Whistleblowing in Software Engineering

A Study of Interactions and Escalations in Whistleblowing Situations

Lucy Jane Hunt

This dissertation is submitted for the degree of Doctor of Philosophy

September 2024

School of Computing and Communications

" If you want to understand the big issues, you need to understand the everyday practices that constitute them."

Lucy Suchman

Declaration and Publications

I declare that the contents of this thesis, unless otherwise referenced, is all my own work and ideas. Any elements of this work which have been published, or are the outcome of research collaborations, are stated at the beginning of the thesis chapter where the content resides. This thesis has not been submitted for the award of a higher degree anywhere other than the Lancaster University, and for any degree other than for the degree of Doctor of Philosophy.

Lucy Jane Hunt

September 2024

Excerpts of this thesis have been published in the following conference:

Hunt, L. and Ferrario, M.A., 2022, January. A Review of How Whistleblowing is Studied in Software Engineering, and the Implications for Research and Practice. In *2022 IEEE/ACM 44th International Conference on Software Engineering: SEIS-Software in Society (ICSE 2022): Proceedings*. IEEE.

Additional publications

Winter, E., Forshaw, S., Hunt, L. and Ferrario, M.A., 2019, May. **Advancing the study of human values in software engineering**. In *2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)* (pp. 19-26). IEEE.

Winter, E., Forshaw, S., Hunt, L. and Ferrario, M.A., 2019, May. **Towards a systematic study of values in SE: tools for industry and education.** In *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)* (pp. 61-64). IEEE.

Armknecht, F., Verbauwhede, I., Volkamer, M. and Yung, M., 2019. **Biggest Failures in Security.** In *Dagstuhl Seminar 19451, Location: Wadern, Germany*. (Participant)

Abstract

High-profile incidents and scandals involving software at Boeing, Volkswagen and the UK Post Office have resulted in loss of life, environmental damage, and societal harm. Why were there no apparent whistleblowers during the development of these systems, why did issues only come to light after the systems were in production? Whistleblowing can be described as a public interest disclosure about organisational wrongdoing or harm. Organisations have a responsibility to demonstrate effective mechanisms for detecting, evidencing, mitigating, and speaking up if professional values, practices, or standards are breached on IT projects. Specifically, the ACM Code of Ethics advises software professionals to "blow the whistle" if leaders do not act to mitigate risks of harm. Recent stories in the media demonstrate that software professionals with insider knowledge of issues at Google, Uber, Twitter, and Facebook do take individual and collective action to disclose wrongdoing and harm, often at great personal cost.

My published literature review finds software engineering research lacks empirical in-practice studies of whistleblowing, reflective of the frequency, sensitive nature, and obtrusiveness of studying such events. In this thesis I ask, "*why and how do software engineers blow the whistle*?" and report on actions taken (or not taken) by software engineering professionals to mitigate harm and wrongdoing in software engineering practice. I use existing whistleblowing research and theories to inform the development of a Whistleblowing in Software Engineering (WISE) analysis framework. The framework guides the systematic analysis and abstraction of story data from interviews. Important findings, specific to software engineering practice, are presented through cases in the health, transport, and nuclear industries; my findings are discussed in light of existing whistleblowing research. Key findings relate to tampering with software artefacts to remove, disguise, or leave evidence of issues, and the creating of team and organisational secrets. The cases notably evidence issues being suppressed or covered up by management, in breach of regulatory standards and compliance processes. I find practitioners motivated to uphold professional values and standards despite the negative consequences for themselves. Some experienced practitioners seek help from professional and regulatory bodies to mitigate concerns; some less experienced staff keep quiet, raise issues discretely with colleagues, or are threatened into complying with management wrongdoing. The cases confirm findings reported in laboratory studies and call for further in-practice studies with researchers given timely access to stakeholders and software artefacts linked to recent or emerging whistleblowing situations.

Acknowledgements

To my anonymous participants – for their patience and generosity of time.

Contents

List of Tables

List of Figures

List of Names, Abbreviations and Acronyms

CERT	Computer Emergency Response Team
ACM	Association of Computing Machinery (Professional Body)
NHS	National Health Service (UK)
Ipsos	Polling and market research company. Known as “Ipsos MORI” prior to February 2022.
BCS	British Computer Society
IEEE	Institute of Electrical and Electronics Engineers
IET	Institute of Engineering and Technology
WISE	Whistleblowing in Software Engineering
EBM	Escalation boundary model
ORID	Objective-Reflective-Interpretive-Decision facilitation

Chapter 1. Introduction

Science and technology, underpinned by software, brings benefits and improves the quality of our lives in many domains from health, transport and government to energy, finance, and communication. Software can also cause harm. These harms may be intentional (malicious or illegal software), caused by negligence (poor practices leading to defective or vulnerable software) or from unforeseen or unplanned outcomes of use. The ACM Code of Ethics (ACM Council, 2018) [the Code] calls for professionals to reflect on the wider impact of their work in supporting public good and public interest. Specifically, that harm should be avoided, especially when the consequences are significant and unjust. A non-exhaustive list of harms, given by the Code, includes physical or mental injury, destruction or disclosure of information, damage to property, reputation, and the environment. The Code additionally states, "*a computing professional has the obligation to report any signs of systems risks that may result in harm*" and that "*if leaders do not act [...] it may be necessary to "blow the whistle*". The Code reflects that capricious or misguided reporting can be harmful too and that computing professionals should carefully assess relevant aspects of the situation before reporting risks. While whistleblowing can reveal issues and mitigate harm, encouraging such actions is inappropriate without understanding the impact and effectiveness to change a situation. This thesis investigates actions, up to and including whistleblowing, taken by software professionals to mitigate harm and uphold professional values and standards in software engineering practice. The UK government[1] states whistleblowers can report the following types of situations they believe happened, are happening, or are to be happening soon:

- a criminal offence, miscarriage of justice or breaking the law,
- health and safety of someone is in danger,
- risk or actual damage to the environment,
- covering up wrongdoing.

In Table 1-1 I construct a definition of whistleblowing for software engineering, anchored on professional codes of conduct (e.g., (ACM Council, 2018)).

Table 1-1 Definition of Whistleblowing in Software Engineering

Whistleblowing in Software Engineering: The process of making a responsible disclosure, internally or externally, of actions and artefacts perceived to be contrary to accepted professional values and standards in the software development lifecycle, carried out to mitigate wrongdoing or harm to self, others, or wider society (Hunt and Ferrario, 2022).

[1] https://www.gov.uk/whistleblowing

1.1 Terminology

The terminology in this thesis is guided by Stol and Fitzgerald's Software Engineering Research ABC Framework (Stol and Fitzgerald, 2018), ISO standard (IOS (International Organization for Standardization), 2021) and the UK government whistleblowing situations. The terminology is summarised in Table 1-2. The ABC framework defines research goals in terms of Actors (A) and the measurement of their Behaviour (B) in a specific research Context (C). ISO standard 32007 (for whistleblowing management systems) defines whistleblowing. The main actors (participants) in my research are software engineering experts and their reporting on the behaviours of individuals (including themselves), roles, teams, and organisations involved in harmful situations with public interest aspects. Other actors studied are software systems and artefacts, (the software engineering aspects) and their role in the situation being studied.

Table 1-2 Terminology Summary

Term	Definition
Actor	Any software engineering artefacts and stakeholders (individuals, teams, and organisations). With a focus on software professionals, software systems, and those creating or impacted by software systems.
Behaviour	The activities, actions, and behaviours of actors (human or systems) measured as part of the research.
Context	The setting of actors (systems or organisations) being studied (e.g., laboratory, field, neutral)
Situation (Harm or Wrongdoing)	Public interest situations that include criminal offences, miscarriages of justice, breaking the law, health and safety of someone, environmental damage or the covering up of wrongdoing in the past present or future.
Whistleblower	A person who reports wrongdoing with reasonable belief the information is true at the time of reporting. Reasonable belief is based on observation, experience or information known to individual, which would also be held by another person in the same circumstances.
Whistleblowing	Reporting of suspected or actual harm and wrongdoing. *Open*: disclosure includes whistleblower identity *Confidential*: whistleblower identity not disclosed without their consent *Anonymous:* disclosure without disclosing whistleblower identity.

1.2 Reporting on Insider Threats

This thesis specifically focuses on how actors involved in software engineering practice respond to perceived breaches of values and standards inside organisations where software is being developed. These situations can be described as "*insider threats*" to organisations, whereby someone maliciously, negligently, or unintentionally acts in a way that could negatively affect the organisation. In their book, The Dark Side of Software Engineering (Rost and Glass, 2011), Rost and Glass review wrongdoing and harm observed while studying IT projects. Rost and Glass find that while there are many studies on the cause and effect of whistleblowing, very few are found in the software

engineering literature, and specifically highlight the lack and difficulty of finding case studies. My research seeks to discover if and how whistleblowing in software engineering practice happens and presents the findings in the form of case studies.

The CERT® Guide to Insider Threats (Cappelli, Moore and Trzeciak, 2015) describes "*technology crimes*" of intellectual property theft, sabotage, fraud, espionage, unintentional incidents, and misuse that puts organisations and the public at risk. CERT® specifically refers to "*software lifecycle development interference*" and discusses the threats from those creating or operating software that is defective, insecure, malicious, or illegal. CERT® guides organisations to consider mitigation strategies for software development lifecycle threats when developing and maintaining systems internally, or when implementing systems acquired from elsewhere. CERT® advises that employees should be able to report "*suspicious events*" without fear of repercussions and so over-come cultural barriers of whistleblowing. Despite legal protection for whistleblowers, their actions often come at great personal cost, with reports of victimisation, being ostracised and feeling forced to leave a workplace (Fitzgerald, 1990; Alford, 2007). Internal or external whistleblowing may cause operational disruption and reputational damage (Near and Miceli, 1996). Both the ACM and CERT® advise *what* should happen (whistleblow to report risk or harm) but do not address the complexity of assessing and deciding *when, how* and to *whom* this reporting should happen.

1.3 Insider Threats and Actions to Mitigate Harm

The ACM describe "*harm*" as negative consequences, especially when those consequences are significant and unjust. The ACM give examples including physical or mental injury, destruction or disclosure of information, and damage to property, reputation, and the environment. The ACM also describes how well-intended actions, including those that accomplish assigned duties, may also lead to harm. When that harm is unintended, those responsible are obliged to undo or mitigate the harm as much as possible. To minimise the possibility of indirectly or unintentionally harming others, computing professionals are advised to follow generally accepted best practices unless there is a compelling ethical reason to do otherwise.

In 2019, Ipsos (known as "Ipsos MORI" prior to February 2022) conducted an online survey of over 1000 UK computing professionals on behalf of Doteveryone (Miller and Coldicott, 2019). Doteveryone were an independent thinktank that explored how technology is changing society. Doteveryone's report based on the Ipsos data, "*People, Power and Technology: The Tech Workers' View"* was the first in-depth research into the concerns of people working across a range of technology roles in the UK. Ethical consent was granted, by Lancaster University's Faculty of Science and Technology Research Ethics Committee, to analyse the Ipsos dataset. This next section discusses the Ipsos findings.

1.3.1 Harms Seen

The Ipsos survey found 28% of respondents had witnessed decisions about a technology that could have negative consequences for people or society. Concerns included the development of products causing addictiveness, job losses or end user isolation, alongside software development lifecycle issues leading to failures in safety, security, and testing practices. The Ipsos findings are examples of insider threat situations warned

about by CERT® and the ACM, which could put individuals, organisations, and the public at risk of harm. Based on the Ipsos findings, Doteveryone reported that tech workers wanting guidance and skills to help navigate new dilemmas, have an appetite for more responsible leadership and want clear regulation so they can innovate with awareness.

1.3.2 Actions Taken After Witnessing Harm

Table 1-3 presents a summary of actions taken by the 287 (28%) Ipsos survey participants who reported witnessing harmful situations. The Ipsos study does not capture situational data to a sufficient level of granularity to understand behaviours or actors (the sequence of events, decisions, and actions taken to report up a harmful situation) nor the detail of the discovery or evidencing the harm. Participants could select more than one action option. The actions range from taking no action (10%), talking to a colleague (54%), to reporting to an external body (29%). The 47% that raised their concerns internally may have done so through formal project processes or through corporate (internal) whistleblowing processes. The 29% reporting concerns externally could be described as having taken external whistleblowing actions; they may have taken internal actions first.

Table 1-3 Actions After Harmful Decisions (Miller and Coldicott, 2019)

Actions taken by tech workers witnessing potential harm (N=287 Participants could select one or more of the actions)	**%**
Raised concerns with a colleague	54
Raised concerns to manager or Human Resources	47
Reported concerns to external body	29
Considered leaving the company	28
Left the company	18
(Took no action)	10

It is reported that 18% of participants left their company, though there is no detail as to what the circumstances were, nor the actions taken before or after leaving. People choosing to leave an organisation is a significant concern for teams and organisations, particularly if people leave without reporting the potential harm. Feeling forced to leave a workplace is reported in general whistleblowing studies (Fitzgerald, 1990; Alford, 2007), the Ipsos data indicates this may also happen in computing professions.

The Ipsos survey is predominantly quantitative data, with summaries of the types of actions a subset of 287 respondents took on witnessing a potentially harmful situation at work. Statista, using data from the UK's Office for National Statistics, reports there were over 408,000 "*programmers and software development professionals*" in the UK in 2020 (Statista, 2021), a time close to when this survey was run. If we extrapolate the Ipsos survey data against the known population of software development professionals, 28% (114,000 people) may have seen harmful issues in their careers to date.

The impact of staff turnover and the resources to replace and train new staff should not be underestimated by organisations (Oxford Economics, 2014), nor the wider impact that unreported harmful software can have on individuals and society. It is concerning that 10% of people witnessing a harmful situation take no action to report it. Existing sociology research suggest people may take no action because they do not have the

necessary power or influence to change a situation (Near and Miceli, 1996) or that it is not their responsibility to report it (Latané and Darley, 1970).

My research reports on the sequence of events, decisions, and actions taken when software engineering actors raise concerns about harm or wrongdoing in software engineering practice.

1.3.3 Outcomes of Actions

Table 1-4 shows Ipsos participant satisfaction with outcomes of raising their concerns, finding most were somewhat or very satisfied with the outcome. While 90% of participants do take some (one or more) actions, without the actual situation of the observed harm this data set gives no insight into *how* a harmful situation is discovered, escalated, and if it is successfully resolved and to what level of satisfaction.

Table 1-4 Raising Concerns and Satisfaction (Miller and Coldicott, 2019)

Satisfaction on raising or reporting concerns	%
Don't know	0.5
Very unsatisfactory	5.5
Somewhat unsatisfactory	4
Neither satisfactory nor unsatisfactory	11.5
Somewhat satisfactory	31
Very satisfactory	47

1.3.4 Summary and Research Gaps

With the potential size, complexity, and distributed nature of IT systems there is a complex chain of authority and responsibility involved in the creating and operating of software systems. Internal processes for raising concerns or issues, such as retrospectives, appraisals, complaints or project and quality audits, exist as formal or obligatory mechanisms for reporting up problems that could precede actions linked to whistleblowing. Coeckelbergh concludes that challenges of awareness and evidencing harm should not prevent individuals and organisations looking at their collective moral responsibilities to society; to speak up, look backwards and forwards from situations, and inspire better practices (Coeckelbergh, 2012). To support such calls for action requires the software engineering community, researchers, and practitioners, to come together and study harmful situations and how they came to be. To look in detail and analyse whistleblowing situations such that it can be used to support awareness raising within the software engineering community is challenging, as evidenced by the scarcity of such studies found in my literature review (Hunt and Ferrario, 2022) and as reported on by Rost and Glass (Rost and Glass, 2011).

1.4 Public Interest Technology-based Media Stories

High-profile scandals involving software at organisations such as Boeing[2], Volkswagen (Trope and Ressler, 2016), and the UK Post Office[3] have resulted in loss of life, environmental damage, societal and individual harm. Heightened awareness of such

[2] https://spectrum.ieee.org/aerospace/aviation/how-the-boeing-737-max-disaster-looks-to-a-software-developer
[3] https://www.postofficetrial.com/2020/01/horizon-trial-judgment-is-handed-down.html

stories from the media, campaign groups, public inquiries and court cases has led to reputation damage, economic losses, fines, and increasing calls for transparency and regulation of the software industry and community on several levels. This section discusses the harms found in a selection of media stories where the cause has, in part, been attributed to software and software engineering practices. The stories provide examples of behaviours and situations, warned about by CERT® and the ACM, which put organisations and the public at risk of harm.

Behaviour: negligence and hiding technical issues causing individual harm, economic losses, and societal impact: For over 20 years, UK campaign groups and journalists have reported on a scandal at the Post Office Ltd[4]. Post Office branches were provided with Horizon branch accounting software that, it has now been revealed, caused discrepancies in branch finance records. Prosecutions for fraud, false accounting and theft were brought against hundreds of branch postmasters leading to jail sentences, bankruptcy, suicides, ruined careers, and reputation in the community. Convictions were based on the wrong assumption that the Horizon IT were working correctly. The Post Office Ltd. and Fujitsu (the software supplier) were subsequently found to be aware of, but covered up, known errors that caused the discrepancies and instead blamed postmasters and their staff. The UK government, the Post Office's only shareholder, is compensating groups of former Post Office workers from a one-billion-pound compensation fund. There are ongoing inquiries and criminal investigations against the Post Office and Fujitsu IT staff that extend beyond the completion of this thesis. The inquiry is still ongoing, recently investigating the known amount of internal evidence regarding technical issues with the IT system. A 2024 ITV drama (Mister Bates vs The Post Office[5]) has brought the issues and the inquiry to the attention of the general public more so than twenty years of investigations and campaigning.

Behaviour: Intentional code causing environmental harm and societal impact: In 2015 American researchers evidenced how software was being misused by Volkswagen to detect and then defeat regulatory emissions tests, allowing nitrogen oxide emissions up to 35 times higher than permitted by the US standard (Trope and Ressler, 2016). Volkswagen chose to deceive the regulators rather than solve the lowering emissions problem. Volkswagen's senior management initially claimed to have no knowledge of the malpractice and initially blamed rogue engineers. A senior Volkswagen software engineer and executive was jailed for their role in the scandal[6,7] along with worldwide inquiries and court cases resulting in massive fines, customer compensation and a new whistleblowing system developed for Volkswagen employees and third parties.

Behaviour: Intentional code hiding fraud, causing individual and societal economical losses: In March 2009, Madoff pleaded guilty to 11 federal crimes and admitted to operating the largest private Ponzi scheme in history. Madoff was sentenced to 150 years in prison[8] and died there in 2021[9]. Madoff's IT experts George Perez and

[4] https://www.postofficetrial.com/2020/01/horizon-trial-judgment-is-handed-down.html
[5] https://en.wikipedia.org/wiki/Mr_Bates_vs_The_Post_Office
[6] https://www.theguardian.com/business/2016/sep/09/volkswagen-engineer-pleads-guilty-conspiracy-emissions-scandal-
[7] https://www.theguardian.com/business/2017/dec/06/oliver-schmidt-jailed-volkswagen-emissions-scam-seven-years
[8] https://www.theguardian.com/business/2009/jun/29/bernard-madoff-sentence
[9] https://www.theguardian.com/us-news/2021/apr/14/bernie-madoff-dies-prison-ponzi-scheme

Jerome O'Hara were also jailed. The American Securities and Exchange Commission (SEC) alleged the defendants used their computer skills to produce false documents and trading records to hide the fraud. It is also alleged they took bribes in return for their silence during their 20-year service for Madoff. "*Without the help of O'Hara and Perez, the Madoff fraud would not have been possible*," said George Canellos, director of the SEC's New York office [10] . "*They used their special computer skills to create sophisticated, credible and entirely phony trading records that were critical to the success of Madoff's scheme.*"

Behaviour: Negligent practices and unplanned outcomes causing societal and individual harm: In the UK, in 2020, automated decision-making software came under scrutiny after a "*mutant*" algorithm used by an exam regulator, (Ofqual), downgraded almost 40% of the A-level grades as assessed by teachers, culminating in the system being scrapped [11] . Other automated decision systems used by UK councils and government bodies for welfare benefits and visa applications have also been "quietly withdrawn" due to poor outcomes[12]. The reasons given ranged from problems in the way the systems worked to concerns about negative effects and bias concerning end users.

As a former software developer, I wonder at what point in the software development lifecycle were issues in the above stories first identified and by whom? How did actors (individuals, teams, and organisations) behave (act, react or intervene) and were there attempts made to resolve or cover issues up? How many situations were observed but went undisclosed, and so in effect were suppressed and became an organisational secret. In all the above stories, questions must be asked of those developing and supporting the systems, but also the behaviours and role of auditors (internal and external) and regulators. Finance, aviation, and automotive industries are highly regulated, yet situations in these industries went unreported for years. Keil and Robey's research (Keil and Robey, 2001) evidence auditors being "at the behest of those that employ them", and not reporting up failing projects as doing so could threaten their future career. My research presents case studies that explore such issues and includes incidents detected by or that should have been reported to internal compliance officers, external regulators, or professional bodies.

1.5 Whistleblowing Stories in Software Engineering

In 2011, Rost and Glass published a book, The Dark Side of Software Engineering: Evil of Computing Projects with a collection of studies from their many years of field-based research (Rost and Glass, 2011). The authors reflect on the apparent lack of whistleblowing in software engineering, despite their research and anecdotes reporting on a broad range of serious project issues from subversion, lying, hacking, and theft of information through to espionage, disgruntlement, and sabotage. Vandekerckhove (Vandekerckhove, 2012) finds whistleblowing from the technology community is rare, when compared to an increasing number of cases reported amongst health, education, and financial professionals. This next section presents a selection of high-profile media stories from the last 15 years evidencing circumstances where blowing the whistle, on

[10] https://www.theguardian.com/business/2009/nov/13/madoff-accomplices-jerome-ohara-george-perez
[11] https://www.theguardian.com/politics/2020/aug/26/boris-johnson-blames-mutant-algorithm-for-exams-fiasco
[12] https://www.theguardian.com/society/2020/aug/24/councils-scrapping-algorithms-benefit-welfare-decisions-concerns-bias

some level, was used to raise concerns and attempt to change outcomes. Internal mechanisms for resolving issues may have been exhausted or ineffective and are not presented in any detail in the news media stories, though subsequent autobiographies and articles do cover, in varying levels of detail, some of the approaches to mitigate harm tried.

In 2013, Edward Snowden leaked documents regarding mass surveillance by the American National Security Agency (Stein, 2013; Tavani and Grodzinsky, 2014; Ring, 2015). In 2018, over four thousand Google employees signed an open letter urging the company not to work on the Pentagon's drone warfare project[13]. A dozen Google employees went on to resign over Google's continued participation in the military project, citing ethical concerns of using artificial intelligence in drone warfare. In 2018 it was revealed that fifty million Facebook profiles had been accessed by Cambridge Analytica[14]. Christopher Wylie went on to reveal Cambridge Analytica's unlawful use of personal data for election campaigning in the USA. In 2020 Susan Fowler spoke to the media [15,16] and subsequently published a book about the experience of speaking up at Uber and how whistleblowing nearly ruined her career. By contrast, and despite significant media attention, no one internal from Volkswagen publicly raised concerns about software that defeated emissions tests (Trope and Ressler, 2016).

Since 2021, there have been a succession of whistleblowing stories by former Google and Facebook employees speaking out about known harms caused by features of their products and business models (e.g., algorithmic bias and putting profit before public good)[17]. In 2022, Mark MacGann, Uber's former chief lobbyist for Europe, revealed how Uber software supported their efforts to flout laws, deceive police, and lobby governments to develop its global business, now operating in over sixty-nine countries[18,19,20]. Uber defended their actions, saying their Greyball software protected their drivers from violent customers and undercover law enforcement activities. Most recently in the USA, in January 2023, Facebook allegedly fired George Hayward for speaking out about Facebook's "*negative testing*" practices[21] that allows companies to surreptitiously run down a user's battery in the name of testing features. Hayward identified that reducing phone battery life could put people at risk, especially in circumstances where communication with the police or rescue workers was necessary. Hayward reported that his manager justified that "*by harming a few we can help the greater masses*".

As a perceived solution to overcoming the fear of whistleblowing, organisations promote policies to protect whistleblowers with systems for securely disclosing and optionally anonymously raising concerns. Volkswagen introduced a new system following the emission detection scandal. In the UK health service, despite having a

[13] 'The Business of War': Google Employees Protest Work for the Pentagon - The New York Times (nytimes.com)
[14] Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach | Cambridge Analytica | The Guardian
[15] https://www.ft.com/content/4e9ce18c-6221-11ea-b3f3-fe4680ea68b5
[16] https://www.theguardian.com/world/2020/mar/01/susan-fowler-uber-whistleblower-interview-travis-kalanick
[17] https://www.theguardian.com/technology/2021/oct/08/tech-whistleblowers-facebook-frances-haugen-amazon-google-pinterest
[18] https://www.theguardian.com/news/audio/2022/jul/12/the-uber-files-the-whistleblower-part-2-podcast
[19] https://www.theguardian.com/news/audio/2022/jul/11/the-uber-files-the-unicorn-part-1
[20] https://www.theguardian.com/technology/2017/nov/21/uber-data-hack-cyber-attack
[21] https://nypost.com/2023/01/28/facebook-fires-worker-who-refused-to-do-negative-testing-awsuit/

well-publicised freedom to speak up policies[22] and systems, there are media reports of retaliation and "*hunting out*" of whistleblowers. For example, a hospital trust was found to be seeking fingerprints from staff to try and identify the author of an anonymous whistleblowing letter, leaving staff feeling mistrustful of systems and processes in place to protect them[23].

1.6 Storytelling in Software Engineering

My research explores stories from software engineering practice. Polkinghorne's narrative analysis paper describes how stories "*express a kind of knowledge that uniquely describes human experience in which actions and happenings contribute positively and negatively to attaining goals and fulfilling purposes*" (Polkinghorne, 1995). Lutters and Seaman (Lutters and Seaman, 2007) describe software engineering "*war stories*" as a form of qualitative data that capture informant accounts of surmounting great challenges and reporting on behaviours in a specific context. Recommendations are given that the rich contextual detail afforded by narratives in stories warrants their inclusion in the methodological tool kit of empirical software engineering research. The benefits could be significant for researchers seeking to understand software engineering practices and for wider society seeking an understanding of how software professionals uphold values and standards and endeavour to act with the public interest in mind. Visualising software stories has been explored by Kuhn and Stocker (Kuhn and Stocker, 2012). Kuhn and Stocker's work was in turn influenced by Ogawa and Ma's storyline approaches for visualising developer interactions with a project (Ogawa and Ma, 2010). Kuhn and Stocker and Ogawa and Ma find that storytelling has more impact when combined with artefacts that trigger memories of experiences, for example revisiting check in comments, source code and photos of development teams. Capturing and presenting data that represents artefacts, actor interactions, events and decisions along a timeline provides a rich data set through which to explore whistleblowing situations in software engineering practice. When technology stories break in the media, how the issues were discovered, and how they evolve and escalate inside or outside the organisation is rarely shared, denying software engineering practitioners and research community the opportunity to learn from it.

My case study research attempts to address this gap of understanding by investigating stories from the point of discovery of an issue. The research is conducted in neutral settings (online interviews) and works with participants to explore a whistleblowing situation they were directly involved with. My WISE framework, based on existing whistleblowing theories and studies, is used to analyse stories, and develop the three presented cases. Systematically analysing, abstracting, and presenting stories (actors and behaviours in specific whistleblowing context) into a structured case report enables stories from different whistleblowing contexts and industries to be compared.

1.7 My Research Objectives

After a 25-year career in the IT industry, my *perception* is that there is a lack of voices from the software engineering community regarding the handling of harmful situations

[22] https://www.england.nhs.uk/ourwork/freedom-to-speak-up/
[23] https://www.bbc.co.uk/news/uk-england-suffolk-59707114

in software engineering practice. I have worked as a software engineer, business analyst and an IT consultant. I have worked on well run and successful IT projects. I have also experienced IT projects, deemed successful by some, but with poor outcomes for software engineering teams, organisations, and end users (direct and indirect). I am motivated to systematically explore challenging real-life situations as opportunities for the reflection on and the improvement of software engineering practices. My research is guided by three key objectives, shown in Table 1-5 that in turn have several research questions each.

Table 1-5 Objectives, Research Questions, and Outputs

Objective	Research Questions	Output
1. To understand what is currently known about whistleblowing in software engineering literature	**RQ1:** What are the primary whistleblowing themes in the software engineering literature? **RQ2:** How is whistleblowing researched in software engineering? **RQ3:** What are the research gaps?	• Literature Review
2. To design a research framework to systematically capture, analyse and explain features of software engineering whistleblowing stories	**RQ4:** How can existing literature, social science theories and models inform a research framework for studying whistleblowing in software engineering? **4-1:** What structures can be used to represent Actors? **4-2:** What structures can be used to represent Behaviours? **4-3:** What structures can be used to represent Context?	• Escalation Boundary Diagram • WISE Analysis Framework • Interview Protocol • Case Study Template
3. To explore experience of upholding values and standards in software engineering practice, with actions up to and including whistleblowing	**RQ5:** How do software engineers whistleblow? **5-1:** What harms or wrongdoing is detected? **5-2:** What software engineering practice aspects are involved? **5-3:** What actions do actors take? **5-4:** What final outcomes are reported?	• Case Report • Case Profile • Story Wheels • Background research

1.7.1 Research Summary

Figure 1-1 presents a visual summary of how the research chapters contribute to the five research questions.

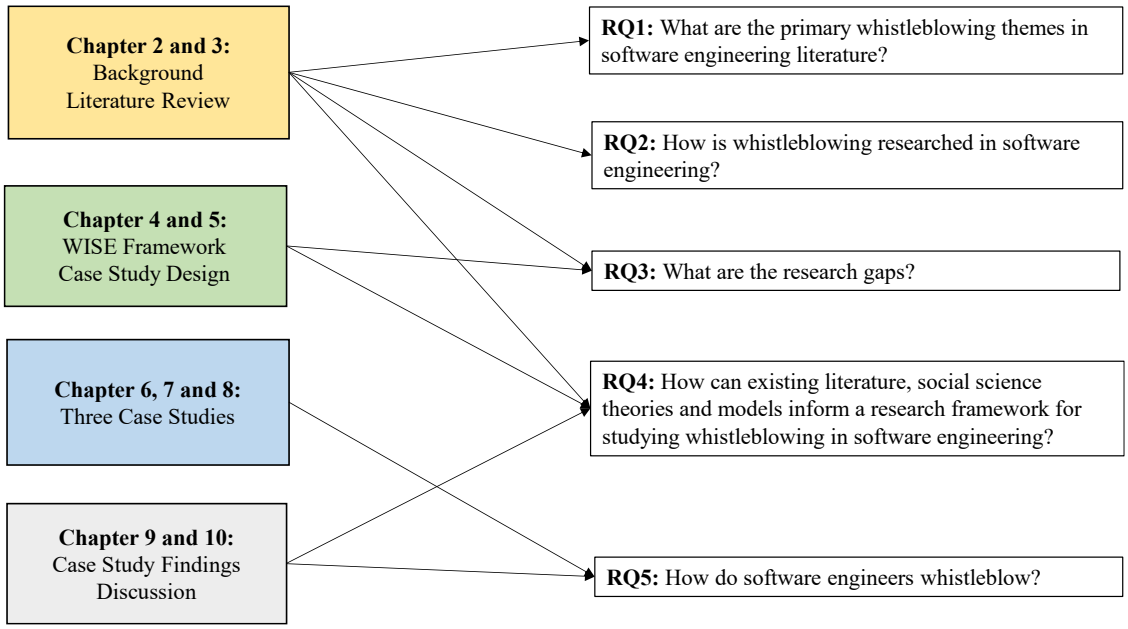


Figure 1-1 Research Chapters and Research Questions

The next three sections introduce each objective and research questions.

1.7.2 Objective 1

To understand what is currently known about whistleblowing in software engineering literature.

My first objective was to understand what is currently known about whistleblowing as found in the software engineering literature. Chapter 2 looks at the whistleblowing research landscape and constructs a definition of whistleblowing specific to software engineering practice. Chapter 3 is a literature review and answers the following research questions:

RQ1: What are the primary whistleblowing themes in the software engineering literature?

RQ2: How is whistleblowing researched in software engineering?

RQ3: What are the research gaps?

RQ1 reports on the primary themes of the review items. RQ2 reports on the actors, behaviours, contexts, and methods of empirical study items. RQ3 identifies research gaps based on the themes found in RQ1 and study types found in RQ2. Laboratory-controlled experiments have contributed to a better understanding of factors affecting the *intention* of whistleblowing on IT projects (Keil and Robey, 2001; Park and Keil, 2007; Park, Im and Keil, 2008; Park, Keil and Kim, 2008; Keil and Park, 2010; Schilhavy and King, 2010; Wang, Keil and Wang, 2015). However, field-based research into how whistleblowing actually happens in software engineering practice is rare. Findings from my literature review feed into Objective 2, RQ4 and the development of my Whistleblowing in Software Engineering (WISE) analysis framework in Chapter 4.

1.7.3 Objective 2

To design a research framework to systematically analyse and explain features of software engineering whistleblowing stories.

My second objective was to design a framework to support my research and specifically the analysis of software engineering stories. The resulting WISE analysis framework is a new approach to guide researchers how to systematically explore, capture and analyse whistleblowing stories across different domains.

In Chapter 4, findings from Chapter 2 (background), Chapter 3 (literature review) and Chapter 5 (research design) support RQ4 and address Objective 2:

RQ4: How can existing literature, social science theories and models inform a research framework for studying whistleblowing in software engineering?

4-1: What structures can be used to represent Actors?

4-2: What structures can be used to represent Behaviours?

4-3: What structures can be used to represent Context?

1.7.4 Objective 3

To explore experience of upholding values and standards in software engineering practice, with actions up to and including whistleblowing.

The final objective is to investigate software actor experiences of upholding values and standards in software engineering practice. Interviews are conducted with seven software engineering experts. The interviews are analysed in a structured and repeatable way using the WISE analysis framework. Three case chapters (Chapter 6, Chapter 7, and Chapter 8) present findings from the analysis and give insight into previously unstudied whistleblowing situations from software engineering practice. Chapter 9 presents a summary of the case study findings and answers RQ5. In Chapter 10 the key findings from the literature review and case studies are discussed.

RQ5: How do software engineers whistleblow?

5-1: What harms or wrongdoing is detected?

5-2: What software engineering practice aspects are involved?

5-3: What actions do actors take?

5-4: What final outcomes are reported?

1.8 Contributions to Knowledge

The prime objective of my thesis is the investigation and presentation of practitioner whistleblowing experiences in software engineering practice. My research is knowledge seeking about the phenomenon of whistleblowing in software engineering practice. I use case study research methods and my WISE analysis framework to systematically analyse actor interactions in public interest software engineering situations. My research focuses on data collection of the observed behaviours of software engineering actors (individuals, teams, and organisations). My contributions to software engineering research knowledge are:

Contribution 1: The first published literature review of how the phenomenon of whistleblowing has been studied in software engineering research. My literature review

finds a prevalent view that reporting harm is a matter of individual moral responsibility. I conclude whistleblowing research in software engineering appears rare (compared to other domains) and lacks field-based research into how whistleblowing happens (or indeed does not happen) in software engineering practice.

Contribution 2: The WISE analysis framework. An actor-behaviour based framework developed to guide the analysis of stories from software engineering practice. The framework allows software engineering whistleblowing stories to systematically be analysed, profiled, and compared.

Contribution 3: A set of three cases from the health, nuclear and automotive industries reporting on situations and whistleblowing actions taken by software engineering practitioners. The cases cover regulatory breaches, health and safety situations, damage to the environment and the covering up of wrongdoing. Reports of dissent, covering up, keeping silent, escalation and retaliation actions are presented. My thesis increases knowledge about the whistleblowing process and actions of software engineer experts in practice.

Contribution 4: I give researchers new insight into the complex challenges of collecting and analysing whistleblowing data. I make recommendations for future researchers regarding the adaptation of my WISE analysis framework to use alternate social science theories and industry domains actors. I propose further in-depth studies of emerging whistleblowing stories to include a wider set of actors (teams, organisations, regulators, and professional bodies) and tighter selection of domains such as health.

1.9 Ethics

My research required ethical approval as the data used is collected and analysed from humans. Lancaster University's Faculty of Science and Technology Research Ethics Committee granted approval (Appendix Ethics Approval).

1. Story interviews and workshops (FST17072 and extensions)
2. Expert interviews online (FST19079, FST19150)
3. Analysis of DotEveryone (Ipsos) dataset (FSTREC20115)

Due to ethical issues around the anonymity of participants, my research does not attempt to gather and present multiple accounts of a story. However, as part of background work on each case, I triangulate the stories through legal cases, news stories, government inquiries, and context specific research on public safety issues. There are ethical challenges in presenting my findings while protecting the identity of participants and their organisations. While some participants stated they do not mind their names being mentioned, it is unethical and irresponsible to do so. Participant names and careers are traceable online, the unintended consequences for individuals and related organisations cannot be predicted. The interviews with participants are in neutral settings (online). The research is not a form of intervention to change a situation or to seek closure on a story for participants, though participants reported it was cathartic to revisit stories from their past. Nor is the research able to predict the likelihood or frequency with which harmful situations and whistleblowing occurs in practice. The interviews are a very small sample set, of a seemingly rare phenomenon in software engineering, and any findings cannot be generalised.

1.10 Thesis Structure

This thesis is structured into eleven chapters, each of which is discussed below, with a summary of the contents of each chapter.

Chapter 2 introduces a background to whistleblowing. I present an overview of existing whistleblowing definitions and related theories, frameworks, and models. I construct a definition of whistleblowing specifically for software engineering practice anchored around the software development lifecycle. I describe codes of conduct, ethics, professional values, and standards that under pin this definition. This chapter feeds into both the literature review and the design of my WISE analysis framework.

Chapter 3 is a Literature Review and addresses Objective 1, to understand what is currently known about whistleblowing in software engineering research. With no existing literature reviews on the phenomenon of whistleblowing in software engineering identified, this chapter presents the methods, results, findings, and discussions of my review. I present primary themes and, research methods used. The findings and research gap analysis inform the design of my WISE framework and case reports.

Chapter 4 presents the WISE Analysis Framework: Describes how the WISE analysis framework was designed and developed. I explain how the framework was initially developed during the literature reviews in Chapter 1 and Chapter 3**,** and evolved through the analysis of captured stories. The framework guided the selection of stories that were developed into cases and presented in Chapter 6, Chapter 7 and Chapter 8.

Chapter 5 - Research Design and Implementation provides an overview and justification of my case study research methods. I present the data collection and analysis techniques and reflect on the empirical validity of these. I introduce the ORID (Objective, Reflective, Insight and Decisions) framework (Stanfield, 2000) adapted to facilitate the interviews used to collect software engineering stories. I discuss the application of the WISE analysis framework and the validity of this research and the practical and ethical considerations which shaped it.

Chapter 6, Chapter 7 and Chapter 8 present case study from each of the Nuclear, Health and Transport industry informed by participant interviews. **Chapter 9** presents a summary of case and story findings and answers RQ5. **Chapter 10** discusses findings from the literature review and case studies.

Chapter 11 presents the conclusions about how my thesis has advanced software engineering knowledge, and how it is positioned amongst previous research studies and the opportunities for future research.

Chapter 2. Background

This chapter gives an overview of the whistleblowing landscape, it covers general definitions, theories, frameworks and models and their relevance and application to my study of whistleblowing in software engineering practice. Definitions and theories of whistleblowing frames the scope of my WISE analysis framework, and the structure of data required to populate case studies. I discuss computing codes of conduct, ethics, professional values, and standards, from which my definition of whistleblowing in software engineering practice is anchored.

2.1 A Definition of Whistleblowing

The study of the whistleblowing phenomenon sits at the intersection of decision sciences, organisational studies, and business ethics. Moral psychology and business ethics scholars Alford (Alford, 2007) and Kenny (Kenny, Fotaki and Vandekerckhove, 2020) have focused on the lives of whistleblowers. With studies spanning over 30 years, Dozier, Near and Miceli have, between them, developed the most widely cited research on whistleblowing effectiveness and decision-making models (Dozier and Miceli, 1985; Near and Miceli, 1985, 1995; Near, Dworkin and Miceli, 1993; Rehg *et al.*, 2008; J. P. Near and M. P. Miceli, 2016). Many definitions of whistleblowing exist (Jubb, 1999) with varying levels of detail regarding roles, responsibilities, obligations, and evidence. A broad view, given by the United Nations, describes disclosures through internal mechanisms, external oversight mechanisms or in some cases to the media[24]. Near and Miceli define whistleblowing specifically with the inclusion of its purpose to "*effect action*" (Near and Miceli, 1985):

> **Whistleblowing:** The disclosure by organisation members (former or current) of illegal, immoral, or illegitimate practices under the control of their employers, to persons or organisations that may be able to effect action.

2.2 Existing Whistleblowing Theories and Research

With a goal of predicting and measuring the effectiveness of whistleblowing, Dozier and Miceli define a 3-step whistleblowing process for an individual having observed some form of wrongdoing to 1) assess if the wrongdoing needs to be reported, 2) if it is their personal responsibility to report, and if 3) there is an alternative reporting mechanism (Figure 2-1).

[24] https://www.undocs.org/A/70/361

Figure 2-1 A 3-step Whistleblowing Model (Dozier and Miceli, 1985)

Many controlled laboratory experiments, in many research domains, cite and base studies on this model. Hypotheses are developed with scenarios designed to isolate and control sets of situational variables regarding actors, their behaviours and human factors influencing the intention or likelihood to whistleblow. Study participants are profiled and their intention to report up or blow the whistle on a particular scenario are assessed via surveys or interviews. The findings are generalised and used to make predictions about individuals taking whistleblowing actions and the effectiveness to handle different types of wrongdoing in various organisational settings. Experimental studies that isolate variables (behaviours) of actors are some distance from the realistic context of an in-practice whistleblowing situation involving many actors and the evolving and potentially changing behaviours of those actors. However, such studies contribute to the knowledge of how whistleblowing decision making may happen.

An extension of Dozier and Miceli's 3-step model is Keenan and McLain's 7-step "*interactionist*" process loop model of whistleblowing (Keenan and McLain, 2017), shown in Figure 2-2. The model represents aspects of how individual actors make sense of a situation with respect to the seriousness of the wrongdoing, their motivation to correct the situation, their personal influence over the situation and the alternatives of seeking others who could correct the wrongdoing. The behaviour of searching for others leads to further re-assessment of the situation and the potential consequences for themselves and others made aware of the situation. Of note is the assessing the complicity of management with regards to the wrongdoing and the consequences for themselves and others because of this. The interactionist model describes four pathways (behaviours) actors might choose – suppression (silence, not reporting), procedural reporting, non-procedural reporting, or correction of wrongdoing through other interventions. Individual differences, situational factors, and organisational influences moderate how individuals and groups make one (or more) decisions regarding whistleblowing actions to correct the wrongdoing within their interpretation of culture and procedures within an organisation.

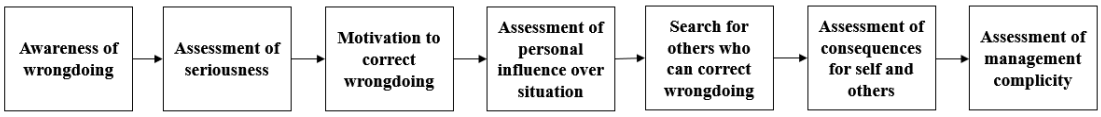


Figure 2-2 Interactionist Whistleblowing Model (Keenan and McLain, 2017)

The data from the Ipsos survey in Table 1-3 reported on six actions taken by participants on witnessing a potentially harmful situation (no action, left the company, considered leaving the company, raised concerns with a colleague, reported company to external body, raised concerns to manager or Human Resources). Ipsos did not include a "*correction through direct intervention*" option. The sequence of decisions and successive actions chosen by participants were also not captured in the Ipsos survey. This is not a criticism of the survey; Ipsos address their own research questions and as

part of this give some insight into harmful software and technology situations that my thesis is specifically concerned with. My research questions seek a different and deeper insight from whistleblowing situations than those reported up by Ipsos study participants – I look at interactions between actors – both stakeholders and software engineering artefacts.

On the use of Social Science Theories: Stol and Fitzgerald reflect that much software engineering research is *not* guided by explicit theories, nor does it produce explicit theory (Stol and Fitzgerald, 2013). Runeson and Host advise that for case studies, that theories may contribute to a framework for the analysis of data (Runeson and Höst, 2009). I am not constructing new theories but look to use relevant theories from reference disciplines and adapt them to the software engineering domain. Lorey, Ralph and Felderer (Lorey, Ralph and Felderer, 2022) warn that ignoring relevant social science theories in software engineering may: 1) undermine the community's ability to generate, elaborate and maintain a cumulative body of knowledge; and 2) lead to oversimplified models of software engineering phenomena. Lorey et al.'s literature review finds that less than two percent of papers use a social science theory, and the theories are rarely tested for applicability to software engineering contexts.

Keenan and McLain's interactionist whistleblowing model guides the design and development of my WISE analysis framework that in turn addresses Objective 2 and RQ4 (use of existing literature, social science theories and models to inform a research framework). The case study interviews with participants are designed to collect the necessary quantitative and qualitative data, using the ORID framework such that RQ5 (how do software engineers whistleblow?) can be answered. In summary, I seek to understand the facts about a situation, the events, and interactions (behaviours) between people, teams, and organisations (actors) related to the software engineering aspects of the situation (context). I seek a participant's assessment of the situation and decisions taken to mitigate harm to themselves or others. I focus on how awareness and knowledge of a situation develops, and to whom situations are escalated and how other actors (people and organisations) react and handle the situation.

In the 1970s, Tajfel and Turner (Tajfel *et al.*, 1979) developed **Social Identity Theory** to describe how individuals see themselves based on the social group(s) to which they belong. Social Identity Theory has been used to explore whistleblowing situation interactions where group identity is observed to influence individual actions and interactions with groups. This is relevant to whistleblowing (or not whistleblowing) as it can help to explore the standards, values, knowledge, and power that motivates individuals and groups (software engineers, teams, and organisations) responding to observed or perceived wrongdoing of groups they are a part of (e.g., a software engineering team or an organisation). Anvari et al. (Anvari *et al.*, 2019) take Social Identity Theory and combine it with four actions available to those seeking to mitigate a harmful situation. These four distinct choices are described as: 1) the power (influence) to change the group's wrongdoing behaviour from within, 2) the seeking of power from an external source to change the group behaviour, 3) powerlessness to change the group behaviour while remaining in the group (silence) or 4) to be able to exit the situation (leave the group). An individual or group of individuals may, for an evolving situation, attempt one or more of these actions to mitigate a harmful situation. Anvari et al. also presents a simple boundary escalation model to describe how whistleblowing escalates upwards from within a team through organisation, industry

and optionally to the public reporting of the wrongdoing. The boundary escalation model is detailed in Chapter 4 as part of the WISE analysis framework.

Anvari et al. report that existing whistleblowing research often focuses on the impact of individual and organisational factors, while overlooking how group memberships affect whistleblowing actions. Actor interactions underpin the WISE analysis framework that identifies interactions between actors (individuals, groups, and software engineering artefacts) up, down and within the layers of the boundary model. In Chapter 10 the findings from the case studies are discussed through the lens of Keenan and McLain's interactionist whistleblowing model and Anvari's whistleblowing actions based on social identity theory.

2.3 Defining Whistleblowing in Software Engineering

This section provides a definition of whistleblowing to focus the context and scope of my research in software engineering practice. Organisations and professional bodies develop codes of conduct or ethics to demonstrate a level of trust that can be attributed to or expected from an individual or organisation with regards to the design, development, operation and support of technology and software systems. The ACM's introduction to its Code of Ethics and Conduct states that "*the Code is designed to inspire and guide the ethical conduct of all computing professionals, including current and aspiring practitioners, instructors, students, influencers, and anyone who uses computing technology in an impactful way*". The ACM Code includes seven principles and nine professional responsibility statements, based on an understanding that the public good should be a primary consideration of computing professionals. My research is specifically focused on Principle 1.2 Avoid Harm:

ACM Principle 1.2: Avoid Harm: To minimise the possibility of indirectly or unintentionally harming others, computing professionals should follow generally accepted best practices unless there is a compelling ethical reason to do otherwise. A computing professional has an additional obligation to report any signs of system risks that might result in harm. If leaders do not act to curtail or mitigate such risks, **it may be necessary to "blow the whistle" to reduce potential harm**.

I construct a definition of whistleblowing in software engineering anchored on professional codes of conduct and software engineering best practice (e.g., (ACM Council, 2018):

Whistleblowing in Software Engineering: The process of making a responsible disclosure, internally or externally, of actions and artefacts perceived to be contrary to accepted professional values and standards in the software development lifecycle, carried out to mitigate harm to self, others, or wider society (Hunt and Ferrario, 2022).

The ACM Code warns that capricious or misguided reporting can be harmful and that computing professionals should carefully assess relevant aspects of the situation before reporting risks. My research reports on how individuals and teams assess situations and how they attempt to mitigate harm from these risks alongside harm mitigation to themselves (or others) as a result of raising concerns. The next section presents the current legal status and protection of whistleblowers and how it relates to software engineering practitioners.

2.4 Legal Status and Protection of Whistleblowers

The early 1970s saw whistleblowing at the centre of heated debates in the technology and engineering community, triggered by Nader's call to scientists and engineers to "*hold responsibility*" (Boffey, 1971). In the 1980s the Challenger disaster re-ignited the discussion in the community of those speaking of potential risks having their concerns ignored or suppressed (Jarman and Kouzmin, 1990). Zelby's 1989 feature article written for the IEEE Technology and Society Magazine (Zelby, 1989) concedes that "*the issue of blowing the whistle is complicated*" and calls for something to be done beyond codes of ethics, standards, and legislation; the article concludes by arguing for the establishment of an ombudsman, seen as a possible mechanism to "*protect both the whistleblower and those on whom the whistle was blown*". The ISO-37002 whistleblowing management system guidelines (IOS (International Organization for Standardization), 2021), first proposed in 2018, was published in 2021 with the following goals:

- encouraging and facilitating the reporting of wrongdoing
- supporting and protecting whistleblowers and other interested parties involved
- ensuring reports of wrongdoing are dealt with in a proper and timely manner
- improving organisational culture and governance
- reducing the risks of wrongdoing

Prof. Wim Vandekerckhove, Convener of the ISO working group that developed the standard, said "ISO 37002 will help to build trust between an organisation and its stakeholders, providing a strong layer of protection against corruption."[25] The ISO standards organisation define a whistleblower as "*a person who reports suspected or actual wrongdoing and has reasonable belief that the information is true at the time of reporting. Reasonable belief is a belief held by an individual based on observation, experience or information known to that individual, which would also be held by a person in the same circumstances*."

