# When changes break things
## What characterizes breaking changes to a live system

Danny Bunschoten
Delft University of Technology
Thesis committee: Prof.Dr.Ir. Diomidis Spinellis, Eileen Kapel, Benedikt Ahrens

## 📖 Introduction  ①

- Companies depend on online availability, reducing downtime is essential
- Learning from previous incidents to more efficiently solve future incidents
- Root cause analysis on 258 incident reports from Google
- Using AI to extract the relevant details
- Linking key numbers and extracted information to TTR to reveal patterns

## 📚 Related Literature  ②

- Many incidents caused by small quickly noticeable changes
- Number of available skilled engineers is a major factor in determining the MTTR
- Proven categories of root causes used in this study

## 🔍 Research Question  ③

**To what extent do different types of software changes correlate with the mean time to repair (MTTR) of incidents at Google?**

## ⚗ Methodology  ④

1. Downloaded 979 incident reports from Google's public repository
2. Filtered to reports containing root cause information (n=258)
3. Preprocessed text using NLP techniques and data cleaning
4. Trained classification models for automated change type identification
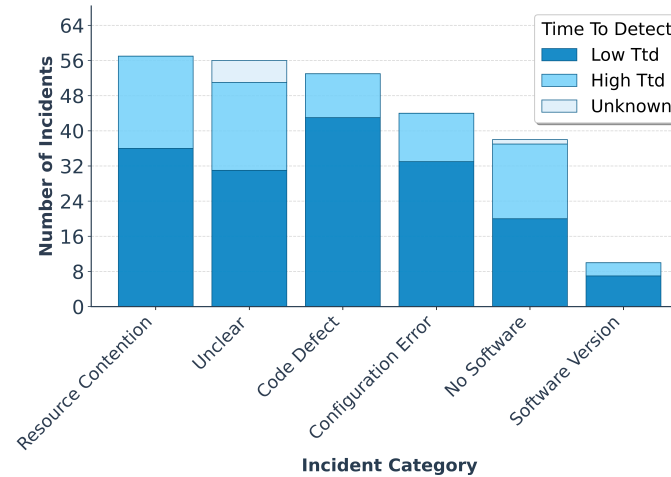5. Statistical evaluation of change impact on MTTR and TTD metrics

## 📊 Results  ⑤



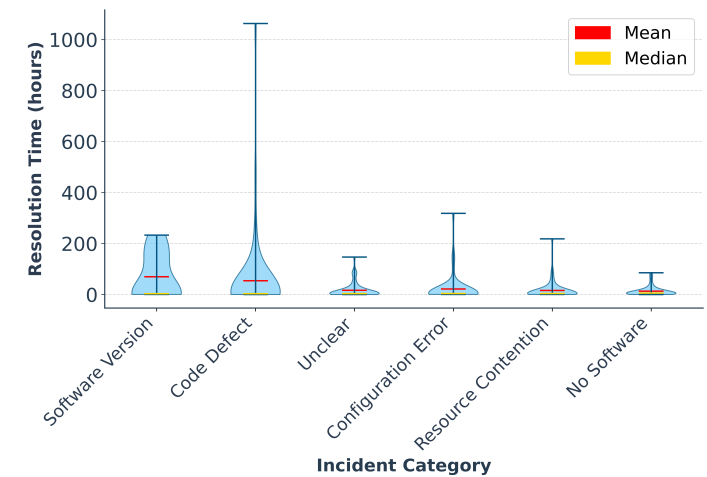Figure 1: TTD distribution per root cause category



Figure 2: TTR distribution per root cause category

### 💡 Key Findings

- **Software version** changes (69.8 hours) and **Code defects** (54.0 hours) have significantly higher MTTRs than average 26.7 hours (Figure 2)
- **Code defects** show the highest standard deviation for resolution time of 163.58 hours (Figure 2)

- **Code defects** and **Configuration errors** have at least 75% of their incidents categorized as **low TTD** (Figure 1)
- **Software version** incidents are the least common incidents, but do exhibit the highest MTTR (Figures 1 & 2)

## 💬 Discussion  ⑥

### ⚠ Limitations

- Limited dataset from only one company (Google)
- Potential selection bias in publicly available reports

### 🚀 Future Work

- Validate findings across multiple companies and industries
- Expand analysis to include incident severity classifications

### ⊘ Key Conclusions

- **Software version incompatibilities** represent the highest-risk change type with 69.8h average MTTR
- **Code defects and configuration errors** are quickly detected but varied in resolution time

- Use **insights** from past incident to improve system reliability
- Organizations should **tailor response strategies** based on root cause patterns