What Secondary Issues Contribute to Operational Problems?

An Investigation Based on Public Postmortems

Alexandru Muresan

Supervisor(s): Diomidis Spinellis[1], Eileen Kapel[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
July 22, 2025

Name of the student: Alexandru Muresan
Final project course: CSE3000 Research Project
Thesis committee: Diomidis Spinellis, Eileen Kapel, Benedikt Ahrens

Abstract

Operational incidents in software-defined systems can lead to significant disruptions, and while primary faults such as bugs or misconfigurations are well studied, secondary issues that exacerbate these failures remain underexplored. This research investigates what secondary issues contribute to operational problems by analyzing 1,500 publicly available incident reports from platforms such as GitHub and the Verica Open Incident Database (VOID). Using a large language model (LLM) and a predefined classification schema, the study extracts and categorizes these issues at scale.

The results show that communication failures (48.2%), monitoring and transparency deficiencies (46.5%), and documentation issues (41.1%) are the most prevalent secondary issues. These often co-occur, with the most common issue pair, communication failures and monitoring deficiencies, appearing together in over 600 reports, suggesting interdependent systemic weaknesses. Furthermore, these secondary issues show strong associations with different primary fault types, such as misconfigurations and software bugs, revealing distinct amplification patterns that affect incident severity and resolution time.

A reproducible data pipeline was developed to enable large-scale analysis, and manual validation of model annotations yielded an accuracy of 81.9%, confirming the reliability of the LLM-based classification approach. The study addresses the feasibility of AI-assisted analysis for postmortem diagnostics and provides actionable insights into operational fragility, emphasizing the need to address not only technical faults but also organizational and process-level weaknesses.

1 Introduction

Operational incidents in software-defined businesses can result in substantial disruptions, ranging from service outages and lost transactions to reputational damage and financial problems. These systems operate under high complexity, where rapid feature deployment and tight integration of services make them vulnerable to cascading failures. Understanding why these incidents occur and how to reduce their impact is a critical goal for any large-scale IT operation.

Historically, research has focused on primary faults as the main causes of failures [1; 2]. This has led to improvements in testing, monitoring, and change control processes. However, secondary issues, which are factors that amplify the severity or duration of an incident without directly causing it, remain underexplored. Despite their critical role in incident outcomes, these systemic weaknesses are less represented in structured analyses [3; 4].

This project investigates what kinds of secondary issues contribute to operational problems by analyzing a broad set of postmortem reports. Unlike earlier studies that examine incident resolution within a specific organization [5; 6], this research takes a cross-organizational view, leveraging a diverse dataset of incident reports from public sources such as GitHub and VOID. Additionally, it builds on the rise of Artificial Intelligence for IT Operations (AIOps) methodologies, which aim to bring machine learning and automation into IT operations [7; 8].

Understanding the co-occurrence of secondary issues, both with each other and in relation to primary faults, is essential because it reveals patterns of operational fragility that are often missed in root-cause centered analyses. These patterns can indicate system dependencies or repeated breakdowns in organizational processes such as communication, recovery preparedness, or visibility. From a sociotechnical systems perspective [9], failures rarely appear from a single point but emerge from interactions among human, technical, and organizational elements. Identifying these recurring clusters of contributing factors supports a more complete incident management strategy and can inform the design of resilient systems that address not only technical robustness but also process integrity.
To address this issue, our study is guided by the following research questions (RQs):

- **RQ1:** Which types of secondary issues are most frequently observed across incident reports?
- **RQ2:** Are there identifiable patterns linking specific secondary issues to primary faults?
- **RQ3:** Which secondary issues are commonly found together?

This project presents a quantitative analysis of postmortem reports to uncover recurring secondary issues that exacerbate operational incidents. It introduces a reproducible, LLM-assisted classification framework and reveals that issues such as communication failures, monitoring gaps, and documentation problems frequently co-occur and compound incident severity. The findings offer actionable insights into systemic weaknesses in incident response and demonstrate the value of going beyond primary fault analysis to better understand the full context of software failures.