Whistleblower protection varies greatly across countries, employment, and contract types. A government employee may be bound by state secret acts, a freelancer by non-disclosure agreements, and in some sectors such as healthcare there are disclosure channels provided[26]. In the UK, Vandekerckhove's 2012 survey of 2000 British adults found 81% of people supportive of whistleblowers (Vandekerckhove, 2012). Despite this, and with laws supporting whistleblowers, media reports and studies overwhelmingly report that those reporting harm or wrongdoing are victimised and ostracised in the workplace by both colleagues and management (Alford, 2007), even when their disclosures are vindicated after lengthy investigations or court cases (Fitzgerald, 1990). In Europe, prior to 2019, few European Union (EU) Member States fully protected whistleblowers by law. To address this fragmented legislative landscape, new EU whistleblower laws were adopted in 2019, requiring protection by both companies and governments, including safe channels for reporting internally and externally[27]. In the UK, protection for whistleblowers is included in the Public Interest

[25] https://www.iso.org/news/ref2703.html
[26] https://www.cqc.org.uk/news/stories/quick-guide-raising-concern-about-your-workplace
[27] https://www.consilium.europa.eu/en/press/press-releases/2019/10/07/better-protection-of-whistle-blowers-new-eu-wide-rules-to-kick-in-in-2021/

Disclosure Act 1998[28]. If a worker makes a protected disclosure, compensation can be claimed for any victimisation following such disclosures.

Research and advocacy practitioners (Vandekerckhove, James and West, 2013) stress the importance for organisations to understand their obligations upon the discovery and disclosure of harmful practices; they have a responsibility to explain and uphold their policies and processes for reporting and managing the issues. Also highlighted is the need for individuals to be aware of the support and protection available to them should they choose to make disclosures. Kenny et al.'s recent organisational research (Kenny, Fotaki and Vandekerckhove, 2020) finds that, even after reprisals, whistleblowers continue to be passionate about their professional integrity and stress the importance of sustained practical and material support for those who blow the whistle.

2.5 Summary

Whistleblowing is a phenomenon that is not specific to software engineering, therefore using general whistleblowing and social identity theories are valid mechanisms to study whistleblowing situations in software engineering.

The software engineering industry is a diverse community including consultants, contractors, third party, and outsourced organisations that may fall outside whistleblower protection policies or laws in a particular organisation or country. When serious harmful software issues are discovered by software engineering teams there should be an understanding of channels and actions available to report up and mitigate harm, which in extreme cases could include whistleblowing. In my case studies participants are asked about their awareness or involvement with professional bodies such as the British Computer Society (BCS), the Institute of Electrical and Electronics Engineers (IEEE), the Institution of Engineering and Technology (IET) and the Association for Computing Machinery (ACM) and the support practitioners receive in whistleblowing situations.

[28] https://www.gov.uk/government/publications/the-public-interest-disclosure-act/the-public-interest-disclosure-act

Chapter 3. Literature Review

This chapter is based on my ICSE 2022 paper (Hunt and Ferrario, 2022): A Review of How Whistleblowing is Studied in Software Engineering, and the Implications for Research and Practice. The review was conducted in 2020, and included items published up to 2019. Whistleblowing is often portrayed as a tragic, heroic, or responsible act of individuals, but with limited value for understanding the specific software engineering aspects behind a story. The first objective of this thesis was to understand how whistleblowing is studied in software engineering and identify the gaps. With no previous literature reviews identified, this chapter presents my review of the literature, based on the following research questions:

RQ1: What are the primary whistleblowing themes in software engineering literature?
RQ2: How is whistleblowing researched in software engineering?
RQ3: What are the research gaps?

The review looked for literature items specifically about whistleblowing in software engineering. My primary search was based on the title, abstract and key words of items as found in other software engineering mapping studies (Glass, Vessey and Ramesh, 2002; Shaw, 2003). I answer the three research questions and conclude with a discussion of the validity issues of this review. The review finds that field-based studies of whistleblowing in software engineering are rare. Findings from this review identify this thesis as a new contribution to software engineering research, as no such literature reviews or whistleblowing case studies have previously been published.

3.1 Method

This literature review is *guided* by Kitchenham's systematic literature review procedures, an approach designed for established research fields (Kitchenham, 2004). Whistleblowing precisely defines the area of interest. Therefore, my search is primarily focused on whistleblowing as found in the title, abstract or key words of items. This is a valid approach and is used in similar mapping studies (Glass, Vessey and Ramesh, 2002; Shaw, 2003). The review reports on representative literature to develop perspectives on whistleblowing. While whistleblowing is not a new phenomenon, based on manual preliminary searches, it is found not to be a widely studied or a well-established area in software engineering. I prioritised coverage and included any peer reviewed articles or studies about whistleblowing in software engineering. Not all items returned are empirical research items and thus cannot be easily compared (Cruzes *et al.*, 2015). I do not use publication dates or citation counts as cut off points for inclusion, this is a review of all available peer reviewed literature up to and including 2019.

The item selection was organised in three broad cycles: 1) find any whistleblowing items in the chosen information sources; 2) remove false positives where whistleblowing is used in a literal meaning in sport, football, sailing, acoustics, or video editing for example, 3) apply software engineering specific selection criteria to the remaining items for their relevance to software engineering practitioners, projects, or practices. The specific steps used to prepare and execute the primary search, and carry out items' selection are described below:

1. Identify the need for a literature review.
2. Formulate research questions.
3. Define search strategy and string (whistleblowing).
4. Search relevant information sources.
5. Remove duplicates (SCOPUS contains ACM and IEEE items).
6. Remove false positives.
7. Apply specific inclusion criteria (software engineering).
8. Apply general inclusion criteria (date, publication year, type).

Selected items were then inductively coded (grounded in data, not existing concepts) by two researchers, to support the analysis and interpretation of the results:

1. Extract meta data from selected items (title, abstract, author, publication, citations, source, country(s), keywords, document type).
2. Researchers independently review and inductively code themes for subset of items.
3. Researchers compare and discuss themes.
4. Researchers agree and define set of primary themes.
5. Researchers independently allocate primary theme to all items.
6. Use Cohen's Kappa (K) to calculate primary theme inter-rater reliability score.
7. Researchers discuss and resolve discrepancies.
8. Researchers run second round of primary theme allocation based on defined themes.
9. Finalise results for analysis and interpretation.

3.2 Search and Selection Strategy

Information Sources: Automated searches were performed on IEEE Xplore Digital Library, ACM Digital Library, and the SCOPUS (computer science) database. The IEEE is the largest professional organisation for technology and the ACM covers computing and information technology; any items in here could be related to software or software professionals. SCOPUS is a broader database covering many disciplines, so the search was limited to computer science (COMP-SCI) items only.

Whistleblowing Search: The word whistleblow precisely defines the area of interest; however, it can be phrased in multiple ways. The following 12 whistleblowing variations were used to search the information sources. Taken from existing generic whistleblowing theories, additional keywords such as "dissent", "silence" and "secrets" were also used, but no relevant additional items were found with their inclusion:

"whistle blow" OR "whistleblow" OR "whistle-blow" OR "whistle blowing" OR "whistleblowing" OR "whistle-blowing" OR "whistle blower" OR "whistleblower" OR "whistle-blower" OR "blowing the whistle" OR "blown the whistle" OR "whistle blown"

Removing False Positives: The titles and abstracts from the whistleblowing search were manually reviewed to remove false positives, for instance where whistleblowing is used in its literal sense (e.g., sport, football, sailing, acoustics, or video editing).

Removing Duplicates: Duplicate entries, where SCOPUS returned matching items from the IEEE or ACM, were identified. SCOPUS and IEEE functionality provided the richest set of export data; therefore, ACM duplicates were the most frequently removed.

Selecting Software Engineering Items: The abstract and body of whistleblowing items were manually reviewed to ensure terms or narratives relevant to software engineering were present:

- Creating software and IT projects (products or services)
- Studies or reports on software engineering practitioners

Inclusion Criteria: Finally, the following inclusion and exclusion criteria were applied to the items:

Accepted

- Any publication year
- Peer reviewed book chapters and magazine items
- Peer reviewed conference and journal papers (any length)

Rejected

- Papers not in English
- Entire books, book reviews, or volumes of proceedings

Volumes of proceedings were rejected as individual relevant items were returned as part of the main search. As an example, Plotkin (Plotkin, 1989) (Economic Survival and Whistleblowing) was returned as an item from a volume of proceedings, although subsequently rejected as not specifically a software engineering item.

3.3 Results

The search for whistleblowing (or its variation) was run against each source database; 311 preliminary items were found across the three source databases, as detailed in Table 3-1.

First-round selection: Metadata from the 311 preliminary items was downloaded into a spreadsheet. An initial search identified 124 false positive or duplicate items, as shown in Table 3-1, giving 187 first round selection items. The IEEE and SCOPUS functionality provided the richest set of export data; therefore, ACM duplicates were the most frequently removed.

Table 3-1 Preliminary (311 items) and First-round Selection (187 items)

Whistleblowing Search	IEEE	ACM	SCOPUS	Total
Preliminary: Title, abstract and keywords	70	94	147	311
False positives or duplicates	13	76	35	124
First round selection	**57**	**18**	**112**	**187**

Second-round software engineering relevancy: The title and abstracts of the 187 items from the first-round selection were then analysed for the word "software" or "engineer", thirty-four unique items were found. A search for "project" (things that software engineers work on) in the 187 items also run and returned 20 items and gave us a further 11 unique items to add to the existing 34 items. Finally, a manual review of the 187 items abstracts and bodies was made to identify software engineering project and product related items not detected by the previous systematic key term search. Examples of words found included "IT", "computing industry", "professional", "safety-critical system" and "technology". This gave a further 29 unique items for inclusion. All items were, where available, downloaded and checked for relevancy based on the inclusion criteria in Section 3.2, 14 were excluded. This gave a final list of 60 items for analysis, as shown in Table 3-2.

Table 3-2 Second-round Software Engineering Relevancy (60 items)

Second Round Selection	Count
Items with “software” or “engineer”	+34
Unique items with “project”	+11
Manual check for relevance to SE	+29
Exclusion criteria applied	-14
Final Accepted Items	**60**

3.4 A Thematic Analysis of Review Items

A thematic analysis followed to inductively code the 60 accepted items:

1. 20 items independently reviewed and coded for themes by two researchers.
2. Themes and sub-themes compared and discussed between researchers.
3. Key theme list agreed and defined (6 primary).
4. All items independently allocated a primary theme.
5. All items independently allocated secondary theme (if identified).
6. Primary inter-rater reliability score calculated (K = 0.82).
7. Researchers discussed and resolved 8 discrepancies.
8. Second round of primary theme allocation run.
9. Disagreed (weakly) on only 2 items.
10. Result set finalised for interpretation and discussion.

For a subset of twenty items, myself and my then supervisor Marie Angela Ferrario independently tagged a primary theme and identified any secondary themes, alongside any sub-themes (unlimited number of themes to each item). The resulting list of themes included: whistleblowing practices, individual attitude and behaviour, professional ethics, technology for whistleblowing, not whistleblowing (not speaking up), management, governance, organisations, case study, stories, human decision making, industry domains (education, government, engineering, health), software engineering practice aspects, security, and privacy.

My then supervisor and I compared and discussed their results for each of the twenty items to understand each other’s applied theme. Of note was the separating out of "human factors" into "personal and social factors" and "organisational and professional issues" - identifying differences between individual and organisational themes. From this we jointly came up with six primary coding themes, with a list of sub-themes in each primary theme as shown in Table 3-3.

Table 3-3 Primary Theme of the 60 Items

Primary Theme	Aspects include	Item Count	Item reference
Human Factors (Personal and Social Factors)	Personal values, motivation, social identity, decision making, mum effect (keeping quiet about issues), social responsibility	20	(Larson, 1971; Perry, 1981; Florman, 1982; Dyro, 1988; Smith, Keil and Depledge, 2001; Smith and Keil, 2003; Keil *et al.*, 2004; Kumagai, 2004; Swierstra and Jelsma, 2006; Keil, Im and Mähring, 2007; Niu, Stylianou and Winter, 2008; Park, Im and Keil, 2008; Brinkman, 2009; Schilhavy and King, 2010; Martin, 2011; Wang and Oh, 2011; Adamson, 2015; Wang, Keil and Wang, 2015; Jaeger and Eckhardt, 2018; Petter, 2018)
Whistleblowing Stories	History of whistleblowing, educational case studies, high profile news stories	15	(Rost and Glass, 2011), (Fitzgerald, 1990; Bowyer, 1997, 2000; K. W. Bowyer, 2001; McCubbrey and Fukami, 2009; Kline, 2010; Adams, 2014; Hintz and Dencik, 2016; Pfleeger, 2016; Gunasekara, Adams and Murata, 2017; Kavathatzopoulos *et al.*, 2017; da Silva and Dobránszki, 2019; Ghoshroy, 2019)
Technology for Protection or Detection of Whistleblower	Technology for protection or detection. Platforms, protocols, vulnerabilities, privacy enhancing	12	(Taniguchi *et al.*, 2005; Okolica, Peterson and Mills, 2006; Bell, 2011; Roth *et al.*, 2013; Bodó, 2014; Hohenberger *et al.*, 2015; Sloan and Hernandez-Castro, 2015; Zakia *et al.*, 2017; Nursalman, Anggraeni and Firdaus, 2018; Sion *et al.*, 2018; Jayakrishnan and Murali, 2019; Maitre *et al.*, 2019)

Primary Theme	Aspects include	Item Count	Item reference
Whistleblowing process	Process, theories, models, motivation, agency theory, decision making, behavioural reasoning theory	6	(Zelby, 1989; Kevin W Bowyer, 2001; Park, Keil and Kim, 2008; Keil and Park, 2010; Oh and Teo, 2010; Tavani and Grodzinsky, 2014)
Organisational and Professional Aspects	Roles, responsibilities, culture, professional code of ethics, deaf effect (not listening to issues), governance, policies, professional bodies, regulation	6	(Park, 1996; Preston, 1998; Keil and Robey, 2001; Park and Keil, 2007; Wang *et al.*, 2017; Noor and Mansor, 2019)
Software Engineering Practice	Misuse of open-source code	1	(Sarkinen, 2007)

All sixty items were independently allocated a primary and, if applicable, secondary themes by the two researchers. After the calibration cycle and two primary theme iterations we agreed strongly (Kappa = 0.96, (Cohen, 1960)) on a primary theme for each item. We discussed and resolved the disagreement between whistleblowing process or organisational aspects. Organising and professional aspects are factors that affect the whistleblowing process and are discussed in stories, so there is overlap in how we each selected the primary theme.

While software engineering practice aspects was a primary theme on only one paper, there were a significant number of items (nearly a quarter) where software engineering aspects were identified as a secondary theme, see Table 3-4. The software engineering practices on these items included software development lifecycle, coding, testing, software piracy, code reuse, misuse of IT, and security concerns. Papers with contrived study scenarios were developed through consultation with software engineering experts into the types of issues software engineers *might* find in practice. I was hoping to find reports and software engineering aspects from practice.

Table 3-4 Analysis of Secondary Themes

Theme	% of items Primary Theme	% of items Secondary Theme
Human Factors	33%	30%
Whistleblowing Stories	24%	5%
Technology for Protection or Detection of Whistleblower	20%	2%
Whistleblowing process	11%	16%
Organisational and Professional Aspects	10%	25%
Software Engineering Practice	**2%**	**23%**

3.5 Findings – Meta Data

Based on item metadata, a summary of citations, temporal and geographical distribution of reviewed items, alongside an overview of the types and publication sources of items was conducted. Figure 3-1 presents a publication timeline of the search results. It shows that generally there has been an increase in publications mentioning whistleblowing, peaking at eighteen items in both 2017 and 2018. A word heat map found "organisation" was used in over half (53%) of abstracts in this two-year period, double the average occurrence for "organisation" over all time (28%).

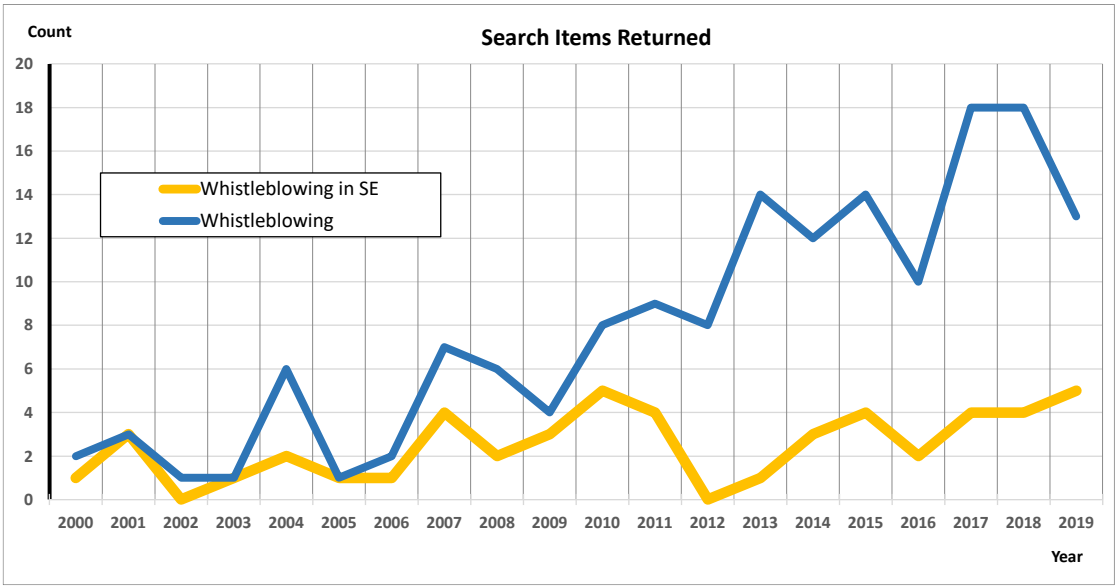


Figure 3-1 Temporal View of Search Results

However, the number of items specifically about *whistleblowing in software engineering* (from the software engineering relevancy selection process) shows low numbers (peak of 5 in 2010 and 2019) with only small changes, both up and down, over time. Overall, the item types break down into three distinct groups - conference and journal papers (49), book chapters (3) and magazine articles (8). The first six items chronologically are whistleblowing and ethics focused magazine articles. Education related papers, with whistleblowing case studies as part of ethics education, start to emerge in the late 1990s. Journal and conference papers begin appearing in the early 2000s with empirical research studies on the intention to whistleblow and on technology solutions (supporting or detecting whistleblowing).

An analysis of the publications for the forty-nine conference and journal papers reveals the majority of papers found were from Information Systems (14 items), Education (7 items), Computing in Society (4 items) or Ethics (3 items) publications. Peer reviewed magazine articles were found in a broad range of publications including IEEE Technology and Society, IEEE Spectrum, IEEE Computer and ACM Inroads.

Authors and Citations. There are 48 primary authors with a total of 96 authors across all items. At the time of searching, there were a total of 655 citations across the 57 non-book chapter items. The collective body of research works from Keil, Park, Wang, and Smith relating to bad news reporting on IT projects are consistently the highest cited. At the time of searching, the citation counts of this groups thirteen papers account for

over three quarters of all citations. Over 50% of the items have less than two citations, with thirteen items having not been previously cited. It seems the field is dominated by a few key authors, potentially leading to a lack of diversity of voices and approaches to this topic.

Participant and Geographical Distribution. Over half the items are from work carried out in the USA, with at least 19 other countries represented. Empirical study participants are evenly split between students and professionals, and there is a similarly even split between Eastern and Western cultures, including two specific cross-cultural studies with the USA and South Korea. India, South America, and Africa are not well represented as authors or participants. Whistleblowing stories referred to in items are predominantly about USA-based organisations where whistleblowing was involved and resulted in a media presence and interest in the story. Given the international spread of the software engineering development community alongside the complexity and global reach of IT systems, future work could look at how whistleblowing happens internationally.

3.6 Findings

With only sixty items identified in nearly fifty years of publications, the main finding was the limited amount of research specifically on whistleblowing in software engineering. This section presents the themes identified in the literature (RQ1). I report on the approaches used to study whistleblowing in software engineering and identify some of their strengths and weaknesses (RQ2). Finally, I examine the gaps in whistleblowing research (RQ3) and how the literature review findings feed into RQ4 and the designing of my WISE analysis framework. In Chapter 9 the literature review and case study findings are presented together and the implications for future research and practice in software engineering is discussed.

3.6.1 RQ1- What are the Primary Whistleblowing Themes?

Human Factors (Personal and Social) were identified as the most frequent primary theme (N=20), which includes aspects of individual values, personal motivations, and social identity (i.e., belonging to a team) as Table 3-3 shows. The fact this is the most recurrent theme is not surprising since whistleblowing is widely portrayed, both by research and media, as being about individuals making decisions according to their conscience, sense of responsibility, and obligation. *Whistleblowing Stories* also feature frequently (N=15), particularly in the context of higher education where whistleblower stories are used in computer and engineering ethics courses to illustrate to students the situations in which their responsibilities may be challenged at work. The *Technology* theme follows (N=12) including ten items primarily concerned with the development of safe and secure channels for safeguarding whistleblowers' communication, alongside two looking at detection of whistleblowing activities. The Whistleblowing Process (N=7) includes items looking at decision-making theories, motivations, and process models for whistleblowing. Organisational and Professional Issues theme (N=7) includes papers investigating organisational culture and structures affecting whistleblowing processes. Finally, only one paper has *software engineering practice aspects* as the primary theme (Sarkinen, 2007), discussing "*dirty code*" where developers take open-source code into closed source systems and infringe on software licensing and intellectual property rights. While not frequent as the primary theme, software engineering aspects were identified as a significant secondary theme, with

fourteen items using software engineering practice aspects as part of their back story in the experimental studies (software development lifecycle, coding, testing, design, open source, software piracy, code reuse, misuse of technology, security issues). However, as discussed previously, these are not studies of in-practice experiences - there is a knowledge gap in the research looking at actual software engineering actor experiences.

3.6.2 RQ2- How is Whistleblowing Researched?

In this section the approaches used to report on or study whistleblowing in software engineering is discussed. It is done in two steps: first, I use Stol and Fitzgerald's ABC framework (Stol and Fitzgerald, 2018) to look at the research goals and context of the items. I then apply Easterbrook et al.'s methods categories (Easterbrook *et al.*, 2008) to guide the discussion around the subset of empirical research papers found. As previously discussed in Section 1.1, Stol and Fitzgerald's ABC research classification takes into consideration the generalisability of actors (A) being studied, the measurements of their behaviour (B), and how realistic a context (C) the study is set in. This framework is particularly relevant to my research because whistleblowing ultimately focuses on actors (e.g., individuals, teams, organisations), whose behaviours (e.g., interactions, speaking-up or keeping quiet) are situated in specific industry domains or work contexts. Laboratory experiments are able to recruit large numbers of actors to study whistleblowing intention and are more easily able to generalise their findings. Field studies of the seemingly rare phenomenon of whistleblowing, are challenging to find sufficient participants to draw conclusions from.

Laboratory experiments are found to be frequently used in the context of project failure or individual reluctance to report up bad news on IT project scenarios. There were few examples harm or misuse from successfully delivered IT systems, such as found in the Volkswagen, Cambridge Analytica, and the UK Post Office Horizon stories. Aspects of software engineering such as software bugs (Park, Keil and Kim, 2008), misunderstood requirements (Smith, Keil and Depledge, 2001), system limitations (Smith, Keil and Depledge, 2001), poor testing (Keil and Park, 2010), fault responsibility of outsourced work (Keil and Park, 2010), ability to hide bad news (Keil *et al.*, 2004) , 3rd party code and fault responsibility (Keil, Im and Mähring, 2007) are used to form a back story to the experiment scenarios. Situational variables are carefully controlled in role play scenarios, with the likelihood (intention) to whistleblow then captured. Extreme scenarios are used to maximise variance. Situational factors such as professionalism (Schilhavy and King, 2010), organisation climate and culture (Keil *et al.*, 2004; Keil, Im and Mähring, 2007) , time urgency (Park, Im and Keil, 2008), harm of consequences (Smith and Keil, 2003; Keil, Im and Mähring, 2007; Park and Keil, 2007; Park, Keil and Kim, 2008) and proximity to victims (Park, Keil and Kim, 2008) are shown to increase the likelihood of reporting bad news. However, the studies focus on an individual's response to a scenario, with only limited reflection on wider individual and group dynamics (inside and outside an organisations) that might occur and have an impact on knowledge of choice of actions.

Natural setting Most natural items were non-empirical position, opinion and technology solutions papers from in-field reports or observations not been based on empirical research studies. **Neutral setting** items included interviews with public whistleblowers and asks "*where are they now, would they do it again*" (Fitzgerald, 1990) with a summary of their situation and discussion to follow. The findings are not detailed, comparable or generalisable because of the low numbers of actors, the specific

context of each situation, and the varying degree of background detail to each story presented. An empirical paper (Keil and Park, 2010), proposes a framework, that underpins several of the previously described experimental lab-based studies looking at factors affecting the assessment of a situation and the willingness of participants to report up on it.

The most cited paper in the review is Keil and Robey's 2001 field study (Keil and Robey, 2001). Interviews are conducted with IT auditors about their experiences of project failure situations. Organisational conditions such as size, structure, culture, audit function power and relationship to senior management, job security are identified as factors affecting why internal auditors may or may not assert their responsibility to report bad news. More studies like this, taken to the software engineering practitioner level could offer much insight for the software engineering community. Smith and Keil's (Smith and Keil, 2003) theory development paper, bringing together factors on the reluctance to report bad news in the software engineering project domain. The papers lead to the group of controlled experiment type studies spanning more than 10 years that contribute to the understanding of factors (human and organisational) affecting responsibility and willingness to report bad news on IT projects.

Context of Situations: Across all review items, there are a range of domains discussed, including safety critical systems (Bowyer, 2000; K. W. Bowyer, 2001), nuclear power (Fitzgerald, 1990; Kline, 2010), health (Dyro, 1988), transport (McCubbrey and Fukami, 2009), academia (da Silva and Dobránszki, 2019) and defence (K. W. Bowyer, 2001; Ghoshroy, 2019) situations. There were four stories where issues were reported to the media (Edward Snowden (Tavani and Grodzinsky, 2014), GEC Nuclear (Fitzgerald, 1990), USA missile defence program (K. W. Bowyer, 2001; Ghoshroy, 2019)), some had tried disclosing issues internally before eventually going to the media or government agencies. Table 3-5 presents a summary of stories mentioned in the literature review items. In the individual stories referred to, all bar two whistleblowers lost their jobs (resigned or fired). Notably three whistleblowers received awards from the IEEE for outstanding service to public interest (Clinch River (Kevin W Bowyer, 2001), Air Shield Incubator (Kumagai, 2004), GEC Nuclear (Fitzgerald, 1990)). The ACM debated awarding, but did not give, Snowden a "Making a Difference Award" (Adams, 2014). The Volkswagen story [29] is included in Table 3-5, it stands out as there were initially no whistleblowers and the story only came to light due to academic researchers publishing their findings; a seemingly rare example of external whistleblowing on an internal software engineering situation.

[29] https://theconversation.com/where-were-the-whistleblowers-in-the-volkswagen-emissions-scandal-48249

Table 3-5 Whistleblowing stories referred to by literature review items.

Story Title	Year	Actor	Actor Role	Potential Harm or Impact	Behaviour - Disclosure Actions
Google Maven	2018	4000+ google staff	Varied.	AI combat drones	Media
Volkswagen	2015	External researcher / regulator	Researcher	Environmental damage Not following regulations	Regulator and Media. Researchers published test reports
NSA / GCHQ	2013	Edward Snowden	Security Analyst	State surveillance	Media. Resigned. Escaped to Russia.
Enron	2001	Sherron Watkins	Vice President	Bankruptcy due to fraud	Internal. Government. Tried to fire.
AirShield Incubator	1995	Salvador Castro	Electrical engineer	Flaw in incubator, product recalled	Internal. Threatened external. Fired.
Challenger Shuttle	1986	Roger Boisjoly	Engineer	Explosion. 7 astronauts died	Internal. Fired.
Hughes Aircraft	1986	Margaret Goodearl and Ruth Aldred	Quality assurance	Failures in testing (falsifying) chips for aircraft, tanks, and missiles	Internal and Government. Left and laid off
Star Wars	1985	David Parnas	Researcher on Advisory Panel	Failure to protect USA. Trustworthy software not possible	Resigned. Campaigned against Star Wars.
Clinch River Breeder	1983	Demetrios Basdekas	Electrical engineer	Problems with reactor	Internal. Project cancelled. Kept his job/career.
FAA	1981	Jim Pope	Aviation engineer	Air safety. Mid-air collision avoidance.	Internal. Public and Congress. Fired.
Ford Pinto fuel tank	1978	Frank Camps	Design Engineer	Safety testing concerns. Cost benefit of human lives if tank explosions.	Internal. Demoted, resigned.
CIRCLE IT project	1977	Virginia Edgerton	Senior Information Scientist	Speed (not in time) of dispatching police cars to emergencies	Supervisor. CIRCLE committee. Fired
GEC Nuclear	1976	Ed Gischel, Rick Parks, Larry King	Engineers	Inadequate testing, unsafe designs. Not understanding consequences.	Internal. Media. Resigned
San Francisco BART	1969/71	3 engineers	Engineers. (Automated trains)	Danger to life. Automated train controls.	Internal. Media. Fired.

3.6.3 RQ3- What are the Research Gaps?

The gap analysis is based on the study types found in RQ2. The combination of case studies, field studies and laboratory-controlled experiments has contributed to a better understanding of factors affecting the likelihood of whistleblowing in software engineering practice. However, the studies found lacked diversity in research perspectives, and the number of authors involved was small.

Easterbrook et al. (Easterbrook *et al.*, 2008) present a list of five research methods most relevant to empirical software engineering (Controlled experiments, Case studies, Survey research, Ethnographies, Action research). Most empirical items found in the literature review were controlled lab-based experiments or neutral survey studies looking at factors that influence a participant's intention to speak up and profiling of characteristics of participants. These contribute to our understanding of human and technical factors that *may* affect whistleblowing in practice. Case studies are present in the review, with varying levels of detail; most report on an incident based on secondary data and just three use primary data sources alongside one autobiographical case study. None follow a detailed case study research method such as that defined by Yin (Yin, 2018).

Ethnographic Field Studies: Whistleblowing in software engineering is a complex social-technical phenomenon and its understanding would benefit from in-depth ethnographic studies to understand how situations of public concern arise and how they are then handled. No ethnographic studies were found. Whistleblowing is a knowledge-based activity, and as such would be difficult to report on from observations alone. Whistleblowing is also of a sensitive nature, making the design and recruitment of participants to any in-practice research studies are challenging, particularly when much of the process leading to whistleblowing may happen under the radar. In addition, situations may take course over a long time, thus requiring extensive research time and resources that may not be easily available. The seriousness of the consequences of any actions leading to whistleblowing may mean participants and their organisations would adjust behaviours during any observation or action period to avoid or reduce the risk of detection or retaliation.

Action research studies were not found in the literature review. Rost and Glass (Rost and Glass, 2011) discuss interventions such as trialling whistleblowing hotlines, perceived as being a possible solution to a problem (options for reporting up concerns). Ethical issues and the potential unintended consequences of encouraging participants to report up issues would need to be thought through and addressed when tackling such research. Additionally, all the challenges described in ethnographic studies could be found for action research studies where participants and their organisations adjust behaviours during the study period.

Frequency of whistleblowing: Whistleblowing seems to be a relatively rare occurrence in the science and technology community (Vandekerckhove, James and West, 2013). No studies reported or estimated the frequency of whistleblowing in software engineering practice. DotEveryone (Ipsos) survey data indicated that twenty-eight percent of tech professionals surveyed had witnessed harmful situations and that ninety percent take some action as a result. This high number could be due to respondents answering questions in a manner viewed favourably by others. To counter this social

desirability bias the Ipsos survey was an anonymous online self-completion survey. There is insufficient participant employment data granularity to estimate a frequency of occurrence of situations arising and actions taken. There is no guarantee a whistleblowing situation would arise during an ethnographic or action research study.

Software Engineering Practice Aspects: The Ipsos survey (Miller and Coldicott, 2019) finds ninety percent of tech workers who see harmful situations do take some action, but what exactly were those actions, how effective was it and what was the final outcome? Very limited published work examining these aspects from a software engineering practice perspective were found. Within the software development lifecycle, research is needed to improve our understanding of the effectiveness of software professional actions to change harmful software engineering situations. These studies need to include actual harms seen attributed to software, software engineering practices or software engineering decisions made, that trigger actions up to and including whistleblowing.

Missing interactions: Whistleblowing stories often describe whistleblower actions, with little focus on the involvement and responses from other software engineering actors, disclosure recipients (internal or external), organisations, and groups in wider society made aware of these software engineering situations. Reports of high-profile software scandals in the media appear to be increasing and so too the scrutiny of software organisations to show consideration of the social and human impact of the systems they design, build, and operate. In 2021, there have been a succession of whistleblowing stories in the media, with former software engineering practitioners from companies such as Google and Facebook speaking out, with evidence, about the harms caused by features of their products (e.g., algorithmic bias and putting profit before public good). Researching new and emerging software engineering stories, presented as comparable cases studies, would help map out and understand the growing involvement of the wider software engineering community and campaign groups, and how that impacts the effectiveness of whistleblowers attempting to mitigate harm. Stories also provide relevant and modern case studies for software engineering education and practice to help address the perception that reporting wrongdoing is more than just a matter of individual morals and responsibility.

3.7 Summary

The findings from my literature review indicate that while whistleblowing is increasingly mentioned in software engineering literature, it is an under-explored area of software engineering research. The software engineering research community is uniquely positioned to explore and advance the understanding of situations in software engineering, with whistleblowing being an extreme form of harmful situation reporting.

Carefully designed approaches are required, sympathetic to the complexities and risks associated with whistleblowing studies, to evidence and reflect on the code, software, systems, or software engineering practices that lead to potential and actual whistleblowing situations. The gaps in methods and knowledge in whistleblowing research found in the literature led me to develop Objective 2 (an analysis framework) and Objective 3 (a study of software engineering expert experiences of whistleblowing situations).

3.8 Limitations and Mitigation

This section discusses the limitations of the literature review and mitigations to address them. The review sought to select and categorise existing literature about whistleblowing in software engineering. I cannot guarantee that some relevant literature was excluded or missed, however the review does not rely on single items for its findings.

3.8.1 Construct Validity – Search Terms

The first threat is of construct validity - do the search strings reflect my search intentions? Whistleblowing is a very specific search term - precisely defining the actions and escalations of a harmful situation in software engineering practice that I am studying. I used twelve variations of whistleblowing to find matching items. Other key words such as "dissent", "bad news reporting", and "misreporting" were applied alongside “whistleblowing” and returned no additional items. The search was specific and sufficient for what I wanted to achieve with a literature review and to position my future research. A future literature review, with adjusted research questions and source databases (e.g. outside the software engineering domain), could look for items reporting on types of harm and wrongdoing attributed to software engineering projects, products, and services in other domains. An example of how this might look is shown in Case A (Chapter 6), where research is cited from medicine looking at critical incidents with ventilators, reviews of medical device issues and who reports issues to USA regulators.

3.8.2 External Validity

The second threat to validity was whether the selected publications provide enough studies to perform the thematic analysis and review of methods found in software engineering research. This thesis began by reviewing a considerable amount of news media and grey literature where whistleblowing is often portrayed as a tragic, heroic, or responsible act of individuals, but with limited value for understanding the specific software engineering aspects behind a story. Manual snowballing checks were conducted, relevant papers were found to already be in the review. It is possible there are papers not retrieved by the search engines or manual snowballing. In mitigation, the three databases used are common sources for literature reviews in software engineering research and whistleblowing is the very specific term we are looking. Additionally manual searches for whistleblowing studies outside of the software engineering domain but relating to software engineering practitioners were conducted. Jalali et al. (Jalali and Wohlin, 2012) report that systematic studies of literature can be done in different ways and that in software engineering, the main recommended first step is using search strings in several databases – as I have done. Their paper compares databases search and snowballing as first step approaches for conducting literature review studies and concluded that none of the first steps significantly outperform the other.

I used Google Scholar as an ongoing check for “whistleblowing in software engineering” items. Google Scholar does not give the option to search on “title and abstract” but does offer a whole article search. The whole article search returned tens of thousands of results, so I ranked by software engineering relevance and took the top 100. Those identified as in scope were found in the review.

Searching whole articles for mentions of whistleblowing returned very large datasets on the specific databases (upwards of 19,000 items). Limiting search to title, abstract and keywords focuses on items that are specifically about whistleblowing in software engineering. Selecting and classifying papers based on their title, abstract and keywords is an approach used in similar software engineering mapping studies (Glass, Vessey and Ramesh, 2002; Shaw, 2003). Not having whistleblowing mentioned in the title, abstract or keywords suggests an item is not primarily about whistleblowing.

3.8.3 Reliability

The final threat is about reliability and concerns to the extent to which the data and the analysis is reliant on the researchers and if we misclassified the items. The extraction of data was conducted by me and was reviewed and discussed frequently with my then supervisor (Marie Angela Ferrario), both of us are experts in the field of Software Engineering. Only two researchers were involved in the thematic analysis. To mitigate this the process included two rounds of independent coding and calibration, where there was strong agreement between the two researchers (Cohen-Kappa = 0.96).

Chapter 4. WISE Framework

Objective 2 is to design a research framework to systematically analyse features of software engineering stories and populate case study reports as described in Chapter 5. This chapter describes the development and use of the WISE framework. The WISE framework is based on existing social science theories and whistleblowing models discussed in Chapter 2. The WISE framework is developed for the capture and analysis of in-practice whistleblowing situations, based on primary data from software engineers directly involved in a situation.

4.1 Escalation Boundary Model

When I first began researching harmful situations in software engineering, I needed a method to analyse a story and compare and contrast actor actions between stories. Anvari et al. (Anvari *et al.*, 2019) present a simple whistleblowing boundary escalation model that shows how whistleblowing actions escalate upwards from within a team through the organisation, industry and into the public domain. I recognised this simple, yet effective diagram could be adapted for the software engineering domain.

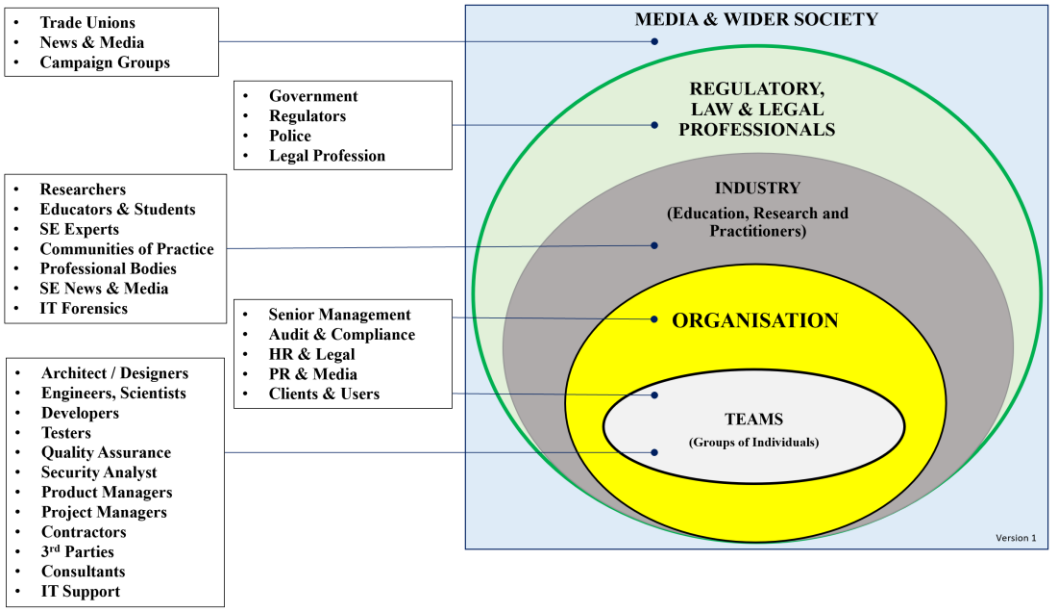


Figure 4-1 Escalation Boundary Model for Software Engineering (Version 1)

My software engineering boundaries are initially informed by grouping lists of actors referred to in whistleblowing stories and studies from existing software engineering literature (see Section 1.4, Section 1.5, Table 3-3, and Table 3-5). An actor is placed

internally or externally to the organisation on version 1 of the escalation boundary model. Initially "industry" and "SE community" groups were in one ellipse outside of the organisation. The choice to split this into Industry and SE Community developed during analysis of stories involving actors outside of specific industry domain but still in the software community more generally, for instance researchers, professional bodies, and technology media.

The boundary escalation model can be annotated to represent interactions between actors in a story, with arrows enumerated in reference to an interaction (step) between one or more actors in a story. Specifically, it can show escalations of a situation upwards, either internally or externally of the organisation. For example, actions and interactions actors take are identified in a story, an example of which is shown in Table 4-1.

Table 4-1 Actor Interaction Data (Example)

Step	Actor 1	Action	Actor 2	Software Artefact	Assessment
1	Engineering Team	*warn Software Engineer* about competency of the Engineering Manager	Software Engineer		Warning
2	Software Engineer	*checks* with others in *Engineering Team* about using *WinOS* in *ventilators*	Engineering Team	WinOS Ventilator	Raise concern to colleagues
3	Software Engineer	*reports* concern of *WinOS* in *medical devices* to *Engineering Manager*	Engineering Manager	WinOS Ventilator	Report up to Manager
4	Engineering Manager	*Engineering manager* suppresses concerns	Software Engineer	EULA for WinOS	Assessment of licence
5	Software Engineer	*checks* with *professional body*	Professional Body	EULA	Request advice

The interaction data set can be searched, filtered, and reported on to populate diagrams such as the escalation boundary model and case report. Narrative quotes from interviews can be linked to each step and populate relevant case report sections. This step data can be used to annotate an escalation boundary model, as shown in Figure 4-2. Here we can see how, having had their concerns ignored, the Software Engineer seeks advice and escalates their concerns to a Professional Body (Step 5).

This escalation boundary model underpins the WISE analysis framework, which in turn supports the design and production of whistleblowing cases in this thesis. The WISE analysis framework provides the basis for classifying the actors and their behaviours (actions and interactions) found in any story. The challenge of the escalation model and interaction data was how to represent actors with multiple group identities. For example, an actor may be a software engineering *contractor* in an organisation and so identifies themselves as in a different group to permanent software engineers who are *employees*. However, as part of their role as a software engineer in the organisation, they are part

of the engineering group developing a software system. Actors may also identify with a professional body or a campaign group within the wider community (e.g., groups within the vicinity of a nuclear power station). It is therefore important to identify which group(s) an actor identifies with when analysing narratives for decisions made and actions taken.

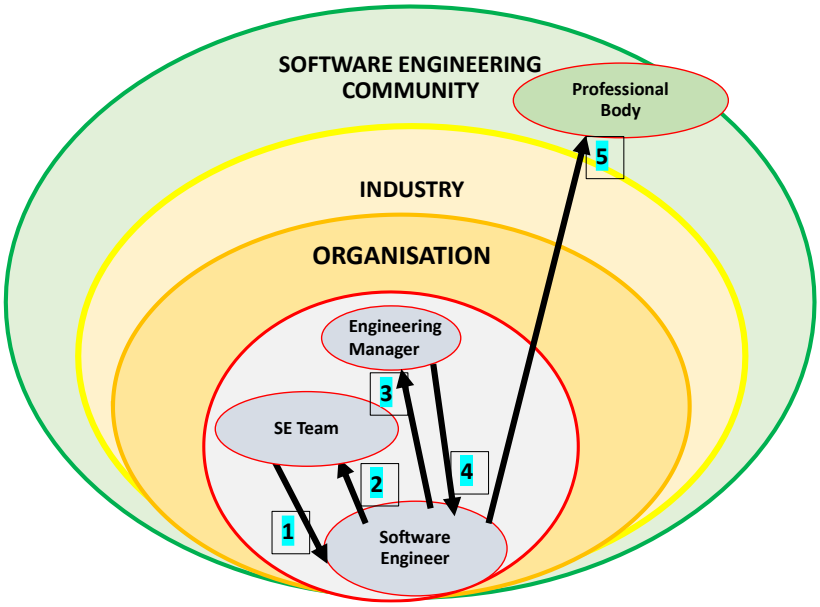


Figure 4-2 Annotated Escalation Boundary Model (Table 4-1 data)

The next section describes each section of the WISE framework and its use to analyse and classify a software engineering story:

(1) Classification of whistleblowing situation and harm in story
(2) Identify actors (stakeholders and software artefacts)
(3) Identify software engineering practice aspects
(4) Identify actions taken by any stakeholders
(5) Identify escalations and whistleblowing by stakeholders
(6) Identify steps taken by actors to assess situation
(7) Identify actors use of "*values and standards*"

4.2 Design of the WISE Analysis Framework

RQ4 guides the design of my WISE analysis framework, with the focus on structures for representing actors and their behaviours in whistleblowing situations. The actions and actor behaviours are further categorised using existing whistleblowing models, theories, and existing research data sets (e.g. Ipsos survey data and actions (Miller and Coldicott, 2019)). This section describes the seven tables (structures) used to represent story data that enable me to answer RQ5 and its sub questions.

4.2.1 Classification of Situations

Table 4-2 shows the four types of government recognised public interest harmful situations that whistleblowers can report in the UK. Stories from the research interviews must include one or more of these types of harms to be included in the study. For

instance, a story may include the original wrongdoing (e.g., not correcting testing failures) plus the subsequent covering up the failures (e.g., backdating test reports).

Table 4-2 Classification of Whistleblowing Situations

ID	Type	Examples of include
1	Legal	Criminal offence, miscarriage of justice, breaking law
2	Health and Safety	Someone is in danger
3	Environment	Risk or actual damage to environment
4	Covering up	Covering up wrongdoing on any of the above

4.2.2 Classification of Actors

Actors are the stakeholders and software artefacts in each story. For each story, the stakeholders were identified and classified into one of the six levels shown in Table 4-3, based on the boundary escalation model described in Section 4.1. Level 0 was added to classify software artefacts referenced in stories. Level 1 is specifically technical teams and individuals within the organisation directly involved with the design, development and support of the software or product found at Level 0. Level 2 represents all other roles and teams within the organisation. Level 3 represents industry domain stakeholders such as rival suppliers. Level 4 represents the broader software engineering community and professional bodies outside of the organisation. Level 5 represents regulators and legal professionals. Finally, level 6 represents wider society and includes the general public, media, and campaign groups.

Table 4-3 Classification of Actors (Stakeholders and Software Artefacts)

ID	Description	*Examples* include
0	Software Artefacts	Code, software, systems, and documentation
1	Individual or Team (Technical)	Software Engineer. Test Engineer. Individuals or teams directly involved in software product.
2	Organisation	HR, Senior Management, Audit, Compliance
3	Industry	Distributor, rival supplier
4	Software Community	Practitioners, Consultants, Educators, Researchers, IT Professional Bodies
5	Regulation and Legal	Police, solicitors, government regulators, external auditors, and standards bodies
6	Wider Society	General public, media, campaign group

4.2.3 Software Engineering Practice Aspects

Software artefacts are classified as actors in a whistleblowing situation, as described in the terminology in Table 1-2. Guided by the Software Engineering Book of Knowledge (P. Bourque and R. E. Fairley, 2014), and through the analysis of the stories, I developed a list software engineering aspects based on software development lifecycle artefacts, tools and processes alongside technical skills and management of the software engineering processes as shown in Table 4-4. These software aspects can be marked up against actor interaction data as shown in the artefact column in the sample of actor interaction data shown in Table 4-1, extracted for story narratives.

Table 4-4 WISE Analysis Framework: Software Engineering Practice Aspects

ID	Software Engineering Practice Aspect
1	Code
2	Software
3	System, product, or service
4	Documentation
5	Agreements and contracts
6	Standards and regulations
7	Processes and practices
8	Software engineering tools
9	Technical skills of teams and individuals
10	Management of software engineering team

4.2.4 Behaviour: Classification of Actions

For each actor, actions taken are identified and coded using the list in Table 4-5. This list is developed from a number of sources. The list starts with actions of committing and observing wrongdoing or harm – the trigger for any situation. I include the five actions from (Anvari *et al.*, 2019) of silence (take no action), conform (comply with instructions), dissent (attempt to change team) and exit situation (team or company). Finally, the six actions reported by Ipsos add three additional entries (Table 1-3).

Table 4-5 WISE Framework: Actor Behaviours

ID	Actor Actions	Event	Anvari	Ipsos
1	Wrongdoing or harm	X		
2	Observe wrongdoing or harm (discover)	X		
3	Take no action (silence)		X	X
4	Comply with instructions (conform)		X	
5	Change team behaviour (dissent)		X	
6	Report concern to external body			X
7	Report concern to a manager			X
8	Raise concern with a colleague			X
9	Exit team		X	X
10	Exit company		X	X

The focus of this thesis is on the escalation actions, I have therefore categorised escalations in more detail based on the escalation boundary model levels. The eight specific actor levels identified are each shown as a reporting point in Table 4-6.

Table 4-6 WISE Framework: Actor Escalation Actions

Level	Escalation Action
1	Report to individual
2	Report to team
3	Report to management
4	Report organisation wide
5	Report to industry
6	Report to software engineering community
7	Report to regulator
8	Report to wider society

Alongside actor actions, the narrative is examined for evidence of the assessment of the situation by actors prior to making decisions and taking actions, as shown in Table 4-7. This is based on Keenan and McLain's 7-step assessment process (Keenan and McLain, 2017) described previously in Section 2.2.

Table 4-7 WISE Framework: Actor Assessment (Keenan and McLain, 2017)

Step	7-Step Assessment process (for any actor)
1	Awareness of wrongdoing
2	Assessment of seriousness
3	Motivation to correct
4	Assessment of personal influence
5	Search for others to correct wrongdoing
6	Assessment of consequences for self and others
7	Assessment of management complicity

Finally, each actor action and associated narrative is reviewed for discussions of professional values and standards. A table of actions was developed iteratively during the design and implementation of the case study interviews as shown in Table 4-8.

Table 4-8 WISE Framework: Values and Standards

Action	Actions relating to values and standards
1	Is action causing harm (potential or actual)?
2	Is action covering up wrongdoing?
3	Is action breaching professional values or standards?
4	Is action upholding professional values or standards?
5	Is action supporting someone else upholding values or standards?
6	Is action evidencing harm or wrongdoing?
7	Is action in retaliation to someone else upholding values or standards?
8	Is action protecting others from consequences of whistleblowing?
9	Is action leaving evidence in software engineering artefacts?
10	Is action mitigating or correcting wrongdoing in software and code?
11	Is action one of refusing to sign off on non-compliant solutions?
12	Is action tampering with software engineering artefacts?

4.3 WISE Case Report

Each case report is made up of the following sections that bring together background research into the case, data collection from interviews, and triangulation of data alongside an analysis of the story based on the categories described in the WISE framework.

Table 4-9 Case Study Sections

Section	Content
Introduction	A short introduction to the actor providing data for the case along with key features of the case
Overview and Triangulation	A summary of the story including the actors discovered, and any key regulators. The content of this section is taken from the case interview and background triangulation research.

Section	Content
Actors	Presents details of all the actors (individuals, groups, organisations) identified in the story.
Public Interest, Potential Harm	Constructed from background research into the context and harms described by the actor.
Software Engineering Practice Aspects	This section presents the identified software engineering practice aspects of the story and any breaches of professional values and standards.
Actors and Actions	This section summarises actors who are aware of the situation. It looks at who does and does not take actions or respond to actions. Quotes from the interview narrative are used to evidence the assessments, decisions, and actions.
Outcomes	This section summarises how the story ends; reflects on actors impacted by the story.

