Systems Theoretic Process Analysis of Sociotechnical Systems

by

Polly Harrington

B.S. Engineering Psychology
Tufts University, 2021

Submitted to the
Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree(s) of

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2025

Signature of Author: _____
Department of Aeronautics and Astronautics
May 16, 2025

Certified by: _____
Nancy G. Leveson, Ph.D.
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by: _____
Jonathan P. How
Richard Cockburn Maclaurin Professor in Aeronautics and Astronautics Chair, Graduate Program Committee

Systems Theoretic Process Analysis of Sociotechnical Systems

by

Polly Harrington

Submitted to the Department of Aeronautics and Astronautics
on May 16, 2025, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

The safety and success of complex modern systems, such as hospitals, aircraft, or software, depend on their ability to integrate people and technical components. For example, doctors must be able to use their computerized surgical tools to treat their patients successfully, airplane pilots must be able to operate the required controls for takeoff and landing, and regulators must be able to interpret the data they receive to make critical decisions. However, designing systems that facilitate safe interactions between humans and technology is not a simple task. System designers must consider not only the constraints of the technical components but also human requirements throughout the entire system. However, accidents in modern systems continue to prove that more work is needed to identify and prevent unsafe interactions between humans and technology

Systems Theoretic Process Analysis (STPA) is a hazard analysis methodology based on systems theory that has been used to improve system safety in various industries, including healthcare, aviation, nuclear power, and automotive design. However, if hazard analysts using STPA lack significant expertise in human factors engineering (HFE), they may be unable to thoroughly and rigorously identify critical unsafe interactions.

This thesis presents a process for utilizing HFE to improve the results of STPA analyses on sociotechnical systems. In particular, the process focuses on the thorough identification of causal scenarios in sociotechnical systems by incorporating relevant human factors concepts. The process allows analysts without significant training in HFE to improve their ability to identify useful scenarios for humans in their system. The effectiveness of the improved process is demonstrated using a healthcare case study on over-the-counter clinical laboratory tests in the United States.

By establishing a process for non-HFE experts to use when conducting STPA analyses, more systems can be developed that enhance human performance rather than increase conflict between humans and the engineered system.

Thesis Supervisor: Nancy G. Leveson, Ph.D.

Title: Professor of Aeronautics and Astronautics

Acknowledgments

First and foremost, I would like to thank the members of the System Safety group, particularly Professor Leveson. It has been an incredible experience to join the lab and collaborate in research. Additionally, I would like to thank Brittany and Rodrigo. I was fortunate to have completed the master's program alongside you both. ESL has not been the same without you.

Finally, thank you to those who attended the Marshmallow Fluff Festival in Fall 2022 and to those who joined in all the events, trips, and activities that followed. You have all made my experience at MIT so much better than I ever imagined it could be.

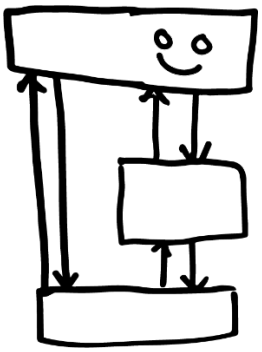

Figure-1. STAMPY, the unofficial STAMP mascot.

Table of Contents

Chapter 1: Introduction

Accidents in critical systems such as aviation, shipping, and healthcare have led to significant losses for both the companies involved and the affected individuals. However, many of these accidents, from crashes to missed medication, are preventable. Traditional hazard analysis methods, the processes by which engineers identify and mitigate safety problems in designs, have improved the safety of relatively simple systems in the last century (N. Leveson, 2011). However, they are unable to sufficiently identify hazards that emerge from the interactions of humans and technology (N. Leveson, 2004). If any humans are included in hazard analyses, it is often only at the operator level and does not include the wider organizational context of managers and other system decision-makers (Hofmann et al., 2017; Nazaruk & John, 2020).

In 2019, for example, two Boeing 737-Max aircraft crashed after a sensor and automation malfunction. Boeing had rushed the release of the 737-Max to compete with the Airbus 320 NEO, which had threatened Boeing's market share. Ultimately, after the grounding of all 737-Max planes worldwide, Boeing incurred losses of over $18 billion (Gelles, 2020). The Boeing 737-Max accidents were initially attributed to technical failures (Nicas et al., 2019) and to the pilots' inability to regain control over the planes (Cook, 2019; Langewiesche, 2019). However, further investigation proved that broader systemic problems with Boeing's safety culture and organizational decision-making contributed to the accidents. While hazard analyses were performed on the plane components and the pilots received extensive training and screening (Merida, 2017), there was minimal safety analysis of the management structure that opted to fix an identified design flaw with automated control system changes rather than changing the fundamental aircraft design.

Another major accident with significant societal impact was the 2024 collapse of Baltimore's Francis Scott Key Bridge after a collision with a container ship. The collision was attributed to technical issues with the ship's power systems (NTSB, 2024; Pollard, 2024). However, the ship had undergone the required safety inspections (Kypriotaki, 2024). What is less understood is how the overall shipping industry, including ports, regulatory authorities, and shipping companies, uses the results of those inspections and other performance data to make safety decisions. For example, in the days before the accident, the container ship lost power several times (Coy, 2024). Despite knowing that the ship's power systems were not fully functional, the sociotechnical systems managing the shipping canal and the shipping companies were unable to prevent the accident. To prevent similar accidents in Baltimore and in other ports around the world, both the technical components of the boats and the sociotechnical system managing international shipping need to be improved.

An accident with similar themes occurred in a Tennessee hospital in 2017. In this incident, a nurse provided incorrect medication to a patient experiencing claustrophobia before a PET scan. However, the nurse accidentally provided a medication called vecuronium instead of Versed, which resulted in the death of the patient (Kelman, 2022). One contextual factor that influenced the accident was that the nursing unit was significantly understaffed at the time of the accident. Therefore, the nurse had to multitask between many critical and time-sensitive tasks with

insufficient support. Furthermore, the nurse was not assigned to a specific section of the hospital. Instead, the nurse was assigned as a "floater," someone who supports different sections of the hospital as needed. Therefore, the nurse had less experience with tasks in the radiology unit than a more specialized nurse would have had (Williams et al., 2023). While the digital medication dispensing cabinet was subject to hazard analysis and the nurse received years of training, the context in which the nurse and cabinet interacted was not subject to the same degree of analysis. Ultimately, the nurse who administered the incorrect medication was criminally charged, while the managers who put the nurse into this under-resourced situation were not subjected to the same scrutiny.

Each of these incidents could have been prevented by a better hazard analysis that could identify unsafe interactions between humans and technology. While individual accidents may be triggered by an action of a system-level operator, the system that created the conditions in which the accident occurred was created by interactions between the technology and managers, operators, regulators, shareholders, and others. As Mica Endsley writes, humans are often "the final dumping ground for the inherent problems and difficulties in the technologies we have created" (Endsley, 2012, p. 553). The ability to identify hazards that arise from human-technology interactions is critical to preventing future accidents from occurring. Unfortunately, while many engineers and social scientists acknowledge that modern systems are usually made up of interactions between technical and social components, few hazard analyses adequately evaluate the interactions between technology and humans (N. Leveson, 2004).

One field working to address the way that sociotechnical systems are analyzed is Human Factors Engineering (HFE), which researches how human capabilities and limitations can inform system design. However, while HFE methods excel at analyzing and understanding human-technology interactions, they have not been widely adopted or have only been incorporated in a cursory manner. In healthcare, for example, the use of HFE in device and hospital design has increased over the last 20 years (Weinger et al., 2011). However, most physicians, pharmacists, and other experts are acutely aware that the technology they use does not always improve the safety of their work (Poon et al., 2021; Wu et al., 2021). The lack of adequate adoption of HFE may be due in part to three factors: the reliance on domain-isolated systems analysis methodologies, the limited scope of HFE analyses, and the lack of HFE training among engineers.

Safety analysts frequently use methods that limit their analyses to either the technical components or the humans in a system. For example, popular hazard analysis methods, including Fault-Tree-Analysis (FTA) and Failure Modes and Effects Analysis (FMEA), only model technical elements of a system when investigating how accidents could occur (Czaja & Nair, 2012). Conversely, hazard analysis methods used frequently by social scientists, such as Systems Engineering Initiative for Patient Safety (SEIPS) and Task Analysis, often exclude technical components and focus primarily on human actions (Baxter & Sommerville, 2011; N. Leveson, 2011).

However, no human-created system is purely technical or purely social (Wilson et al., 2012). Even as the role of automation increases, humans are still involved in building, directing, and

supervising the automation. Drawing artificial barriers between humans and technology in systems results in analyses that only address some of the myriad ways a system could perform unsafely. Without considering both humans and technology together, interactions between the human and technical components of the system may be missed entirely.

Furthermore, the scope of HFE analyses is also usually limited to the system user or operator. However, the interaction of humans in the broader system may also significantly impact safety. For example, an electronic health record interface that appears safe in usability testing may not perform as expected in a hospital because hospital managers made unsafe implementation decisions, such as approving incorrect laboratory test menus or not enabling certain features. Implementation decisions are often rushed and underinformed, especially because there is insufficient analysis of what information and resources managers need to inform EHR implementation decisions (N. Leveson et al., 2023).

Finally, even if an organization identifies that incorporating human factors considerations into their analyses would benefit them, many systems engineering teams do not have sufficient human factors expertise to do so. Identifying safety concerns related to human factors is not straightforward. Engineers frequently believe they can easily identify human-system design flaws without HFE training because they are human. However, this has not proven to be the case (Wickens, 2002). Human factors experts draw from years of training in ergonomics, cognitive psychology, and engineering (Karwowski, 2012), and many engineering programs do not consider human factors in their required coursework (Dadmohammadi et al., 2017). Only 15 universities in the United States have a Human Factors undergraduate degree program registered with the Human Factors Engineering Society (HFES) (HFES, 2025) and only 22 universities offer HFES accredited graduate degrees. As a result, most engineers lack adequate training in HFE (Fossum et al., 2018). Teams without effective training in HFE may be unable to perform a holistic human factors safety analysis sufficiently or may devalue the importance of HFE (Czaja & Nair, 2012).

To address these three factors, this thesis demonstrates how the hazard analysis method Systems Theoretic Process Analysis (STPA) can be used to conduct a thorough HFE analysis of a sociotechnical system. STPA is based on systems theory, which, at a high level, posits that in complex systems, “the whole is more than the sum of its parts” (Bertalanffy, 2009). In other words, a system’s behavior emerges from the interactions between different components. Therefore, the system in its entirety must be considered as a whole, including both humans and technological components. Additionally, STPA is particularly well suited to human factors analyses because the way system components are modelled, using control theory, is similar to the way that human information processing is currently understood (Proctor & Van Zandt, 2018).

The choice of STPA addresses the inability of traditional methods to address interactions between humans and technology. STPA also enables the analysis of humans in the system beyond the system operator or user. However, without HFE expertise, analysts using STPA may not be able to identify hazards caused by HFE concerns effectively (Czaja & Nair, 2012). Therefore, this thesis provides detailed scenario archetype templates that facilitate the identification of loss scenarios in sociotechnical systems. These archetypes are based on an

expanded model of human behavior informed by HFE research and enable those without significant HFE training to use STPA to conduct more thorough and rigorous analyses of sociotechnical systems.

Previous work by Megan France extended STPA to model interactions between the operator and the controlled system (France, 2017). This thesis expands France's work by providing guidance that analysts can use to improve the results of their STPA analyses on sociotechnical systems at the hierarchical levels above the operator, including within the management and organizational structure.

Chapter 2: Literature Review

The following chapter reviews the current best practices for analyzing safety in sociotechnical systems and evaluates why they have been unable to reliably identify and prevent unsafe interactions between humans and technology. Then, STPA is introduced in detail to demonstrate how it improves the results of hazard analyses on sociotechnical systems.

Many methods of hazard analysis are used to evaluate safety in sociotechnical systems. Many of the most common methodologies, including Failure Modes and Effects Analysis (FMEA), Fault Tree Analysis (FTA), and Hazard and Operability (HAZOP) studies, were developed for technical systems and were later augmented to include users and humans (Sharit, 2012). Human factors engineers have also developed system analysis tools, such as Human Factors Analysis and Classification System (HFACS), Systems Engineering Initiative for Patient Safety (SEIPS), and Task Analysis (TA).

In the first section, each of these methods is described in depth. The following section reviews the underlying limitations that prevent them from thoroughly identifying hazards.

2.1 Popular Hazard Analysis Methods used in Sociotechnical Systems

Methods used to identify hazards vary by industry. Some of the most common methods are described below.

2.1.1 Failure Modes and Effects Analysis (FMEA)

FMEA is one of the most popular risk analysis methods (Vincoli, 2006). For example, in healthcare, the Joint Commission, which oversees hospital certification in the United States, and the Food and Drug Administration (FDA) recommend FMEA as a hazard analysis method for medical devices (Joint Commission Resources, 2020).

At a high level, FMEA analyzes what would happen to the system if each system component failed and classifies each potential failure by severity (Stephans & Talso, 1993). Therefore, it is excellent for identifying single-point failures, which are ways an entire system could fail if only one component breaks (Vincoli, 2006).

More specifically, FMEA analyzes the potential failure modes of each component and considers them in the context of all operational modes. The goal is to identify which component failures could cause the most severe accidents in each operational mode. The identified loss pathways are then designed out or prevented through the addition of barriers (N. Leveson, 2023; Stephans & Talso, 1993; Vincoli, 2006).

According to the System Safety Analysis Handbook (1993), the basic steps of an FMEA are to identify the:

1. Components or processes of interest
2. System-level consequences are to be prevented
3. Failure modes for each component

4. Impact of failures on the system
5. What mitigation or barriers currently exist
6. The probability and severity of each failure

Traditional FMEAs are conducted on technical systems. However, HFE researchers have also expanded FMEA methods to include human operators. One such method, Human-FMEA, analyzes human errors rather than component failures. Once the potential human errors are identified, each potential error is analyzed in the context of all operation modes (Sharit, 2012).

An example Human-FMEA that looks at each step in a process for a human is shown below in Figure 2. 1 An example section of an human-FMEA analysis on a sociotechnical system. The section depicted analyzes the task of the FDA identifying audit targets when regulating medical devices.. The example excerpt is from an FMEA that depicts the process of the FDA identifying a manufacturer to audit.

Subtask	Step (e.g., critical task)	Risk ID Number (ID)	Use Error (Failure Mode)	Cause of Failure	Hazardous Situation	Harm	Probability	Criticality	Mitigation
Identify Audit Target	Review Safety Data	1	Doesn't view all data	data is not all in the same place	Manufacturer with unsafe devices is not audited	Unsafe manufacturer not corrected	3	1	Check information against existing accounts
	Identify manufacturers over target	2	Selects manufacturer with safety data below target	Data is duplicated in database	Audit applied to a manufactuer without significant problems	Wasted time and resources	3	1	Display message that informs users to check that the entered data is accurate
		3	Evaluates incorrect date period	Did not correctly set date parameters	Uncompleted steps	Interface won't provide accurate search results	2	1	Ensure date period defaults to set guidelines
	Evaluate Mitigating Factors	4	Doesn't notice manufacturer has an Audit already in place	Display does not indicate companies under audit	Wasted time and resources	Account isn't made	3	1	Display an indication that a company is already under review

Figure 2. 1 An example section of an human-FMEA analysis on a sociotechnical system. The section depicted analyzes the task of the FDA identifying audit targets when regulating medical devices.

While FMEAs are described as "universally applicable to systems" (Stephans & Talso, 1993), they struggle to handle non-failure causality modes. Furthermore, their applicability to complex human decisions has been critiqued (N. Leveson, 2023). Because humans do not "fail" in the same way as a mechanical system, human errors must be contorted to fit the framework. Another well-known limitation of FMEA is its inability to meaningfully identify critical events that could be triggered by multiple failures (Stephans & Talso, 1993).

2.1.2 Fault Tree Analysis (FTA)

FTA was developed in the 1960s and has remained largely unchanged since (N. Leveson, 2023). Given the current system design, FTA first identifies the hazards, or undesirable system states, and then determines the chain(s) of events that could lead to their occurrence. Events are

connected via logical gates (e.g., AND or OR gates). The series of events is modeled until the analyst determines that the “primary events” of each chain have been identified. The fault tree is then analyzed to determine what combinations of events could cause the hazard in question. Event probabilities are often calculated and used to determine the likelihood of the hazard occurring given the current system design (N. Leveson, 2023; Sharit, 2012; Stephans & Talso, 1993).

According to the System Safety Analysis Handbook (1993), the basic steps of an FTA are to:

1. Define the top event or the failures of interest
2. Define the boundaries of the analysis
3. Define the tree structure
4. Identify paths of failures for all branches in the fault tree
5. Identify the minimum cut set of events in the tree that could lead to the top-level failure

Work has been done to enable the analysis of a system's organizational or social components using FTAs. For example, the Management Oversight and Risk Tree (MORT) builds on an FTA analysis and acknowledges that accidents are usually caused by a multitude of human and technological factors (Knox & Eicher, 1976). MORT builds fault trees of accident prevention barriers with three branches: specific control factors, management system factors, and risk factors (known and accepted risks) (Sharit, 2012). MORT has been used across many industries, including defense (Knox & Eicher, 1976).

An example FTA for the FDA conducting an audit of a device manufacturer is depicted in Figure 2. 2 below. The top-level “failure” event is the FDA's failure to identify unsafe diagnostic test devices. The events that could lead to that outcome build the rest of the fault tree.

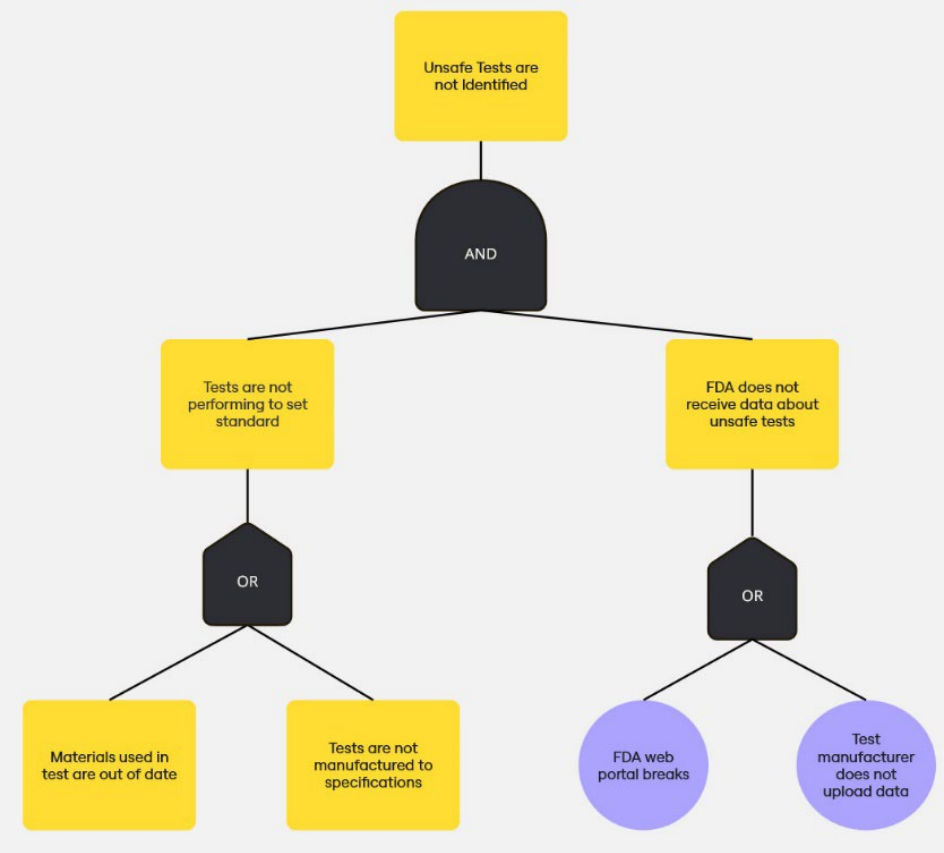

Figure 2. 2 An example section of an FTA analysis on a sociotechnical system. The fault tree depicts the events that could lead to the FDA not identifying an unsafe laboratory testing device.

Unlike FMEA, FTA is a top-down analysis. By starting with the high-level events to be prevented and moving down to understand what events could cause them, FTAs generate fewer results that are not meaningful to an analysis than an FMEA (Stephans & Talso, 1993).

One of the major risks associated with quantitative FTAs is that they require a precise probability for all events in the cut set. If not all probabilities are well understood, the ultimate probability for the top-level event will not be accurate (Stephans & Talso, 1993). If inaccurate probabilities are used in an FTA, the analysts will not have a realistic understanding of the risks in their current design. One common way that probabilities can be miscalculated is if the underlying events in the fault tree are treated as if they are independent. Often failures will be caused by the same conditions. For example, a power outage could trigger failure events on different sides of a fault tree.

2.1.3 Hazard and Operability (HAZOP) Studies

HAZOP studies use guidewords to identify how the system may deviate from the design's intended function (Crawley & Tyler, 2015). Analysts take the intended operational states and question what consequences could emerge if the system deviates from those conditions. Common guidewords include "too much," or "too little, "reverse, "before," and others (Crawley & Tyler, 2015). HAZOP is most commonly used in the process industry (Kariuki & Löwe, 2007).

The steps in a HAZOP analysis are to identify:

1. Each step in a process
2. The intentions and parameters of each step
3. The possible deviations in each step, using guidewords
4. The Consequences from each potential deviation
5. Causes of identified deviations
6. Current mitigations
7. Missing mitigations (Stephans & Talso, 1993)

HAZOP has been expanded to include human factors considerations by looking at human decision-making using guidewords such as "Wrong operation on right object," "Wrong selection made," or "wrong information communicated," among others (Crawley & Tyler, 2015). Human-augmented HAZOPs use human-error taxonomies to search for potential human errors at different steps of a manufacturing process or other sequential processes.

Figure 2. 5 shows an example excerpt from an HAZOP analysis of the FDA's process of auditing a device manufacturer. Specifically, Figure 2. 5 depicts deviations from the step of "Identify manufacturers who require an audit."

Guideword	Deviation	Causes	Consequences	Safeguard	Severity	Likilyhood	Ranking	Reccomendations
Too little	No manufacturers are identified	Doesn't view all data,data is not all in the same place	Unsafe manufacturer not corrected	Checklist to review all data sources	Manufacturer with unsafe devices is not audited	1	3	Combine data streams into one portal
	Not all manufacturers who need an audit are identified	Selects manufacturer with safety data below target Data is duplicated in database	Wasted time and resources	Script to remove data duplication	Audit applied to a manufactuer without significant problems	2	2	Display message that informs users to check that the entered data is accurate
Too much	Too many manufacturers are identified for review	Data is inaccurate	Account isn't made	Data cleaned before use	Wasted time and resources	1	3	Display an indication that a company is already under review

Figure 2. 11 An example section of a HAZOP analysis on a sociotechnical system. The section depicted analyzes the task of the FDA identifying audit targets when regulating laboratory test devices.

2.1.4 Human Factors Analysis and Classification System (HFACS)

HFACS is a human-centered risk analysis tool. First used and created by the US Navy, HFACS is now used across a wide variety of industries, including healthcare and construction (HFACS, Inc, n.d.; Jalali et al., 2023).

HFACS is based on Reason's Swiss Cheese model of accident causality. Reason's Swiss Cheese Model asserts that accidents happen when the vulnerabilities of all accident-prevention barriers line up. Therefore, the model assumes accidents will be prevented if any barrier vulnerabilities are fixed in the chain of events. HFACS evaluates each barrier and identifies where vulnerabilities align and would allow an accident to happen (HFACS, Inc, n.d.).

However, as opposed to other methods based on the Swiss Cheese model, HFACS explicitly analyzes organizational system attributes including management (Jalali et al., 2023). To conduct

an HFACS analysis, four categories of system attributes are evaluated: organizational influences, supervisory factors, preconditions for unsafe acts, and unsafe acts. Within each of these categories, there are several sub-categorizations that can direct the analysis. For example, to analyze the supervisory factor, analysts are guided to consider inadequate supervision, planned inappropriate operations, failure to correct known problems, and supervisory violations. Overall, HFACS contains nineteen subcategories across the four main factors(Jalali et al., 2023).

The basic steps of an HFACS analysis are to identify (Jalali et al., 2023):

1. What unsafe acts must be avoided
2. What errors or violations could lead to unsafe acts
3. The preconditions that could lead to unsafe acts
4. The supervisory factors that could lead to unsafe preconditions
5. The organizational influences that could lead to unsafe supervisory factors

Figure 2. 20 depicts the HFACS model with each of the categories and subcategories considered in the framework.

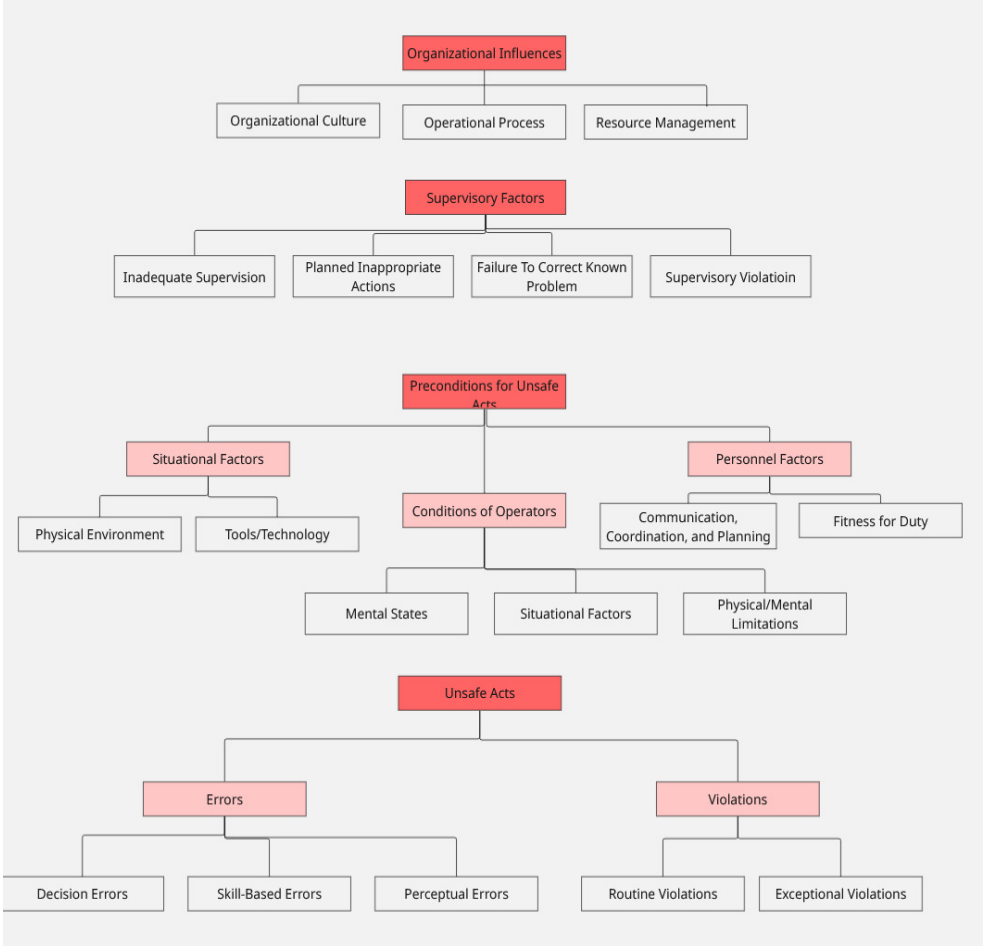

Figure 2. 20 shows the HFACS model. The hierarchical relationship between the four main classes of system attributes and their corresponding subattributes are depicted.

