. Currently, contemporary SSM has a large, multipurpose, and flexible methodology in various ways by practitioners and academics [9].

The SSM implementation enables simultaneously visualization to cover many worldviews and facilitates the exchange of views [15]. Simply, it comprises the following stages: (1) Comparison of models with "reality", (2) Defining desired changes. Examination of feasibility, (3) Actions. Figure 1 visualizes the steps of SSM. Cezarino et al. [16] revealed these fundamental baselines of SSM:

- Identifying the occurred problem or situation.
- Building a structure model from a system through investigation by providing pictures and involving related facts, especially issues that worry people.

- The systems should cover these main elements (but not limited to): client, actors, desired transformation, organizational world view, owners, and environmental constraints.
- Designing the conceptual models of the ideal situation for each of the elements required.
- Shifting from the system to the real world by repeating the comparison between the second and fourth steps. It leads to a decision with appropriate argumentation and adjustment. Also, it will expose the organizational ability to adapt and construct the basis of discussion and debate that ought to lead to eventual consensus.
- Considering the organizational cultures and including them in the feasibility study.

In their experiment, Damenu and Beaumont [17] found that SSM was effective for analyzing the ill-defined and socially complex problem situations that require systems thinking. It had strong suitability for conducting the analysis, presenting the discussion, and engaging with stakeholders towards the alignment with effective approaches for assessing culture. Therefore, the current states and challenges in a specific phenomenon can be effectively addressed [17]. Augustsson et al. [7] revealed that most of SSM implementation had aimed at problem structuring and proposing improvements (40.8%), problem structuring (16.3%), and evaluation (12.2%). They also underlined its purpose for policy, knowledge management, and teamwork improvement. This methodology guides the researchers to generate a conceptual model as a visual representation that simplifies and clarifies the linkages and elements of the concrete problem and the proposed solution for easier, and understandable situations [14]. Not only enable researchers to identify issues, SSM implementation also allows the stakeholders to present their perspectives on the multiple and overlapping issues that they face by illustrating the reality of multiple, valid perspectives that various individuals hold clearly [18]. The investigation in SSM processes further emphasized how researchers can gain valuable insight into a study context by engaging in analysis of the roles, norms, values, power, culture, and politics of the problem situation [18]. These indicators signalize the promising benefits and success stories as supporting reasons to utilize SSM for evaluating the existing helpdesk process model and proposing an improved one.

In the middle of SSM implementation, the researchers adopt the CATWOE as a mnemonic reminder to accommodate this information [6, 7]:

- Customers: the entities that are affected by the problematic situation and the improved resolution.
- Actors: the entities who directly engaged in the situation and in performing the improved resolution.
- Transformation: the change process.
- Worldview: the fundamental belief that underlies the improved resolution is worthwhile and important.
- Owners: the actors who are responsible for the improved resolution and who authorize the decided actions.
- Environmental constraints and enablers: the external factors that may influence the problematic situation and the improved resolution.

After filling in the CATWOE elements, SSM will guide the root definition using the PQR formula by answering the questions: what should be done (P), how it should be done (Q), and why it should be done (R). Augustsson et al. [7] also

reminded Five E's as criteria for outcome assessment against the proposed resolutions: (1) Efficacy, (2) Efficiency, (3) Effectiveness, (4) Ethicality, and (5) Elegance.

3. METHODOLOGY

3.1 Research classification

This research deployed the classification as introduced by Thornhill et al. [19] in their Research Onion (see Figure 2). In the first layer, related to philosophy, this research adopts an interpretivist approach, emphasizing the subjective understanding of phenomena in case studies. Also, the fact collection processes from every stakeholder indicate interpretivism. Later, this research applies the deductive approach, following the basic theory of SSM with collected and interpreted data to generate a model of a helpdesk process model. In the Strategy layer, this research was classified as case study research since it brought a specific case of a chaotic software project maintain phase. It also took the Mono-Method category in Choice later, especially contextual data as a baseline for the qualitative approach. In the time horizon layer, this study picked the Cross-sectional considering the research results were captured at a particular time as a single snapshot. Last, this research relied on the techniques of interview, observation, and statistical analysis as part of the sixth layer.