4.4 Use of WISE Framework

Chapter 5 describes the design of my case study approach and the collection of data from interviews with software engineering experts. After transcription and anonymisation of interview transcripts, the interviews were analysed guided by the ORID approach (facts, feelings, insight, decisions), WISE analysis framework and the case template. My *units of analysis* are the interactions that happen at an individual, team and organisation level (internally and externally); specifically seeking to explore the event triggers, decisions, actions, and interactions between actors. Due to small number of interviews and uniqueness of the context of each situation, a thematic analysis is not appropriate. Due to a necessarily small number of available interviewees given the context of the research around individual actions of whistleblowers we cannot make generalisations; nor is that the purpose of this research. The WISE analysis framework provides the basis for a deductive analysis to identify the actors and their actions in each story. This data set creates an abstracted version of each story that supports the presentation of stories, through structured case studies, in a format that can be explored and compared using existing whistleblowing models and theories.

4.4.1 Facts from Interviews

The first step in the WISE framework analysis is to identify the individual stories from the interviews (some participants recounted more than one story and some anecdotes). For each story I identified all the harms or wrongdoing, actors, and technical artefacts (code, software, systems in the software development lifecycle) discussed. This was initially done with post-it notes and highlighter pen in printed transcripts as shown in Figure 4-3 and Figure 4-4 below, with data obfuscated to protect participants. This data was recorded in an excel spreadsheet. This actor identification activity was repeated in nVivo. This enabled me to cross-check for gaps with the post-it based spreadsheet data. All subsequent data analysis was done in excel workbooks, Microsoft Visio diagrams and hand drawn figures. I did not feel it necessary to use software such as nVivo to support the analysis of the stories. I wanted to embed myself in the data and story narrative to maintain a close connection with the specifics of each story throughout the analysis and on into the case development. Additionally, Excel and Visio were more flexible ways of developing and iterating on the presentation the data analysis.

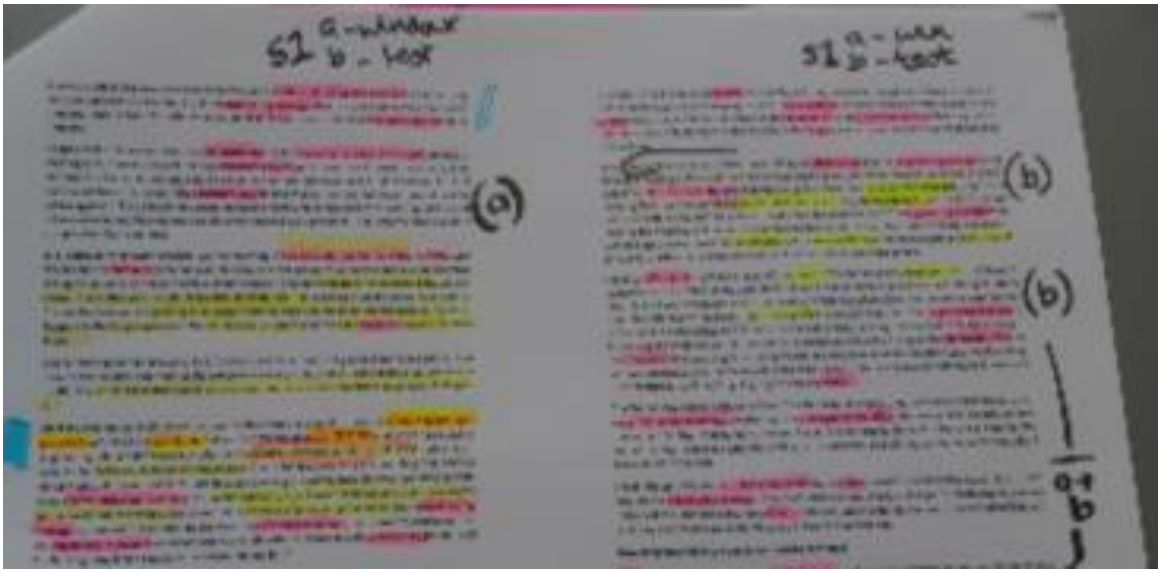


Figure 4-3 Highlighting Stories, Actors, and Actions in Interview Transcript

Figure 4-4 List of Actors and Software Engineering Aspects

The next step was to look at the interactions between the actors and artefacts – to extract the story events (actions). This was first done by highlighting the transcripts and using post-it notes as shown in Figure 4-3 and Figure 4-4, that were then transferred into a spreadsheet. For each actor, I identify what actions and interactions were reported or taken by the participant in relation to software engineering aspects (e.g., the code, an end user license agreement). This story data set is then systematically analysed against each of the WISE Framework classification tables to support the development of the case studies and to answer RQ5.

4.5 Summary

This chapter gives a summary of the categories in the WISE framework and how they are developed and used for the analysis of whistleblowing stories in software engineering. The framework is adaptable, future versions could incorporate new types of actions and findings from other whistleblowing research studies. The framework could also be developed for other domains such as medicine, where the actors, their escalation hierarchy and practice specific aspects would need to be developed to replace the software engineering actors and aspects in the WISE framework.

Chapter 5. Case Study Design

Easterbrook et al. (Easterbrook *et al.*, 2008) reflect that software engineering researchers often select methods with little understanding of the purpose of a given method and its relative strengths and weaknesses. This chapter presents the justification and design of my case study research. It shows how I am guided by leading case study researchers in social science and specifically software engineering (Runeson *et al.*, 2012). Figure 5-1, based on (Yin, 2018), describes the six activities for developing and implementing my case study research method – plan and design study, prepare to collect data, collect data, analyse data and present findings.

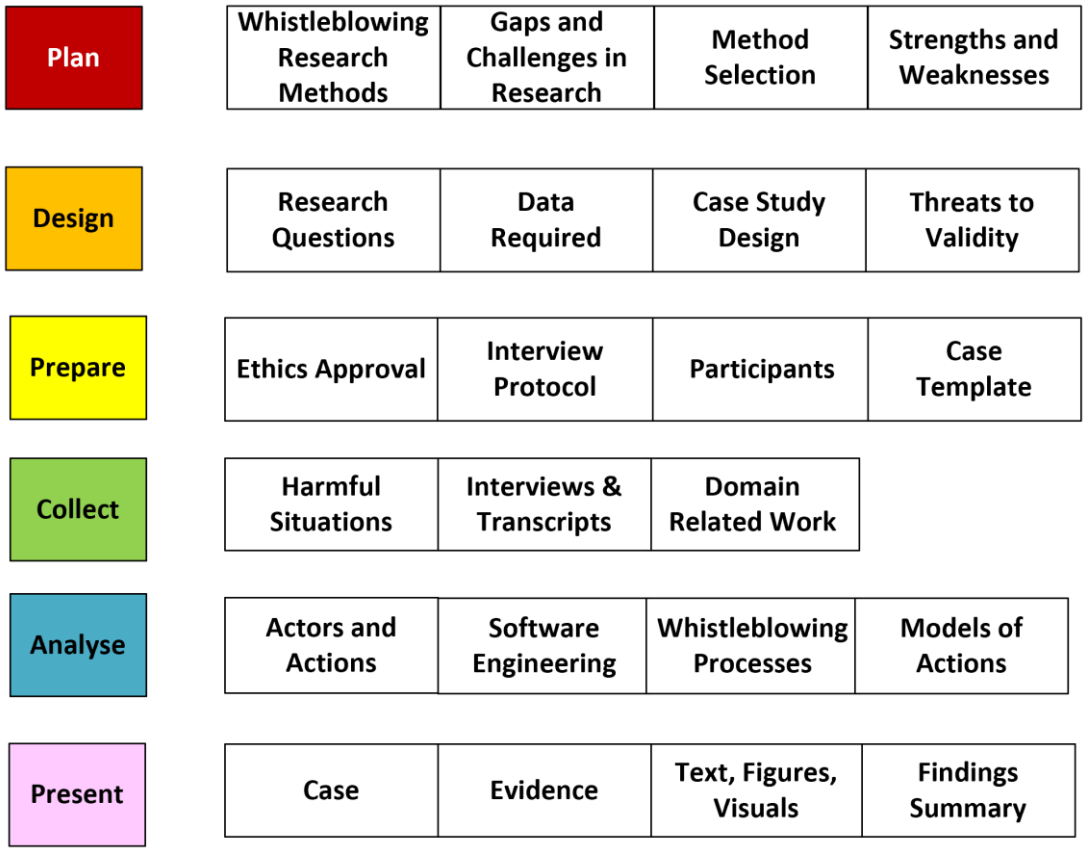


Figure 5-1 Activities to Plan and Implement My Case Study Research

Section 5.1 (plan and design) reviews methods used to study whistleblowing in software engineering from RQ2 and RQ3 and describes why I selected case study research to address research gaps and answer RQ5. **Section 5.2** (**preparation and ethics**) reports on the preparation for the research, including the ethics and recruitment

of participants. **Section 5.3** describes how the ORID interviews were designed and piloted. **Section 5.4** describes the protocol for the main case study interviews. **Section 5.5** describes how the collected study data was analysed, including the use of my WISE analysis framework. **Section 5.6** describes how the three case studies were selected. **Section 5.7** discusses the quality and validity of the research.

5.1 Plan and Design

Investigating how and why whistleblowing is, or indeed is not, happening in software engineering practice requires carefully designed studies, sympathetic to the complexities and risks associated with those involved in any whistleblowing situations.

5.1.1 Reflection on Methods

I first discuss five of the most relevant empirical software engineering methods as guided by (Easterbrook *et al.*, 2008) and how they relate to my selection of case study research methods. Controlled experiments do not satisfy Objective 3 which is the study of real-world software engineering experiences. As discussed in the literature review findings (Section 3.6.2), ethnographic and action research methods were not found, likely due to challenges with finding willing organisations and participants and the length of time a study may have to run to discover a whistleblowing situation. Additionally, the consequences (for me, participants, and their organisations) on discovering whistleblowing situations in an organisation are unknown. Any research would have to be done with the consent of a team or organisation. The seriousness of actions up to and including whistleblowing may mean individuals and organisations adjust their behaviours during any observation or action period. This would all affect the validity of any data collection.

I explored designing a survey to capture stories. On trialling this with my then supervisor (Marie Angela Ferrario), even with the simplest of stories we found it difficult to capture the level of data required to answer my actor and actions-based research questions without making a survey long, unwieldy, and difficult for participants. Keil et al. (Keil and Park, 2010) reflect on the challenges of research studying the complex phenomenon of whistleblowing and the subtlety of detecting factors in written surveys, they propose future work to study "*failed IT projects*" and interview IT professionals on these projects. I concluded case studies, based on individual participants' stories, in a neutral setting after the event, was a suitable and effective way to take this research forward given the time and resources available.

5.1.2 Selecting Case Study Research in Software Engineering

Case studies can be used for investigating software engineering practices and are characterised by their flexible nature, multiple forms of data collection informed by qualitative data (Talbot, 2016). Case studies exist widely outside of academia, appearing in magazines, documentaries, and newspapers as ways to engage viewers in a topic and present specific cases of interest to an audience. Case studies are found in education to support professional development activities presenting evidence of in practice experiences and learning opportunities. Corporations produce case studies to market their products and services. Yin describes these as examples of *non-research* and *teaching practice* case studies, for which no explicit research procedures may have been followed to create them. Case studies may therefore not always be recognised as

a formal research method. Dozier, Miceli and Near highlight that whistleblowing is a dynamic process, and studies should seek to account for the actions of all actors involved, not just the whistleblower. Unlike experiments that isolate and tightly control aspects of scenarios, my case study is primarily based on the reality and interpretation of events, actions and interactions with other actors as described by participants. Yin advises that conducting *case study research*, demands a higher set of expectations and procedures to be managed and demonstrated, as detailed in these foundational statements:

1. Investigates a contemporary phenomenon within real world context,
2. Boundaries between phenomenon and context may not be evident,
3. Illuminate decisions or set of decisions,
4. Many more variables of interest than data points,
5. Benefits from prior development of theoretical positions to guide design
6. Relies on multiple sources of evidence, with triangulation of data.

My research meets the first five of these statements. Whistleblowing is a contemporary phenomenon, and my research is done in real world context. The context of the organisational situation and making decisions to whistleblow is difficult to separate. A particular actor may behave differently to the same triggers in a different organisational context. The case studies have many factors and variables of interest, and I have no control over those factors and variables present in any situation under study. I take a small set of data points (actors, their interactions and software engineering aspects) from the stories collected in interviews. I use existing theories to design the WISE analysis framework which guides the data collection and analysis. The nature of whistleblowing makes item 6, relying on multiple sources of evidence, challenging. Specifically, my research is focused on data from the perspective of a software engineering expert with observations of other actor actions and behaviours. I conduct triangulation of data through open-source research and present it in each case, being mindful of protecting individuals and organisations in the case story.

To address RQ5, interview data is collected and analysed using the WISE framework. The findings and presented in a case report, based on the WISE analysis framework described in Chapter 4.

5.2 Preparation and Ethics

This section reports on the preparation for the case study research. The research was undertaken with the full understanding of the participants involved, using their expert software engineering knowledge to explore situations they were personally involved in.

5.2.1 Ethics Approval

The research project required ethical approval as data is collected and analysed from humans. Ethical approval was granted from Lancaster University's Faculty of Science and Technology Research Ethics Committee who oversees the ethical review of proposed research. Copies of the application paperwork is in Appendix Ethics Approval.

1. Story interviews and workshops (FST17072)
2. Expert interviews (FST19079, FST19150)
3. Approval for analysis of secondary Ipsos dataset (FSTREC20115)

5.2.2 Participant Recruitment

This section describes the approach taken to recruit study participants. In the literature review study items, up to 3,000 participants (software professionals or students) took part in predominantly lab-based experiments. I cannot be sure of exact numbers as I do not know if any of the studies overlapped. In my research I am looking specifically for participants who are software engineering experts and have been involved in harmful software engineering situations with a public interest and whistleblowing aspects. I raised awareness of my research interests through professional body workshops and my presentations on topics such as "Do No Harm" and "Human Values in Software Engineering", a list of which are included in Table 5-1. My workshop participants were made aware of my research and were invited to be in touch with me regarding the sharing of their experiences of upholding values and standards in software engineering practice or to pass on my details to other who may be interested to speak with me. I attended workshops and presentations from software engineering individuals and groups with abstracts that indicated their sessions were about upholding values and standards in the tech industry.

Table 5-1 Events Participated in to Raise Awareness of my Research.

Dates	Event	Audience
2018 –2021	6 x Higher Education workshops – values in computing, do no harm. Face to face & online. (Lancaster, UCLAN, Leeds Trinity)	Students & Researchers N= 200
2019 - 2021	Software Practice Advancement specialist group of BCS. 3 workshops: Computing for Humanity, Values in Computing and Do No Harm? [London, Leeds, online]	Industry N=70
2019	Workshop on Empirical Software Engineering in Industry – BCS Manchester Branch.	Researchers & Industry N=40
2019	Biggest Failures in IT Security Dagstuhl, Germany	Researchers N=30
2021	BCS Webinar: Do No Harm?	Industry, students, researchers, N=20
2022	Private workshop for a software engineering consultancy, online.	Industry N=10

5.2.3 Participant Selection

Of an estimated total audience of 370, made up of computer science students, researchers, educators, and software engineering professionals, 18 people contacted me to express an interest in my studies. Many people I spoke to at events genuinely felt they had not seen harmful situations where professional values and standards were breached such that it was a public interest concern. At my workshops there was an element of "*it is rare, but it does happen*". People reflected on near misses and how issues had been overcome by raising concerns and taking actions to mitigate the harm. However, a reflection does not lead to a willing participant, or a participant may not have sufficient recall of what happened to make the story a suitable incident that I can systematically study. I was conscious not to probe people about their personal stories openly at events. People were in a public space (predominantly online), it would have been wrong to discuss their situations openly as we could not know who else was

present, listening, recording, or watching. However, when approached, I would take the conversation somewhere private from the main event. If the following criteria were met, and the potential participant is a software engineering expert, they were invited to take part in the study:

- Public interest aspects to a situation
- Harm or wrongdoing situation in software practice

People excluded from this study: Two people approached me regarding situations in academia, one was involved in an ongoing case. I was able to give some general advice and useful resources. Neither met the software practice criteria for this study. Two potential participants were public whistleblowers – Alison McDermott (former Human Resources Consultant to Sellafield) and Peter Duffy (former Consultant Surgeon at University Hospitals of Morecambe Bay NHS Foundation Trust). Peter's case involved elements of reporting up of issues relating to the misuse of technology including emails and NHS records[30] as reported on by Computer Weekly. I privately discussed these two on-going cases (over four hours with Peter and ongoing emails, similar with Alison). Their experiences of the nuclear industry and the NHS did contain elements of technology issues, though not specifically about software engineering, so on this aspect alone they were not candidates for participation. Additionally, neither Peter or Alison are software engineering experts, their cases are ongoing and involve public media campaigns and employment tribunals. I sat in on Alison's Employment Tribunal (December 2021). I developed an informal analysis of Alison's story. In both Peter and Alison's cases aspects of their stories (the harm seen, actions and interactions with other actors) could have been analysed with my WISE framework, but this would be future work and would involve the WISE framework being adapted to include actors and escalation boundary models relevant to their specific industry roles and domains.

Exclusions from study: Consent and selection for participation was designed to be clear, unambiguous, and auditable. I spoke at length with potential participants before and after consent was given. Together we considered the consequences of participation, and the impact on their wellbeing. The research is not about exploring individual personal grievances at work, although there were possible elements of this within the data as there is a focus on the tensions between individuals, teams, and organisations. The research was not an opportunity for an intervention into an ongoing situation, such expectations would put me and the participant in difficult situations moving forward. People involved in on-going court cases, employer disputes, or seeing me as an intervention to a situation were therefore not invited to participate in the study.

My research explores software engineering situations with a public interest aspect to it. If potential participants were primarily recounting anecdotes or stories without some public potential public risk from harm or insufficient detail for me to capture and analyse, they too were excluded. In summary, the following exclusions were applied:

- Participants involved in on-going court cases or employment dispute,
- Participants seeing me as a whistleblowing mechanism or intervention,
- Participants presenting short anecdotes or second-hand stories,

[30] https://www.computerweekly.com/news/366539133/Medical-regulator-drops-probe-into-NHS-whistleblower-Peter-Duffy-amid-dispute-over-email-evidence

- Low public interest aspects or risks from software or IT projects

5.2.4 Participant Scoping Discussions and Interviews

I had scoping discussions (via a phone call, email, or face to face) with nine potential participants. I explained my software engineering background and the focus of my research into harmful situations seen in software engineering practice. I explained that their interviews are not being published but would form part on an anonymised data set, also not published, that I would use for my analysis. I sent participants an information sheet and consent form (Appendix Consent Forms). Forms were returned via email (5) or by post (2) before setting dates for interviews. Two potential participants dropped out at this point. I conducted semi-structured interviews with 7 software engineering experts, the design of semi-structured interviews is discussed in the next sections.

All interviews were conducted and recorded via Microsoft Teams or Zoom. I had the backup of a local digital voice recording device. In total there were over 14 hours of scoping discussions and recorded interviews with participants, alongside many emails directly and indirectly linked to the research. In some cases, it was up to four months between the first contact and the interview. In others it was within a couple of weeks as detailed in Table 5-2.

	P1	P2	P3	P4	P5	P6	P7
Date first contact	11/20	09/20	11/20	04/21	05/21	10/21	02/22
Date of prelim chat	01/21	10/20	12/20	04/21	05/21	11/21	02/22
Date of interview	03/21	03/21	04/21	06/21	06/21	12/21	03/22
Number of emails	52	96	35	40	12	165	42
Phone calls / zoom / in person talk	2	2	4	2	2	3	3
Length of interview (mins)	92	91	100	168	80	171	61
Length of interview (words)	9,384	12,085	11,735	23,289	6,791	20,923	5,949

Table 5-2 Participant Engagement Summary

The interview transcripts are analysed using the WISE analysis framework, as described in Chapter 4. First a section on the design and structure of the interviews themselves and the pilot study (Section 5.3.1) that was conducted to prepare myself for designing and running the main interviews.

5.3 Interview Planning, Design, and Pilot

The design of my interviews were guided by two key books – Kvale's " InterViews: An Introduction to Qualitative Research Interviewing" (Kvale, 1994) and Stanfield's "The Art Of Focused Conversations" (Stanfield, 2000) provides practical guidance for running research interviews, it also provides a conceptual framework for how to think about interview research and focuses heavily on the methods, planning and preparation required for running successful interview studies. My interview planning and design is based on Kvale's seven stages of interview research as show in Table 5-3.

Kvale's stages 1 and 2 have already been addressed in the first three chapters of this thesis, where I have reported on my literature review findings, developed research questions, and clarified the scope and purpose of using case study research. Stages 5, 6 and 7 are addressed in later sections of this chapter. This section is specifically looking at stage 3 and 4, how to design and conduct interviews to capture the data required to answer my research questions guided by Stanfield and Kvale's books.

Table 5-3 Stages of Interview Research (based on Kvale's 7 stages)

Stage	Description
Thematising	Review existing research literature for current knowledge. Develop themes to be investigated. Clarify purpose of research.
Designing	Methodologically well-controlled design for obtaining intended data and knowledge.
Interviewing	Craftmanship and the role of the researcher. Conducting interviews based on an interview guide.
Transcribing	Preparing interview material for analysis.
Analysing	Appropriate methods of analysis.
Verifying	Generalisability, reliability, and validity of interview findings.
Reporting	Communication of findings. Consideration of ethical aspects.

The structure of my interviews follows Stanfield's "ORID" framework, described as a method of leading people through phases of facts and reflections, enabling them to process their experience. The process involves asking four layers of questions, with each layer building on the previous. It is based on the principle that people first need to be cognisant of the facts, to then deal with their emotional responses (reflections) on the situation to then undertake analysis and decision-making about the situation. It is shortened to ORID, with each letter representing a phase in the process:

- 'O' for objective – known facts
- 'R' for reflective – how people feel
- 'I' for interpretive – what insights, issues, or challenges identified
- 'D' for decisional – decision or response – what should or could we do

Specifically, I was looking at ORID for its 1) usability by researchers, 2) repeatability across participants, 3) adaptability for different stories, and 4) usefulness for helping participants recall and explore a story.

5.3.1 Interview Pilot Study

A pilot study contributed to the design of my main interviews and gave me practice at conducting ORID interviews (an activity recommended by Kvale). Key learning aspects regarding the pilot for me were seeing how the focus on the facts of the story grounded all the subsequent discussions about feelings and insight into the situation and appeared helpful for participants. Pilot participants were local students, academics, and software professionals interviewed about their reflections on a software engineering-based whistleblowing story. Pilot participants are not based on people involved in the actual study interviews. This process is to prepare me for running interviews.

With ethical consent granted, I recorded face to face interviews with six participants in a room at Lancaster University. Participants were made aware the interview topic was to be about the Facebook Cambridge Analytica story, and that we would be reviewing

an interview with Christopher Wiley. Participants were given the option to suspend the interview if they felt this was not something they wanted to listen to or discuss. As part of interview, participants were shown a publicly available YouTube video (13-minutes) of Christopher Wiley[31], talking about his role as a whistleblower at Cambridge Analytica and the misuse of personal data. Participants were interviewed about the story, guided by pre-prepared questions tied to the ORID framework, shown in Table 5-4. The interview was semi-structured and flexible to the participant responses. However, the overarching ORID process was adhered to within the time constraints (one hour) to trial all sections of the ORID approach. The transcription, analysis, and findings from the pilot study (how people respond to a public interest whistleblowing story) are tangential to my research questions and are not presented in this thesis.

Table 5-4 Pilot Interview Questions

ORID stages	Sample questions / question guide
Topic	Facebook Cambridge Analytica Whistleblower - Christopher Wiley
Objective	**Facts:** Who did you watch? What did they talk about? How did they sound? Who are main characters in the story? Recall of facts about the story (names, dates, places, money). What words stood out for you?
Reflective	**Feelings:** How did watching the interview make you feel? Reactions or emotions? Anxiety, concerns, inspiring? What are the most critical parts of the story and why? Did anything surprise you? How did you feel at the end?
Interpretive	**Personal:** What key points were made – why are they important? **Organisational:** What underlying organisational issues do you think there are? What motivates Christopher Wiley / others in the story? How do you think people at Facebook or Cambridge Analytica feel? **Society** Who do you think is responsible for what happened? How might this story affect our futures – self / society – threats or opportunities?
Decisional	**Decision:** What can we do about this situation? What do you think needs to be protected / defended / secured? What can software engineers / researchers do? What can wider society do?
Debrief	How did you find the sound, video, and the story? How did you find the session structure?
Follow up (+ 2 weeks)	Strength of feeling about story and any actions taken as a result of session

As part of the interview, participants recalled personal stories linked to Facebook, that in one case brought up an emotional response that they need time to think about before discussing. At both a practical and personal level, I had to learn to bite my tongue in the interviews, I was not there to discuss my thoughts and opinions on the Cambridge Analytica story. I was there to be a researcher, to follow the interview guide as designed.

[31] https://www.theguardian.com/uk-news/video/2018/mar/17/cambridge-analytica-whistleblower-we-spent-1m-harvesting-millions-of-facebook-profiles-video

I learnt to become comfortable with silence in an interview while participants made notes or thought about their responses to questions. I experienced a technical failure of my dictaphone (failure without an audible alert), this was awkward as 15 minutes elapsed and went unrecorded. We restarted the interview, we attempted to repeat the missing 15 minutes and then continued with the rest of the interview.

Based on the structure and experience of running the pilot, I developed ORID style interviews for the main case study interviews.

5.4 Main Case Study Interviews

The interviews with my case study participants were designed in four sections. The two main interview sessions (participant biography and story) are topped and tailed with the administration of running the interview and explaining what happens next, as summarised in Table 5-5.

Table 5-5 Interview - Main Sections

Section	Description
1. Introduction and logistics	Introducing myself. Setting the scene of my research. Pause or withdrawal protocol. Timings. Technical failure or interruption. Personal life or career areas not to discuss or refer to?
2. Biography	Participant career, involvement with professional bodies.
3. Story	Facts, feelings, insight, and decisions (ORID)
4.What next	Transcription (by me) and review (by participant) of interview. Opportunity to correct, redact, delete, or add to transcript.

During the first ten minutes of each interview, I briefly introduced myself and set the scene of the research. During interviews I had a distress protocol in place for participants wishing to terminate an interview or withdraw from the study, though this was not used. I explained that if participants wanted to pause or withdraw at any point that is OK and we will cease the interview. I set a plan in place for any technical failures or interruptions. Reliance on the internet connection and technology was a concern for me, although overall the flexibility outweighed issues (location, time available, travel) of arranging to meet in person. Online presented the challenge of detecting frozen screens – I made sure to nod and "umm", so participants knew I was still present in the online room. I also kept a close eye on movement and sound from participants, I had two incidents of lost internet connection and of Teams not alerting me that a participant was waiting to be let into the room.

Distress Protocol: I discussed with participants the stories we were planning to cover in the interviews. I asked participants if they envisage any discomfort or issues coming up because of discussions around the story or if there are any personal life or career areas they do not want to discuss or refer to. As part of the debrief I asked participants about the experience of the interview and any issues they wish to discuss. Participants could ask to hear or see their interview recording. Participants could also view, comment on, and amend their interview transcripts. Some participants asked for sections of interview content to be withdrawn, transcripts were marked to indicate which areas of consent is withdrawn from (e.g., a discussion of family or personal health issues). This data was not quoted or referred to in any of the subsequent analysis or case

reports. Transcripts were anonymised and organisations obfuscated. All analysis and case studies referred only to a participant or organisation by letter (A, B, C).

5.4.1 Case Study Main Interview – Section 2 - Biography

Section 2 of the interview recognises that whistleblowing episodes do not exist in isolation. I therefore look to capture elements of a participant's life before and after the episode. It acts as a warm up where we informally discuss their career, skills, and any engagement with professional bodies. It provides a further opportunity for participants to disclose sensitive areas of their lives or careers not to be discussed, or to be discussed but not recorded or analysed. Guide prompts are shown in Table 5-6.

Table 5-6 Section 2: Biography and Warm Up Guide (20 mins)

Career - education, industry sectors, roles, skills, and years of experience **Specialisms** – e.g., security, safety critical, high reliability, finance, health **Technology** – embedded systems, security, coding languages, platforms etc **Contracts** – permanent, temp, consultant, service provider, freelance. **Membership** of professional bodies. Your interests in IT / computing / technology came from where? How has technology / IT profession changed since you started?

5.4.2 Case Study Main Interview – Section 3 – Story Capture

Section 3 of the interview is based on the ORID facilitation technique and guides the participant through their story. Having spoken and emailed participants prior to the interview I was aware of the story to be covered. The first part of section three contains the objective questions, focusing on gathering facts about their story – the actors involved, the software, the harm or wrongdoing observed, the sequence of events and any known outcomes. The guide for the objective questions is shown in

Table 5-7. With the key facts of the story captured, I move to explore a participant's reflection on their story guided by questions in Table 5-8. During the interview, additional facts were sometimes remembered out of sequence by participants, or additional stories or anecdotes were introduced.

Table 5-7 Section 3: Objective Questions Guide

Capturing facts, decision points, actions, and outcomes. Who (people, organisations) were involved? What software or software engineering practices involved? What points in the SDLC? What breaches, risks or harms did you become aware of, how and when? Who or what could be harmed if risks not raised or listened to? Was technology defective, vulnerable, malicious, or illegal? How did you assess risk of harm and impact? How did you gather evidence? What needed to change and how quickly? When did intention to report up become action? What did people suggest you do next? Who tried to prevent you taking action and how? What was overall outcome - was the harm / risk / awareness addressed?

The interview had to be flexible and move backwards and forwards appropriately between facts and reflection questions. If time was becoming constrained, I proposed that we return to additional anecdotes and stories in a follow up interview (this was not required).

Table 5-8 Section 3: Reflective Questions Guide

Reflection on story
How did you feel when you first discovered the breach, harm, or risks?
How do you think the situation came about and why?
Did you feel responsible for reporting the issue?
What motivated you to raise the issue? Did you have any allies?
How would you describe the culture of the organisation and their response?
How do you think other stakeholders felt and acted in this situation and why?
Did the organisation see you as threat?
How effective do you think your actions were?
How has the organisation and people involved changed since?
How has the episode changed your career?

Following the capture and discussion of the main story, the final part of the session asked questions about participant insight from the story and how the incident changed their ways of working then and now, as shown in Table 5-9. I also ask participants to think about what others could learn from their story.

Table 5-9 Section 3: Interpretive and Decisional Questions Guide

Interpretive
What are the key learning points from this episode?
How could outcomes have been different? (better or worse)
If you could go back in time, what would you do differently?
How could the organisation have responsibly disclosed the issues?
Decisional
Has it changed your way of working? Would you do it all again?
What should or could be changed or improved in SE practice?
How could we (SE community) learn from this?
(Policy, procedures, and processes; support, advice, and protection)

At the end of each interview, I describe the next steps of how I transcribe each interview and give the participant secure online access to their own transcript for corrections, redactions, deletions, or additions. Three participants followed up the interview with extra clarifying details regarding discussions in the interviews. One participant still regularly emails me on a variety of themes around whistleblowing. Three participants noticeably did not use the name of people or organisations during the recorded interviews or gave pseudonyms for them. ("*Let's just call him John. (He wasn't called John)*"). One participant was particularly keen to obfuscate the organisation involved as it had demonstrably retaliated against a former employer in an employment tribunal. The employee had spoken about an organisation's harmful practices during a court case, and in the ruling the judge agreed there were concerns to be investigated further, though ruled not directly related to the tribunal. Despite not mentioning the name of the organisations during the interviews, personal data collected from the interview enabled me to track down, through internet searches, the unnamed organisations. This underlines the importance of not publishing the interviews or my data sets and the careful anonymisation and obfuscation of case content.

5.4.3 Post Interview Activities

After each interview I would make personal notes and reflections about the interview and the interview process. I would draw a diagram to capture my first impression of the

story, the actors, the interactions, and the timeline of events. Examples are shown in Figure 5-2.

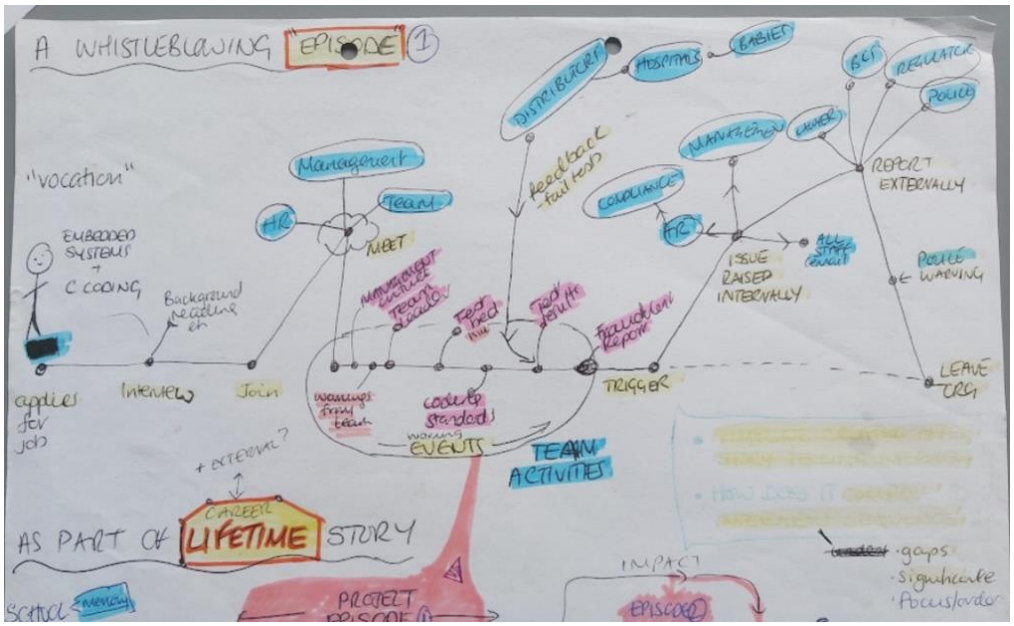


Figure 5-2 Hand Drawn Story Timeline – Post Interview

5.5 Use of WISE Analysis Framework

The study interviews were analysed using the WISE analysis framework and case template described in Chapter 4. My *units of analysis* are the interactions that happen at an individual, team and organisation level (internally and externally); specifically seeking to explore the event triggers, decisions, actions, and interactions between actors. I annotate an escalation boundary model, to show interactions between actors and giving me the first insight into the "shape" of the story. I make notes of public interest and harm aspects of the situations discussed and how they related to technical and software engineering aspects of the situation. I talked with my then supervisor (Marie Angelo Ferrario), about the interviews, their content (the main story and any other anecdotes), how I felt about the interviews and their inclusion in the study.

5.5.1 Triangulation of Case Story

Before and after study interviews, I conduct background research into participants and their situations. Before the interview it was useful for me to find out something about the organisation's field of work to understand something of the technical aspects of situations participants may discuss. After interviews I could triangulate and on some level corroborate the content of their story. For instance, I reviewed standards and guidelines, I retrieved public inquiry documents from the National Archives [32]. Participants shared their CVs with me, others had career timelines available on LinkedIn[33]. I searched LexisNexis[34] for court case records for any data relevant to

[32] https://www.nationalarchives.gov.uk/
[33] www.linkedin.com
[34] https://www.lexisnexis.com/uk/legal/

individuals or organisations. I looked through UK Parliamentary website [35] and Hansard[36] for references to the issues and stories in parliamentary debates, policy and law making. For protection of participants, I have not presented any specific content relating to my background research that might identify organisations or whistleblowers in the story. For example, in the health story, I provide background into the use and design of ventilators and the software to manage and control the ventilator. I provide an overview of actions taken by regulators, end users and manufacturers to correct issues. I provide references to medical studies about ventilator failures in medical practice. I do not present full details of field safety notices issued by the case organisation.

5.6 Selection of Three Cases

The participant interviews, and the stories therein, create a unique data set reporting on software engineering practitioner experiences of harmful or whistleblowing situations. Table 5-10 presents a list of the primary stories captured from expert interviews.

Table 5-10 Stories from Expert Interviews

Story	Title	Context	Section Reference
1	Ventilator Design	Health	Chapter 6
2	Defeating emissions tests	Automotive	Chapter 7
3	Safety critical systems	Nuclear power	Chapter 8
4	Train Protection System	Transport	Appendix E
5	Algorithms in Air Traffic Control	Transport	Appendix E
6	Systems integration	Transport	Appendix E
7	Poor practices in IoT solution	Smart City	Appendix E

Each of the stories in Table 5-10 are analysed for evidence of the four key Anvari whistleblowing actions of dissent, whistleblow, silence or exit as shown in Table 5-11.

Table 5-11 Whistleblowing Action, by Story

Anvari Whistleblowing Action	Story
Dissent – attempt to change team or behaviour	1,2,4,5,6,7
Whistleblow – seek external stakeholder to change team or behaviour	1,3
Take no action – silence or consent to behaviour	1,2,3,6
Exit – choose to leave team or organisation	1,4,6,7

For example, Story 1 contains all four actions, whereas Story 2 only has dissent and take no action. Stories 1, 2 and 3 from Table 5-10 were selected to develop into cases A, B and C as they give good coverage of Anvari's actions and of whistleblowing harms and wrongdoing types recognised by the UK government, as shown in Table 5-12. These three cases are sufficient to explore my research questions within the time and resource constraints of this PhD thesis. Extracts of stories 4, 5, 6 and 7 are included in Appendix E Story Extracts.

[35] https://www.parliament.uk/
[36] https://hansard.parliament.uk/

Table 5-12 Whistleblowing Types of Harm Coverage, by Case

Harm / Wrongdoing Type	Case
Law or Regulation breach	A, B, C
Health and Safety	A, C
Environment	B, C
Covering up	A, B, C

5.7 Quality and Validity

Guided by (Yin, 2018), in this section I look at the threats to validity (construct, internal, external and reliability) of case study research and my application of it to whistleblowing in software engineering research. I discuss the strengths, weaknesses and actions taken to mitigate limitations of studies.

5.7.1 Construct Validity

Construct validity assesses if appropriate operational measures have been identified for the concepts being studied. Studying the escalation of whistleblowing situations is complex. I use existing whistleblowing theories and studies to design and guide the data collection and analysis in my research as described in this chapter. My discussions with participants before, during and after interviews make it clear that they have been recruited to capture how potentially harmful public interest situations are discovered and escalated in software engineering practice. My ORID interviews are structured with the primary goal to objectively gather the harms and wrongdoing, the actors and actor interactions around software engineering aspects of a situation. A limitation for each interview, and thus case study, is that I only have one source of primary evidence (the participant) to capture primary data from. To mitigate this, I do conduct background research into the participant, the organisation, and the domain. As an example, in Case A, I conduct a review of studies of critical incidents with ventilators and I query regulator databases for incidents and product recalls. Additionally, all participants reviewed their transcribed interview and case participants reviewed the actor models developed for their stories. In both corrections to transcription or analysis of their stories were made – for instance gaps or errors in the description or responsibility of a particular actor.

5.7.2 Internal Validity

My research develops exploratory case studies. Yin says internal validity tests are for *explanatory* or *causal studies* only. However, I do reflect that in some of my participant stories there could be alternative or "*rival explanations*" for how and why a situation came to be and how other actors are observed or reported reacting to a situation. There may be missing facts about how a situation starts or escalates. On a couple of occasions in interviews, short anecdotes were shared regarding grievances about an individual or organisation rather than a public interest harmful software situation. All case studies contain situations described by participants as causing harm or wrongdoing. The UK government states whistleblowers can report situations *they believe* happened, are happening, or are to be happening soon. Open-source research into participants, the organisation, and the domain did not help to mitigate internal validity issues, as no previous studies of any of my participant's stories have been published. Future work would look to find multiple perspectives from different actors during or following a

whistleblowing incident and to seek access to software engineering artefacts to explore the situation.

The WISE framework data classification, particularly regarding actions, is based on existing social science theories and whistleblowing models. Other theories and models are available that may give alternate classifications for data and thus findings. The WISE framework and data classifications can of course be extended. Part of the research goal is to understand the applicability of existing theories and models. Gaps in existing models are identified with the finding of whistleblowing actions specifically relating to software engineering artefacts. This contributes to knowledge in the software engineering research domain.

5.7.3 External Validity

External validity deals with whether study findings are generalisable beyond the immediate study. A frequent criticism of case study research is that findings cannot easily be generalised. My case studies are not intended to be generalisable or to generate theories or whistleblowing prediction models applicable to a wider population of software engineers. Empirical experiments, such as those found in literature review, present generalised findings but are limited in their ability to predict or show how whistleblowing happens in practice. A large sample of participants can be recruited to test hypotheses through predetermined scenarios. Each participant gives a measure of their intention to "*report up*" on a controlled situation. The frequency of this occurrence of whistleblowing in engineering practice is not known. To run live field-based studies (e.g., ethnographic or interventions) could be looking for a needle in a haystack, an observable whistleblowing situation may not happen in the research time frame. My research cannot control the participants, variables or scenarios, its strength is that it reports decisions, intentions and actual actions taken to report up harmful situations. Each case is unique to the participant, the context of their situation and organisations, however my analysis of these situations is systematic and structured.

5.7.4 Reliability

To evidence the reliability of a method, the study needs to evidence how data collection and analysis can be repeated with the same results, and thus minimise errors and bias in the study. The anonymisation and obfuscation of people required considerable care and effort, to avoid detracting from presenting the story characteristics and narrative. Section 5.3 and 5.5 describe the procedures for running this study, performing the analysis, and populating the case template and story wheels. Reporting on case studies can lead to potentially long documents that would be challenging to repeat by another researcher. My communications and interviews with informants are indeed long, I could not present or share all the data and discussions captured for ethical reasons. There was much more data captured than story narratives used to answer my research questions.

5.8 Summary

Good case studies are hard to do and require a broad set of skills to design and run the studies. Piloting the ORID interviews improved my personal skills. The design and development of the WISE framework and case format went through many iterations. Findings are related to the specific context of the situations under study. The abstraction of actor actions and interactions do allow comparisons (but not generalisations) between

case studies. My presentation of cases is based on a simple case template, with clear well labelled heading sections to guide the reader through the case and evidence. A systematic analysis of narrative extracts identifies key aspects of the case. Infographics (boundary escalation models and story wheels) are used to concisely summarise and present findings. A known limitation is that there can be more than one interpretation of the facts and actions, and that selective presentation and interpretation of facts can influence the direction of findings and conclusions. My research goal is to be curious not judgemental. I focus on the facts of the situation, as reported to me by participants. I seek to triangulate this data where possible. Future research should involve more than one researcher conducting the interviews and analysis.

Chapter 6. Case A (Health)

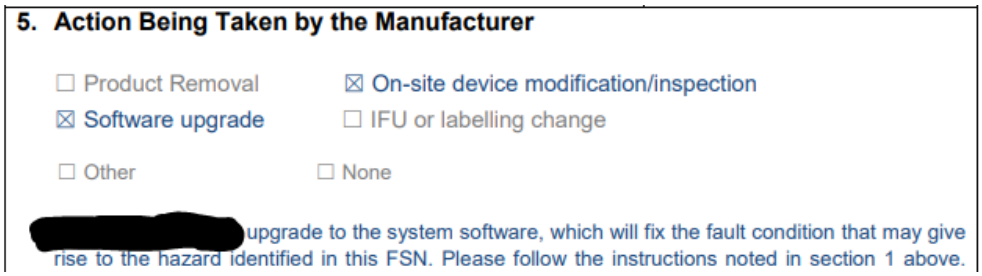
This case is based on interviews, phone calls and emails with Actor A, an experienced software engineer and safety critical systems professional. Actor A discusses a situation from within the last ten years, occurring during the design and development of medical ventilators. There are at least thirty actors identified, both internal and external to the organisation. The ventilator design and safety standard issues raised are reported internally and then externally to relevant professional bodies and regulators in the UK and Europe. The organisation, following advice from Actor A, engages a consultancy to improve their software engineering processes, and subsequently starts using QAC (now Helix QAC[37]), a longstanding best of breed static analysis tool, compliant to internationally recognised coding guides for safety critical systems (Bagnara, Bagnara and Hill, 2018).

6.1 Overview and Triangulation of Data

Case A is about an organisation of less than fifty people. The organisation has a good reputation for developing healthcare devices including ventilators. The organisation trades internationally. In the United States, the Food and Drug Administration[38] (FDA) regulates medical devices. In the UK, the Medicines and Healthcare products Regulatory Agency[39] (MHRA) administers and enforces the law on medical devices and has a range of investigatory and enforcement powers to ensure the safety and quality of devices used with patients. At the time of engagement with Actor A, there were more than sixty companies registered as supplying ventilator machines in the UK.

Actor A did not know if the organisation's ventilators failed during use with patients. The distributor pre-installation tests should detect issues; however, the falsified test specifications could mean issues are missed in the future. The outcome of the referral to the MHRA was not known. We could infer it was not taken to court as Actor A heard nothing further about their report that required them to give evidence. The organisation may have been offered guidance for improvement in their practices. Field safety notices may have been issued to distributors and clinicians regarding the ventilators. Since the incident, publicity relating to the organisation describes the evolution of their ventilators, including updates to hardware, the operating system and software. Urgent Field Safety Notices (FSNs) have been found for their ventilators with details of actions being taken by manufacturers. The dates and full details are not reported here as they would make the organisation identifiable.

[37] https://www.perforce.com/products/helix-qac
[38] https://www.fda.gov/medical-devices
[39] https://www.gov.uk/government/organisations/medicines-and-healthcare-products-regulatory-agency



5. Action Being Taken by the Manufacturer

☐ Product Removal ☒ On-site device modification/inspection

☒ Software upgrade ☐ IFU or labelling change

☐ Other ☐ None

[redacted] upgrade to the system software, which will fix the fault condition that may give rise to the hazard identified in this FSN. Please follow the instructions noted in section 1 above.

Figure 6-1 Field Safety Notice - Actions Taken by Manufacturer.

6.2 Actors

Figure 6-2 illustrates the actors involved in this case story including those cited in the wider context of the incident (e.g., patients, doctors, hospitals) although not directly active in the incident.

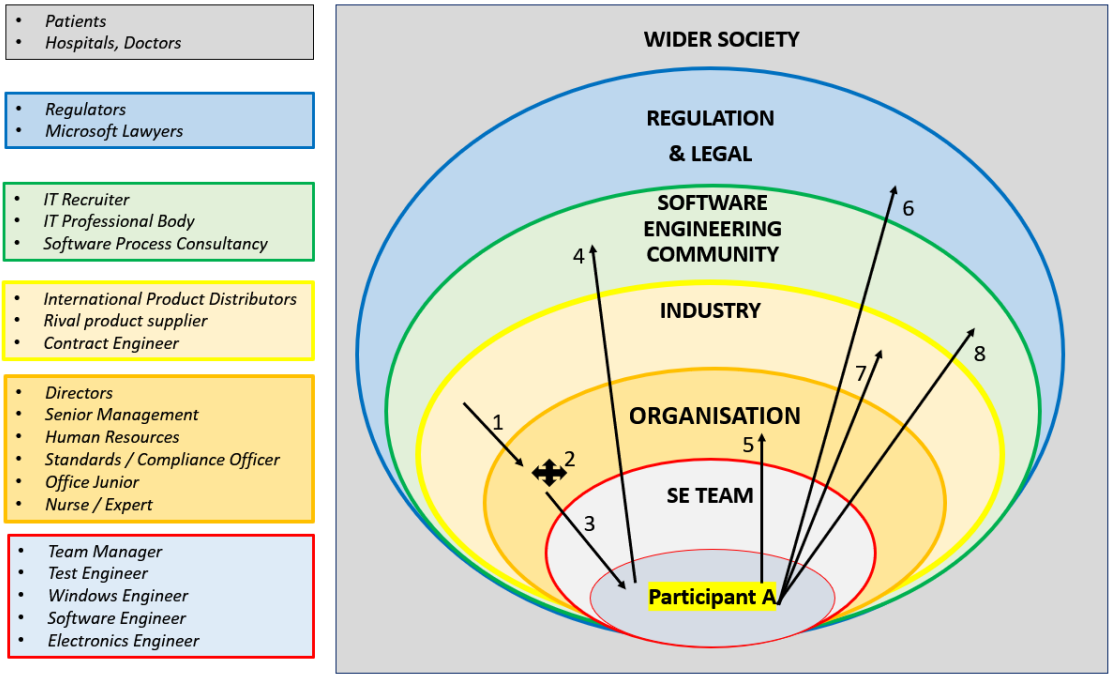


Figure 6-2 Actors and Escalation Steps for Case A

This case touches on all levels of the escalation boundary model. Over a period of several months the number of people aware of the issues grows. By the end of the story most people in the organisation are aware of the issues, as are several external individuals and organisations, including IT consultants, distributors, a professional body, an IT recruiter and two regulators. Hospitals and clinicians (wider society) may have been notified of concerns through the distributors and regulators.

An initial incident involved decisions to use embedded Microsoft Windows 7 Operating System in medical devices based on a comparison to a rival's product. This raised Actor A's concerns regarding safety standards in the organisation's engineering practices. There then follows a main incident, where an international distributor reports ventilators were failing pre-installation tests (Step 1 in Figure 6-2). Internally, test specifications were observed to be rewritten and backdated by the engineering manager at the

organisation (Step 2). An office junior makes Actor A aware of this test report issue (Step 3). Actor A seeks guidance from a professional body (Step 4), warns the organisation (Step 5) and refers the organisation to the regulator over this issue (Step 6) and warns the distributor about what happened (Step 7). An IT consultancy, subsequently engaged to help with software engineering process improvements, also reported the organisation to the regulator. Actor A was dismissed after they raised their concerns about the backdating of test reports. Key actors at the centre of the issues were indirectly reported to have later been dismissed (the Test Engineer) or replaced (the Engineering Manager) following these incidents. Actor A summarised the story as being a combination of inept management, testing failures, ineffectual compliance, and sluggish regulators. After the incident Actor A warned an IT recruiter about the incident at the organisation (Step 8), after being approached regarding a new vacancy at organisation A.

6.3 Public Interest and Wrongdoing

This case covers three whistleblowing harm situations – the initial breach of regulations and then the subsequent covering up of testing failures. Additionally, the potential harm to ventilator users is identified and discussed.

Ventilator-induced lung injury is an acute lung injury inflicted or aggravated by mechanical ventilation and might contribute significantly to the morbidity and mortality of critically ill patients (Beitler, Malhotra and Thompson, 2016). Ventilators are a complex system of electronic sensors and mechanical parts, controlled via a microcontroller, with embedded software and a user interface. Should a ventilator malfunction, it needs to be quickly detected for corrective action for the patient to be taken. The ventilator may need to be reset, restarted or the patient moved on to alternate ventilation. Actor A outlined one of the potential harms that ventilators can cause to already critically ill patients and specifically to neonatal babies.