The remainder of the paper is organized as follows: Section 2 discusses relevant background. Section 3 outlines the methodology used to extract and analyze the incident reports. Section 4 presents the Ethics of the research and the use of AI throughout the project. Section 5 presents our findings on secondary issue patterns. Section 6 offers a discussion and interpretation of the results. Section 7 addresses the limitations of the project and directions for future research, followed by concluding remarks in Section 8 .

2 Related Literature and Background Information

To understand the significance of secondary issues in operational reliability, we first examine prior research on system failures. This section reviews both technical fault studies and sociotechnical perspectives, highlighting the gaps in existing literature that this project aims to address.

Much of the foundational research on system reliability has focused on primary faults which are immediate technical causes such as software bugs, hardware failures, and misconfigurations [10; 1]. These studies have driven progress in preventive practices including testing, change management, and observability, all aimed at reducing the frequency and impact of such failures.

In contrast, secondary issues have received less systematic attention. These include operational shortcomings such as untested rollback procedures, inadequate monitoring, and unclear team responsibilities. Although frameworks such as AIOps aim to automate detection and response using machine learning and telemetry data [7; 8], they typically treat failures as isolated events rather than components of broader systemic patterns. Similarly, studies on change-induced incidents and internal incident management practices [6; 5] provide valuable information on organizational workflows but often lack generalizability between organizations.

Recent work has begun to examine the sociotechnical nature of incident response. Sillito and Kutomi [11] analyzed postmortem reports to identify how knowledge gaps, poor coordination, and tool limitations delay effective recovery. Their findings highlight that many incident outcomes are shaped less by the technical fault itself and more by how teams interpret and respond to it. However, the study is constrained by its reliance on a small set of voluntary disclosures, which can overrepresent well-documented high-impact failures.

Complementing these qualitative insights, efforts such as BugSwarm [3] and Zhou et al.'s machine learning-based root cause classifier [2] demonstrate the potential of scalable and data-driven approaches to operational diagnostics. Although both focus primarily on reproducible bugs and root causes, their methodologies inform the use of large language models to extract higher-order operational patterns from textual reports.

Finally, broader assessments of software failure research call attention to its methodological gaps. Amusuo et al. [12] criticize the lack of standardization in the field in the study of failures, while Gazzola et al. [4] show that many real-world problems emerge only in production, highlighting the need to study how systems behave in actual production, not just in theory or lab environments. These perspectives highlight the importance of expanding research beyond fault localization to include the systemic conditions that influence the way that incidents unfold.

This study contributes to this goal by systematically identifying and analyzing secondary issues across a large and diverse dataset of postmortem reports. In doing so, it aims to reveal recurring operational weaknesses that are often overlooked but critically shape the trajectory of real-world incidents.

The following section outlines the methodology used to identify and analyze secondary issues from postmortem reports using large-scale data extraction and language model-assisted classification.

3 Methodology

This section outlines the methodological framework used to identify and analyze secondary issues across incident reports. It details the data collection pipeline, the use of large language models for classification, and the process of building and validating a taxonomy for consistent annotation of secondary factors. This study adopts a repository mining approach to investigate secondary issues that contribute to operational incidents in software-defined systems. A large dataset of postmortem reports was collected from public sources and processed using an automated pipeline. The methodology combines data extraction, large language model-based annotation, and quantitative analysis [13] to identify recurring patterns in incident outcomes. The approach enables scalable analysis of real-world operational failures across organizational boundaries and supports the research questions through structured data classification and co-occurrence measurement.

3.1 Data Collection

Incident postmortem reports were sourced from publicly accessible repositories, primarily VOID [14] and a GitHub repository [15] containing around 200 reports. Since these platforms do not offer standard APIs for data retrieval, a custom Python-based scraper was developed. This scraper sends structured POST requests, handles pagination using offsets, and extracts relevant fields such as incident dates, impact descriptions, and involved technologies. The collected data is stored in JSON format, with mechanisms to append only new records, ensuring consistency for additional runs. Although 10,000+ reports were availablle on VOID, only the first **1,500 reports** were scrapped and downloaded for further processing throughout this project. This was done in order to maintain relevancy of the data taking the most recent ones, the 1500 reports are ordered chronologically descending from **December 2024 until November 2021** covering a 3 year span.