2.1.5 Systems Engineering Initiative for Patient Safety (SEIPS)

SEIPS is a model for improving safety in healthcare focused on the humans in the system (Carayon et al., 2006). SEIPS was developed to simplify systems engineering principles for increased application by non-HFE experts and ensures analyses of sociotechnical systems consider more than the individual workers in a system (Holden & Carayon, 2021). Like HFACS, SEIPS is based on Reason's Swiss cheese accident causality model but modifies it slightly into the "work-system model."

The work system model examines the interactions among five components of a sociotechnical system: the human, tasks, tools, organizational environment, and physical environment. This model posits that each human in the system accomplishes tasks using tools while being influenced by the organizational and physical environment (Carayon et al., 2006). SEIPS analyses rely on a broad swath of information sources, including surveys of staff, observations, available environmental/building information, job descriptions, and others. The five work-system components are used to categorize information found.

The process of a SEIPS analysis is less defined than the previous methods. However, the basic process is to analyze the work system, the processes, the employee and organizational outcomes, and the patient outcomes, and question how they could contribute to unsafe patient care (Carayon et al., 2006).

Figure 2. 23 below shows a SEIPS model for the process of a device manufacturer going through approval and performance monitoring. Each device is subject to several systems within the FDA throughout its product lifecycle. In the SEIPS analysis, each work system would be analyzed for potential safety risks in the tasks, tools, organizational conditions, and the physical environment.

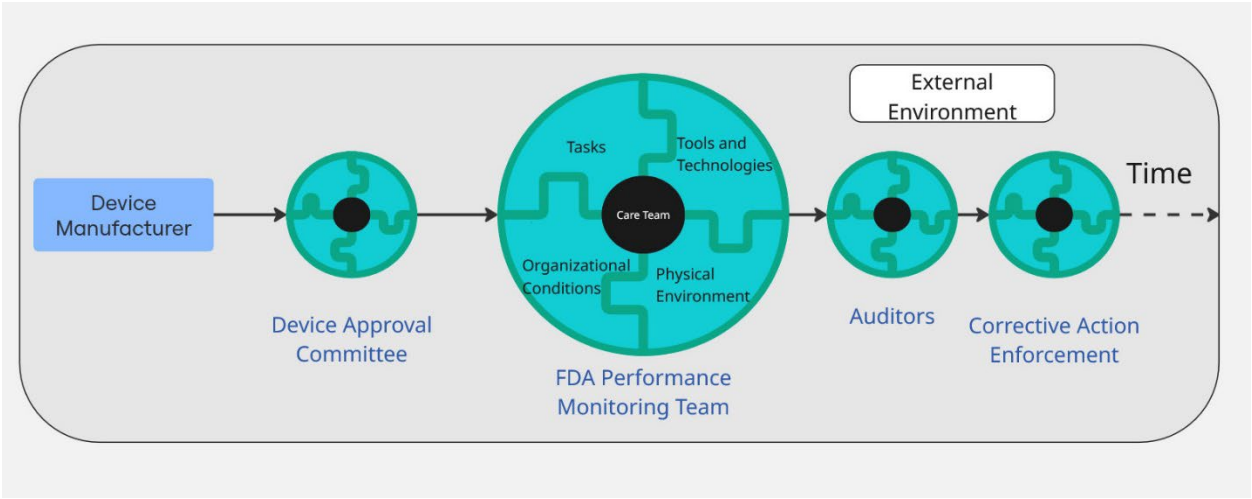

Figure 2. 29 shows the various worksystems that a device manufacturer will interact with as their clinical lab test is developed and released. Each work system has tasks, tools, organizational conditions, and a physicial environment that all impact safety.

2.1.6 Task Analysis (TA) and Human Reliability Analysis (HRA)

TAs are a common tool HFE practitioners use to identify potential pitfalls of processes or designs. There are numerous types of TAs, including Hierarchical TAs (HTAs), Cognitive TAs

(CTAs), and Emotional TAs (Intriligator, 2022). However, at the basic level, a TA identifies the steps needed to achieve a goal. As described by Erik, a TA identifies "WHO does WHAT and WHY" (Erik, 2012). TAs are commonly used when a process is too complex for a single person to comprehend fully and can be especially useful when a process involves collaboration between several people (Erik, 2012).

Depending on the type of TA, steps may be conceptualized as decision points, physical actions, or sub-goals. For example, CTAs evaluate how people think and make decisions as they complete a task. There is no set definition of what comprises the most granular chunk of a TA. The determination of the granularity of a TA is a judgment call and depends on the context of the overall task and goals of the analysis (Sharit, 2012). For example, a hospital could conduct a TA of a surgical preparation procedure that includes the administration of medication via an IV bag. If the hospital's goals include determining how many clinicians should be involved in the procedure, it may not include the sub-steps of the task "connect the IV." However, if the hospital is trying to understand how long the preparation might take or identify potential hazards, it may be necessary to dissect the IV connection task further.

The basic steps of a TA are to:

1. Identify the main goal or task of the human or team
2. Identify the steps needed to complete that goal or task. These are often the "decision points" where a user will make different choices depending on the task context
3. Iteratively refine the tasks into more detailed sub-tasks until the analyst determines they have sufficient detail

Figure 2. 38 below shows an example TA

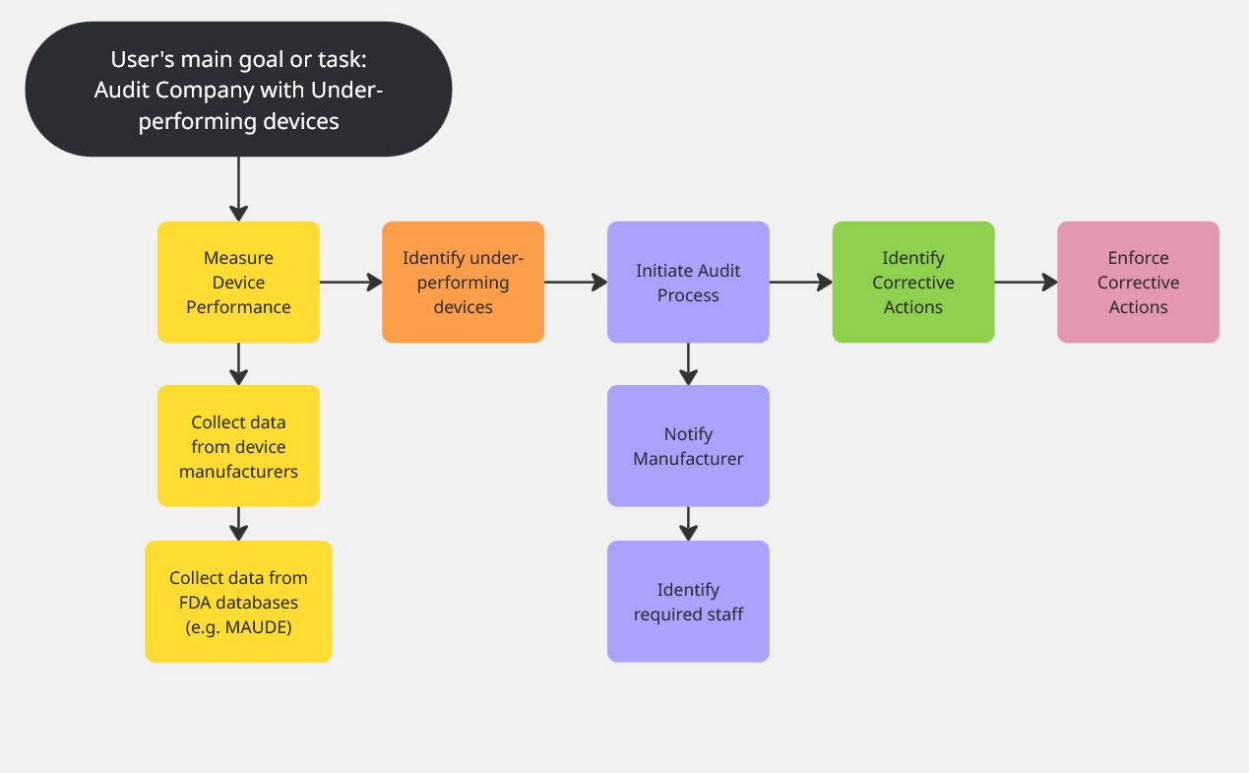

Figure 2. 38 is a high-level task analysis of the FDA auditing a company with underperforming or unsafe devices. Some of the main tasks are shown with their corresponding sub-tasks.

A Human Reliability Analysis (HRA) is a TA that evaluates what errors are possible at each step (Wilson & Norris, 2005), similar to HAZOP. For many HRAs, the end goal is error rate prediction (Birch et al., 2023). These error rate predictions are usually based on expert estimates of human behavior (Birch et al., 2023). HRAs are included in IEEE standards, ASME standards, and SHARP standards(Sharit, 2012) and they are used across many industries such as nuclear energy and chemical processing (Kariuki & Löwe, 2007).

Many industries, including rail, medical device manufacturing, and aviation, require risk analyses that produce quantitative probabilistic risk estimates. Therefore, companies and regulators try to plug humans into traditional analysis techniques (Wilson & Norris, 2005) by assigning probabilities to each identified potential human error (Majewicz et al., 2020; Xi et al., 2017). However, as stated earlier, human decision-making is biased by the context in which the human operates. Probabilistic risk calculations rely on assumptions of randomness in human decision-making and often ignore the biases present in an environment where the human is tasked with making the decision in question. Human decision-making is not random and cannot be accurately analyzed using traditional risk probability estimation techniques.

2.2 Gaps in Applying Common Safety Analysis Models and Techniques to Sociotechnical Systems

Most safety-critical industries, such as aviation and healthcare, are required to apply one or more of the methods listed above (N. Leveson, 2023; Stephans & Talso, 1993). One of the reasons that accidents continue to happen, despite the use of the identified hazard analysis methods, is that systems have grown in complexity since many of them were introduced (N. Leveson, 2004). These methods are unable to identify hazards stemming from:

- Humans in the system other than the operator,
- Non-failure events,
- Complex interactions between humans and technology,
- Non-linear system behavior.
- Sociotechnical systems involve humans beyond the operator

All methods discussed in the previous section have been used to analyze sociotechnical systems. However, the human-technology interactions modeled by the identified methods are often limited to computer interfaces and physical controls (Food and Drug Administration, 2016; Hofmann et al., 2017; Wiklund, 2022).

The methods listed above often struggle to fully analyze the system surrounding higher-level decision-makers in systems, including managers, designers, and organizational leadership.

One of the reasons that these methods struggle to incorporate higher-level decision making is that they rely on tasks being well-defined. Because most methods of incorporating humans into the identified methods begin with a TA, they are not adequate for complex managerial tasks that do not have a defined step-by-step procedure that happens in the same way each time. Task Analysis works best on repetitive tasks with defined steps, which may work for operator functions but rarely work well for those higher in the system (Rasmussen, 1990).

2.2.1 Adverse events in sociotechnical systems have causes beyond failures

One reason why the traditional hazard analysis methodologies cannot sufficiently identify all safety hazards in modern systems is that they are based on a model of accident causation that assumes that failures of components or humans are the sole cause of accidents. However, many accidents occur when all components are working as designed, but the interaction between components causes the system to behave unsafely (N. Leveson et al., 2023).

For example, FTA assumes that each event in the fault tree is a failure. FTA analyses do not consider whether system-level failures could arise even if no component experiences a failure (N. G. Leveson, 2023).

Accidents caused by interactions between components are unlikely to be caught in advance when testing and requirements focus on preventing component-level failures and do not identify how the system could perform unsafely, even when all components are acting as designed.

2.2.2 Sociotechnical systems have complex interactions between components

All of the identified hazard analysis methods rely on analytic decomposition, which is breaking down systems into parts to analyze separately with the assumption that if each

component performs adequately and safely when analyzed independently, the whole system will work when assembled (N. Leveson et al., 2023). For analytic decomposition to work, system components must not have significant interactions when the system is assembled; each component must behave the same way independently as it would within the system (N. Leveson, 2011). However, modern systems rely on significant component interaction, also called coupling. As a system becomes tightly coupled, it becomes impossible to guarantee that how a component acts on its own is how it will act in the context of the fully assembled system (N. Leveson et al., 2023). The assumption of independent components is particularly hazardous in social systems because humans are always influenced by their environment, and their decision-making cannot be separated from the context in which it occurs (Klein, 2008; Rasmussen et al., 1990; Tversky & Kahneman, 1974)

Furthermore, not only is there often an assumption that system components will not interact, but there is also an assumption that any failure events will be independent. Assuming independence allows a calculation of the statistical probability of the undesired event using a probabilistic quantitative analysis of the combination of preceding events. However, if two or more of the events have a common cause, the calculated risk will be incorrect (N. G. Leveson, 2023).

2.2.3 Sociotechnical systems are dynamic and non-linear

The identified hazard analysis tools are also all based on a linear model of accident causality, which assumes all events, such as accidents, are preceded by a linear chain of causal events. Within the event chain, each event causes the next event directly and sequentially (N. Leveson, 2023). Logically, it follows from this reasoning that stopping any event in the sequence will prevent the final accident. Modern systems, however, do not always follow a linear chain of events (Sterman, 2009). Events or actions taken in the past often have delayed or compounded effects that emerge over time. When an accident happens and is investigated, events that contributed to the unsafe system context, such as maintenance, may appear entirely disconnected and irrelevant to investigators.

System analysis methods developed by human factors researchers, such as SEIPS or HFACS, can theoretically identify organizational or contextual factors, but they lack an underlying model of accident causality that forces analysts to consider and identify larger systemic factors (Baker, 2022). When the accident causality model is based on a linear chain-of-failures model, investigation and prevention of hazards often end with the first human who could have made a decision that would have avoided the accident, regardless of the system context. When human operators are blamed, solutions tend to be limited to retraining, termination, or even legal prosecution of low-level employees (N. Leveson, 2023; Williams et al., 2023). For example, a meta review of HFACS analyses found that 80% of the incidents analyzed using the HFACS framework labeled the main cause of the accidents as unsafe acts or preconditions for unsafe acts and were focused on the “immediate environment in which work is performed” as opposed to the wider organizational structure (Jalali et al., 2023).

2.3 Hazard Analysis Based on Systems Theory

Relying on systems engineering tools that are unable to identify major hazards in sociotechnical systems is dangerous. Humans are prone to minimizing risk, especially when faced with evidence that confirms this bias (Tversky & Kahneman, 1974). It is paramount for systems engineers to utilize methodologies that can handle the complex, coupled, non-linear systems built today, especially regarding human interaction.

System-theory-based methods, such as Systems Theoretic Process Analysis (STPA), mitigate the challenges listed above. Systems theory originated in the biological sciences with researchers including Von Bertalanffy, who emphasized the age-old idea that "the whole is more than the sum of its parts" (Bertalanffy, 2009). In biology, this makes intuitive sense; no one body part can be well understood without understanding the human body holistically. However, systems theory applies far beyond biological systems and is extraordinarily helpful for any complex system. Systems Theory is particularly useful when dealing with systems that are not large enough that Bayesian statistics apply, but are not simple enough to be trivial (Bertalanffy, 2009).

Systems theory was developed when Bertalanffy and others recognized that many fields were discovering laws or principles (such as growth or competition) that aligned closely with those developed in other unrelated fields. The only commonality across the disciplines was the focus on system behavior (Bertalanffy, 2009).

Systems theory enables the identification and analysis of emergent properties. Emergent properties stem from the interactions between system components. For example, a bicycle cannot be analyzed for stability or speed by examining its tires, gear mechanism, handlebars, or rider independently. Only when all the components are together and interacting do properties, such as stability and speed, emerge. Many critical system properties are emergent, including, but not limited to, safety, reliability, profitability, and maintainability.

Systems theory can be applied to any system with emergent properties. In systems theory, a system is defined as a group of components working together to achieve a goal (Weinberg, 2001). Systems theorists emphasize that systems are simplified models based on human perception and interpretation of reality (Von Bertalanffy, 1972). That is, there are no natural barriers that define one entity as part of a system and define another entity as outside of it. Every system can be broken into more detailed subsystems or abstracted into broader systems (N. Leveson, 2011). Engineers draw boundaries between systems to simplify cognitive tasks and solve particular problems.

2.3.1 Systems Theoretic Process Analysis (STPA

Systems Theoretic Process Analysis (STPA) is a hazard analysis method based on a systems theoretic accident model called Systems Theoretic Accident Model and Processes (STAMP). In the STAMP model, in order for a system to effectively produce the emergent property, there must be adequate control and feedback relationships that allow controllers within the system to monitor the system's performance and provide the correct control inputs (N. Leveson, 2011). STPA analyzes the control and feedback relationships in a system to identify how losses can be prevented (N. G. Leveson et al., 2012).

Controllers in a system may include humans, organizations, computers, or mechanical devices. Controllers use control actions to modify the state of the system. Control actions include policy changes, electronic signals, commands, and directions, among others. According to Leveson, the four essential conditions a controller must possess to be successful are a "goal condition," an "action condition," an "observability condition," and a "model condition" (N. G. Leveson, 2017).

A goal condition is the behavior or status that the controller wants the controlled process or component to exhibit. The controller cannot select appropriate control actions if the goal condition is misunderstood or unclear.

An action condition is the ability of a controller to make appropriate changes to the process/component. If the controller is unable to provide adequate controls, even if it knows what behavior it wants to see, it will be unable to do anything to move the system to the goal condition.

The observability condition is a controller's ability to perceive the system's actual behavior. Without adequate feedback on the system's behavior, the controller cannot determine if the system is meeting the goal condition or if control actions are necessary. The observability condition also includes the timing of the feedback. For example, does the controller receive the information in time to make the required decisions? A sensor that updates once a minute is sufficient for some systems, but may not provide sufficient data in others.

Finally, the model condition requires that the controller has an adequate understanding of the system and the effect of their controls on it (i.e., a "process model"). For humans, this system understanding is often called a mental model (Rasmussen, 1987). If a controller cannot predict how their actions will impact the system, it will be difficult, if not impossible, for them to select the best control actions (France, 2017).

Together, these conditions create a control loop. STPA models systems using the control loops within the system. These models are called control structures, and they allow analysts to identify how a system's controls may be insufficient to maintain the system-level emergent properties. A thorough STPA analysis identifies missing or insufficient controls, feedback, or entire control-feedback loops in the system by examining each control loop in the model, as well as reviewing the model holistically.

STPA is a top-down analysis, which starts with the losses that stakeholders want to prevent and identifies potential causes of those losses. Therefore, STPA analyses only generate relevant scenarios that lead to losses. In contrast, many traditional hazard analyses, such as FMEA, work from the bottom up, starting at a component level and generating numerous scenarios that do not necessarily result in a loss (N. Leveson, 2023).

2.3.2 STPA Basics

There are four main steps in an STPA analysis (N. G. Leveson et al., 2012). The four steps are broadly shown below in Figure 2. 47.

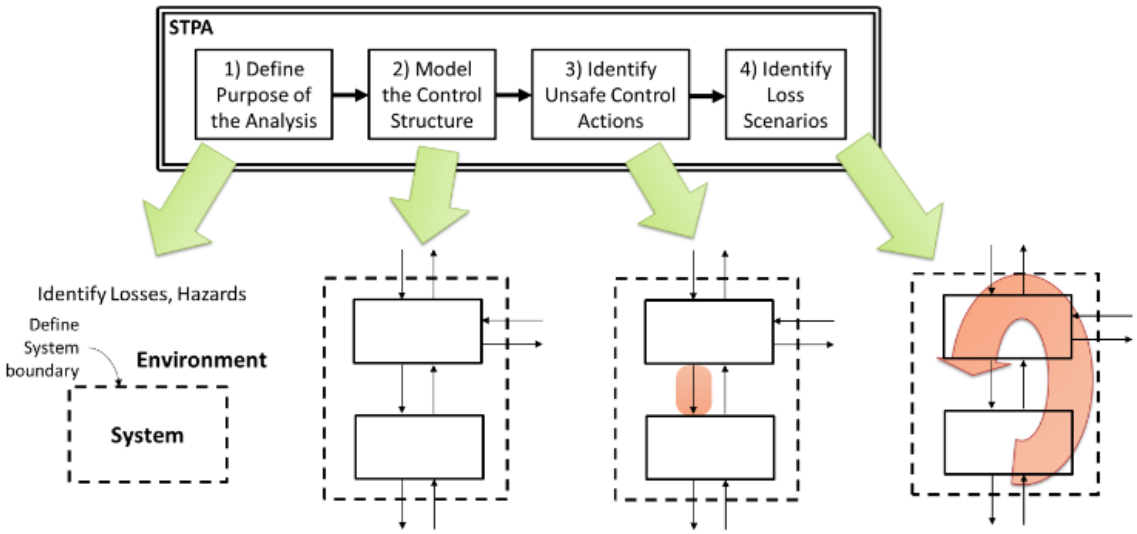

Figure 2. 47 depicts the STPA process. The figure is from the STPA handbook (Leveson, 2011)

Step 1: Identifying the losses and hazards.

The first step in any analysis is to document the project's specific goals. In an STPA, that means defining the system-level losses and hazards (N. Leveson, 2011).

Losses.

Losses are defined as anything of value to the stakeholders. While in most systems, the primary loss is injury or death to people, stakeholders can consider other types of losses. Typical losses include harm to the surrounding environment, loss of trust, and monetary losses.

Hazards.

Hazards are defined as "A hazard is a system state or set of conditions that, together with a particular set of worst-case environmental conditions, will lead to a loss" (N. Leveson & Thomas, 2018). Hazards must be connected with one or more losses (N. G. Leveson et al., 2012). If a hazard does not lead to a loss in certain environmental conditions, then it is not a loss. For traceability between the steps of the STPA, it is helpful to number both the losses and hazards, and for each hazard, denote which loss it is tied to.

Step 2: Building the control structure model.

Step two of STPA builds a model of the feedback-control relationships in the system. These models are referred to as control structures.

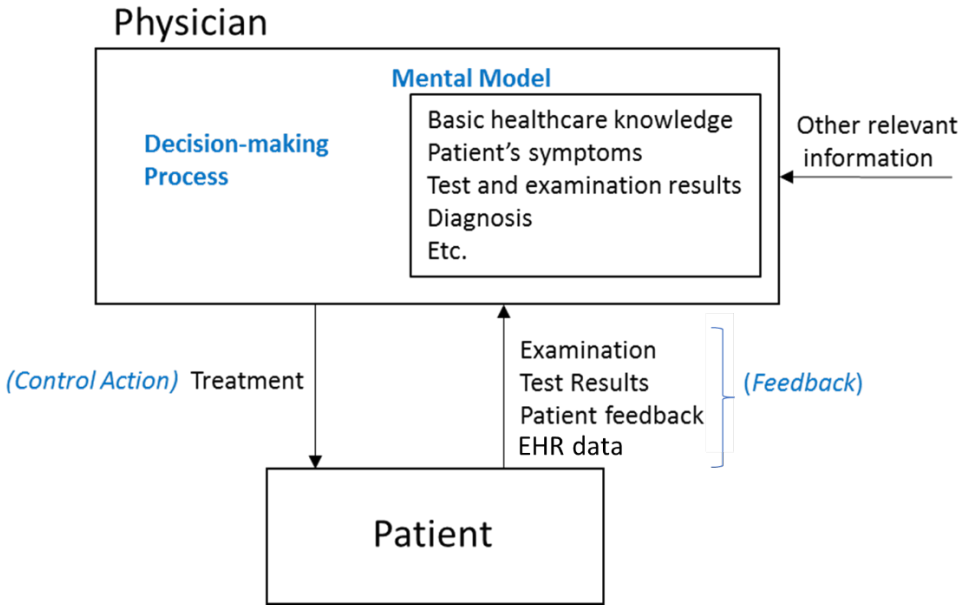

Figure 2. 56 depicts a basic feedback-control loop in a healthcare system between a physician and a patient. This figure originally appeared in (Leveson et al., 2023)

Figure 2. 50 shows an example of a basic feedback control loop: a controller at the top (a physician) and a controlled process at the bottom (a patient or the patient's health). The physician has the control action "Treatment" that he or she can use to impact the patient's health. The physician informs the selection of a treatment action using feedback from the patient, including examinations, test results, patient comments, and data from Electronic Health Records (EHRs). The feedback informs the physician's mental model of how the patient is doing. The mental model (or process model for non-human controllers) is also influenced by factors such as training, basic healthcare knowledge, or previous diagnosis information.

To build the model, other feedback-control loops that impact the safety outcomes of the controlled process are identified and combined. These models are referred to as control structures.

Models will never be complete in that they will never represent the system entirely (N. Leveson, 2023; N. Leveson et al., 2023).Therefore, the object is to create an acceptable and useful model of the system.

Step 3: UCA identification.

Once the control structure is established, identifying Unsafe Control Actions (UCAs) is straightforward. For each controller and list of control actions, the analysts must consider the contexts in which the various control actions would become unsafe.

The STPA handbook defines a UCA as "a control action that, in a particular context and worst-case environment, will lead to a hazard" (N. Leveson, 2011). Proper UCA syntax requires

a controller, a type, a control action, and the context that makes the control action unsafe. Figure 2. 63 below shows the four types of UCAs.

Figure 2. 63 Shows the various kinds of UCAs using the example of a medical practitioner providing treatment to a patient

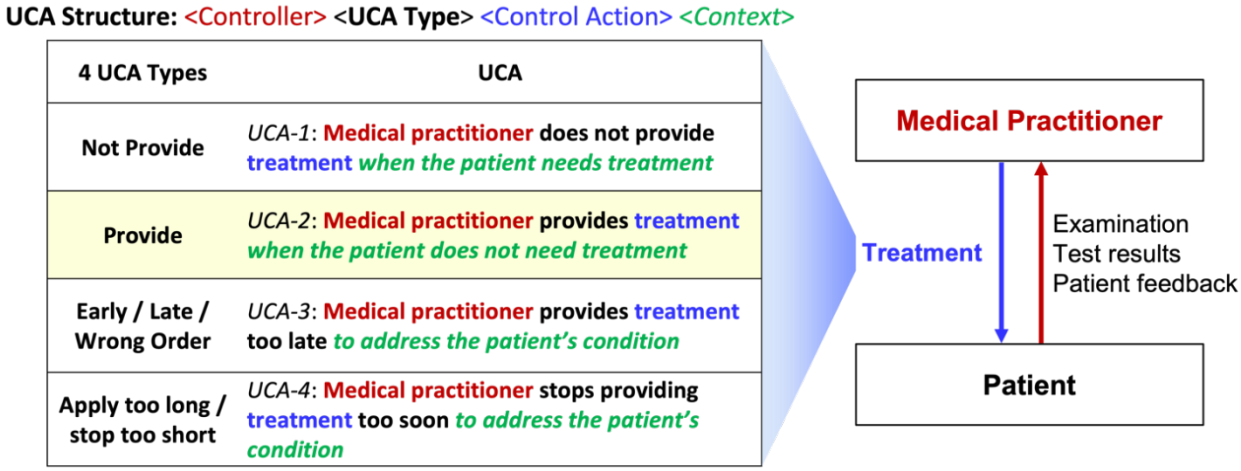

Figure 2. 63 shows the various components of a UCA and the different types of UCA using the example control loop of a medical practitioner and a patient. This figure originally appeared in (Leveson et al., 2023)

Figure 2. 63 also shows four examples of UCAS that were generated from a medical practitioner's "provide treatment" control action on a patient.