Actor A.1: *In a ventilator, breathing is a cyclic process - you breathe in, you breathe out, you breathe in, you breathe out. The machine delivers a pressure volume profile over the breath cycle. It's inherently cyclic, and it's inherently time triggered because you set the number of breaths per minute.*

Actor A tells of how a nurse expert explained the harm a ventilator can do to a baby:

Actor A.2: *A nurse who worked for the company said, if you are ventilating a baby you can cause, in a matter of a couple of breaths, fatal volutrauma. Trauma because you are pushing inappropriately large volume of air into the baby. The lungs expand and the blood vessels rupture and the poor baby drowns in his own blood. You can have quite high pressure, be it for short periods of time, but you have to make sure that you monitor the volume. Worst case is that you kill a sick baby in a few breaths of it being hooked up.*

6.3.1 Studies of Critical Incidents with Ventilators

A 2011 clinical study of 3 years of critical incidents concerning anaesthetic equipment (Cassidy, Smith and Arnot-Smith, 2011), found that of 1029 incidents, 338 (32%) were categorised as equipment issues. The most common issue was faulty equipment, which accounted for 113 (33%) of equipment incidents; unfamiliarity or incorrect use of equipment accounted for 72 (21%) incidents. Across all incidents, most (89%) caused no patient harm; however, 30 incidents were judged to have led to moderate or severe harm where a patient required manual ventilation until ventilator self-corrected or for a patient to be moved to alternate equipment. Although faulty equipment was identified, user error or unfamiliarity also played a part. Another clinical study in 2011 (Welters *et al.*, 2011) over a period of 90-months in a 13-bed adult general intensive care unit (ICU), found 1127 critical incidents were reported. The most common incident was faulty equipment, accounting for 33.4% of incidents. Researchers noted that serious equipment failure was reported to the relevant medical regulator for further investigation with suppliers and for notification to others with similar devices. These studies do not report on the technical detail of failures (software, hardware, or mechanical failures). Cassidy et al. conclude, in relation to intermittent and known faults in practice, that "*It is surely indefensible to continue to use equipment when its safe functioning is not guaranteed"*(Cassidy, Smith and Arnot-Smith, 2011).

Alemzadeh et al. studying Food and Drug Administration (FDA) reports in 2013, find that malfunctioning medical devices are one of the leading causes of serious injury and death in the USA (Alemzadeh *et al.*, 2013). Between 2006 and 2011, there were 5,294 recalls and approximately 1.2 million adverse events reported to the FDA. Almost 23 percent of these recalls were due to computer-related failures, of which approximately 94 percent presented medium to high risk of severe health consequences (serious injury or death) to patients. The researchers are computer scientists studying the technical details of events and product recalls from database records. Across their six-year measurement period, the number of computer-related recalls almost doubles. The study findings and discussions emphasise the importance of devices designed with well-defined safety requirements and implementations that employ robust error-detection techniques and rigorously validated fail-safe mechanisms.

Fu et al. (Fu *et al.*, 2018) reviewed medical device recalls from the FDA database over a 3-year period between 2014 and 2016, to identify potential software-related causes for recall. In communication with manufacturers, data was gathered relating to 100 software-related recalls. Subsequent analysis of the gathered data found four main categories of issues: Control Flow Fault, Calculation Fault, System Integration Fault and Human-Machine Interaction Fault. Major challenges were found relating to the reliability, safety, and security of medical devices. At the time of Fu et al.'s study, the FDA database did not contain root causes of software failures in the recall record. The FDA website does now include basic root causes including software design change, manufacturing, and deployment, change control, design, and manufacturing process, and use environment. A recent search of the database reveals that since 2003 the FDA have recalled 429 ventilator devices, of which 52 have software recorded as the root cause (48 items for software design, 2 for change control and 2 for in use environment). Issues detailed include malfunctioning safety mechanisms, screens freezing, issues with software updates and software unexpectedly stopping or restarting, sometimes without audible or visible alerts to nurses and clinicians monitoring patients. Intensive care depends on the stable functioning of technical equipment. These study results show

technical failures lead to events that compromise patient safety and call for a need for development and implementation of strategies for prevention and early detection of errors.

6.3.2 Who Reports Software Issues in Medical Devices?

The clinical studies (Cassidy, Smith and Arnot-Smith, 2011; Welters *et al.*, 2011) take reports of incidents directly from medical clinicians in practice. The technology expert led studies (Alemzadeh *et al.*, 2013; Fu *et al.*, 2018) use reports from USA public regulator databases. The Manufacturer and User Facility Device Experience (MAUDE) database contains medical device reports submitted to the FDA by mandatory reporters (manufacturers, importers, and device user facilities) and voluntary reporters such as health care professionals, patients, and consumers. The MHRA (UK) website does not currently have a database of device recalls, instead it shares links to manufacturers' Field Safety Notices published on webpages or via spreadsheets at weekly and yearly intervals. In the last three years, there have been 30 Field Safety Notices for issues relating to ventilators. I have found no UK research studies of Field Safety Notices comparable to the American studies. I have made no attempt to synthesis data from across the mixed format documents, spreadsheets, and archived websites. Additionally, the UK notices are less descriptive of the source of the incident as shown by examples in Table 6-1.

Table 6-1 Example UK Field Safety Notices - Description of Incidents

*"This voluntary medical device safety notice is being issued following **reports from seventeen customers** stating that during alarm conditions, the audible alarm may not sound and/or the omni-directional LED visual alarm may not illuminate as described in the Operator Manual".*
*"**Within our global vigilance activities, we have become aware of one case** in which an obstructed breathing system was used on a patient during anaesthesia. The patient reportedly became hypoxic and had to be reanimated."*
*"In the **course of our global market surveillance activities**, we have become aware of sporadic cases in which the above-mentioned products restarted…"*

The level of detail regarding source of incidents in field safety notices or recalls varies from report to report and between countries and agencies. The USA databases are more detailed and usable than searching the UK regulator websites, documents, spreadsheets, and archived website files. Specifically, the American MAUDE database includes the source and occupation of the reporter. No studies were found that examine who reports up what issues and incidents to manufacturers and regulators. In this case three stakeholders were known to report the issues to the regulators (Actor A, ventilator distributors and software process consultants). Actor A could have made an anonymous report but chose to make the organisation aware of their reporting actions.

6.4 Software Engineering Aspects

This case covers a broad range of software engineering aspects. Actor A is concerned about risks to patients if a ventilator malfunctions or stops working. Their concerns are increased with an observed lack of adherence to software engineering standards to deliver and test code used in safety critical products. With good software engineering

practices in place, alongside reputable sensors, electronics, and mechanical features of the ventilators, the organisation could argue that a safe and reliable product will be delivered. However, the falsifying of documentation and test reports, indicate weaknesses in development and support processes, alongside the Standard Officer's lack of understanding of software engineering process weakens this argument. This section discusses software engineering issues identified by Actor A and how they tried to address them.

6.4.1 No Concerns with Existing Products

Actor A initially had no major concerns about existing ventilator products and the industrial DOS based operating system it was based on. Nor did they question the technical competence of the electrical and hardware engineers building the existing ventilator products. Actor A recognised that the organisation was, with good justification, looking to make their products cheaper to manufacture and easier to test, and thus looking to re-engineer and improve their products. Actor A did not agree with the proposed technical solution to do so.

Actor A.3: *The version of the product, this flagship ventilator, was actually using an industrial DOS as its main operating software. Which was reasonably common at the time if you had something that needed to show a GUI, a touchscreen GUI and has to handle file access (the ventilation data is recorded). A DOS level of functionality is OK. It doesn't have much in that you don't really want to use. Industrial DOS type operating systems have been used in critical products for some time, so that didn't raise any big flags with me. They [the organisation] recognise that they needed to update the products, make it cheaper to manufacture and test, which is not unreasonable. The bad engineering wasn't driven by money. It was driven by a lack of competence. What they were seeking to do was entirely reasonable, both from a business and technical point of view.*

6.4.2 New Product Issues and Rival Product

Actor A's concerns were raised when it was proposed that the new ventilator product be based on a Windows Embedded operating system, as found in a rival's product. Actor A was particularly concerned given the observed lack of technical competence across the broader product and software development life cycle within the organisation.

Actor A.4: *There arose a situation, in which they bought in a competitor's product. Partly to reverse engineer it, to see what a market-leading ventilator from a rival company does. [Our existing device] had been in the market for a while, and they needed to re-engineer it because they felt that in some ways it wasn't keeping up with the competition. Which I think is why they bought in the [rival] ventilator, which they perceived as the main competition and that they had to be comparable with. As far as I could make out the Engineering Manager wanted to see whether he could get away with using Windows embedded as the operating system for their new ventilator.*

6.4.3 Licensing of Devices (FDA 510(k) Rule)

The rival product was based on a Microsoft operating system and achieved its license through the United States Federal Drug Agency (FDA) 510(k) rule, which states a

device can be approved if it is substantively equivalent to a previously licensed device[40]. The FDA advise [41] that infrastructure (e.g., changing programming language), architecture (e.g., porting to new OS) or core algorithm changes may impact performance and thus a new 501(K) may be required. Actor A did not believe their new product was going to be substantively equivalent or built and tested to the required safety standards of the previously approved device.

Actor A.5: *The rival device was approved by the US Food and Drug Administration, under the 510(k) rule. It is a rule, which enables you to get certification for a device, which is largely similar to an already certified device. If you make incremental changes over a period, you can use 510(k) rule to get an incremental certificate because it is only an incremental change. If you start from a device which was working properly [their existing product] but take several incremental changes [replace the operating system], it can take you quite a long way away from what you know a safety critical systems engineer would actually consider a prudent choice.*

6.4.4 Windows End User License Agreements

The End User License Agreement (EULA) from Microsoft for Windows details its fault tolerance, restrictions of use, warranties, and liabilities. It states that Windows may not be used in any device or system in which a malfunction would result in foreseeable risk of injury or death to any person. An extract of the EULA in Figure 6-3.

15. NOT FAULT TOLERANT. The software is not fault tolerant. *[OEM]* installed the software on the device and is responsible for how it operates on the device.

16. RESTRICTED USE. The Microsoft software was designed for systems that do not require fail-safe performance. You may not use the Microsoft software in any device or system in which a malfunction of the software would result in foreseeable risk of injury or death to any person. This includes operation of nuclear facilities, aircraft navigation or communication systems and air traffic control.

Figure 6-3 Extract of Microsoft EULA for Embedded Windows

Actor A looked into the EULA in relation to safety critical devices and, from their knowledge and experience of safety critical systems, had concerns about engineering in the required redundancy to be safe.

The Engineering Manager told Actor A he took their concerns to a meeting with a Microsoft UK distributor that also had lawyers for advice. The Engineering Manager reported as saying it was okay to proceed. Actor A tries to convince the Engineering Manager that his decision is not in line with safety standard practices and would not meet the FDA 501(k) rule that their rival's device is approved under. Actor A does not know what exactly was explained to the Microsoft lawyers and UK distributors, and if there were early indicators of potential issues with a new ventilator.

[40] https://www.fda.gov/medical-devices/device-approvals-denials-and-clearances/510k-clearances
[41] https://www.fda.gov/regulatory-information/search-fda-guidance-documents/deciding-when-submit-510k-software-change-existing-device

Actor A.6: *I eventually, against his [Engineering Manager] instructions, looked and found a sticker on the inside of the competitor ventilator. It was inside some kind of access panel, and it said it was Windows 7 embedded. That shocked me because I would not regard Windows 7 or any Windows system as suitable for a safety critical product. The first thing I did was to check what the license agreement said.*

The end user license agreement contained a wording to the effect that "the license is not valid if you use this in a setting where harm or damage can be caused, injury or damage can be caused". Now my view with that is well, that doesn't surprise me. Microsoft is selling a desktop operating system for desktop users. It is not (primarily) engineered to be an operating system for safety critical devices. It has many functions in it that you do not need in a ventilator.

The machine delivers a pressure volume profile over the breath cycle. It's inherently cyclic, and it's inherently time triggered because you set the number of breaths per minute. I thought, "Occam's razor". You don't need an operating system like Windows to do that. You can't really engineer the required redundancy into it to be safe. All of my engineering instincts said, no, we do this from the ground up. We use modern safety rated processes.

6.4.5 Software of Unknown Provenance and Standards

Actor A described the Windows operating system as Software of Unknown Provenance (SOUP**).** IEC 62304 standard[42] spells out a risk-based decision model on when it is acceptable to use SOUP in a medical device. It defines standards and testing requirements to support why and how SOUP software should be used.

Actor A.7: *All of my engineering instincts said, no, we do this from the ground up. We use modern safety rated processes. We use all the techniques that are recommended in ISO-IEC 61508, and we do it as a pucker engineering job.*

We don't just buy what in the medical software standards call SOUP - which is an acronym for Software of Unknown Provenance. One of the few acronyms in computing that I think is actually apt and I have used since. Just because you've got a widely used desktop operating system, from a safety critical point of view, you weren't privy to how it was developed and don't have access to the development artefacts, therefore that is software of unknown provenance.

The IEC 61508 standard, referred to by Actor A, covers aspects to be considered when programmable electronic systems are used to carry out safety functions. Products developed on a Windows embedded operating systems would have to ensure that any software malfunction or failure would not cause harm (injury or death) to a patient. In the case of ventilators, visible warnings and audible alarms are the methods used to alert doctors, nurses, or carers to respond quickly and appropriately to the detection of software malfunctions or failure so as to prevent harm to patients.

Actor A was concerned that guidance from appropriate standards was not being followed. However, they do reflect on the challenges with following standards. The

[42] https://www.iso.org/standard/38421.html

organisation did proceed to hire consultants to help them improve their software engineering processes and tools to improve their code quality. Specifically, they bought in QAC (now Helix QAC) a long standing best of breed static analysis tool, compliant to internationally recognised coding standards for safety critical systems[43].

Actor A.8: *They (the organisation) said they comply with the relevant standard and regulations. But the problem is that - and it's a problem with process standards in all industry sectors - recommended practices 61508, for example, recommends practices, or highly recommends practices, that you should use in the software development cycle. But there's a huge problem in most industrial software development cycles – "use that method here", "use that method there". The technical coherence in terms of the output of one sub-process going into another is just not there. There are relatively few examples of being able to use best processes and practice techniques from the beginning to the end of the process. And to tool processes in such a way that the output from one phase is understood by the next phase. It is a chronic weakness.*

There was a guy who was responsible, as a sort of compliance officer. He was mainly concerned with looking at compliance in terms of what the product did, as regards the product standards. He was not a systems engineer himself. And he did not appear to understand the technical coherence problem. And would not have known, for example, if you said look at ISO-IEC 61508, he would say that's not the standard we follow.

6.4.6 Falsifying Test Reports

In this case, the distributor pre-installation checks prevent potentially faulty ventilators being distributed to hospitals. The subsequent falsifying of test reports and the concerns regarding poor software engineering practices calls in to doubt the quality of the product development lifecycle and the future reliability of pre-installation checks to catch such issues with the ventilators. Management chose to cover up the issues rather than investigate and correct them. It was subsequently reported (after Actor A left the organisation) that the Test Engineer had not been carrying out all the necessary tests before shipping the devices. The Engineering Manager and the Test Engineer believed they could cover up information relating to test failures.

Actor A.9: *The Engineering Manager was chairing the meeting, and he said his approach to dealing with this problem was "well OK, if it doesn't pass the test that we have in the specification, what tests will it pass?" Now I have no doubt that the falsified report did pass all the tests that they carried out on it. But whether those were the appropriate tests is open to doubt, whether even the specified tests were actually technically appropriate, I thought it was open to doubt because I wasn't very impressed with their systems engineering.*

6.4.7 Bringing in Experts and Static Analysis Tools

Actor A observes poor software engineering practices, including technical decisions made by the manager, poor adherence to standards, weak bug tracking and the lack of impact analysis.

[43] https://www.perforce.com/products/helix-qac

Actor A.10: *They did look at them (failures, impacts and frequency), but I didn't see any mechanism for feeding that back into the engineering process. It was only on a problem list. There was one guy there was using Bugzilla to track reports. It records the fault, but there wasn't a process where you looked at impact analysis. What else could that fault cause? What could be causing that fault? Could it affect other things? It wasn't systematic, it was just clerical. They did minimal level of tracking in their systems, engineering procedures and software engineering procedure. It was very weak, technically.*

Actor A reflected on how many people in the organisation were technical and competent professionals should have also raised issues:

Actor A.11: *Eight, I would have expected to (be technically competent), if they were competent, to have picked up on it and say hang on this isn't right.*

Actor A had some success making improvements to the organisation, advising the bringing in of consultants and new static code testing tools. Actor A makes the incoming process consultants aware of the test report situations. The incoming consultants are later also reported to have raised concerns about the organisation with the MHRA, although no additional details were given. The history of the organisation and how others had previously attempted to improve group practices is not known.

Actor A.12: *One of the things I recommended was that they get in a code auditing tool called QAC, which is very good. One of the best of breed C static analysers on the market. Especially for adherence to coding standards.*
It was evident to me, in that company, that nobody had the depth of understanding of systems engineering in general, and software engineering in particular that could understand that they didn't know what they didn't know. I had said look you [they] want to get some consultancy advice on how good the process is - and I persuaded them to hire in a quite large consultancy. They did that and were just on the point of hiring them when I walked out when they fired me.

6.5 Actor Actions

Case A reports on a broad range of actions taken by Actor A and others inside and outside the organisation.

6.5.1 Raising Concerns about the Engineering Manager

Actor A initially described the organisations as a "*typical small enterprise, small niche enterprise doing a particular kind of product. Quite a good reputation*." Quite soon after joining, the development team raised concerns about the competency of the Engineering Manager to Actor A. Anecdotally Actor A reported that an engineer in the team left the organisation because of the competency of the Engineering Manager.

6.5.2 Investigating Issues and Asking for Advice

Actor A investigates the rival's ventilator product and in particular the End User License Agreement and safety standards that new medical devices using embedded Microsoft win7OS must comply with (see Software Engineering Aspects in Section 6.4 for details). With concerns identified, they escalate their concerns to the Engineering Manager.

Actor A.13: *Yes, I did ring up an IT professional body…I spoke to somebody in the in the safety critical systems side of things. They said make sure that you publicise concerns to the company. So, I copied in a lot of people. And the firm held that against me - it clearly embarrassed the Engineering Manager – "why did you have to send it to all those people". Well, the reason was to make it quite clear that I'm bleating about this. This was just on the Windows issue - which I emailed all the directors about. I discharged my responsibility there.*

The Engineering Manager reported back as saying it was okay to proceed. Actor A again tries to convince the Engineering Manager that his decision is not in line with current safety standard practices and would likely not meet the FDA 501(k) rule that their rival's device is approved under. Actor A discusses the situation with an IT professional body and was given the advice to publicise their concerns within the company.

6.5.3 Engineering Manager's Justification

Actor A recognised that the organisation was, with good justification, looking to make their products cheaper to manufacture and easier to test, and thus looking to re-engineer and improve their products. But they did not agree with their proposed solution. The Engineering Manager believed the decision to move to the Windows platform was acceptable and instructed the team to implement the changes. Actor A was concerned that the Engineering Manager had not reported the true facts of the situation to the either the distributors or Microsoft legal representatives. Actor A was not entirely sure that the Engineering Manager did have a meeting with Distributors or Microsoft as Actor A was subsequently "*kept out of the loop*" on this work.

6.5.4 Distributor Complaints

Approximately five months after the Windows incident, an international distributor raises a complaint to the organisation about a device failing pre-installations checks. Through open-source checks with regulator and distributor websites I am aware of more recent issues with ventilators (Figure 6-1 Field Safety Notice - Actions Taken by Manufacturer.)

Actor A.14*: About four and a half months later I was told by an office junior that he [Engineering Manager] was holding a meeting… One of our distributors in […] is complaining that when they unpacked the products, they're not meeting their basic out of the box pre-installation checks - they are failing their tests. I said, "don't we test them before they go?" He said yes. I said what's the problem then? The Engineering Manager was chairing the meeting, and he said his approach to dealing with this problem was "well OK, if it doesn't pass the test that we have in the specification, what tests will it*

Actor A.14 cont. *pass?" He then rewrote the test report on it and backdated it by about six months. And that's fraud, plain and simple.*

Actor A is first made aware of the issue by an Office Junior, reporting that the Engineering Manager held a meeting about the pre-installation checks and was adjusting and back-date test reports. We do not know if the Office Junior raised their concerns in the meeting or with anyone else in the organisation. We do not know when other staff inside the organisation first became aware of the situation formally, through the test report failures from the distributor, or informally, from other internal tests, discussions, or emails. The actions of Actor A and their subsequent dismissal did make most of the company aware of the situation and issue. Actor A did not make the organisation aware of how they found out about the distributor report.

6.5.5 Evidence Gathering

Actor A decides they must gather evidence regarding the falsified test report and obtained both a digital and hard copy version of the signed off backdated report. They replaced the hardcopy report (taken from the Engineering Manager's office) with blank paper so not be accused of stealing paper from the company while seizing the evidence. Actor A reported doing everything by the book throughout the reporting process to limit any retaliation by the organisation.

6.5.6 Escalate to MHRA

Actor A speaks directly to the company Standards (compliance) Officer, who appeared to be aware of the situation and was unhappy that Actor A was going to hand the falsified test report to the regulator. Actor A then reports the issue directly to the MHRA (both by email and then in person to their offices) and complains the UK regulator was sluggish in their reaction. Actor A also contacted the international distributor, who in turn contacted the international regulator who is more responsive. Directors and other employees (including Standards Officer) are formally notified, by email, that the issue has been reported to the MHRA. Actor A was open about their reporting of the situation and was not concerned about their anonymity. They were concerned for other staff and did not want to "*drop them in it*".

Actor A.15*: And eventually, when the meeting (which had taken place over two days), was finished - and this effectively fraudulent report was printed, I actually went into the Engineering Manager's office put it in a plastic bag, took it out to the car park and put it in a grit bin in next to my car. I thought I'm seizing the evidence on this. I seized the evidence on a Tuesday. I rang up the standards officer in the company, who was singularly unimpressed, and told him what the situation was. His first thing was will you please give me the report. And I think hello, I think he's been a part of this, he knows he's been caught. I said no, I'm going to give it to the MHRA… who took about a fortnight to acknowledge it. I think that was in November. Eventually I went to the MHRA up in Victoria in, I think the middle of December, or maybe the week before Christmas and actually handed the report over to them and told them what had happened.*

6.5.7 Retaliation

Within a week of reporting the organisation to the MHRA, Actor A was called into a meeting with HR and was dismissed. Actor A was relatively new to the organisation when the issues were discovered and raised. They acknowledge they are outspoken and had expected the retaliation. Actor A did not want to stay working at the organisation or take them to an employment tribunal. They felt they had to speak up because it went against their professional principles not to do so. The organisation had the opportunity to take corrective actions regarding the testing of ventilators but chose not to. Actor A is a mature and experienced professional, confident in their evaluation of the situation that he issues needed to be raised and appropriate standards and processes should have been adhered to. It is not known if management had previously dismissed employees for speaking out.

Actor A.16*: I expected a little of the brown stuff to hit the fan. When that happens, it ruffled a few feathers. But I think by that time, after the Windows Licensing debacle, I thought I might not spend too much time here. The Wednesday passed without incident. The Thursday, as I recall, they decided, at short notice, to do my probationary interview. …I said I will be fired by the end of the week, if not by the end of the day. I could see what the writing on the wall was. And all they wanted to know was what I knew about this [test]report.*
I said I thought this was a probationary interview, but they turned it round to this issue. I just turned around the Engineering Manager, I said not only are you utterly and abjectly incompetent, but you are also dishonest too. That basically, forced them to fire me, which didn't bother me. It's a way of walking out and they must pay you a month salary in lieu.

Once Actor A was dismissed the organisation instructed other employees not to talk with them. The Test Engineer was subsequently reported to have been dismissed and the Engineering Manager replaced, we don't know details of the investigation that lead to these actions and how directly they related to the issues Actor A raised with the MHRA.

Actor A.17: *The firm had told everybody not to speak to me. This is routine. This happens. If you leave, the people are told do not speak to you. The only guy who did speak to me was an electronic engineer who was there on a contract basis. He thought "he was contractor, so what if they fire me". He wasn't going to be told who he couldn't talk to. And it was through him I learned that an engineer whose responsibility was to test machines before they went off to the distributors, was actually fired for not doing so.*
In particular, they found that the guy who was responsible for performing the last test before they went off to the distributors, was actually just signing the forms and not doing the tests. So, it's a chronic management failure. But you deal with that by rooting out where it's going wrong in the process. You don't deal with it by falsifying a test report.

6.5.8 Fear of Speaking Up

Actor A had concerns regarding the organisational management and treatment of staff. When they first started, other staff informally warned them about the competency of the

Engineering Manager. Later they observed the Windows Engineer carried out the Engineering Manager's request to implement Windows 7 in the new ventilator, justifying it because he could not afford to lose his job if he spoke up. Actor A did say they did not have the same worries, they treated the work as a short-term contract and was happy to escalate the concerns and not be bound by being an employee when it comes to ethical conduct.

Actor A.18*: That engineer had gone through a very costly divorce and was in an awkward situation financially, so I didn't land him in it. He wasn't in a position, he felt, where he could challenge him (the Engineering Manager), because he couldn't afford to lose the job. Although I was technically an employee on the payroll, I was perfectly happy to regard it as a short, not very well-paid contract. Although I was an employee and nominally having to do what I was told, I regarded myself as bound by the [Professional Body] code of conduct, employment status or not. You do not buy my ethical conscience simply by making me an employee. And if you think you can intimidate me about such matters, you're wrong.*

Actor A felt professionally responsible, they had the skills and experience of working in safety critical systems for many years to be confident to speak up. They acknowledge they are outspoken, and in this organisation did not trust senior management.

Actor A.19: *People that I've worked will say, not only do I not suffer fools gladly. Once I've decided that somebody is a fool, I tend to suffer them with a spectacular lack of grace.*
(In response to falsifying test reports) I sent an email to everybody who'd been at that meeting. That said, "it's widely known that I regard the Engineering Manager as incompetent. At my last meeting with him, I told him I considered him dishonest as well. We know we've been having trouble with ventilators not meeting their specifications. The way you deal with this is not to hold a meeting, find out what test it will pass, and backdate the report. That's fraudulent."

6.5.9 Protecting Others

Actor A responded to the falsifying reports situation without involving the office junior, who had informally reported the situation to them. Actor A took it upon themselves to get a physical copy of the falsified report (and replace it with blank paper so not to be accused of stealing themselves) without involving the office junior.

6.6 Outcomes

6.6.1 Regulator Referral

The final outcomes were the organisation being referred to the regulator (three times, once by Actor A, then by a consultancy engaged to help with process improvements and also a European distributor). Actor A reported that the UK regulator was somewhat sluggish to respond compared to international regulators.

6.6.2 Dismissal and Replacement of Staff

Actor A was dismissed as a result of speaking up. Other key actors at the centre of the issues were reported to have been dismissed (Test Engineer) and replaced (Engineering Manager). When asked about taking the organisation to an Employment Tribunal over their dismissal, Actor A said they would not do so.

Actor A.20: *There's no point in going to an employment tribunal unless you are particularly aggrieved, and wanted to stay in the job, or were particularly aggrieved that you were booted out. I'm not going to complain about losing a job because crooks fired me. I'm just going to leave, just walk away from them. Walking away and staying absolutely silent would not have been ethical in my point of view, but walking away and telling the regulator and other firms who need to deal with them I think is reasonable.*

6.6.3 Future Career and Warning Recruitment Agency

Actor A was open about their reasons for leaving this organisation – both with future employers and IT recruitment agencies. When an IT recruiter contacted Actor A about another role at the same organisation, Actor A advised caution to placing other people there. Actor A reports they spoke about issues on the ventilator project with a recruitment agency:

Actor A.21: *Some agencies contacted me, because I had this firm on my CV, and said that they're recruiting again. I said please don't send me there, they fired me and I told them why. I said I would be very careful in advising your contractors about this. You may be putting a contractor into a position where, through no fault of yours or his own, he's put in a position where he feels he's asked to do unethical things.*

At a subsequent job interview, Actor A felt they were offered the role, in part because of their record on speaking up.

Actor A.22: *...And they [new organisation interviewing] said, well, tell us about yourself. I said, let me tell you about my last project. They fired me, and I told them what happened. When I joined the company, they said they said one thing we'd like you to do - you must familiarise yourself with the company ethics policy - they actually maintain a whistleblowing function in their own company for ethical purposes. Because I blew the whistle, actually was one of the things that help me get the job. Things come around, but I think as far as I'm aware, such companies are in a minority.*

6.7 Case Summary

Actor A provided a rich and in-depth view of the technical, organisational, and human aspects of the story. There are a broad range of individuals, teams and organisations identified both internally and externally to the ventilator company that observed or were made aware of the falsified test specifications. The case evidences actions of people keeping quiet, covering up issues, speaking up, complying with management, or exiting the organisation because of factors relating to the incident. Actor A explains their actions are motivated by their professional responsibility, to speak up about poor

product management, poor organisation culture while also having an awareness of protection of self and others involved in reporting up the situation.

Chapter 7. Case B (Transport)

This case is based on interviews, phone calls and emails with Actor B about their experiences during the development phase of a vehicle Engine Management System. The wrongdoing was not raised outside of the development team, though the team did write comments into the code and documentation relating to the harmful code. This was Actor B's first job after finishing university. They subsequently went on to work on vehicle security systems, methods of software assurance, static analysis of software, and safety critical systems.

7.1 Overview and Triangulation of Data

Case B is about a reputable vehicle manufacturing group based in Europe, shipping vehicles internationally. Their vehicles contain a software-based Engine Management System that monitors and controls how the engine performs (power, fuel economy and emissions). On behalf of the UK government, the Vehicle Certification Agency (VCA) is the designated Type Approval authority for automotive products, this includes emission (homologation) tests. Vehicle Type approval is the UK system by which manufacturers prove an example of their vehicle meets all the required safety and regulatory standards to the VCA. Vehicles cannot legally be sold without this certification. Certification is applied to all future examples of that vehicle with the same specification.

The events in the story were triggered when an algorithm was requested to adjust the engine's performance and emissions while regulatory tests were being run on the vehicle. The requirement came from a senior engineer outside of the software engineering team and was endorsed by the software engineering team manager. The software engineering team, made up of four developers, initially refused to implement the requirement, but eventually complied with the request. The vehicle received certificates for meeting performance standards and was sold in the UK. The company still manufactures engines and vehicles today, although few examples of this particular model are likely to be on the road. This case situation is prior to the Volkswagen scandal. Case B demonstrates that prior to the 2015 Volkswagen scandal at least one vehicle manufacturer in the UK successfully evaded regulatory standards for emission tests through the misuse of software.

7.2 Actors

Figure 7-1 illustrates key actors in the incident and those cited in the wider context of the incident. It is a small self-contained story with no escalation of the issues outside of

the software engineering team. Vehicle owners and regulators are unaware that they were misled by the organisation. It is not clear if senior management or shareholders were aware of the misuse of software and wrongdoing.

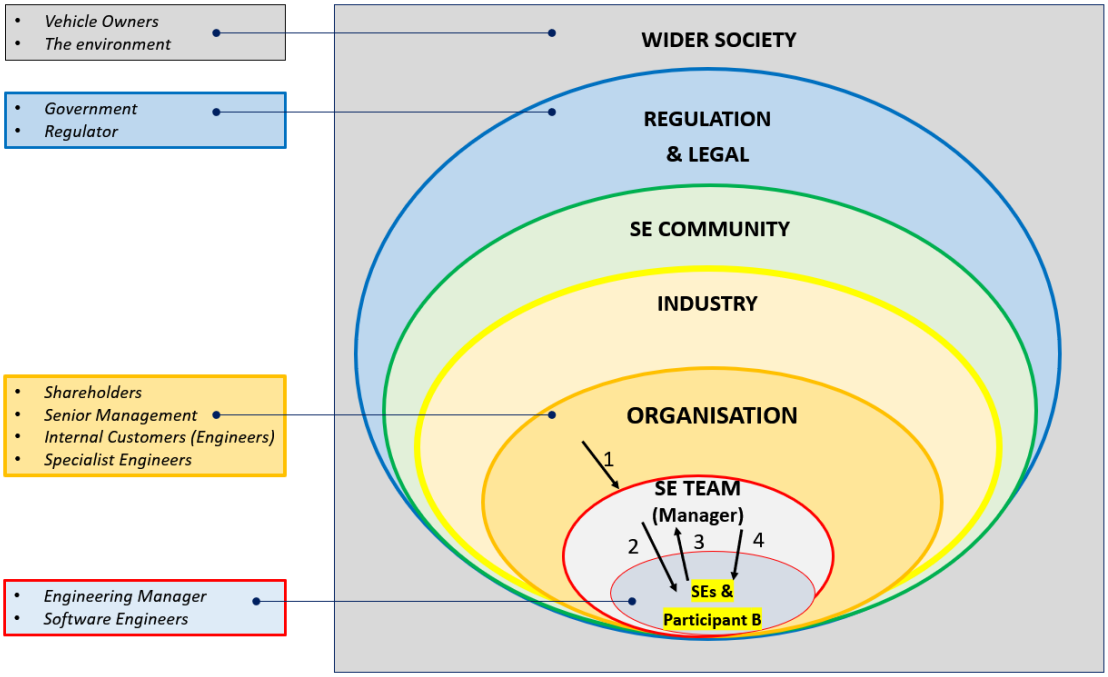


Figure 7-1 Actors and Escalation Steps for Case B

The software engineering team implemented the test detection feature as instructed by specialist engineers (step 1). There were discussions between the developers and team manager not to implement the code (steps 2, 3 and 4), but no escalations of the issue went outside of the team, the team implemented the code due to threats made about their jobs if they did not. It is not known how widely the emission detection software was known about outside of the team (e.g., a tester or compliance officer?) or within the automotive or software industry more generally. Actor B reported that the vehicle went into production, if the issue was raised, it was not declared or corrected while Actor B worked there.

7.3 Public Interest and Wrongdoing

Case B covers three whistleblowing situations. The initial breach of regulations, the subsequent covering up of code design, and the potential harm to people and the environment.

In 1978 General Motors introduced the first electronic control systems into vehicles. Since then, the Electronic Control Unit (ECU) has evolved from controlling the internal combustion engine into a complex Engine Management Systems (EMS) controlling many aspects of modern vehicles. Most non-electric vehicles on the road have internal combustion engines. The EMS takes input from the driver (e.g., brakes, pedals, gears, and cruise control settings) alongside other monitoring and control subsystems in the vehicle (e.g., diagnostics and engine sensors). The EMS software then controls the valves, fuel injection and spark ignition through parameter settings to operate a well-tuned engine delivering power, fuel economy while minimising emissions.

Public Interest: Emissions from combustion engines are a source of pollutants in the form of nitrogen oxides (NOx) and are a significant contributor to ozone and fine particle pollution. The World Health Organisation has reported there are more than 70,000 scientific papers to demonstrate that air pollution is affecting our health. The inhalation of fine particles can damage lung tissue and cause or worsen respiratory conditions such as asthma, emphysema, and bronchitis. It can aggravate existing heart disease. Global regulations are in place to reduce pollution through NOx emissions and thus reduce health risks. It is these regulations and emission tests that Volkswagen evaded. With the testing protocols published, engineers could build software to detect when vehicles were running under lab conditions. In lab conditions the emission control systems were used to reduce emissions to acceptable government set levels. In real driving conditions, emission reduction controls were bypassed thus boosting performance and fuel efficiency rating of a vehicle. The driver gets improved performance, the emissions tests and government appear to meet their emission targets. The organisation in Case B is not Volkswagen, though there are many similarities between the two as described by Actor B during the interviews. The next section provides background to emission test evasion.

7.3.1 Background to Emission Test Evasion

In 2014, the California Air Resources Board (CARB) commissioned a study on emissions discrepancies between European and US models of vehicles, with scientists at West Virginia University specifically looked to detect emissions during live road tests, compared to testbeds. In 2015 CARB reported that software was being misused by Volkswagen, allowing NOx emissions up to 35 times higher than permitted by the US standard. Volkswagen had chosen to cheat the regulators for over 7 years, rather than solve the lowering emissions problem.

In 2016 a UK government policy paper published findings into whether vehicle manufacturers were manipulating emissions tests in the UK[44]. They did not detect evidence of test cycle manipulation strategies as used by the Volkswagen Group. However, tests found higher levels of NO_x emissions in test track and real-world driving conditions than those in the laboratory for all manufacturers' vehicles, with results varying significantly between different makes and models. Existing laboratory tests, designed to ensure emission limits, were shown to be inadequate. Since 2017, UK vehicles have had to meet emissions limits in real driving conditions and across a wide range of operating temperatures. Car firms lobbied to delay the implementation of new limits and tests[45], citing the high cost of meeting stricter environmental controls. Additionally, roadside checks[46] now include looking for emissions "*cheat devices*" used by drivers to cut the cost of operating a vehicle. Regulators have found evidence of:

- devices designed to stop emissions control systems working
- cheap, fake emission reduction devices
- illegal engine modifications which result in excessive emissions

[44] https://www.gov.uk/government/consultations/improving-air-quality-reducing-nitrogen-dioxide-in-our-towns-and-cities
[45] https://www.ft.com, carmaker lobby EU
[46] https://www.gov.uk/government/news/lorry-emissions-checks-to-start-at-the-roadside

In 2015, after a year-long investigation, an international team of investigators identified the Volkswagen defeat device as a piece of code labelled "*acoustic condition*" which activated emissions-curbing systems when the car's computer identified it was undergoing a test (Contag *et al.*, 2017). "*The Volkswagen defeat device is arguably the most complex in automotive history*" Levchenko's team examined 900 versions of the engine control unit's code and found that 400 included information to circumvent emissions tests. The "*acoustic condition*" label indicates code to control the sound the engine makes. The label was a euphemism for conditions occurring during an emissions test. Researchers found a less sophisticated technique in the Fiat 500X, where the car's on-board computer allowed the emissions-curbing system to run for the first 26 minutes and 40 seconds after the engine starts— roughly the duration of many emissions tests.

Case B has several similarities to the Volkswagen and Fiat mechanisms for defeating emission tests. Researchers note that for both Volkswagen and Fiat, the vehicle Engine Control Units are manufactured by automotive component maker Bosch. Bosch was recorded as having warned Volkswagen not to enable the override feature of the ECU in production vehicles (Trope and Ressler, 2016). Bosch was given a ninety million euro fine for their involvement in the Volkswagen scandal[47].

7.3.2 Disclosure of Issues at Volkswagen

When the Volkswagen story first broke, their board of directors claimed to have no knowledge of the malpractice and blamed "*a few rogue engineers*". Following public inquiries in the USA, two senior managers were identified at the centre of the design and implementation of the defeat device and software. The public inquiry reported that some engineers did raise concerns in 2006, but managers instructed them to continue with it, and so the use of the defeat device became routine. Several car breakdowns were blamed on the vehicles remaining in 'test' mode while being driven on the road. Engineers worked to solve the breakdowns and were encouraged to conceal the rogue software and to destroy the related documents[48]. Six Volkswagen executives and managers were eventually charged in the USA over their role in the scandal. James Liang, a senior Volkswagen engineer pleaded guilty and was jailed for more than three years[49]. He was released from jail in December 2019. At Volkswagen a new whistleblowing system was introduced to eliminate the "*culture of silence*" that helped emissions cheating devices and software remain unreported for many years. In Case B the issues have not knowingly been disclosed publicly.

7.4 Software and Engineering Aspects

7.4.1 Quality of Code and Software Engineering practices

The code in Case B is not known to be of poor quality, it delivered the requirement that went into production vehicles. The software engineering practices, and team's technical skills are also not known to be a factor in the case.

[47] https://www.dw.com/en/bosch-pays-90-million-euro-fine-over-diesel-scandal/a-48843405
[48] https://www.bbc.co.uk/news/business-38603723
[49] https://www.bbc.co.uk/news/business-41053740

7.4.2 Technical Specification (Agreements) and Testing

The technical specification of the vehicle is provided to vehicle owners and regulators, with assurance that the specification, including emissions standards, have been tested and met. Unless emission tests were run on live road conditions, these tests would not identify any issues with the specification.

7.4.3 Unethical Code and Hiding Evidence

The software is not known to be of poor quality but is known to be misused to defeat an emissions test. The software engineering team evidenced the source of the requirement in the published design drawings but were not allowed to explicitly name it as a "*test detection*". Their manager compromised the extent to which the documentation referred to test detection, thus covering up the issue from people that might read the documentation but not have access to the source code. The dialogue between the team and their manager indicates a recognition of what was being coded was unethical. The team did leave evidence of "*defeat emission test*" labels in the source code.

Actor B.1: *It didn't sit well with any of the [*software*] engineers in the department. There were four of us on the team, who were responsible for doing that particular engine management system. All four of us flatly refused to do it. Our boss, the managing engineer, if you like, he was an engineer by background, but he actually directed that if we didn't do it we would be out of a job. So, we did the code.*

We used a very descriptive flag name. It was called "emissions test detection." We didn't hide it in any way, shape, or form. We tried not to hide it. And when the software design document, the drawings capturing the design of the software were reviewed by management - they were not happy with this flag name. So, we actually gave it the name of the engineer who directed that the emissions test should be detected. We called it "[Joe Bloggs] mode." That was as far as we could get away with.

...the unhappiness with the name was from the Engineering Team Manager – the specialist manager never realised his name would be going into the documentation! The naming of the software variables was completely internal to the engine management systems group.

7.4.4 Code to Detect Emission Tests

The protocol for emissions test is published and specialist engineers knew they could detect, with a fair degree of confidence, when a car was in an emissions test condition.

Actor B.2: *There were several internal customers who were experts in the internal combustion engine who we worked with to refine the control system. Basically, the requirement came from outside the software team, but was endorsed by the software team manager.*
The emissions tests were part of the type approval or homologation. That's what Volkswagen was caught with. It's quite easy to do because you know that the vehicle is going to be kept at a standard temperature for about 24 hours before it goes into the emissions test. So, if the vehicle has been at 25 degrees plus or minus so many degrees, for 24 hours, and then sits idling for one minute before anybody touches anything, then you've got a fair degree of confidence to back off the fuel and pass the emission test.

7.5 Actor Actions

7.5.1 Organisation Justification

Actor B described the harm of cheating the emissions tests and related it to Volkswagen's recent case of cheating emission test and environmental impact. Actor B did reflect that the green argument is stronger now than it was at time of this incident. They report that the organisation justified the requirement as being driven by a belief that customers wanted performance, while letting the government believe fuel economy standards were met.

Actor B.3: *It's quite pertinent in the light of the Volkswagen fiasco of a few years ago, when they were discovered fiddling emissions tests. We were directed, by management, to detect an emissions test and change the fuel parameters on an engine management system once we have detected that emissions test.*
The justification given for us by the engineer who proposed it was "Well, it means that we can give the customers the driveability they want and let the government get their satisfaction that we've got the right fuel economy." At the time, the green argument was not quite so strong as it is today.

7.5.2 Fear of Speaking Up, Protecting Self (and Team)

The software engineering team were not comfortable implementing the test detection feature, they discussed it and knew it was unethical. The team raised their concerns with their manager and initially refused to implement the feature. Their manager threatened them with job loss if they did not implement it. Actor B felt there was no real regulator they could have gone to, and they could not raise their concerns anonymously. They mitigated harm to themselves by not speaking up. The team also protected themselves by indicating who the requirement came from in the documentation. The team published design documentation referring to *"<Specialist Engineer Name Mode>*" as a label for the test detection feature. Software variables, internal to the engine management system, specifically referred to "*emissions test detection*." The Specialist Engineer was not aware that his name was referenced in the documentation. Stakeholders outside of the organisation would not have been aware of the code or the documentation.

Actor B.4: *the "Engineering Manager" within the "Engineering Department" was the person who insisted that the algorithm be included, however it was a manager from an internal customer who had proposed the algorithm. There were several internal customers who were experts in the internal combustion engine who we worked with to refine the control system. Basically, the requirement came from outside the software team, but was endorsed by the software team manager.*

Actor B reflects that the incident was early in their career, and that they would not be as compliant now.

Actor B.5*: I think if I could go back and have my time again, would I have been as compliant as I was? Hell no. But as somebody who's 18 months out of college, just married and in his first job? I didn't know what I was doing.*

7.5.3 What Vehicle Owners Want

Actor B reflects on the role of customers, suggesting that vehicle owners would not have paid for environmental protection or economy options on the vehicle if given a choice (at the time). UK government report indicates drivers use cheating devices to improve performance rather than control emissions.

Actor B.6: *When I was first at the manufacturing group, they offered the customer the chance of putting on run flat tyres or a stereo system. So, give the customer no cost safety or no cost luxury. They didn't get rid of many run flat tyres. People opted for the radio system. Customers don't want ecology or economy - until they need it. If you can give them bells and whistles and make them more comfortable, the customer will take it every time. Don't expect them to pay for safety. Don't expect them to pay for economy. They're looking for fluff. The psychology goes against people trying to make things better from a safety and economic benefit.*

7.6 Outcomes

7.6.1 Short Term Profit

Actor B reflected on the tension of organisational short-term profits over the longer (green) sustainability.

Actor B.7: *I think we put way too much store on short term profitability. We've got to get away from that short term profitability, and into long term sustainable mentality - for all parties, not just a shareholder.*
Engineering requires better educated customers and it requires better educated managers. Project managers who are willing to look from the customer perspective, the user perspective, not just the shareholder perspective.

7.6.2 Lasting Impact

The emission detection software went into production vehicles, with certificates indicating they meet specified performance standards. Regulatory tests were subverted. Regulators and future owners were misled. Actor B reflects how this incident affected their future career, helping them better reflect on non-technical challenges faced by engineers.

Actor B.8: *while it represents a deliberate attempt to subvert a regulatory test, it actually formed one of my ethical foundations for my subsequent career. Being early in my career, it was perhaps my first major ethical challenge and illustrated the importance of engineers being able to take a wider perspective than just meeting a customer's technical requirements. When I subsequently became involved directly in safety, my earlier exposure to an ethical challenge served me well and allowed me to better understand the non-technical nature of some challenges faced by engineers.*

Actor B reported it was cathartic to revisit stories from their past and reflect on how it influenced their future career.

7.7 Case Summary

In Case B there are a broad range of actors from the software engineers and their management through to vehicle owners and wider society impacted by cover ups and environmental pollution. Vehicles from organisation B were sold in the UK with certificates that misled regulators and vehicle owners. Case B evidences software engineers documenting issues and trying to raise their concerns about the software. Case B demonstrates an organisation complicit in disguising unethical code and suppressing concerns. Ultimately the engineers followed instructions from senior management and kept quiet about the situation to protect their career.

Chapter 8. Case C (Nuclear)

The case is based on interviews, phone calls and emails with Actor C, a software engineering expert and professional body representative. The events in the story were triggered when a professional body was made aware of issues relating to the design of a computer-based protection system for a proposed nuclear reactor. While the original incident is over 25 years old, it relates to the use of technical standards, formal methods and verification of systems that are relevant today with the evolving of new technologies in safety critical systems. New sets of risks and concerns from errors, malfunctioning, cyber-attacks, and safety need to be understood and appropriate methods and standards used to regulate those building and operating such software-based systems.

8.1 Overview and Triangulation of Data

Case C is based around the time when the approval for a nuclear power station at Hinkley Point in Somerset was being sought. A 1988 inquiry was chaired by Michael Barnes, QC, reported on issues including the *"safety and the impact on health of the proposed pressurised water reactor*." Many national and local bodies submitted evidence arguing for and against the development. While the plans to build Hinkley Point C were eventually approved, they were dropped by subsequent governments.

The events in the story were triggered when a professional body was made aware of issues relating to the design of a computer-based protection system for a proposed nuclear reactor. Concerns were raised by the organisation's internal quality assurance team that the protection system would not meet the regulatory requirements. A report containing evidence of not meeting standards was suppressed from the nuclear regulator. Actor C followed the planning proposal inquiry, with a particular interest on how it reported on the safety of computer-based protection systems. Actor C gave evidence at a regional inquiry day and published an article in a broadsheet newspaper at the time.

8.2 Actors

Figure 8-1 illustrates the actors in Case C story. The case spans all levels of this escalation boundary model from those that build the system all the way through to wider society, raising their concerns of the impact of nuclear reactors and any failures on the local community and wider environment.

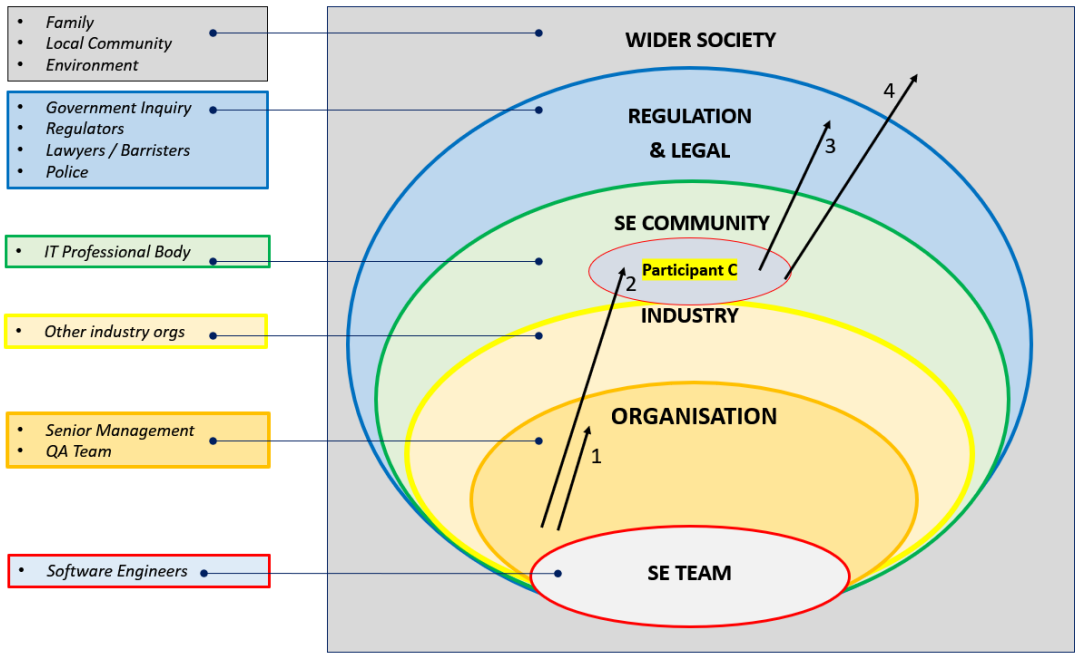


Figure 8-1 Actors and Escalation Steps for Case C

There is little known about what happened inside the organisation. It is known that the Organisation's QA Team (verification and validation) produced a report that was suppressed by the organisation management (Step 1). There is nothing else known about what else happened internally at the organisation. Did the engineers and software engineers believe the system was acceptable, did they attempt to raise concerns if not? Did anyone else inside the organisation raise informally or formally the report findings or its suppression? Actor C receives a report from inside the organisation of concerns about suppression of report that regulatory requirements would not be met (Step 2). Actor C, through their role with a professional body, takes their public safety concerns to a public inquiry (Step 3) and writes about concerns more generally in the media (Step 4). Following the public inquiry, Actor C experienced retaliation events, with threats to their career in the nuclear industry and a house break in, believed to be linked to people trying to find physical copies of the suppressed QA Team report. Concerns raised by the professional body in turn changed how quality assurance of software in the safety critical industry was done and led to new static analysis tools were developed (Jones, 1991).

8.3 Public Interest and Wrongdoing

This case covers all four whistleblowing harm situations – the initial breach of regulations and subsequent covering up of breach. Additionally, the potential harm to people and the environment is identified. Actor C described how they and the professional body were concerned about safety critical software and their responsibility to uphold standards when such systems are being developed. An example of proximity being an issue, and not wanting a Bhopal scale accident to happen in a populated area.

Actor C.1: *One of the things that I was concerned about was the vulnerability of the control systems for the major hazard sites in the country. Oil refineries, chemical plants and big fuel stores, places where if there's an accident, because there are quite often these are in in places where there are people around or if the wind is blowing, lots of people get hurt. You don't want a Bhopal scale accident if you can possibly avoid it.*

Actor C recalled attending conferences and being involved in early discussions about the risk assessment of safety critical software and how it was sometimes wrongly compared to hardware risks and failure modes.

