International Journal of Innovations & Research Analysis (IJIRA)
ISSN :2583-0295, Impact Factor: 5.449, Volume 03, No. 04(II), October- December, 2023, pp 136-142

SOFTWARE-RELATED PRODUCT HARM CRISIS: ANALYZING CODING SCHEMES TO DEVELOP MANAGEMENT FRAMEWORK (SR-PHC-MF)

Manoj Kumar Rana*
Dr. Shalini Devi**

ABSTRACT

Product harm crisis (PHC) involves cross section of multiple disciplines. It is a critical event for any business as it directly impacts end-consumers' safety. Social media and online consumer platforms help to amplify the message. If not handled well, this has potential to negatively impact the brand reputation and financial viability of the firm which is in middle of such crisis. Understanding the role of software in causing and then managing the impact of such crisis is not much studied. Software solutions are critical components of product development across various stages including conceptualization, design and development, testing, distribution and after sales support. Software based solutions are also used by support teams to assess customer and channelize communications in case of Product harm crisis event. This paper covers the Software-Related PHC events and suggests a framework to handle those. The trend of increased utilization of software in product development process and increasing complexity of software systems including opaqueness caused due to software components modularity, artificial intelligence and machine learning algorithms.

Keywords: *Software Related, Product Harm Crisis, Software Related Product Harm Crisis Management Framework, Software Failures.*

Introduction

Heath, R. L., & Palenchar, M. J. (2000), "A product harm crisis refers to a situation where a product, whether due to design flaws, manufacturing defects, or other issues, causes harm to consumers, leading to a significant negative impact on the reputation and financial stability of the company responsible for the product.". A company may face safety concerns, product defects or other miscellaneous issues related to product. These issues have potential to harm consumers with varied intensity. Handling PHC requires effective utilization of mix of communication, correction, and reputation management. Various studies are done in past across different domains like Automobile, Telecom, Healthcare, CPG, FMCG products etc., highlighting the impact of Product Harm Crisis situations on specific brands (Backhaus M. & Fischer, M. 2016) as well as entire industry (Birsch D. & Fielder John H. 1994).

Nowadays, software is providing major functionalities across stages of product conceptualization, design and development, distribution and after sales support for all these domains. Often, the software has a huge contribution in meeting safety requirements of the product, this implies that failure or malfunction might result into, or add to, a fatal accident. Even though software engineering

* AGM, HCL Technologies, Noida, UP, India.
** Associate Professor, Keshav Mahavidyalaya, University of Delhi, Delhi, India.

practices and tools-chain have matured consistently, still software are prone to defects due to various factors including missed requirements, programming defects, test insufficiency etc, (Santos et. al. 2021). The involvement of latest trends including Cloud Computing, Machine Learning and Artificial intelligence enables a whole new set of opportunities and capabilities to software, not available earlier. However, these also introduces new set of defects which are not only difficult to fix due to black box nature of underlying algorithms but also difficult to notice for situations including in-built bias (Barocas et. al. 2016), or subtle data shifts (Chehreghani et.al. 2007) etc., until it is too late. Such conditions make product predict wrong outcomes and thus has potential to cause harm for basing decisions on these wrong predictions.

This paper evaluates Software Related product harm crisis situations and provide a framework to address these challenges.

Literature Review

Different definitions are provided for the term 'Product harm crisis' across literature. The definitions vary based on situation, application, domain etc.

Heath R. L., & Palenchar M. J. (2000) defines it as "A product harm crisis refers to a situation where a product, whether due to design flaws, manufacturing defects, or other issues, causes harm to consumers, leading to a significant negative impact on the reputation and financial stability of the company responsible for the product.".

Product harm crisis is also defined as "well publicized instances of defective or dangerous products" (Dawar & Pillutla 2000). For the purpose of this research, we have expanded this definition to also include potential for a severe financial or reputation loss due to software functionality loss or data leakage to the consumers. Some of the pertinent examples are Mattel's toys recall in 2007 (Teagarden, 2009), Toyota's automobile recall in 2009 after a car crash (Andrews et. al. 2011), Boeing 737 MAX fatal crashes due to MCAS software issue (Dominici G. & Lisi D. 2020), Heartbleed bug in OpenSSL cryptographic library (Durumeric, Z. et al. 2014). and Equifax data breach due to Apache Struts vulnerability (Lee W. 2019). Defective products may harm end users, companies, and even entire society. Product harm crisis also impact the company brand value as end-users consumes negative information, builds negative attitudes toward the brand, company, and sometimes entire industry (Siomkos & Kurzbard 1994). During the PHC process companies incur the additional cost, enhance their current operations, and attempts to rebuild the reputation in long-run. Hence it is critical for stakeholders involved, including but not limited to company management and supply chain involved, to understand product harm crisis in details including underlying causes.