To generate UCAs from the control structure, the control actions of each controller are reviewed. For the simple control loop in Figure 2. 63, the only modeled control action is “provide treatment."

First, for each control action, contexts in which not providing the control action could be unsafe are identified. For example, Figure 2. 63 lists the UCA, "Medical practitioner does not provide treatment when the patient needs treatment.”

The other three categories of UCAS are providing the control action in an unsafe context, providing the control action too early, too late, or in the wrong order, and applying the control action for too long or stopping the control action too early. Each of these four types must be considered for each control action in the model.

Step 4: Identifying causal scenarios.

Once UCAs are generated, causal scenarios are identified. Given the current system design, scenarios take a UCA and explain why it may reasonably occur. For example, why would the UCA, as shown in Figure 2. 63, "Medical practitioner does not provide treatment when the patient needs treatment," reasonably occur?

At a basic level, scenarios describe why a control action that turns out to be unsafe was selected or why a safe control action was not correctly executed (N. Leveson & Thomas, 2018). These two branches of scenarios have many different subcategories, including incorrect mental models, conflicting feedback, or misaligned incentives.

2.4 Conclusion

This chapter reviewed common methods for hazard analysis in sociotechnical systems, exploring the gaps and difficulties within these methods to identify what an improved method would require. STPA was introduced as a method that addresses the identified gaps. However, STPA can be further improved for applications to sociotechnical systems by explicitly including processes for identifying scenarios using HFE. In the next chapter, STPA will be discussed with additional processes to make applications to sociotechnical systems more thorough.

Chapter 3: Human Factors in STPA

This chapter outlines the process of Systemic Technological Process Analysis (STPA) applied to a sociotechnical system and provides techniques for incorporating Human Factors Engineering (HFE) concerns into the analysis. The first three steps of STPA are briefly described, but they remain largely unchanged when applied to a sociotechnical system. The significant contribution of this thesis, and this chapter specifically, is the process for identifying causal scenarios by considering human requirements in each part of a control loop.

3.1 Losses and Hazards

The first step of STPA is to define the losses and hazards that must be prevented. The process of identifying losses and hazards for a sociotechnical system is nearly equivalent to identifying losses in technical systems.

3.1.1 Losses

Losses are system-level outcomes that the stakeholders want to prevent. STPA is a top-down analysis method that only identifies causal scenarios that could lead to the specified losses. The results of the analysis, therefore, are dependent on what losses are selected. For example, an STPA analysis could focus only on loss of life or only on the loss of customer satisfaction. While there may be overlap between the results of the two studies, the hazards, UCAs, and Scenarios could be largely distinct. System stakeholders may focus on a specific loss, even if they understand that other losses are significant.

For most safety analyses, the primary loss is loss of life or injury. Most stakeholders will find harm to humans relevant and potentially directly tied to other concerns, such as monetary or reputation loss.

For a healthcare system, losses considered may include:

L-1 Loss of life or injury to patients

L-2 Loss of life or injury to employees

L-3 Damage to equipment or facilities

L-4 Loss of reputation

L-5 Loss of financial viability

3.1.2 Hazards

The next step is to identify system hazards. Hazards are system states that could lead to a loss in a worst-case environment.

Like losses, hazards must be kept at the system level. Therefore, hazards cannot include any specific system component, such as employees, equipment, or IT infrastructure. They may,

however, include system outputs. For example, a hospital may consider a hazard relating to patients receiving inadequate care. Additionally, hazards should never include device failures or decisions made by individual controllers. Two reasonable hazards for a healthcare system may be:

H1. Patients receive less than the acceptable standard of care

H2. Patients lose trust in the healthcare system

3.2 Control structure

Once the losses and hazards are determined, the control structure is modeled. Because models cannot include every detail of the system, deciding what to include or not include in the model is a critical choice. The level of abstraction needed depends on the context of the analysis. The level of abstraction manages the model's "apparent complexity" (Rasmussen, 1985). If too many components and relationships are modeled, the apparent complexity may be too high to understand the system meaningfully. If the model contains too few elements, an insufficient number of unsafe interactions may be identified. Different levels of abstraction may be useful for different system analyses.

Figure 3. 1 shows a simple control structure that shows many of the relationships in sociotechnical systems. Typically, there are regulatory groups or other government-level controllers at the top. Below the regulators are the regulated organizations that directly interact with the controlled process. Figure 3. 1 shows a control structure where all control-feedback loops exist and are complete. However, real systems may lack entire control loops or adequate feedback control channels.

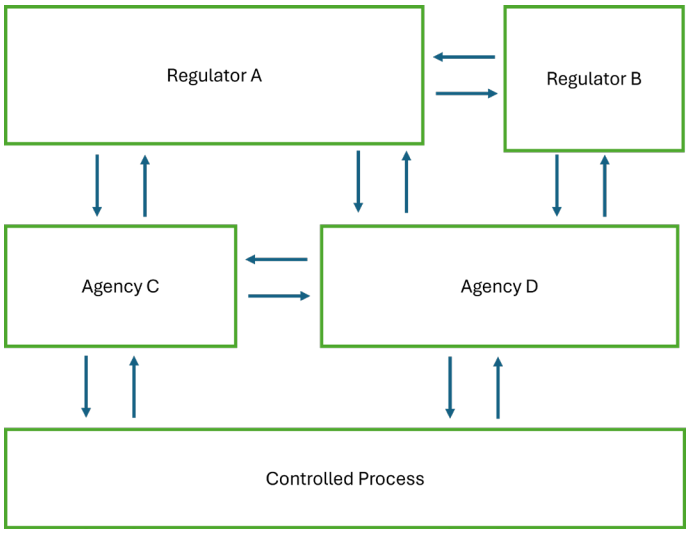

Figure 3. 1 depicts a high-level generic control structure of a sociotechnical system.

One of the challenges to modeling sociotechnical systems is that, usually, no individual or organization intentionally designed the systems in their entirety. Instead, the systems evolved slowly and have a tangled web of interconnections (Nemeth, 2004). For example, healthcare as an industry evolved over hundreds (if not thousands) of years(Perry et al., 2021). Within the

United States, each regulatory body was created at a different time to solve a specific problem (Smith, 2023). However, as technology evolved, the boundaries between the regulatory sectors changed. For example, medical devices are regulated by the Food and Drug Administration (FDA). For many years, medical devices such as blood pressure cuffs or stethoscopes did not contain computers or software. Then, as digital medical records were growing in use, the federal government created a new agency regulating medical health information technology (the ONC/ASTP). However, more and more medical devices now include software components, resulting in overlap and gaps between the jurisdictions of the two agencies (FDA et al., 2014; N. Leveson et al., 2023).

One of the first challenges is determining how detailed the control structure must be to identify hazards in the given system adequately. If the control structure includes too much detail, it reduces the intellectual manageability of the model, making it difficult for people to understand and interpret the depicted information. If the control structure is too simplistic, it will not capture critical safety-related interactions. This problem can be solved by using different levels of abstraction, depending on the specific question being asked. There is no reason that only one model or one level of abstraction must be used to answer all questions.

Abstraction choices may be easier to understand through a more ubiquitous model—the map. Every map is a model of a geographical system. However, maps appear remarkably different depending on the specific problem being solved at any given time.

For example, navigational maps on phones in driver view mode present a limited set of data points relevant to a driver. These maps often show street names and the approximate size and shapes of each block, but do not give details about the buildings on the streets or the level of incline of the road. When following driving directions, the map may include information about stoplights and highlight the suggested route. However, limited information about the geographical area is included. Some drivers may wish for more details to be included, while others may think the map is too crowded with unnecessary information. However, most people can navigate unfamiliar locations using this system model.

A subway map, on the other hand, looks completely different. Although it represents the same location as the driving map, the size and shape of the individual blocks are distorted or removed to convey the relationship between the subway lines. This model helps subway passengers select the correct line and direction. Additionally, because most cities have chosen a similar way of communicating subway information, passengers can quickly understand how to move around a city and transfer between lines in any city they travel to (Kent, 2021). However, by limiting the scope of information, some subway passengers may not know how close stops are to each other above ground. Including the necessary details to convey the distance between stops to passengers may help some passengers navigate, but could cause confusion and reduce comprehensibility for many more.

In both maps, abstraction is necessary to make the model valuable. In addition, no objective "correct" model perfectly represents the system (a city, in this example). Maps include or exclude various pieces of information, depending on the intended use of the map. Models should only be

as detailed as needed to facilitate the decision-making and problem-solving required for the users (Machol & Miles, 1973). Different models and levels of abstraction are necessary to solve different problems.

3.3 Unsafe Control Actions

As with steps one and two, the process of identifying Unsafe Control Actions (UCAs) in a sociotechnical system is similar to that of a technical system.

In step three of STPA, each controller's available control actions are evaluated to understand in which contexts the control action would be unsafe. UCAs are defined as "a control action that, in a particular context and worst-case environment, will lead to a hazard" ((N. G. Leveson et al., 2012).

As described in Chapter Two, each UCA has a specific set of components: the control action, the UCA classification, and the context (N. G. Leveson et al., 2012). The different categories of control actions are providing the control action, not providing the control action, providing the control action with unsafe timing or sequencing (too early/too late/out of order), and providing the control action with unsafe duration (stopped too soon, applied too long). These four categories cover the complete set of possible ways each control action could lead to a hazard (N. G. Leveson et al., 2012).

Not all control actions will have UCAs for every category. For instance, some control actions are discrete and do not have a duration. One example of a discrete control action is the FDA's audit of a medical device company. An audit is either conducted or not; audits cannot be applied for too long or too short. Problems that arise from incomplete or insufficient audits will be captured at a later point in the analysis. Therefore, there will be no UCAs for the last category.

UCAs are identified by evaluating the context of each control action of every controller. Table 3. 1 shows an example list of UCAs for a physician in a healthcare system.

Table 3. 1 shows a subset of UCAs for a physician in a healthcare system

Control Action	Applied	Did not Apply	Too late	Too Long
Provide Treatment	UCA 1.1: Provided treatment when patient did not need treatment. UCA 1.2: Provided treatment for condition that patient did not have	UCA 1.3: Did not apply treatment when patient needed treatment	UCA 1.4 Applied treatment too late to address patient's condition.	UCA 1.5: Applied treatment too long after condition was mitigated. UCA 1.6: Stopped treatment before condition was mitigated

3.4 Scenarios

Scenarios explain why a controller would reasonably apply a UCA to a system. The process of identifying scenarios in sociotechnical systems relies on an understanding of how humans behave in complex systems. Therefore, HFE expertise is necessary for thorough scenario identification in sociotechnical systems. In this section, a process for using HFE to identify causal scenarios is presented. The new process expands on a new method for developing causal scenarios using four high-level scenarios.

3.4.1 New approach to scenario generation

A new approach to identifying scenarios has been developed to formalize the process of scenario identification. This new approach generates exactly four high-level scenarios for each UCA, then evaluates each high-level scenario to identify specific causal scenarios (Thomas, 2024). The four classes of scenarios are defined using output and input functions. Each of the four high-level classes of scenarios is described in the sections below.

Class One

In class one scenarios, the controller receives feedback that correctly depicts the state of the system but executes the specific unsafe control action anyway.

The general high-level archetype of this scenario is:

- Output: UCA (<Controller> provides <Control Action> when <Context>)
- Input: <Input> correctly showed that <Context>

To use the generic archetype, the variables within the archetype are replaced with the context of the UCA in question. For example, for the UCA: “FDA does not audit device manufacturer when its devices are performing below set standard,” the variable <controller> is replaced with FDA, the variable <control action> is replaced with “does not audit” and the variable <context> is replaced with “devices are performing below set standard.” In this example, the only variable not given by the UCA is <input>, which in this example might be “device performance reports.”

Therefore, the example class one scenario for the same UCA is:

- Output: FDA does not audit a device manufacturer when its devices are performing below the set standard
- Input: Device performance reports correctly showed that devices are performing below the set standard

The model of a class one scenario is depicted below in Figure 3. 2.

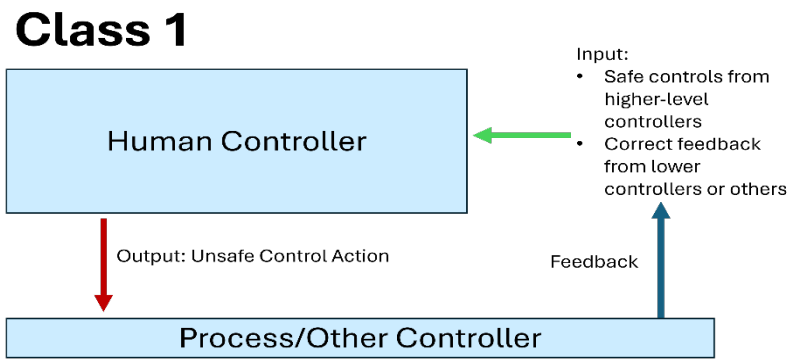

Figure 3. 2 depicts a class one scenario.

Class Two

In class two scenarios, controllers receive feedback that does not adequately represent the state of the system and execute an unsafe control.

The general high-level archetype of this scenario is:

- Output: UCA (<Controller> provides <Control Action> when <Context>)
- Input: <Feedback/Input> to <Controller> does not adequately indicate <Context>

The example high-level class two scenario for the UCA "FDA does not audit a device manufacturer when its devices are performing below the set standard" is:

- Output: FDA does not audit a device manufacturer when its devices are performing below the set standard"
- Input: Device performance reports sent to FDA do not adequately indicate that the devices are performing below the set standard"

The model of a class two scenario is depicted below in Figure 3. 3.

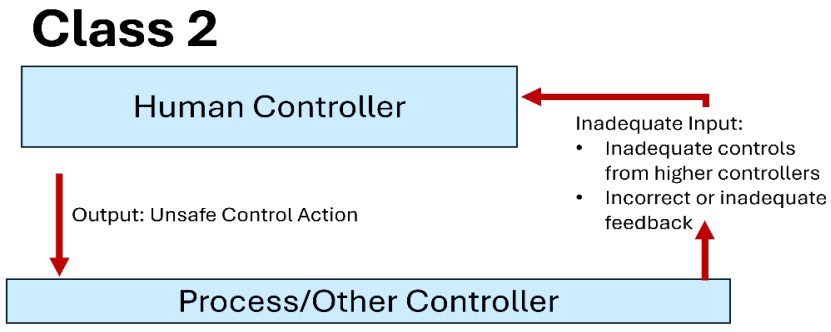

Figure 3. 3 depicts a class two scenario

Class Three

Class three scenarios are focused on the controller output and the control path. In a class three scenario, a controller provides a safe control action, but the process receives a UCA.

The general high-level archetype of this scenario is:

- Output: <Controller> does not provide <UCA> but <Process> receives <UCA>

The example high-level class three scenario for the UCA "FDA does not audit a device manufacturer when its devices are performing below the set standard" is:

- Output: FDA does provide instructions to conduct an audit, but the manufacturer is not audited when its devices are performing below the set standard.

The model of a class three scenario is depicted below in Figure 3. 4.

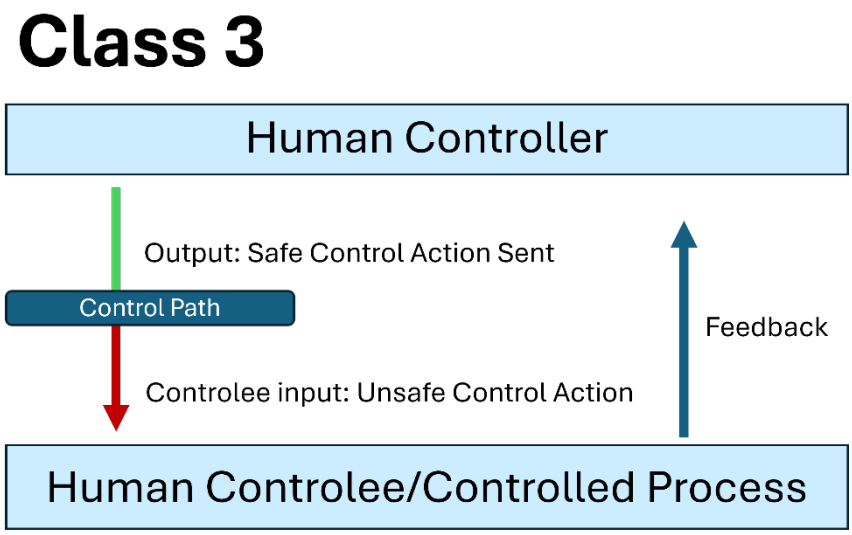

Figure 3. 4 depicts a class three scenario

Class Four

Class four scenarios are focused on the controlled process or controlled entity. In class four scenarios, the process does not receive a UCA. However, the process acts as if a UCA had been provided.

The general high-level archetype of this scenario is:

- Controlee/Process Input: <Safe Control Action (SCA)>
- Controlee/Process Output: <Process> provides <UCA>

The example high-level class four scenario for the UCA "FDA does not audit a device manufacturer when its devices are performing below the set standard" is:

- Controlee/Process Input: FDA audits a device manufacturer when its devices are performing below the set standard."

- Controlee/Process Output: Manufacturer continues to produce unsafe devices.

The model of a class four scenario is depicted below in Figure 3. 5

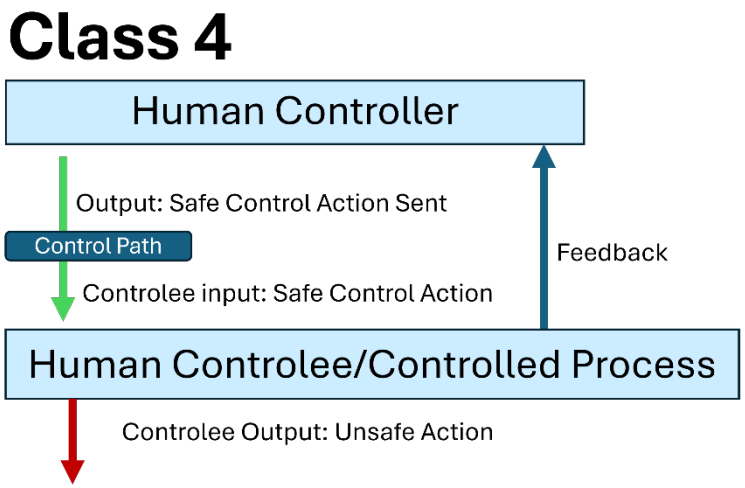

Figure 3. 5 depicts a class four scenario

3.4.2 Controller Models

To utilize the new approach to scenario generation, Thomas uses the generic controller model shown in Figure 3. 6 (2024). The model models the way in which a controller uses inputs from the environment and the system to arrive at control decisions to understand how a controller could reasonably select an unsafe control action.

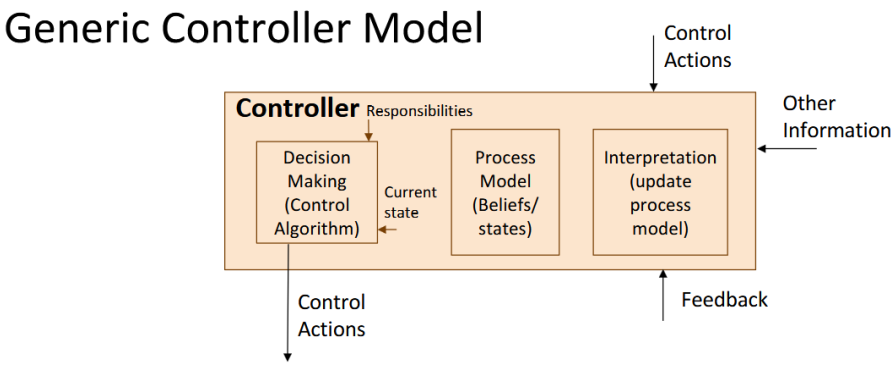

Figure 3. 6 depicts a generic controller model from (Thomas, 2024)

The six components of a generic control loop depicted in Figure 3. 6 are explained below:

Responsibilities

These are the specific tasks or processes that a controller is obligated to do in accordance with its role. For example, an insulin dispenser has the responsibility to provide insulin dosages as needed, a doctor has the responsibility to provide treatment to patients who need it, and a

hospital financial manager has the responsibility to ensure that the hospital's income is sufficient to cover costs.

Decision-making algorithm

The method by which the controller selects which control action will fulfill its responsibilities.

Process model and beliefs

The model of the system that the controller uses to understand the current state of the system and how the system will respond to different controls.

How process models are updated

The process by which a controller uses information from the environment or other controllers to maintain the accuracy of its process model.

Current state of the system

The attribute that determines how the controller responds to input. In software systems, the current state is often the system mode. For example, an automated controller for a defibrillator may provide a shock if it is in "emergency" mode and the user presses "start." If the defibrillator is in "tutorial" mode, the same "start" input will not result in a shock (Montague & Verdeja, 2021).

Control actions from other controllers, feedback, and other information

Any information from the system or environment that impacts controller behavior.

The controller model in Figure 3. 6 is generic enough to represent both technical and human controllers. However, to aid scenario identification for human controllers, the generic controller model is adapted to a more specific human controller model shown below in Figure 3. 7.

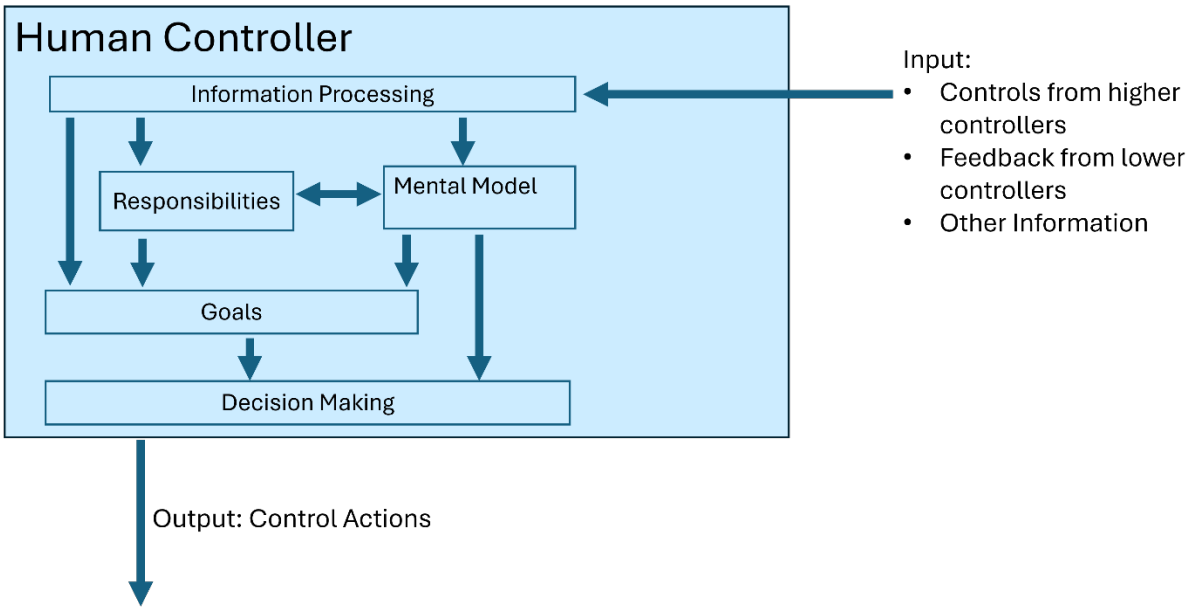

Figure 3. 7 depicts a generic human controller model

While the novel contribution of this chapter is how the model is used to identify scenarios, the model in Figure 3. 7 has three significant differences from Thomas's generic controller model:

Interpretation is changed to information processing, which is how human factors researchers refer to models of information flow in humans (Proctor & Van Zandt, 2018). Information processing models enable analysis of if and how humans receive and process sensory input. Human information processing is distinct from computerized controllers' interpretation because, unlike computers, which can be programmed to use every piece of data they receive, humans can dynamically select what information to pay attention to but are not able to attend to or interpret all the data they receive (Wickens & Carswell, 2012).

Human controllers' process models are referred to as mental models. The mental model is how humans store their understanding of the system, or system components, in their memory (Rasmussen, 1987). Mental models in humans are dynamic and developed by experience and training, whereas a technical controller's process model may be programmed and static or otherwise controlled by the system design.

The current state of a human controller is replaced with the human controller's current goal. The state of an automated controller changes how it responds to input. For example, a vending machine will not dispense a product if it is in the "unpaid" state, even if an item is selected. If money is inserted into the vending machine and it changes to the "paid" state, the machine will dispense a product if the same item selection is entered. Humans do not have equivalent states that can be modeled usefully; humans only have two states: consciousness and unconsciousness. Instead, humans change how they respond to input based on their goals (Carayon et al., 2012). For example, a clinician with the goal of cost reduction may select a different treatment than a clinician with the goal of maintaining a relationship with a specific pharmaceutical company, even if the two physicians receive the same information and input from their patient (Zarei et al., 2023). Different inputs may change which goal a controller is prioritizing at any time.

In the following sections, the process for using the human controller model in Figure 3. 7 to identify low-level scenarios for each of the four scenario classes using HFE is explained.

3.4.3 Identifying scenarios using the human controller model

Each of the four low-level scenarios must be further analyzed to identify detailed lower-level causal scenarios with more system context to describe why they could have occurred. This section reviews HFE topics relevant to each of the six components in the human controller model in order to explain why a controller would find it reasonable to select an unsafe control action.

For each UCA, first, the four scenarios are defined. Then, the process provided in this section is used to expand the four scenario classes into more detailed causal scenarios. To avoid repetition, only the additional context of the scenarios is provided in the low-level scenario archetypes that follow.

For example, the full Class One scenario archetype is:

- Output: <Controller> provides <Control Action> when <Context> because <explanation>
- Input: <Input> correctly showed that <Context>

The provided low-level scenario archetypes only represent the expanded <explanation> variable.

The low-level scenario archetypes include new variables that were not used in the high-level scenario class definition archetypes. All archetype variables used in the scenario archetypes are defined in Table 3. 2 below. The bold variables are pre-defined by the UCA, the variables in italics must be identified by system experts based on the context of the UCA and the system:

Table 3. 2 defines the variables used in the scenario archetypes provided in the subsequent section

Variable	Significance
<UCA>	Unsafe Control Action.
<SCA>	Safe Control Action. This is the safe control action given the context from the base UCA
<Controller>	The entity that provides the UCA being analyzed
<Controlee> or <process>	The receiver of the UCA. The controlee can be the process being controlled by the system or another controller.
<superior controller>	A higher-level entity that sends a control action to the controller
<peer controller>	An entity in the system that does not have control over the controller but may have overlapping responsibilities with the controller.
<input>	The relevant feedback or environmental data
<goal>	The system state that the controller is trying to achieve
<system goal>	The system state that the system as a whole is trying to achieve

Controller Responsibilities

Controller responsibilities are the tasks or processes that a controller is obligated to do, in certain contexts and in accordance with their role. Causal scenarios that originate from inadequate responsibilities involve the specific controls that a controller has access to and whether the controller has sufficient authority and accountability to make safe control decisions in all system contexts where necessary. Two main categories of responsibility-based scenarios exist: 1) controllers have insufficient responsibility or authority, and 2) multiple controllers share a responsibility.