Actor C.2: *At that time there was a growing interest in safety critical software. There were international conferences…There was lots of talk about probabilistic risk assessment of hardware, fault trees and failure modes, and effect analysis on hardware. When it came to software, assumptions were that the same mechanisms worked. People produced arguments about the individual probability of failure of an assignment statement. How often do assignment statements fail?! How often do do-loops fail? For an off the shelf transistor or resistor, you might be able to get that kind of statistic, you can't for an individual programming language construct.*

Actor C also recounted the beginnings of the IEC standards for safety critical software and hardware, and that today there are still flaws in the standards, and concerns that lead people to trust software "*beyond its trustworthiness*".

Actor C.3:
People recognised that software had the potential to cause computer-based protection systems and control systems to fail, obviously. And there was concern about it. The Health and Safety Executive had a team of specialist engineers who were responsible for programmable electronic systems, and for doing the inspection assurance, run by a man called Ron Bell. He had the foresight to set up an International Standardisation Committee, looking into developing standards for building these programmable electronic systems, and in fact, that led to the British Standard, I think, 1508, which finally became the international IEC Standard 61508 which is the international standard for safety critical hardware and software for programmable electronic systems.

There are still fundamental flaws in that standard, as there were at the time. So, the professional body was very much on the case trying to look at people who were building safety critical software and seeing what expertise was being brought to bear, to really provide adequate assurance that these systems were safe enough. Because we were concerned that people were basically trusting software beyond its trustworthiness.

8.4 Software Engineering Aspects

Software engineering aspects from inside the organisation were briefly covered in the interview, as Actor C was not privy to the internal activities of the organisation. Concerns were raised from inside the QA team, doing verification and validation of the computer based primary protection system for a proposed nuclear power plant and reactor. The insight and reflections on these aspects, discussed with Actor C, are presented here.

8.4.1 Awareness of Issues with the Design

The organisation's QA Team are aware of issues (not meeting regulatory standards) and publish them internally in their QA Report.

8.4.2 Concerns of the Professional Body

Actor C discussed the concerns of the professional body, that testing alone was not sufficient to validate the systems. That the software itself needed to be analysed.

Actor C.4: *There were eight or so of us drawn from both academia and from industry in the Safety Critical Group. We became aware there was concern from inside the verification and validation team regarding the proposed primary protection system for a proposed nuclear power plant and reactor. There were concerns it would not meet the regulatory requirements, requiring them to produce a system that had high confidence of a failure probability on demand of 10^{-4}.*
They (the organisation) felt they couldn't produce scientifically valid evidence that was better than 10^{-3}. They were an order of magnitude out. And as I recall, the rest of us on the committee didn't believe in the 10^{-4} either. It seemed to us that the amount of testing that would need to be done in order to get to a 99% confidence level wasn't feasible, and that in any case, probabilistic analysis was not good enough. You needed something more rigorous. You needed to analyse the software itself. The task force was therefore very interested in how they were going to develop assurance of a computer based primary protection system for a nuclear reactor.

8.4.3 Analysable Software

Actor C's philosophy is that you must build safety critical software to very rigorous standards, to the most rigorous standards you possibly can. Actor C's belief was that the organisation was not doing so.

Actor C.5: *Software developers have unique strength, which is software is a mathematical object and therefore reasoning about software is entirely feasible. Complexity can be huge, so you have to design the software to be analysable.*
But in principle you can do it, and then if you're professional about what you're doing, and you're building something where you really want to be sure it's not going to fail, there's no alternative, you have to do that as well as testing it. .... does it actually do what it what it's supposed to do and nothing else? And are there any combinations of inputs that could possibly cause it to crash for internal reasons? For you know, arrays going out of bounds or integer overflows, or you know those sorts of entirely technical internal software issues. There's no other way of doing it. You've got to do it by proof, and that is completely feasible. It irritates me to this day that in general people don't do that. Some companies do but most companies don't.

They went on to discuss the standards of standards and use of formal methods:

Actor C.6: *There's a lot of lip service paid to the importance of very high quality in security critical and safety critical systems. There is immense industry resistance to the use of formal methods. Because they are seen as firstly being a technology that most of IT industry doesn't have. They don't know how to do it. They don't know how to train people. There is a great fear that it's extremely expensive. And therefore, it would put up costs and damage their businesses.*

Actor C also discussed the use of third-party software components in safety critical systems, where you cannot be assured of the quality of such systems.

Actor C.7: *…and one of the things you can't do, is to build out of components that somebody else is built that haven't followed those sort of standards. And given that almost all IT system these days are built out of APIs, and you know online components in component libraries on GitHub, that people simply draw down and bolt together. It not a practical task to assure the quality of such systems built that way, and so it requires a revolution in the way that software is developed. If we're really going to be able to build systems that are really safe and really secure and where you can provide strong assurance that that's true.*

8.5 Actor Actions

8.5.1 Awareness of Issues with the Design and Testing

The organisation's QA Team are first aware of issues (not meeting regulatory standards) and publish them internally in their QA Report. It is not known how this QA report was shared or discussed internally. The report was instructed to be deleted and not shared externally with the nuclear inspectorate. People within the organisation complied with the management request not to publish the report externally. We do not know how the software engineering teams, their managers and the QA teams responded to the findings of the QA team or being told to suppress the report.

8.5.2 The Leaked Report

Actor C discussed being notified of an internal report regarding the design of the computer-based protection system failing to meet regulatory requirements. Actor C viewed this notification as an anonymous whistleblowing action. Actor C does not name the person reporting the issue.

Actor C.8: *The thing that shocked me was internal reports about failing to meet regulatory requirements had been suppressed inside the organisation. That the internal assurance team were told to delete the report they wrote, from the repository of reports that were available to the Nuclear Inspectorate - who were in turn checking the work that was being done and providing the oversight. They were to remove the report and anything that indicated it had ever existed. This raising awareness was an informal whistleblowing action to the professional body.*

8.5.3 A Public Inquiry

The professional body was aware of a public inquiry into Hinkley Point C. Actor C followed the inquiry proceedings and was concerned that the software-based protection system was not mentioned. Actor C was granted permission, on behalf of the professional body, to give evidence regarding safety concerns.

Actor C.9: *There was a public inquiry into building a new nuclear power station. I was tasked with the responsibility, on behalf of the professional body, of keeping an eye on that public inquiry. To see at what point they looked at the safety of the primary control system, which we knew was computer based. It was a public inquiry, and they published the transcript of all their proceedings. They were doing it systematically and in stages.*

Actor C.10 …*I read all the transcripts of the safety area. I was shocked they hadn't mentioned the computer systems at all. The fact that the primary protection system was software based, and that this was novel, was not mentioned in the safety part of the public inquiry.*
The public inquiry moved to a phase where they were holding regional days, to allow the communities that were relatively close to the site to have their say. I asked for permission to raise the safety of the primary control protection system. I was granted that, on behalf of the professional body. I went along and gave evidence saying we were shocked they covered safety but hadn't mentioned the computer system at all. I got a polite hearing and a few questions asked.

8.5.4 Protecting the Whistleblower

Actor C did not reveal the existence of the suppressed report to the inquiry for fear of putting the whistleblower in danger. Actor C felt that the professional body putting forward their concerns was sufficient.

Actor C.11: *I didn't feel it was appropriate to reveal the existence of the suppressed report to the public inquiry because the danger was that I would put at risk the whistleblower. I felt that if we (the professional body) were, as a professional society, putting forward serious concerns about the ways in which the primary protection system could be assured, that was enough, that was doing the job that needed to be done.*

8.5.5 Retaliation Against the Professional Body

There was formal feedback sent to the professional body, via lawyers, from the organisation building the nuclear reactor.

Actor C.12: *A week later I got a letter from eminent legal professionals, representing the people building the reactor, asking me what I thought I was doing raising this issue at the public inquiry, and surely this is something that the professional body should have raised in private. But nobody had approached us about it.*

Nobody ever approached us to say, you know, this is what we're doing, what do you think about it? As it happened, the professional body had, as a member, somebody who worked for the organisation on the assurance side on the task force. So, it wasn't as if they didn't know that the professional body had set up a task force for safety critical computing.

8.5.6 Retaliation Against Individual

There was some retaliation to speaking up at the public inquiry. Actor C believes their house was broken into by people looking for a copy of the suppressed report produced by the internal QA team.

Actor C.13: *Some while after my appearance at the inquiry, my house was broken into. I was at work. My partner was at work. The house was locked up. We had a cleaner coming around who had a key to the front door. She couldn't get in. The front door was on the chain. And that didn't make any sense to her, so she contacted one of us and we came back to find that the back door of the house was open. The front door was on the chain and one of the windows was broken.*

Actor C.14: *... We called the police; it was clear that forced entry had been made through a kitchen window. The people who had broken in had walked past an expensive camera that was sitting on a table in the kitchen. And walked past cash that we left out for the cleaner. They had gone upstairs to the half-landing that I was using as my office and had gone through a bag that contained all my papers. They had clearly put the chain on the front door to stop themselves being disturbed. Having been disturbed by the cleaner trying to get in, they had fled out of the back door, leaving the back door unlocked (because they couldn't lock it behind them).*

I have always believed there was a concern that I possessed a copy of the internal report that had been hidden from the nuclear inspectorate. And they wanted to make sure that it disappeared. But of course, I have no evidence for that, but that's the only explanation that I, or indeed, I suppose the police could think. The view of the police, after they looked around and had seen this chain of events was "have you got any secret papers in the house?" It was clear that somebody was looking for papers and weren't thieves. This wasn't people breaking to see what they could nick. I said no, I don't have any secret papers in the house. The police didn't take it any further. That was the last we heard about it.

Actor C was also warned that their own organisation (not involved in the story prior to this point) would not work in the nuclear industry in the future, they reflect on the penalties of doing the right thing.

Actor C.15: *...the nuclear inspector would insist on better assurance. But of course, it made them cross. And I was told at a meeting subsequently that my company would never ever get any work from the nuclear industry... so you know that there is a penalty to trying to do the professional thing.*

8.5.7 Role of Professional Bodies

Actor C reflected on the role of professional bodies to uphold the standards:

Actor C.16: *That in a sense, that's why professional society have to be professional [following industry retaliation and threats]. It's like they have to be independent and why it's important that people are able to work on professional societies without any real threat. Although, of course, I was known to the nuclear developers.*

8.6 Outcomes

8.6.1 Delays

Actor C believes the professional body interventions cost the nuclear industry time and money. Actor C reflects that the developers involved might argue that their testing would have been sufficient:

Actor C.17: *Our intervention cost them a delay and quite a lot of money to have to build the tools that enable them to do rigorous code analysis on the protection system, to satisfy the inspectorate. I would argue it added to the safety of the system. The original developers would argue their testing would have been good enough and who knows?*

8.6.2 Changes to Quality Assurance

As a consequence of the concerns that the professional body raised, quality assurance of software in the safety critical industry was done differently and new static analysis tools were developed. Testing alone was not sufficient.

Actor C.18: *As a consequence of the concern that the Professional Body had raised about the assurance ... it was decided [at the organisation] that they had to do assurance differently. The testing wouldn't do. That what they had to do, therefore, was to use static analysis tools. They ended up writing a disassembler, to take the code that they had written for the primary protection system and to turn it back into the intermediate language (for analysis). The Ministry of Defence MALPAS system, which was a static analysis tool that had been produced for analysing code to look for Trojan code and do dataflow analysis, on software for security reasons. The director of RSRE (Royal Signals and Radar Establishment) declassified it in order that it could be used for safety critical work in industry.*

8.7 Case Summary

There are a broad range of actors from the software engineers through to those impacted by proximity to harm from nuclear reactor (people and environment). There are breaches of regulations and covering up of the breach. However, there were only two clear actions of reporting up inside the organisation – the initial report being published (but suppressed) internally, followed by the external leaking of the report to the IT professional body. Once the IT professional body was informed, concerns about safety critical systems were raised more widely within the nuclear industry and more widely to society through the news media and the public inquiry.

Chapter 9. Case Findings

This chapter summarises and discusses the findings from the case studies in the form of answers to RQ5. The findings (F1 to F16) are summarised in Table 9-6 and discussed in Chapter 10 with reference to the literature review.

9.1 RQ 5-1: What Harm or Wrongdoing is Detected?

Coverage of government defined whistleblowing harm or wrongdoing situations, by case, is presented by story wheels in Figure 9-1. Table 9-1 presents a summary of stories from participant interviews, the actors at risk of harm, the harmful behaviours described and the industry domain the story is set in. The majority relate to safety critical software found in health, nuclear power, automotive, and passenger transportation systems. Other entries relate to issues with smart city technology (IoT devices).

Aspect	Legend	Case Study A (Health)	Case Study B (Automotive)	Case Study C (Nuclear)
Whistleblowing Situation	1. Law or regulation breach 2. Health and Safety (human) 3. Risk or actual damage to environment 4. Covering up wrongdoing			

Figure 9-1 Whistleblowing Situations in Each Case

In all three cases software engineering experts witness regulation breaches within their organisations. Case A relates to fraudulent activities regarding a manager changing and backdating pre-installation system checks, system specifications and test reports to cover up test failures. Case B reports on code developed to evade vehicle emission tests, a breach of vehicle regulations that was not reported to the regulator. In Case C a safety critical report is suppressed from a nuclear regulator. The actors at risk of harm vary across each case. In Case A health and safety issues specific to patients on ventilators. Case B and C are raising risks of both environmental and health and safety-based harms. Stories 4, 5 and 6 involve public transport with vehicle operators, passengers, and the general public at risk of harm. Story 7 involves networked IoT devices with emergency features. Extracts of stories 4, 5, 6 and 7 can be found in Appendix E (Story Extracts).

Table 9-1 Harm or Wrongdoing Detected (RQ5-1)

Case (Story)	Situation or System	Actor at risk of harm	Harm or Wrongdoing	Domain (Context)
A	Ventilator design and test reports	Ventilator patients	Falsifying pre-installation check lists and backdating test reports. **Regulatory breach**. **Poor software engineering practices (testing processes).**	Health
B	Defeating emissions tests	Environment. General public	Adjust engine performance during emissions tests. **Regulatory breach. Disguising of functional requirement in code and documentation**.	Transport
C	Computer based nuclear protection system	General public Environment	System **not meeting regulatory requirements** for safety critical systems. **Suppressing of reports** indicating non-compliance with regulatory requirement.	Energy
(4)	Train protection system	Passengers and crew	Insufficient test equipment to ensure embedded software in safety warning system worked as designed. **May not meet rail standards.**	Transport
(5)	Design of air traffic control solution	Passengers and crew	Over engineered solution (unnecessary abstraction) impacts testing capability and coverage. **Concerns recorded and monitored**.	Transport
(6)	System integration testing	Passengers and crew	Unable to evidence software engineering and systems integration processes complied with industry safety standards. **Regulatory breach. Cover up.**	Transport
(7)	Poor quality solutions	General public	Quick fix solutions proposed by manager **not compliant with security standards**. Instructed to sign off on **sub-standard emergency feature**.	Smart City

Figure 9-2 shows visually how the different reporting up pattern is in each of the cases. Of particular note is Case B where management were successful in suppressing the software engineering team concerns over regulation breaches (**F1**) despite evidence of the wrongdoing captured in software artefacts (**F2**).

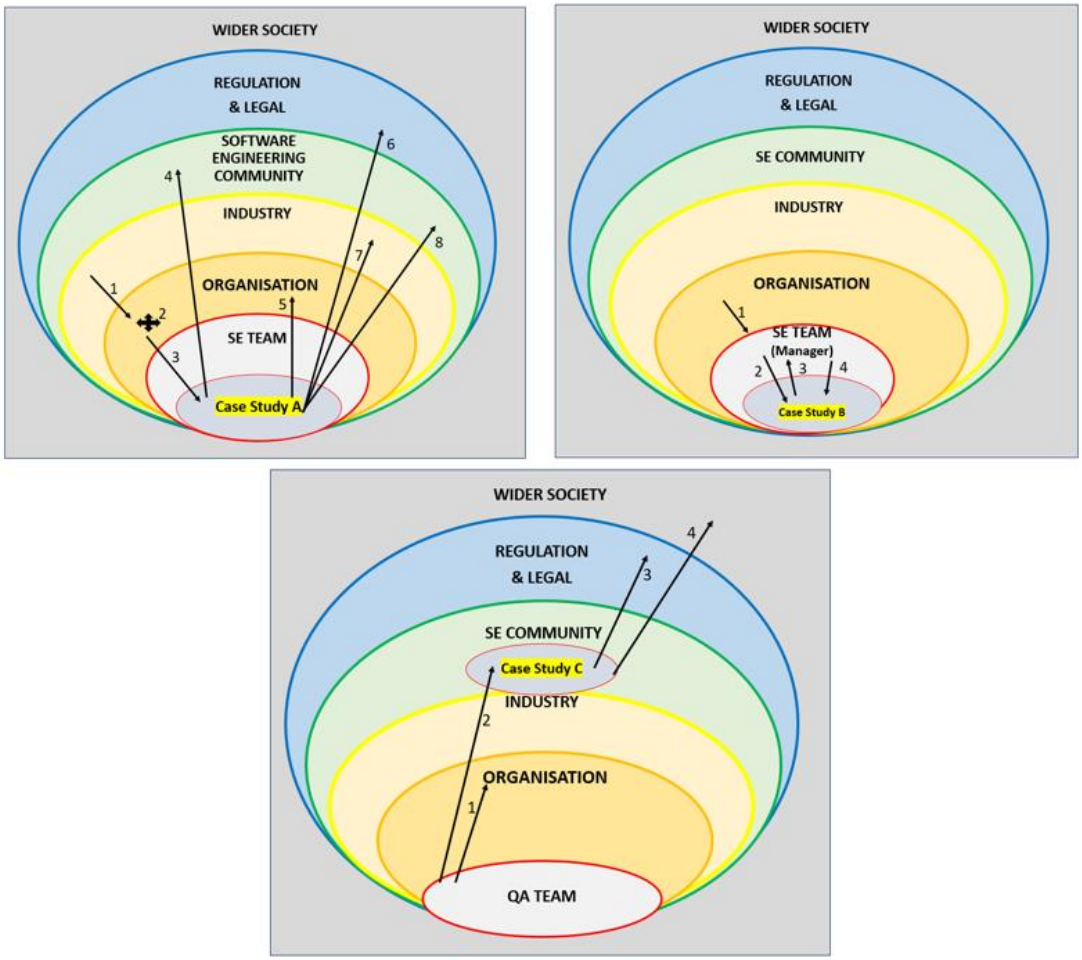


Figure 9-2 Comparing Case Reporting Actions

In all three cases management roles in the organisations are found to be complicit in disguising or suppressing concerns raised by software engineering experts in regard to regulations and regulators which deterred actors from reporting up the situation (**F1, F3**). The concerns raised in the cases are evidenced by internal software engineering artefacts (code, documentation, test reports or technical reports) written by software engineering experts (**F2**). In Case A and C this evidence was an enabler for reporting up the situation. In all the cases management had mechanisms (instructing the cover ups and the making of threats to teams and individuals) to suppress the evidence of risk or the breach of regulations (**F3**). These three findings suggest management have a poor attitude towards software engineering experts and of their compliance and responsibility to the regulators.

In Case A management actions were not sufficient to prevent a software engineer reporting the organisation directly to regulators, having taken advice from a professional body. In Case C concerns were indirectly raised through a professional

body taking part in a public inquiry and writing articles in mainstream news media. In Case B, a team of software engineers subsequently had their collective concerns suppressed by management (**F3**). The regulator was not made aware of the breach of regulations, all risks and concerns were kept within the software engineering team. In all three case studies (and the stories listed in Table 9-1) issues or decisions regarding software products is known about more widely than the software engineering team designing and implementing the solutions. This knowledge becomes a team and organisational "*secret*" (**F4**) with group complicity to suppress or cover up issues.

9.2 RQ 5-2: What Software Engineering Practice Aspects are Involved?

This section focuses on the technical aspects of the cases and stories. Figure 9-3 gives a high-level summary of the software engineer aspects found in each case. Case A is rich in the technical content, with a broad range of software engineering issues raised and discussed. Case B was from the perspective of a software engineer involved in the misuse of code to detect and evade emission tests; there were no concerns raised about the quality of the software, tools or processes followed. Case C was from the perspective of a professional body to whom a concern had been raised about the safety of a computer-based protection system not meeting required standards. The primary actor in Case C had limited internal knowledge of the organisation and technical people involved in the situation. The actors and professional bodies in Case A and C do have expertise of the practices, standards and tools required to develop safety critical systems. Stories 4, 5 and 7 are from the perspective of software engineers concerned with project testing and security issues. Story 6 is a safety engineer wrangling with an engineering design team reluctant to follow contractual quality assurance processes.

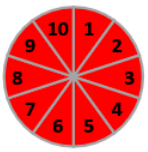
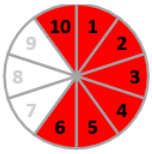
Aspect	Legend	Case Study A (Health)	Case Study B (Automotive)	Case Study C (Nuclear)
Software Engineering	1. Code 2. Software 3. System, Product or Service 4. Documentation 5. Agreements and Contracts 6. Standards, Regulations 7. SE Processes and Practices 8. SE Tools 9. Technical skills of team 10. Management			

Figure 9-3 Software Engineering Practice Aspects in Each Case

9.2.1 Code, Software, Systems, Products or Service

In all three case studies the technical details, functionality or quality assurance of software systems are described. Case A identifies poor system design and management of software engineering processes, specifically the testing processes (**F5**). Case B describes the misuse of code to detect emission tests and the tampering with code to disguise the true functionality (**F6**). Case C reports on concerns with the design and quality assurance (validation and verification) of safety critical software systems. Case A and C additionally raise concerns with the use of "*Software of Unknown Provenance*" in safety critical systems where components are used that are designed, developed, or tested and with limited access to development artefacts of those using the component. Stories 4 and 6 also report on poor design, management, and testing processes. Story 6 also raises concerns with third party components and their integration into the transport system solution.

9.2.2 Software Engineering Skills, Processes, Practices and Tools

In Case A significant issues were raised with the design of solutions and the software engineering processes, practices, and tools to build and test (**F5**). The technical skills of software engineering teams were not raised as an issue in Case B and the software went into production vehicles. In Case C the professional body had no inside knowledge of the software skills or processes, only awareness of the Quality Assurance team performing verification and validation activities on the system designs.

Stories in Table 9-1 all demonstrate concerns from poor quality software engineering processes (**F5**). In particular the testing, auditing and quality assurance of software and software engineering processes alongside management instructing software experts to sign off on sub-standard processes and systems. In Story 5 concerns were raised about an over engineered air traffic control system that was difficult to test, the organisation recorded and monitored the risk. In Story 7 actors refused to code the proposed quick fix poor-quality solution and went on to develop a robust secure solution. Also in story 7, on discovering the IT project issues, software engineers raised and then addressed concerns with the code; referring specifically to "*harm to software codebases*" and the subsequent consequences for the project team and project deliverables.

9.2.3 Technical Documentation and Licensing Agreements

In all three cases technical documentation could evidence the software risks and wrongdoing (**F2)**. Case A references Microsoft licensing agreements and the exclusions of use in relation to safety critical systems. In Case A copies (paper and electronic) of the back dated test reports are gathered and provided to the regulator. In Case B, the software documentation counters the disguising of the true functionality of the code (**F6**) being implemented by naming the engineering manager requesting the requirement (**F2**). In Case C there is a suppressed QA report (**F2 and F3**) highlighting not meeting minimum safety standards. In contrast, in Story 6 (Table 9-1) there is a lack of documentation to evidence compliance with software engineering and integration testing procedures that in turn lead to a safety engineer refusing to sign off assurance of a transport project.

9.2.4 Professional Standards and Regulations

In Case A and C, software engineering experts are reported to be strongly motivated by professionalism and of the upholding standards both from the technical and ethical aspects (**F7**). Experts were insistent on not being silenced and sought mechanisms for speaking up about the situations, despite the retaliation and impact on themselves personally (**F8**). Case B involves a software engineer in their first job after leaving university, and reports finding it difficult to know how to speak up (**F9**), but that the incident guided their future responses to situations where professionalism and ethics were challenged (**F10**). In Case A and B, the engineering manager or team leader were complicit in breaching regulations and the cover up of issues through threats to people's jobs (**F1 and F3**). In Case C a report was suppressed from the regulator though no direct evidence was given of the internal suppression mechanisms. Stories 4, 5, and 7 were from software engineering experts citing their professional standards as the motivation to correct the poor practices that could lead to harmful (health and safety of humans) situations (**F7**).

9.3 RQ 5-3 What Actions Do Actors Take?

This section looks at the actions taken by actors witnessing or suspecting harm or wrongdoing issues. Stories were selected to take forward as case studies based on their coverage of types of harm or wrongdoing and of actions taken. Table 9-2 shows presence of dissent, whistleblowing, silence or exiting the situation in each of case studies (**F16**) and discussed previously during case selection in Section 5.6. All four actions are identifiable in Case A with the software engineer blowing the whistle and exiting the organisation, there were other actors in the organisation consenting (**F12**) or being silent (**F11**) about the harm or wrongdoing. In Case B there was initial dissent within the software engineering team when asked to implement emission detection code, then followed by consent and compliance with wrongdoing (**F12**), though evidence was left in the code and documentation as to the source and purpose of the requirement (**F2**). In Case C it is unknown what if any dissent happened internally at the organisation or if anyone left as a result. It was known that internally the report was suppressed by management (**F1**) and that concerns were raised externally (**F15**), and the name of the internal whistleblower protected (**F8**). The next three sections report on actors and actions taken in the case studies.

Table 9-2 Actor Actions, based on Anvari Model

	Case			Story			
Action identified	**A**	**B**	**C**	**4**	**5**	**6**	**7**
Dissent	Yes	Yes	Unknown	Yes	Yes	Yes	Yes
Whistleblow	Yes	No	Yes	No	No	No	No
Silence/Consent	Yes	Yes	Yes	No	No	Yes	No
Exit	Yes	No	Unknown	Yes	No	Yes	Yes

9.3.1 Human Actors Involved

The blue wheels in Figure 9-4 give an overview of the groups of human actors involved in each case. All cases involved domain and industry specific regulators and the upholding professional software engineering standards. Case A and C involved actors actively seeking guidance and support from the software engineering community (a professional body) and blew the whistle on the issues. Case B involved a junior software engineer not long out of university who sought no advice and did not report issues outside of the software engineering team or organisation (**F9**).

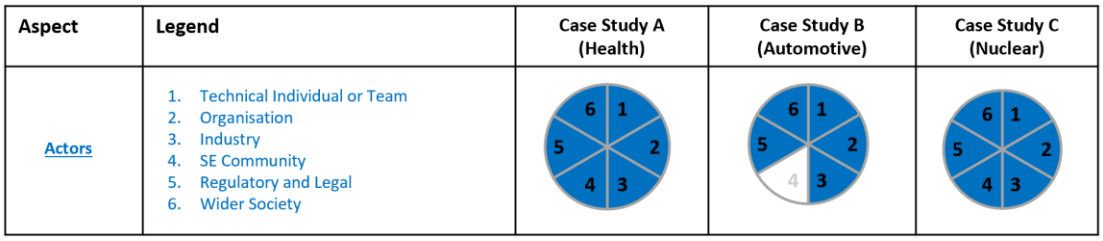


Figure 9-4 Actor Profile Wheels for Each Case

All cases covered issues of professional ethics and standards, regulation breaches and the potential impact on end users and on wider society. In Case A and B, the regulatory issues were known to the software engineering teams. In Case A, alongside the software

engineer, distributors and external consultants reported further issues to regulators. In Case B, there was no evidence anyone outside the organisation was aware of the situation and so an "*organisational secret*" was established regarding the wrongdoing (**F4**). Case C focuses on a professional body who had standards and regulatory concerns reported to them by an internal organisational actor. The issues were subsequently raised within the nuclear industry, the software engineering community, at a public inquiry, and through a public newspaper article. There was no evidence that actors in Case C reported their concerns to the nuclear regulator.

9.3.2 Actions Taken by Actors

The green wheels in Figure 9-5 shows a summary of decisions and actions taken by those aware of the issues arising in each case story.

Aspect	Legend	Case Study A (Health)	Case Study B (Automotive)	Case Study C (Nuclear)
Known Actions Taken (by any story actor)	1. Took no action (keeping quiet) 2. Complied with instructions 3. Attempt to change team or organisation 4. Report concerns to external body 5. Raised concerns to a manager 6. Raised concerns with colleague 7. Exit team 8. Exit company			

Figure 9-5 Whistleblowing Related Actions in Each Case

In Case A and B several people inside the organisations are perceived to take no action or comply with management instructions **(F11 and F12)**. In Case A pre-installation check lists and test reports were observed to be changed and backdated in a team meeting. In Case B emissions test detection code was implemented and released into production cars. In Case C a technical report was suppressed from regulators, the actors complicit in this is not reported. In Stories 6 and 7 actors refused (dissented) to comply with management instructions to sign off sub-standard solutions, however it was reported that other experts were sought who did then sign the solutions off.

In Case A and B attempts are made to raise concerns regarding the design of software with colleagues and then team managers **(F13 and F14)**. In Case A this action is successful in bringing in external consultants and a new static analysis tool to help improve the testing. No attempt was made to change the team or organisation regarding the standards breach; concerns were taken directly to the regulator (**F15**). In Case B the software engineering team raise concerns and initially refuse to implement functionality requested by senior engineers and team managers. Following threats to their jobs the software team go on to implement the feature (**F8 and F12**).

In Case C, other than complying with management to suppress the QA report, little is known about internal actions taken. The organisation in Case C was later reported to have built software testing tools to enable more rigorous code analysis on nuclear protection systems that would in turn satisfy the nuclear inspectorate standards. In Story 5 design concerns were raised internally with colleagues regarding the unnecessary complexity (too much abstraction) of an air traffic control solution. While the solution was not redesigned, the concern was formally recorded and monitored within the team.

It is not known if the code went into production or if concerns were raised with regulators.

In Case A the organisation is reported to the regulator, and a professional body is consulted about the situation **(F15)**. In Case B concerns were not reported to any external body. In Case C a professional body is made aware of issues with safety standards at Organisation C **(F15)**, the professional body in turn becomes involved in a public enquiry and with a standardisation of processes for software in safety critical systems.

In Case A there is evidence of people leaving the organisation by choice or through dismissal linked to the reporting up of issues **(F10)**. The team manager and a test engineer involved in the wrongdoing are known to have been dismissed and left the organisation at some later date. In Case B and C it is not known if anyone exited the organisation because of the incidents. In Stories 4, 5, and 7 software engineers left the organisation by choice immediately or at the end of their contracts, in part due to concerns and issues raised **(F10)**. In Story 6 a safety engineer initially remained with the organisation but was moved onto different projects, although further quality issues caused them to subsequently leave the organisation by mutual agreement.

9.3.3 Escalations

This section looks specifically at the reporting up escalation actions in each case. Figure 9-6 shows the profile wheel from each case (**F13 and 14**).

Aspect	Legend	Case Study A (Health)	Case Study B (Automotive)	Case Study C (Nuclear)
Reporting Up	1. Report to an individual 2. Report to team 3. Report to management 4. Report organisation wide 5. Report to industry 6. Report to SE community 7. Report to regulator 8. Report to wider society			

Figure 9-6 Reporting Up Actions Identified in Each Case

In Case A the primary incident is first informally reported up by an office junior to a software engineering expert (colleague) in the organisation **(F13)**. The software engineering expert then reports up the situation internally and externally **(F14 and F15)**. In Case B the harmful situation is discussed within software engineering team and is raised with the team manager (**F13 and F14)**. It is not known how Case C was reported up internally (if at all), other than a report was produced for management that was subsequently suppressed.

In Case B the action of the software team documenting the source of the requirement could be classified as type of reporting, though the team would not know if, how or by whom the document might be subsequently read internally or externally and for what purpose. It is also not known if any record is made of the team initially refusing to code the requirement or how the team were threatened with losing their jobs if they did not comply with instructions. Based on the software artefacts alone the team could be seen as complicit with the requirement; they wrote the code, and the code knowingly went into production **(F2 and F6)**.

9.3.4 Decisions and Actions

Findings on how decisions are made, by software engineering experts, to blow the whistle are presented in Table 9-3, based on Keenan and McLain's whistleblowing model (Keenan and McLain, 2017) . In Case B there was no record of searching for others to correct wrongdoing and in turn there was no whistleblowing actions taken. In all other aspects the case studies evidence applicability of existing decision theories and process models (**F16**).

Table 9-3 Assessment and Decision to Blow the Whistle (F16)

	Case		
7-Steps of whistleblowing	**A**	**B**	**C**
1. Awareness of issue	Yes	Yes	Yes
2. Assessment of seriousness	Yes	Yes	Yes
3. Motivation to correct	Yes	Yes	Yes
4. Assessment of personal influence	Yes	Yes	Yes
5. Search for others to correct wrongdoing	Yes	*No*	Yes
6. Assessment of consequences to self	Yes	Yes	Yes
7. Assessment of management complicity	Yes	Yes	Yes
Blew the whistle?	**Yes**	**No**	**Partially**

In Case A the time urgency of harm to neonatal babies is the primary concern (**step 1 and 2**). Additionally, serious concerns about management behaviour covering up the situation alongside of poor software engineering practices and understanding of software standards (**step 7**) was evidenced. In Case B a team of software engineers are concerned being asked to code software to detect emissions tests (**step 1 and 2**). The organisation justifies their decisions as, in their opinion, car buyers prefer engine performance over reducing impact on the environment and meeting government regulations. This may explain other actors not seeing the situation serious enough to report (**step 2**). In Case C serious concerns are raised about the computer based nuclear protection systems and the impact on the UK should any such safety critical system fail (**step 1 and 2**).

In all case studies actors reported on being motivated to correct the situation (**step 3**). In Case A and C actors sought the help of others outside of their organisation (**step 5**). The software engineering team in Case B was motivated to correct the situation (**step 3**) and initially refused to code the emission detection code. As an inexperienced software engineer, they did not seek help from others outside the software engineering team or organisation (**step 5**). In all case studies actors assessed the consequences of speaking up for both themselves (**step 6**) and others. In Case A, an office junior witnessed the wrongdoing, and chose not to formally report the issue themselves. Instead, they sought help (**step 5**) to discreetly raise concerns, suggesting they feared the consequences to themselves (**step 6**) from management known to be complicit in the wrongdoing (**step 7**). The office junior's involvement was not made known to management by the software engineer. The software engineers in Case A and B expected serious consequences for themselves (**step 6**) given management's complicity (**step 7**) in the situation.

In Case B, the software engineer reported on being inexperienced in industry situations, this was their first job after university. The personal consequences so early in their career (**step 6**) and complicity of management (**step 7**) led to no attempts to seek others

(**step 5**) or report up the situation. In Case C, a member of a professional body was made aware of issues at the organisation and deemed them serious enough for them to act (**step 1, 2 and 3**). While the professional body member did not work at or have direct influence on Organisation C, they knew of mechanisms to raise concerns in industry and more widely (**step 4 and 5**). The professional body member was aware of potential consequences to themselves (**step 6**) and the complicity of management (**step 7**) suppressing the QA report. As an experienced software engineering expert and because of their role in the professional body, there was a strong motivation to uphold industry standards (**step 3**) in this case and more widely (see Story 4 in Table 9-1, involves the same Actor).

9.3.5 Enabling or Inhibiting Factors

Table 9-4 summarises the factors reported as motivating, enabling, or inhibiting actors from taking actions in the cases and stories. Chapter 10 discusses these factors in relation to existing literature theories, models, and research findings.

Table 9-4 Enabling, Motivating, or Inhibiting Factors Reported

Enablers	Motivators	Inhibitors
• Years of experience • Confidence • Professional bodies • Standards and regulations • Evidence wrongdoing • Reporting mechanisms • Anonymity • Ability to protect self • Ability to protect others • Group action	• Professionalism • Immediacy of harm • Seriousness of issues • Proximity of issue	• Lack of confidence • Lack of experience • Management threats • Management complicity • Career threats • Not knowing process • No anonymity • Wrongdoing justification

9.4 RQ 5-4: What Final Outcomes are Reported?

In Case A internal and external reporting brought about changes in staff and software engineering practices. Actor A was dismissed from the organisation so, other than informal conversations with another former employee, knew little detail of what subsequently happened and how it came about. In Case B no known changes were reported within the organisation and vehicles went into production. In Case C external reporting influenced the future of standards for safety critical systems which in turn lead the organisation to develop new tools for more rigorous code analysis on safety protection systems. These outcomes are summarised in Table 9-5.

Table 9-5 Summary of Organisational Outcomes

Case	Known Organisational Outcomes
A	Organisation referred to regulator. IT consultants engaged to improve software engineering processes. Dismissal of Software Engineer
B	Financial benefits from vehicle sales. Ongoing use of code in breach of regulations
C	New tools and quality assurance processes at organisation

In Case A and B the outcomes are based on short term changes seen or reported. Study participants did not work at the organisations at the time of the study; it is not possible to reflect on the long-term changes or impact of the incident. Even if access to the organisations had been sought tracking down changes could be challenging and not necessarily attributable to these incidents.

In story 4 it is reported that the train protection solution did not go into production vehicles. In story 5, the potential concerns were recorded by the organisation, but the software engineering involved eventually left the organisation and was not aware if designs caused testing issues. In story 6 the safety engineer eventually left the organisation, reporting that while two projects did not complete, the passenger transport system, with some public controversy, was delivered.

9.5 Key Findings

Table 9-6 Key Findings from Case Study

ID	Theme	Short description	Case
F1	**Organisational and Personal Factors**	Complicity of management in wrongdoing	A, B, C
F3		Mechanisms to cover up, suppress harm or wrongdoing	A, B, C
F4		Team or organisational secrets about harm or wrongdoing	A, B, C
F8		Protection of self or others when reporting up situation	A, C
F9		Inexperience to raise issues outside of team	B
F10		Outcomes and impact of whistleblowing on career	A, B, C
F2	**Software Engineering Practice Aspects**	Evidence of issues in software engineering artefacts	A, B, C
F5		Poor quality software development lifecycle processes	A
F6		Tampering with artefacts to disguise wrongdoing	A, B
F7		Professionalism and upholding of technical standards	A, C
F11	**Actor Actions (Process)**	Taking no action	A, B, C
F12		Complying with management instructions	A, B, C
F13		Raising concerns with colleagues	A, B
F14		Raising concerns with managers	A, B
F15		Reporting externally	A, C
F16	**Model &Theory**	Coverage of steps in whistleblowing theories and models	A, B, C

9.6 Summary

Figure 9-7 brings together all the story wheels from the three cases:

Aspect	Legend	Case Study A (Health)	Case Study B (Automotive)	Case Study C (Nuclear)
Software Engineering	1. Code 2. Software 3. System, Product or Service 4. Documentation 5. Agreements and Contracts 6. Standards, Regulations 7. SE Processes and Practices 8. SE Tools 9. Technical skills of team 10. Management			
Known Actions Taken (by any story actor)	1. Took no action (keeping quiet) 2. Complied with instructions 3. Attempt to change team or organisation 4. Report concerns to external body 5. Raised concerns to a manager 6. Raised concerns with colleague 7. Exit team 8. Exit company			
Actors	1. Technical Individual or Team 2. Organisation 3. Industry 4. SE Community 5. Regulatory and Legal 6. Wider Society			
Reporting Up	1. Report to an individual 2. Report to team 3. Report to management 4. Report organisation wide 5. Report to industry 6. Report to SE community 7. Report to regulator 8. Report to wider society			
Whistleblowing Situation	1. Law or regulation breach 2. Health and Safety (human) 3. Risk or actual damage to environment 4. Covering up wrongdoing			

Figure 9-7 Case Summary of Story Profile Wheels

Chapter 10. Discussion

Driven by knowledge and research gaps identified in software engineering literature, my goal was to develop a method to study the actions and interaction of actors involved in whistleblowing situations in software engineering practice. This chapter discusses the key findings of my case study research and its positioning amongst existing literature in whistleblowing research, and specifically in software engineering research. The discussions are structured based on the whistleblowing themes (RQ1) and research findings and gaps (RQ2 and RQ3) identified in the literature review in Chapter 3. Table 9-6 presented the summary of the case study findings from Chapter 1, grouped by literature review themes which guides their discussion in this chapter.

10.1 Models and Theory in Software Engineering Research

Runeson and Host guide that theories should contribute to a framework for the analysis of data in case studies (Runeson and Höst, 2009). Stol and Fitzgerald reflect that much software engineering research is *not* guided by explicit theories (Stol and Fitzgerald, 2013) and Lorey et al.'s literature review (Lorey, Ralph and Felderer, 2022) finds that, when used, theories are rarely tested for applicability to software engineering contexts. Whistleblowing is a phenomenon that is not specific to software engineering, it sits at the intersection of decision sciences, organisational studies, and business ethics. General whistleblowing models are therefore applicable mechanisms to explore whistleblowing situations in software engineering organisations. My WISE analysis framework is underpinned by existing models, frameworks, and data sets as described in Chapter 4.

10.2 Organisational and Personal Factors

Management complicity covering up issues (Keil *et al.*, 2004), *and organisational secrets* (Grey and Costa, 2016) were findings in the three case studies. Case A evidenced actors deterred from speaking up to management but seeking colleagues to discretely report the situation to. In Case B the software engineering team initially refused to implement code and challenged the organisation's requests. However, following management threats to comply software engineers implemented the code. In Case C we know management was complicit in the suppression of a report, but we do not know the mechanisms of how this was actioned internally. In Case C it is reported that the organisation may suspect their suppressed report had been leaked to the professional body. In all cases organisations were seeking to withhold information from regulators. The cases evidence examples of the Mum Effect and Deaf Effect. The Mum effect (Smith, Keil and Depledge, 2001; Park, Im and Keil, 2008) describes employee reluctance to transmit bad news about issues on a project. The Deaf effect (Keil and

Robey, 2001) describes management reluctance to hear (turn a deaf ear) to people reporting issues.

In Case A and C, experts expressed having knowledge and experience of appropriate technical standards, professional codes of practice and reporting mechanisms to support taking action (McNamara, Smith and Murphy-Hill, 2018). Actors report developing this knowledge over years of experience and in doing so gain the confidence to speak up, alongside the motivation of maintaining their professional standards despite retaliation (Alford, 2007; Schilhavy and King, 2010; Vandekerckhove, James and West, 2013; Kenny, Fotaki and Vandekerckhove, 2020). In contrast, lack of knowledge and confidence is reported as an inhibitor, particularly by more junior actors which combined with management complicity and threats to their career, inhibit decisions to take action. Actors did not judge others for not speaking up about the situation, indeed the protection of other actors from negative personal consequences regarding the disclosures was discovered during case study, and not was not action in WISE analysis framework.

A recent study of ethical concerns of software engineers, reports on the need to help practitioners identifying concerns and building their collective power to resolve the issues (Widder *et al.*, 2023). In Case A the software engineer chose to act alone to protect others. In Case B the software engineering team initially refused to code the system. In Case C, an individual at the organisation disclosed the situation to a group at a professional body to flag and address concerns. In all three cases the harm and wrongdoing cannot be solved by any one individual. Different tactics were used in each situation to attempt to resolve the concerns. In Case A corrective action was needed to ensure test processes were being followed and that the solution design complied with safety critical standards – this was achieved. In Case B the desired solution was not to write code to cheat the emissions test, and instead accept performance limitations. In Case C safety concerns need to be addressed with better validation and verification tools, this was achieved.

Recommendation 1: The effectiveness of taking collective action in software engineering warrants further study including the roles of trade unions, professional bodies, and regulators. For a study to specifically look at trade unions, who protect worker rights and public interests, where there has been an emergence of technology specific trade unions (Google[50] and Microsoft[51]). At the time of writing the Public and Commercial Services union were supporting striking government IT workers from Fujitsu[52]).

Lack of knowledge and experience (Reijenga, Aslam and Guzmán, 2023) *alongside the personal consequences for self and others* (Keenan and McLain, 2017) was described as inhibitors to whistleblowing and were in part linked to management's complicity in covering up, suppressing situations, and threatening employees. In Case A, a software engineer protected others from the consequences of being involved in reporting the organisation to the regulator. The software engineer took on the responsibility of

[50] https://www.theguardian.com/technology/2021/jan/04/more-than-200-us-google-employees-form-union
[51] https://www.wired.co.uk/article/united-tech-and-allied-workers-union
[52] https://www.theguardian.com/business/2024/jan/16/fujitsu-it-workers-hmrc-systems-to-go-on-strike

speaking up, of evidencing the issues, and ultimately lost their job as a result. In Case B, the software engineering team was threatened with loss of jobs if they did not implement the code as instructed. Software engineers reported on not having sufficient knowledge, experience, or confidence to speak up. In Case C the professional body intentionally did not mention their informant at the inquiry to protect them from retaliation. In all the case studies there was retaliation actions directed at actors who raised concerns (Alford, 2007).

Anonymity (Bodó, 2014) *while evidencing and disclosing wrongdoing* are factors in the three cases. There are actors in all three cases reported as not speaking directly to senior management because, even with an anonymous action, the source of the disclosure would be attributable to themselves or one their close team because of the nature of evidence to be provided (Larson, 1971). In Case A and C, actors protected the original observer of the wrongdoing and in effect enabled them to speak up anonymously (Miceli, Roach and Near, 1988) while still supplying detailed evidence. The role of management in the cases are predominantly negative and appear to inhibit reporting up by their complicity in causing harm or wrongdoing (Keenan and McLain, 2017), they do so by being able to justify the harm to actors, or by making threats to actors. In Case B, initial dissent against management (refusal to code) was enabled through group action (Anvari *et al.*, 2019), though inexperience and threats to their careers subsequently caused the team to comply with management instructions.

10.2.1 Perception of Seriousness of Issues

Safety critical systems in highly regulated industries were features of most of cases and stories captured. Given whistleblowing is defined in terms of breaching regulations and standards and the potential impact to cause harm to people or the environment, this is not an unsurprising finding.

Recommendation 2: Future research focusing on whistleblowing stories in critical national infrastructure domains could give specific and actionable insights raised by expert actors in these domains.

Table 9-4 presented details of factors reported to motivate, enable, or inhibit whistleblowing. Motivating factors related to immediacy of harm (Park, Im and Keil, 2008), proximity to victims (Park, Keil and Kim, 2008), and seriousness of harm from the situation (Smith and Keil, 2003; Keil, Im and Mähring, 2007; Park and Keil, 2007; Park, Keil and Kim, 2008). In the case study these factors were each found at different points in the product lifecycle. In Case A ventilators were in production and being delivered, thus action was needed to be taken with some urgency. Cases B and C were less urgent as the issues were discovered during early design phases, though the consequences of harm are more far reaching on the general public. Additionally, in Case C there was a proximity to the proposed nuclear development by the professional body member raising the issues that also motivated them to speak up.

Reflections on the seriousness of the harm and wrongdoing (Park, Keil and Kim, 2008; Wang, Keil and Wang, 2015) and how other actors might perceive the situation was discussed in the cases and why other actors may feel the situation is not serious enough to report up. In Case A the quality of previous products could have been deemed sufficient to assure quality of newly designed products. In Case B the perceived concern for environmental impact was low and management's belief was that customers were

more concerned with performance than emissions. In Case C existing validation and verification checks might be deemed sufficient by the organisation. However, as per the definition of whistleblowing, actors in the cases had a reasonable belief their concerns were valid. The handling of the situations with cover ups, threats, and suppression of evidential documents suggests organisations had concerns about their product designs that needed to be protected. Unmanaged disclosures could cause reputational damage to the organisation (Near and Miceli, 1996).

Recommendation 3: future studies to explore specifically how management and organisations triage whistleblowing situations, with a focus on the reaction to whistleblowing actors and the availability of software engineering artefacts to support the disclosures.

10.2.2 Keeping Mum and Deaf Effect

Two papers from the literature review specifically study employees keeping quiet or management not listening to issues on IT projects (Smith, Keil and Depledge, 2001; Park, Im and Keil, 2008). Several other papers discuss it as part of their findings (Keil and Robey, 2001; Smith and Keil, 2003; Keil *et al.*, 2004; Keil, Im and Mähring, 2007; Park and Keil, 2007; Park, Im and Keil, 2008; Park, Keil and Kim, 2008; Keil and Park, 2010; Vandekerckhove, James and West, 2013; Wang *et al.*, 2017; Petter, 2018). The mum effect is found to be a behaviour for both students and practitioners, including software developers and internal IT auditors. Job security, organisational size and the audit function relationships to senior management are identified as factors affecting if internal auditors remain quiet or assert their role responsibility to report bad news. Papers reporting on the deaf effect (Keil and Robey, 2001; McCubbrey and Fukami, 2009; Ghoshroy, 2019) focus on the heightened risk of costly project failures when not listening for or not speaking up early about issues. The difficulty of cause and attribution of blame and punitive economic punishment for software project failures are a known issue in software engineering literature and practice (Charette, 2005). However, the impact of keeping quiet or not being listened to can have wider implications than economic ones: from disrupting democratic processes, to costing people lives.

10.2.3 Professional Bodies

Two of my cases demonstrated how involving a professional body can support software experts to raise concerns – either through guidance (Case A) or as a mechanism to anonymously raise an issue outside of the organisation (Case C). Kline (Kline, 2010) and Hersh (Hersh, 2002) call for a study on the actions (and in-actions) of professional bodies in respect of whistleblowing cases. Whilst research and advocacy groups stress the importance for organisations to understand their obligations, and for individuals to be aware of the support and protection available to them should they choose to take whistleblowing action (Vandekerckhove, James and West, 2013).

An IEEE statement elaborates why they (the IEEE) are not always in a position to support whistleblower cases: "*diverse opinions may flow from the same set of circumstances [...] and there are always at least one other side to a story [...] the IEEE does not have the responsibility to make a case for the member submitting a complaint or request for support*" (Perry, 1981). Zelby (Zelby, 1989) calls for a procedure to protect both parties in the whistleblowing process. Regulators and professional bodies have a role in protecting the general public and to consider the needs of those who work

in their industry. Both must raise awareness to overcome the perception that whistleblowing is an individual act of responsibility and advocate for improved support mechanisms and tactics for software development teams discovering issues through the different stages in the software development lifecycle. Very recently (February 2024) the British Computer Society published a report on AI and professional standards[53] and makes a recommendation that *"Technology professionals should expect strong and supported routes for whistleblowing and escalation when they feel they are being asked to act unethically or, for example, to deploy AI in a way that harms colleagues, customers or society*".

10.3 Software Engineering Practice Aspects

Table 10-1 shows the similarities of existing media stories and software engineering practice aspects of situations found in the case studies. As with stories in my research (Table 9-1), issues are found across the software development lifecycle from poor or unsafe designs through to the misuse or technical issues of production systems. In the Ipsos survey (Section 1.3.1), participants reported on software development lifecycle issues leading to failures in safety, security, and testing practices. A notable absence in the existing literature was technical details of how whistleblowing situations started and evolved. In the Ipsos survey (Miller and Coldicott, 2019) twenty-nine percent of tech workers surveyed had seen harmful situations, and of those ninety percent had taken some action as a result. However the data was not granular enough to support my research question about the software engineering aspects and artefacts of a situation.

Table 10-1 Existing Software Engineering Stories Related to Case Study

Organisation	Software Engineering Aspects	Relates to Case
UK Post Office	Covering up technical issues and documentation	*Case A:* covering up technical issues by changing documentation
Volkswagen	Intentional misuse of software; disguising code behaviour	*Case B*: Misuse of code, labelling of code to hide functionality
Bernie Madoff	Intentional misuse of software; fake trading records	*Case A*: falsifying test reports
GEC Nuclear	Inadequate testing, unsafe designs. Not understanding consequences	*Case C*: quality assurance of computer based nuclear protection
Hughes Aircraft	Failures in testing (falsifying) chips for aircraft, tanks, and missiles	*Case A*: changing documents and falsifying test reports.