3.2 Data Processing

To identify secondary issues within the incident narratives, the collected reports were processed using LLama 3 70B model [16] via the Groq API. Each report's "Full text description" was given through a prompt that asked the model to extract contributing factors that increased the severity or recovery time of the incident, excluding the primary fault. The prompt was given 7 categories or types of secondary issues to use for the processing of the 1,500 reports. Prompts were limited to 4,000 characters to remain within the API's rate limits. The model responses were parsed and stored in a structured JSON file, indexed by report ID. The script included rate limiting and error handling to ensure robustness and scalability.

A taxonomy of secondary problems was developed following established guidelines for developing taxonomies in information systems research. The process was guided primarily by the Extended Taxonomy Design Process (ETDP) as outlined by Kundisch et al. [17], and inspired by the earlier methodology proposed by Nickerson et al. [18]. These frameworks require iterative, design-driven development cycling between empirical observation and conceptual clarification. An initial version of the taxonomy was created using outputs from a large language model (LLM) and improved with qualitative examples derived from prior research [19; 20]. Through multiple iterations, the categories were refined to ensure they were distinct, well-defined categories that together captured the full range of issues seen in postmortem reports. Modifications included merging semantically close categories and introducing new labels where recurring patterns appeared in uncategorized reports. This structured and iterative process helped establish a robust taxonomy suitable for large-scale analysis and aligned with rigorous empirical standards.

To investigate the second research question, a dedicated processing pipeline was established to integrate the primary fault classifications with the previously extracted secondary issues. Primary fault data, sourced from a json file containing the primary faults of the same 1,500 reports, was loaded, providing a structured record of incident IDs and their assigned primary fault labels. At the same time, the detailed secondary issue descriptions were processed. This involved using regular expressions to extract and normalize individual secondary issue labels (converting to lowercase and removing punctuation), which were then mapped to their corresponding incident IDs. Finally, these two distinct datasets were combined based on their shared incident identifiers, creating a unified structure that associated each primary fault with a comprehensive list of its accompanying secondary issues. This integrated dataset served as the foundation for analyzing co-occurrence patterns between primary fault types and the array of contributing secondary factors.

To analyze the co-occurrence of secondary issues across incident reports, a Python-based script was developed to process and structure the extracted labels. Using regular expressions, the script parsed the annotated text to extract issue categories per report and normalize them by removing punctuation and enforcing lowercase formatting. The structured data was then processed using the *pandas* library, which facilitated the construction of a co-occurrence matrix. In this matrix, both rows and columns represented distinct secondary issue categories, and cell values reflected how often each pair of issues appeared together in the same report. To avoid duplication and ensure matrix symmetry, only the lower triangle of the matrix was considered.

Using *pandas*, the co-occurrence counts were extracted, sorted, and exported as a CSV file containing the co-occurring secondary issue pairs. This tabular output enabled downstream analysis and visualization of systemic patterns across incidents.

3.3 Validation of Results

To evaluate the reliability of the secondary issue classifications generated by the LLama 3 70B model, a two-step validation strategy was employed. The first and most important validation step consisted of a manual review. A representative set of 100 incident reports was randomly sampled and annotated by hand according to the established secondary issue taxonomy. These annotations served as ground truth for evaluating the LLM's predictions.

The LLama 3 70B model's output was then directly compared against these human-labeled reports. Out of 299 total secondary issue labels generated by the model across the selected reports, 245 were consistent with the manual annotations, resulting in an overall accuracy of 81.94%. These results demonstrate that the model tends to produce contextually appropriate classifications, especially for well-defined categories. However, some misclassifications occurred in edge cases involving semantically overlapping labels, suggesting the model may still over-predict.

As a secondary step, the outputs of the 70B model were compared with those of a smaller model, LLama 3 8B Instant model, using the same prompts and report set. Inter-model agreement was assessed using Cohen's Kappa coefficient, a standard measure of inter-rater reliability that accounts for chance agreement. The comparison yielded consistently low agreement scores across all categories, with most values falling below $\kappa = 0.2$, indicating only slight agreement. This finding reinforced the decision to rely primarily on manual verification, as the 8B model was found to be overly permissive and inconsistent.