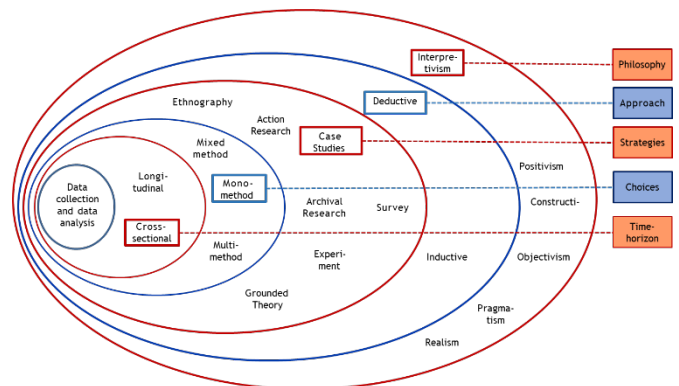


Figure 2. Research classification (adopted from [19])

3.2 Research phases

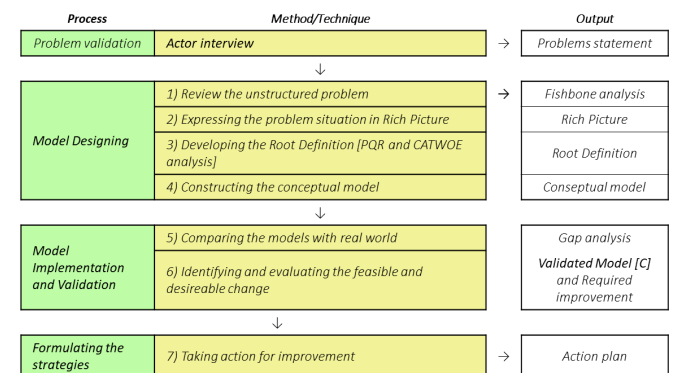


Figure 3. Research stages

Figure 3 describes a series of research stages, including input, process, and output information at each stage. Considering that this research is a case study-based and

applied type of research, all data is perceived from research objects: parties who are stakeholders in implementing the NMA software. The data was collected using interview techniques, focus group discussions, and field observations. Parties involved in the interview and FGD process included XYZ Corp. as the software house and ABC Corp. as the client and user. This research recruited the following entities representing XYZ Corp.: software developer, project manager, and the XYZ Corp board of directors. Meanwhile, entities representing ABC Corp. are the operators of recruiting new members and registrants/applicants for membership at ABC Corp. Also, the observations were performed before and after the helpdesk process model was implemented. This research conducts triangulation to confirm findings and evidence to reduce subjectivity.

4. RESULTS AND ANALYSIS

4.1 Model designing: 1st to 4th step of Soft System Methodology

In step 1 of SSM, this research focused on defining the problems. It conducted interviews and observations on related evidence to uncover facts and their relationships. This research highlighted the lack of a complaint handling system that occurred in the NMA project. Ideally, the qualified helpdesk provides three key benefits: resolving and following up on issues, enhancing software quality assurance, and improving stakeholder satisfaction. In contrast, the existing helpdesk process model had three negative impacts: uncontrolled issue handling, poor software quality assurance, and low stakeholder satisfaction. This research decomposed the causative analysis into four aspects: People, Data, Processes, and Technology (refer to the four elements of information systems). Through those aspects, this article identified the relevant triggers and targeted them as root causes (see Figure 4) that need practical solutions towards qualified helpdesk process model.

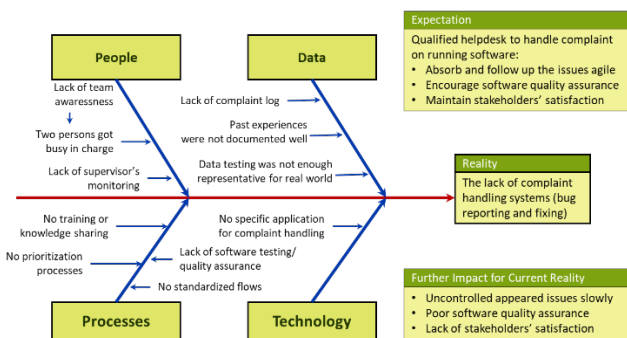


Figure 4. Root cause analysis using fishbone diagram

Step 2 of SSM was depicting or visualizing the problem by including the actors along with the chronology of the problem. Figure 5 shows a Rich Picture of the existing situation, representing the process of software users' complaints (version 0 of helpdesk process model). Components in Rich Picture were visualized by identifying the real stakeholders during project implementation. The Applicant discovered a software failure and contacted the Operator, the initial point of contact for the NMA applicant. Applicants believe that a lengthy repair process could prompt them to cancel their membership

registration at ABC Corp. Meanwhile, Operators felt anxiety because they had repeatedly received complaints from Applicants that the software still contained bugs after being released. This anxiety stemmed from decreased credibility and performance indicators, primarily due to the software development team is slow and unresponsive bug fixes.