Section 9.2 presented a broad range of findings about software engineering artefacts with key findings related to the tampering with software artefacts to remove, disguise, or leave evidence of issues, and the creating of organisational secrets about issues regarding the software or product under development. CERT® refers to "*software lifecycle development interference*" as an insider threat (Cappelli, Moore and Trzeciak, 2015). As my study participants were no longer working at the case study organisations it was not possible to access software engineering artefacts to explore software engineering aspects in more detail and in particular how such organisational secrets are maintained. Background research to Case B looked at the Volkswagen scandal and reflects on developers disguising the true functionality of code. Trope described these solutions as a new insider cyberthreat (Trope and Ressler, 2016). I am interested to

[53] https://www.bcs.org/articles-opinion-and-research/living-with-ai-and-emerging-technologies-meeting-ethical-challenges-through-professional-standards/

know how, as software engineers join and leave projects, the history of the project and issues are maintained. Could future engineers joining an organisation rediscover the organisational secret (Grey and Costa, 2016) , and if so, how would it be dealt with? What are the risks of people leaving an organisation, taking evidence with them, and then disclosing the issue? In the UK, former IT staff from Fujitsu are speaking up about historic concerns with the Horizon system[54] and the extent of the subsequent cover up.

While whistleblowing may be assumed beneficial for society, encouraging it may be inappropriate without understanding factors that increase the likelihood of its effectiveness to change a situation (Near and Miceli, 1995). Understanding how software products involved in harmful situations came to be, relies on specialised technical knowledge and recall of the situation. Both (Kuhn and Stocker, 2012) and (Ogawa and Ma, 2010) find software artefacts trigger memories of experiences, such as revisiting check in comments, source code and photos of development teams. Recommendation 4 sets the groundwork for a rapid response to emerging software situations in specific industry domains, such that the knowledge and recall of a situation by actors involved can be captured using the WISE case study method and run by researchers with knowledge and expertise in that domain. Preparing for and communicating such studies with individuals, groups, and organisations, while challenging to set up could reveal evidence and insight into how such situations evolve.

Recommendation 4: Develop professional networks and collaborations with professional bodies and domain specific industry experts and regulators with which to conduct future research. Obtain consent for access to organisations, software artefacts and actors involved in recent or emerging harmful or whistleblowing situations for research studies.

10.3.1 Software Engineering Background Literature

The background literature searches in Case A (health domain) found clinical studies of critical incidents (Cassidy, Smith and Arnot-Smith, 2011; Welters *et al.*, 2011) where faulty equipment accounted for around a third of all incidents. The studies do not report on the technical detail of failures, the researchers were not software engineers. Cassidy et al. conclude "*It is surely indefensible to continue to use equipment when its safe functioning is not guaranteed"*(Cassidy, Smith and Arnot-Smith, 2011). Alemzadeh et al. (Alemzadeh *et al.*, 2013) studying Food and Drug Administration (FDA) reports, and Fu et al. (Fu *et al.*, 2018) reviewing medical device recalls from the FDA database[55] both find malfunctioning medical devices are a leading causes of serious injury and death in the USA. The publicly accessible Manufacturer and User Facility Device Experience (MAUDE) database contains FDA reports from manufacturers, importers, and user facilities alongside voluntary reporters such as health care professionals, patients, and consumers. The MHRA (UK) website does not currently have such a database of device recalls, instead links to manufacturers' Field Safety Notices (PDFs) are published on webpages or via spreadsheets at weekly and yearly intervals, it was challenging to compile a UK overview of software and hardware device failures from this data.

[54] https://www.theguardian.com/uk-news/2024/jan/14/a-tragedy-is-not-far-away-25-year-old-post-office-memo-predicted-scandal
[55] https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfres/res.cfm

Recommendation 5: A cross-discipline study of software in clinical settings with better access to field safety notices, root causes and actors involved in incidents. This would provide a source of stories to generate multiple cases. The boundary escalation model in the WISE analysis framework would need to be extended to represent the actors and reporting pathways found in the clinical domain.

Wang et al. (Wang, Keil and Wang, 2015) make recommendations for safety critical project managers to provide training and tools to ensure employee sensitivity to consequences and probability of failures in safety critical systems, and that even low-risk concerns should not be ignored. Evidence from Case A suggests it is the manager and compliance officers that required such training.

10.3.2 Technology For Detection or Protection of Whistleblowers

With whistleblowing sometimes described as an insider threat to an organisation (Cappelli, Moore and Trzeciak, 2015), two papers in the literature review present studies developing technical solutions for the detection of whistleblowing activities through email traffic (Okolica, Peterson and Mills, 2006) and information analysis (Maitre *et al.*, 2019). There are also eight papers looking at technology supporting the protection of identity or making anonymous the source of a whistleblowing submission. Although, as Larson reflected, the confidentiality of disclosures may not prevent discovery of whistleblowers by other means (Larson, 1971). Two items critique available whistleblowing platforms ahead of developing or selecting a whistleblowing platform solution (Zakia *et al.*, 2017; Jayakrishnan and Murali, 2019). Bodo (Bodó, 2014) reports on aspects of privacy enhancing technologies, such as Wikileaks, that enable insiders to anonymously share private organisational information. On the detection side, Okolica et al.'s work on the ENRON case (Okolica, Peterson and Mills, 2006) describes using email analysis to identify signals for potential whistleblowers - i.e., people having a hidden interest in sensitive topics. Maitre et al. (Maitre *et al.*, 2019) present a solution for discovering weak signals in web pages and twitter content, to extracting information "*possibly sent by whistleblowers*". To avoid such network detection, an innovative solution called AdLeaks, is provided by Roth et al. (Roth *et al.*, 2013), using online advertising as a cover for submitting and recovering whistleblowing reports. This theme raises issues of how employee activity is monitored internally and externally and by whom, and if actions taken can be identified as activities relating to both covering up of issues or revealing of whistleblowing actions (e.g., seeking evidence, insider threats, regulators or the media looking for evidence and stories).

In the case studies, while the need for anonymity and protection of whistleblowers is presented, using technology to protect (or detect) the actions of those speaking up was not found. In Case A the software engineer chose not to report the situation anonymously to the regulator; post, email and an in person visit to the regulator were the reporting mechanisms used. In Case B the software engineering team having raised their concerns with the team manager believed any anonymous reporting up would have been attributable to them. In Case C the anonymity of Organisation C's whistleblower was maintained by the professional body. The WISE framework looks at to whom a situation is escalated but did not explicitly capture the means by which it is done, however the interview narratives did in most cases capture this. Mechanisms for reporting up situations should be explicitly added into future iterations of the WISE framework.

I reflect on the possibility that technology that can help evidence and support the reporting of software engineering harm and wrongdoing, can also be used against those discovering and speaking up about such issues. The use of software artefacts to evidence or cover up issues was present in the case studies, as was tampering with or suppressing artefacts. In Case B, the artefacts would confirm that the software team implemented and are complicit the fraudulent requirement. However, without knowledge of the issue or access to such artefacts the regulators or other investigators (e.g., compliance auditors) would be unaware of such evidence without having to trawl lines of codes or large design documents to surface it. Pfleeger (Pfleeger, 2016) responds to those calling for greater transparency of software after the Volkswagen scandal though disregards the idea that with open-source code someone could have seen the relevant code segment and blown the whistle earlier. The effort (manual or automated) to scrutinise such systems, made of millions of lines of code, is near impossible and would require specialist skills and knowledge to navigate the codebase. Additionally, in Case B and other automotive literature, there are reports of hiding or obfuscating actual functionality. As AI and machine learning technologies advance, would it be possible to detect obfuscated or tampered with code?

The Post Office public inquiries are beset with challenges seeking evidence of what happened with reports of evidence lost or destroyed. In a legal paper on recommendations for the probity of computer evidence, linked to the Post Office trials, Marshall et al. report that the kind of documents that are likely to exist, and ought to be disclosed are not generally well-understood by people without professional computing or software engineering knowledge. This highlights a procedural dilemma: when a request for a disclosure of a software issue is made, the lack of specific software engineering knowledge to locate and examine the required evidence may result in the dismissal of the request. The Marshall et al. paper proposes a two-step software-issue disclosure process emerging from cases involving the cover-up of harmful software practices and products. Such a solution would support legal challenges would benefit all stakeholders in the software development lifecycle, and in particular help reduce the burden on individuals and organisations seeking to produce evidence in *support or defence* of whistleblowing actions. Stakeholders in the nuclear public inquiry, referred to in Case C, may have found such a solution advantageous.

In the literature review specifically, (Zelby, 1989) calls for a procedure to protect both parties in the process; the whistle-blower in legitimate instances and those on whom the whistle was blown frivolously or maliciously. Papers by Park, Keil and Kim (Park, Keil and Kim, 2008; Keil and Park, 2010) describe basic whistleblowing models and processes that guide their laboratory experiments. Oh and Teo (Oh and Teo, 2010) look at a behavioural reasoning process for whistleblowing and apply it to a software piracy experiment. Tavani and Grodzinsky (Tavani and Grodzinsky, 2014) develop a "*trust and betrayal*" framework to look at the relationship between employee and employer when a whistleblowing situation arises, applied to the analysis of the Edward Snowdon story and the perception of being a hero or a traitor. My research develops and applies a framework for the capture and analysis of in-practice situations, based on primary data from software engineers directly involved in a story. The WISE framework could be used to analyse secondary data sets to identify actors, decisions and actions taken by software engineers. The generated profile wheels (similar to Figure 9-7) showing the themes of that data set. Throughout my research I have gathered many harmful tech stories. Future research could analyse and catalogue the stories, coded up using the

WISE analysis framework. This story bank would be available for researchers, educators, students, and IT professionals.

10.4 Whistleblowing Process

Whistleblowing stories from the media were frequently referenced in the literature review, discussing harm and wrongdoing found in practice. Table 3-5 gave examples of stories found. Regulation breaches, health and safety issues, damage to the environment, and covering up wrongdoing were all aspects found present; safety critical systems were frequently discussed. Whistleblowing is often depicted as a personal act of moral responsibility, and as a heroic one, given the harrowing consequences (Alford, 2007) that whistleblowers may face. Whistleblowing stories have captured media attention in both news, TV series and films. *Mr Bates vs The Post Office* has been watched by over 10 million viewers [56] and is widely referred to as one of the largest miscarriages of justice in British history.

The educational power of telling stories (Preston, 1998) has been understandably harnessed in computing and engineering ethics courses (Bowyer, 1997, 2000; Kevin W Bowyer, 2001; Brinkman, 2009; Kline, 2010; Martin, 2011; McNamara, Smith and Murphy-Hill, 2018). The media attention and teaching practice risks perpetuating the view that whistleblowing is a personal matter and of individual responsibility, whereas research and software engineering practice shows the responsibility for technology systems design, deployment and operations is much wider and distributed (Coeckelbergh, 2012). My case studies present whistleblowing stories from the perspective of an individual involved in the situation and include their reflection on the involvement of others. In Case A and C software experts express no regret for what they did and would do so again. In Case B, a now senior and experienced safety software expert reflected how it was cathartic to revisit stories from their past and reflect on an incident that "*served me well and allowed me to better understand the non-technical nature of challenges faced by engineers"*.

10.4.1 Group Identity

Central to whistleblowing is the choice made by individuals or groups to disclose perceived wrongdoing, with the purpose of mitigating the harm or to rectify the situation. The observation of wrongdoing and deciding to disclose concerns presents several dilemmas relating to the protection of oneself and others. There may be feelings of obligation towards colleagues and the organisation (Park, Keil and Kim, 2008; Wang and Oh, 2011; Wang *et al.*, 2017), but also concerns for harmed parties internally and externally (Smith, Keil and Depledge, 2001; Park and Keil, 2007). Gundlach, Douglas, and Martinko (Gundlach, Douglas and Martinko, 2003) discuss perspectives of power and suggest whistleblowers consider the economic and psychological costs and benefits of taking action. Near and Miceli (Near and Miceli, 1995) report that power may intimidate an employee's decisions to blow the whistle, particularly when accusations are about higher status members of an organisation and the cost of challenging these individuals risks their current and future careers. Anvari (Anvari *et al.*, 2019) guides us to look at the groups an actor identifies with in relation to the situation and how they act within a group (dissent, silence, exit, external reporting). In Case A and B there is

[56] https://uk.news.yahoo.com/mr-bates-vs-post-office-205200132.html?

evidence of group consensus about wrongdoing and that management was not correcting the harmful situation. In Case A, a software engineer acts (dissents then reports externally) on behalf of others in the organisation who have informally raised concerns with them. In Case B, the software engineering team have concerns and initially dissent to implement the feature.

10.4.2 Actor Actions

In the general whistleblowing literature, several long-standing theories and frameworks are widely used to predict factors influencing if an *individual* is more or less likely to whistleblow and report up wrongdoing. As described in Chapter 4, these theories and frameworks inform my WISE analysis framework which looks at the actions and actors in the software engineering domain. The WISE framework is extendable, and during the analysis of the case, data additional actions were added, as captured in Table 4-8, to support the "upholding values and standards" analysis. One key addition was identifying actors tampering with or suppressing software engineering artefacts. This finding is very specific to the software engineering lifecycle and is directly related to "holding onto organisational secrets" within software engineering artefacts. A second addition was actors writing evidence into software engineering artefacts to protect themselves from future inquiries about the source of a harmful requirement. Software engineering experts have mechanisms to leave evidence in technical artefacts without the knowledge of others. The potential asymmetry of knowledge between software engineers and management is a strength and opportunity that could be used to overcome threats from management (Keil *et al.*, 2004).

10.4.3 Holding Organisational Secrets

In Case A at least three actors are aware of the changing and backdating of test reports. In Case B the software engineering team and at least two managers are aware of the wrongdoing in the code. In Case C senior management were definitely aware of the issues raised in the report. Organisational secrets are being kept (Grey and Costa, 2016). In my case studies, management behaviour regarding software engineering experts and external regulators was concerning as it suggests management has developed mechanisms for organisational secret keeping. Teams were found unwilling to speak up about their collective concerns for fear of retaliation or job loss, and so the secret keeping and complicity in a situation continues. These findings draw attention to regulator's accessing evidence regarding the compliance of software systems, products, and processes particularly from those trusted with testing or auditing such systems. In all my cases and stories it could be argued that issues stem from a "*few rogue individuals*", as was Volkswagen's initial defence [57], that later turned into an acknowledgement that whole chain of actors was aware of the code used to defeat emissions tests[58]. Organisational secrets, the detail of which may be evidenced in software artefacts, warrants further study.

[57] https://www.independent.co.uk/news/business/news/volkswagen-emissions-scandal-a-few-rogue-engineers-are-to-blame-says-vw-chief-executive-a6687201.html
[58] https://www.theguardian.com/business/2015/dec/10/volkswagen-emissions-scandal-systematic-failures-hans-dieter-potsch

Recommendation 6: Research into the vulnerabilities and bugs in software is widely studied resulting in tools and mechanisms to test, detect and alert organisations and developers to correct the issues. How can software features, the documenting of which may disguise the true use or purpose, be found in systems that may contain many millions of lines of code?

10.5 Whistleblowing Research and Gaps

Investigating how and why whistleblowing is, or is not, happening in software engineering practice requires carefully designed studies, sympathetic to the complexities and risks associated with studying individuals and organisations involved in such situations. On conducting my software engineering literature review, the studies I anticipated finding were scarce, and where items existed, they were not to a level of granularity with respect to the software engineers, escalations, and the software engineering aspects of situations. This section discusses research and gaps found in the literature review.

The gap analysis in Section 3.6.3 found field studies and laboratory-controlled experiments have contributed to a better understanding of factors affecting the likelihood of whistleblowing in software engineering practice. However, the studies found lacked diversity in research perspectives, and the number of authors involved was small. Most empirical items found in the review were controlled lab-based experiments or neutral survey studies looking at factors that influence a participant's intention to speak up and profiling of characteristics of participants. Case studies were present in the literature review, with varying levels of detail; most report on an incident based on secondary data and just three use primary data sources alongside one autobiographical case study. None followed a detailed case study research method such as that defined by Yin (Yin, 2018) or Runeson et al. (Runeson *et al.*, 2012). The existing literature does contribute to our understanding of human and technical factors that *may* affect whistleblowing in software engineering practice, however these studies are not reported as using whistleblowers as the main actor in the study.

10.5.1 Frequency of Whistleblowing Situations

Whistleblowing is a phenomenon of relatively rare occurrence in the science and technology community (Vandekerckhove, James and West, 2013). There were no studies found attempting to estimate the frequency of whistleblowing situations occurring in software engineering practice. This pushed me to look more widely for *any* technology studies that included data or references to actors taking whistleblowing actions that could help make some estimate as to frequency. One such study by Ipsos (Miller and Coldicott, 2019) in 2019 was introduced at the start of this thesis. While the whistleblowing literature in software engineering has not evolved significantly since this, in late 2023 Reijenga, Aslam and Guzman (Reijenga, Aslam and Guzmán, 2023) published a survey of 147 software professionals regarding their views of whistleblowing and software (Hunt and Ferrario, 2022). The Reijenga et al. survey looked specifically at software related wrongdoing (e.g., privacy or security breaches). Findings from both studies indicate that 29% and 22% respectively of the survey populations had witnessed harm or wrongdoing. The Ipsos study reported that 90% of participants witnessing a harmful situation took some action, from talking to a colleague

(54%), to reporting internally (47%) or to an external body (29%). The Reijenga et al. study found 48% reported taking some action on witnessing a situation. Of the 48% taking some action, 52% informally reported the situation to co-workers or management, 33% reported formally within the organisation and 27% spoke directly to those involved in the wrongdoing. There is insufficient granularity in the data to estimate a frequency of occurrence of situations arising. We can only make crude estimates about the number of people seeing harmful situations (29% and 22% according to (Miller and Coldicott, 2019; Reijenga, Aslam and Guzmán, 2023) respectively), the surveys do not ask how often or how many harmful situations participants have seen.

Understanding the frequency of occurrence of whistleblowing is complex and has implications for the choice of research methods, as there is no guarantee a whistleblowing situation would arise during a study period. The relevancy of frequency is discussed in the next section.

10.5.2 Ethnographic and Action Research

Whistleblowing in software engineering is a complex social-technical phenomenon and its understanding would benefit from in-depth ethnographic studies to understand how situations of public concern arise and how they are then handled. No ethnographic studies were found in the literature review. Whistleblowing is a knowledge and decision-based activity, and as such would be difficult to report on from observations alone. Whistleblowing is also of a sensitive nature, making the design and recruitment of participants to any in-practice research studies are challenging, particularly when much of the process leading to whistleblowing happens under the radar. In addition, situations may take course over a long time, requiring extensive research time and resources that may not be easily available. The seriousness of the consequences of any actions leading to whistleblowing may mean participants and their organisations would adjust behaviours during any observation or action period to avoid or reduce the risk of detection or retaliation. Action research studies were also not found in the literature review. Ethical issues and potential unintended consequences of engaging and potentially encouraging participants to report up issues in an organisation would need to be carefully addressed when tackling such research.

10.5.3 Missing perspectives and interactions

In the literature, whistleblowing stories are often described from the whistleblowers' perspective, with less attention on the involvement and responses from other software engineering practitioners, disclosure recipients (internal or external), organisations, and groups in wider society made aware of these software engineering situations. Reports of high-profile software scandals in the media appear to be increasing and so too the scrutiny of software organisations to show consideration of the social and human impact of the systems they design, build, and operate. In 2021, there have been a succession of whistleblowing stories in the media, with former software engineering practitioners from companies such as Google and Facebook speaking out, with evidence, about the harms caused by features of their products (e.g., algorithmic bias and putting profit before public good). Case A and B did not involve the general public and media. Case C did, using national newspaper to publish a letter regarding future concerns with nuclear safety and the development of safety critical systems more generally.

Recommendation 7: Researching modern emerging software engineering stories, presented as comparable cases, to map out and raise awareness of the growing involvement of the software engineering community, media, and campaign groups to publicise issues and mitigate harm.

10.5.4 Research Implications

To be able to study an organisation before, during and after a harm or wrongdoing incident comes to light is the ideal; to speak with multiple actors with various roles and responsibilities and to collect a data set from multiple perspectives. In reality, even if researchers were present at *just the right time* it would be challenging to reliably gather and analyse data sets, while protecting the safety and confidentiality of actors involved in the situation. (Anvari *et al.*, 2019) highlighted "*just right timing*" and suggests a baseline survey data set at a time T1, to then return to the organisation at time T2 and to then run a study of the current organisation climate and any harm or wrongdoing situations discovered between times T1 and T2.

(Singer and Vinson, 2002) report on the rise in popularity of empirical methods in software engineering research. Surveys, experiments, metrics, case studies, and field studies are examples of empirical methods used to investigate software engineering processes. Major ethical issues identified by Singer and Vinson include informed consent, scientific value, beneficence, and confidentiality. People complicit in the harm or wrongdoing may not wish to reveal what actually happened, may be protecting their or other roles in the situation (as observed in my case studies) and the possible career and legal consequences linked to the situation for themselves, other individuals, and the organisation more generally.

10.6 Summary

The findings from my literature review indicate that while whistleblowing is increasingly mentioned in software engineering literature, it is an under-explored area of software engineering research. The literature review concludes that further empirical research is required to understand how whistleblowing happens in software engineering practice. The gaps in methods and knowledge in whistleblowing research found in the literature led me to develop Objective 2 and 3 and for them to be addressed by designing a framework to conduct case study research based on in-depth interviews with software engineering experts. My research specifically sought software experts that had witnessed or acted in potential whistleblowing situations (the 22% and 29% in the Ipsos and Reijenga et al. surveys). There are extensive initiatives in the research and practitioner community to detect and reduce harm from software engineering practices and systems (e.g., values and fairness of software (Brun and Meliou, 2018; Hussain, Mougouei and Whittle, 2018; Winter *et al.*, 2019) and responsible AI (Barredo Arrieta *et al.*, 2020)). While not directly addressing the challenges of whistleblowing in practice, these initiatives guide practitioners on how to discover and reflect on the consequences (unintended or not) of software systems and the associated implementation practices. The initiatives may help mitigate potential harm and reduce need for whistleblowing; however, mechanisms are still required to raise concerns if a software development team or organisation choose or fail to recognise, reduce, or prevent identifiable harms. An assumption we must make is that there will always be scenarios where, despite best endeavours, harmful situations will arise caused by malicious, negligence, or unintentional acts that could negatively affect an organisation.

Chapter 11. Conclusions

Prior to starting my PhD, I worked as a software engineer, business analyst and IT consultant for over twenty-five years. I frequently read Computer Weekly and followed the emerging Post Office scandal since 2013. In the UK there are criminal investigations against the Post Office and Fujitsu IT staff[59] for cover ups linked to IT systems that left hundreds of Subpostmasters suspended, sacked, or criminally prosecuted for theft, fraud, and false accounting. External whistleblowing from campaign groups and regulatory bodies, after many years, finally brought this story to the public attention. It is not known if internal software engineering teams tried to evidence and speak out about the issues, and if they did, how effective it was. My research suggests software engineering practitioners find it difficult to report issues, and even more so to speak up when management are complicit in the harm. Software engineers report they comply with management orders even though they do not agree with the functionality or directly benefit from it themselves. By complying with management instructions engineers may take on a responsibility that, like Volkswagen engineers, could see them prosecuted and even jailed[60].

My research started by collating a considerable amount of grey literature and news media about whistleblowing linked to software issues (Section 1.4, Section 1.5). This in turn led to a literature review on whistleblowing (Chapter 3), which in turn guided the development of the WISE analysis framework (Chapter 4) and a case study based on in-depth interviews with software engineering experts working in health, transport and energy sectors. Key findings relate to tampering with software artefacts to remove, disguise, or leave evidence of issues, so creating team and organisational secrets. The cases and stories notably evidence issues being suppressed or covered up by management, in breach of regulatory standards and compliance processes. I found practitioners motivated to uphold professional values and standards despite the negative consequences for themselves. Some experienced practitioners seek help from professional and regulatory bodies to mitigate concerns; some less experienced staff keep quiet, raise issues discretely with colleagues, or are threatened into complying with management wrongdoing.

Statista (a global data and business intelligence platform) report there were over sixty-two million people working in the global IT industry in 2023[61], with software engineers, user support specialists and systems analysts being three of the major job roles in this

[59] https://www.theguardian.com/uk-news/2024/jan/05/post-office-criminal-investigation-potential-horizon-accounting-fraud
[60] https://www.bbc.co.uk/news/business-41053740
[61] https://www.statista.com/statistics/1126677/it-employment-worldwide/

domain. Three of the main IT professional bodies are the IEEE[62], the ACM[63] and (in the UK) the BCS[64] who, between them, currently have over six hundred thousand members, which is less than one percent of the global work force. However, IT professionals do not need to be a member of such professional bodies to seek guidance from such bodies. The ACM Code of Ethics (ACM Council, 2018) states, "*a computing professional has the obligation to report any signs of systems risks that may result in harm*" and that "*if leaders do not act [...] it may be necessary to blow the whistle*". The goal of this thesis was to investigate software engineering professionals taking or responding to whistleblowing actions. The WISE framework is currently presented as a research tool. Future work could present it as a practitioner focused resource to raise awareness of what "blowing the whistle" entails in software engineering, and how to assess and decide to proceed with whistleblowing actions that go beyond formal policies and procedures set out by organisations. More widely there are opportunities to work across domains with whistleblowing institutions and charities to develop WISE style frameworks for other domains.

Recommendation 8: The software engineering community must continue working to make software, decisions, and artefacts transparent and auditable, this will help identify harmful situations in a timely fashion to mitigate harmful outcomes.

Understanding software engineering practices and being transparent about points of success and failure may help practitioners "*avoid misguided reporting*" and support them to "*carefully assess relevant aspects of the situation*" before taking on whistleblowing as advised by the ACM Code of Ethics (ACM Council, 2018).

11.1 Research Aims and Objectives

I set out the following three objectives to achieve this goal.

Objective 1: To understand what is currently known about whistleblowing in software engineering literature. I conducted a systematic survey of software engineering literature to understand the current themes and research gaps. I found that whistleblowing is an under-researched area, in particular lacking in field-based studies.

Objective 2: To design a research framework to systematically capture, analyse, and present features of software engineering whistleblowing stories. I explain why case studies were selected and the scope, design, and features of the study. I report on the preparation including the ethics and recruitment of participants. I describe how the collected study data was analysed and presented in a case template and visually in story wheel profiles.

Objective 3: To explore experience of upholding values and standards in software engineering practice, with actions up to and including whistleblowing. Research was conducted in a structured and repeatable way using the WISE framework. The three case study chapters give insight into previously unstudied whistleblowing situations from software engineering practice. Chapter 1 presents a synthesis of the case study

[62] https://www.ieee.org/
[63] https://www.acm.org/
[64] https://www.bcs.org/

findings. Chapter 10 discusses the key findings from existing literature and my case studies from practitioner and research perspectives.

11.2 Research Contributions

My research makes the following contributions to knowledge:

Contribution 1: The first published literature review of how the phenomenon of whistleblowing has been studied in software engineering research. I conclude whistleblowing research in software engineering appears rare (compared to other domains) and lacks field-based research into how whistleblowing happens (or indeed does not happen) in software engineering practice. My case study research confirms findings reported from laboratory studies and calls for further in-practice studies with researchers given timely access to stakeholders and software artefacts linked to emerging whistleblowing situations.

Contribution 2: The WISE analysis framework. An actor-behaviour framework developed to guide the analysis of stories from software engineering practice. The framework allows software engineering whistleblowing stories to systematically be analysed, profiled, and compared. The framework is extendable and could be adapted to include alternate whistleblowing theories, models and industry domains.

Contribution 3: A set of three cases from the health, nuclear and automotive industries reporting on situations and whistleblowing actions taken by software engineering practitioners. The cases cover regulatory breaches, health and safety situations, damage to the environment and the covering up of wrongdoing. Reports of dissent, covering up, keeping silent, escalation and retaliation actions are presented. My thesis increases knowledge about the whistleblowing process and actions of software engineer experts in practice.

Contribution 4: I give new insight into the complex challenges of collecting and analysing whistleblowing data. Firstly presenting the challenges of finding and engaging with participants willing to share their stories. Then the systematic approach of capturing and analysing stories by abstracting actors and their actions and interactions with other actors, enabling stories from different contexts to be compared. I make recommendations for future researchers regarding the adaptation of my WISE analysis framework and how alternate whistleblowing theories and industry domains could be used to develop other explanations of how situations come to be. I propose further in-depth studies of emerging whistleblowing stories to include a wider set of actors and actions.

11.3 Limitations

This section summarises the limitations of this thesis. Sections 3.8 and 5.7 discuss in detail aspects of quality, validity, and mitigations for the literature review, interviews, and case studies.

11.3.1 Construct Validity

Construct validity assesses if appropriate operational measures have been identified for the concepts being studied. Whistleblowing is a very specific term, precisely defining

the escalation of wrongdoing or harmful situations. Terms relating to whistleblowing, whistleblowers and harm are defined up front in Section 1.1 based on government and standard body definitions. Section 2.1 introduces existing whistleblowing definitions and develops a version specific to software engineering practice. In the literature review "whistleblowing" was specific and sufficient for what I wanted to achieve with a literature review and to position my subsequent research. Discussions with interview participants made it clear they were recruited to explore public interest situations in software engineering practice.

11.3.2 Internal Validity

My research develops exploratory cases. Yin says internal validity tests are for *explanatory* or *causal studies* only (Yin, 2018). However, I reflect that in some of my participant stories, there are "rival explanations" for how and why a situation came to be. All case studies contain situations believed by participants to be causing harm or wrongdoing. The UK government states whistleblowers can report situations *they believe* happened, are happening, or are to be happening soon.

The WISE framework action data classification is based on existing social science theories and whistleblowing models. Other theories and models are available that may give alternate classifications for data and findings. The WISE framework and data classifications are created from multiple models and existing studies; the data sets in the framework can be adapted or extended.

11.3.3 External Validity

External validity deals with whether study findings are generalisable beyond the immediate study. A frequent criticism of case study research is that findings cannot easily be generalised. Three cases studies were decided sufficient to explore my research questions within the time and resource constraints of this PhD thesis. My case studies are not intended to be generalisable or to generate theories or whistleblowing prediction models applicable to a wider population of software engineers. Future studies could look to develop more domain-based case studies, such as issues with the development of software in health devices described in Case A, to find generalisable results from within a very specific domain and context.

11.3.4 Reliability

The reliability of the research methods needs to evidence how data collection and analysis can be repeated with the same results, and thus minimise errors and bias in the study. Sections 5.3 and 5.5 describes the procedures for running this study, performing the analysis, and populating the case template and story wheels based on analysis using the WISE framework. I worked alone for this section of the thesis. Future studies would look to work with other researchers to independently analyse the interviews and discuss findings.

11.3.5 Case Outcomes

Reporting on outcomes was a challenging aspect of the cases, there was little insight into changes directly attributable to reporting up a situation. This is a weakness of the study, of working with participants that have left an organisation where the incidents

occurred and with no access to software engineering artefacts relating to the incident. There is an opportunity for future research to look at the impact of whistleblowing incidents or regulation breaches on software engineering practices, products and more generally to the culture of the organisation.

11.4 Future Research

This section discusses future work that could be conducted to investigate the key findings regarding 1) seriousness of issues 2) organisational secrets 3) tampering with software artefact evidence and 4) actor actions and group identity.

11.4.1 Insider Threats to Critical National Infrastructure

Safety critical systems in highly regulated industries have featured frequently in this thesis. Future research focusing specifically on recent whistleblowing stories in critical national infrastructure domains would give more insight into the potentially serious issues raised by actors in these industries, and with a focus on the software engineering aspects of these industries. To develop a case study approach where multiple perspectives of a story will meet the triangulation of data principle for case study research, that is absent in my case studies. To look at factors of information asymmetry and secret keeping with practitioners, their managers, and senior managers. A wider understanding of how expert actors interact would be developed. In particular the different views of the seriousness of such issues and how they should be escalated and resolved. To develop a longitudinal study to look at the ongoing impact and effect of the incident and any subsequent external investigations conducted by regulatory bodies. A Delphi style study (Schmidt *et al.,* 2001) could be conducted where several software experts (including whistleblowers) consider the views of other experts on selected case studies from practice. For these experts to specifically consider what are insider threats to our critical national infrastructure. This method would incorporate the use of the WISE (Chapter 4) and ORID (Section 5.3) frameworks used to design and conduct the expert interviews.

11.4.2 Cross-Disciplinary Case Study

A future literature review, with adjusted research questions and source databases, could look for items reporting on the types of harm and wrongdoing attributed to software engineering products in other domains. An example of how this might look is developed in Case A (Chapter 6), where medical research is cited that looks at the triggers and reporting of critical incidents with ventilators and medical devices and software engineering research studies the technical root causes of incidents. Case study research, conducted by a cross-disciplinary team, would produce more constructive and practitioner insight than that of studies run by medical or software engineering researchers alone.

11.4.3 Practitioner Awareness of Whistleblowing Processes

While whistleblowing can reveal issues and mitigate harm, encouraging such actions is inappropriate without understanding the impact and effectiveness to change a situation. The ACM code of conduct says computing professionals should carefully assess relevant aspects of a situation before blowing the whistle. Future work with practitioners and professional bodies and regulators could develop industry specific case studies and checklists to support IT professionals considering taking whistleblowing actions.

Research suggests that seeing connections between the consequences of software decisions through examples of similar stories may influence and inform future decision making (McNamara, Smith and Murphy-Hill, 2018). Developing and publishing whistleblowing cases studies, to the software engineering community could help support this awareness and connection.

References

ACM Council (2018) *ACM Ethics | The Official Site of the Association for Computing Machinery's Committee on Professional Ethics*. Available at: https://ethics.acm.org/ (Accessed: 7 April 2020).

Adams, A.A. (2014) 'Report of a Debate on Snowden's Actions by ACM members', *ACM SIGCAS Computers and Society*, 44(3), pp. 5–7.

Adamson, G. (2015) 'Ethical Challenges for Future Technologists: The Growing Role of Technology and the Growing Ethical Responsibility of the Technologist', in *2015 IEEE International Symposium on Technology and Society (ISTAS)*. IEEE, pp. 1–6.

Alemzadeh, H. *et al.* (2013) 'Analysis of Safety-critical Computer Failures in Medical Devices', *IEEE Security and Privacy*, 11(4), pp. 14–26.

Alford, C.F. (2007) 'Whistle-blower Narratives: The Experience of Choiceless Choice', *Social Research: An International Quarterly*, *74(1)*, pp. 223–248.

Anvari, F. *et al.* (2019) 'The Social Psychology of Whistleblowing: An Integrated Model', *Organizational Psychology Review*, 9(1), pp. 41–67.

Bagnara, R., Bagnara, A. and Hill, P.M. (2018) 'The MISRA C coding standard and its role in the development and analysis of safety-and security-critical embedded software'. In *International Static Analysis Symposium* (pp. 5-23). Cham: Springer International Publishing.

Barredo Arrieta, A. *et al.* (2020) 'Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges Toward Responsible AI', *Information Fusion*, 58, pp. 82–115.

Beitler, J.R., Malhotra, A. and Thompson, B.T. (2016) 'Ventilator-Induced Lung Injury', *Clinics in Chest Medicine*, 37(4), p. 633-646.

Bell, G.B. (2011) 'Digital Whistleblowing in Restricted Environments', *Journal of digital information*, 12(3).

Bodó, B. (2014) 'Hacktivism 1-2-3: How privacy enhancing technologies change the face of anonymous hacktivism', *Internet Policy Review*, 3(4), pp.1-13.

Boffey, P.M. (1971) 'Nader and the Scientists: A call for responsibility', *Science (American Association for the Advancement of Science)*, pp. 549–551. Available at: https://www.jstor.org/stable/1730939 (Accessed: 27 July 2024).

References

Bowyer, K. (1997) ‘Case study resources for an ethics and computing course’, in *Proceedings Frontiers in Education 1997 27th Annual Conference. Teaching and Learning in an Era of Change*, pp. 469–473.

Bowyer, K.W. (2000) ‘Goodearl and Aldred versus Hughes aircraft: A whistle-blowing case study’, in *Proceedings - Frontiers in Education Conference*. USA: IEEE Computer Society (FIE ’00), p. S2F/2–S2F/7.

Bowyer, K. W. (2001) ‘“Star Wars” revisited - A continuing case study in ethics and safety-critical software’, in *International Symposium on Technology and Society, Proceedings*. USA: IEEE Computer Society (ISTAS ’01), pp. 51–60.

Bowyer, Kevin W (2001) *Whistle Blowing*. Wiley-IEEE Press.

Brinkman, B. (2009) ‘The Heart of a Whistle-Blower: A Corporate Decision-Making Game for Computer Ethics Classes’, in *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery (SIGCSE 09), p. 316-320.

Brun, Y. and Meliou, A. (2018) Software fairness. In *Proceedings of the 2018 26th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering* (pp. 754-759).

Cappelli, D., Moore, A. and Trzeciak, R. (2015) *The Cert Guide to Insider Threats: How to prevent, detect, and respond to Information Technology crimes (Theft, Sabotage, Fraud)*, Addison-Wesley.

Cassidy, C.J., Smith, A. and Arnot-Smith, J. (2011) ‘Critical incident reports concerning anaesthetic equipment: analysis of the UK National Reporting and Learning System (NRLS) data from 2006-2008*’, *Anaesthesia*, 66(10), pp. 879–888.

Charette, R.N. (2005) ‘Why Software Fails’, *IEEE Spectrum*, 42(9), pp. 42–49.

Coeckelbergh, M. (2012) ‘Moral responsibility, technology, and experiences of the tragic: From Kierkegaard to offshore engineering’, *Science and engineering ethics*, 18(1), pp. 35–48.

Cohen, J. (1960) ‘A Coefficient of Agreement for Nominal Scales’, *Educational and Psychological Measurement*, 20(1), pp. 37–46.

Contag, M. *et al.* (2017) ‘How They Did It: An Analysis of Emission Defeat Devices in Modern Automobiles’, in *Proceedings - IEEE Symposium on Security and Privacy*, pp. 231–250.

Cruzes, D.S. *et al.* (2015) ‘Case studies synthesis: a thematic, cross-case, and narrative synthesis worked example’, *Empirical Software Engineering*, 20(6), pp. 1634–1665.

Dozier, J.B. and Miceli, M.P. (1985) ‘Potential predictors of whistle-blowing: A prosocial behavior perspective’, *Academy of management Review*, 10(4), pp. 823–836.

Dyro, J.F. (1988) ‘Meditation on ethics in clinical engineering practice’, *IEEE Engineering in Medicine and Biology Magazine*, 7(2), pp. 77–80.

Easterbrook, S. *et al.* (2008) 'Selecting empirical methods for software engineering research', in *Guide to advanced empirical software engineering*. Springer, pp. 285–311.

Fitzgerald, K. (1990) 'Whistle-blowing: not always a losing game', *IEEE Spectrum*, 27(12), pp. 49–52.

Florman, S.C. (1982) 'Careers: A skeptic views ethics in engineering: Competing with recession and unemployment for people's concern, ethics still manages to arouse lively debates', *IEEE Spectrum*, 19(8), pp. 56–57.

Fu, Z. *et al.* (2018) 'Study of Software-Related Causes in the FDA Medical Device Recalls', in *Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems, ICECCS*. Institute of Electrical and Electronics Engineers Inc., pp. 60–69.

Ghoshroy, S. (2019) 'The Price for Blowing the Whistle when Facing Ethical Dilemmas', in *2019 IEEE International Symposium on Technology and Society (ISTAS)*, pp. 1–6.

Glass, R.L., Vessey, I. and Ramesh, V. (2002) 'Research in software engineering: an analysis of the literature', *Information and Software technology*, 44(8), pp. 491–506.

Grey, C. and Costa, J. (2016) *Secrecy at Work: The Hidden Architecture of Organizational Work*. Stanford University Press.

Gunasekara, G., Adams, A.A. and Murata, K. (2017) Ripples down under: New Zealand youngsters' attitudes and conduct following Snowden. *Journal of Information, Communication and Ethics in Society, 15*(3), pp.297-310.

Gundlach, M., Douglas, S. and Martinko, M. (2003) 'The decision to blow the whistle: A social information processing framework', *The Academy of Management review*, 28(1), pp. 107–123.

Hersh, M.A. (2002) 'Whistleblowers—heroes or traitors? Individual and collective responsibility for ethical behaviour', *Annual reviews in Control*, 26(2), pp. 243–262.

Hintz, A. and Dencik, L. (2016) 'The Politics of Surveillance Policy: UK Regulatory Dynamics after Snowden', *Internet Policy Review*.

Hohenberger, S. *et al.* (2015) 'An overview of ANONIZE: A large-scale anonymous survey system', *IEEE Security & Privacy*, 13(2), pp. 22–29.

Hunt, L. and Ferrario, M.A. (2022) 'A review of how whistleblowing is studied in software engineering, and the implications for research and practice', In *Proceedings of the 2022 ACM/IEEE 44th International Conference on Software Engineering: Software Engineering in Society* (pp. 12-23) pp. 12–23.

Hussain, W., Mougouei, D. and Whittle, J. (2018) 'Integrating Social Values into Software Design Patterns'.

IOS (International Organization for Standardization) (2021) *Whistleblowing Management Systems—Guidelines. ISO 37002:2021*. Available at: https://www.iso.org/standard/65035.html (Accessed: 27 July 2024).

J. P. Near and M. P. Miceli (2016) 'After the wrongdoing: What managers should know about whistleblowing', *Business Horizons*, 59(1), pp. 105–114.

Jaeger, L. and Eckhardt, A. (2018) 'When Colleagues Fail: Examining the Role of Information Security Awareness on Extra-Role Security Behaviors'.

Jalali, S. and Wohlin, C. (2012) 'Systematic literature studies: Database searches vs. backward snowballing', *International Symposium on Empirical Software Engineering and Measurement*, pp. 29–38.

Jarman, A. and Kouzmin, A. (1990) 'Decision pathways from crisis - A contingency-theory simulation heuristic for the Challenger Shuttle disaster (1983-1988)', *Contemporary Crises*, 14(4), pp. 399–433.

Jayakrishnan, H. and Murali, R. (2019) 'A Simple and Robust End-to-End Encryption Architecture for Anonymous and Secure Whistleblowing', in *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pp. 1–6.

Jones, L. (1991) 'Development of Safety Critical Systems and Software for Ministry of Defence', *Safety and Reliability*, 11(2), pp. 6–17.

Jubb, P.B. (1999) 'Whistleblowing: A restrictive definition and interpretation', *Journal of Business Ethics*, 21(1), pp. 77–94.

Kavathatzopoulos, I. *et al.* (2017) 'Snowden's revelations and the attitudes of students at Swedish universities'. *Journal of Information, Communication and Ethics in Society*, *15*(3), pp.247-264.

Keenan, J.P. and McLain, D.L. (2017) 'WHISTLEBLOWING: A CONCEPTUALIZATION AND MODEL'. In *Academy of Management Proceedings* (Vol. 1992, No. 1, pp. 348-352). Briarcliff Manor, NY 10510: Academy of Management.

Keil, M. *et al.* (2004) ''Why Didn't Somebody Tell Me?: Climate, Information Asymmetry, and Bad News About Troubled Projects', *Data Base for Advances in Information Systems*, 35(2), pp. 65–84.

Keil, M., Im, G.P. and Mähring, M. (2007) 'Reporting bad news on software projects: the effects of culturally constituted views of face-saving', *Information Systems Journal*, 17(1), pp. 59–87.

Keil, M. and Park, CW. (2010) 'Bad news reporting on troubled IT projects: Reassessing the mediating role of responsibility in the basic whistleblowing model', *Journal of Systems and Software*, 83(11), pp. 2305–2316.

Keil, M. and Robey, D. (2001) 'Blowing the whistle on troubled software projects', *Communications of the ACM*, 44(4), pp. 87–93.

Kenny, K., Fotaki, M. and Vandekerckhove, W. (2020) 'Whistleblower subjectivities: Organization and passionate attachment', *Organization Studies*, 41(3), pp. 323–343.

Kitchenham, B. (2004) 'Procedure for undertaking systematic reviews', *Computer Science Department, Keele University (TRISE-0401) and National ICT Australia Ltd (0400011T. 1), Joint Technical Report*.

Kline, R.R. (2010) 'Engineering case studies: Bridging micro and macro ethics', *IEEE Technology and Society Magazine*, 29(4), pp. 16–19.

Kuhn, A. and Stocker, M. (2012) 'CodeTimeline: Storytelling with versioning data', *Proceedings - International Conference on Software Engineering*, pp. 1333–1336.

Kumagai, J. (2004) 'The whistle-blower's dilemma', *IEEE Spectrum*, 41(4), pp. 53–55.

Kvale, S. (1994) *InterViews: An introduction to qualitative research interviewing*. Sage Publications, Inc.

Larson, H.T. (1971) 'On whistle blowing', *Computer*, 4(4), p. 34.

Latané, D. and Darley, J. (1970) 'The unresponsive bystander: Why doesn't he help me?' Appleton-Century-Crofts.

Lorey, T., Ralph, P. and Felderer, M. (2022) 'Social Science Theories in Software Engineering Research', *Proceedings - International Conference on Software Engineering*, 2022-May, pp. 1994–2005.

Lutters, W.G. and Seaman, C.B. (2007) 'Revealing actual documentation usage in software maintenance through war stories', *Information and Software Technology*, 49, pp. 576–587.

Maitre, J. *et al.* (2019) 'A meaningful information extraction system for interactive analysis of documents', in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 92–99.

Martin, C.D. (2011) 'Reasoning with ethics', *ACM Inroads*, 2(1), pp. 8–9.

McCubbrey, D.J. and Fukami, C. V (2009) 'ERP at the Colorado department of transportation: The whistle blower's dilemma', *Communications of the Association for Information Systems*, 24(1), p. 7.

McNamara, A., Smith, J. and Murphy-Hill, E. (2018) 'Does ACM's Code of Ethics Change Ethical Decision Making in Software Development?', in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. New York, NY, USA: Association for Computing Machinery (ESEC/FSE 2018), pp. 729–733.

Miceli, M.P., Roach, B.L. and Near, J.P. (1988) 'The Motivations of Anonymous Whistle-Blowers: The Case of Federal Employees', *Public Personnel Management*, 17(3), pp. 281–296.

Miller, C. and Coldicott, R. (2019) 'People, power and technology: The tech workers' view', *Retrieved from Doteveryone website: https://doteveryone. org. uk/report/workersview*.

Near, J.P., Dworkin, T.M. and Miceli, M.P. (1993) 'Explaining the Whistle-Blowing Process: Suggestions from Power Theory and Justice Theory', *Organization Science*, 4(3), pp. 393–411.

Near, J.P. and Miceli, M.P. (1985) 'Organizational dissidence: The case of whistle-blowing', *Journal of business ethics*, 4(1), pp. 1–16.

Near, J.P. and Miceli, M.P. (1995) 'Effective-whistle blowing', *Academy of management review*, 20(3), pp. 679–708.

Near, J.P. and Miceli, M.P. (1996) 'Whistle-blowing: Myth and reality', *Journal of Management*, 22(3), pp. 507–525.

Niu, Y., Stylianou, A.C. and Winter, S.J. (2008) 'Blowing the Whistle on Unethical Information Techmology Practices: The Role of Machiavellianism, Gender and Computer Liteacy', *AMCIS 2008 Proceedings*, p. 269.

Noor, N.R.A.M. and Mansor, N. (2019) 'Exploring the Adaptation of Artificial Intelligence in Whistleblowing Practice of the Internal Auditors in Malaysia', *Procedia Computer Science*, 163, pp. 434–439.

Nursalman, M., Anggraeni, R. and Firdaus, M.Z. (2018) 'Application of Layered Architecture in Whistleblowing Information System for Supporting Good University Governance in Indonesia University of Education', in *2018 International Conference on Information Technology Systems and Innovation (ICITSI)*, pp. 49–54.

Ogawa, M. and Ma, K.L. (2010) 'Software evolution storylines', in *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 35–41.

Oh, L.-B. and Teo, H.-H. (2010) 'To blow or not to blow: An experimental study on the intention to whistleblow on software piracy', *Journal of Organizational Computing and Electronic Commerce*, 20(4), pp. 347–369.

Okolica, J.S., Peterson, G.L. and Mills, R.F. (2006) 'Using Author Topic to detect insider threats from email traffic', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Berlin, Heidelberg: Springer-Verlag (ISI'06), pp. 642–643.

Oxford Economics (2014) *The Cost of Brain Drain A report for Unum*.

P. Bourque and R. E. Fairley (2014) *Guide to the software engineering body of knowledge (SWEBOK), version 3.0*.

Park, C., Im, G. and Keil, M. (2008) 'Overcoming the mum effect in IT project reporting: Impacts of fault responsibility and time urgency', *Journal of the Association for Information Systems*, 9(7), p. 1.

Park, C. and Keil, M. (2007) 'Organizational Factors and Bad News Reporting on Troubled IT Projects', *AMCIS 2007 Proceedings*, p. 92.

Park, C., Keil, M. and Kim, J.W. (2008) 'The effect of IT failure impact and personal morality on IT project reporting behavior', *IEEE Transactions on Engineering Management*, 56(1), pp. 45–60.

Park, P.D. (1996) 'Whistleblowing: an employer's view how vulnerable is your company', *Engineering Management Journal*, 6(4), pp. 183–186.

Perry, T.S. (1981) 'Ethics: Knowing how to blow the whistle: Speaking out about an employer's unethical practices may bring public esteem to the whistle blower, but the many possible pitfalls must be considered', *IEEE Spectrum*, 18(9), pp. 56–61.

Petter, S. (2018) 'If you can't say something nice: Factors contributing to team member silence in distributed software project teams', in *SIGMIS-CPR 2018 - Proceedings of the 2018 ACM SIGMIS Conference on Computers and People Research*. New York, NY, USA: Association for Computing Machinery (SIGMIS-CPR'18), pp. 43–49.

Pfleeger, C.P. (2016) 'Looking into Software Transparency', *IEEE Security and Privacy*, 14(1), pp. 31–36.

Plotkin, S.C. (1989) 'Economic survival and whistle-blowing: One solution', *Conference on A Delicate Balance: Technics, Culture and Consequences, TCAC 1989*, pp. 86–89.

Polkinghorne, D.E. (1995) 'Narrative configuration in qualitative analysis', *International journal of qualitative studies in education*, 8(1), pp. 5–23.

Preston, D. (1998) 'What makes professionals so difficult: An investigation into Professional Ethics teaching', in *Proceedings of the Ethics and Social Impact Component on Shaping Policy in the Information Age, ACM POLICY 1998*. New York, NY, USA: Association for Computing Machinery (ACM POLICY '98), pp. 58–67.

Rehg, M.T. *et al.* (2008) 'Antecedents and outcomes of retaliation against whistleblowers: Gender differences and power relationships', *Organization Science*, 19(2), pp. 221–240.

Reijenga, S., Aslam, K. and Guzmán, E. (2023) 'Whistleblowing in the Software Industry: a Survey', in. Institute of Electrical and Electronics Engineers (IEEE), pp. 1–12.

Ring, T. (2015) 'The enemy within', *Computer Fraud & Security*, 2015(12), pp. 9–14.

Rost, J. and Glass, R.L. (2011) *The dark side of software engineering: evil on computing projects: Whistleblowing Chapter 7*. John Wiley & Sons.