Together, these validation efforts highlight both the value and the limitations of using large language models in structured classification tasks. While the LLama 3 70B model shows promising accuracy when benchmarked against human judgment, caution is still required when interpreting its outputs, especially without expert oversight.

4 Ethics of the research and Use of AI

Responsible and ethical research practices are critical in the context of large-scale data collection and AI-assisted analysis. This section explains how the project ensures reproducibility, ethical data usage, and transparency in both automation and human oversight.

4.1 Responsible research

Reproducibility. This research is designed to be reproducible through the provision of a complete and transparent workflow. All scripts used for data collection, processing, and analysis are implemented in Python and organized in a publicly accessible repository. The source data, incident reports from VOID and GitHub, is also publicly available, allowing others to use the same inputs. Clear documentation accompanies the scripts to ensure that other researchers can follow the same steps and verify results.

Replicability. To ensure replicability, a replication package will be available via Zenodo [21]. This package includes:

- Python scripts used for data collection from the Verica Open Incident Database (VOID) [14] and the public GitHub repository [15].
- A dataset of 1,500 incident reports processed by the two LLMs used in the study.
- Prompt templates and scripts used to interface with the LLMs.
- A text file containing the manual annotations of 100 randomly selected incident reports.
- All figures presented in the report, along with the corresponding source code.
- A README file with detailed instructions for reproducing the analysis and navigating the repository.

This replication package is intended to support transparency and facilitate independent verification of the study's findings. By providing both the raw inputs and the full processing pipeline, the project enables other capable researchers to build upon this work or adapt its methods to related domains.

Ethical Integrity. The study adhered to ethical research standards throughout the data collection and analysis phases. Web scraping was performed responsibly by respecting the terms of use of the target websites and implementing request throttling to avoid server overload. No authentication barriers were bypassed, and only publicly accessible data was gathered. The study refrained from collecting any personally identifiable information or sensitive content, thereby ensuring compliance with responsible data usage principles.

4.2 Use of AI

Artificial intelligence played a significant role in both the analytical and editorial aspects of this research. Large language models, specifically LLama 3 models accessed via the Groq API, were used to interpret incident reports by identifying and categorizing secondary issues that contributed to the severity or duration of operational incidents. This automated interpretation enabled a scalable and consistent analysis of qualitative data that would have been difficult to achieve manually.

In addition to analysis, AI tools were used to support the writing process. Language models assisted with grammar correction, sentence restructuring, and formatting suggestions, helping to improve clarity and coherence throughout the report. They were used as a means of improving efficiency and productivity rather then replacing critical thinking. These editorial contributions were limited to stylistic refinement, no AI-generated content was used to formulate research questions, draw conclusions, or conduct core analysis. The use of such tools was conducted in line with academic ethical standards, ensuring that the intellectual contributions remain those of the author.

5 Results

Here, we present the findings from our analysis of 1,500 incident reports. The results are structured around the three research questions, revealing which secondary issues are most common, how they relate to primary faults, and how often they appear together in operational failures.

5.1 RQ1: Most Frequent Secondary issues

To address this research question, all extracted secondary issues were categorized according to the predefined schema and counted across the dataset. This section presents the frequency distribution of these categories, highlighting which types of secondary issues are most prevalent in publicly available incident reports. The results provide insight into the most common systemic weaknesses contributing to the impact or recovery of operational failures. These categories are defined as follows:

Monitoring & Transparency Deficiencies: This category includes cases where lack of alerts or limited observability into system behavior delayed the detection or diagnosis of the incident. For example, teams may have been unaware of a service failure until it was reported by users.

Rollback Preparedness: Issues under this category refer to insufficient or untested rollback mechanisms. This includes scenarios where teams attempted to revert to a previous stable state but were hindered by incomplete recovery procedures.

Automation Gaps: This includes reliance on manual interventions for processes that could have been automated, such as deployment or scaling. Manual execution increases the likelihood of errors and often slows down recovery.

Communication Failures: Covers breakdowns in coordination, such as unclear roles during the incident, or miscommunication between teams. These issues often lead to extended resolution times and confusion during recovery.

Documentation Issues: Refers to outdated, missing, or unclear internal documentation that impedes troubleshooting or executing recovery procedures. Teams encountering unfamiliar systems or edge cases may lack proper guidance in these situations.