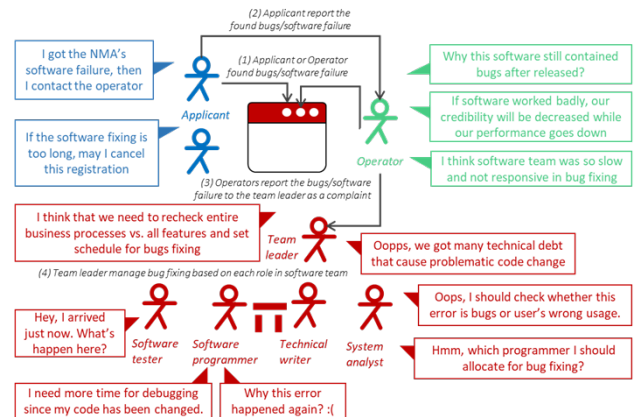


Figure 5. Rich picture from the existing situation

The red entity illustrates a software developer consisting of a Team Leader, System Analyst, Technical Writers, Software Programmers, and Software Testers. The team leader and a System Analyst became the Operator's contact person for information on occurring bugs. Then, they allocated a suitable programmer to handle bug fixing after initially diagnosing the cause of the error. In some cases, the complaints were not caused by software bugs only, but also by software usage that did not follow the business processes or new requirements. Software programmers assigned to handle bug fixing certainly experience the challenge of ensuring that source code improvements do not affect features already working well but only fix errors that occur. Finally, software testers who had just joined when the situation was critical needed clarification about what was happening here. Software testers must learn business processes to prepare a stock of test cases for quality assurance. This situation aligned with Pargaonkar's opinion about organizational challenges in allocating the necessary human resources for comprehensive testing, particularly for large-scale or time sensitive projects [20].

On the other hand, three technical writers had a relatively small workload compared to different roles in the software development team. Due to fluctuating issue quantities and repetitive cases, operators escalated these issues to project sponsors. Therefore, the situation was tense, and the Project Manager should intervene to control the conflicts. This situation could cause project sponsors' and users' dissatisfaction with software adoption.

The next step is to define the root of the problem based on the visualization and description provided in steps 1 and 2. In step 3 of SSM, defining the root of the problem utilizes CATWOE and PQR analysis techniques. CATWOE analysis aims to describe the actors in the system to be built, while PQR analysis will supply an overview of how the system works. This research deployed the CATWOE (see Table 1) and PQR analysis components through an interview process and observation of expected conditions in the future.

In step 4 of SSM, the result of combining CATWOE and PQR analysis techniques is the Root Definition of the system to be built (in this context, the design of the NMA software

helpdesk process model). The Root Definition is: A system owned by the software house, especially product owner (O), and implemented by the helpdesk, product owner, and other software developer team in XYZ Corp.(A), to do modelling the helpdesk process model in NMA software during after-release and maintenance period(P) by build effective communication mechanism for complaint issuance, actively bugs monitoring, and implement risk management during bug fixing(Q) to customer: Applicants in ABC Corp. and NMA operators (C) to achieve controlled users' complaints and their satisfaction(R) within the constraint Limited funding, limited human resources, some remote software developer, different corporate's culture, risky reputation(E).

Table 1. CATWOE components in Soft System Methodology (SSM)

Element	Description on Software Project Context
CATWOE Analysis	
Customer (C): Who receives the impact of activities?	Applicants for ABC Corp. and NMA operators. (1) Helpdesk: they who receive, distribute, and update the complaint status.
Actor (A): Who implements the activities?	(2) Product owner: someone who leads software development and maintenance. (3) XYZ Corp.: an institution that develop software (1) Scattered or sporadic communication style for complaint handling during software release and maintenance.
Transformation (T): What should be changed to convert inputs into outputs?	(2) Qualified and standardized helpdesk process model to handle complaints as mandatory service. Qualified helpdesk process model can encourage software development teams to trace complaints, engagement with users, and project stakeholder satisfaction and trustiness.
Worldview (W): Which way of views able to stop the activities?	The software house, especially product owner.
Owner (O): Who can stop the activities?	Limited funding, limited human resources, some remote software developer, different corporate's culture, risky reputation.
Environment Constraint (E): What obstacles are found within the environment?	
PQR Analysis	
(P) what it does	Modeling the helpdesk process in NMA software during after-release and maintenance period. Build effective communication mechanism for complaint issuance, actively bug monitoring, and implement risk management during bug fixing.
(Q) how it is done	Controlled users' complaints and their satisfaction.
(R) reason for the activity	

Based on the CATWOE definition and PQR analysis results, this research formulates a helpdesk handling flow in version 1. Version 1 represents the initial step, which will

gather feedback from relevant stakeholders to ensure the readiness and availability of competencies. The helpdesk process model generally consists of five steps, visualized in Figure A1, involving five roles. The first two roles are Technical Writer and System Analyst, who receive information regarding complaints about bugs experienced by users (Applicants or Operators). Next, the Technical Writer and System Analyst inform the Product Owner (PO) and Software Programmer (SP) about the issue to be addressed. After being fixed by the SP, the Software Tester (ST) performs validation and verification in the Development and Staging environments. The SP escalates the software change to Production if the validation and verification are passed.