Roth, V. *et al.* (2013) 'A secure submission system for online whistleblowing platforms', in *International Conference on Financial Cryptography and Data Security*, pp. 354–361.

Runeson, P. *et al.* (2012) *Case Study Research in Software Engineering: Guidelines and Examples, Case Study Research in Software Engineering: Guidelines and Examples*.

Runeson, P. and Höst, M. (2009) 'Guidelines for conducting and reporting case study research in software engineering', *Empirical Software Engineering*, 14(2), pp. 131–164.

Sarkinen, J. (2007) 'An open source (d) controller', in *INTELEC 07-29th International Telecommunications Energy Conference*, pp. 761–768.

Schilhavy, R.A.M. and King, R.C. (2010) 'Who Says Professionals Are Ethical? A Cross-sectional Analysis of Ethical Decision Making, Attitudes and Action.', in *AMCIS*, p. 568.

Schmidt, R. *et al.* (2001) 'Identifying software project risks: An international Delphi study', *Journal of Management Information Systems*, 17(4), pp. 5–36.

Shaw, M. (2003) 'Writing good software engineering research papers', in *25th International Conference on Software Engineering, 2003. Proceedings.*, pp. 726–736.

da Silva, J.A.T. and Dobránszki, J. (2019) 'A new dimension in publishing ethics: social media-based ethics-related accusations', *Journal of Information, Communication and Ethics in Society*.

Singer, J. and Vinson, N.G. (2002) 'Ethical issues in empirical studies of software engineering', *IEEE Transactions on Software Engineering*, 28(12), pp. 1171–1180.

Sion, L. *et al.* (2018) 'Risk-based design security analysis', in *Proceedings - International Conference on Software Engineering*. New York, NY, USA: Association for Computing Machinery (SEAD '18), pp. 11–18.

Sloan, T. and Hernandez-Castro, J. (2015) 'Forensic analysis of video steganography tools', *PeerJ Computer Science*, 1, p. e7.

Smith, H.J. and Keil, M. (2003) 'The reluctance to report bad news on troubled software projects: a theoretical model', *Information Systems Journal*, 13(1), pp. 69–95.

Smith, H.J., Keil, M. and Depledge, G. (2001) 'Keeping mum as the project goes under: Toward an explanatory model', *Journal of Management Information Systems*, 18(2), pp. 189–227.

Stanfield, B. (2000) *The art of focused conversation: 100 ways to access group wisdom in the workplace., Toronto: Canadian Institute of Cultural Affairs*. New Society Publishers.

Statista (2021) *Programmers & software developers in the UK 2019 | Statista*. Available at: https://www.statista.com/statistics/318818/numbers-of-programmers-and-software-development-professionals-in-the-uk/ (Accessed: 17 January 2021).

Stein, J. (2013) ‘The end of national security reporting?’, *IEEE Security and Privacy*, 11(4), pp. 64–68.

Stol, K.J. and Fitzgerald, B. (2013) ‘Uncovering theories in software engineering’, *2013 2nd SEMAT Workshop on a General Theory of Software Engineering, GTSE 2013 - Proceedings*, pp. 5–14.

Stol, K.-J. and Fitzgerald, B. (2018) ‘The ABC of software engineering research’, *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 27(3), pp. 1–51.

Swierstra, T. and Jelsma, J. (2006) ‘Responsibility without moralism in technoscientific design practice’, *Science, technology, & human values*, 31(3), pp. 309–332.

Tajfel, H. *et al.* (1979) ‘An integrative theory of intergroup conflict’, *Organizational identity: A reader*, 56, p. 65.

Talbot, D. (2016) ‘Data-mining your psyche: The latest data-driven political pitches target you based on your personality, not just your demographics. But does such profiling work?’, *Technology Review*, 119(3), pp. 88–91.

Taniguchi, N. *et al.* (2005) ‘DECIDE: A scheme for decentralized identity escrow’, in *Proceedings of the ACM Conference on Computer and Communications Security*. New York, NY, USA: Association for Computing Machinery (DIM ’05), pp. 37–45.

Tavani, H.T. and Grodzinsky, F.S. (2014) ‘Trust, Betrayal, and Whistle-Blowing: Reflections on the Edward Snowden Case’, *SIGCAS Comput. Soc.*, 44(3), pp. 8–13.

Trope, R.L. and Ressler, E.K. (2016) ‘Mettle Fatigue: VW’s Single-Point-of-Failure Ethics’, *IEEE Security and Privacy*, 14(1), pp. 12–30. Available at:

Vandekerckhove, W. (2012) *UK Public attitudes to whistleblowing*. London.

Vandekerckhove, W., James, C. and West, F. (2013) *Whistleblowing: the inside story-a study of the experiences of 1,000 whistleblowers*. Public Concern at Work.

Wang, J. *et al.* (2017) ‘Impacts of organizational commitment, interpersonal closeness, and Confucian ethics on willingness to report bad news in software projects’, *Journal of Systems and Software*, 125, pp. 220–233.

Wang, J., Keil, M. and Wang, L. (2015) ‘The effect of moral intensity on it employees’ bad news reporting’, *Journal of Computer Information Systems*, 55(3), pp. 1–10.

Wang, J. and Oh, L.-B. (2011) ‘The Impact Of Relationships And Confucian Ethics On Chinese Employees’ Whistle-Blowing Willingness In Software Projects.’, in *PACIS*, p. 208.

Welters, I.D. *et al.* (2011) ‘Major sources of critical incidents in intensive care’, *Critical Care*, 15(5), pp. 1–8.

References

Whistleblowing for employees: What is a whistleblower - GOV.UK (no date). Available at: https://www.gov.uk/whistleblowing (Accessed: 27 July 2024).

Widder, D.G. *et al.* (2023) 'It's about power: What ethical concerns do software engineers have, and what do they (feel they can) do about them?', in *ACM International Conference Proceeding Series*. Association for Computing Machinery, pp. 467–479.

Winter, E. *et al.* (2019) 'Advancing the study of human values in software engineering', in *Proceedings - 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2019*.

Winter, E., Forshaw, S., Hunt, L. and Ferrario, M.A. (2019) *'*Towards a systematic study of values in SE: Tools for industry and education*'. In 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER) (pp. 61-64). IEEE.*

Yin, R.K. (2018) *Case study research and applications*. 6th edn, *Sage Publications*.

Zakia, I. *et al.* (2017) 'Aspiration and complaint system: From literature survey to implementation', in *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*, pp. 1–6.

Zelby, L.W. (1989) 'Whistle-blowing-'Somebody has to take a stand'', *IEEE Technology and Society Magazine*, 8(3), pp. 4–6.

Appendix A. Footnotes

This appendix summarises the media stories referenced in footnotes. As these media stories are only available online and might at some point in be taken down, this appendix contains the headline and introduction from the link, accessed between 1st and 10th August 2024.

URL ID	URL	Headline and introduction from online content.
1	https://www.gov.uk/whistleblowing	Whistleblowing for employees. What is a whistleblower. Who to tell and what to expect. If you're treated unfairly after whistleblowing.
2	https://spectrum.ieee.org/aerospace/aviation/how-the-boeing-737-max-disaster-looks-to-a-software-developer	"I have been a pilot for 30 years, a software developer for more than 40. I have written extensively about both aviation and software engineering. Now it's time for me to write about both together. The Boeing 737 Max has been in the news because of two crashes, practically back-to-back and involving brand new airplanes. In an industry that relies more than anything on the appearance of total control, total safety, these two crashes pose as close to an existential risk as you can get. Though airliner passenger death rates have fallen over the decades, that achievement is no reason for complacency."
3, 4	https://www.postofficetrial.com/2020/01/horizon-trial-judgment-is-handed-down.html	Horizon trial judgment is handed down. The handing down of the Horizon trial judgment on 16 dec 2019 was expected to be a damp squib. After all, the parties had settled five days previously. But it all went off. You can read my report here. The following is about ensuring the judge's exact comments in court are made available to all as they could be of benefit in any forthcoming legal actions. The most interesting bit came at the end of the hearing. Having handed down the judgment and dealt with housekeeping, Sir Peter Fraser gave the following announcement: "Based on the knowledge that I have gained both from conducting the trial and writing the Horizon Issues judgment, I have very grave concerns regarding the voracity of evidence given by Fujitsu employees to other courts in previous proceedings about the known existence of bugs, errors and defects in the Horizon system. These previous proceedings include the High Court in at least one civil case brought by the Post Office against a sub- postmaster and the Crown Court in a greater number of criminal cases, also brought by the Post Office against sub-postmasters and sub-postmistresses. After very careful consideration, I have therefore decided, in the interests of justice, to send the papers in the case to the Director of Public Prosecutions, Mr Max Hill QC, so he may consider whether the matter to which I have referred should be the subject of any prosecution."

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
5	https://en.wikipedia.org/wiki/Mr_Bates_vs_The_Post_Office	***Mr Bates vs The Post Office*** is a four-part British television drama series for ITV, written by Gwyneth Hughes, directed by James Strong and starring an ensemble cast led by Toby Jones. The series is a dramatization of the British Post Office scandal, a miscarriage of justice in which hundreds of Subpostmasters were wrongly prosecuted (privately and publicly) for theft, false accounting or fraud due to a faulty computer system called Horizon. It was broadcast on four consecutive days from 1 January 2024. The series takes its name from the court case instigated by the former Subpostmaster Alan Bates and others.
6	https://www.theguardian.com/business/2016/sep/09/volkswagen-engineer-pleads-guilty-conspiracy-emissions-scandal-	**Volkswagen engineer pleads guilty to conspiracy in emissions scandal**. Volkswagen engineer James Liang pleaded guilty on Friday to one count of conspiracy in the company's emissions cheating scandal and has agreed to cooperate in the widening criminal investigation. This is the first criminal charge in the justice department's year-long investigation into the company's rigging of federal air pollution tests. VW previously admitted to cheating on US diesel emissions tests for several years after researchers found about 500,000 of its cars contained a software that would reduce emissions while the car was undergoing a government-administered emissions test. When the car was on the road and the cheat device was turned off, the cars emitted up to 40 times the legal limit for nitrogen oxide, according to the Environmental Protection Agency. Such levels can cause respiratory problems. VW agreed to pay $15.3bn to its customers and regulators in July. The settlement did not preclude criminal charges.
7	https://www.theguardian.com/business/2017/dec/06/oliver-schmidt-jailed-volkswagen-emissions-scam-seven-years	**Oliver Schmidt jailed for seven years for Volkswagen emissions scam.** A senior Volkswagen executive was sentenced to seven years in prison by a US court on Wednesday after being found guilty of concealing software used to evade pollution limits on nearly 600,000 diesel vehicles. Oliver Schmidt, a German national who was the general manager in charge of VW's environmental and engineering office in Michigan, had pleaded guilty to his part in the cover-up and argued he was "misused" by VW in its attempts to circumvent US emissions tests. But at the sentencing in Detroit judge Sean Cox sided with the prosecution. "It is my opinion that you are a key conspirator in this scheme to defraud the United States," Cox told Schmidt in court. "You saw this as your opportunity to shine … and climb the corporate ladder at VW." Schmidt read a written statement in court acknowledging his guilt and broke down when discussing his family's sacrifices on his behalf since his arrest in January. "I made bad decisions and for that I am sorry," he said. Alongside the sentence Schmidt was fined $400,000. Both the jail term and the fine were at the top end of sentencing guidelines. Schmidt, who oversaw emissions at VW's office in Michigan from 2012 to early 2015, met with key California regulators in 2015 but did not disclose the rogue software. The government said he later misled US investigators and destroyed documents. Schmidt's lawyers argued that his role only heated up in 2015, years after others at VW hatched the scheme, which violated the Clean Air Act.

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
8	https://www.theguardian.com/business/2009/jun/29/bernard-madoff-sentence	**Bernard Madoff receives maximum 150-year sentence** The disgraced financier Bernie Madoff has been sentenced to the maximum 150 years in prison for masterminding a $65bn (£38bn) fraud that wrecked the lives of thousands of investors. The US district judge Denny Chin described the fraud as "staggering" and said the "breach of trust was massive" and that a message was being sent by the sentence. There had been no letters submitted in support of Madoff's character, he said. Victims in the courtroom clapped as the term was read out. The sentence, which means the 71-year-old fraudster will end his days in prison, was handed down at an emotional hearing in a lower Manhattan courtroom, where victims were given the chance to tell how the fraud had destroyed their livelihoods.
9	https://www.theguardian.com/us-news/2021/apr/14/bernie-madoff-dies-prison-ponzi-scheme	**Bernie Madoff, financier behind largest Ponzi scheme in history, dies in prison.** Bernard Madoff, the one-time Wall Street titan who orchestrated one of the largest frauds in history, has died in prison aged 82. Madoff, known as Bernie, was a former chairman of the Nasdaq stock exchange, and was regarded for years as an investment sage. But unbeknown to his thousands of victims, he was running a Ponzi scheme that wiped out at least $17.5bn in savings. Imposing a 150-year sentence in 2009, judge Denny Chin called Madoff's crimes "extraordinarily evil". His criminal behavior devastated the lives of his victims, leading to suicides, bankruptcies and home losses.
10	https://www.theguardian.com/business/2009/nov/13/madoff-accomplices-jerome-ohara-george-perez	**Madoff's former IT experts arrested over $65bn fraud.** Two computer programmers who worked for fraudster Bernard Madoff were arrested today on charges of helping to cover up his $65bn (£39bn) swindle. The US attorney's office in New York said that Jerome O'Hara and George Perez were arrested at their homes by the FBI. It said the pair formerly worked for Bernard L Madoff Investment Securities on the infamous 17th floor of his Manhattan offices from where he ran his scam. They are charged with conspiracy, falsifying books and records of a broker-dealer, and falsifying books and records of an investment adviser. The securities and exchange commission allege the defendants used their computer skills to produce false documents and trading records to hide the fraud. It is also alleged that they took bribes in return for their silence during their 20-year service for Madoff.
11	https://www.theguardian.com/politics/2020/aug/26/boris-johnson-blames-mutant-algorithm-for-exams-fiasco	**Boris Johnson blames 'mutant algorithm' for exams fiasco.** Boris Johnson got an angry response after telling school pupils that the exam results crisis was caused by a "mutant algorithm" and he was glad it had been "sorted out". The National Education Union (NEU) called the prime minister "brazen" after he appeared to shrug off responsibility for the fiasco. Johnson was giving a speech to students on Wednesday at Castle Rock school in Coalville, Leicestershire. As well as welcoming them back to school and reassuring them it was safe, he said: "I'm afraid your grades were almost derailed by a mutant algorithm, and I know how stressful that must've been for pupils up and down the country." He added: "I'm very, very glad that it has finally been sorted out."

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
12	https://www.theguardian.com/society/2020/aug/24/councils-scrapping-algorithms-benefit-welfare-decisions-concerns-bias	**Councils scrapping use of algorithms in benefit and welfare decisions.** Councils are quietly scrapping the use of computer algorithms in helping to make decisions on benefit claims and other welfare issues, the Guardian has found, as critics call for more transparency on how such tools are being used in public services. It comes as an expert warns the reasons for cancelling programmes among government bodies around the world range from problems in the way the systems work to concerns about bias and other negative effects. Most systems are implemented without consultation with the public, but critics say this must change. The use of artificial intelligence or automated decision-making has come into sharp focus after an algorithm used by the exam regulator Ofqual downgraded almost 40% of the A-level grades assessed by teachers. It culminated in a humiliating government U-turn and the system being scrapped. The fiasco has prompted critics to call for more scrutiny and transparency about the algorithms being used to make decisions related to welfare, immigration, and asylum cases.
13	https://www.nytimes.com/2018/04/04/technology/google-letter-ceo-pentagon-project.html	**'The Business of War': Google Employees Protest Work for the Pentagon.** WASHINGTON — Thousands of Google employees, including dozens of senior engineers, have signed a letter protesting the company's involvement in a Pentagon program that uses artificial intelligence to interpret video imagery and could be used to improve the targeting of drone strikes.The letter, which is circulating inside Google and has garnered more than 3,100 signatures, reflects a culture clash between Silicon Valley and the federal government that is likely to intensify as cutting-edge artificial intelligence is increasingly employed for military purposes. "We believe that Google should not be in the business of war," says the letter, addressed to Sundar Pichai, the company's chief executive. It asks that Google pull out of Project Maven, a Pentagon pilot program, and announce a policy that it will not "ever build warfare technology." That kind of idealistic stance, while certainly not shared by all Google employees, comes naturally to a company whose motto is "Don't be evil," a phrase invoked in the protest letter. But it is distinctly foreign to Washington's massive defense industry and certainly to the Pentagon, where the defense secretary, Jim Mattis, has often said a central goal is to increase the "lethality" of the United States military.
14	https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election	**Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach.** The data analytics firm that worked with Donald Trump's election team and the winning Brexit campaign harvested millions of Facebook profiles of US voters, in one of the tech giant's biggest ever data breaches and used them to build a powerful software program to predict and influence choices at the ballot box. A whistleblower has revealed to the *Observer* how Cambridge Analytica – a company owned by the hedge fund billionaire Robert Mercer and headed at the time by Trump's key adviser Steve Bannon – used personal information taken without authorisation in early 2014 to build a system that could profile individual US voters, in order to target them with personalised political advertisements. Christopher Wylie, who worked with a Cambridge University academic to obtain the data, told the *Observer*: "We exploited Facebook to harvest millions of people's profiles. And built models to exploit what we knew about them and target their inner demons. That was the basis the entire company was built on."

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
15	https://www.ft.com/content/4e9ce18c-6221-11ea-b3f3-fe4680ea68b5	Uber whistleblower Susan Fowler: 'Everything was chaos'. Whistleblowing is such a lonely business that I was prepared for Susan Fowler to be a rather withdrawn, standoffish lunch date. Instead, she arrives grinning, wearing a bright-pink jumper and exclaiming how happy she is to be here. Although we have never met before, she reaches out for a hug hello and immediately starts chatting. Now 28, Fowler was just 25 when she published a blog post about the sexual harassment and discrimination she encountered while working at Uber. The blog was a sensation, read six million times in the first few weeks. Fowler's meticulous, authoritative dissection of the company exposed the rot in tech's shiniest start-up and precipitated the exit of its founder, Travis Kalanick, as chief executive. Fowler was dubbed Silicon Valley's suffragette. She appeared to be the rare whistleblower able to speak out without retaliation. In reality, her life became tumultuous as soon as she pressed publish. Her email and social media accounts were hacked. Friends and family were contacted by people she suspects were private investigators and she was followed to and from work by strangers. While she was being featured on the cover of Time magazine and hailed as the FT's person of the year in 2017, she was growing more isolated and afraid, unable to sleep at night.
16	https://www.theguardian.com/world/2020/mar/01/susan-fowler-uber-whistleblower-interview-travis-kalanick	**Susan Fowler: 'When the time came to blow the whistle on Uber, I was ready'**. Before Susan Fowler was a whistleblower she was a violinist, and before she was a violinist she fed fruit flies to spiders that were milked for their venom at a small Arizona business known as Spider Pharm. In February 2017, Fowler was thrown into the public eye after she published a damning blogpost exposing the toxic sexism she experienced working as a software engineer at Uber. And in her new memoir, *Whistleblower*, she explains how she came to shake up one of the world's most valuable startups.
17	https://www.theguardian.com/technology/2021/oct/08/tech-whistleblowers-facebook-frances-haugen-amazon-google-pinterest	'Welcome to the party': five past tech whistleblowers on the pitfalls of speaking out. When Frances Haugen revealed she was the Facebook whistleblower who supplied internal documents to Congress and the Wall Street Journal, she joined a growing list of current and former Silicon Valley employees who've come forward to call out military contracts, racism, sexism, contributions to climate crisis, pay disparities and more in the industry. In the past days, the Guardian spoke with five former employees of Amazon, Google, and Pinterest who've spoken out about their companies' policies. The conversations revealed Haugen's experience has been singular in some respects. Few of them received the international praise bestowed upon her. Some of them said they have faced termination, retaliation, harassment and prolonged litigation.
18	https://www.theguardian.com/news/audio/2022/jul/12/the-uber-files-the-whistleblower-part-2-podcast	The Uber files: the whistleblower (part 2) - podcast.

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
19	https://www.theguardian.com/news/audio/2022/jul/11/the-uber-files-the-unicorn-part-1	The Uber files: the unicorn (part 1) - podcast
20	https://www.theguardian.com/technology/2017/nov/21/uber-data-hack-cyber-attack	**Uber concealed massive hack that exposed data of 57m users and drivers.** Uber concealed a massive global breach of the personal information of 57 million customers and drivers in October 2016, failing to notify the individuals and regulators, the company acknowledged on Tuesday. Uber also confirmed it had paid the hackers responsible $100,000 to delete the data and keep the breach quiet, which was first reported by Bloomberg. “None of this should have happened, and I will not make excuses for it,” Uber’s chief executive, Dara Khosrowshahi, said in a statement acknowledging the breach and cover-up. “While I can’t erase the past, I can commit on behalf of every Uber employee that we will learn from our mistakes.”
21	https://nypost.com/2023/01/28/facebook-fires-worker-who-refused-to-do-negative-testing-awsuit/	**Facebook secretly killed users batteries, worker claims in lawsuit.** Facebook can secretly drain its users’ cellphone batteries, a former employee contends in a lawsuit. The practice, known as “negative testing,” allows tech companies to “surreptitiously” run down someone’s mobile juice in the name of testing features or issues such as how fast their app runs or how an image might load, according to data scientist George Hayward. “I said to the manager, ‘This can harm somebody,’ and she said by harming a few we can help the greater masses,” said Hayward, 33, who claims in a Manhattan Federal Court lawsuit that he was fired in November for refusing to participate in negative testing. Hayward worked on Facebook’s Messenger app, which allows users to send written messages or make phone or video calls — and is a crucial communication tool in many countries, he said.
22	https://www.england.nhs.uk/ourwork/freedom-to-speak-up/	**Freedom to Speak Up.** NHS England aims to ensure everyone working within the NHS feels safe and confident to speak up. We encourage our NHS leaders to take the opportunity to learn and improve from those who speak up. We want everyone working in the NHS to feel safe and confident to speak up and all NHS leaders to welcome this opportunity to learn and improve. We seek to improve the quality of speaking up arrangements across the NHS in a number of ways. Firstly, we evaluate concerns raised by people working within the NHS about the way NHS organisations operate; their cultures and the quality of care they provide. Secondly, we provide a scheme for people that require support after they have spoken up. Finally, we use staff experiences; learning from the handling of speak up matters and best practice to form the basis of policy, guidance and resources. These further support leadership teams to improve operational arrangements around Freedom to Speak Up.
23	https://www.bbc.co.uk/news/uk-england-suffolk-59707114	**Chairwoman quits over West Suffolk Hospital whistleblower handling.** The chairwoman of an NHS trust criticised for asking staff for fingerprints as it hunted a whistleblower has announced she is stepping down. Sheila Childerhouse said she took personal responsibility for the failings at West Suffolk Hospital. Last week a report highlighted how staff were also targeted when they spoke out over a drug-taking colleague.

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
24	https://www.undocs.org/A/70/361	**United Nations: Promotion and protection of the right to freedom of opinion and expression.** In the report, submitted in accordance with Human Rights Council resolution 25/2, the Special Rapporteur on the promotion and protection of the right to freedom of opinion and expression addresses the protection of sources of information and whistle-blowers. Everyone enjoys the right to access to information, an essential tool for the public's participation in political affairs, democratic governance and accountability. In many situations, sources of information and whistle-blowers make access to information possible, for which they deserve the strongest protection in law and in practice. Drawing on international and national law and practice, the Special Rapporteur highlights the key elements of a framework for the protection of sources and whistle-blowers.
25	https://www.iso.org/news/ref2703.html	**International Standards Organization: Beating bribery and corruption.** Good governance in any organization involves demonstrating accountability and fostering a "speak up" culture. To address the importance of having a secure and effective way that employees can report concerns about wrongdoing, a new ISO standard for whistleblowing has just been published. ISO 37002, Whistleblowing management systems – Guidelines, provides guidance for implementing, managing, evaluating, maintaining and improving a robust and effective management system for whistleblowing. It is non-sector-specific and can be used by organizations of all sizes, including SMEs, as well as those with international operations. Following the three principles of trust, impartiality and protection, the standard covers the identification and reporting of such concerns and how they are assessed and addressed. Its use will not only minimize or prevent potential losses but also ensure compliance with organizational policies and legal and social obligations.
26	https://www.cqc.org.uk/news/stories/quick-guide-raising-concern-about-your-workplace	**Care Quality Commission. Quick guide to raising a concern about your workplace.** Have you got a concern about something you've seen or experienced at your place of work? We have published a quick guide on whistleblowing to help you decide what to do next and how you can tell us about it. Our quick guide has been written for health and care professionals that need to raise a concern about their workplace. It gives you helpful advice on speaking out about poor care and what protection you will have from the law if you do.
27	https://www.consilium.europa.eu/en/press/press-releases/2019/10/07/better-protection-of-whistle-blowers-new-eu-wide-rules-to-kick-in-in-2021/	**European Council. Better protection of whistle-blowers: new EU-wide rules to kick in in 2021**. The EU is to guarantee a high-level protection to whistle-blowers across a wide range of sectors including public procurement, financial services, money laundering, product and transport safety, nuclear safety, public health, consumer and data protection. Today the Council formally adopted new rules on whistle-blowers protection. The new rules will require the creation of safe channels for reporting both within an organisation - private or public - and to public authorities. It will also provide a high level of protection to whistle-blowers against retaliation and require national authorities to adequately inform citizens and train public officials on how to deal with whistle-blowing. The legislation will now be formally signed and published in the Official journal. Member states will have two years to transpose the new rules into their national law.

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
28	https://www.gov.uk/government/publications/the-public-interest-disclosure-act/the-public-interest-disclosure-act	**Guidance: The Public Interest Disclosure Act Published 1 May 2013.** The Act protects workers from detrimental treatment or victimisation from their employer if, in the public interest, they blow the whistle on wrongdoing. The Act protects most workers in the public, private and voluntary sectors. The Act does not apply to genuinely self-employed professionals (other than in the NHS), voluntary workers (including charity trustees and charity volunteers) or the intelligence services.
29	https://theconversation.com/where-were-the-whistleblowers-in-the-volkswagen-emissions-scandal-48249	**Where were the whistleblowers in the Volkswagen emissions scandal?** The "defeat device" used by Volkswagen to cheat emissions testing in its diesel vehicles may be history's most costly software-related blunder. But why did nobody in the German car giant speak out when questions were raised over how it intended to use the engine management software in some of its engines? As the Notice of Violation from the the United States Environmental Protection Agency (EPA) explains, the software in Volkswagen's EA189 diesel engines detected the precise conditions that indicated when a government emissions test was being run. Then, and only then, did the control software fully enable the anti-pollution devices fitted to the vehicle. At all other times, the "road calibration" resulted in nitrogen oxide emissions up to 35 times higher than permitted by the US standard.
30	https://www.computerweekly.com/news/366539133/Medical-regulator-drops-probe-into-NHS-whistleblower-Peter-Duffy-amid-dispute-over-email-evidence	Medical regulator drops probe into NHS whistleblower Peter Duffy amid dispute over email evidence. The UK's chief regulator for doctors has dropped an investigation into an NHS whistleblower who exposed hundreds of cases of harm at a hospital trust in north-west England, following a dispute over the authenticity of emails put forward as evidence by the trust. Consultant urologist Peter Duffy, 61, has faced disciplinary proceedings at the General Medical Council (GMC), which could have led to him being barred from practice, for over two years. The case centred on the contents of two disputed emails that Morecambe Bay Trust (UHMBT) produced as evidence a number of years after Duffy blew the whistle on patient safety at the trust.
31	https://www.theguardian.com/uk-news/video/2018/mar/17/cambridge-analytica-whistleblower-we-spent-1m-harvesting-millions-of-facebook-profiles-video	Cambridge Analytica whistleblower: 'We spent $1m harvesting millions of Facebook profiles' – video

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
32	https://www.nationalarchives.gov.uk/	Uk Government, National Archives
33	www.linkedin.com	LinkedIn
34	https://www.lexisnexis.com/uk/legal/	LexisNexis - Legal
35	https://www.parliament.uk/	UK Government - Parliament
36	https://hansard.parliament.uk/	UK Government – Hansard
37, 43	https://www.perforce.com/products/helix-qac	**Helix QAC: Best Static Code Analyzer for Functional Safety and Standards Compliance**. For over 30 years, Helix QAC has been the trusted static code analyzer for C and C++ programming languages. With its depth and accuracy of analysis, Helix QAC has been the preferred static code analyzer in tightly regulated and safety-critical industries that need to meet rigorous compliance requirements. Often, this involves verifying compliance with coding standards — such as MISRA and AUTOSAR — and functional safety standards, such as ISO 26262. Helix QAC is certified for functional safety compliance by TÜV-SÜD, including IEC 61508, ISO 26262, EN 50128, IEC 60880, and IEC 62304. In addition, it is also certified in ISO 9001 | TickIT plus Foundation Level, which is one of the most widely adopted standards to ensure that your requirements are not only met but exceeded as well. Trust Helix QAC as the best static code analyzer C for static code analysis C++.
38	https://www.fda.gov/medical-devices	**US Food and Drug Administration: Medical Devices**
39	https://www.gov.uk/government/organisations/medicines-and-healthcare-products-regulatory-agency	**Medicines & Healthcare products, Regulatory Agency.** The Medicines and Healthcare products Regulatory Agency regulates medicines, medical devices and blood components for transfusion in the UK

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
40	https://www.fda.gov/medical-devices/device-approvals-denials-and-clearances/510k-clearances	**510(k) Clearances.** Section 510(k) of the Food, Drug and Cosmetic Act requires device manufacturers who must register, to notify FDA of their intent to market a medical device at least 90 days in advance. This is known as Premarket Notification - also called PMN or 510(k). This allows FDA to determine whether the device is equivalent to a device already placed into one of the three classification categories. Thus, "new" devices (not in commercial distribution prior to May 28, 1976) that have not been classified can be properly identified. Specifically, medical device manufacturers are required to submit a premarket notification if they intend to introduce a device into commercial distribution for the first time or reintroduce a device that will be significantly changed or modified to the extent that its safety or effectiveness could be affected. Such change or modification could relate to the design, material, chemical composition, energy source, manufacturing process, or indications for use.
41	https://www.fda.gov/regulatory-information/search-fda-guidance-documents/deciding-when-submit-510k-software-change-existing-device	**Deciding When to Submit a 510(k) for a Software Change to an Existing Device.** This guidance will assist industry and Agency staff in determining when a software (including firmware) change to a medical device may require a manufacturer to submit and obtain FDA clearance of a new premarket notification (510(k)).
42	https://www.iso.org/standard/38421.html	**IEC 62304:2006 Medical device software — Software life cycle processes.** Defines the life cycle requirements for medical device software. The set of processes, activities, and tasks described in this standard establishes a common framework for medical device software life cycle processes.
43	https://www.perforce.com/products/helix-qac	**Helix QAC: Best Static Code Analyzer for Functional Safety and Standards Compliance**. For over 30 years, Helix QAC has been the trusted static code analyzer for C and C++ programming languages. With its depth and accuracy of analysis, Helix QAC has been the preferred static code analyzer in tightly regulated and safety-critical industries that need to meet rigorous compliance requirements. Often, this involves verifying compliance with coding standards — such as MISRA and AUTOSAR — and functional safety standards, such as ISO 26262. Helix QAC is certified for functional safety compliance by TÜV-SÜD, including IEC 61508, ISO 26262, EN 50128, IEC 60880, and IEC 62304. In addition, it is also certified in ISO 9001 | TickIT plus Foundation Level, which is one of the most widely adopted standards to ensure that your requirements are not only met but exceeded as well. Trust Helix QAC as the best static code analyzer C for static code analysis C++.

URL ID	URL	Headline and introduction from online content.
44	https://www.gov.uk/government/consultations/improving-air-quality-reducing-nitrogen-dioxide-in-our-towns-and-cities	**Consultation outcome - Improving air quality: reducing nitrogen dioxide in our towns and cities**
45	https://www.ft.com/content/e05baf80-c57f-11e3-a7d4-00144feabdc0	**Carmakers lobby to delay EU efforts to upgrade emission testing.** The European car industry is lobbying to delay the introduction of a tough new emissions testing regime designed to combat fears that carmakers are gaming the system to boost their efficiency ratings. European cars are as much as 30 per cent less efficient than their manufacturers claim, according to the International Council on Clean Transportation, as carmakers take advantage of an archaic testing system that they want to keep in place for at least seven years. The EU has said it wants to replace the 1970s test with a much more stringent regime by 2017. The new test would remove the loopholes that carmakers currently use to boost their efficiency ratings, including disconnecting the battery to stop it charging during the test, overinflating the tyres to reduce resistance and performing the test at optimum temperatures.
46	https://www.gov.uk/government/news/lorry-emissions-checks-to-start-at-the-roadside	**News story - Emissions cheat devices to be included in roadside checks of lorries**
47	https://www.dw.com/en/bosch-pays-90-million-euro-fine-over-diesel-scandal/a-48843405	**Bosch pays 90-million-euro fine over diesel scandal.** The penalty may be significantly less than the ones handed out to Volkswagen, Audi and Porsche, but auto parts supplier Bosch has become the latest big-name casualty of the "Dieselgate" scandal. German auto parts supplier Bosch was on Thursday ordered by prosecutors to pay a fine of €90 million ($100 million) over its role supplying components in the "Dieselgate" emissions cheating scandal. Stuttgart investigators "levied a fine against Robert Bosch GmbH for negligently infringing its quality control obligations," they said in a statement, adding that the company had agreed not to contest the fine. Beginning in 2008, Bosch "delivered around 17 million motor control and mixture control devices to various domestic and foreign manufacturers, some of whose software contained illegal strategies," the prosecutors found. "Cars fitted with the devices emitted more nitrogen oxides than allowed under regulations."

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
48	https://www.bbc.co.uk/news/business-38603723	**VW papers shed light on emissions scandal.** "Volkswagen obfuscated, they denied, and they ultimately lied." These were the words of the US Attorney General Loretta Lynch, as she set out how the German carmaker would be punished for attempting to hoodwink the US authorities over the emissions produced by its diesel cars. It has been a tough week for Volkswagen. It has been fined $4.3bn, agreed to plead guilty to criminal charges - and six executives are facing charges. One of them, Oliver Schmidt, has spent the past few days in a Miami jail. Others may yet find themselves in the firing line. But because of this, we now have a very clear idea not only of what Volkswagen was doing wrong, and how it went about it, but also the measures that were taken to conceal that wrongdoing.
49	https://www.bbc.co.uk/news/business-41053740	**VW engineer jailed for emissions scandal.** A former Volkswagen engineer who helped develop a device that enabled cars to evade US pollution rules has been sentenced to more than three years in prison and ordered to pay $200,000. James Liang, 63, was the first person prosecuted in the emissions scandal. The US investigation has led to charges against seven others in the US and sparked probes in other countries. Volkswagen has admitted guilt, agreeing to spend as much as $25bn to address US claims. Liang co-operated with prosecutors, who argued that his help with the investigation warranted a reduction in the possible punishment to three years in prison and a $20,000 fine. But US District Court Judge Sean Cox opted for a harsher penalty of 40 months and a $200,000 penalty, saying he wanted to send a message to others in the car industry. "This is a very serious and troubling crime against our economic system," he said.
50	https://www.theguardian.com/technology/2021/jan/04/more-than-200-us-google-employees-form-union	**More than 200 US Google employees form a workers' union.** More than 200 Google employees in the United States have formed a workers' union, the first group at a big tech company to do so as the industry faces a reckoning over years of unchecked power. The elected leaders of the Alphabet Workers Union announced the organization in a New York Times opinion piece on Monday, saying they aimed to ensure employees work at a fair wage, without fear of abuse, retaliation or discrimination.
51	https://www.wired.co.uk/article/united-tech-and-allied-workers-union	**Google and Microsoft staff set to join the UK's first tech trade union.** A group of workers have launched the UK's first union branch dedicated exclusively to the tech sector to tackle issues ranging from working conditions to racial injustice and the climate crisis. The United Tech and Allied Workers (UTAW), a branch of the Communications Workers Union, plans to recruit tech and digital workers, as well as non-tech workers employed by tech companies. "The need for trade union organising is as acute in tech as anywhere else. Workers have seen through the bubble of ping pong tables, free t-shirts and desk beers," says John Chadfield, one of the founding members of the branch. He expects the branch to recruit at least one hundred members in the first two months. A spokesperson for the branch says that workers from companies including Google, Microsoft, ASOS, Monzo and Deliveroo are planning to join.

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
52	https://www.theguardian.com/business/2024/jan/16/fujitsu-it-workers-hmrc-systems-to-go-on-strike	**Fujitsu IT support workers who protect HMRC systems to go on strike.** Staff at Fujitsu, the technology firm at the centre of the Post Office Horizon scandal, are due to go on strike tomorrow, which their trade union said could disrupt HM Revenue and Customs at the busiest time of year for tax collection. About 300 staff, most of whom work in IT support for HMRC at sites in Telford and Stratford, east London, will go on strike in protest at a pay offer that the Public and Commercial Services union (PCS) said was 10 times less than what Fujitsu is offering staff in Japan.
53	https://www.bcs.org/articles-opinion-and-research/living-with-ai-and-emerging-technologies-meeting-ethical-challenges-through-professional-standards/	**Living with AI and emerging technologies: Meeting ethical challenges through professional standards.** With the rapid development and growing demand for the use of Artificial Intelligence (AI) there are new and serious challenges for the ethical use of these systems. The issues are currently most visible in emerging generative AI platforms (used for creating content such as text and images). They include misinformation, intellectual property and plagiarism, bias in training data, and the ability to influence and manipulate public opinion (for example during elections). The Post Office Horizon IT Scandal has highlighted the vital importance of independent standards of professionalism and ethics in the application, development and deployment of technology.
54	https://www.theguardian.com/uk-news/2024/jan/14/a-tragedy-is-not-far-away-25-year-old-post-office-memo-predicted-scandal	**'A tragedy is not far away': 25-year-old Post Office memo predicted scandal.** In any big scandal with the power to dominate the nation's attention, there are inevitably key moments when events could have been stopped in their tracks. Yet few early warnings could have been as prescient as a seven-page memo handed to a Post Office official 25 years ago. During a fractious meeting at Newcastle rugby club in 1999, the note set out a litany of concerns from subpostmasters in the north-east of England who had been piloting the now infamous Horizon accounting system. The issues, including with balancing their accounts, were causing stress and forcing some to work well into the night. Soon after those concerns were raised, subpostmasters gathered again to discuss the potential severity of the problems. "The difficulties and trauma being experienced by some subpostmasters were giving rise to concerns for their health and emotional wellbeing," the meeting was told. "It was felt by some that a tragedy was not far away if something was not altered soon. The software was considered to be poor quality and not intended to run such a huge network."

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
55	https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfres/res.cfm	**Medical Device Recalls:** This database contains Medical Device Recalls classified since November 2002. Since January 2017, it may also include correction or removal actions initiated by a firm prior to review by the FDA. The status is updated if the FDA identifies a violation and classifies the action as a recall and again when the recall is terminated. FDA recall classification may occur after the firm recalling the medical device product conducts and communicates with its customers about the recall. Therefore, the recall information posting date ("create date") indicates the date FDA classified the recall, it does not necessarily mean that the recall is new.
56	https://uk.news.yahoo.com/mr-bates-vs-post-office-205200132.html?	**Mr Bates vs The Post Office confirms record-breaking viewing figures.** Mr Bates vs The Post Office has confirmed record-breaking viewing figures for ITV. According to figures released by ITV, the show, which tells the story of the British Post Office scandal, is officially the most watched programme so far in 2024 and it is ITV's biggest new drama in over a decade – even beating Downton Abbey's 2010 launch. The first episode of the four-part drama has been watched by 10.9m viewers, while the series has now averaged 9.8m viewers across all four episodes, with the first three episodes having been watched by over 10.6m viewers. All four episodes currently stand as the most watched programme on any channel so far this year and the series, including the documentary, has garnered 16.6 million streams.
57	https://www.independent.co.uk/news/business/news/volkswagen-emissions-scandal-a-few-rogue-engineers-are-to-blame-says-vw-chief-executive-a6687201.html	**Volkswagen emissions scandal: Few rogue engineers are to blame, says VW chief executive.** Volkwagen's exhaust emissions scandal, which is set to cost the world's biggest car maker billions of pounds, was caused by "a couple of software engineers" who did it for unknown reasons, Michael Horn, VW's US chief executive has claimed. Testifying before a congressional committee in Washington, DC, Mr Horn apologised and promised a full investigation. "This was not a corporate decision," he said, but something done by German software engineers. "I agree it's hard to believe," he admitted when challenged by sceptical committee members. The claims came as German investigators raided the VW headquarters in Wolfsburg in search of evidence to clarify who was responsible for the cheating.
58	https://www.theguardian.com/business/2015/dec/10/volkswagen-emissions-scandal-systematic-failures-hans-dieter-potsch	**VW admits emissions scandal was caused by 'whole chain' of failures.** Volkswagen has admitted for the first time that the diesel emissions scandal was the result of a collection of failures within the company, rather than just the actions of rogue engineers. Hans Dieter Pötsch, the VW chairman, said there had been a "whole chain" of errors at the German carmaker and there was a mindset within the company that tolerated rule-breaking. VW provided the most detailed explanation so far about how the diesel emissions scandal occurred at a press conference in Germany. Pötsch said engineers had installed defeat devices in engines after realising they could not hit emissions targets for diesel cars in the US by "permissible means". Nine managers have been suspended over possible involvement in the scandal. Although Pötsch said no senior executives were believed to have been actively involved in cheating emissions tests, he warned: "This is not only about direct but overall responsibility."

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
59	https://www.theguardian.com/uk-news/2024/jan/05/post-office-criminal-investigation-potential-horizon-accounting-fraud	**Post Office under criminal investigation for potential fraud over Horizon scandal.** The Post Office is under criminal investigation over "potential fraud offences" committed during the Horizon scandal, the Metropolitan police have confirmed for the first time. Officers are "investigating potential fraud offences arising out of these prosecutions", for example "monies recovered from sub-postmasters [operators] as a result of prosecutions or civil actions", Scotland Yard said on Friday evening. It is not clear whether the investigation relates to individual staff members or the Post Office as a corporate entity. Between 1999 and 2015, more than 700 post office branch managers were wrongly handed criminal convictions after faulty Horizon accounting software made it appear as though money was missing from their outlets.
60	https://www.bbc.co.uk/news/business-41053740	**VW engineer jailed for emissions scandal.** A former Volkswagen engineer who helped develop a device that enabled cars to evade US pollution rules has been sentenced to more than three years in prison and ordered to pay $200,000. James Liang, 63, was the first person prosecuted in the emissions scandal. The US investigation has led to charges against seven others in the US and sparked probes in other countries. Volkswagen has admitted guilt, agreeing to spend as much as $25bn to address US claims. Liang co-operated with prosecutors, who argued that his help with the investigation warranted a reduction in the possible punishment to three years in prison and a $20,000 fine. But US District Court Judge Sean Cox opted for a harsher penalty of 40 months and a $200,000 penalty, saying he wanted to send a message to others in the car industry. "This is a very serious and troubling crime against our economic system," he said.
61	https://www.statista.com/statistics/1126677/it-employment-worldwide/	**Full-time employment in the information and communication technology (ICT) industry worldwide in 2019, 2020 and 2023(in millions)** The worldwide full-time employment in the ICT sector is projected to reach 55.3 million in 2020 (pre-corona estimation), an increase of 3.9 percent over 2019. Software developer/engineer, user support specialist and systems analyst are three major job roles in the ICT industry.
62	https://www.ieee.org/	IEEE and its members inspire a global community to innovate for a better tomorrow through highly cited publications, conferences, technology standards, and professional and educational activities. IEEE is the trusted "voice" for engineering, computing, and technology information around the globe.
63	https://www.acm.org/	ACM offers the resources, access and tools to invent the future. No one has a larger global network of professional peers. No one has more exclusive content. No one presents more forward-looking events. Or confers more prestigious awards. Or provides a more comprehensive learning center.

Appendix A Footnotes

URL ID	URL	Headline and introduction from online content.
64	https://www.bcs.org/	We're over 70,000 members in 150 countries, and a wider community of business leaders, educators, practitioners and policy-makers all committed to our mission. As a charity with a royal charter, our agenda is to lead the IT industry through its ethical challenges, to support the people who work in the industry, and to make IT good for society.

Appendix B. Ethics Approval

Ethics Approval FST19079 (Interviews)

Ethics approval FSTREC ref: FST19079

Dear Lucy,

Thank you for submitting your research ethics application for the project '**Values in Computing**' for review. The application has been reviewed by members of the FST Research Ethics Committee and I can confirm that approval has been granted for this project.

As principal investigator your responsibilities include:

- ensuring that (where applicable) all the necessary legal and regulatory requirements in order to conduct the research are met, and the necessary licenses and approvals have been obtained;
- reporting any ethics-related issues that occur during the course of the research or arising from the research (e.g. unforeseen ethical issues, complaints about the conduct of the research, adverse reactions such as extreme distress) to the Research Ethics Officer;
- submitting details of proposed substantive amendments to the protocol to the Research Ethics Officer for approval.

Please contact the Research Ethics Officer, Becky Case (fst-ethics@lancaster.ac.uk 01542 593987) if you have any queries or require further information.

Kind regards,

Becky Case

Becky Case
Research Ethics Officer – FST & FHM
A34, Faculty of Science and Technology, Mon, Tues & Thurs
Bowland Main, Research Services, Weds
B14, Faculty of Health and Medicine, Fri
Lancaster University
Lancaster
LA1 4YR
Tel: 01524 (5)93987
E-mail: fhmresearchsupport@lancaster.ac.uk, fst-ethics@lancaster.ac.uk

Next committee deadlines:
FHM – 12th February 12.00
FST – 18th March 12.00

Ethics Approval FST19150 (Interviews and Workshops)

Lancaster University

Applicant: Lucy Hunt
Supervisor: Maria Ferrario
Department: SCC
FSTREC Reference: FSTREC19150

17 June 2020

Re: FSTREC19150
Values in Computing

Dear Lucy Hunt,

Thank you for submitting your research ethics application for the above project for review by the **Faculty of Science and Technology Research Ethics Committee (FSTREC)**. The application was recommended for approval by FSTREC, and on behalf of the Chair of the Committee, I can confirm that approval has been granted for the amendment to this research project.

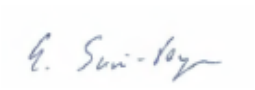
As principal investigator your responsibilities include:

- ensuring that (where applicable) all the necessary legal and regulatory requirements in order to conduct the research are met, and the necessary licenses and approvals have been obtained;
- reporting any ethics-related issues that occur during the course of the research or arising from the research to the Research Ethics Officer at the email address below (e.g. unforeseen ethical issues, complaints about the conduct of the research, adverse reactions such as extreme distress);
- submitting details of proposed substantive amendments to the protocol to the Research Ethics Officer for approval.

Please contact me if you have any queries or require further information.

Email:- fst-ethics@lancaster.ac.uk

Yours sincerely,



Dr. Elisabeth Suri-Payer,
Interim Research Ethics Officer, Secretary to FSTREC.

Ethics Approval FST20115 (DotEveryone Data Set)

Applicant: Lucy Hunt
Supervisor: Dr Maria Angela Ferrario, Dr Alistair Baron.
Department: SCC
FSTREC Reference: FST20115

11 May 2021

Re: FST20115
Values in Computing

Dear Lucy,

Thank you for submitting your research ethics application for the above project for review by the **Faculty of Science and Technology Research Ethics Committee (FSTREC).** The application was recommended for approval by FSTREC, and on behalf of the Chair of the Committee, I can confirm that approval has been granted for this research project.

As principal investigator your responsibilities include:

- ensuring that (where applicable) all the necessary legal and regulatory requirements in order to conduct the research are met, and the necessary licenses and approvals have been obtained;
- reporting any ethics-related issues that occur during the course of the research or arising from the research to the Research Ethics Officer at the email address below (e.g. unforeseen ethical issues, complaints about the conduct of the research, adverse reactions such as extreme distress);
- submitting details of proposed substantive amendments to the protocol to the Research Ethics Officer for approval.

Please contact me if you have any queries or require further information.

Email: fst-ethics@lancaster.ac.uk

Yours sincerely,

T. Morley

Tom Morley,
Research Ethics Officer, Secretary to FSTREC.

Appendix C. Consent Forms

Participant Consent Form

Participant Consent Form

Project Title: Values in Computing **(STORY XXX)**

Name of Participant: _____

Pseudonym to be used in research: _____
(Please leave blank if you prefer the researchers to select a pseudonym)

The purpose of this consent form is to check that you understand what will be required of you, if you agree to take part in this research, and how any information you give will be used in the study.

☐	1. I confirm that I have read and understood the Participant Information Sheet for the above study
☐	2. I have had the opportunity to consider the information, ask any questions about the research and have had these questions answered satisfactorily.
☐	3. I agree to participate in this study. I understand that my participation is voluntary, and I can choose to opt-out of all or part of the study at any time as described in the Participation Information Sheet, within two weeks of participation.
☐	4. I understand that I have the right to withdraw, without giving any reasons for this, at any point during the study.
☐	5. I agree for any interviews I give to be audio recorded.
☐	6. I agree that photographs of me can be taken.
☐	7. I agree that video recordings of me can be taken.
☐	8. I agree that any quotations from what I say during an interview can be used in publications. I understand that my quotations will be used anonymously.
☐	9. I understand that any personal data I provide will be retained and processed by the researcher in accordance with GDPR, as covered by the Data Protection Information Act 2018.
☐	10. By providing contact details, I understand that I will be contacted further by the team with updates about the study.

Participant email/phone number: _____
(optional)

Participant signature:

Researcher signature: _____

Date: _____

Participant Information Sheet (FSTREC 19150, 20063, 19079)

Project Title: Values in Computing **(expert interview)**

Researcher: Lucy Hunt L.hunt1@lancaster.ac.uk

Supervisor: Dr Maria Angela Ferrario

About the research

This project is a systematic approach to the elicitation, articulation, and deliberation of human values in software production. With an increasing number of high impact software incidents, we need to improve software engineering working practices to be more mindful of the wider ethical, social and human impact of what software does or could do. Our aim is to help software engineering teams understand how they and others feel about the creation, impact and use of software. Software Engineering teams that are more aware of the potential impact of their technical decisions and working practices may anticipate and reduce vulnerabilities that lead to incidents and outcomes affecting themselves, end users and wider society.

About the team

The researchers in this team are part of the Values in Computing (ViC) research group, in the School of Computing and Communications at Lancaster University.

Why have you been approached?

You have been approached to take part in this project because you are actively engaged in software production or have an interest in understanding the role of values in the design, development and use of software and digital technologies more broadly. For this reason, we would like to interview you or invite you to a focus group to discuss this. Interviews and workshops will take place online or via email if you have access to the technology to do so. Interviews should last up to 90 minutes. Workshops may last up to 3 hours. Following on from the sessions we may email you for a short (less than 5 minutes) response about participating in the session. You have the opportunity to contact the researchers during or after the study, online, face to face or via email, for any reason.

What does 'Informed Consent' mean?

Before the study commences, you will be asked to sign a consent form to confirm that you have read and received this information sheet and that you are willing to volunteer in this research. You do not have to take part in this study if you do not want to. You have the right to terminate the interview or the focus group session at any point and you are not obliged to answer questions if you do not wish to. You may withdraw from the research without any negative consequences. If you wish to completely withdraw from this research, please do so within two weeks of the interview or the focus group, so that the data collected from you can be excluded from the analysis. Additionally, you can choose not to answer specific questions during the interview or have specific sections of the interview withdrawn (e.g., if there is a specific incident you do not want to include) within two weeks of the interview.