Inadequate control authority

In order to provide a safe control action, a controller must have the responsibility and authority to make that action. If a controller does not have the authority to make the correct control decision given the system context, they will make an unsafe decision regardless of the quality of feedback or their ability to problem-solve. For example, a hospital manager may not have the authority or responsibility to hire additional nurses, even if they receive feedback that shows that they have insufficient staffing.

Detailed scenario archetypes stemming from inadequate control authority are provided in Table 3. 3.

Table 3. 3 contains scenario archetypes related to inadequate control authority.

Class	Detailed Causal Archetype
One	<Controller> did not have the responsibility to <SCA> given <Context> indicated by <Input>
	<Controller> had the responsibility to execute <UCA> regardless of <Input>
Two	The <Controller> did not have the responsibility to question the <Input>; instead, it had the responsibility to make control decisions based on the <Input>.
	<Controller> knows that <SCA> is necessary. However, they believe that no one else has executed the <SCA> yet, but the <Peer Controller> has. The control action may be unsafe if duplicated.
	<Controller> has the responsibility to verify <Input> before making a control decision. However, the <Controller> may rarely encounter errors, so they may skip the verification step to save time.
	<Controller> knows that <SCA> is necessary. However, because of <Input>, they believe that it has not been executed by <Peer Controller>, but <Peer Controller> has already done so.
	<Controller> knows that <SCA> is necessary. However, they believe that it has already been executed by <Peer Controller>, but <Peer Controller> has not.
	The <Controller> has the responsibility to request updated <Input>, but does not realize that their <Input> is outdated.
Three	<Control path> only sends control actions after they are verified by another <Controller>who disapproved of the <SCA>

Four	<SCA> is outside of the responsibilities of <Controller> so <SCA> is ignored by <process>

Shared Responsibilities

Other scenarios that may arise from inadequate allocation of controller responsibilities involve controls that are shared between multiple controllers. Shared responsibilities can lead to unsafe control selection when controllers are unsure who is responsible for executing a control action in each context. If every controller assumes that someone else will take responsibility, the control action may not be executed at all. In large sociotechnical systems, individuals often make assumptions about the responsibilities of external organizations or other people (Dewar, 2002). In a hospital, for example, many members of the IT department may be aware that a specific control is needed, such as implementing a software update. However, if everyone assumes that someone else is responsible for the update, the update will not happen (N. Leveson et al., 2023).

In addition to controls never being implemented, many control actions require coordination between multiple people or organizations. Lack of coordination can cause hazards when the actions of different controllers are canceled out or otherwise negatively interact with each other. If controllers are unaware of the responsibilities and actions of other controllers, they could lose coordination and select unsafe control actions. Moreover, sociotechnical systems adapt and change over time. Therefore, responsibility allocations that were safe in the past must be updated and coordinated as the system changes.

Detailed scenario archetypes stemming from shared responsibility are provided in Table 3. 4.

Table 3. 4 contains scenario archetypes related to shared responsibility.

Class	Detailed Causal Archetype
Class One	<Controller> knows <SCA> is needed but believes that <Peer Controller> is responsible for executing <SCA>. The control action may be unsafe if duplicated, so <Controller> does not execute the control.
	<Controller> knows that <SCA> is necessary. However, they believe that it has not already been executed by <Peer Controller>. <Peer Controller> executed the control action, but there is a time delay on the system impact. <Input> may only indicate whether the effect has occurred, rather than whether the control itself has been engaged.
Class Two	<Controller> uses <Input> to determine whether a control has been executed by others in the system. It may be possible for another <Controller>to execute the control action without changing <Input>.
Class Three	<Controller> does not execute <UCA> but another <Controller>enacts it anyway.

Class Four	<Controlee> has a default setting that may be unsafe if no controls are provided by any controller.

Mental Models

While there are many definitions of the term mental model (Rasmussen, 1987; Rouse & Morris, 1985), this thesis uses the definition given by Rasmussen that mental models "are used to predict future events and responses of the environment to human actions; to find causes for observed events; to determine proper changes in the environment to obtain desirable responses" (Rasmussen, 1987, p. 10). In essence, mental models refer to the way humans store their understanding of system behavior in order to predict future system states and identify appropriate actions (Rasmussen, 1987). Humans' ability to safely select control actions relies on whether their mental model of the system correctly matches the behavior of the real system.

Mental models are critical for humans to both use feedback to understand the current system state and to help run internal "what-if" hypothesis tests, where different inputs are tested and potential outputs are compared without making any changes to the system itself (Rasmussen, 1987; Rasmussen et al., 1990; Sharit, 2012).

A well-developed mental model, created through experience and training, can reduce the required effort to maintain control over the system (Endsley, 1995). Mental models help humans identify the current system state by matching current perceptual data to previous system behavior. Humans can use mental models to find close matches even if the available information is incomplete or inconsistent (Endsley, 2012).

Scenario archetypes from inadequate mental models are provided in Table 3. 5.

Table 3. 5 contains scenario archetypes related to inadequate mental models.

Class	Detailed Causal Archetype
Class One	<Controller> is unable to identify the correct control action associated with <Input>. The <Controller>may not have sufficient experience to have a well-developed mental model or may be stressed/distracted/fatigued, etc.
Class Two	The <Controller>'s mental model is that <Input> is a direct indication of system status; however, the <Input> is a measure of a different construct that may not always align.
Class Three	The <Controlee>'s mental model of the system leads them to believe that the <SCA> is unsafe, so they do not adhere to it.
Class Four	<Controlee> interprets the control in a different way than was intended by the <Controller> due to mismatched mental models.

Memory and Recall

Mental models use a combination of short-term and long-term memory. Humans have a theoretically infinite long-term memory (Wang et al., 2003). However, creating and storing

mental models in long-term memory in ways that can be readily accessed and used is highly dependent on training and experience (Gobet & Simon, 1998). Information existing in long-term memory does not guarantee that information will always be retrieved in the appropriate context (Dismukes, 2006). Experience and training differentiate between a human who can recall the appropriate information at the right time and one who cannot.

When operators lack a substantial, pre-existing mental model in their long-term memory that is easily accessible, they rely more heavily on their short-term memory. Short-term memory is extremely limited, and information is easily lost from short-term memory when people become distracted or their attention is redirected (Endsley, 1995).

Familiarity with a system increases the amount of information a human can keep in their short-term memory because they can chunk the perceptual information into denser blocks (Gobet & Simon, 1998). For example, an experienced doctor may see a patient's chart and be able to keep a set of symptoms as one block in short-term memory if those symptoms are often clumped and fall under a typical diagnosis. A doctor working outside of their specialty may need to store each piece of diagnostic data separately, allowing them less capacity to store other information.

Scenario archetypes that involve memory and recall are shown in Table 3. 6.

Table 3. 6 contains scenario archetypes related to memory and recall.

Class	Detailed Causal Archetype
Class One	<Controller> has limited familiarity with the system and takes too long to identify what perceptual cues are useful for addressing the current system context. The <Controller>'s training did not prepare them to identify the safe control action when <Input> emerged. This context was not covered in the training due to the <Context>. Over time, <Controller>'s mental model shifted to relying on <Input> to determine their action selection. <Controller> may not have experienced a system state where <Input> was accurate, but other forms of feedback were necessary to make a safe decision. The decision was needed quickly, and <Controller>'s mental model required more cognitive resources than they had available at the moment. The <Controller> had not experienced this <Context> before, but they had experienced the same <Input> before. Their mental model may therefore be unaware that the <Input> could correspond to multiple system states.
Class Two	The <Controller>'s mental model relied solely on <Input> as a decision-making factor because they could not recall other <Inputs>.

Class Three	
Class Four	<Controlee> receives <SCA>, but the <SCA> may be generic, and the <Controlee>is unable to translate the general advice into their mental model of their system.

Information Processing

It is not sufficient for a human controller to have a detailed and robust mental model; they must also continually update their mental model as the environment and system context change and evolve over time. Humans cannot directly utilize every piece of data in their environment; they have limited cognitive resources that must be split over perceiving data using sensory organs (sight, hearing, touch), directing attention to the sensory input in order to understand it, and selecting what information to use to update their mental models of system behavior (Baddeley & Hitch, 1974). HFE researchers refer to this process as information processing. Human information processing is a mostly subconscious process, but the limitations at each step have important implications for when and if a mental model is adequately updated (Wickens & Carswell, 2012).

Interpretation of available feedback

In order to adequately maintain an accurate mental model, humans must not only constantly perceive critical information from their environment, but also direct cognitive resources to attend to the sensory data in order to comprehend its significance (Baddeley & Hitch, 1974). Experiments have shown that humans store a limited amount of sensory data in working memory, even if they are not paying attention to the input (Endsley, 1995). For example, students attending lectures are often able to recall the last sentence their professor said if questioned, even if they were not paying attention. However, the same student may be unable to recall what was said a few sentences earlier because information stored in working memory is quickly erased if it is not attended to (Baddeley & Hitch, 1974).

The decision to attend to a sensory input is not always a conscious decision. Humans have a limited amount of cognitive bandwidth to dedicate to tasks at any given time. If humans are focused on a task, for example, they may not have sufficient cognitive resources to allocate to the interpretation of new data (Endsley, 2012; Wickens & Carswell, 2012). Therefore, even if a display changes information or an alarm goes off, the human will not necessarily have the capacity to interpret or comprehend the change.

One factor that determines the speed and accuracy of humans' ability to condense perceptual information into an understanding of the current system state is the training and experience of the human. Experience allows humans to quickly sort through perceptual information and match it with system states they have previously experienced (Gobet & Simon, 1998). Therefore, experts exert far fewer mental resources to understand the perceptual information they have received and have more cognitive resources free to attend to unexpected input (Proctor & Van Zandt, 2018). Unsafe control decisions may be caused by insufficient training requirements or inadequate motivation for lower-level controllers to stay in their positions long enough to obtain experience.

For example, suppose a new nurse is attending to a patient when multiple alarms go off. The nurse will spend significant mental resources perceiving all the information suddenly flooding his environment. The nurse will, therefore, have reduced cognitive resources remaining to understand the patient's status based on the various alarms. As a result, the nurse may have insufficient cognitive resources to predict the patient's future status based on the patient's current status. An experienced nurse, on the other hand, may have experienced this combination of alarms before. Therefore, fewer mental resources will be expended by the nurse on perception and understanding. The experienced nurse will have more bandwidth to predict the patient's future status and determine the necessary intervention. In this example, one of the critical differences between the new nurse and the experienced nurse is the comparative detail of their mental models.

Furthermore, a controller's access to data and information is often influenced by others in the system. Controllers must ensure they provide appropriate levels of feedback or access to information to others in the system when possible. If a controller receives too much data, they may not be able to identify the most critical information and may expend too many mental resources on irrelevant details. While there is often a belief that more information is better, decision-making becomes worse if too much information is provided, especially during stressful situations (Wickens et al., 2013). On the other hand, if too little information is provided, the controller may not be able to identify changes in system state.

Scenario archetypes that result from the interpretation of feedback are provided in Table 3. 7.

Table 3. 7 contains scenario archetypes related to interpreting feedback.

Class	Detailed Causal Archetype
Class One	<Controller> had an accurate mental model before a system change; however, once the system behavior changed, the controller's mental model did not. Therefore, they interpreted <Input> incorrectly. <Controller> was inundated with <Input> and was unable to identify what was causing the system to change states. There may have been no direction from the system to guide the response or interpretation of the <Input>. The <Controller>'s mental model did not update when the <Input> changed because they were focused on another source of <Input>. The <Controller>'s mental model did not update when the <Input> changed because it conflicted with their initial hypothesis of the system state, and the <Input> was not salient enough to prompt a change in their hypothesis. The <Controller>'s mental model did not update when the <Input> changed because other feedback sources were showing conflicting information, and the <Controller> was

	overwhelmed and could not determine which <Input> was the most relevant.
Class Two	The <Controller> was overwhelmed with <Input> data and focused solely on <Input> to maintain their focus, but was unable to recognize that <Input> conflicted with other data sources. <Controller> had no other forms of <Input> to challenge the information provided by <Input>. <Controller> believed that the inputs used to monitor the system state were based on different underlying data sources. However, there were underlying relationships between the Inputs such that if one was incorrect, the others were also incorrect. <Controller> believed <Input>, but the information was an indication that it was no longer reliable, for example, a dial reaching its maximum value.
Class Three	<Controlee> cannot receive the <SCA>, so the <SCA> was either mistranslated or ignored.
Class Four	<Controller> had an accurate mental model before a system change; however, once the system behavior changed, the controller's mental model did not. Therefore, they interpreted <Input> incorrectly. <Controller> was inundated with <Input> and was unable to identify what was causing the system to change states. There may have been no direction from the system to guide the response or interpretation of the <Input>. The <Controller>'s mental model did not update when the <Input> changed because they were focused on another source of <Input>. The <Controller>'s mental model did not update when the <Input> changed because it conflicted with their initial hypothesis of the system state, and the <Input> was not salient enough to prompt a change in their hypothesis. The <Controller>'s mental model did not update when the <Input> changed because other feedback sources were showing conflicting information, and the <Controller> was overwhelmed and could not determine which <Input> was the most relevant.

Salience

Because not all information in the environment will be interpreted, the human brain must determine which information is most valuable to pay attention to at any one time (Wickens &

Carswell, 2012). Humans select what to pay attention to using the salience of the sensory input or their expectations from previous experience (Wickens, 2002). Salience refers to the degree to which a perceptual cue sticks out from the environment, while experience governs where humans expect to locate critical information. A well-established mental model can direct attention to environmental stimuli with the highest expected value for relevant system information (Vidulich & Tsang, 2012).

For high-level decision-makers, there is often an abundance of system information available; however, some information may be more difficult to obtain or utilize. Decisions about what information to attend to may be based on a trade-off between the value expected from the information and the perceived level of effort to obtain the data (Wickens & Carswell, 2012). For example, hospital administrators may know that input from clinicians could improve their decision-making. However, the administrators may believe it would be too costly in time or money to solicit input from clinicians and rely instead on assumptions or readily available information.

Scenario archetypes stemming from the salience of feedback are provided in Table 3. 8.

Table 3. 8 contains scenario archetypes related to the salience of feedback.

Class	Detailed Causal Archetype
Class One	<Controller> may not have expected to find valuable information from <Input>; they may have developed a habit over time of relying solely on other sources of Input. <Controller> receives more <Input> from <Peer Controller> than others. They therefore develop a mental model that <Input> represents the state of the system. However, another <Peer Controller> may experience a different perspective but not have the time or resources to report.
Class Two	The most salient piece of <Input> available to the <Controller> was <Input> Obtaining an improved <Input> source may have been difficult or costly.
Class Three	<Controller> may have used an outdated control path mechanism to send the <SCA>. The old control path may still technically function, but may not be monitored as routinely.
Class Four	<Controller> issued <SCA> in a format that did not catch the attention of the <Controlee>. The control might have been buried in other less critical information, or in a format that <Controlee> believes usually does not contain useful information.

	<Controller> believes that another task is a higher priority. <Controlee> may not have made the importance of <SCA> clear enough to redirect the energy and attention of <Controller>

Biases

Humans are extraordinary pattern recognizers and can make informed inferences and deductions based on scant data. However, to manage complexity, human cognition uses biases and heuristics to make assumptions about systems (Tversky & Kahneman, 1974). Biases and heuristics are innate cognitive methods humans use to make quick judgments about their environment. Heuristics are critical for managing complex systems, but they are not always accurate. However, as expertise and experience increase, the heuristics and shortcuts humans employ become more accurate (Lehto et al., 2012).

One factor that can reduce the quality of decision-making is the speed at which the decision is required. Humans keep mental models at different levels of abstraction. Controllers may have a detailed mental model that can consider dozens of factors. However, controllers under time pressure may not have time to use such a mental model to run mental hypothesis tests (Rasmussen, 1990). Therefore, if a decision is rushed, the controller may need to use a simpler mental model that considers fewer contextual factors and is subject to more approximate heuristics.

One heuristic humans use to evaluate feedback is correlating event frequency with future event likelihood (Wickens & Carswell, 2012). For example, suppose a hospital has only experienced one outbreak of a dangerous strain of bacteria in the last decade. Hospital managers may not interpret signals of infection across different departments as a widespread problem that requires immediate intervention. Safety management systems must, therefore, calibrate alarms and event flagging carefully. Missed events can lead to clear harm, including accidents, while false alarms can lead to response delays and alarm fatigue (Wickens & Carswell, 2012). Alerts for events that are infrequent may need to be accompanied by supporting data so that controllers understand what is triggering the alarm.

Scenario archetypes stemming from human cognitive biases are provided in Table 3. 9.

Table 3. 9 contains scenario archetypes related to cognitive biases.

Class	Detailed Causal Archetype
Class One	<Controller> did not believe <Input> source because there was insufficient corroborating information, and the system state <Input> indicated was rare. <Controller> believed that <UCA> would address the <Context> because of training or education. <Controller> believes that <Input> requires <UCA> because the most recent incidents where <Input> was true, <UCA> was used.

	The <Controller> had less time than usual to make a decision. They may not have been able to consider all factors when making the decision.
Class Two	<Controller> believed <Input> because the system state it indicated was typical or expected.
Class Three	<Controller> sends <SCA>, but it is passed through a group that makes a change that they don't realize will change the impact of the <SCA>.
Class Four	<Controlee> received <SCA> but had not or rarely received this command previously and waited for confirmation to execute the requested action. <Controlee> did not verify system state indicated by <SCA> because it was a routine action

Hypothesis Testing

Sometimes, feedback and hypothetical tests alone are insufficient to keep a mental model current. When a controller does not understand the system's current state or how it would react to different inputs, humans often test hypotheses by making changes to the system to observe the results (Rasmussen, 1990).

For example, if a nurse is attending to a patient and the equipment goes dark, she may hypothesize that the power has gone out. She may try to turn on a light in another room to determine whether the problem is localized to her room. She may look out the window to see if other buildings have power. If the power is on in other locations, she may attempt to reset the equipment.

System designers may inadvertently limit the ability of system controllers to conduct hypothesis testing adequately, potentially leading to a loss scenario. Accidents often occur when humans cannot intervene before negative consequences from a test occur (Rasmussen et al., 1990).

Scenario archetypes that involve hypothesis testing are provided in **Error! Reference source not found.**.

Table 3. 10 contains scenario archetypes related to hypothesis testing.

Class	Detailed Causal Archetype
Class One	The <Controller>'s mental model did not update when the <Input> changed because it conflicted with their initial hypothesis of the system state, and the <Input> was not salient enough to prompt a change in their hypothesis. The <Controller>'s mental model did not update when the <Input> changed because other feedback sources were showing conflicting information, and the <Controller> was

	overwhelmed and could not determine which <Input> was the most relevant. <Controller> knew that the system was in a new state due to <Input>. However, they did not know how this new state affected the impact of their controls. They may try <UCA> to test the system impact, but did not know that the effects of <UCA> would be hazardous given <Context>
Class Two	The <Controller>'s mental model was updated when the <Input> changed, and other <Inputs> that were correct appeared unreliable. The <Controller> did not receive <Input> in time and was unable to determine why the system was behaving in a certain way. Therefore, they needed to conduct hypothesis tests on the system to troubleshoot. The <Controller> believed that <UCA> would be a safe test, as it would provide essential information on the system's state. However, given <Context>, the test was unsafe. <Controller> received <Input>, but the <Input> could be true in multiple system states. Therefore, they needed to conduct tests. <Controller> believed that <UCA> would be safe and give them important information on the state of the system. However, given <Context>, the test was unsafe.
Class Three	<Controller> was conducting small hypothesis tests that were not intended to be implemented at the system level. However, the <Controlee>interpreted the action as a sign that it was the correct action to implement system-wide.
Class Four	The <Controller>'s mental model did not update when the <Input> changed because it conflicted with their initial hypothesis of the system state, and the <Input> was not salient enough to prompt a change in their hypothesis. The <Controller>'s mental model did not update when the <Input> changed because other feedback sources were showing conflicting information, and the <Controller> was overwhelmed and could not determine which <Input> was the most relevant. <Controller> knew that the system was in a new state due to <Input>. However, they did not know how this new state affected the impact of their controls. They may try <UCA> to test the system impact, but did not know that the effects of <UCA> would be hazardous given <Context>

Decision-Making Process

Decision-making processes refer to the methods controllers use to select between multiple control actions. One of the most common models used to understand human behavior and decision-making today classifies human behaviors into three categories: skill-based, rule-based, and knowledge-based (Rasmussen, 1983).

Skill-based behaviors are frequently repeated and require minimal active mental energy. Skill-based decisions happen quickly and often subconsciously. A soccer player dribbling does not need to think about each individual muscle flexion as they make their way down the field. Other common skill-based behaviors include touch typing for an experienced secretary or route manufacturing steps. Skill-based decision making is outside of the scope of this thesis because those types of actions do not have a significant impact on safety-critical decisions in organizational or managerial contexts (Rasmussen, 1983).

Rule-based behaviors are selected by matching the current situation to a previously known behavior. Rule-based behaviors require active effort to identify the appropriate action, but limited problem-solving is necessary. Other examples of rule-based behavior include expert technicians troubleshooting a frequent problem or doctors diagnosing a common ailment. Some rule-based behaviors may transition to skill-based behaviors as humans gain more practice and experience (Rasmussen, 1983).

Finally, knowledge-based behaviors describe human responses to novel or challenging conditions. Knowledge-based behaviors emerge when humans encounter unfamiliar problems or system states. Examples of knowledge-based behaviors include developing new experimental treatments or making strategic management decisions to improve patient throughput in a hospital (Rasmussen, 1983).

As situations and contexts evolve, people transition between skill-based and rule-based decisions, or from rule-based to knowledge-based decisions. Staying at an inappropriate decision-making level can lead to an inappropriate response (Rasmussen, 1983).The next sections describe how rule and knowledge-based decision-making can lead to unsafe control action selection.

Rule based

Controllers often develop rules for responding to repeated system contexts. They may also receive training or instruction that provides a list of procedures to follow in specific contexts. In humans, these set responses are called scripts (Schank & Abelson, 1977).

Scenarios may arise when controllers develop scripts that default to a specific control action in certain contexts that do not include steps to evaluate all sources of feedback. Consequently, a controller could systematically overlook a critical piece of feedback, even if it provides useful information. Such patterns may be provided in training or may have developed over time (Thomas, 2024).

Additionally, unlike computers, humans cannot simply learn new scripts and delete out-of-date scripts. As systems evolve and change, humans may identify an inadequate script that no

longer works in the new context. Pilots who move from Boeing planes to Airbus Planes, for example, must learn an entirely new set of scripts. The pilots who are new to Airbus only need to learn the scripts applicable to an Airbus. However, the transferred pilots may require additional training to identify which of their Boeing scripts are no longer helpful or actively harmful.

One challenge humans may encounter with rule-based decision-making is identifying when to deviate from a script and transition to a different mode of problem-solving. For example, a doctor with a patient experiencing high blood loss will follow a standard routine to mitigate the concern. If the default actions are ineffective, the doctor must deviate from their script. Knowing when to move away from a specific script is difficult. Humans are prone to paying more attention to information that confirms their hypothesis and minimizing information that conflicts (Wickens & Carswell, 2012).

Table 3. 11 shows scenario archetypes for rule-based decision making.

Table 3. 11 contains scenario archetypes related to rule-based decision making.

Class	Detailed Causal Archetype
Class One	<Controller> has developed an incorrect script as a response to <Input>, either due to negative transfer, system changes, or training. The <Input> was associated with too many scripts, and the <Controller> could not determine which one was correct. Earlier <Input> prompted the <Controller>to invoke a script that did not involve checking or attending to <Input>.
Class Two	<Input> was not specific enough to allow the <Controller>to realize that their trained scripts were insufficient to handle the situation.
Class Three	<SCA> had previously been accompanied by another control. <Controlee> may have learned to wait for the additional control before changing their behavior.
Class Four	<SCA> was responded to by <Controlee> in a particular way in the past. However, after a change to the system, <SCA> had to be responded to in a new way.

Knowledge Based

If humans are unable to identify a script that works for the current system context, they move to knowledge-based decision-making. In these cases, humans use their knowledge of the system, their mental model of how it functions, and other information to identify and evaluate new potential control actions.

One form of knowledge-based problem-solving is creative problem-solving, which involves inventing novel solutions and assessing their potential effectiveness. Creative problem-solving takes significant cognitive resources and time. If humans become fatigued during the process of creative problem-solving, they are at risk of falling into cognitive tunneling. Cognitive tunneling

describes a state where the person is unable to come up with new ideas and stays too focused on one channel of reasoning (Wickens & Carswell, 2012).

Another way that knowledge-based decision-making can lead to unsafe decisions is when the information used to make a decision is based on an incorrect assumption. Unfortunately, humans have difficulty differentiating between assumptions and known facts in their mental models and are unable to recognize when they are using facts or assumptions to make a decision (Lehto et al., 2012; Wickens et al., 2013). Scenarios should identify the incorrect assumption itself, how the assumption emerged, and why the system is not set up to identify or correct it.

Unidentified reliance on an inaccurate assumption is not limited to individuals. Erroneous assumptions are often held widely across an organization or industry. For example, individuals within a particular field may have been subject to the same training or other potentially erroneous information. Additionally, information imparted to many through conferences, presentations, and publications may be misleading or inaccurate. Publication biases are a well-known source of industry bias; for example, in psychology, many long-held beliefs about human behavior were called into question after it was revealed that the studies could not be replicated (Korbmacher et al., 2023; Open Science Collaboration, 2015).

Knowledge-based decisions may lead to unsafe control actions when controllers lack the necessary knowledge to predict how the system state will respond to different control inputs. When humans make choices, they rely on their mental model of the system to predict how different control actions will impact the system's future state (Endsley, 1995; Rasmussen, 1987).

Mental models are particularly critical for allowing humans to conduct mental “what-if” hypotheses and compare the potential predicted outcomes. The usefulness of this approach will depend on the accuracy of the user’s mental model. If the mental model cannot run “what-if” tests, humans may need to run diagnostic tests on the system itself. For example, a hospital group managing dozens of locations may be unable to predict how changing a work process will impact efficiency. Instead of rolling out the change to all locations, the group may test out the change on one or two locations first. Diagnostic tests could also include a physician providing different treatments to patients without a precise diagnosis. The physician will use the outcome of the treatments to help inform their mental model of the patient’s condition and rule out or include different diagnoses. While these tests are often helpful, they increase the risk of providing an inappropriate control action for the system. For example, a doctor may provide a treatment that causes the patient significant harm while they are trying to rule out different diagnoses.

One reason humans struggle to develop adequate mental models is that actions frequently have multiple effects in complex systems. For example, mode confusion is when a controller makes a decision based on an incorrect belief about the current system mode (Sarter & Woods, 1995). Often, in modern complex systems, one specific action has different effects depending on the mode of the system. A controller who believes the system is in mode A may believe control action Z will have a safe impact on the system. However, if the system is actually in Mode B, Control action Z may instead create an unsafe system state. A simple example of how control inputs can change meanings depending on system mode is how pressing the “volume up” button

on many smartphones usually increases the volume. However, if the phone is in camera mode, the volume-up button will take a picture.