4.2 Model implementation and validation: 5th to 6th step of Soft System Methodology

Following step 5 of SSM, the developed helpdesk flow design (Version 0, Figure A1 in Appendix) received feedback from the stakeholders in this software development project, including the Board of Directors of the software house and the internal software team. Several practical inputs became considerations to decompose the activities into more detailed ones, even adding several activities. For example, TW and SA have received complaints several times from users who believe they have encountered bugs, but these issues were due to incorrect usage. It indicates that each report should be checked to see whether the incoming report is a bug or a human error from the user's side. Also, this research highlighted that complaints are often repetitive. These repetitive incidents require the helpdesk to recall how similar cases have been handled in the past. On the other hand, recurring cases (if they are indeed bugs) should receive more attention to prevent them from happening again.

XYZ Corp's top management also provided feedback as input for improvement in the current flow. The first input pertains to the time estimate or SLA (Service Level Agreement) that requires definition as a work reference. Each bug handled must have an estimated completion time considering the severity and ongoing workload. It ensures the team has a definitive target that encourages effective and efficient work. Moreover, the estimated work time is based on handling similar bugs or those of the same scale, so that the team has a regular and measurable work culture. In addition, top management provided input to encourage regular progress reports on bug handling to APL and OPE, even if the software team had not yet fully resolved the bugs. It leads to transparency and more honest interactions between the project team and APL and OPE. Top management expects the team to report progress earlier, rather than OPE or APL, who first ask about the progress of their complaints. Based on bug handling, top management requires a supply of a recap or overview of the quantity and severity of bugs handled by the team. It aims to build a positive image of XYZ Corp. in front of ABC Corp. as a client. By relying on quantitative and transparent data, top management hopes that ABC Corp. will gain positive impressions of the improved software and the team's performance during implementation, thereby ensuring project continuity and even increasing the chances of implementing subsequent software.

After those several iterations in step 5, this research came to the step 6 of SSM. By considering feedback and many best practices in various other projects, the author developed Version 2 of the helpdesk process model, as shown in Figure

A2. In version 2, the role responsible for handling all reports and complaints is the Technical Writer (TW), who assesses reports based on three criteria:

- bug identification.
- identification of recurring cases.
- initial estimation of the severity.

The first criterion aims to distinguish between complaints that are bugs and those that are users' mistakes when using the software. After several months of receiving complaints, some were not bugs, but they were handled as if they were bugs, thereby consuming work time. Mistakes in software usage should not be recorded as bugs to confirm that the software has minimal bugs (to maintain the software's quality and reputation). However, their mistakes serve as input for improving user experience, thereby minimizing the likelihood of similar mistakes in the future. Related to second criteria, checking whether the error is a repeated case (as recorded in the catalog) aims to speed up handling based on previous experience and provide input for the PO to formulate preventive action in the future. In the last criterion, the team can estimate earlier how significant the impact that the bug brings since it will influence the required time, required resources, and risk of failure. If the severity is too high, the team should notify the XYZ top management to obtain necessary support and approval for any risky handling.

As shown in Figure A2 (Appendix), PO and SA perform a bug diagnosis in the Production environment while formulating a handling strategy involving the Software Programmers and a targeted timeline. This research confirms the importance of implementing the C-P-D principles: Corrective, Preventive, and Detective. Implementing these principles manifests the concept of risk management, which emphasizes the need to handle risks (software errors in this case) and prioritizes continuous improvement. As reminded by Bauskar et al. [21], risk management is essential and relevant for every project that seeks to avoid and suppress unanticipated costs, basically calling for pre-emptive action. Elokby et al. [22] also underlined its importance to achieve the success of project as whole. Continuous improvements are achieved by correcting errors that occur (in source code and database), preventing similar cases, and installing an error detection system to identify risks early. Prevention and detection efforts also involve improving venues and user satisfaction to minimize errors, increase team agility, and strengthen software quality.

TW utilizes two layers of communication media. Layer 1 communication is WhatsApp, intended for APL and OPE to interact with TW. To facilitate coordination with OPE staff at ABC Corp, TW also created a dedicated WhatsApp Group (WAG) with them. Layer 1 aims to define the integrated interface of the official helpdesk process model's window. Layer 2 communication is a dedicated WAG for all team members, led by the PO. This second-layer system was derived from an evaluation of the previous model, which did not define an effective and documented communication process within the internal team. Through this WAG, incoming complaints can be addressed more quickly, including an initial diagnosis by the SA and related responses from programmers.