Software plays an increasingly important role in different stages of managing product harm crisis. Software defects, vulnerabilities, or malfunctions can contribute to product failures and harm. (Boehm B. W. 1987). The design and quality of software play a crucial role in preventing product harm. Communication is critical to handle a product harm crisis situation. Software tools facilitate effective communication internally and externally, aiding in crisis management. (Coombs W. T. 2014). Supply chain management software traces and identifies affected products and thus respond quickly to product harm incidents. (Chopra S. & Meindl P. 2007).

Further, software tools help companies in maintaining regulatory compliance, reducing any adverse legal impact during a product harm crisis situation.

The Software industry is very dynamic due to fast paced innovations happening around concepts, frameworks, tools, applications, and other capabilities. Software usage has increased applications due to explosive growth of data, cheaper storage, faster network and cost-effective infrastructure rentals in form of cloud computing. Software practitioners make lots of crucial decisions during different stages of software development lifecycle including software conceptualization, design and development, implementation, and post deployment stages. These decisions are made at different organizational levels. Decisions vary from requirements clarifications, implementation decisions during development, project management gating criteria, release management, portfolio management etc. Some of these software decisions at various organizational levels could be unsuccessful, leading to failure conditions and corresponding upstream negative impact.

Santos et. al. (2021), analyzed software failure causes including their type, programmatic context, layer, and source-code where these manifested. They observed that 84% software failure causes were categorized into memory addressing issues, responsiveness, and inadequate exception handling. Irrespective of the failed-code type, the major failure causes were related to programming.

These programming related failures are remain mostly hidden once these passes gating criteria of go/no-go decision at deployment stage, and later manifests into product harm situations depending upon severity levels.

Simone L. (2013) found that in the analysis of recalls reported from 2005 through 2011, 19.4% of medical device recalls were related to software. The recalls related to software were approximately 6% in 1980s. The increase is attributed to combination of various factors including increase in software complexity, increased coverage of business workflows using software, more products utilizing software, increased products' 'softwarization'. Understanding about how software-related problems can result in recalls provides businesses an opportunity to incorporate this information into their design and manufacturing and other business workflows to produce safer alternatives.

Simone L. (2013) explored the multiple terms that are used in literature to imply the role that software plays in recall process and thus potential product harm crisis situations. These terms include Software faults, Software Error, Software problem, software failure and Software related. As there is inconsistency of usage of these terms, he divided these recalls into two mutually exclusive categories – Software related recalls and recalls not related to Software. Software related recalls include a spectrum of reported problems such as when software is a contributing factor in crisis, when it is part of a poorly defined system level risk control measure, or when new software is released as part of corrective or preventive action for non-software error. Table 1 summarizes sub distinctions for the term 'Software Related':

Table 1: Sub distinctions for term 'Software-Related'

Term	Explanation
Software-related and due to a software error	If the cause is attributed to software error or includes a software error
Software-related	If the causal analysis results into causes not due to software error but somehow software is involved. Or if the cause is not known but software is somehow involved.

Objectives of the Study

The current study aims to understand the role of software in product harm crisis situations. As nowadays software has ubiquitous presence in end-to-end product development, the study aims to understand software related product harm crisis situation. It aims to understand the common causes of those software failures. Based on this information, the study develops a framework to handle software related product harm crisis situations.

Research Problem

Typical PHC research is typically mapped to one of the 3 phases pre-recall, during a product recall and post-recall period. Considerable time is lost between recall decision, actual product recall and moment the product is considered suitable for use again **(Van Heerde, 2007).**

The current research focuses on understanding the causes, impacts and mitigation strategies for software-related product harm crisis. The study utilizes the characteristics of software development processes, lifecycle, tools and technologies to enable that. This research attempts to answer the following research questions:

- What are the primary causes of software-induced product harm crises across different industries?
- What mitigation strategies can be implemented to prevent and handle software related product harm crisis?