Furthermore, the side effects of an action may continue beyond what was predicted. For example, in one real healthcare system, an insurer believed a hospital had too many adverse events. The insurers paid for the additional care patients required after an adverse event. The insurer believed refusing to cover patient care resulting from adverse events would incentivize the hospital to perform better. When the policy was changed, the hospital had less money to cover its operating costs. To compensate for the lost income, the hospital needed to bring in additional patients and increase its treatment volumes without increasing staffing or equipment resources. In the end, the hospital's adverse event rate increased instead of decreasing (Stringfellow et al., 2009).

Table 3. 12 has scenario archetypes for knowledge-based decisions.

Table 3. 12 contains scenario archetypes related to knowledge-based decisions.

Class	Detailed Causal Archetype
Class One	The <Controller> lacked sufficient time and mental resources to identify a novel solution to the <Context>. No previous solution would have been safe in this context. The <Controller>'s mental model was not granular enough to run satisfactory "what if" tests to evaluate control options. <Controller> was unaware that the <UCA> they chose would have side effects beyond the desired effect.
Class Two	<Input> could not provide <Controller> with information about the effects of the available controls. <Input> was insufficient to keep <Controller> aware of the <Controlee>'s mode. The <UCA> would have been safe if the <Controlee> were in a different mode.
Class Three	<Controlee> changed modes between the control action being sent and the control action being received.
Class Four	<Controller> may not have understood why <SCA> was issued. Because they have access to a different set of information, they may ignore or otherwise not exercise full control.

View of the problem

In addition to difficulties with rule- or knowledge-based decision-making, humans can reach different decisions about which controls to select based on how they view the system and their position within it.

For example, controllers may have different problem-solving biases depending on their placement in the control hierarchy. Lower controllers are biased toward solutions that avoid the worst-case scenario, while higher-level controllers are more inclined towards solutions that prioritize the best-case outcome (Wickens et al., 2013). These distinct preferences result in different decisions and behaviors. One of the ways that controllers can impact how lower-level controllers problem-solve and arrive at solutions is through changing the way that decisions are framed (Wickens et al., 2013).

Humans also have difficulty selecting the appropriate control action in scenarios where they misattribute the risk or severity profile. Regardless of the likelihood, humans are more likely to take preventative measures when the potential loss is severe. However, they are less likely to use preventative measures when the severity of the loss appears smaller, even if it is a frequent event (Wickens et al., 2013).

Furthermore, in many systems, major losses are rare. Consequently, people often overestimate their own abilities because they have not personally experienced such a loss. This overconfidence can lead to the onset of riskier behaviors (Wickens et al., 2013) as controllers are less likely to predict losses or hazards when making control decisions.

Scenario archetypes based on a controller's system perspective are shown in Table 3. 13.

Table 3. 13 contains scenario archetypes related to a controller's view of the system.

Class	Detailed Causal Archetype
Class One	<Controller> prioritizes the best-case outcome and is unaware of <Context> that would change the effect of <UCA>. The existing <Input> may be technically correct, but it is insufficient to predict the outcome of <UCA>. Because the <Controller> perceived the risk of error to be minimal, they were less attentive to feedback such as <Input>. <Controller> did not believe <Input>, because no loss had happened previously in their experience. <Input> was insufficient to change their mental model of the current system's behavior.
Class Two	<Controller> prioritizes the best-case outcome over possible hazards, but the overall system has the opposite priority. Because <Controller> was prioritizing a best-case outcome, they may have a lower perceived value from conflicting information.
Class Three	<Controlee> does not believe <SCA> is necessary. They may have received similar controls and ignored them without consequence in the past.

Class Four	<Controlee> ignores <SCA> because it has received instructions or training to prioritize a different outcome.

Optimization and experimentation

Human problem-solving is not limited to solving problems that job tasks pose. Often, problem-solving involves optimizing behaviors to achieve comfort or efficiency. Over time, this optimization can result in the elimination of safety-related preventative measures such as donning protective equipment, setting up safeguards, and completing all steps in a process (Rasmussen, 1990). To prevent such manners of problem-solving, processes must be designed to make unsafe actions difficult and safe actions straightforward (N. Leveson, 2011). Furthermore, there must be ways for controllers to identify when a shortcut or optimization step makes their process less safe. Often, accidents are the only signal controllers have regarding whether a process has been changed too much in an effort to improve efficiency.

Table 3. 14 shows scenario archetypes based on optimization and experimentation.

Table 3. 14 contains scenario archetypes related to optimization and experimentation.

Class	Detailed Causal Archetype
Class One	The <Controller> was experimenting to make a process more efficient. The <Controller> further reduced safety margins on <Control action> because they had received no negative feedback the last time <UCA> was executed. <Controller> did not believe that <Input> indicated <UCA> would lead to negative consequences, as previous instances of <UCA> had not resulted in negative consequences. <Controller> did not realize that <Control Action> was set to be strict enough that any deviation from <Safe Control Action> would lead to a hazard.
Class Two	<Controller> does not verify <Input> because previous verification steps did not change their decision-making.
Class Three	<Controller> provided safe control action to <Controlee> that was too difficult or time-intensive for <Controlee> to follow every time.
Class Four	While the <Controller> provided <SCA>, there was no <Input> from the <Controlee> indicating that the control was adequate. Over time, the <Controlee> may have stopped fully following the <SCA>.

Controller Goals

Unlike technical system components that have designated “states,” humans do not have true modes. Instead, humans change how they respond to input based on their goals. Different inputs may change which goal a controller is prioritizing at any time.

Misaligned Goals

If the goals of controllers (either individuals or organizations) are not synchronized with broader system goals, unsafe scenarios will arise. One common way goals lose alignment is when organizations provide incentives to their members that do not align with the system's goals (Carayon et al., 2012; N. Leveson, 2011). For example, suppose a healthcare organization aims to maximize patient throughput but does not provide incentives to employees when patient numbers go up or disincentives when patient numbers decrease. In such an organization, employees will be unlikely to cooperate with the push to increase throughput unless they are incentivized to work towards the same goal that management has.

Incongruous statements and actions from higher-level controllers can lead employees to make assumptions about which control actions will yield the best outcome for the system or themselves. Inaccurate assumptions about the state of other controllers can lead to scenarios where feedback or controls are interpreted differently than expected (Colquitt et al., 2011). For example, an employee may have faulty assumptions about the goals of their supervisors. Employees may believe their supervisor's top priority is workplace efficiency, not safety. Employees who assume their boss does not prioritize safety may choose riskier controls that prioritize immediate efficiency. Communications and incentive structures must align with the overall system goals to prevent unsafe action selection.

Another example is a company that claims its priority is safety and verbally informs its employees that safety is the top priority. Nevertheless, if the company only provides incentives for achieving productivity metrics, such as reducing downtime or increasing output, it may not adequately reward employees who make safety-minded decisions. In that case, employees will have the goal of avoiding negative consequences and will be less likely to make safety-minded control actions in the future.

Goals can also conflict between different people in a system. Often, unsafe outcomes occur when humans operating the system have different goals than higher-level controllers. This conflict creates a mismatch between the goal conditions of higher and lower controllers. For example, consider a control loop between hospital management and clinicians. Management might have the goal of reducing missed critical laboratory test results. Therefore, management institutes a policy that requires clinicians to acknowledge all lab results within one hour of receipt. Clinicians may have the goal of avoiding potential reprimands. Consequently, clinicians may create habits of marking all lab results as acknowledged before truly evaluating their values.

Table 3. 15 contains scenario archetypes based on misaligned goals.

Table 3. 15 contains scenario archetypes related to misaligned goals between a controller and the broader system.

Class	Detailed Causal Archetype
Class One	<Controller>'s goal of <goal> conflicts with the system level goal of <system goal> because <Context>

	<Controller> misinterpreted the command from <higher level controller> because they had the wrong goal in mind for system performance The <Controller> was incentivized to maximize a different parameter than what was best for the system. They may have known that the control would lead to an unsafe result, but believed the <UCA> would lead to the best outcome for them.
Class Two	Communication from <Superior Controller> was interpreted in a way that changed the goal state of the <Controller>
Class Three	<Controller> sees that they need to improve safety, but believes that the <UCA> will improve performance. However, they don't realize that <Controlee> will find an unsafe workaround to achieve the requirements in the <UCA>.
Class Four	Controlee receives <SCA>, but the <SCA> may not come with enough incentives for them to follow through.

Switching between Goals

Human controllers can have multiple goals at once, and their goals often conflict (Simon, 1957; D. D. Woods, 2000). For example, a hospital administrator may have the dual goals of averting accidents and increasing profitability. These two goals do not always conflict; a hospital with too many accidents will not be profitable. However, when making decisions such as setting appropriate staffing levels or setting patient throughput metrics, the goals of profitability and safety will conflict.

The incentives or motivation structures of systems and organizations often drive an individual's personal goals. Companies often strive to enhance performance by fostering a competitive environment among employees. However, while competition can lead to higher motivation, it can also negatively distort performance (Colquitt et al., 2011; Schein, 2015). Competition can change a person's goals from improving the outcome of the work to improving how their work compares to others. Unsafe control decisions occur when people believe they will do better if others do worse. Organizational psychology has repeatedly found that promoting teamwork is essential to producing productive outcomes, but often, the incentives in place do not reward good team members (Schein, 2015).

Additionally, controllers may not know how to prioritize tasks that are assigned to them. If instructions from other controllers are sent without clear prioritization, the controller in question may be unable to distinguish when a high-priority event needs to be responded to before any other tasks.

Table 3. 16 contains scenario archetypes based on a controller changing goals.

Table 3. 16 contains scenario archetypes related to a controller's goals changing.

Class	Detailed Causal Archetype
Class One	<Controller> ignores <Input> because they are focused on improving a different metric due to their perception of the incentive structure.
Class Two	<Controller> relies on <Input> to make a decision, but the <Input> is insufficient to prompt them to switch to a different goal when necessary.
Class Three	<Controller> issues a <SCA>, but the <Controlee>to which they issue it has a different goal for system performance due to previous controls, and they ignore or misinterpret the <SCA>.
Class Four	<Controlee>responds to events labeled as high priority by <Controller> every day that turn out to be insignificant tasks. In that case, an actual high-priority alert will not seem unusual nor stick out to <Controlee> as requiring immediate attention.

Controls and Inputs from other Controllers and the Environment

Control decisions are not only impacted by a controller's internal decision-making process. Input from the environment or other controllers will impact how the controller in question behaves.

Environmental Factors

One of the most significant factors that influences the decision-making of controllers is the availability of adequate resources (Rasmussen et al., 1990). When humans do not believe they have adequate resources to successfully meet set goals, their engagement and performance with the system are diminished (Demerouti et al., 2001; Luczak et al., 2012). Furthermore, insufficient resources limit the degrees of freedom a controller has when making a decision (Rasmussen, 1990).

Resources and environment can range from the physical environment and financial support to staffing experience, training, and retention. Insufficient staffing, for example, stretches the ability of every controller to complete all necessary tasks adequately. Even if a major decrease in workforce seems to perform adequately initially, fatigue from burnout after extended periods of high workloads is a major contributor to decreased performance (Demerouti et al., 2001).

Unsafe actions are often the result of demand-resource mismatches (Rasmussen, 1986). Demand-resource mismatches are situations that require more resources than are available. Demand-resource mismatches often occur because the resources needed in unusual situations may rise significantly above those needed in normal operations. The rise in necessary resources may continue to escalate as off-nominal situations increase the required speed and cognitive effort of necessary decisions. If the resources are not available as the demand increases, the situation can continue to deteriorate until an accident occurs (D. D. Woods, 2000).

Demand-resource mismatches often occur when automated systems experience unplanned or unsafe behavior and the humans monitoring the system cannot diagnose and solve the problem correctly. In these cases, humans are tasked with intervening to correct a system when it is operating well outside of normal parameters. Workload burdens can increase significantly during such events, and penalties for mistakes will be more severe than typical (D. Woods, 1995).

In systems with a chronic demand-resource mismatch, individuals working in the system have only enough time to stay on top of the most basic routine tasks. Humans working in demand-resource mismatched systems rarely have time to improve workflows even if improvements could lighten their workload. Therefore, there is no time to make processes more efficient. Furthermore, employees often have less time with and attention from supervisors who have a greater ability to change workflows (Tucker & Edmondson, 2003).

Another effect of a demand-resource mismatch is the system slipping into an unsafe state over time. For example, under-resourced systems can slip into unsafe states if tasks such as maintenance, updates, and evaluation are delayed or canceled. Deferred maintenance can be particularly hazardous because, in the event of unusual events, fewer resources are available to manage the extra workload. Therefore, even small events escalate quickly into major losses.

Scenario archetypes based on environmental factors are shown in Table 3. 17

Table 3. 17 contains scenario archetypes related to environmental factors.

Class	Detailed Causal Archetype
Class One	<Controller> received instructions from <Superior Controller> to execute <UCA>. <Controller> may have received negative feedback from previous instances of questioning directives from <Superior Controller>. <Controller> received instructions from <Superior Controller>to execute <UCA>. <Superior Controller> may not have sent a <UCA> request before. Therefore, <Controller> did not question the instructions. <Controller> may have access to <Input>, but did not believe that it would change their decision.
Class Two	<Controller> believed that the resources necessary for <UCA> were already in place. However, they were unaware that the resources were insufficient.
Class Three	The <SCA> may have gone to many different types of organizations. One <Controlee> may have had a different context or level of resources that made the SCA not safe in their particular context.
Class Four	<Controlee> receives <SCA>, but <Controlee> does not have the resources to manage the additional workload. Therefore, <Controlee> must choose between executing the SCA and executing their other tasks. <Controller> may not

	have control over the resources of the <Controlee> or may not have believed that the control would require additional resources.

Conflicting or Insufficient feedback

While humans do not experience data corruption in the same way that a computer system does, information can still be distorted and altered as it is transmitted between individuals and organizations.

The information needed to make a control decision may be in conflict. It is much more challenging for people to maintain a clear understanding of the system's state when their sources of information provide conflicting information. High-level controllers must ensure that there are straightforward ways for others in the system to verify information (N. Leveson, 2011).

Humans struggle to direct their attention when different sources of information conflict (Carroll & Sanchez, 2021). If the wrong data is selected, humans will not update their mental model correctly, even if the data needed to correctly modify their mental model exists (D. D. Woods, 2000). Humans also struggle to redirect their focus and attention as the world changes (D. D. Woods, 2000). If humans are used to getting information from specific sources, they may not look for data in other places.

Additionally, system controllers may not update their mental model correctly, even when the feedback is accurate, because they cannot redirect their attention from irrelevant information to the relevant scenario. Humans often continue to follow their initial plan even when conflicting information arises (De Keyser & Woods, 1990). Conflicting information must be extremely salient to be noticed in such cases. Humans who are unable to redirect their attention appropriately risk becoming fixated on one hypothesis and ignoring or not seeking out other possibilities.

Furthermore, providing feedback is often not the main goal of humans within a system. For example, doctors may consider writing reports of near misses or technology errors as less important than responding to patient questions or reviewing labs. This lack of prioritization is compounded if reporting is difficult or time-intensive to complete. Moreover, providing adequate feedback may lead to punishment. A doctor who reports a technology issue and receives retraining or reprimands for “user error” will be less likely to submit similar comments in the future.

Another feedback design consideration is how decision support tools are designed. Humans are generally able to both identify linear trends using raw data and use those trends to adequately predict future system states. However, humans are not good at predicting future system states that are changing non-linearly (Sterman, 1989; Wickens et al., 2013). Decision-making guidance may need to be supplied to help humans identify non-linear trends. If a trend in the system develops non-linear behavior, the feedback that was provided previously may no longer be sufficient.

Table 3. 18 contains scenario archetypes based on inadequate feedback

Table 3. 18 contains scenario archetypes related to inadequate feedback.

Class	Detailed Causal Archetype
Class One	<Controller> does not check the <Input> source regularly because it rarely updates with valuable information. <Controller> does not trust <Input> because it is inconsistent or has been inaccurate recently. <Controller> develops a hypothesis of the system state and does not notice that <Input> is inconsistent with that hypothesis. <Input> is technically accurate, but it is displaying information about a change in the system that is difficult for humans to interpret without additional details correctly.
Class Two	<Controller> does not receive <Input> because the people who could send the report do not believe that sending the information is the best use of their time. <Controller> does not receive <Input> because the people who could send the report believe they could be disciplined for submitting a report due to prior experience.
Class Three	<Controller> notices that <Controlee> is engaging in unsafe behavior so sends a <SCA>. However, the <Controlee> is not looking for outside <Input> and does not interpret the <SCA>.
Class Four	<Controlee> receives <SCA>, but the <SCA> may include instructions that require the <Controlee> to do something only in a specific context. The <Controlee>may not have adequate <Input> to identify that context.

Relationships between controllers

Feedback may also be insufficient if the relationship between controllers degrades. Political, social, and other categories of interpersonal relationships influence human and organizational decisions. Individuals will change their choice of control actions maintain a particular relationship, especially if that relationship influences their work and goals (Lehto et al., 2012). Organizations may choose to share sensitive information with each other if they receive critical information in return. For example, airlines share safety insights with each other because if any airline experiences accidents, sales across all airlines will drop (N. Leveson, 2011). On the other hand, organizations may stop sharing information if the relationship deteriorates and less value is obtained through collaboration. Keeping relationships intact can be critical to the function of the overall system. If such relationships are not considered when making control decisions, critical system interdependencies may weaken over time.

The relationships between controllers also influence the level of trust among controllers. Safety-critical industries require higher trust interactions and relationships between hierarchical

levels (Schein, 2015). Safety-critical systems must build and maintain trust, both within individual organizations and between different organizations. When controllers do not trust other members of a system, safety is significantly impaired (N. Leveson, 2011). Furthermore, a lack of trust often leads to reduced communication and assumptions about each other controllers' goals and objectives. For example, suppose employees believe they will be punished for deviating from the standard operating procedures. In such an environment, the employees will not discuss or reveal workarounds to management even when the workarounds are necessary for achieving other management-set metrics (e.g., output per day or turnaround time). Without awareness of how work is actually conducted, management will be unable to identify when the system migrates to an unsafe state.

Table 3. 19 contains archetypes for scenarios involving relationships between controllers.

Table 3. 19 contains scenario archetypes related to relationships between controllers.

Class	Detailed Causal Archetype
Class One	<Controller> does not trust that the <Input> they are receiving is accurate because they believe the source of the <Input> is withholding or editing the data.
Class Two	"Because the <Controlees> supervised by <Controller> do not trust <Controller>, they do not share complete information that <Controller> needs to make decisions. <Controller> no longer receives <Input> from <Peer Controller> because the peer relationship has degraded or a voluntary information sharing agreement has lapsed. <Controller> no longer receives <Input> from <Peer Controller> because they have stopped sharing information with that <Peer Controller> or have otherwise damaged the relationship between the two organizations or individuals. <Controller> is unaware of the actual processes used to complete a task. The <UCA> may have been safe in the context of the process the <Controller> has documented; however, workarounds changed the context, making the <UCA> unsafe. Workarounds may not be communicated to higher-level controllers.
Class Three	The <SCA> is safe, but the <Controlee> does not trust it, given the history of previous control actions.
Class Four	The <SCA> may be technically safe, but the <Controlee> believes that following through with it would weaken a critical relationship.

3.5 Conclusion

To conclude, the techniques in this chapter enable a more thorough analysis of sociotechnical systems using STPA. Particular focus is given to the identification of detailed scenarios in sociotechnical systems. Detailed low-level scenario archetypes for each of the four classes of scenarios are provided for each component of a control loop.

Overall, this chapter provides forty-six class one, thirty-three class two, seventeen class three, and seventeen class four scenario archetypes.

In the next chapter, the process of identifying scenarios provided in this chapter will be applied to a real system as a case study.

Chapter 4: Application to the US Laboratory Data Safety Management System for Over-the-Counter Diagnostic Tests.

To demonstrate how the process for scenario identification, as shown in the previous chapter, can be applied to complex socio-technical systems, this chapter reviews a case study of an STPA on a sociotechnical healthcare system. Specifically, the following chapter highlights results from an STPA analysis of the Over the Counter (OTC) diagnostic test safety management system in the United States.

Studies have been conducted on individual components within the OTC diagnostic test system, but few analyses have been performed on the system as a whole. The use of OTC tests has increased significantly since the COVID-19 pandemic. An STPA analysis can identify hazards in the current system design and can help model and understand how the increased use of OTC tests is impacting the broader healthcare system.

The OTC diagnostic test system in the US is an ideal candidate for a sociotechnical STPA. The OTC system is highly sociotechnical; the tests themselves are highly technical products, but the social system of regulators, manufacturers, and users is equally important to consider when identifying systemic hazards. While studies have been done on how consumers use OTC tests (O'Laughlin et al., 2022; Todsen et al., 2023), fewer studies have been conducted on the impact of decisions made at the organizational level. An STPA focused on hazards in the sociotechnical safety management structure of the system will identify opportunities for system-wide improvements.

The work presented in this chapter is an extension of an STPA done on the OTC system in 2024 (N. Leveson et al., 2023). This chapter will compare the scenarios generated from this process with the initial list of scenarios to demonstrate the ability of the process to enable the identification of previously unidentified scenarios. Because this thesis is primarily concerned with the process of scenario generation, the losses and hazards, control structure, and UCAs are adopted from the original project.

4.1 System Overview

OTC tests are clinical diagnostic laboratory tests that have been adopted for use by patients at home. OTC tests are either self-administered by the patient or administered by a non-professional caregiver (CMS, 2022). Currently, few OTC tests are approved by the Federal Food and Drug Administration. However, the necessity of at-home tests for COVID-19 during the COVID-19 pandemic has brought more pressure to approve other health tests (Jean et al., 2021). Other common OTC tests include pregnancy tests and blood glucose monitors. Concerns with increasing the availability of OTC tests include patients' ability to successfully use OTC tests and a lack of test result data for public health monitoring of communicable diseases (McPhillips,

2022). This sizeable sociotechnical system includes federal regulators, public health agencies, test manufacturers, and health information technology in addition to millions of patients.

The OTC diagnostic test system in the United States has numerous technical components, including the OTC devices themselves, electronic health records, and public health databases. However, these technical elements are embedded within a broader social system that encompasses doctors, hospital administrators, regulators, laboratory technicians, and others. While significant analysis has been done on the technology itself (Center for Devices and Radiological Health, 2021; Lindner et al., 2021; Todsen et al., 2023; World Health Organization, 2015), there has been insufficient analysis of how the complex social safety management system controlling the technical elements impacts safety.

The goal of this STPA analysis is to identify current gaps in the current safety management system for OTC diagnostic tests and to anticipate potential future gaps that may emerge as OTC tests become more prevalent. Additionally, the analysis would identify recommendations to improve the system's safety. The work shown in this chapter is a continuation of a larger project, which evaluated the Safety Management system of OTC tests and Point of Care tests (N. Leveson et al., 2024). This thesis expands the results of the initial STPA study using the process provided in Chapter 3.

4.2 STPA Analysis

4.2.1 Losses and Hazards

Like most complex systems, the OTC diagnostic test system has a plethora of stakeholders, each with distinct needs, goals, and desires. The selection of losses is critical because STPA is a top-down analysis. The final results will only capture data relevant to those needs if the correct losses and hazards are identified. The priorities of the stakeholders in this analysis were patient safety and overall trust in the healthcare system. Therefore, the following losses were considered:

L1: Loss of life or injury to patients

L2: Loss of reputation or trust in the laboratory-data HIT system

With the losses defined, hazards were identified.

H1: Patients receive less than the acceptable standard of care (L-1)

H2: Laboratory ecosystem stakeholders, including patients (public), lose trust in the laboratory data being collected, shared, analyzed, and reported (L-2)

Developing the language for the losses and hazards required significant effort and refinement. The stakeholders for this project included regulators, clinicians, patients, and many others. The fields represented by the stakeholders and the analysts often used the same words, but with different implications. For example, doctors may use the term "complexity" to refer to biological systems that are not designed or engineered, whereas engineers use the term to describe any system that is intellectually unmanageable.

Initially, H-1 was phrased as “patients receive insufficient care.” However, physicians and other stakeholders noted that "insufficient care" could be interpreted too widely. For example, physicians pointed out the distinction between a patient who dies or is injured from a condition for which there is no known treatment and a patient who dies or is injured from receiving care that was not aligned with current medical guidance. Changing the phrasing from "insufficient care" to "less than the acceptable standard of care" helped communicate to those in the medical community that the object of the study was not to blame physicians for the limitations of human mortality, but instead to focus on why patients may not receive the current medical standard of care. An article by Perry et al. discusses some of the other language difficulties between systems analysts and healthcare professionals (2021).

4.2.2 Control Structure

For the analysis in this chapter, the control structure in Figure 4. 1 is used. This is an abstracted control structure from the one in the original study. In this model, the controlled process is the databases that store test result information. If this data is collected, it is stored either in a public health database or in an application created by the test manufacturer. Except for the data layer, every controller is an individual human or a collection of humans.

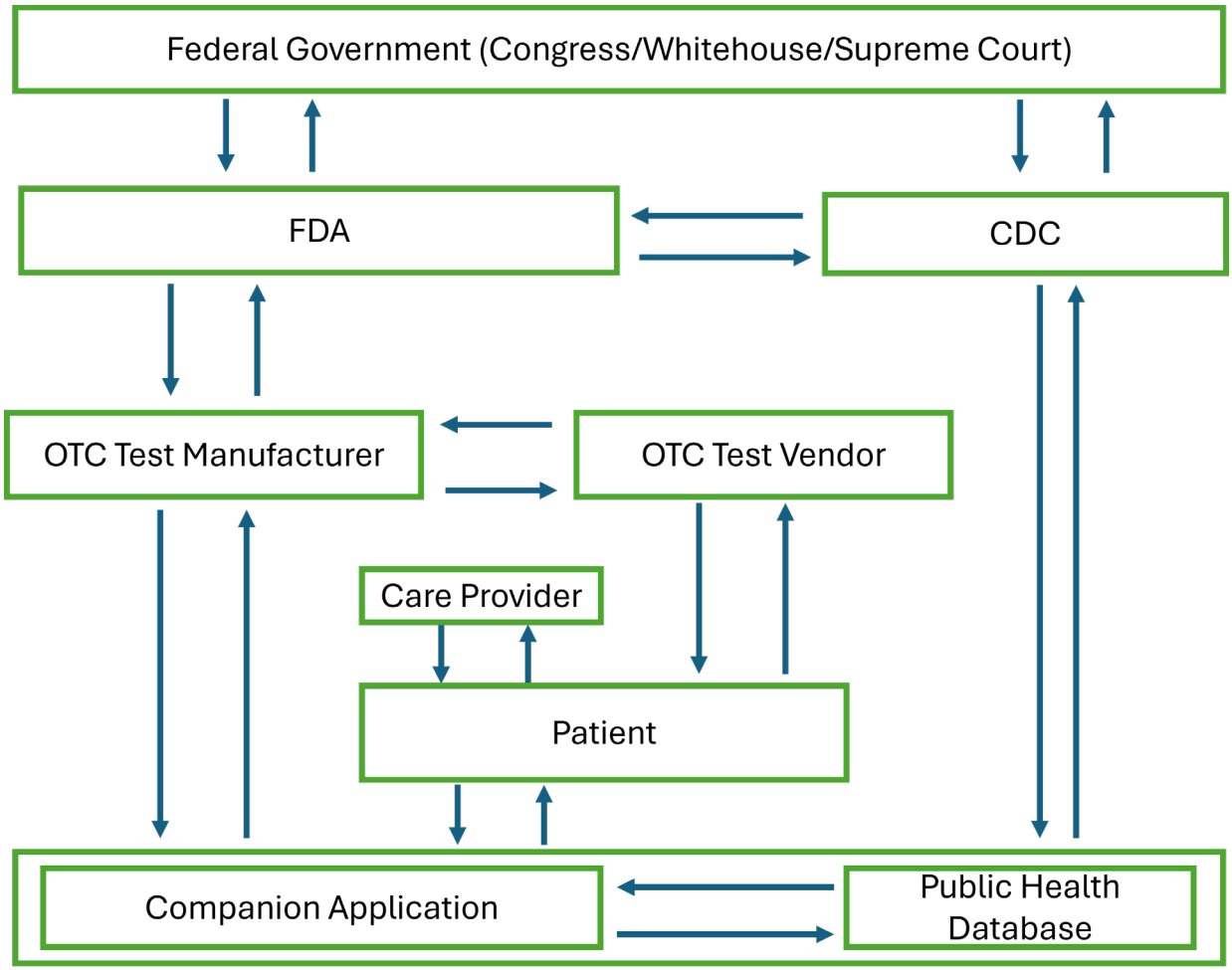

Figure 4. 1 depicts the control structure of the OTC diagnostic testing system in the United States. This figure is adapted from (Leveson et al., 2024).