Every time an anomaly occurs, TW will identify whether it is a bug or not. If the anomalous condition is not a bug, the TW will provide corrective instructions to the users (OPE or APL) so they can return to proper use. If it is a real bug, TW will report it to the team through WAG and record it in bug

records with adequate evidence. WAG usage also contributes to transparency in the distribution of bug fixing work since the bugs are related to specific features and specific skill sets. Although bug reports are still ongoing at the beginning, this helpdesk process model encourages software testers to develop test cases based on the complaints. In the previous flow, software testers had no specific workload until the programmer submitted a fix. Time-consuming test case creation can be pushed forward with the bug fix, allowing software testers to immediately execute the fix once the programmer notifies them that it is complete.

Table 2. Summary of formulated strategies in 7th step

Plans	Targeted Time	Target Deliverable	Person in Charge (PIC)
Aspect: Data			
Centralizing the log recording	Week 1	Centralized log recording	TW
Improving the test case generation	Week 2	Test cases	ST
Building the learning lesson	During implementation	Retrospective catalog	PO
Aspect: People			
Distributing the job more transparency	During implementation	Job measurement	PO and SA
Monitoring the bugs fixing through weekly report	During implementation	Weekly report for the Board of Directors	PO
Determining the targeted time for received complaint	During implementation	Targeted time in bug logbook	PO and SA
Aspect: Processes			
Training and simulation	Week 1	Tacit and implicit knowledge	All members
Defining the prioritization of bugs fixing	During implementation	Priority label in bug logbook	PO and SA
Aspect: Technology			
Utilizing the mature platform for recording and communication	Week 1	Adopted platforms	TW
Ensuring the data protection	Week 2	Tested data management	TW

This research identified several significant differences between the proposed design (Version 2, Figure A2 in Appendix) and the previous flow (Version 1, Figure A1 in Appendix) in complaint handling. The proposed design allocates TW and SA as frontlines that receive complaints and are the initial interface for the software team to application users. Meanwhile, the previous version still allowed users to contact the software team members sporadically. Following the bugs as informed by TW and SA in the WAG, the SP perform the bug fixing. The WAG usage considers the WhatsApp application's price, which is free and commonly used by people in Indonesia with good usability, including the user's technological literacy profile and age. However, the

WAG has a general purpose for communication, so it complicates the documentation and repository processes. To anticipate that limitation, the PO provides complaint and bug-reporting records filled in by the TW, called as Complaint Book. It consisted of several parts as follows:

- List of received complaints, including timestamp, evidence, and initial bug identification
- List of bugs that occurred, including timestamp, estimated time, diagnosis, priority, and person in charge (PIC)
- List of bug handling, including timestamp, taken actions, and evidence

The record-filling processes are currently limited to TW for new entries and SP for development progress information, without providing access to filling for application users (APL or OPE). Their inability to access the complaint and bug reporting records aims to prevent sensitive information related to application vulnerabilities, including preventing irresponsible persons who might exploit these vulnerabilities. Ideally, the software teams have specific applications to accommodate the recording processes as detailed before. However, the software teams can still rely on existing mature platforms, such as Excel or Spreadsheet, by adjusting the security and usability aspects.

4.3 Formulating the strategies: 7th step of Soft System Methodology

In step 7 of SSM, this research focused on formulating tactical plans to implement the developed helpdesk model system. The tactical plans cover these items: necessary plan, targeted time, targeted products or services, required input, and person(s) in charge (PIC). These items should be defined to ensure the appropriateness of designed tactical plans and reduce the failure chances. The tactical plans also accommodate four aspects as mentioned in the root cause analysis (see Figure 4): Data, People, Process, and Technology. Table 2 unveils the formulated tactical plans as the necessary strategies. By formulating these strategies, the helpdesk process model would be more prepared for implementation readiness.