Categorization Failures: Includes incidents that were misclassified in severity or type, leading to an inappropriate response. For instance, a high-impact issue being labeled as low priority can significantly affect time to resolution.

Miscellaneous / Unknown: A catch-all category for secondary issues that do not clearly fit into the other six. This includes vague or context-specific problems that could not be confidently categorized.

Based on the run of the llama 3 model, here are the results:

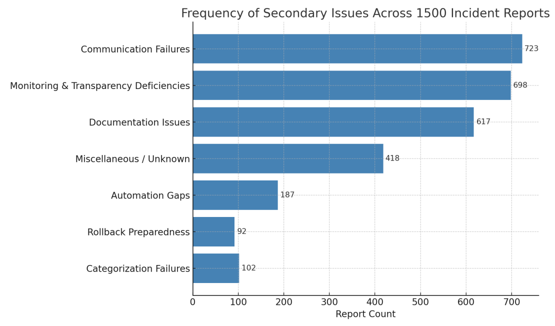

Figure 1: Frequency of secondary issues across 1500 incident reports

Figure 1 presents all the 7 secondary issues from the decided taxonomy together with their occurrences across the 1,500 processed reports. As it can be observed from the chart above, the most common secondary issue is **Communication Failures** with 723 appearances (48.2%), followed by **Monitoring & Transparency Deficiencies** with 698 (46.5%) and **Documentation Issues** with 617 (41.1%).

5.2 RQ2: Link between secondary issues and primary faults

The following figures presents the six secondary issues associated with each of the primary fault types, it was decided to remove the unknown label from both datasets as it did not add any value to the paper. In order to represent this, a heatmap visualization was chosen which contains the secondary issues on the vertical axis and primary faults on the horizontal axis. In the heatmap itself, the numbers in the boxes represent the common number of occurrences between the secondary issues and primary faults within the processed reports.

These charts provide a comparative overview of how often different secondary issues co-occur with each primary fault category. For example, if a specific secondary issue is present in 50% of all incidents involving a particular fault, its bar will reflect that proportion. This approach facilitates the identification of common patterns and highlights which operational weaknesses tend to appear together in the context of certain root causes.

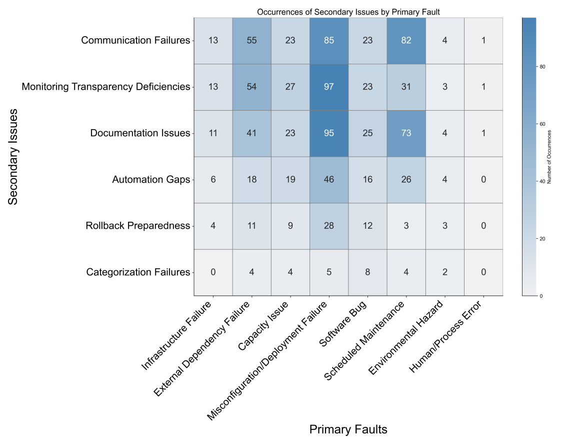

Figure 2: Secondary issues observed in incidents labeled with Primary faults

5.3 RQ3: Co-occurrence of secondary issues

To understand how secondary issues tend to co-occur in incident reports, we built a co-occurrence matrix based on the presence of normalized issue labels within the same report. For each pair of secondary issues, we counted how many times both were mentioned together.

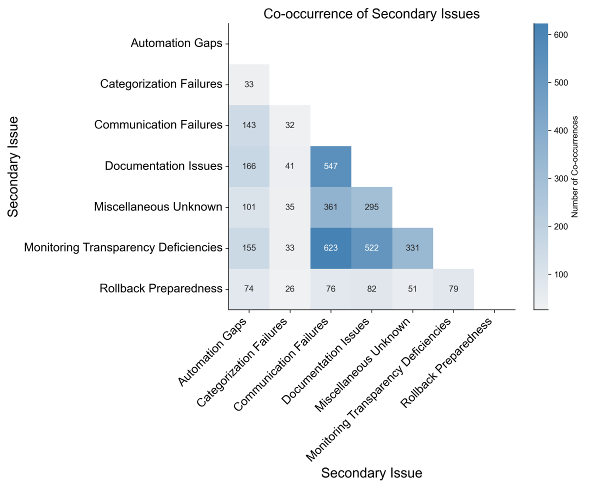

Figure 3: Heatmap of the co-occurring secondary issue pairs across incident reports.