This is an abstracted model of the system. Every organization could be further broken down into departments, and there are dozens of other groups that interact with the components depicted. However, maintaining this level of abstraction enables the analyst to understand the most critical control loops within the system's control hierarchy. A more detailed control structure of the system can be found in Appendix A.

The federal government comprises the three main branches: Executive, Judicial, and Legislative. It provides the legal framework and funding that allows regulatory authorities to create and enforce safety regulations and, therefore, has some of the most influential controls in the system.

The Food and Drug Administration (FDA) regulates medical devices, including OTC tests. Before the OTC test manufacturers can market their tests to customers, they must obtain approval from the FDA. The FDA also has the authority to conduct audits of manufacturers and impose corrective actions on companies that produce unsafe devices.

The Centers for Disease Control and Prevention (CDC) is the regulatory agency responsible for public health in the United States. It collects data on communicable disease outbreaks and

publishes guidance to protect Americans against disease. The CDC sets the data standards for any clinical test results that it requests. While certain test results taken in a traditional clinical lab must be reported to the CDC, there are no such requirements for OTC test results.

Both the FDA and the CDC are within the federal Health and Human Services department. Regarding OTC testing, they are the two agencies with the strongest controls, directly overseeing certification and future system evolution.

OTC Test Manufacturers develop, manufacture, and sell OTC tests. They have the most control over the functionality and performance of OTC test technology, as their resource allocations and design decisions directly impact the safety of these devices. The manufacturers also determine whether to create a companion application that patients can use to record their test results and report them to public health agencies, such as the CDC.

Test Vendors are the entities responsible for selling tests or providing them to users. These might include pharmacies, online stores, local government agencies, and others. The vendors select which tests to sell and have influence over what information the patient sees when making a purchase.

Care providers include clinicians, nurses, and anyone who works for a care facility to provide diagnosis or treatment to a patient. In this system, they may administer treatment based on the results of an OTC test and offer recommendations for OTC tests.

Patients have control over how and when they use OTC tests, as well as whether to share the test results with public health agencies or companion applications. Some patients, such as those with diabetes, may use OTC tests daily, while others may only take one if required, for example, patients who need to take a COVID-19 test before international travel.

4.2.3 Unsafe Control Actions

Each controller's control actions are listed in Table 4. 1 below. The controls not included in the original analysis are marked with an asterisk.

Table 4. 1 contains the controls available to each controller in the OTC diagnostic testing system

Controller	Control Actions
Federal Government:	Provide regulatory authority* Provide funding*
FDA	Create regulations to authorize tests* Approve OTC tests Issue corrective action to an OTC manufacturer Audit EHR developers for conformity to regulations*
CDC:	Set standards for reporting OTC data Publish public health guidance Identify and monitor outbreaks*
OTC Manufacturer	Release OTC device and instructions Provide data collection mechanism Select data standards to implement
Test Vendor	Sell or provide test to patient Sell medication
Care Provider:	Provide treatment to patient Prescribe/ recommend OTC test to patient
Patient:	Acquire OTC test Follow OTC pre-test instructions or test procedures Interpret test results Upload test results or personal information to database Seek Medical treatment

With the available control actions identified, UCAs are identified. This chapter will analyze the UCAs for the in the control loop between the FDA and the OTC test manufacturer, as shown in Figure 4. 2.

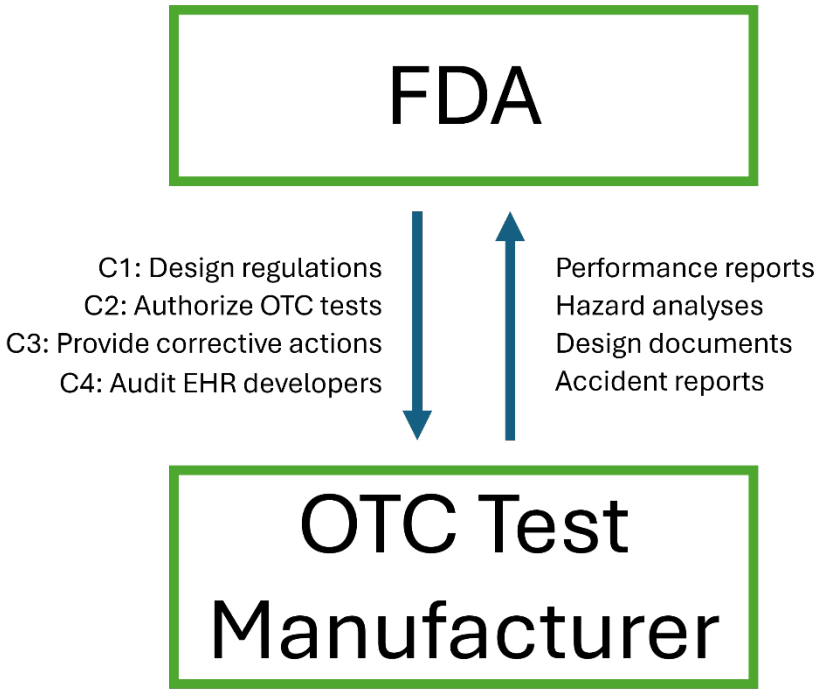

Figure 4. 2 depicts the control loop between the FDA and an OTC test manufacturer.

The four control actions that will be analyzed in this chapter are:

C1: Create regulations to authorize tests

C2: Approve OTC tests

C3: Issue corrective action to an OTC manufacturer

C4: Audit EHR developers for conformity to regulations

The UCAs for these four control actions are shown in Table 4. 2. UCAs that were not included in the original analysis are marked with an asterisk. Appendix B contains the UCAs for the other controllers in the system.

Table 4. 2 contains the UCAs identified for the FDA in the OTC clinical diagnostic system in the United States.

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long
Create regulations to authorize tests	UCA 1.1: The FDA does not update regulations to authorize tests when OTC technology is updated such that existing regulations are no longer sufficient. * UCA 1.2: The FDA does not create regulations to authorize tests that require the collection of information needed to monitor OTC test safety * UCA 1.3: The FDA does not create regulations that enforce the collection of information needed by other federal agencies (CDC). *	UCA 1.4: The FDA creates regulations that are insufficient to manage safety effectively. * UCA 1.5: The FDA creates regulations that conflict with the regulations of a different agency. * UCA 1.6: The FDA creates regulations that cannot be met by any OTC test. * UCA 1.7: The FDA creates regulations that require more work to administer than the resources available. * UCA 1.8: The FDA creates regulations that motivate regulated parties to behave unsafely*	UCA 1.9: The FDA removes regulations when they are still necessary to control safety. * UCA 1.10: The FDA provides changes to regulatory authorities too frequently to understand the impact of regulations on safety. *	N/A
Approve OTC Tests	UCA 2.1: FDA does not approve an OTC test when that test would enable better patient care decisions. UCA 2.2: The FDA authorizes a test too late to	UCA 2.3: The FDA approves a test that does not conform to regulated standards* UCA 2.4: The FDA approves a test that users are unable to use safely*	UCA 2.6: FDA approves an OTC device too late to get critical data during a health emergency	N/A

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long
	control the spread of an emergent disease	UCA 2.5: FDA approves an OTC test that does not facilitate data reporting by test users when that data is needed to inform public health decisions or test decisions.		
Issue corrective action to an OTC manufacturer	UCA 3.1: FDA does not issue corrective action to an OTC manufacturer following a series of inappropriate results from an OTC device.	UCA 3.2: FDA issues a corrective action to an OTC manufacturer whose device is performing according to regulations such that patients lose access to a critical test. UCA 3.3: The FDA provides corrective actions that are insufficient to control the identified problems. *	UCA 3.4: FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device.	UCA 3.5: The FDA applies a corrective action to an OTC manufacturer for too long following the resolution of a problem with an OTC device.
Audit to OTC manufacturers	UCA 4.1: The FDA does not audit a company with manufacturing processes that do not meet FDA regulations. *	UCA 4.1: The FDA audits a company in a way that is insufficient to identify processes that do not meet regulations. *		N/A

4.2.4 Scenarios

For the scenarios, each UCA is evaluated using the process outlined in Chapter 3 to understand why the controller might believe it is reasonable to provide the unsafe control action in the unsafe context.

To identify the scenarios, each UCA is used to identify the variables in the detailed scenario archetypes. Then, the complete scenario archetypes are used as prompts to investigate whether the system is designed in a way that the scenario is reasonable.

For example, the archetype variables from UCA 3.4 from Table 4. 2: "FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device" are shown in Table 4. 3.

Table 4. 3 defines the variables used in the scenarios for UCA 3.4.

Variable	Variable value
Full UCA	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device
Controller	The FDA
Controlee	OTC test manufacturer
Superior Controller	Federal Government
UCA, control only	The FDA does not provide a corrective action in time
SCA, control only	FDA provides a corrective action in time
Context	Devices provided a series of inappropriate results

In total, one hundred and thirteen detailed scenario archetypes were provided in Chapter 3 and used to create one hundred and thirteen prompts for consideration for further analysis. These prompts were refined with input from subject matter experts (SMEs) and additional research. The full list of generated scenario prompts is in Appendix C, but thirty-three are shown in Table 4. 4. In column two of Table 4. 4, the scenario archetype is listed. The corresponding scenario prompt adjusted with the specifics of UCA 3.4 is in column three. The scenario prompts that were developed into full scenarios are highlighted.

Table 4. 4 depicts the original scenario archetype provided in chapter 3 and the corresponding scenario prompt that has had the variables replaced with the context of UCA 3.4

ID	Class	Archetype	Scenario prompt The FDA issues a corrective action to an OTC manufacturer too late, following a series of inappropriate results from an OTC device, because…
3.4.A	One	<Controller> has limited familiarity with the system and takes too long to identify what perceptual cues are useful for addressing the current system context.	The FDA has limited familiarity with the system and takes too long to identify what perceptual cues are useful for addressing the current system context.
3.4.B	One	<Controller> had an accurate mental model before a system change; however, once the system behavior changed, the controller's mental model did not. Therefore, they interpreted <Input> incorrectly.	The FDA had an accurate mental model before a system change; however, once the system behavior changed, the controller's mental model did not. Therefore, they interpreted <Input> incorrectly.
3.4.C	One	<Controller> was inundated with <Input> and was unable to identify what was causing the system to change states. There may have been no direction from the system to guide the response or interpretation of the <Input>.	The FDA was inundated with <Input> and was unable to identify what was causing the system to change states. There may have been no direction from the system to guide the response or interpretation of the <Input>.
3.4.D	One	<Controller> did not believe <Input> source because there was insufficient corroborating information, and the system state <Input> indicated was rare.	FDA did not believe the <Input> source because there was insufficient corroborating information, and the system state <Input> indicated was rare.
3.4.E	One	The <Controller>'s mental model did not update when the <Input> changed because it conflicted with their initial hypothesis of the system state, and the <Input> was not salient enough to prompt a change in their hypothesis.	The FDA's mental model did not update when the <Input> changed because it conflicted with their initial hypothesis of the system state, and the <Input> was not salient enough to prompt a change in their hypothesis.
3.4.F	One	The <Controller> lacked sufficient time and mental resources to identify a novel solution to the <Context>. No previous solution would have been safe in this context.	The FDA lacked sufficient time and mental resources to identify a novel solution to the <Context>. No previous solution would have been safe in this context.
3.4.G	One	<Controller> was unaware that the <UCA> they chose would have side effects beyond the desired effect.	The FDA was unaware that not providing a corrective action in time they chose would have side effects beyond the desired effect.
3.4.H	One	Because the <Controller> perceived the risk of error to be minimal, they were less attentive to feedback such as <Input>.	Because the FDA perceived the risk of error to be minimal, they were less attentive to feedback such as <Input>.

ID	Class	Archetype	Scenario prompt The FDA issues a corrective action to an OTC manufacturer too late, following a series of inappropriate results from an OTC device, because…
3.4.I	One	<Controller> ignores <Input> because they are focused on improving a different metric due to their perception of the incentive structure.	The FDA ignores <Input> because they are focused on improving a different metric due to their perception of the incentive structure.
3.4.J	One	<Controller> received instructions from <Superior Controller> to execute <UCA>. <Controller> may have received negative feedback from previous instances of questioning directives from <Superior Controller>.	The FDA received instructions from the Federal Government to delay the provision of the corrective action. The FDA may have received negative feedback from previous instances of questioning directives from the Federal Government
3.4.K	One	<Controller> does not check the <Input> source regularly because it rarely updates with valuable information.	The FDA does not check the <Input> source regularly because it rarely updates with valuable information.
3.4.L	One	<Controller> does not trust that the <Input> they are receiving is accurate because they believe the source of the <Input> is withholding or editing the data.	The FDA does not trust that the <Input> they are receiving is accurate because they believe the source of the <Input> is withholding or editing the data.
3.4.M	Two	Obtaining an improved <Input> source may have been difficult or costly.	Obtaining an improved <Input> source may have been difficult or costly.
3.4.N	Two	The <Controller> did not have the responsibility to question the <Input>; instead, it had the responsibility to make control decisions based on the <Input>.	The FDA did not have the responsibility to question the <Input>; instead, it had the responsibility to make control decisions based on the <Input>.
3.4.O	Two	The <Controller> has the responsibility to request updated <Input>, but does not realize that their <Input> is outdated.	The FDA has the responsibility to request updated <Input>, but does not realize that its <Input> is outdated.
3.4.P	Two	The <Controller>'s mental model is that <Input> is a direct indication of system status; however, the <Input> is a measure of a different construct that may not always align.	The FDA's mental model is that <Input> is a direct indication of system status; however, the <Input> is a measure of a different construct that may not always align.
3.4.Q	Two	<Controller> believed that the inputs used to monitor the system state were based on different underlying data sources. However, there were underlying relationships between the Inputs such that if one was incorrect, the others were also incorrect.	The FDA believed that the inputs used to monitor the system state were based on different underlying data sources. However, there were underlying relationships between the Inputs such that if one was incorrect, the others were also incorrect.

ID	Class	Archetype	Scenario prompt The FDA issues a corrective action to an OTC manufacturer too late, following a series of inappropriate results from an OTC device, because…
3.4.R	Two	<Controller> received <Input>, but the <Input> could be true in multiple system states. Therefore, they needed to conduct tests. <Controller> believed that <UCA> would be safe and give them important information on the state of the system. However, given <Context>, the test was unsafe.	FDA received <Input>, but the <Input> could be true in multiple system states. Therefore, they needed to conduct tests. The FDA believed that not providing a corrective action earlier would be safe and give them important information on the state of the system. However, given <Context>, the delay of the corrective action was unsafe.
3.4.S	Two	<Input> could not provide <Controller> with information about the effects of the available controls.	<Input> could not provide the FDA with information about the effects of the available controls.
3.4.T	Two	<Controller> relies on <Input> to make a decision, but the <Input> is insufficient to prompt them to switch to a different goal when necessary.	The FDA relies on <Input> to make a decision, but the <Input> is insufficient to prompt them to switch to a different goal when necessary.
3.4.U	Two	<Controller> does not receive <Input> because the people who could send the report do not believe that sending the information is the best use of their time.	The FDA does not receive <Input> because the people who could send the report do not believe that sending the information is the best use of their time.
3.4.V	Two	<Controller> does not receive <Input> because the people who could send the report believe they could be disciplined for submitting a report due to prior experience.	The FDA does not receive <Input> because the people who could send the report believe they could be disciplined for submitting a report due to prior experience.
3.4.W	Two	Because the <Controlees> supervised by <Controller> do not trust <Controller>, they do not share complete information that <Controller> needs to make decisions.	Because the OTC test manufacturers supervised by the FDA do not trust the FDA, they do not share complete information that the FDA needs to make decisions.
3.4.X	Two	<Controller> is unaware of the actual processes used to complete a task. The <UCA> may have been safe in the context of the process the <Controller> has documented; however, workarounds changed the context, making the <UCA> unsafe. Workarounds may not be communicated to higher-level controllers.	The FDA is unaware of the actual processes used to complete a task. The FDA not providing a corrective action may have been safe in the context of the process the FDA has documented; however, workarounds changed the context, making the delay of a corrective action unsafe. Workarounds may not be communicated to higher-level controllers.
3.4.Y	Three	<Control path> only sends control actions after they are verified by another <Controller>who disapproved of the <SCA>	<Control path> only sends control actions after they are verified by another <Controller>, which did not approve of the corrective action in time

ID	Class	Archetype	Scenario prompt The FDA issues a corrective action to an OTC manufacturer too late, following a series of inappropriate results from an OTC device, because…
3.4.Z	Three	<Controller> issues a <SCA>, but the <Controlee>to which they issue it has a different goal for system performance due to previous controls, and they ignore or misinterpret the <SCA>.	The FDA provides a corrective action in time, but the OTC test manufacturer to which they issue it has a different goal for system performance due to previous controls, and they ignore or misinterpret the corrective action.
3.4.AA	Four	<Controlee> interprets the control in a different way than was intended by the <Controller> due to mismatched mental models.	The OTC test manufacturer interprets the control in a different way than was intended by the FDA due to mismatched mental models.
3.4.BB	Four	<Controlee> receives <SCA>, but the <SCA> may be generic, and the <Controlee>is unable to translate the general advice into their mental model of their system.	OTC test manufacturer receives a corrective action in time, but the corrective action may be generic, and the OTC test manufacturer is unable to translate the general advice into their mental model of their system.
3.4.CC	Four	<Controller> issued <SCA> in a format that did not catch the attention of the <Controlee>. The control might have been buried in other less critical information, or in a format that <Controlee> believes usually does not contain useful information.	The FDA issued a corrective action in a format that did not catch the attention of the OTC test manufacturer. The control might have been buried in other less critical information, or in a format that the OTC test manufacturer believes usually does not contain useful information.
3.4.DD	Four	<Controller> believes that another task is a higher priority. <Controlee> may not have made the importance of <SCA> clear enough to redirect the energy and attention of <Controller>.	The FDA believes that another task is a higher priority. The OTC test manufacturer may not have made the importance of the FDA providing a corrective action clear enough to redirect the energy and attention of the FDA.
3.4.EE	Four	<Controlee> received <SCA> but had not or rarely received this command previously and waited for confirmation to execute the requested action.	The OTC test manufacturer received a corrective action from the FDA in time, but had not or rarely received this command previously, and waited for confirmation to execute the requested action.

The scenario prompts from Table 4. 4 were then evaluated to identify whether or not that scenario is reasonable in the OTC test system. The following five scenarios were identified as reasonable and expanded into full scenarios. An additional six detailed scenarios are included in Appendix D.

Table 4. 5 shows a completed scenario for the scenario prompt 3.4.B.

Table 4. 5 contains the completed scenario based on scenario prompt 3.4.B for UCA 3.4.

UCA	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device
Scenario ID	3.4.B
Scenario Prompt	The FDA had an accurate mental model before a system change; however, once the system behavior changed, the controller's mental model did not. Therefore, they interpreted <Input> incorrectly.
Completed Scenario	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device because the FDA's mental model of user behavior with this OTC device is no longer accurate. Due to pressure on users to ensure a certain result on the test, patients may develop and publicize "hacks" to obtain the desired outcome. Even if the identified mechanism is something the FDA can fix with a corrective action, the FDA may not have feedback in place to monitor patient use of OTC devices (Lorch, 2021).

Table 4. 6 shows a completed scenario for the scenario prompt 3.4.I.

Table 4. 6 contains the completed scenario based on scenario prompt 3.4.I for UCA 3.4.

UCA	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device
Scenario ID	3.4.I
Scenario Prompt	The FDA ignores <Input> because they are focused on improving a different metric due to their perception of the incentive structure.
Completed Scenario	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device because the FDA is focused on approving more devices. The FDA may receive directions from the federal government to be less punitive to developers and focus on approving new devices instead of identifying problems with current tests on the market. These directions may take the form of a change to the regulatory structure or through direct or indirect social pressure (Foley, 2022).

Table 4. 7 shows a completed scenario for the scenario prompt 3.4.J.

Table 4. 7 contains the completed scenario based on scenario prompt 3.4.J for UCA 3.4.

UCA	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device

Scenario ID	3.4.J
Scenario Prompt	The FDA received instructions from the Federal Government to delay the provision of the corrective action. The FDA may have received negative feedback from previous instances of questioning directives from the Federal Government
Completed Scenario	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device because the FDA received instructions from the federal government not to provide a corrective action. The FDA may have received negative feedback from previous instances of applying corrective actions to OTC devices and wants to ensure that they are able to continue its work in other areas. The Federal Government may not understand the safety risks of underperforming OTC devices and may be incentivized to advocate on behalf of companies that may feel their business would be unduly disrupted. The FDA may not have a corrective action that is strong enough to improve the device's performance without significant pushback from the affected company.

Table 4. 8 shows a completed scenario for the scenario prompt 3.4.R.

Table 4. 8 contains the completed scenario based on scenario prompt 3.4.R for UCA 3.4.

UCA	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device
Scenario ID	3.4.R
Scenario Prompt	FDA received <Input>, but the <Input> could be true in multiple system states. Therefore, they needed to conduct tests. The FDA believed that not providing a corrective action earlier would be safe and give them important information on the state of the system. However, given <Context>, the delay of the corrective action was unsafe.
Completed Scenario	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device despite receiving reports that the device did not perform adequately from several users. However, such reports could be true even if the tests performed as expected. All laboratory tests have an error margin such that some false positives or false negatives are expected. Therefore, the FDA needs to conduct additional investigations to identify whether the devices were truly underperforming (Todsen et al., 2023). The FDA may believe that delaying the corrective action would enable it to conduct more thorough investigations. However, because the OTC tests are malfunctioning, the delay allows more unsafe tests to flood the markets, making future corrective actions less impactful.

Table 4. 9 shows a completed scenario for the scenario prompt 3.4.U.

Table 4. 9 contains the completed scenario based on scenario prompt 3.4.U for UCA 3.4.

UCA	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device

Scenario ID	3.4.U
Scenario Prompt	The FDA does not receive <Input> because the people who could send the report do not believe that sending the information is the best use of their time.
Completed Scenario	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device because the FDA does not receive adequate feedback. The individuals who have the best understanding of how to report problems with OTC tests include doctors, laboratory technicians, and other healthcare professionals who may be familiar with reporting mechanisms for other FDA-approved products. However, healthcare professionals will rarely interact with a patient's OTC tests unless the patient comes in for a confirmation test. A doctor may run a confirmation test on a patient that shows that the OTC test gave an inaccurate answer. However, the doctor is not required to submit a report based on the confirmation test. Furthermore, even if the doctor wanted to submit a report, the doctor may not have enough information about the device the patient used, and the doctor may not trust that the patient conducted the test adequately.

These eleven detailed scenarios (five above, six in appendix D) are in addition to the six scenarios identified from the same UCA in the original study, which are presented in Appendix E. The scenario prompts, generated from a rigorous analysis of a human controller model in a control loop, increased the thoroughness of the STPA results in a sociotechnical system. Although the results in this chapter are only shown for a small portion of the system, the process can be easily applied to any of the other UCAs identified.

Chapter 5: Conclusions

Over the 20th century, safety standards and hazard analyses that focused on improving simple systems did improve safety in many industries (N. Leveson, 2011), including healthcare (Institute of Medicine (US), 2008), manufacturing (Hofmann et al., 2017), and automotive design (Akamatsu et al., 2013). However, industries still experience major accidents that result in significant loss of life, monetary losses, and environmental damage (Bates & Singh, 2018; Gelles, 2020; Leigh, 2011; Witte, 2024). Major accidents today are often a result of interactions between system components, rather than a failure of an individual component (Carayon, 2006; Gurses et al., 2012; N. G. Leveson et al., 2012). Often, the interactions that lead to accidents are between humans and technical system components or in the safety management system (Fossum et al., 2018; N. Leveson, 2011). Because sociotechnical systems control the safety of every industry, we need better ways to systemically anticipate design problems stemming from the unsafe design of sociotechnical systems that manage and operate technical devices.

To identify unsafe interactions before major accidents occur, we need improved methods of hazard analysis that can manage the complexity of the systems we design today. STPA is a hazard analysis method that has made significant progress in closing this gap by modeling the system using control-feedback loops. The resulting causal scenarios enable the system to be re-designed in a way that mitigates the potential hazards (Baker, 2022; France, 2017; N. G. Leveson et al., 2012; Thomas, 2023).

However, the process of identifying causal scenarios for human controllers is difficult for analysts who do not have significant training in human factors (Czaja & Nair, 2012). Many engineers do not receive human factors training (Dadmohammadi et al., 2017) and may therefore be unable to complete as thorough an analysis as necessary.

This thesis bridges the gap by providing a clear process to facilitate the identification of causal scenarios in STPA. By identifying and modeling critical human factors considerations within a control loop and providing detailed scenario archetypes for each of the four classes of scenarios, this process enables non-human factors specialists to thoroughly identify detailed scenarios stemming from the design of the sociotechnical system.

5.1 Contributions

Humans are amazing problem solvers, which is why sociotechnical systems, like healthcare, rely on their staff to avoid the consequences of unsafe system design (Tucker & Edmondson, 2003). Rarely do humans intentionally choose unsafe actions. When accidents do occur after what seems like a human error, there is usually a rational explanation for why the unsafe action seemed reasonable at the time(Carayon, 2006; Gurses et al., 2012; Rasmussen et al., 1990). While human cognition is complex and not fully understood, it is possible to improve the design of sociotechnical systems based on well-established human factors principles.

