Understanding Software Failures Through Incident Report Analysis

Author: Iulia-Maria Aldea[1] Responsible professor: Prof. Dr. Ir. Diomidis Spinellis[1] Supervisor: Eileen Kapel[1]

[1]EEMCS, Delft University of Technology

1. INTRODUCTION

Background

- **Incident** = unplanned disruption + urgency [1]
- **Artificial Intelligence for Development Operations (AIOps):** understand past incidents to predict and/or mitigate new ones (Fig. 1)

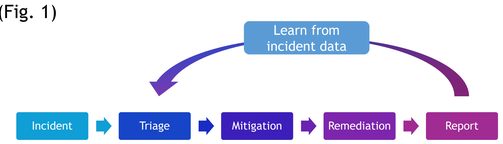

Figure 1: Learning from incident reports

- No standardized reporting [2] => current research **specialized**:
  - For company: ING [3], ANT Group [4]
  - For technology: Cloud Based Systems [5]

Goals of This Research

- E*xtraction of incident characteristics* from reports
- *Similarity* and *pattern identification*

Research Questions

"What are the characteristics of incident reports caused by software changes"

RQ1: What general characteristics of incidents are evident across the collected incident reports and how can they be automatically extracted?

RQ2: What is the relationship between an incident's cause, impact and remediation that follows from the established characterization?

RQ3: What types of software changes associated with incident occurrence are observed in the dataset?

RQ4: What recurring patterns can be identified, and what are the three most prominent clusters based on incident similarity?

2. METHODOLOGY

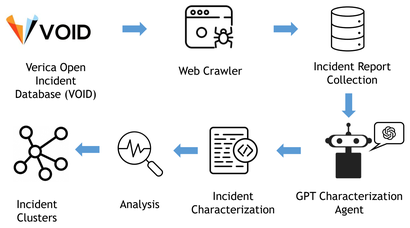

Figure 2: Research workflow

- Research Workflow (Fig. 2): Extract incidents (source VOID); Characterize with GPT 4.1 Mini Model; Analyse (Fig. 3); Establish incident archetypes
- We analyse: **348** incident reports
- Manual validation: **34** reports

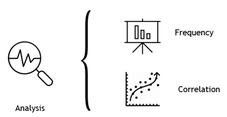

Figure 3: Analysis pipeline

3. RESULTS

Model Performance

GPT 4.x Mini family evaluation:

- Ground truth: **34** manually labeled incidents

Metrics used for evaluation:

- Jaccard Similarity for remediation and impact tags: measures similarity between set of labels (Fig. 4)
- Cohen's Kappa for *cause*, *severity* and *mitigation*: measures agreement; accounts for chance (Fig. 5)

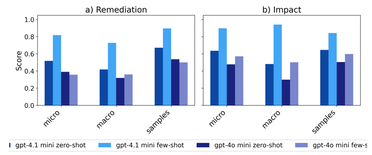

Figure 4: Jaccard Similarity comparison between models

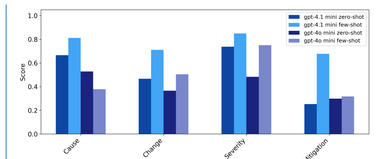

Figure 5: Cohen's Kappa comparison between models

GPT-4.1 Mini model under few-shot prompting reaches an overall accuracy of above 80% when predicting incident characteristics in accordance with the defined taxonomy.

Frequency Analysis

Table 1: Most Frequent Incident Characteristics

Category	Value	Count	Percent
Cause	Code Defect	85	28.24
	Capacity Issue	79	26.25
Severity	Major	233	77.41
Mitigation	Reduction	274	91.03
Impact	Partial Production Outage	228	75.75
	Degraded Service/Performance	225	74.75
Remediation	Hot Fix	118	39.20
	Infrastructure Change	75	24.92

- *Code Defects* and *Capacity Issues* account for ~55% of all reported incidents (Table 1).