Figure 3 presents the co-occurring issue pairs. The most common co-occurrence was between **Communication Failures** and **Monitoring & Transparency Deficiencies**, appearing together in 623 reports. Other high-frequency pairs include **Communication Failures** with **Documentation Issues** (547 reports), and **Documentation Issues** with **Monitoring & Transparency Deficiencies** (522 reports).

6 Discussion

This section interprets the findings presented in Section 5, explaining their significance and placing them within the broader context of existing literature on operational incidents and secondary issues.

The analysis of 1,500 public incident reports from Figure 1 reveals that **Communication Failures**, **Monitoring & Transparency Deficiencies**, and **Documentation Issues** are the most frequently observed secondary issues contributing to operational problems. This finding aligns with existing research highlighting the importance of human factors, knowledge gaps, and coordination problems in system reliability [20; 19]. The high prevalence of these issues suggests that weaknesses in communication, observability, and knowledge management are pervasive systemic factors that amplify the impact or duration of operational failures.

While these top three secondary issues are consistently present across various primary fault types, their relative prominence shifts. For instance, as we can notice from Figure 2 **Communication Failures** and **Monitoring & Transparency Deficiencies** are particularly pronounced with *External Dependency Failure*, likely due to the critical need for rapid coordination with external teams and clear visibility into third-party service health. In incidents stemming from *Misconfiguration*, the **Documentation Issues** and **Monitoring & Transparency Deficiencies** secondary issues become more prominent, emphasizing the role of clear instructions and early detection of anomalous behavior. For *Software Bugs*, issues such as **Monitoring & Transparency Deficiencies**, **Documentation Issues**, and **Communication Failures** remain key, indicating their broad impact regardless of the primary cause. These patterns indicate that secondary issues act as critical amplifiers, often impacting the initial problem regardless its origin [9].

The co-occurrence analysis further illuminates the interconnectedness of these weaknesses. According to Figure 3 the most frequent co-occurring pairs are **Communication Failures** with **Monitoring & Transparency Deficiencies** (623 reports), **Communication Failures** with **Documentation Issues** (547 reports), and **Documentation Issues** with **Monitoring & Transparency Deficiencies** (522 reports). This strong interdependency suggests that addressing individual secondary issues in isolation may be insufficient. For example, poor communication can exacerbate monitoring gaps, and inadequate documentation can hinder effective communication during an incident. This reinforces the notion that effective incident response requires an ample approach to mitigate these interconnected weaknesses, rather than focusing solely on the primary fault.

A somewhat unexpected finding is the relatively lower frequency of **Automation Gaps** and **Rollback Preparedness** compared to the top three issues and compared to the initial results from the GitHub repository, where these two were the most common in the 200 reports dataset. While these are crucial aspects of operational resilience, their less frequent appearance as amplifying factors in the analyzed public postmortems could suggest varying levels of organizational maturity in these areas or a reporting bias in public incident narratives [12; 4]. It is also possible that the impact of these issues might be implicitly attributed to other categories, such as **Communication Failures** or **Monitoring & Transparency Deficiencies**, if they lead to extended resolution times that are then documented through a lens of detection or coordination delays [20].

7 Limitations and Future Work

This study has several limitations that inform promising directions for future research. Below, we outline these grouped by methodological, data-related, and validation aspects.

Project Duration and Iteration Constraints. The limited timeframe of the project constrained the extent to which the methodology could be refined. Some implementation choices, such as prompt formulation, model selection, and classification refinement, had to be fixed early and could not benefit from multiple rounds of tuning or comparative experimentation.

Dataset Size The analysis was based on 1,500 incident reports sourced from public platforms such as GitHub and VOID. While this is a meaningful sample size for exploratory research, it captures only a subset of postmortems generated in industry. Publicly available reports tend to be better written and biased toward high-impact failures from organizations with mature post-incident processes. As a result, certain secondary issues or failure contexts may be overrepresented, while others, especially those from smaller teams or internal incidents, may be underreported.