It spans across all three phases as describes above.

Research Methodology

The current research utilizes case studies based qualitative research method for in-depth exploration. It is based on secondary sources from academic literature, industry reports and case studies published for software related product harm crisis. Case studies are selected from different industries for the products where software defines the core functionality or assisted functionality to cause severe damage in case of failures. It utilizes qualitative research method such as Content Analysis to systematically study the content of mainly textual information. There are 7 coding schemes used to capture the key themes in the content.

Research Design and Approach

The current research utilizes mix-methods research design. This approach enables detailed exploration of software related product harm crisis. The Content Analysis research method uses 7 coding schemes:

- Interface inconsistencies
- Usage and Context
- Algorithmic Errors
- Error Detection and Recovery Mechanism
- Data Drift
- Maintenance and Upgrade
- Unforeseen Events

The following case studies are also utilized for this exploration:

- Boeing 737 MAX fatal crashes due to MCAS software issue
- Equifax data breach due to Apache Struts vulnerability

Research Findings

For Research Questions

Research questions Q1 and Q2 are studied with the help of available research literature, software industry trends and best practices from software product development fields. Further applicability of these findings were applied to both of the identified cases studies.

Research Question, Q1: What are the primary causes of software-related product harm crises across different industries?

Based on the Content Analysis, the repeatedly occurring themes are tabulated. The Table 2 below, provides a summary of problems themes resulting into software related failures.

Table 2: Common Problem Themes Causing Software related Failures

	Coding Schemes	Identified Themes
1	Interface inconsistencies	Incorrect / mixed / missing data, Database access issues, data refresh issues, device / module interfaces and communications
2	Usage and Context	Missed / incomplete / incorrect workflows or state of operations
3	Algorithms errors	Incorrect or inadequate calculations, longer delays leading to timeout, insufficient or wrong domain assumptions, Biases specially for Machine learning and AI implementations
4	Error Detection and Recovery Mechanism	Inadequate or missing error detection and recovery implementations, lack of mitigations for non-software related vulnerabilities.
5	Data Drift	Change in user interaction with product, deployment environment, additional interfaces
6	Maintenance and Upgrade	Installation of incorrect software version, configuration data or calibration data. Insufficient software installation or upgradation procedures, loaded / installed software does not run
7	Unforseen events	Hacking, cybersecurity breaches, critical service outages

Research Question, Q2: What mitigation strategies can be implemented to prevent and handle software related product harm crisis?

To mitigate software related product harm crisis, Software Related Product Harm Crisis Management Framework (SR-PHC-MF) is created. The framework amalgamates the best practices from Software practice with product harm crisis stages. This enables robust software offering component, preventive care and early detection of issues, and quick response to product harm crisis events to minimize negative impact on product, brand, company and industry. By responding in a transparent manner with required alacrity both reputation and monetary losses can be contained. Table 3 summarizes the Software Related Product Harm Crisis Management Framework (SR-PHC-MF).

Table 3: Software Related Product Harm Crisis Management Framework (SR-PHC-MF)