5. DISCUSSION

5.1 Theoretical implications

This research has successfully extended the successful application of subsystem methodology theory to the context of software engineering theory. It aligns with the research results from Luong et al. [8]. If they had created a framework for health domain, this research generated helpdesk process model, including the workflow, for software project domain. Similar artifacts were also successfully produced in this study, with certain specifications for the helpdesk model development case. The existence of SLA as targeted execution time was also aligned with research about helpdesk process model by Setyowibowo et al. [23]. Besides software engineering, another theory successfully incorporated in this research is project management knowledge. Project management theory, which previously focused solely on the environmental, time, and cost domains, has now been expanded to encompass environmental, scheduling, financing, and stakeholder satisfaction. This research supports the effort

to realize a project that not only successfully builds a product that meets the engineering requirements but also achieves stakeholder satisfaction through the provision of implementation services, particularly related to management, specifically related to the management of complaints arising from bugs in the software product resulting from this project. Interestingly, the theory of soft systems methodology itself is a long-standing theory with various implementation variations. The results of this study indirectly extend the relevance of the method, which remains applicable in the current era. This research contributes to the successful maintenance of old theories that can still be applied with formal adjustments.

5.2 Practical implications

The results of this study have several practical implications. First, the helpdesk model developed provides short-term and long-term solutions for software development companies to provide helpdesk services capable of handling user complaints. Furthermore, the helpdesk process model has capability to accommodate input from various stakeholders, prioritizing software quality and user satisfaction. Furthermore, the developed helpdesk model can be replicated across various projects with detailed adjustments. Hence, this research results fulfill the replication and scalability aspects. Moreover, the results of this study serve as a reference for developing several standards, procedures, and work instructions that are adequate and aimed at meeting project requirements. Thus, the general foundation for project management prioritizes customer satisfaction, alongside other key project quality dimensions: scope, schedule, and cost.

6. CONCLUSION

This research has successfully achieved the targets set out in the first and second research questions. In relation to the first research question, this research has produced a helpdesk model based on the identification of business process problems, the roles of each stakeholder, and the necessary artifacts and documents. The problem identification and definition were addressed through a root cause analysis approach and the CATWOE framework. By relying on an iterative process of developing the proposed helpdesk model, this research successfully developed a helpdesk model for a software development project. Meanwhile, this research solved the second research question by formulating several strategies or tactical plans aimed at optimizing the developed helpdesk model and minimizing the risk of failure. Through these strategies, the NMA development team and the XYZ company were able to implement this helpdesk model effectively. In addition, this research's results can be replicated to achieve scalability in various other software projects.

7. RECOMMENDATION

This study also has several limitations, particularly regarding collecting quantitative data when testing the design's effectiveness and efficiency. This study recommends further research for quantitative tests by considering several variables: the duration resulting from the efficiency of complaint handling and the reduction of repetitive bugs. In addition, this

study recommends further research, specifically a feasibility test of the helpdesk process model when replicated for other software cases. It will ensure the maturation of the helpdesk process model as part of the mandatory service for the software house. Related to software quality aspects, this research also suggests further research to decompose the strategies towards each aspect (such as usability, performance, and security) as specific KPI and targeted programs. Hence, the helpdesk process model can support software development team achieves the qualified project maintenance period.

ACKNOWLEDGMENT

This work was supported by the Bandung Techno Park, Telkom University, as business incubation and commercialization unit who facilitate the case study. This work was also supported by Directorate of Research and Community Service, Telkom University, under the Basic Internal Grant "Navigating the Reliable Product on Empirical Software Engineering Project" in fiscal year 2025.