Validation Process and Expertise. Manual validation of model-generated classifications was conducted by a single student annotator. Although this allowed for an initial precision estimate (approximately 80%), the lack of multiple annotators or expert reviewers limits the strength of the validation. Another aspect that might impact this is the relatively small size of the ground truth sample which although randomly chosen could impact a larger dataset differently. A more robust evaluation would involve multiple independent raters and measurement of inter-rater agreement using metrics such as *Cohen's Kappa*.

Model Generalization and Taxonomy Robustness. The classification taxonomy was developed through iterative refinement and informed by prior literature, but was not independently evaluated for generalizability across domains. Future work could further assess its completeness and adaptability to incident reports from other industries or technical stacks.

Directions for Future Research. Several extensions could improve and validate this work:

- Incorporate private or industry-shared postmortems to improve dataset diversity and generalizability.
- Include structured metadata such as incident severity, response duration, or affected components to enable more contextualized insights.
- Conduct expert-led manual annotation of a larger sample to evaluate and improve classification accuracy.
- Explore alternative LLMs or prompt strategies to test the robustness of the pipeline and minimize bias in output frequency.
- Revisit the correlation between primary faults and secondary issues using larger or stratified datasets to test whether observed patterns hold beyond the most frequent categories.

8 Conclusions

This research explored how secondary issues contribute to operational problems in software-defined systems by analyzing 1,500 publicly available incident postmortems. By leveraging a structured classification schema and large language models, the study surfaced recurring systemic weaknesses that often aggravate the severity or duration of incidents. **Communication Failures**, **Monitoring & Transparency Deficiencies**, and **Documentation Issues** emerged as the most common secondary issues, frequently appearing together and in combination with primary faults such as *Misconfigurations* and *Software Bugs*. These patterns suggest that operational fragility is rarely the result of a single flaw. More often, it stems from the interaction between technical and organizational deficiencies that damage incident response and recovery.

Beyond classifying frequent patterns, this work introduced a scalable and reproducible pipeline for extracting and analyzing secondary issues using AI-assisted methods. It demonstrates the potential of combining structured prompt design with statistical analysis to uncover meaningful insights from unstructured technical reports. While the analysis was constrained by the scale of the dataset and the limitations of automated validation, the findings offer practical implications for improving incident preparedness and response strategies. Future research can build on this foundation by incorporating richer metadata, refining the classification schema through expert feedback, and extending the approach to a larger and more diverse set of incident reports. This work highlights the value of looking beyond root causes to understand the broader ecosystem of failures that shape real-world operational outcomes.

A Appendix

A.1 Use of LLM

For this project two LLM's were used for the processing of the 1500 reports and they were given the following prompt: "You are an expert in post-incident analysis. Given the following incident report, identify any secondary issues—problems that didn't directly cause the incident, but contributed to its severity, impact, or delayed recovery. Use only the following categories:

- **Monitoring & Transparency Deficiencies** (e.g., missing alerts, poor observability, limited visibility)

- **Rollback Preparedness** (untested rollback plans, incomplete recovery procedures)
- **Automation Gaps** (reliance on manual processes, lack of deployment automation)
- **Communication Failures** (unclear roles, delayed updates, miscommunication)
- **Documentation Issues** (missing, outdated, or unclear procedures)
- **Categorization Failures** (incorrect severity tagging, misclassified incidents)
- **Miscellaneous / Unknown** (for issues that do not fit any other category)

```
Incident Report:
{incident text inserted here}
```
List the secondary issues as bullet points."

References

[1] D. Oppenheimer, A. Ganapathi, and D. A. Patterson, "Why do internet services fail, and what can be done about it?" in *4th Usenix Symposium on Internet Technologies and Systems (USITS 03)*, 2003.