			Practices	Description
Bidirectional Traceability through Connected Tooling	Industry Best Practices supported by standardized Artefacts and Templates	Data Driven Robust Governance promoting Transparency	**Preventive Measures**	
			Robust Development Practices	Application of industry recognized best practices and coding standards
			Through Test and Quality Assurance Planning	Identification of rigorous testing methodologies, exit criteria, adoption of tool-based automation for sanity and quick regression.
			Adoption of Secure Development Lifecycle	Adoption of security practices throughout development lifecycle, including Risk assessment for requirements, threat modeling and Design review, Static analysis, Security Testing and code review, Security assessment and secure configuration, and monitoring and real-time handling for security events. Adoption of DevSecOps, which is a security-conscious adoption of DevOps.
			Legal and Compliance Adherence	
			Legal Preparedness by Design	Early sign-off of legal compliance requirements by legal experts, review of legal compliance controls at each exit phase, inclusion of cross geographies legal experts during product acceptance testing. Plan to acquire conformance certificates for geographies of operations.
			Data Protection Mechanism	Adherence to data compliance requirements specific to geographies of operations particularly ensure industry best practices for Personal Identification Information (PII), data in motion and data at rest. Stronger encryption levels are highly recommended.
			Rapid Response and Resolution	
			Incident Response Team	Identify roles, names, key contact information, Tools and Standard Operating Procedures (SOP) along with SLAs, thresholds, and escalation mechanism. Include participation across departments and roles. Create backup of critical roles.
			Agile Recovery Strategies	Conduct frequent pivots around approaches based on response data received from end users. Create customizable templates, document explicit success criteria as part of selected product harm crisis handling approach. Utilize common dashboard for status reporting with no or minimal manual intervention.
			Early Detection and Monitoring	
			Continuous Monitoring	Implement monitoring tools with thresholds actions configured. These should be done for both leading and lagging measures.
			User Feedback Mechanism	User feedback, from different channels, are fed into common dashboard. Proactive monitoring probes are implemented which proactively tracks change in users' sentiments across channels. In case of a structured feedback, ensure users are educated to avoid false alarms.
			Effective Communication and Transparency	
			Crisis Communication Plan	Create clear communication protocols including chain of command and escalation process. Establish communication channels for both internal and external stakeholders. Prepare pre-approved templates and key messages for different crisis situations. Have a robust media relations plan in place. Identify key stakeholders, including customers, suppliers, and community members. Develop targeted communication strategies for each stakeholder group.
			Users Communication Channels	Monitor all user communication channels including customer portal, user feedback platforms, product forums etc. Users should be communicated back minimally through the same channel. In case of multi-homing, deployed software tools should be able to provide consolidated and consistent view across all such channels.

			Post Incident Analysis and Learnings	
			Root Cause Analysis (RCA)	Tools like fish-bone analysis, 5-Whys etc. to be used to understand root cause of Product Harm Crisis event. RCA helps to fine-tune the current crisis response or predict / avoid / delay future Product Harm Crisis events.
			Continuous Improvements	Document the context of Product Harm Crisis event, root cause, impact, and learnings. Share it with stakeholders. These must be part of Organization wide Knowledge Management in searchable form.

Application of Research Findings to Case Studies

- **Case Study 1**: Boeing 737 MAX fatal crashes due to MCAS software issue

Case Details: Two crashes related to Boeing 737 MAX aircrafts happened within 6 months period. The Lion Air Flight 610 crashed in Indonesia on 29th October 2018, while Ethiopian Airlines flight 302 crashed on 10th March 2019. All passengers and crew members lost their lives in both these cases.

Key Findings: Both these crashes were found to be linked to Maneuvering Characteristics Augmentation System (MCAS). MCAS is a flight control software system responsible to automatically adjust the horizontal stabilizer trim to prevent a stall. This system relies on the data from a single Angle of Attack (AoA) sensor. This sensor measures the angle between the aircraft nose and oncoming air.

Following issues were held responsible for these crashes:

- Overreliance on a single sensor: If this sensor is faulty then MCAS could push aircraft's nose down.
- Multiple nose-down commands in short period of time: If MCAS read high AoA, then it issued multiple nose-down commands in short period of time, Pilot could not act so fast to counteract.
- Pilot Awareness: MCAS was not covered in flight manuals, not specific training was provided to pilots in this case for this feature.

Applicable Coding Schemes from Findings

Interface inconsistencies, Usage and Context, Error Detection, and recovery mechanism.

- **Case Study 2**: Equifax data breach due to Apache Struts vulnerability

Case Details: Equifax is one of the major credit reporting agencies. The data breach impacted nearly 147 million people. The sensitive financial and personal information including names, SSN numbers, birth dates, addresses, driver's license numbers etc. were compromised.

Following issues were held responsible for this data breach and loss of financial and personal information:

- Equifax used Apache Struts which is an open-source framework for developing Java based web applications. This version of Apache Struts had a vulnerability identified by tag 'Apache Struts CVE-2017-5638'. Attackers exploited this vulnerability to execute arbitrary commands on the web servers of Equifax. Though the breach took place in May 2017, Equifax could not discover it until July 2017.
- Equifax was aware of the Apache Struts vulnerability and had been notified about the available patch by Apache Software Foundation. But it did not apply this security patch which was marked necessary.