Human factors research has led to improvements in the designs of countless devices and interfaces (Meister, 2018). However, the research on human decision-making must also be

applied to humans beyond the system operators (Hofmann et al., 2017). The safety management system surrounding technical systems makes important decisions about regulations, available resources, scheduling, system design, and many others. Each of these decisions has a significant effect on the ultimate safety of the system, but the systems in which these decisions are made are rarely subject to the same analysis as the technology itself (Carayon, 2006). By focusing on the way in which higher-level decision-makers interact, this thesis provides a way to improve the design of sociotechnical systems.

By identifying and modeling critical human factors considerations within a control loop and providing detailed scenario archetypes for each of the four classes of scenarios, this thesis presents a process for non-human factors specialists to use in thoroughly identifying and detailing scenarios stemming from the design of sociotechnical systems that could result in losses.

5.2 Limitations

The process described in this thesis was tested by comparing the results of an earlier analysis with those obtained using the new process. Although the new process identified significantly more scenarios, this comparison was not a rigorous validation method. Validations with more controls are necessary to fully identify the strengths and weaknesses of the process provided.

Furthermore, the process outlined in this thesis is limited to identifying causal scenarios. STPA analysts who are not human-factors experts may also struggle to identify UCAs or to complete other steps of the analysis. More guidance may be needed to improve the results of the earlier steps in an STPA analysis.

Finally, the scenario prompts occasionally result in repeated content. While the model of the human controller shown in Chapter 3 is a useful model of human behavior in a system, there is an overlap across the different cognitive components. For example, information processing is influenced by the goals the controller has because a controller will direct more attention to sensory information from areas where they expect the most valuable information to come from. Therefore, the prompts from these two categories may lead to a duplication of a causal scenario.

5.3 Future Work

The one hundred and thirteen scenario archetypes provided in this thesis provide an excellent way to ensure thoroughness when analyzing human controllers. However, given the number of UCAs in a system, the number of scenarios this process provides could easily be overwhelming. An improved version of this process would enable the analyst to efficiently identify which of the scenario archetypes are most appropriate for the current UCA. For example, the method of identifying scenarios provided in this thesis lacks the innate ability to filter for applicability by context. In a more robust and usable version of this process, the STPA analyst would be able to enter information from the first three stages of STPA, including controllers, control actions, losses, hazards, and have a tool to help them identify the most applicable scenario archetypes. One simple example of such a context filter is the ability to identify when a control action is shared between controllers and only provide prompts to consider shared control archetypes in

those cases. Another opportunity to provide more guidance would be the ability to filter for scenario archetypes that are only relevant to certain categories of UCAs (applies or does not apply, for example).

Furthermore, the guidance in this thesis primarily refers to scenarios. There may be opportunities to use the human controller model in Chapter 3 to provide a more rigorous process for identifying UCAs. One potential piece of such a process would be an improved method for maintaining coherence as the analysis iteratively goes between scenario and UCA identification. For example, when identifying scenarios in a class four scenario, one may identify that the reason the process would not respond to the controller's safe control action is due to another controller's UCA. A more robust model and process would provide a mechanism for tracking newly generated UCAs and ensuring that they are thoroughly analyzed as well.

Finally, as discussed in the limitations section, the process presented in this thesis requires further validation. One potential validation method would be to compare the results of two similar groups conducting an STPA on the same sociotechnical system, with and without the proposed process. The outputs could then be compared to determine whether the group using the provided process was able to complete a more thorough analysis.

5.4 Conclusions

Humans make many high-level decisions that have a profound impact on safety. These decisions are rarely systemically analyzed or subject to safety reviews. This thesis provides a process to thoroughly consider human factors in the process of STPA of a sociotechnical system. This process identifies how the design of the system could lead to hazards due to interactions of humans, both at the operator level and at the managerial level. The process provided in this thesis enables STPA analysts who may have limited understanding of human factors to improve the thoroughness of their analysis of sociotechnical systems.

References

Akamatsu, M., Green, P., & Bengler, K. (2013). Automotive technology and human factors research: Past, present, and future. *International Journal of Vehicular Technology*, *2013*(1), 526180. https://doi.org/10.1155/2013/526180

Baddeley, A. D., & Hitch, G. (1974). Working memory. In G. H. Bower (Ed.), *Psychology of learning and motivation* (Vol. 8, pp. 47–89). Academic Press. https://doi.org/10.1016/S0079-7421(08)60452-1

Baker, E. W. (2022). *Safety in hospital medication administration applying STAMP processes* [Masters Thesis, Massachusetts Institute of Technology]. https://hdl.handle.net/1721.1/143213

Bates, D. W., & Singh, H. (2018). Two decades since to err is human: An assessment of progress and emerging priorities in patient safety. *Health Affairs*, *37*(11), 1736–1743. https://doi.org/10.1377/hlthaff.2018.0738

Bertalanffy, L. von. (2009). *General system theory: Foundations, development, applications* (Rev. ed., 17. paperback print). Braziller.

Birch, D., Miller, E., & Bradley, T. (2023). Human reliability analysis using a human factors hazard model. *Journal of System Safety*, *58*(2), 7–29. https://doi.org/10.56094/jss.v58i2.251

Carayon, P. (2006). Human factors of complex sociotechnical systems. *Applied Ergonomics*, *37*(4), 525–535. https://doi.org/10.1016/j.apergo.2006.04.011

Carayon, P., Hoonakker, P., & Smith, M. J. (2012). Human factors in organizational design and management. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (4th ed). Wiley.

Carayon, P., Hundt, A. S., Karsh, B., Gurses, A. P., Alvarado, C. J., Smith, M., & Brennan, P. F. (2006). Work system design for patient safety: The SEIPS model. *Quality & Safety in Health Care*, *15*(Suppl 1), i50–i58. https://doi.org/10.1136/qshc.2005.015842

Carroll, M. B., & Sanchez, P. L. (2021). Decision making with conflicting information: Influencing factors and best practice guidelines. *Theoretical Issues in Ergonomics Science*, *22*(3), 296–316. https://doi.org/10.1080/1463922X.2020.1764660

Center for Devices and Radiological Health. (2021, August 25). *Over-the-Counter (OTC) Medical Devices: Considerations for Device Manufacturers*. FDA; FDA. https://www.fda.gov/medical-devices/products-and-medical-procedures/over-counter-otc-medical-devices-considerations-device-manufacturers

CMS. (2022). *Over-The-Counter (OTC) Home Testing and CLIA Applicability Frequently Asked Questions*. Centers for Medicare & Medicaid Services (CMS). https://www.cms.gov/files/document/over-counter-otc-home-testing-and-clia-applicability.pdf

Colquitt, J. A., LePine, J. A., & Wesson, M. J. (2011). *Organizational behavior: Improving performance and commitment in the workplace* (2nd ed.). McGraw-Hill Irwin.

Cook, J. (2019, March 11). What you need to know about the Boeing 737 MAX 8 that crashed in Ethiopia. *ABC News*. https://abcnews.go.com/Politics/boeing-737-max-crashed-ethiopia/story?id=61606129

Coy, P. (2024, March 26). Accidents like the Baltimore bridge collision are far too common. *The New York Times*. https://www.nytimes.com/live/2024/03/26/opinion/thepoint

Crawley, F., & Tyler, B. (Eds.). (2015). *HAZOP: Guide to best practice* (3rd Ed.). Elsevier. DOI:10.1016/B978-0-323-39460-4.00010-4

Czaja, S. J., & Nair, S. N. (2012). Human factors engineering and systems design. In *Handbook of Human Factors and Ergonomics* (1st ed.). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118131350

Dadmohammadi, Y., Salehi, S., Kiran, R., Jeon, J., Kang, Z., Cokely, E. T., & Ybarra, V. (2017, October 9). *Integrating human factors into petroleum engineering's curriculum: Essential training for students*. SPE Annual Technical Conference and Exhibition. https://doi.org/10.2118/187241-MS

De Keyser, V., & Woods, D. D. (1990). Fixation errors: Failures to revise situation assessment in dynamic and risky systems. In A. G. Colombo & A. S. de Bustamante (Eds.), *Systems Reliability Assessment* (pp. 231–251). Springer Netherlands. https://doi.org/10.1007/978-94-009-0649-5_11

Demerouti, E., Bakker, A. B., Nachreiner, F., & Schaufeli, W. B. (2001). The job demands-resources model of burnout. *Journal of Applied Psychology*, *86*(3), 499–512. https://doi.org/10.1037/0021-9010.86.3.499

Dewar, J. A. (2002). Step 2: Identifying load-bearing, vulnerable assumptions. In *Assumption-based planning: A tool for reducing avoidable surprises* (pp. 64–90). Cambridge University Press. https://doi.org/10.1017/CBO9780511606472.005

Dismukes, K. (2006). Concurrent task management and prospective memory: Pilot error as a model for the vulnerability of experts. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *50*(9), 909–913. https://doi.org/10.1177/154193120605000910

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 32–64. https://doi.org/10.1518/001872095779049543

Endsley, M. R. (2012). Situational awareness. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (4th ed). Wiley.

Erik, H. (2012). Task analysis: Why, what, and how. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (4th ed). Wiley.

FDA, FCC, & ONC. (2014). *FDASIA Health IT Report: Proposed Strategy and Recommendations for a Risk-Based Framework*. https://www.fda.gov/media/87886/download

Foley, K. E. (2022, August 24). *Trump White House exerted pressure on FDA for Covid-19 emergency use authorizations, House report finds*. POLITICO. https://www.politico.com/news/2022/08/24/trump-white-house-exerted-pressure-on-fda-for-covid-19-emergency-use-authorizations-house-report-finds-00053428

Food and Drug Administration. (2016). *Applying Human Factors and Usability Engineering to Medical Devices* [FDA-2011-D-0469]. https://www.fda.gov/media/80481/download

Fossum, K. R., Danielsen, B.-E., Aarseth, W., & Johnsen, S. O. (2018). A project management issue of new technology developments: A case study on lack of human factors' attention in human–robot interaction. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, *232*(2), 164–173. https://doi.org/10.1177/1748006X17728601

France, M. (2017). *Engineering for humans: A new extension to STPA* [Masters Thesis, Massachusetts Institute of Technology].

https://dspace.mit.edu/bitstream/handle/1721.1/112357/1008570407-MIT.pdf?sequence=1&isAllowed=y

Gelles, D. (2020, January 29). Boeing expects 737 Max costs will surpass $18 billion. *The New York Times*. https://www.nytimes.com/2020/01/29/business/boeing-737-max-costs.html

Gobet, F., & Simon, H. A. (1998). Expert chess memory: Revisiting the chunking hypothesis. *Memory*, *6*(3), 225–255. https://doi.org/10.1080/741942359

Gurses, A. P., Ozok, A. A., & Pronovost, P. J. (2012). Time to accelerate integration of human factors and ergonomics in patient safety. *BMJ Quality & Safety*, *21*(4), 347–351. https://doi.org/10.1136/bmjqs-2011-000421

HFACS, Inc. (n.d.). *The HFACS Framework*. Retrieved November 27, 2024, from https://hfacs.com/hfacs-framework.html

HFES. (2025). *Academic Programs*. Human Factors and Ergonomics Society. https://www.hfes.org/Education-Career-Resources/Academic-Programs

Hofmann, D. A., Burke, M. J., & Zohar, D. (2017). 100 years of occupational safety research: From basic protections and work analysis to a multilevel view of workplace safety and risk. *Journal of Applied Psychology*, *102*(3), 375–388. https://doi.org/10.1037/apl0000114

Holden, R. J., & Carayon, P. (2021). SEIPS 101 and seven simple SEIPS tools. *BMJ Quality & Safety*, *30*(11), 901–910. https://doi.org/10.1136/bmjqs-2020-012538

Institute of Medicine (US). (2008). The Changing Nature of Health Care. In *Evidence-Based Medicine and the Changing Nature of Healthcare: 2007 IOM Annual Meeting Summary*. National Academies Press (US). https://www.ncbi.nlm.nih.gov/books/NBK52825/

Intriligator, J. (2022). Multidimensional task analysis (MTA): A new design method for human factors practitioners. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *66*(1), 2229–2233. https://doi.org/10.1177/1071181322661534

Jalali, M., Dehghan, H., Habibi, E., & Khakzad, N. (2023). Application of "Human Factor Analysis and Classification System" (HFACS) model to the prevention of medical errors and adverse events: A systematic review. *International Journal of Preventive Medicine*, *14*, 127. https://doi.org/10.4103/ijpvm.ijpvm_123_22

Jean, S., Burnham, C.-A. D., Chapin, K., Garner, O. B., Pant Pai, N., Turabelidze, G., & Butler-Wu, S. (2021). At-Home Testing for Infectious Diseases: The Laboratory Where You Live. *Clinical Chemistry*, *68*(1), 19–26. https://doi.org/10.1093/clinchem/hvab198

Joint Commission Resources. (2020). *Joint Commission International Survey Process Guide for Hospitals: Including Academic Medical Center Hospitals* (7th edition). https://prod-ebooks-s3.s3.amazonaws.com/9935462109906761.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAUL46FQS2N4CUAQLP%2F20241024%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241024T214159Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=e6f0fdd7474cfbd3f7fc26f10e2999a19cf24c8622209f37b9b4e4fd56a0a28e

Kariuki, S. G., & Löwe, K. (2007). Integrating human factors into process hazard analysis. *Reliability Engineering & System Safety*, *92*(12), 1764–1773. https://doi.org/10.1016/j.ress.2007.01.002

Karwowski, W. (2012). The Discipline of Human Factors and Ergonomics. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (4th ed). Wiley.

Kelman, B. (2022, March 25). Former nurse found guilty in accidental injection death of 75-year-old patient. *NPR*. https://www.npr.org/sections/health-shots/2022/03/25/1088902487/former-nurse-found-guilty-in-accidental-injection-death-of-75-year-old-patient

Kent, A. J. (2021). When topology trumped topography: Celebrating 90 years of Beck's underground map. *The Cartographic Journal*, *58*(1), 1–12. https://doi.org/10.1080/00087041.2021.1953765

Klein, G. (2008). Naturalistic decision making. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *50*(3), 456–460. https://doi.org/10.1518/001872008X288385

Knox, N. W., & Eicher, R. W. (1976). *MORT User's Manual* (No. ERDA-76-45-4). Aerojet Nuclear Company. https://www.osti.gov/servlets/purl/7279266

Korbmacher, M., Azevedo, F., Pennington, C. R., Hartmann, H., Pownall, M., Schmidt, K., Elsherif, M., Breznau, N., Robertson, O., Kalandadze, T., Yu, S., Baker, B. J., O'Mahony, A., Olsnes, J. Ø.-S., Shaw, J. J., Gjoneska, B., Yamada, Y., Röer, J. P., Murphy, J., … Evans, T. (2023). The replication crisis has led to positive structural, procedural, and community changes. *Communications Psychology*, *1*(1), 1–13. https://doi.org/10.1038/s44271-023-00003-2

Kypriotaki, A. (2024, March 27). Baltimore bridge incident: The day after. *SAFETY4SEA*. https://safety4sea.com/baltimore-bridge-incident-the-day-after/

Langewiesche, W. (2019, September 18). What really brought down the Boeing 737 Max? *The New York Times*. https://www.nytimes.com/2019/09/18/magazine/boeing-737-max-crashes.html

Lehto, M. R., Nah, F. F.-H., & Yi, J. S. (2012). Chapter 7: Decision-Making Models, Decision Support, and Problem Solving. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (4th ed). Wiley.

Leigh, J. P. (2011). Economic burden of occupational injury and illness in the United States. *The Milbank Quarterly*, *89*(4), 728–772. https://doi.org/10.1111/j.1468-0009.2011.00648.x

Leveson, N. (2004). A new accident model for engineering safer systems. *Safety Science*, *42*(4), 237–270. https://doi.org/10.1016/S0925-7535(03)00047-X

Leveson, N. (2011). *Engineering a safer world: Systems thinking applied to safety*. MIT Press.

Leveson, N. (2023). *An introduction to system safety engineering*. The MIT Press.

Leveson, N. G., Fleming, C. H., Spencer, M., Thomas, J., & Wilkinson, C. (2012). Safety assessment of complex, software-intensive systems. *SAE International Journal of Aerospace*, *5*(1), 233–244.

Leveson, N., Thomas, J., Harrington, P., Rodrigo, R., Powell, S., & Keller, A. (2023). *System safety within laboratory data exchanges report*. http://psas.scripts.mit.edu/home/wp-content/uploads/2023/10/System-Safety-within-Laboratory-Data-Exchanges-Report.pdf

Leveson, N., Thomas, J., Harrington, P., Rodrigo, R., Powell, S., & Keller, A. (2024). *Addendum to the system safety within laboratory data exchanges report: Over-the-counter and point-of-care testing*. http://psas.scripts.mit.edu/home/wp-content/uploads/2023/10/System-Safety-within-Laboratory-Data-Exchanges-Report.pdf

Leveson, N., & Thomas, J. P. (2018). *STPA handbook*. MIT Partnership for Systems Approaches to Safety and Security. psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf

Lindner, A. K., Nikolai, O., Rohardt, C., Kausch, F., Wintel, M., Gertler, M., Burock, S., Hörig, M., Bernhard, J., Tobian, F., Gaeddert, M., Lainati, F., Corman, V. M., Jones, T. C., Sacks, J. A., Seybold, J., Denkinger, C. M., & Mockenhaupt, F. P. (2021). Diagnostic accuracy and feasibility of patient self-testing with a SARS-CoV-2 antigen-detecting rapid test. *Journal of Clinical Virology*, *141*, 104874. https://doi.org/10.1016/j.jcv.2021.104874

Lorch, M. (2021, July 6). How children are spoofing Covid-19 tests with soft drinks. *BBC*. https://www.bbc.com/future/article/20210705-how-children-are-spoofing-covid-19-tests-with-soft-drinks

Luczak, H., Kabel, T., & Licht, T. (2012). Chapter 14: Task Design and Motivation. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (4th ed). Wiley.

Machol, Robert. E., & Miles, R. F. (1973). The engineering of large scale systems. In R. F. Miles (Ed.), *System concepts*. Wiley.

Majewicz, P. J., Blessner, P., Olson, B., & Blackburn, T. (2020). Estimating the probability of human error by incorporating component failure data from user-induced defects in the development of complex electrical systems. *Risk Analysis*, *40*(1), 200–214. https://doi.org/10.1111/risa.12798

McPhillips, J. C., Jacqueline Howard,Deidre. (2022, April 18). *Rise in at-home testing means we could be undercounting Covid-19 cases even more than before*. CNN. https://www.cnn.com/2022/04/18/health/covid-at-home-testing-data/index.html

Meister, D. (2018). *The history of human factors and ergonomics*. CRC Press. https://doi.org/10.1201/9781315276069

Merida, E. (2017, March 9). Boeing 737 MAX 8 Earns FAA Certification. *MediaRoom*. https://boeing.mediaroom.com/2017-03-09-Boeing-737-MAX-8-Earns-FAA-Certification

Montague, G., & Verdeja, A. (2021). *Automated external defibrillator systems and methods of use* (United States Patent No. US20210093876A1). https://patents.google.com/patent/US20210093876A1/en

Nazaruk, M., & John, T. (2020, November 1). *Closing the gap in human factors: Everybody has a role to play*. Journal of Petroleum Technology. https://jpt.spe.org/closing-gap-human-factors-everybody-has-role-play

Nemeth, C. P. (2004). The human-made environment. In *Human factors methods for design: Making systems human-centered*. CRC Press.

Nicas, J., Kitroeff, N., Gelles, D., & Glanz, J. (2019, June 1). Boeing Built Deadly Assumptions Into 737 Max, Blind to a Late Design Change. *The New York Times*. https://www.nytimes.com/2019/06/01/business/boeing-737-max-crash.html

NTSB. (2024). *Marine investigation preliminary report: Contact of containership Dali with the Francis Scott Key Bridge and subsequent bridge collapse* (No. DCA24MM031). https://www.ntsb.gov/investigations/Documents/DCA24MM031_PreliminaryReport%203.pdf

O'Laughlin, K., Espinosa, C. C., Smith-Jeffcoat, S. E., Koh, M., Khalil, G. M., Hoffman, A., Rebolledo, P. A., Schechter, M. C., Stewart, R. J., Silva, J. da, Biedron, C., Bankamp, B., Folster, J., Gargis, A. S., Bowen, M. D., Paulick, A., Wang, Y. F., Tate, J. E., Kirking, H. L., … Team, C. C.-19 E. R. G.-10 F. (2022). Specimen self-collection for SARS-CoV-2

testing: Patient performance and preferences—Atlanta, Georgia, August-October 2020. *PLOS ONE*, *17*(3), e0264085. https://doi.org/10.1371/journal.pone.0264085

Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, *349*(6251), aac4716. https://doi.org/10.1126/science.aac4716

Perry, S. J., Catchpole, K., Rivera, A. J., Parker, S. H., & Gosbee, J. (2021). 'Strangers in a strange land': Understanding professional challenges for human factors/ergonomics and healthcare. *Applied Ergonomics*, *94*, 103040. https://doi.org/10.1016/j.apergo.2019.103040

Pollard, J. (2024, March 26). What we know about the Baltimore bridge collapse. *AP News*. https://apnews.com/article/baltimore-key-bridge-collapse-what-to-know-127d6ae38d63561ca4c1f18b3d508ba6

Poon, E. G., Trent Rosenbloom, S., & Zheng, K. (2021). Health information technology and clinician burnout: Current understanding, emerging solutions, and future directions. *Journal of the American Medical Informatics Association*, *28*(5), 895–898. https://doi.org/10.1093/jamia/ocab058

Proctor, R. W., & Van Zandt, T. (2018). *Human factors in simple and complex systems* (3rd edition). CRC Press, Taylor & Francis Group.

Rasmussen, J. (1983). Skills, rules, and knowledge; Signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-13*(3), 257–266. IEEE Transactions on Systems, Man, and Cybernetics. https://doi.org/10.1109/TSMC.1983.6313160

Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to cognitive engineering*. New York : North-Holland. http://archive.org/details/informationproce0000rasm

Rasmussen, J. (1987). *Mental models and the control of actions in complex environments*. Risø National Laboratory.

Rasmussen, J. (1990). The role of error in organizing behaviour. *Ergonomics*, *33*(10–11), 1185–1199. https://doi.org/10.1080/00140139008925325

Rasmussen, J., Nixon, P., & Warner, F. (1990). Human error and the problem of causality in analysis of accidents [and discussion]. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *327*(1241), 449–462.

Rouse, W. B., & Morris, N. M. (1985). *On looking into the black box: Prospects and limits in the search for mental models* (Nos. 85–2; p. 61). Center for Man-Machine Systems Research. https://apps.dtic.mil/sti/pdfs/ADA159080.pdf

Sarter, N. B., & Woods, D. D. (1995). How in the World Did We Ever Get into That Mode? Mode Error and Awareness in Supervisory Control. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 5–19. https://doi.org/10.1518/001872095779049516

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding: An inquiry into human knowledge structures* (p. 248). Lawrence Erlbaum.

Schein, E. H. (2015). Organizational psychology then and now: Some observations. *Annual Review of Organizational Psychology and Organizational Behavior*, *2*(Volume 2, 2015), 1–19. https://doi.org/10.1146/annurev-orgpsych-032414-111449

Sharit, J. (2012). Human error and human reliability analysis. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (4th ed). Wiley.

Simon, H. A. (1957). *Models of man; Social and rational* (pp. xiv, 287). Wiley.

Smith, K. (2023). A (Brief) History of Health Policy in the United States. *Delaware Journal of Public Health*, *9*(5), 6–10. https://doi.org/10.32481/djph.2023.12.003

Stephans, R. A., & Talso, W. W. (Eds.). (1993). *System safety analysis handbook* (First Edition). System Safety Society.

Sterman, J. D. (1989). Misperceptions of feedback in dynamic decision making. *Organizational Behavior and Human Decision Processes*, *43*(3), 301–335. https://doi.org/10.1016/0749-5978(89)90041-1

Sterman, J. D. (2009). *Business dynamics: Systems thinking and modeling for a complex world* (Nachdr.). Irwin/McGraw-Hill.

Stringfellow, M. V., Dierks, M., & Leveson, N. (2009). Healthcare industry incentive structures pressure system operators to operate in a high-risk risk state. *Proceedings of the International System Dynamics Conference*. https://www.researchgate.net/publication/237410573_Healthcare_Industry_Incentive_Structures_Pressure_System_Operators_to_Operate_in_a_High-risk_Risk_State

Thomas, J. (2023, June). *Empirical evaluations of STPA in the aviation industry*. STAMP Workshop, Massachusetts Institute of Technology. http://psas.scripts.mit.edu/home/wp-content/uploads/2023/2023-06-06-1010__John-Thomas__PUB.pdf

Thomas, J. (2024). *STPA step 4 building scenarios: A formal scenario approach*. STAMP Virtual Conference. https://psas.scripts.mit.edu/home/wp-content/uploads/2024/STPA-Scenarios-New-Approach.pdf

Todsen, T., Jakobsen, K. K., Grønlund, M. P., Callesen, R. E., Folke, F., Larsen, H., Ersbøll, A. K., Benfield, T., Gredal, T., Klokker, M., Kirkby, N., & von Buchwald, C. (2023). COVID-19 Rapid Antigen Tests With Self-Collected vs Health Care Worker–Collected Nasal and Throat Swab Specimens: A Randomized Clinical Trial. *JAMA Network Open*, *6*(12), e2344295. https://doi.org/10.1001/jamanetworkopen.2023.44295

Tucker, A. L., & Edmondson, A. C. (2003). Why hospitals don't learn from failures: Organizational and psychological dynamics that inhibit system change. *California Management Review*, *45*(2), 55–72. https://doi.org/10.2307/41166165

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *Science*, *185*(4157), 1124–1131. https://doi.org/10.1126/science.185.4157.1124

Vidulich, M. A., & Tsang, P. S. (2012). Mental workload and situation awareness. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (4th ed). Wiley.

Vincoli, J. W. (2006). *Basic guide to system safety* (2nd ed). Wiley-Interscience.

Von Bertalanffy, L. (1972). The history and status of general systems theory. *The Academy of Management Journal*, *15*(4), 407–426. https://doi.org/10.2307/255139

Wang, Y., Liu, D., & Wang, Y. (2003). Discovering the capacity of human memory. *Brain and Mind*, *4*(2), 189–198. https://doi.org/10.1023/A:1025405628479

Weinberg, G. M. (2001). *An introduction to general systems thinking* (Silver anniversary ed). Dorset House.

Weinger, M. B., Gardner-Bonneau, D., Wiklund, M. E., & Kelly, L. M. (Eds.). (2011). *Handbook of human factors in medical device design*. CRC Press.

Wickens, C. D. (2002). Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*, *3*(2), 159–177. https://doi.org/10.1080/14639220210123806

Wickens, C. D., & Carswell, C. M. (2012). Information processing. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (4th ed). Wiley.

Wickens, C. D., Hollands, J. G., Banbury, S., & Parasuraman, R. (Eds.). (2013). *Engineering psychology and human performance* (4. ed., international ed). Pearson.