How will we run online sessions?

If we arrange an online session with you, we will be using Lancaster University's Microsoft Teams, a fully GDPR compliant communication and data sharing platform. We will check that you have access to the necessary technology to take part (internet connection, computer or laptop with speaker, microphone and ideally a webcam). Participants will be sent a Teams Meeting link and joining instructions, along with

guidance for using Teams Meetings (link provided by Lancaster University IT). We will run a brief pre-session check to ensure the sound and video is working satisfactorily for us both.

https://answers.lancaster.ac.uk/display/ISS/Joining+online+meetings%2C+webinars+and+other+events+in+Microsoft+Teams

In the unlikely circumstances that Teams is not possible or preferable for participants we will consider other platforms or email interviews. The privacy statements and reputations of software and service providers will be checked before proceeding.

How will we ensure privacy of online sessions?

Conducting sessions remotely will require both participant and researcher privacy while engaging in the online session. Participants and researchers may be in their own home or working from a shared office and may not want to have their participation overheard or disclosed to others. We will arrange sessions to suit both researcher and participant privacy requirements. If participants cannot ensure their privacy, we will rearrange sessions for a time and place when they can. Similarly for the researchers, we will ensure sessions are run at times and location that privacy can be ensured. If a session is interrupted by others not involved in the session, conversations and recordings will be stopped. Reasons for this could be someone entering the room, a phone call needing to be taken or participant or researcher having to leave the room (e.g., for unplanned childcare reasons). The session will only resume if both the participant and researcher is able to do so. The session will be rearranged if it cannot be continued.

Recording sound

We would like to take audio recordings of the interviews, group discussions and workshops. These audio recordings will be transcribed. Parts of the recordings may be used in publications, such as newspaper articles, written reports, public presentations, and on the Lancaster University website and respective social media channels. All personal data will be fully anonymised and kept confidential. Non-personal data will be used for research and cannot be confidential. In addition, we are committed to withholding any data that could be used to identify you, such as employer name, address, etc. Therefore, no one will be able to identify you.

Photographs / Video Recordings

We would like to take photographs of workshops and discussions. If you agree, photographs and video recording–may be used in publications such as newspaper articles, essays, reports, public presentations or websites including the Lancaster University website and its social media channels.

How will we run email interviews?

Conducting interviews via email will require participants and researchers to ensure the privacy and security of data (which may include personal or sensitive data) sent between them. We will agree, via a phone call, which email accounts the researcher and participants will be using and agree a strong password to protect documents sent between researcher and participant. If documents are accidentally sent to another email addresses, password protecting the document reduces risk of disclosures. No interview

questions, including personal or sensitive data, will be asked or sent in the main body of an email. Questions and answers will always be sent in password protected documents attached to an email. The interview may take place over a number of days or even weeks. Once the interview study is completed the participant will be advised to delete all emails received and sent relating to the research. The researcher will store the interview files securely on Lancaster University OneDrive and then delete them from the email system.

Confidentiality and anonymity

All personal data collected from you will be treated with confidentiality. This means that only the research team will have access to any of the raw information that can be specifically associated with you. Any information that is shared beyond this team will be anonymised. Your name and address will be removed, and we will use a pseudonym to refer to you instead. This will apply to any publications or presentations or any discussions with other colleagues in the University. Data that can be used to identify you will also be removed. We will keep personal details (such as your name and contact email, if you provide this) and research content (e.g., interview transcriptions) in separate encrypted and password protected files. Non-personal data will be used for research and cannot be confidential.

How will the data be used and protected?

We will treat data that you have provided in accordance with GDPR, as covered by the Data Protection Information Act 2018. This means that any personal information stored in physical format (paper, readily playable recordings) will be stored in a locked office in Lancaster University premises. For email interviews, no interview questions, including personal or sensitive data, will be asked for or sent in the main body of an email. Questions and answers will always be sent in password protected documents attached to an email. These files will subsequently be stored on a secure and password protected server and deleted from the email system. Any personal information that is stored electronically will be stored on a secure and password protected server. Any personal information that is transported electronically on a mobile device (such as a laptop) will be encrypted and/or password protected. The information collected will be used to inform the development of further research and may be included in academic publications, online reports and presentations. Only anonymised information will be retained indefinitely for on-going research purposes. We will keep the raw data for up to 10 years after the data is collected; after that, the data will be destroyed. For further information about how Lancaster University processes personal data for research purposes and your data rights please visit our webpage: www.lancaster.ac.uk/research/data-protection.

Who has reviewed the project?

This study has been fully reviewed by the Faculty of Science and Technology Research Ethics Committee. If you have more questions please contact Maria Angela Ferrario, School of Computing and Communications, Room C19, C Floor, InfoLab21, Lancaster University, Lancaster LA1 4WA, via email at m.ferrario@lancaster.ac.uk

Issues or complaints

If you have any concern about this study and wish to speak to someone outside the study, you may contact: Prof. Nick Race, Director of Research, School of Computing and Communications, Room D33, InfoLab21, South Drive, Lancaster University, Lancaster.LA1 4WA, UK Tel: +44 (0)1524 510123 n.race@lancaster.ac.uk

Participant Information Sheet (FST17072)

Project Title: Values-First SE

Researchers: Lucy Hunt, Emily Winter, Steve Forshaw

Principal Investigator: Dr Maria Angela Ferrario, Lecturer, School of Computing and

Communications, Lancaster University. Email: m.ferrario@lancaster.ac.uk

About the research

Values-First SE is a systematic approach to the elicitation, articulation, and deliberation of human values in software production. Given the pervasiveness of software and its impact on society, there is a need to make more visible the values-sets built into software systems to better communicate, understand and potentially anticipate how software may behave. However, the interplay between values held by industry, research sponsors, clients, and end-users are complex, difficult to articulate and rarely fully captured by current SE decision-making processes. Values-First SE aims to investigate and make explicit the role that values play during software production processes.

About the team

The researchers in this team are part of the School of Computing and Communications at Lancaster University.

Why have you been approached?

You have been approached to take part in this project because you are actively engaged in software production or have an interest in understanding the role of values in the design and development of software and digital technologies more broadly. For this reason, we would like to interview you or invite you to a focus group to discuss this. You have the opportunity to contact the researchers during or after the study, face to face or via email, for any reason.

What does 'Informed Consent' mean?

Before the study commences, you will be asked to sign a consent form to confirm that you have read and received this information sheet and that you are willing to volunteer in this research. You do not have to take part in this study if you do not want to. You have the right to terminate the interview or the focus group session at any point and you are not obliged to answer questions if you do not wish to. You may withdraw from the research without any negative consequences. If you wish to withdraw from this research, please do so within two weeks of the interview or the focus group, so that the data collected from you can be excluded from the analysis.

Recording sound

We would like to take audio recordings of the interviews, group discussions and workshops. These audio recordings will be transcribed. Parts of the recordings may be used in publications, such as newspaper articles, written reports, public presentations, and on the Lancaster University website and respective social media channels. All personal data will be fully anonymised and kept confidential. Non-personal data will be used for research and cannot be confidential. In addition, we are committed to withholding any data that could be used to identify you, such as employer name, address, etc. Therefore, no one will be able to identify you.

Photographs / Video Recordings

We would like to take photographs of workshops and discussions. If you agree, photographs and video recording may be used in publications such as newspaper articles, essays, reports, public presentations or websites including the Lancaster University website and its social media channels.

Confidentiality and anonymity

All personal data collected from you will be treated with confidentiality. This means that only the research team will have access to any of the raw information that can be specifically associated with you. Any information that is shared beyond this team will be anonymised. Your name and address will be removed, and we will use a pseudonym to refer to you instead. This will apply to any publications or presentations or any discussions with other colleagues in the University. Data that can be used to identify you will also be removed. We will keep personal details (such as your name and contact email, if you provide this) and research content (e.g., interview transcriptions) in separate encrypted and password protected files. Non-personal data will be used for research and cannot be confidential.

How will the data be used and protected?

We will treat data that you have provided in accordance with the EU General Data Protection Regulation 2016/679 (GDPR). This means that any personal information stored in physical format (paper, readily playable recordings) will be stored in a locked filing cabinet in a locked office in Lancaster University premises. Any personal information that is stored electronically will be stored on a secure and password protected server. Any personal information that is transported electronically on a mobile device (such as a laptop) will be encrypted and/or password protected. The information collected will be used to inform the development of further research and may be included in academic publications, online reports and presentations. Only anonymised information will be retained indefinitely for on-going research purposes. We will keep the raw data for up to 10 years after the data is collected; after that, the data will be destroyed.

Who has reviewed the project?

This study has been fully reviewed by the Faculty of Science and Technology Research Ethics Committee. If you have more questions please contact Maria Angela Ferrario,

School of Computing and Communications, Room C19, C Floor, InfoLab21, Lancaster University, Lancaster LA1 4WA, via email at m.ferrario@lancaster.ac.uk

Issues or complaints

If you have any concern about this study and wish to speak to someone outside the study, you may contact: Prof. Nick Race, Director of Research, School of Computing and Communications, Room D33, InfoLab21, South Drive, Lancaster University, Lancaster.LA1 4WA, UK Tel: +44 (0)1524 510123 n.race@lancaster.ac.uk

Appendix D. Whistleblowing Terminology

Taken from online browsing platform for ISO standards. ISO 37002:2021(en), Whistleblowing management systems — Guidelines

Term	Definition
wrongdoing	action(s) or omission(s) that can cause harm. Wrongdoing or the resulting harm can have happened in the past, is currently happening or can happen in the future. Potential harm can be determined by reference to a single event or series of events. • breach of law (national or international), fraud, corruption, bribery. • breach of the *organization* or other relevant code of conduct, breach of organization *policies* ; • gross negligence, bullying, harassment, discrimination, unauthorized use of funds or resources, abuse of authority, conflict of interest, gross waste or mismanagement. • actions or omissions resulting in damage or risk of harm to human rights, the environment, public health and safety, safe work-practices, or public interest.
whistleblower	person who reports suspected or actual *wrongdoing* , and has reasonable belief that the information is true at the time of reporting. Reasonable belief is a belief held by an individual based on observation, experience or information known to that individual, which would also be held by a person in the same circumstances. Examples of whistleblowers include, but are not limited to, the following: • *personnel* within an *organization*; • personnel within external parties, including legal persons, with whom the organization has established, or plans to establish, some form of business relationship including, but not limited to, clients, customers, joint ventures, joint venture partners, consortium partners, outsourcing providers, contractors, consultants, sub-contractors, suppliers, vendors, advisors, agents, distributors, representatives, intermediaries, and investors. • other persons such as union representatives. • any person formerly or prospectively in a position set out in this definition.
whistleblowing	reporting of suspected or actual *wrongdoing* by a *whistleblower* . A report of wrongdoing can be verbal, in person, in writing or in an electronic or digital format. It is common to distinguish: • open whistleblowing, where the whistleblower discloses information without withholding their identity or requiring that their identity be kept secret. • confidential whistleblowing, where the identity of the whistleblower and any information that can identify them is known by the recipient but is not disclosed to anyone beyond a need-to-know basis without the whistleblower's consent, unless required by law. • anonymous whistleblowing, where information is received without the whistleblower disclosing their identity. *Organizations* can use an alternative term such as "speak up" or "raise a concern", or an equivalent.

Appendix D : Whistleblowing Terminology

Term	Definition
triage	assessment of the initial report of *wrongdoing* for the purposes of categorization, taking preliminary measures, prioritization and assignment for further handling. The following factors can be considered: likelihood and severity of impact of wrongdoing or suspected wrongdoing on the *personnel* , *organization* and *interested party* , including reputational, financial, environmental, human or other damages.
detrimental conduct	threatened, proposed or actual, direct or indirect act or omission that can result in harm to a *whistleblower* or other relevant *interested party*, related to *whistleblowing*
corrective action	action to eliminate the cause of a *nonconformity* and to prevent recurrence
risk	effect of uncertainty on *objectives.* An effect is a deviation from the expected — positive or negative. Uncertainty is the state, even partial, of deficiency of information related to, understanding or knowledge of, an event, its consequence, or likelihood. Risk is often characterized by reference to potential events and consequences or a combination of these. Risk is often expressed in terms of a combination of the consequences of an event the associated likelihood of occurrence.

Appendix E. Story Extracts

This appendix provides short, edited extracts from stories not developed into case reports.

Story ID	Title	Organisation Context	Type
4	Train Protection System	Transport	Story
5	Algorithms in Air Traffic Control	Transport	Story
6	Systems integration	Transport	Story
7	Poor practices in IoT solution	Smart City	Story

Story 4 - Train Protection System

The train protection system works by having metal coils in the tracks, which radiate a signal of between 60 and 65 kilohertz; the unit on the train detects the signals from them. So, if you are heading to a red signal and you haven't started decelerating, the signal that comes up from the coils is detected by the unit – “we should be slowing down” and slams on the brakes. The firm had a perfectly good unit, by an electronic engineer implementing it as a gate array. It's a finite state machine, which is defined in the relevant railway standards. They’d decided to do a software implementation of train protection warning system because “that seems to be the way to go”. Now quite why they decided to do that when they had a perfectly adequate working system done in hardware, I just don't know.

The system is relatively simple as a finite state machine, a field programmable gate array can do the job very easily. In fact, the optimum design for the train borne unit for a train protection warning system, has the finite state logic in a gate array, and has a digital signal processor on it to clean up the signal coming from the track bed. The rail track bed is electromagnetically a very noisy place. If you want to put computing power into it, the place to put it is in signal processing and signal acquisition. The finite state logic of “if that's happened and that’s happened, then do that or if you get that, do that”. That is easily done in electronics. If you're doing best practice, best practice design, the sensible place to put all the software is in digital signal processing.

There were three of us working on it. Two electrical engineers and me, I was the only software engineer on the process. We would first get working a prototype, because they only had an individual software process system involved. They were using an Intel 8051 processor for this particular box, it was underpowered. It could not detect the signal quickly enough. I’d agreed with the electronic engineer that we would have a level triggered interface between the electronic detection and the software. I did the finite state software. The engineer tweaked the basic operating loop of the 8051 to cope with the speed at which we had to process things, he changed it to an edge triggered interface.

However, if you missed the point where it changes from low to high, you missed the signal. That had quite a profound effect - it meant it didn't work with the software as I'd written it. He tried to explain it to me, but I couldn't get the thing to work.

They had an emulator, for testing, but that didn't work. I was left with no means of going into the embedded system to find out why isn't this working. If you are testing software on an embedded target, you need tools which will enable you to interrogate the target, or at least a simulation of that target's code in an emulator to say "oh, that's what it's doing - it's missing the signal there". Now the firm were late getting that, and it didn't work, and they never bothered to do anything about it.

Also, I'd found that when working from the rail standard, there was a wording whose significance I'd missed. And it meant I was doing an interleaved concurrency when you actually need true parallel concurrency, a true threaded concurrency of independent threads. This meant I had to rejig the software. That wasn't a big deal. I was at fault here; I hadn't twigged the significance of the rail track standard wording. It said, "if you start a sequence of responding to one signal from a coil, it should not impede you responding to other sequences". The standard interpretation of that, which I didn't know at the time, was "that if you go over one coil and you're processing the signal from that, once you go over another coil, you can't just ignore the other coil".

I was asking questions about it - does it ever happen as the coils are so spaced (a mile apart), has the train always finished processing one set of safety functions before it hits the next. But I got no answers. So I went ahead with the interleaved concurrency, which is fairly simple. It was then told to me what the standard means. I rejigged the software in terms of the state machine. It was OK, but it still wasn't detecting the signal from the electronics. It got to the stage that I've got to have test equipment that enables me to look into what's happening. And they wouldn't provide it. Eventually I got so dissatisfied I just walked out. I suspect that the manager of the group, who was a production engineer, had just been promoted from production foreman. He was much more likely to listen to the electronic engineers than the software engineers. And I was the only software engineer on the project, involving safety critical software, and that's a red flag. If you're the only software engineer involved, and you're working with engineers who can write programs, but are not themselves software engineers, you can be outvoted. Particularly electronic engineers who are inclined to believe that they are software engineers too. They know how to program, but they are not taught anywhere near the amount of computer science that they need to know, in order to understand what best practice in software engineering is. And they may take it personally. "What do you mean, I'm not a software engineer?". Well, what did you study at uni? "Electronics and digital systems?" Where did you acquire expertise in software engineering? "Well, I've been programming for the past three years". When I programmed for three years, I certainly wasn't competent to do safety critical stuff.

Story 5 - Air Traffic Control

Someone came to me and was concerned about the algorithm used to track military aircraft across the sectors of the airspace. Civil aircraft follow designated routes; there are designated approaches. Military aircraft need to get out of a space as quickly as possible, causing the least disruption to civil flight. You don't track or expect a military flight to follow the civil airways, you just track it across a sector. The sectors in UK airspace are polygonal, they're not always convex polygons sometimes they've got re-

entrant angles. Also, there will be one polygon between certain altitudes and another polygon at different altitudes.

One particular member of the staff was concerned about how the algorithm that tracked the civil aircraft across these sectors work. Now it happened to be programmed by a guy I knew previously. And frankly, if I trust anybody to get it right, I trust him to get it right. But I went back to the formal specifications which had been prepared by one of the specialists. And it staggered me what this person had done.

The formal specification of the airspace was a set of disjoint volumes, and that is as far as it went. It's not even a topology, I mean topology is weaker than a geometry. It wasn't even a topology and not even a basis for topology. I would have triangulated the air space. You could have it, so that each sector at each level, was resolved into triangles. Where a triangle at one level, overlaps with one at a lower level. It was triangles all the way down, so whichever way you were crossing a sector, and even if you were changing altitude, you were only ever crossing a triangle. There's two ways to cross a triangle. You can cross it at one point you just touch in apex, you can cross it at two vertices, in which case you're traveling along the side, you can cross it at one vertex and the middle of a side. Or one part of a side. So only four cases, so it's easy to test. Much easier to test than crossing a polygon that's got re-entrant angles.

I recommended that it be triangulated – abstraction mustn't become a mantra. There is a given geometry for an airspace. It's a stereographic projection, you know what the geometry is. And my view, having an abstract formal specification, because in software engineering you don't want premature implementation decisions. But abstracting away from a geometry that you were given, to something less than a topology struck me as asinine. And I raised a memo on that.

It upset the engineer; it upset the person who had written the formal specification. But it was acknowledged as identifying a genuine concern and they recorded it, they put it into their systems and would monitor any problems in that particular piece of software. To see if we ought to consider re-engineering it. I regarded that as a perfectly acceptable thing to do. It did create a bit of an atmosphere in the office. They came to me and said is there a bit of an atmosphere? Well, yes. Do you want to work across the office? It was towards the last month of contract, so I just worked in another area of the office.

Story 6 - Passenger Transport

I was a Senior Safety Engineer; my role was looking at safety assurance of the system they were developing. It quickly became apparent to me and my colleague there were some serious deficiencies in the way the software had been put together. That's slightly unfair. There were deficiencies in the process used to put the software together. There was no assurance evidence available to show what processes had been followed. The software worked, we just didn't know how well it worked, what its capacity was, what its upper limits were and what its safe operating envelope was. We couldn't say that the software didn't work. But we were having trouble providing evidence that assurance had been done. Ultimately, the company were too far down the road to actually put processes in place. I didn't want them to go back and re-engineer everything from the word dot. That's unrealistic. But for them to strengthen up their processes from today and going forward - is a valid approach.

We were being asked to come up with a statement that IEC 50129 had been followed, despite no independent assurance within the company, other than a sweeping statement from the software lead “yeah, of course my team have followed 50129. Why are you doubting me?” That did not sit well with me. My concern is that customers are not well enough informed, and ultimately the customer is taking on the risk without actually realising it. My big concern is an unethical business practice where we're setting our customers up for failure, and the management or developers are getting the rewards, and the customers are taking all the risk. That's not the way it should be.

The software team lead came out with some quite profane language about “what are these two guys doing here? They are just getting in our way. Get rid of them.” So that was the point at which I realised I wasn't wanted by my peers (the people I was trying to help). And management were coming down on software team rather than that of technical quality. They effectively wanted a post hoc verification, but you can't bolt safety on afterwards, and that's what organisations were trying to do. To cut a long story short, we lost a major contract around this time. At same time I was expressing doubts about my colleagues, which didn't sit well with the management. So, I cut my losses and transferred to a different division. That was very much and out of the frying pan and into the fire though….

We were specifying the systems going on to this vehicle, at a design level rather than manufacturing and production. We were looking at the train control system, the air conditioning, the HVAC system, the electrical systems, the passenger information system. And brakes as well. I was the safety engineer of the project. I worked for a Reliability Maintainability and Safety group with extensive experience of the MIL standard documentation (US Department of Defence), effectively an independent review. By independent, I mean independent of the design engineers. We were there confirming the design meets its reliability, maintainability, and safety targets.

I was tasked to ensure that [EN 50126, EN 50128 and EN 50129], requirements of the contract, were followed. These require that the reliability, availability, and maintainability targets are set as design requirements. I was embedded in the design team and was working with the designers to get safety requirements understood, captured, and in the design.

At the time, we had an internal process called “design for safety”, recognising you have to move safety from a post design confirmation into a design activity - so I was really encouraged by that. That was positive. But the project engineer says “you're interpreting 50129 wrong. It's an after the fact confirmation.” We had this discussion about EN 50129, whether it was safety by design or safety confirmation post design. It didn't matter how many people pointed out it started at the requirements phase, was embedded within the design team and has a V&V (verification and validation) after the fact. He said nope, I just want the V&V “I don't care what is done, but you will be signing to say that all the right processes have been followed.” I was being directed by the project engineer to put my professional engineering seal onto the documents to say that the right thing had been done. This person was telling me that I wasn't allowed to do the right thing. This went on for some months. Then we got delays and cost constraints creeping into the project. And the procurement department tell you that they have already picked the supplier of the braking system.

Safety Integrity Level (SIL) is a measure of how good your systems processes are. So SIL-4 is the highest integrity. SIL-0 is no inherent integrity. A braking system that you want to rely upon needs to be, probably, SIL4. That says how well that braking system has been engineered, how rigorous the engineering process is. When a procurement expert tells me they can't afford a braking system SIL-4, that they can only buy a SIL-2 system, it tells me they don't understand what I've been asking for (what they're asking for). I went to the brake supplier to do an audit, to see whether there was any way that we could take their existing processes and support them making a claim of meeting the requirements of EN 50129. And the answer was "no". There were missing pieces in the requirements. It was an evolutionary product they had been making since, well, they've been in the business a long time. The EN 50129 standard was trying to get the systems specification upfront and feed the requirements to other components.

I raised the issue with my boss in the Group. I raised the issues with the project engineer who was responsible for the vehicle. I raised the issue with the chief engineer who is responsible for engineering assurance. I raised the issue with the head of the customer site. Unfortunately, everybody fell back to "we've got to fall into line with what project manager say because he has the ear of the customer." My boss had a countdown on his whiteboard as to how many weeks until his retirement…. Not helpful.

The engineers understood where I was coming from. I gave a couple of classes to the engineers on what is the SIL standards because historically they worked to the MIL standard (which is the component-based approach). I was asking them to change to a system-based approach. At the working level, I had traction. At the management level they didn't want to make waves, and is one thing I have found particularly, where there is a huge gulf between management and engineers. Management fell into line with what corporate management was saying, for the shareholder's benefit. Making a simplistic generalisation, I felt that once somebody is tagged with that management label, they became much more compliant. Large bonuses if they met all of their goals and delivering shareholder benefit?

Ultimately, I ended up signed off as medically unfit for work due to mental health issues. I could not work for them - they didn't want me there. I didn't want to be there; it was by mutual agreement that we parted ways.

Story 7 - Smart IoT

I used to work for a consultancy. If you've got a software problem, you'd hire us. The client had written a lot of complicated software and gotten themselves into a hard mess. It's not just one technology, it's a real mixture of things, complicated, their own custom hardware, software, and special networks. People riding around in trucks, installing things by the side of the road. And the wonderful possibility, because of the way it's set up with hardware, if you make a mistake in the remote update part of the software, you could have million-dollar bug sending people out on trucks, repairing gadgets for the rest of the year. It could destroy their productivity and I think it's about $600 a shot, to go out and service them. I was dealing with a team of two developers and the overall boss of the department who runs lots of different teams. Our team sweeps up all the bugs. A tidal wave of problems that then belong to you. It's just lovely!

We were writing software tools that would allow you to debug and fix problems remotely. They wanted remote screenshots - "the device is not working, show us what's

on the screen". And a requirement came from the project manager, who was also the Scrum Master "can't you just open the port, and we can debug it". Well, yeah, and every hacker on the planet can do that too. The amount of damage to the client could be massive. The attitude was you can take a risk for us. They were happy to ask me for things that they had no courage to write down. I was wondering if I have enough indemnity insurance for this. Fixing bugs on software that's not quite like anything I've seen before is kind of risky. I don't want to be the guy on the front page of The Register after creating a $1,000,000 bug!

I could see in the code base that developers from a long time ago already had utilities to do this. But current team didn't know about it, and they were trying to think "shall we buy a tool to do this" or "shall we use remote debugging?" But I could see stuff in the code base that could already do this. I made it work, but my relationship with the boss was destroyed. They thought I was obstructive. After I finished, I got a lot of really good feedback. But what's the point in doing what's best for the client? This was a hill I should not want to die on.

The Scrum Master is responsible for the welfare of the team and for making sure that we do quality work. On the other hand, he's the CTO of the international arm of the consultancy. I'm not in a position to argue with this guy, I'm a contractor. It's a much more delicate position than an employee. If they don't like you - it can be "I'm sorry, you'll just have to go". This is the complexity of trying to defend myself professionally and working with somebody interested in short term results. He's responsible for delivery. And it would look awesome if I could deliver something, a bit brutal and nasty, where I take all the risk. He can employ the Volkswagen defence "I didn't know, some boffin must have…."

Where did the design of the system sit?

Oh, that's another interesting story. There must be tools that you can use to point at the source code repository; that would mine the social history of the project.

- How many people worked on this?
- A timeline of who worked on it?
- How long has this project existed for?
- What is the longest period anyone worked on this codebase?
- Are there particular groups of people that came and went?
- Any particular important days when lots of people finished and/or started?

It turns out that on this project, the typical lifetime of a programmer on that project is about a year, and it's had about 3 generations of people work on it. GitHub has all the right data for that kind of thing. It's not that the decisions are explicit, but it tells you a lot more than just the code. It tells you a lot about the environment.

Appendix F. Literature Review

Word Analysis of Whistleblowing Items Abstracts

Word analysis of whistleblowing item abstracts (before the software engineering sift)

A	B	C	D	E
ID	Location	Words	Count	Percentage
10	abs	"whistle"	161	85%
19	abs	stud* (study, studies)	58	31%
28	abs	research	53	28%
54	abs	organisation / organization	53	28%
14	abs	ethic*	41	22%
38	abs	anonym*	41	22%
53	abs	law / legal	40	21%
18	abs	model	38	20%
17	abs	develop*	37	20%
36	abs	news / story	33	17%
37	abs	security	31	16%
34	abs	behav* (behaviour)	30	16%
30	abs	method*	26	14%
40	abs	risk	25	13%
21	abs	theor*	24	13%
32	abs	decision	23	12%
39	abs	privacy	23	12%
11	abs	software	22	12%
13	abs	project	21	11%
15	abs	profession*	18	10%
25	abs	responsib*	18	10%
12	abs	engineer	17	9%
16	abs	program*	17	9%
23	abs	societ*	17	9%
27	abs	survey	16	8%
48	abs	snowden	16	8%
9	abs	whistle AND software	14	7%
22	abs	framework	13	7%
35	abs	scenario	13	7%
55	abs	teach* or educat*	13	7%
29	abs	empirical	11	6%
44	abs	corruption	11	6%
45	abs	fraud	11	6%
46	abs	wrong* (doing, doer)	11	6%
20	abs	case stud* (study, studies)	10	5%
41	abs	threat	10	5%
56	abs	consequence*	10	5%
31	abs	respondent	9	5%
50	abs	wikileaks	7	4%
52	abs	journali*	6	3%
24	abs	population	5	3%
33	abs	dilemma	5	3%
42	abs	insider threat	5	3%
26	abs	disclosure	4	2%
43	abs	information security	4	2%
57	abs	harm	4	2%
47	abs	dissent	3	2%
49	abs	assange	3	2%
51	abs	dark web	2	1%
8	abs	whistle AND software engineer	1	1%

Inter-rating Cohen-Kappa

	Observered	MAF									
		1	2	3	4	5	6				
	LH (primary)	WB Process	Human Factors	WB Tech	Org / Prof	Stories	SE				
1	WB Process	6	0	0	0	0	0	6			
2	Human Factors	0	20	0	0	0	0	20			
3	WB Tech	0	0	12	0	0	0	12			
4	Org / Professional	1	0	0	5	1	0	7			
5	WB Stories	0	0	0	0	14	0	14			
6	SE Aspects	0	0	0	0	0	1	1			
		7	20	12	5	15	1		Total	60	
	2 discrepencies						Total	60		Concordant	58	96.55%
										Non	2	

	given observed marginals										
	Chance expected	MAF									
		1	2	3	4	5	6				
	LH (primary)	WB Process	Human Factors	WB Tech	Org / Prof	Stories	SE	Fixed Marginal		Check Total	
1	WB Process	0.70	2.00	1.20	0.50	1.50	0.10	6		6.00	
2	Human Factors	2.33	6.67	4.00	1.67	5.00	0.33	20		20.00	
3	WB Tech	1.40	4.00	2.40	1.00	3.00	0.20	12		12.00	
4	Org / Professional	0.82	2.33	1.40	0.58	1.75	0.12	7		7.00	
5	WB Stories	1.63	4.67	2.80	1.17	3.50	0.23	14		14.00	
6	SE Aspects	0.12	0.33	0.20	0.08	0.25	0.02	1		1.00	
	Fixed Marginal	7	20	12	5	15	1		Total	60	
	Check Total	7.00	20.00	12.00	5.00	15.00	1.00	60.00			
							Total	60		Concordant	13.87
										Non-Con	46.13

	Excess Observed	(58 - 13.87)	44.13
	Non Expected Concord	(60 - 13.87)	46.13
	Cohen Kappa	Excess / Non	0.956644

Appendix G. Survey Pilot

This Appendix presents a Qualtrics survey piloted as an alternative to interviews.

Introduction:

In the School of Computing and Communications, at Lancaster University, we are researching how software engineering (SE) teams become aware of technical decisions and working practices that could cause harm to individuals, teams, organisations, the SE community, or wider society.

Participation:

If you are or have been actively engaged in software production or have an interest in understanding the role of professional standards and values in the design, development and use of software and digital technologies more broadly - we invite you to take part in this survey. The survey is in three parts.

- Basic details about you
- Your membership of professional bodies
- About harmful SE practices, software, and situations you have observed.

The survey is anonymous. Any dates, individuals, or organisations you name in free format fields will be redacted and anonymised. Additionally, the data itself will not be publicly shared. If you would like to contact us regarding this survey or our research more generally: my contact details XXXXXX.

Section 1 – Basic Details

QID	Question	Response
1a,b,c	Age, gender, country of residence	Number; checklist; drop down list
2	Years of IT/SE experience	Number ranges: 1-2-5-10-15-20-25-30-45-50-55-60+
	Current role; previous roles	Free format
	Industry sectors you have worked in	Free format

Section 2 – Professional Bodies

<table>
<tr><th>QID</th><th>Question</th><th>Response</th></tr>
<tr><td>3</td><td>Membership of a Professional Body?

Currently, ever (if not currently), years of membership
<table>
<tr><th></th><th>Currently</th><th>Ever</th><th>Years</th></tr>
<tr><td>IEEE</td><td>On / Off</td><td>On / Off</td><td>Num</td></tr>
<tr><td>ACM</td><td>On / Off</td><td>On / Off</td><td>Num</td></tr>
<tr><td>BCS</td><td>On / Off</td><td>On / Off</td><td>Num</td></tr>
<tr><td>IET</td><td>On / Off</td><td>On / Off</td><td>Num</td></tr>
<tr><td>Non IT PB</td><td>On / Off</td><td>On / Off</td><td>Num</td></tr>
<tr><td>Trade union</td><td>On / Off</td><td>On / Off</td><td>Num</td></tr>
<tr><td>Other</td><td></td><td></td><td></td></tr>
</table></td><td>Alternative – all free format?

Non IT PB – academic, sector specific</td></tr>
<tr><td>3a</td><td>About your involvement in professional bodies
Why did you join?
How do you use your membership?</td><td>Free format</td></tr>
<tr><td>3b</td><td>Are you aware of Code of Conduct / Ethics for your professional bodies?
Not aware, Aware, Skimmed, Read, Used => 3c</td><td>Tick all that apply</td></tr>
<tr><td>3c</td><td>Why did you read the codes?
Interest, Guidance, Issue Resolution, Procedures, other</td><td>Tick all that apply</td></tr>
<tr><td>3d</td><td>Are you aware of Code of Conduct / Ethics for your employer / client?
Not aware, Aware, Skimmed, Read, Used => 3e</td><td>Tick all that apply</td></tr>
<tr><td>3e</td><td>Why did you read the codes?
Interest, Guidance, Issue Resolution, Procedures, other</td><td>Free format</td></tr>
</table>

Section 3 – Harmful situations

Please read the following taken from the ACM Code of Ethics:

"Harm" means negative consequences, especially when those consequences are significant and unjust. Examples of harm include unjustified physical or mental injury, unjustified destruction or disclosure of information, and unjustified damage to property, reputation, and the environment. This list is not exhaustive. Well-intended actions, including those that accomplish assigned duties, may lead to harm. When that harm is unintended, those responsible are obliged to undo or mitigate the harm as much as possible. To minimize the possibility of indirectly or unintentionally harming others, computing professionals should follow generally accepted best practices unless there is a compelling ethical reason to do otherwise. Additionally, the consequences of data aggregation and emergent properties of systems should be carefully analysed.

A computing professional has an additional obligation to report any signs of system risks that might result in harm. If leaders do not act to curtail or mitigate such risks, it may be necessary to "blow the whistle" to reduce potential harm. However, capricious

or misguided reporting of risks can itself be harmful. Before reporting risks, a computing professional should carefully assess relevant aspects of the situation.

Please cast your mind back over the last 10 years of your SE career and think about SE workplace situations that were or could have been harmful. This section is free format – for you to write as much or as little as you wish or are able to share with researchers.

4	About a situation	ORID GUIDANCE
4a	**Describe a harmful SE workplace situation you were involved in or observed.**	Events that made you aware of the situation. Facts about the situation. Who was involved – what happened, when and why? Was anyone / anything harmed?
4b	**How and why do you think the situation came about?** **Who was responsible for the situation?**	Your interpretation of the situation
4c	**How did it make you feel then?** **How does it make you feel now?**	Your feelings about the situation
4d	**What did you / others do about the situation?**	Decisions you / others made about the situation: • Speak out to SE team • Speak up internally (organisation) • Stayed silent • Left team or company • Blew the whistle • Other
4e	**What was the outcome?** **Was it satisfactory?**	For you, for your team, organisation, wider society?
4f	**How could the situation have been handled differently?**	

You can report on more than one situation – ADD EXTRA “Q4 RESPONSES”

Appendix H. Case Study – Checklist

This appendix is based on Runeson et al.'s Case Study Research in Software Engineering (Runeson *et al.*, 2012), giving guidance for reading and reviewing Case Study Research.

A1: Design of the Case Study
1. What is the case and its units of analysis?
2. Clear objectives, preliminary research questions, hypotheses defined in advance?
3. Is the theoretical basis—relation to existing literature or other cases—defined?
4. Are the authors' intentions with the research made clear?
5. Is the case adequately defined (size, domain, process, subjects…)?
6. Is a cause–effect relation under study?
7. Does the design involve data from multiple sources (data triangulation), using multiple methods (method triangulation)?
8. Is there a rationale behind the selection of subjects, roles, artifacts, viewpoints, etc.?
9. Is case relevant to validly address the research questions (construct validity)?
10. Is the integrity of individuals/organizations taken into account?
A2: Data collection
11. Is a case study protocol for data collection and analysis derived?
12. Are multiple data sources and collection methods planned (triangulation)?
13. Are measurement instruments and procedures well defined?
14. Are planned methods and measurements sufficient to fulfil objective of the study?
15. Is the study design approved by a review board, and informed consent obtained?
16. Is data collected according to the case study protocol?
17. Is the observed phenomenon correctly implemented?
18. Is data recorded to enable further analysis?
19. Are sensitive results identified (for individuals, the organization or the project)?
20. Are the data collection procedures well traceable?
21. Does the collected data provide ability to address the research question?

A3: Data Analysis and Interpretation
22. Is the analysis methodology defined, including roles and review procedures?
23. Chain of evidence with traceable inferences from data to research questions and existing theory?
24. Are alternative perspectives and explanations used in the analysis?
25. Is a cause–effect relation under study? If yes, is it possible to distinguish the cause from other factors in the analysis?
26. Are there clear conclusions from the analysis, including recommendations for practice/further research?
27. Are threats to the validity analysed in a systematic way and countermeasures taken? (Construct, internal, external, reliability)
A4: Reporting and Dissemination
28. Are the case and its units of analysis adequately presented?
29. Are the objective, the research questions and corresponding answers reported?
30. Are related theory and hypotheses clearly reported?
31. Are the data collection procedures presented, with relevant motivation?
32. Is sufficient raw data presented (e.g. real-life examples, quotations)?
33. Are the analysis procedures clearly reported?
34. Threats to validity reported along with countermeasures taken to reduce threats?
35. Report ethical issues openly (personal intentions, integrity issues, confidentiality)
36. Does the report contain conclusions, implications for practice and future research?
37. Does the report give a realistic and credible impression?
38. Is the report suitable for its audience, easy to read and well structured?
A5: Reader's Checklist
39. Are the objective, research questions, and hypotheses (if applicable) clear and relevant? (items 1, 2, 5, 29, 30)
40. Are the case and its units of analysis well defined? (1, 5, 28)
41. Is the suitability of the case to address the research questions clearly motivated? 8, 9, 14

42. Is the case study based on theory or linked to existing literature? 3
43. Are the data collection procedures sufficient for the purpose of the case study (data sources, collection, validation)? 11, 13, 16, 18, 21, 31
44. Is sufficient raw data presented to provide understanding of the case and the analysis? 32
45. Are the analysis procedures sufficient for the purpose of the case study (repeatable, transparent)? 22, 33
46. Is a clear chain of evidence established from observations to conclusions? 6, 17, 20, 23, 25
47. Are threats to validity analyses conducted in a systematic way and are countermeasures taken to reduce threats? 27, 34, 37
48. Is triangulation applied (multiple collection and analysis methods, multiple authors, multiple theories)? 7, 12, 22, 24
49. Are ethical issues properly addressed (personal intentions, integrity, confidentiality, consent, review board approval)? 4, 10, 15, 19, 35
50. Are conclusions, implications for practice and future research, suitably reported for its audience? 26, 29, 36, 37, 38

Appendix I. SE Practice Scenarios and Hypotheses

Experimental Hypotheses from Laboratory Experiments

Scenario	Hypotheses	N	Title
Project status / failure technological boundaries pushed many unexpected technical glitches failure to survive in marketplace.	**H1: A greater perception that information ought to be communicated reflected in a higher level of perceived personal responsibility for reporting.** **H2: The level of perceived personal responsibility for reporting inversely related to an individual's reluctance to transmit negative information.** **H3a: Individuals will exhibit a greater reluctance to report negative information when perceive a high risk of negative personal consequences** **H3b: Individuals will assess risk of negative personal consequences for external reporting options to be higher than internal reporting options** **H3c: Individuals will exhibit more reluctance to report through external than through internal reporting channels.** **H4: Individuals will be more likely to claim that a project's status ought to be reported when perceived project risk is high.** **H5: Perceived project risk will increase with perceived impact.** **H6: There will be a significant negative relationship between perceived risk propensity and perceived project risk.** **H7: Individuals are more likely to assess that negative information ought to be reported when they perceive higher levels of wrongdoing.** **H8: Individuals are more likely to assess a personal responsibility to report a project's status when they perceive higher levels of wrongdoing.**	136	Keeping Mum as The Project Goes Under: Toward an Explanatory Model Organizational Factors and Bad News Reporting on Troubled It Projects
Project status / failure system limitations	**H1: A stronger assessment that information ought to be communicated reflected in a higher assessed level of personal responsibility for reporting.** **H2: Higher levels of assessed personal responsibility will be associated with less reluctance to report bad status news.** **H3: Organizational climates where individuals encouraged to report negative information lead to stronger assessments that individual personally responsible for reporting.** **H4: When negative information can be hidden effectively from superiors, individuals will be less inclined to assess that such information ought to be reported.**	122	'Why Didn't Somebody Tell Me? Climate, Information Asymmetry, And Bad News About Troubled Projects

Appendix I : SE Practice Scenarios and Hypotheses

Scenario	Hypotheses	N	Title
Software failing. external vendor. Reporting, blame shift	**H1 Controlling for culture, a blame-shifting opportunity associated with a greater willingness to report bad news than if no such blame-shifting opportunity is presented.** **H1a Korean subjects will be more willing to report bad news when there is a blame-shifting opportunity than when there is not.** **H1b US subjects will be more willing to report bad news when there is a blame-shifting opportunity than when there is not.** **H2 When a blame-shifting opportunity is present, US subjects will exhibit greater willingness to report bad news than Korean subjects** **H3 In the absence of a blame-shifting opportunity, Korean subjects will exhibit greater willingness to report bad news than US subjects.**	146	Reporting Bad News on Software Projects: The Effects of Culturally Constituted Views of Face-Saving
Unethical IT use, IPR, privacy violations	**H1a: Higher Machiavellian individuals will be less likely to report IT-related intellectual property infractions.** **H1b: Higher Machiavellian individuals will be less likely to report IT-related privacy infractions.** **H2a: Females are more likely to report IT-related intellectual property infractions than males.** **H2b: Females are more likely to report IT-related privacy infractions than males.** **H3a: Individuals with higher levels of computer literacy as measured by programming experience will be less likely to report IT-related intellectual property infractions.** **H3b: Individuals with higher levels of computer literacy as measured by programming experience will not be less likely to report IT-related privacy infractions.** **H4a: Gender will moderate the relationship between Machiavellianism and reporting IT-related intellectual property infractions. Relationship will be stronger for males.** **H4b: Gender will moderate the relationship between Machiavellianism and reporting IT-related privacy infractions. The relationship will be stronger for males.** **H5a: Computer literacy as measured by programming experience will moderate the relationship between Machiavellianism and reporting IT-related intellectual property infractions. The relationship will be stronger for those with more programming experience.** **H5b: Computer literacy as measured by programming experience will not moderate the relationship between Machiavellianism and reporting IT-related privacy infractions. The relationship will not be stronger for those with more experience.**	72	Blowing The Whistle on Unethical Information Technology Practices: The Role of Machiavellianism, Gender, and Computer Literacy
IT Project Reporting Outsourced IT projects, external vendor blame shifting	**H1: A stronger assessment that information ought to be communicated will be associated with a stronger assessment of personal responsibility for reporting.** **H2: A stronger assessment of personal responsibility will be associated with greater willingness to report bad news.** **H3: When fault responsibility can be placed on an external vendor, individuals are more likely to assess that negative information ought to be reported.** **H4: When fault responsibility can be placed on an external vendor, individuals will be more willing to report bad news.** **H5: When higher levels of time urgency are perceived, individuals are more likely to assess that the project status ought to be reported.** **H6: When higher levels of time urgency are perceived, individuals are more likely to perceive themselves as having a personal responsibility to report the project's status.**	159	Overcoming The Mum Effect in IT Project Reporting: Impacts of Fault Responsibility and Time Urgency

Appendix I : SE Practice Scenarios and Hypotheses

Scenario	Hypotheses	N	Title
Bodily harm or financial loss, scope of impact: one in a billion or 99% Radiation treatment, wealth management	**H1. When the type of impact involves bodily harm, this will have a greater effect than financial loss on perception of impact.** **H2. The scope of impact will positively affect the perceived impact.** **H3. The greater the perceived impact of IT failure, the more likely people assess that the situation should be reported.** **H4a. Individuals with higher levels of morality are more likely to assess that the bad news concerning a project and its status ought to be reported.** **H4b. Individuals with higher levels of morality are more likely to assess a personal responsibility to report a project's status.** **H4c. Individuals with higher levels of morality will be more willing to report bad news.** **H5. Individuals with greater willingness to communicate will be more willing to report bad news.** **H6. A stronger assessment that information ought to be communicated will be reflected in a higher assessed level of personal responsibility for reporting.** **H7. Higher levels of assessed personal responsibility will be associated with greater willingness to report bad news.**	155	The Effect of IT Failure Impact and Personal Morality on IT Project Reporting Behaviour
vignettes of ethical IT dilemmas, programmer hacking into bank software, employee using computer equipment for personal work.	**H1A: Higher levels of professionalism will increase ethical IT behaviour intention.** **H1B: Higher levels of professionalism will increase ethical IT whistleblowing intention.** **H2A: Higher levels of Machiavellianism will decrease ethical IT behaviour intention.** **H2B: Higher levels of Machiavellianism will decrease ethical IT whistleblowing intention.** **H3A: The caring work climate will have a positive impact on levels of professionalism, and a negative impact on levels of Machiavellianism.** **H3B: The laws, codes, and rules work climates will have a positive impact on levels of professionalism.** **H3C: The instrumental work climate will have a negative effect on levels of professionalism, and a positive effect on levels of Machiavellianism.** **H4A: Higher levels of moral recognition will increase moral equity judgments, but decrease moral relativism judgments.** **H4B: Higher levels of moral equity judgments will increase ethical IT behaviour and whistleblowing.** **H4C: Higher levels of moral relativism judgments will decrease ethical IT behaviour and whistleblowing.**	240	Who Says Professionals Are Ethical? A Cross-Sectional Analysis of Ethical Decision Making, Attitudes and Action
Project delay. Fault in code. Job loss threat.	**H1. The effect of assessment on willingness to report is partially mediated by responsibility.** (across three secondary data sets from other studies)	661	Bad News Reporting on Troubled IT Projects: Reassessing the Mediating Role of Responsibility in Whistleblowing Model

Appendix I : SE Practice Scenarios and Hypotheses

Scenario	Hypotheses	N	Title
Software piracy. Social pressure	**H1. Individuals who have more reasons for supporting the act of whistleblowing on software piracy will have a more favourable attitude toward whistleblowing.** **H2. Individuals who have more reasons against the act of whistleblowing on software piracy will have a more unfavourable attitude toward whistleblowing.** **H3. Individuals who have more reasons for supporting the act of whistleblowing on software piracy will have a stronger intention to whistleblow.** **H4. Individuals who have more reasons against the act of whistleblowing on software piracy will have a stronger intention not to whistleblow.** **H5. Individuals who perceive greater control over the act of whistleblowing on software piracy will have a stronger intention to whistleblow.** **H6. Individuals who perceive more social pressure from important others to whistleblow on software piracy will have a stronger intention to whistleblow.** **H7. Individuals with more favourable attitudes toward the act of whistleblowing on software piracy will have a stronger intention to whistleblow.** **H8.With the same attitude toward software piracy whistleblowing, individuals with internal LOC would be more likely to whistleblow than individuals with external LOC.** **H9. The presence of monetary incentives would strengthen the relationship between attitude toward the act of whistleblowing on software piracy and the intention to whistleblow.** **H10. With the same attitude toward software piracy whistleblowing, individuals with a bad relationship with their company would be more likely to whistleblow compared to individuals with a good relationship with their company.** **H11. A higher level of legal protection provided to the individuals would strengthen the relationship between the attitude toward the act of whistleblowing on software piracy and the intention to whistleblow.**	290	To Blow or Not to Blow: An Experimental Study on The Intention to Whistleblow on Software Piracy
failing software project	**H1: The level of employees' organizational commitment will positively affect their willingness to report bad news in software projects.** **H2: The level of employees' closeness with the wrongdoer will negatively affect their willingness to report bad news in software projects.** **H3: Employees' beliefs on the Confucian ethical principle on loyalty between sovereign and subject will positively affect their willingness to report bad news in software projects.** **H4: Employees' beliefs on the Confucian ethical principle on trust between friends will positively moderate the relationship between employees' closeness with the wrongdoer and their willingness to report bad news in software projects.**	144	The Impact of Relationships and Confucian Ethics on Chinese Employees' Whistle-Blowing Willingness in Software Projects

Appendix I : SE Practice Scenarios and Hypotheses

Scenario	Hypotheses	N	Title
Safety critical system, bug reporting	**H1a: The magnitude of consequences caused by a software bug in a safety critical system has a positive direct effect on willingness to report bad news** **H2a: The probability of effect caused by a software bug in safety critical system has a positive direct effect on willingness to report bad news** **H3b: The probability effect caused by a software bug in a safety critical system has an indirect effect on willingness to report bad news mediated by morality judgement** **H3a: Temporal immediacy of the negative consequences caused by a software bug in a safety critical system has a positive direct effect on willingness to report bad news** **H3b: Temporal immediacy of the negative consequences caused by a software bug in a safety critical system has an indirect effect on willingness to report bad news that is mediated by morality judgement** **H4a: Social consensus against delivering a safety critical system with a software bug has a positive direct effect on willingness to report bad news** **H4b: Social consensus against delivering a safety critical system with software bug has indirect effect on willingness to report bad news mediated by morality judgement** **H5a: Proximity to victims of safety critical systems containing a bug has a positive direct effect on willingness to report bad news** **H5b: Proximity to victims of safety critical systems containing a bug has an indirect effect on willingness to report bad news that is mediated by morality judgement**	173	The Effect of Moral Intensity on IT Employees' Bad News Reporting
Project failure. technological boundaries pushed. unexpected technical glitches. Failure to survive in marketplace	**H1: Affective organizational commitment (AOC) will positively affect an individual's willingness to report bad news in software projects.** **H2: Interpersonal closeness with the wrongdoer (ICW) will negatively affect an individual's willingness to report bad news in software projects.** **H3a: The effect of affective organizational commitment (AOC) on willingness to report bad news is strengthened by loyalty between sovereign and subject (CESS).** **H3b: The effect of affective organizational commitment (AOC) on willingness to report bad news is weakened by loyalty between sovereign and subject (CESS).** **H4a: The negative effect of interpersonal closeness with the wrong- doer (ICW) on willingness to report bad news is strengthened by trust between friends (CETF).** **H4b: The negative effect of interpersonal closeness with the wrong- doer (ICW) on willingness to report bad news is weakened by trust between friends (CETF).**	158	Impacts of Organizational Commitment, Interpersonal Closeness, and Confucian Ethics on Willingness to Report Bad News in Software Projects
Potential problems associated with complying with the offending team member's request.	**H1: Higher the level of experience of an individual, the less likely the individual will engage in team member silence behaviours about an offending team member's actions.** **H2: The level of experience of an individual within a team will be positively related to his/her personal responsibility to report.** **H3: When the offending team member has a more prominent role within the team, an individual will be more likely to engage in team member silence behaviours.** **H4: When the offending team member has a more prominent role within the team, an individual will perceive a higher sense of personal responsibility to report.** **H5: A team member with a stronger sense of a personal responsibility to report will be less likely to engage in team member silence behaviours.**	231	If You Can't Say Something Nice: Factors Contributing to Team Member Silence in Distributed Software Project Teams