REFERENCES

- [1] Colakoglu, F.N., Yazici, A., Mishra, A. (2021). Software product quality metrics: A systematic mapping study. *IEEE Access*, 9: 44647-44670. <https://doi.org/10.1109/ACCESS.2021.3054730>
- [2] Van Herck, R.E., Vangehuchten, L. (2024). Unpacking the art of customer complaint handling in Spanish and British telecom emails: A cross-cultural webcare study with a human touch. *International Journal of Business Communication*, 61(1): 115-147. <https://doi.org/10.1177/23294884231201142>
- [3] Aboalghanam, K.M., Alzghoul, A., Alhanatleh, H. (2024). An analysis of service quality and complaint handling in the Jordanian healthcare sector: Implications for TQM and customer retention. *Innovative Marketing*, 20(1): 51-65. [https://doi.org/10.21511/im.20\(1\).2024.05](https://doi.org/10.21511/im.20(1).2024.05)
- [4] Binboga, B., Gumussoy, C.A. (2024). Factors affecting agile software project success. *IEEE Access*, 12: 95613-95633. <https://doi.org/10.1109/ACCESS.2024.3384410>
- [5] Balinado, J.R., Prasetyo, Y.T., Young, M.N., Persada, S.F., Miraja, B.A., Redi, A.A.N.P. (2021). The effect of service quality on customer satisfaction in an automotive after-sales service. *Journal of Open Innovation Technology Market and Complexity*, 7(2): 116. <https://doi.org/10.3390/joitmc7020116>
- [6] Checkland, P., Poulter, J. (2006). *Learning for Action: A Short Definitive Account of Soft Systems Methodology and its Use for Practitioners, Teachers and Students*. Chichester: Wiley.
- [7] Augustsson, H., Churruca, K., Braithwaite, J. (2020). Change and improvement 50 years in the making: A scoping review of the use of soft systems methodology in healthcare. *BMC Health Services Research*, 20: 1063. <https://doi.org/10.1186/s12913-020-05929-5>
- [8] Luong, T.T., Huynh, V.N., Kim, E. (2022). A hybrid use of soft systems methodology for developing a framework of evidence-based teaching for hospitality and tourism instructors in Vietnam. *Systemic Practice and Action Research*, 36: 241-274. <https://doi.org/10.1007/s11213-022-09609-9>
- [9] Hindle, G. (2023). Introduction to soft systems methodology. *Journal of Systems Thinking*, 3(1): 1-13. <https://doi.org/10.54120/jost.000008>
- [10] Nakakawa, A., de Manuel Keenoy, E., Zabala, A.F., Berastegui, D.V., Suarez, J.T. (2023). Strategic objectives for aligning healthcare and IT practices in providing integrated care for multimorbid patients: A soft systems methodology perspective. *Systemic Practice and Action Research*, 37: 59-80. <https://doi.org/10.1007/s11213-023-09643-1>
- [11] Washizaki, H. (2024). *Guide to the Software Engineering Body of Knowledge v4.0*. IEEE Computer Society.
- [12] Jackson, M.C. (2003). *Systems Thinking: Creative Holism for Managers*. Chichester: Wiley.
- [13] Susanty, A., Purwanggono, B., Faruq, C.A. (2022). Electricity distribution efficiency analysis using data envelopment analysis (DEA) and soft system methodology. *Procedia Computer Science*, 203: 342-349. <https://doi.org/10.1016/j.procs.2022.07.043>
- [14] Kumar, V., Vrat, P., Shankar, R. (2024). Managing implementation issues in industry 4.0: A soft systems methodology-based approach. *Systemic Practice and Action Research*, 37: 1069-1100. <https://doi.org/10.1007/s11213-024-09686-y>
- [15] Markopoulos, D., Tsolakidis, A., Karanikolas, N., Marinagi, A., Skourlas, C. (2024). Applying soft system methodology for a clearer understanding of the future intensive care units. In *Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics*, Lamia, Greece, pp. 163-170. <https://doi.org/10.1145/3635059.3635084>
- [16] Cezarino, L.O., Liboni, L.B., Oliveira, M.F., Caldana, A.C.F. (2015). Soft systems methodology and interdisciplinarity in management education. *Systems Research and Behavioral Science*, 33(2): 278-288. <https://doi.org/10.1002/sres.2383>
- [17] Damenu, T.K., Beaumont, C. (2017). Analysing information security in a bank using soft systems methodology. *Information & Computer Security*, 25(3): 240-258. <https://doi.org/10.1108/ICS-07-2016-0053>
- [18] Proches, C.N.G., Bodhanya, S. (2015). An application of soft systems methodology in the sugar industry. *International Journal of Qualitative Methods*, 14(1): 1-15. <https://doi.org/10.1177/16094069150140010>
- [19] Thornhill, A., Lewis, P., Saunders, M.N.K. (2009). *Research Methods for Business Students*. New York: Pearson.
- [20] Pargaonkar, S. (2023). Advancements in security testing: A comprehensive review of methodologies and emerging trends in software quality engineering. *International Journal of Science and Research (IJSR)*, 12(9): 61-66. <https://doi.org/10.21275/SR23829090815>
- [21] Bauskar, S.R., Madhavaram, C.R., Galla, E.P., Sunkara, J.R., Gollangi, H.K., Rajaram, S.K. (2024). Predictive analytics for project risk management using machine learning. *Journal of Data Analysis and Information Processing*, 12(4): 566-580. <https://doi.org/10.4236/jdaip.2024.124030>
- [22] Elokby, E.A., Alawi, N.A., Abdelgayed, A.T.A., Alhodiany, Z.M. (2021). Does project risk management matter for the success of information technology projects in Egypt. In *2021 2nd International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*,