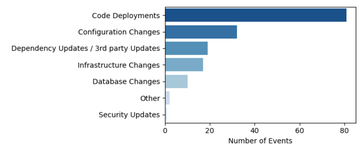

Figure 6. Distribution of incident inducing changes

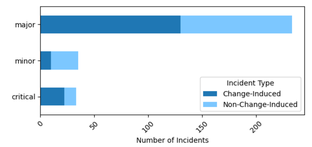

Figure 7. Severity of change-induced incidents

- 53.82% of incidents are caused by software changes, predominately *Code Deployments* (Fig. 6).
- Most change-induced incidents are attributed *Major* severity level (Fig. 7)
- *Code Deployments* account for most change-induced incidents.
- 75% of incidents result in *Partial Production Outages* and/or *Degraded Service*

Predictive Relationships

Figure 8: Increased likelihood severity and remediation mappings of change-induced incidents

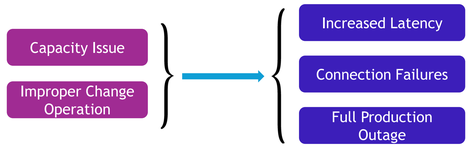

Figure 9: Statistically relevant relationships between cause and impact of an incident

Incident Archetypes

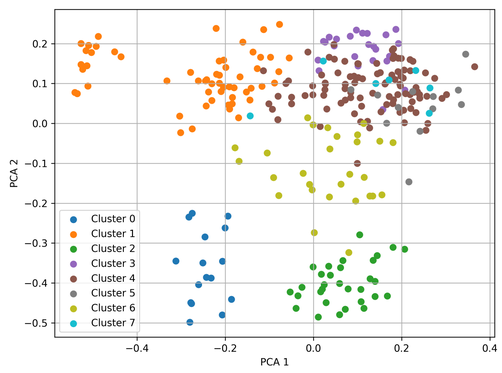

Figure 10. Cluster visualization (GMM, 8 components, PCA)

- Clustering: GMM (Gaussian Mixture Model, 8 components; PCA)
- Clustering Performance Metrics:
  - Silhouette Score 0.06
  - Calinksi-Harabasz Index 10.73
  - Davies Bouldin 2.89
- GMM identifies **8** clusters (Fig. 10).
- Clusters are **not** immediately separable, however there are clusters that stand out.

The top three most common archetypes involve capacity-driven major outages, code defect–related major partial outages, and critical outages driven by improper change operations and code defects.

4. CONCLUSION

Findings

- We propose an *incident characteristics taxonomy.*
- We find:
  - Slightly over half of the incidents are *change-induced* → room for improvement in development pipelines.
  - *Change-induced* incidents are more likely to be *critical* and require *manual recovery* (Fig. 8) → missing or insufficient automated recovery mechanisms.
  - *Improper Change Operations* are more likely to lead to *Full Production Outages* (Fig. 9) which demand *Rollback* remediation → importance of fallback mechanisms.
  - Dominant clusters → applicability of unsupervised clustering for incident similarity.
- We contributes to AIOps research (Fig. 11)

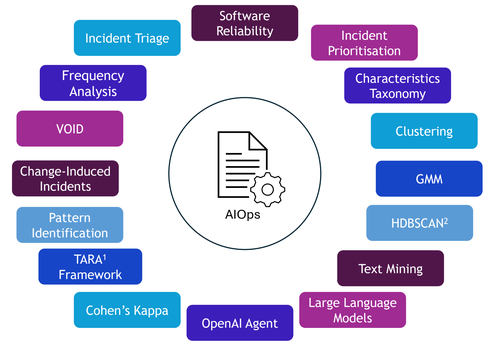

Figure 11: Topics

[1] TARA: Transference, Avoidance, Reduction, Acceptance
[2] HDBSCAN: Hierarchical Density-Based Spatial Clustering of Applications with Noise

Actionable Insights

- Remediation efforts focus on damage containment
  ⟹ need for targeted playbooks
- *Configuration Changes* and *3rd Party Updates* cause incidents reasonably often
  ⟹ need for testing beyond code correctness

Future Work

- Manually label a larger dataset (>100) of incidents to increase trust in the obtained results. Increase labeling quality by using multiple annotators.
- Investigate performance of more model families.
- Fine-tune an agent for better performance on incident characteristics extraction.

5. REPRODUCTIBLE RESEARCH

We contribute to the efforts of open research by making available the data and scripts used.

Code

References

TUDelft