[2] J. Zhou and S. Li, "Distance based root cause analysis and change impact analysis of performance regressions," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–9, 2015. [Online]. Available: https://www.researchgate.net/publication/282311296

[3] D. A. Tomassi, N. Dmeiri, Y. Wang, A. Bhowmick, Y.-C. Liu, P. Devanbu, B. Vasilescu, and C. Rubio-González, "Bugswarm: Mining and continuously growing a dataset of reproducible failures and fixes," 2019. [Online]. Available: https://arxiv.org/abs/1903.06725

[4] L. Gazzola, L. Mariani, F. Pastore, and M. Pezz'e, "An exploratory study of field failures," 2017. [Online]. Available: https://arxiv.org/abs/1708.09494

[5] E. Kapel, L. Cruz, D. Spinellis, and A. van Deursen, "Enhancing incident management: Insights from a case study at ing," in *Proceedings of the 1st IEEE/ACM Workshop on Software Engineering Challenges in Financial Firms*, ser. FinanSE '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1–8. [Online]. Available: https://doi.org/10.1145/3643665.3648048

[6] E. Kapel and B. Ahrens, "On the difficulty of identifying incident-inducing changes," in *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice*, 2024.

[7] A. A. Hinai and M. A. Mazroui, "Optimizing and enhancing it operation operating models through artificial intelligence," in *2024 2nd International Conference on Computing and Data Analytics (ICCDA)*, Nov. 2024, pp. 1–5.

[8] R. Colomo-Palacios, L. Rijal, and M. Sánchez-Gordón, "Aiops: A multivocal literature review," in *Artificial Intelligence for Cloud and Edge Computing*. Springer, 2020, pp. 31–50. [Online]. Available: https://www.rcolomo.com/papers/412.pdf

[9] N. G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. The MIT Press, 01 2012. [Online]. Available: https://doi.org/10.7551/mitpress/8179.001.0001

[10] B. Beyer, C. Jones, J. Petoff, and N. R. Murphy, *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media, 2016.

[11] J. Sillito and L. Kutomi, "Failures and fixes: A study of software system incident response," *arXiv preprint arXiv:2008.11192*, 2020. [Online]. Available: https://arxiv.org/abs/2008.11192

[12] P. C. Amusuo, A. Sharma, S. R. Rao, A. Vincent, and J. C. Davis, "Reflections on software failure analysis," *arXiv preprint arXiv:2209.02930*, 2022. [Online]. Available: https://arxiv.org/abs/2209.02930

[13] T. Hirsch and B. Hofer, "Root cause prediction based on bug reports," in *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2020, pp. 171–176.

[14] Verica Inc., "The void: Verica open incident database," https://www.thevoid.community/database, 2025, accessed: 2025-06-08.

[15] danluu, "A list of post-mortems!" https://github.com/danluu/post-mortems, 2025, accessed: 2025-06-05.

[16] A. Grattafiori, A. Dubey, A. Jauhri *et al.*, "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[17] D. Kundisch, J. Muntermann, A. M. Oberländer, D. Rau, M. Röglinger, T. Schoormann, and D. Szopinski, "An update for taxonomy designers: Methodological guidance from information systems research," *Business and Information Systems Engineering*, vol. 64, no. 4, p. 421 – 439, 2022, cited by: 120; All Open Access, Hybrid Gold Open Access. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85117576515&doi=10.1007%2fs12599-021-00723-x&partnerID=40&md5=6371611ad46912488aa79ab4f9a86d78

[18] R. C. Nickerson, U. Varshney, and J. M. and, "A method for taxonomy development and its application in information systems," *European Journal of Information Systems*, vol. 22, no. 3, pp. 336–359, 2013. [Online]. Available: https://doi.org/10.1057/ejis.2012.26

[19] J. Sillito and M. Pope, "Failing and learning: A study of what is learned about reliability from software incidents," in *2024 IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 2024, pp. 295–302.

[20] P. Dogga, S. Jha, Z. Wang, J. Schenkel, and A. Ghosh, "Taxonomy, insights, and tools for root

cause labelling of incidents," in *2023 USENIX Annual Technical Conference (USENIX ATC 23)*. USENIX Association, 2023. [Online]. Available: https://www.usenix.org/system/files/atc23-dogga.pdf

[21] I. Aldea, M. Georgiev, J. Rutkowski, A. Mureşan, and D. Bunschoten, "What can we learn from incident reports? - web scraper," Jun. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.15711606