Applicable Coding Schemes from findings: Usage and Context, Maintenance and Upgrade, Unforeseen Events

Discussion and Conclusion

The current research provides valuable insights into challenges faced with software related product harm crisis situations. It expands the Product Harm Crisis to include damages beyond physical harm. It enables study of contemporary issues like data loss/leakages, cyber-attacks, Machine learning and artificial intelligence specific challenges like biases and data drifts etc. The explored causes are tabulated, and SR-PHC-MF provides actionable recommendations for industries, policymakers and software developers to enhance reliability and safety of such software products to identify current product harm conditions. and avoid possible future occurrences.

Recommendations and Scope for Further Research

Current study expanded the definition of Product Harm Crisis to take care of contemporary issues which directly don't cause physical harm but exposes consumers to life changing scenarios. Future research may expand to cover more cases studies from more industries. They may also evaluate if attribution for harm is significantly changed based on consumer knowledge of software cause for the harm. More dedicated studies may also be conducted on specific problems for the impact of contemporary issues like data drift or algorithmic biasness.

Références

1. Heath, R. L., & Palenchar, M. J. (2000). The importance of interpersonal communication in crises. In R. L. Heath & M. J. Palenchar (Eds.), Crisis communications: Lessons from September 11 (pp. 185-194). Rowman & Littlefield.
2. Sharma, C. (2022). Tragedy of the Digital Commons. North Carolina Law Review, Forthcoming.
3. Mitroff, I. I., Pauchant, T. C., & Shrivastava, P. (1987). Conceptualizing and measuring crisis-induced stress. Journal of Behavioral Medicine, 10(3), 235-266.
4. Barocas, S., & Selbst, A. D. (2016). Big Data's Disparate Impact. California Law Review, 104, 671. Available at SSRN or http://dx.doi.org/10.2139/ssrn.2477899
5. Boehm, B. W. (1987). Improving Software Productivity. ACM Computing Surveys (CSUR), 19(3), 189–204.
6. Birsch, D., & Fielder, J. H. (1994). The Ford Pinto Case: A Study in Applied Ethics, Business, and Technology. State University of New York Press. ISBN: 978-0791422347
7. Chehreghani, M. H., Rahgozar, M., Lucas, C., & Chehreghani, M. H. (2007). A heuristic algorithm for clustering rooted ordered trees. Intelligent Data Analysis, 11(4), 355–376. doi:10.3233/ida-2007-11404
8. Chopra, S., & Meindl, P. (2007). Supply Chain Management: Strategy, Planning, and Operation. Pearson.
9. Coombs, W. T. (2014). Ongoing Crisis Communication: Planning, Managing, and Responding. Sage Publications.
10. Dominici, G., & Lisi, D. (2020). The Boeing 737 MAX and the Crisis of the Aviation Duopoly. Journal of Strategic Marketing, 1-12.
11. Durumeric, Z., et al. (2014). The Matter of Heartbleed. Proceedings of the 2014 Conference on Internet Measurement Conference, 475-488.
12. Indonesian National Transportation Safety Committee (KNKT). (2019). Final Report: Lion Air Flight 610.
13. Ethiopian Aircraft Accident Investigation Bureau (AIB). (2020). Final Report: Ethiopian Airlines Flight 302.
14. Lee, W. (2019). Equifax Data Breach: An Overview. Journal of Information Privacy and Security, 15(1), 56-66.
15. National Transportation Safety Board (NTSB). (2019). Boeing 737 MAX: Overview of the Lion Air Flight 610 Accident.
16. National Transportation Safety Board (NTSB). (2019). Boeing 737 MAX: Overview of the Ethiopian Airlines Flight 302 Accident.
17. U.S. House Committee on Transportation and Infrastructure, Majority Staff. (2020). Boeing 737 MAX: Ensuring the Safety of the Commercial Aviation Fleet.
18. Equifax. (2017). Equifax Announces Cybersecurity Incident Involving Consumer Information.
19. Federal Trade Commission (FTC). (n.d.). The Equifax Data Breach: What to Do.
20. U.S. House of Representatives, Committee on Oversight and Government Reform. (2018). Report on the Equifax Data Breach.
21. Chopra, S., & Meindl, P. (2007). Supply Chain Management: Strategy, Planning, and Operation. Pearson.

□❍□