APPENDIX

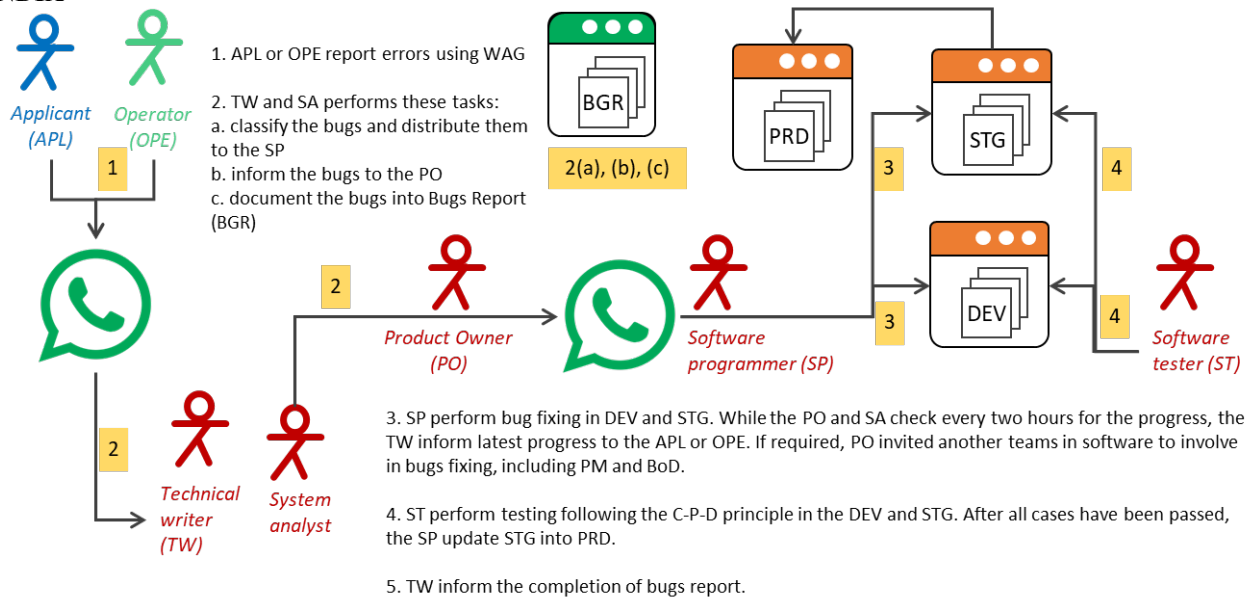


Figure A1. Helpdesk process model version 1

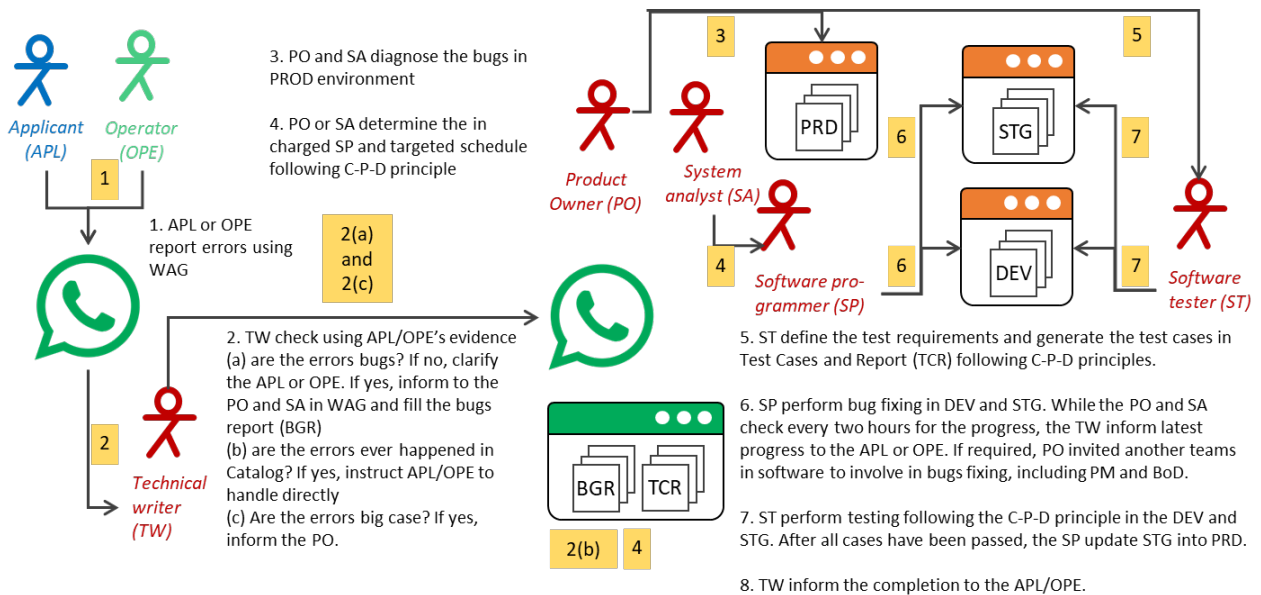


Figure A2. Helpdesk process model version 2