Wiklund, M. E. (2022). Reflecting on Four Decades of Progress in Applying Human Factors Engineering to Medical Devices. *Human Factors in Healthcare*, 100024. https://doi.org/10.1016/j.hfh.2022.100024

Williams, K. N., Fausett, C. M., Lazzara, E. H., Bitan, Y., Andre, A., & Keebler, J. R. (2023). Investigative approaches: Lessons learned from the RaDonda Vaught case. *Human Factors in Healthcare*, *4*, 100054. https://doi.org/10.1016/j.hfh.2023.100054

Wilson, J. R., Mills, A., Clarke, T., Rajan, J., & Dadashi, N. (2012). *Rail human factors around the world: Impacts on and of people for successful rail operations*. CRC Press.

Wilson, J. R., & Norris, B. J. (2005). Rail human factors: Past, present and future. *Applied Ergonomics*, *36*(6), 649–660. https://doi.org/10.1016/j.apergo.2005.07.001

Witte, B. (2024, May 2). Maryland officials release timeline, cost estimate, for rebuilding bridge. *AP News*. https://apnews.com/article/baltimore-bridge-collapse-body-found-cdd8441c5dff48028d1e141b943ca31e

Woods, D. (1995). *Cognitive demands and activities in dynamic fault management: Abductive reasoning and disturbance management*. 63–92. https://www.researchgate.net/publication/262401824_Cognitive_demands_and_activities_in_dynamic_fault_management_abductive_reasoning_and_disturbance_management

Woods, D. D. (2000). Perspectives on human error: Hindsight biases and local rationality. In F. T. Durso (Ed.), *Handbook of Applied Cognition* (p. 141(32)). Wiley. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b4f626576f05e623fae22426ffef140d11999841

World Health Organization. (2015). *Improving the quality of HIV-related point-of-care testing: Ensuring the reliability and accuracy of test results*. World Health Organization. https://iris.who.int/handle/10665/199799

Wu, D. T. Y., Xu, C., Kim, A., Bindhu, S., Mah, K. E., & Eckman, M. H. (2021). A Scoping Review of Health Information Technology in Clinician Burnout. *Applied Clinical Informatics*, *12*(03), 597–620. https://doi.org/10.1055/s-0041-1731399

Xi, Y. T., Yang, Z. L., Fang, Q. G., Chen, W. J., & Wang, J. (2017). A new hybrid approach to human error probability quantification–applications in maritime operations. *Ocean Engineering*, *138*, 45–54. https://doi.org/10.1016/j.oceaneng.2017.04.018

Zarei, E., Ghaffari, A., Nikoobar, A., Bastami, S., & Hamdghaddari, H. (2023). Interaction between physicians and the pharmaceutical industry: A scoping review for developing a policy brief. *Frontiers in Public Health*, *10*, 1072708. https://doi.org/10.3389/fpubh.2022.1072708

Appendix A: Detailed Control Structure of OTC diagnostics

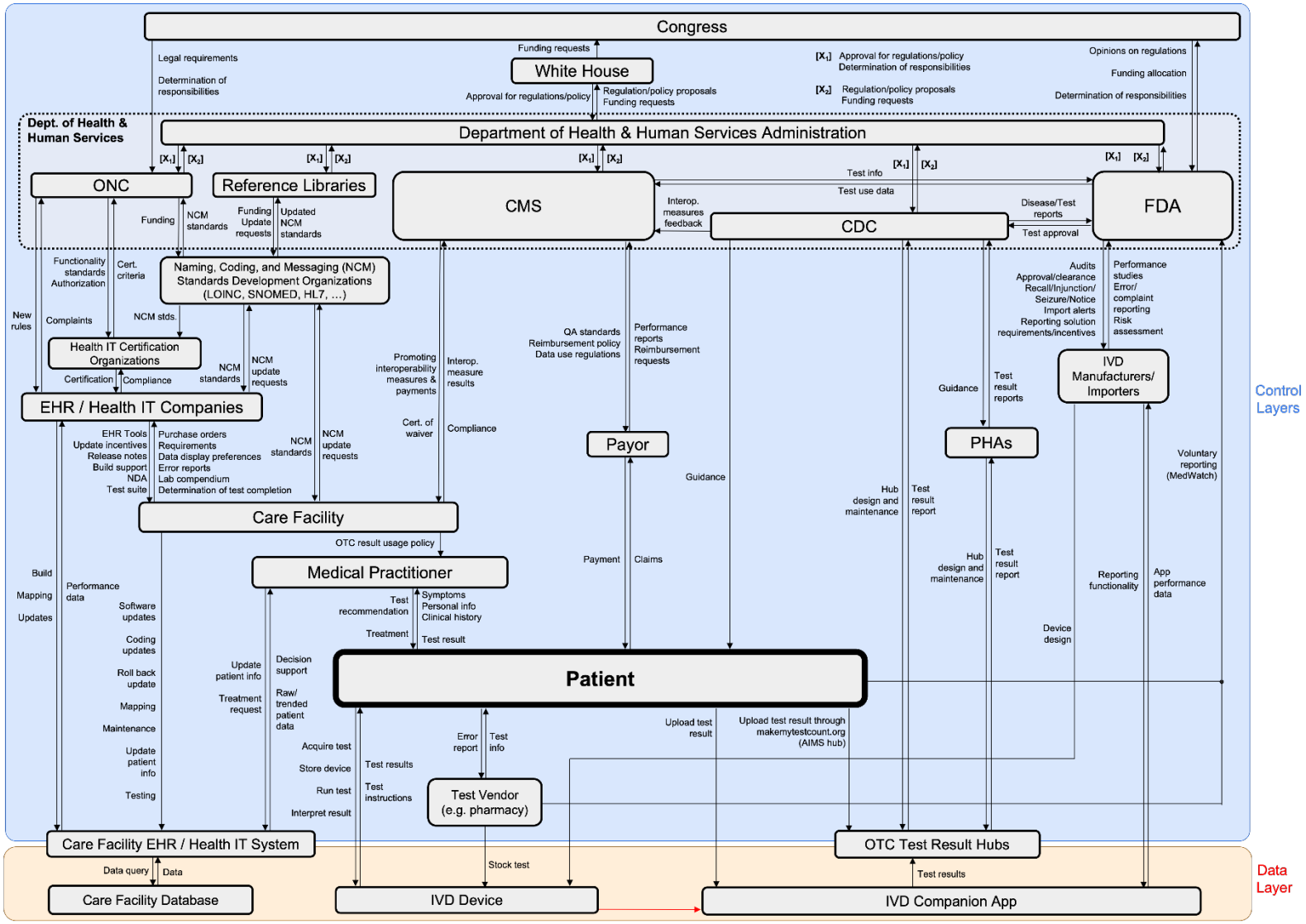

Figure Appendix-1. .Detailed control structure for OTC testing safety management system

Appendix B: UCAs for all controllers in control structure

Table Appx.1: Medical Practitioner UCAs

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long
Prescribe/ recommend OTC test to patient	UCA: Medical practitioner does not prescribe/ recommend an OTC test when the test is available and continuous monitoring improves patient care decisions, and patient cannot access traditional clinical laboratory testing	UCA: Medical Practitioner prescribes/recommends traditional clinical laboratory test that is inaccessible to patient (costs, location) when OTC tests are available and accessible to patient UCA: Medical practitioners prescribes/recommends test that is inappropriate to monitor patient's condition	UCA: Medical practitioner prescribes/recommends test too late to impact care decisions	UCA: Medical practitioner stops prescribing/ recommending or monitoring tests too soon to observe trend in patient condition
Provide treatment to patient	UCA: Medical practitioner does not provide treatment when patient needs treatment to avoid harm	UCA: Medical practitioner provides treatment when patient does not need any treatment UCA: Medical practitioner provides treatment that	UCA: Medical practitioner provides treatment too late to avoid patient harm UCA: Medical practitioner provides treatment before the patient's condition has been identified	UCA: Medical practitioner stops providing treatment too early, before patient condition has been resolved UCA: Medical practitioner provides treatment for too long after patient

		does not match the patient's condition		condition has been resolved

Table Appx.2: FDA UCAs

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long
Create regulations to authorize tests	The FDA does not update regulations to authorize tests when OTC technology is updated such that existing regulations are no longer sufficient. * The FDA does not create regulations to authorize tests that require the collection of information needed to monitor OTC test safety * The FDA does not create regulations that enforce the collection of information needed by other federal agencies (CDC). *	The FDA creates regulations that are insufficient to manage safety effectively. * The FDA creates regulations that conflict with the regulations of a different agency. * The FDA creates regulations that cannot be met by any OTC test. * The FDA creates regulations that require more work to administer than the resources available. * The FDA creates regulations that motivate regulated parties to behave unsafely*	The FDA removes regulations when they are still necessary to control safety. * The FDA provides changes to regulatory authorities too frequently to understand the impact of regulations on safety. *	N/A

Approve OTC Tests	FDA does not approve an OTC test when that test would enable better patient care decisions. The FDA authorizes a test too late to control the spread of an emergent disease	The FDA approves a test that does not conform to regulated standards* The FDA approves a test that users are unable to use safely* FDA approves an OTC test that does not facilitate data reporting by test users when that data is needed to inform public health decisions or test decisions.	FDA approves an OTC device too late to get critical data during a health emergency	N/A
Issue corrective action to an OTC manufacturer	FDA does not issue corrective action to an OTC manufacturer following a series of inappropriate results from an OTC device.	FDA issues a corrective action to an OTC manufacturer whose device is performing according to regulations such that patients lose access to a critical test. The FDA provides corrective actions that are insufficient to control the identified problems. *	FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device.	The FDA applies a corrective action to an OTC manufacturer for too long following the resolution of a problem with an OTC device.
Audit to OTC manufacturers	The FDA does not audit a company with manufacturing processes that do not meet FDA regulations. *	The FDA audits a company in a way that is insufficient to identify processes that do not meet regulations. *		N/A

Table Appx.3: OTC Manufacturer

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long
Release OTC device and instructions	**UCA:** OTC manufacturer does not release OTC test for which there is no adequate replacement in the market	**UCA:** OTC manufacturer releases OTC device that has been insufficiently tested on particular demographics (e.g., children) **UCA:** OTC manufacturer releases OTC device that was approved with inadequate validation data **UCA:** OTC manufacturer releases OTC device without accessible device usage instructions	**UCA:** OTC manufacturer releases OTC device too soon before sufficient testing has been performed on particular demographics (e.g., children)	N/A
Provide data collection mechanism	**UCA:** OTC manufacturer does not provide data collection mechanism when data is needed to inform regulatory or public health guidance	**UCA:** OTC manufacturer provides data collection mechanism that does not collect sufficient data to be used by PHAs or regulatory agencies **UCA:** OTC manufacturer provides data collection mechanism that patients are not willing to use	N/A	N/A
Select data standards	**UCA:** OTC manufacturer does not select data	**UCA:** OTC manufacturer selects a data standard that is	N/A	N/A

to implement in HIT system	standards to implement in HIT system when data needs to be shared with external groups	not compatible with data standards used in HIT systems from competitors or other stakeholders		

Table Appx.4: Test Vendor

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long
Sell or provide test to patient	UCA: Test vendor does not stock particular OTC tests when patients served by the vendor have no adequate replacement for it	UCA: Test vendor stocks OTC test that does not perform to expected performance levels UCA: Test vendor stocks OTC test without accessible instructions for when to purchase OTC test	UCA: Test vendor stocks OTC test too late after its results become valuable to inform patient care	UCA: Test vendor keeps stocking OTC test for too long after it is known that test does not perform to expected performance levels
Sell Medication	UCA: Vendor does not sell treatment to patient based off of OTC results that do not reflect the patient's condition	UCA: Vendor sells treatment to patient based off of OTC results that do not reflect the patient's condition		

Table Appx.5: Patient

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long

Acquire OTC test	**UCA:** Patient does not acquire OTC test when test would be helpful to inform patient decision-making	**UCA:** Patient acquires OTC test that is not the best/most appropriate test to diagnose a disorder/disease **UCA:** Patient acquires OTC test that does not perform to expected performance levels	**UCA:** Patient acquires OTC test too soon before learning what test can help inform their decision-making **UCA:** Patient acquires OTC test too late after test would be needed to inform patient decision-making	N/A
Follow OTC pre-test instructions or test procedures	**UCA:** Patient does not follow OTC pre-test instructions or test procedures when procedures are necessary for validity of test results	**UCA:** Patient follows OTC pre-test instructions or test procedures incorrectly when procedures are necessary for validity of test results **UCA:** Patient follows OTC pre-test instructions or test procedures when those procedures can harm their health	**UCA:** Patient follows OTC pre-test instructions or test procedures too soon before test is to be conducted, when timing of procedures is crucial for validity of test results **UCA:** Patient follows OTC pre-test instructions or test procedures too late before test is to be conducted, when timing of procedures is crucial for validity of test results	**UCA:** Patient stops following OTC pre-test instructions or test procedures too soon before test is to be conducted, when timing of procedures is crucial for validity of test results
Interpret test results	**UCA:** Patient does not interpret OTC test results when interpretation of results is necessary to inform patient's decision-making	**UCA:** Patient interprets OTC test result as invalid when test result was valid **UCA:** Patient interprets OTC test result as valid when test result was invalid **UCA:** Patient misinterprets OTC test result (units,	**UCA:** Patient interprets OTC test result before the test result is available/ready **UCA:** Patient interprets OTC test result too late after test accuracy window has ended	

		measured quantity, etc.) when correct interpretation of results is necessary to inform patient's decision-making		
Upload test results or personal information to database	**UCA:** Patient does not enter data into database when data is necessary to inform patient care **UCA:** Patient does not enter new personal data into OTC companion app when patient condition has changed	**UCA:** Patient enters incorrect data into OTC companion app when data is necessary to inform patient care **UCA:** Patient enters incomplete data into OTC companion app when data is necessary to inform patient care **UCA:** Patient enters other patient's personal data into database	**UCA:** Patient enters data too late after data is needed to inform patient care	N/A
Seek Medical treatment	**UCA:** Patient does not seek medical treatment when treatment is needed to avoid harm	Patient seeks medical treatment when treatment will cause harm	N/A	N/A

Table Appx.6: CDC

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long
Set standards	**UCA:** The CDC does not set standards for reporting OTC	**UCA:** The CDC sets standards for reporting OTC	**UCA:** The CDC sets standards for reporting	N/A

for reporting OTC data	data when data needs to be aggregated for use by the agency	data that patients or providers are unable to comply with **UCA:** CDC sets conflicting standards from other public health agencies for reporting OTC data	diagnostic data too late after providers or device manufacturers have already implemented other standards	
Provide healthcare guidance	**UCA:** The CDC does not provide healthcare guidance that may provide value to patients' cases	**UCA:** The CDC provides healthcare guidance that conflicts with current/previous guidance **UCA:** The CDC provides health guidance that is too stringent for institutions or individuals to follow	**UCA:** The CDC does not provide guidance in time to limit disease outbreak **UCA:** The CDC does provide healthcare guidance too early before sufficient data is received	**UCA:** The CDC removes healthcare guidance when the guidance is still relevant for patient safety outcomes **UCA:** The CDC maintains healthcare guidance when it is no longer relevant for patient safety outcomes
Identify and monitor outbreaks	**UCA:** The CDC does not identify a disease outbreak	**UCA:** The CDC identifies a non-existent outbreak	**UCA:** The CDC identifies an outbreak too late to apply corrective measures	**UCA:** The CDC stops monitoring an outbreak before it is over

Table Appx.7: Federal Government

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long
Provide regulatory authority	**UCA:** The Federal Government does not give any agency responsibility	**UCA:** The Federal Government assigns overlapping regulatory	**UCA**: The Federal Government removes a safety-critical responsibility	N/A

	over safety-critical component of OTC testing **UCA**: The Federal Government does not update a regulatory authority's statutory boundary when it is insufficient to enforce safety **UCA:** The Federal Government does not assign responsibility to a new agency when regulatory need is outside the scope of an existing agency **UCA:** The Federal Government does not expand a regulatory agency's statutory boundary to cover technologies that have emerged or undergone significant changes since previous statutory boundaries were enacted.	responsibilities to different agencies **UCA:** The Federal Government assigns a responsibility to a new agency when that responsibility is within the scope of an existing agency **UCA:** The Federal Government updates a regulatory authority's statutory boundary in a way that removes critical parts of a safe control loop design **UCA:** The Federal Government updates a regulatory authority's statutory boundary too frequently, causing confusion regarding regulatory scope **UCA:** The Federal Government expands the statutory boundary of multiple regulatory agencies to cover the same regulatory gap in a way that is not meaningfully different **UCA:** The Federal Government expands a	from an agency without reassigning it **UCA:** The Federal Government assigns a responsibility for longer than is relevant and helpful (resource waste) **UCA:** The Federal Government updates a regulatory authority's statutory boundary too soon after another regulatory boundary change **UCA:** The Federal Government updates a regulatory authority's statutory boundary too late after it is deemed insufficient to enforce safety **UCA:** The Federal Government expands a federal regulatory agency's statutory boundary too late after technologies have emerged or undergone significant changes since previous statutory boundaries were enacted	

		regulatory authority's statutory boundaries in a way that diminishes the safety of the regulated industry		
Provide funding	**UCA:** The Federal Government does not allocate sufficient funding to agencies whose services support safety-critical processes (or their oversight) **UCA:** The Federal Government does not issue sufficient funding for agencies to address emergent safety-critical reports	N/A	**UCA:** The Federal Government stops issuing funding to agencies whose services support safety-critical processes (or their oversight)	

Appendix C: Full list of all scenario prompts

UCA	Detailed Scenario Archetype	Scenario Prompt for UCA 3.4
Class One	<Controller> did not have the responsibility to <SCA> given <Context> indicated by <Input>	The FDA did not have the responsibility to provide a corrective action in time given <Context> indicated by <Input>
Class One	<Controller> knows <SCA> is needed but believes that <Peer Controller> is responsible for executing <SCA>. The control action may be unsafe if duplicated, so <Controller> does not execute the control.	The FDA knows that providing a corrective action in time is needed but believes that <Peer Controller> is responsible for providing the corrective action in time. The control action may be unsafe if duplicated, so FDA does not execute the control.
Class One	<Controller> knows that <SCA> is necessary. However, they believe that it has not already been executed by <Peer Controller>. <Peer Controller> executed the control action, but there is a time delay on the system impact. <Input> may only indicate whether the effect has occurred, rather than whether the control itself has been engaged.	The FDA knows that providing a corrective action in time is necessary. However, they believe that it has not already been executed by <Peer Controller>. <Peer Controller> executed the control action, but there is a time delay on the system impact. <Input> may only indicate whether the effect has occurred, rather than whether the control itself has been engaged.
Class One	<Controller> is unable to identify the correct control action associated with <Input>. The <Controller>may not have sufficient experience to have a well-developed mental model or may be stressed/distracted/fatigued, etc.	The FDA is unable to identify the correct control action associated with <Input>. The FDA may not have sufficient experience to have a well-developed mental model or may be stressed/distracted/fatigued, etc.
Class One	<Controller> has limited familiarity with the system and takes too long to identify what perceptual cues are useful for addressing the current system context.	The FDA has limited familiarity with the system and takes too long to identify what perceptual cues are useful for addressing the current system context.
Class One	The <Controller>'s training did not prepare them to identify the safe control action when <Input> emerged. This context was not covered in the training due to the <Context>.	The FDA's training did not prepare them to identify the safe control action when <Input> emerged. This context was not covered in the training due to the <Context>.

Class One	Over time, <Controller>'s mental model shifted to relying on <Input> to determine their action selection. <Controller> may not have experienced a system state where <Input> was accurate, but other forms of feedback were necessary to make a safe decision.	Over time, the FDA's mental model shifted to relying on <Input> to determine its action selection. The FDA may not have experienced a system state where <Input> was accurate, but other forms of feedback were necessary to make a safe decision.
Class One	The decision was needed quickly, and <Controller>'s mental model required more cognitive resources than they had available at the moment.	The decision was needed quickly, and the FDA's mental model required more cognitive resources than they had available at the moment.
Class One	The <Controller> had not experienced this <Context> before, but they had experienced the same <Input> before. Their mental model may therefore be unaware that the <Input> could correspond to multiple system states.	The FDA had not experienced this <Context> before, but they had experienced the same <Input> before. Their mental model may therefore be unaware that the <Input> could correspond to multiple system states.
Class One	<Controller> had an accurate mental model before a system change; however, once the system behavior changed, the controller's mental model did not. Therefore, they interpreted <Input> incorrectly.	The FDA had an accurate mental model before a system change; however, once the system's behavior changed, the controller's mental model did not. Therefore, they interpreted <Input> incorrectly.
Class One	<Controller> was inundated with <Input> and was unable to identify what was causing the system to change states. There may have been no direction from the system to guide the response or interpretation of the <Input>.	The FDA was inundated with <Input> and was unable to identify what was causing the system to change states. There may have been no direction from the system to guide the response or interpretation of the <Input>.
Class One	The <Controller>'s mental model did not update when the <Input> changed because they were focused on another source of <Input>.	The FDA's mental model did not update when the <Input> changed because they were focused on another source of <Input>.
Class One	The <Controller>'s mental model did not update when the <Input> changed because it conflicted with their initial hypothesis of the system state, and the <Input> was not salient enough to prompt a change in their hypothesis.	The FDA's mental model did not update when the <Input> changed because it conflicted with their initial hypothesis of the system state, and the <Input> was not salient enough to prompt a change in their hypothesis.

Class One	The <Controller>'s mental model did not update when the <Input> changed because other feedback sources were showing conflicting information, and the <Controller> was overwhelmed and could not determine which <Input> was the most relevant.	The FDA's mental model did not update when the <Input> changed because other feedback sources were showing conflicting information, and the FDA was overwhelmed and could not determine which <Input> was the most relevant.
Class One	<Controller> may not have expected to find valuable information from <Input>; they may have developed a habit over time of relying solely on other sources of Input.	The FDA may not have expected to find valuable information from <Input>; they may have developed a habit over time of relying solely on other sources of Input.
Class One	<Controller> receives more <Input> from <Peer Controller> than others. They therefore develop a mental model that <Input> represents the state of the system. However, another <Peer Controller> may experience a different perspective but not have the time or resources to report.	The FDA receives more <Input> from <Peer Controller> than others. They therefore develop a mental model that <Input> represents the state of the system. However, another <Peer Controller> may experience a different perspective but not have the time or resources to report.
Class One	<Controller> did not believe <Input> source because there was insufficient corroborating information, and the system state <Input> indicated was rare.	The FDA did not believe <Input> source because there was insufficient corroborating information, and the system state <Input> indicated was rare.
Class One	<Controller> believed that <UCA> would address the <Context> because of training or education.	The FDA believed that delaying the corrective action would address the <Context> because of training or education.
Class One	<Controller> believes that <Input> requires <UCA> because the most recent incidents where <Input> was true, <UCA> was used.	The FDA believes that <Input> requires delaying the corrective action because the most recent incidents where <Input> was true, a delay was required.
Class One	The <Controller> had less time than usual to make a decision. They may not have been able to consider all factors when making the decision.	The FDA had less time than usual to make a decision. They may not have been able to consider all factors when making the decision.
Class One	The <Controller>'s mental model did not update when the <Input> changed because it conflicted with their initial hypothesis of the system state, and the <Input> was not salient enough to prompt a change in their hypothesis.	The FDA's mental model did not update when the <Input> changed because it conflicted with their initial hypothesis of the system state, and the <Input> was not salient enough to prompt a change in their hypothesis.

Class One	The <Controller>'s mental model did not update when the <Input> changed because other feedback sources were showing conflicting information, and the <Controller> was overwhelmed and could not determine which <Input> was the most relevant.	The FDA's mental model did not update when the <Input> changed because other feedback sources were showing conflicting information, and the FDA was overwhelmed and could not determine which <Input> was the most relevant.
Class One	<Controller> knew that the system was in a new state due to <Input>. However, they did not know how this new state affected the impact of their controls. They may try <UCA> to test the system impact, but did not know that the effects of <UCA> would be hazardous given <Context>.	The FDA knew that the system was in a new state due to <Input>. However, they did not know how this new state affected the impact of their controls. They may try delaying the corrective action, but did not know that the effects of the delay would be hazardous given <Context>.
Class One	<Controller> has developed an incorrect script as a response to <Input>, either due to negative transfer, system changes, or training.	The FDA has developed an incorrect script as a response to <Input>, either due to negative transfer, system changes, or training.
Class One	The <Input> was associated with too many scripts, and the <Controller> could not determine which one was correct.	The <Input> was associated with too many scripts, and the FDA could not determine which one was correct.
Class One	Earlier <Input> prompted the <Controller>to invoke a script that did not involve checking or attending to <Input>.	Earlier <Input> prompted the FDA to invoke a script that did not involve checking or attending to <Input>.
Class One	The <Controller> lacked sufficient time and mental resources to identify a novel solution to the <Context>. No previous solution would have been safe in this context.	The FDA lacked sufficient time and mental resources to identify a novel solution to the <Context>. No previous solution would have been safe in this context.
Class One	The <Controller>'s mental model was not granular enough to run satisfactory "what if" tests to evaluate control options.	The FDA's mental model was not granular enough to run satisfactory "what if" tests to evaluate control options.
Class One	<Controller> was unaware that the <UCA> they chose would have side effects beyond the desired effect.	The FDA was unaware that delaying the corrective action would have side effects beyond the desired effect.

Class One	<Controller> prioritizes the best-case outcome and is unaware of <Context> that would change the effect of <UCA>. The existing <Input> may be technically correct, but it is insufficient to predict the outcome of <UCA>.	The FDA prioritizes the best-case outcome and is unaware of <Context> that would change the effect of delaying the corrective action. The existing <Input> may be technically correct, but it is insufficient to predict the outcome of delaying a corrective action.
Class One	Because the <Controller> perceived the risk of error to be minimal, they were less attentive to feedback such as <Input>.	Because the FDA perceived the risk of error to be minimal, they were less attentive to feedback such as <Input>.
Class One	<Controller> did not believe <Input>, because no loss had happened previously in their experience. <Input> was insufficient to change their mental model of the current system's behavior.	The FDA did not believe <Input>, because no loss had happened previously in their experience. <Input> was insufficient to change their mental model of the current system's behavior.
Class One	The <Controller> was experimenting to make a process more efficient. The <Controller> further reduced safety margins on <Control action> because they had received no negative feedback the last time <UCA> was executed.	The FDA was experimenting to make the process more efficient. The FDA further reduced safety margins on the control action because they had received no negative feedback the last time they delayed the corrective action.
Class One	<Controller> did not believe that <Input> indicated <UCA> would lead to negative consequences, as previous instances of <UCA> had not resulted in negative consequences.	FDA did not believe that <Input> indicated that delaying the corrective action in time would lead to negative consequences, as previous instances of delaying a corrective action had not resulted in negative consequences.
Class One	<Controller> did not realize that <Control Action> was set to be strict enough that any deviation from <Safe Control Action> would lead to a hazard.	FDA did not realize that <Control Action> was set to be strict enough that any deviation from <Safe Control Action> would lead to a hazard.
Class One	<Controller>'s goal of <goal> conflicts with the system level goal of <system goal> because <Context>.	The FDA's goal of <goal> conflicts with the system-level goal of <system goal> because <Context>
Class One	<Controller> misinterpreted the command from <Superior Controller> because they had the wrong goal in mind for system performance.	The FDA misinterpreted the command from the Federal Government because they had the wrong goal in mind for system performance

Class One	The <Controller> was incentivized to maximize a different parameter than what was best for the system. They may have known that the control would lead to an unsafe result, but believed the <UCA> would lead to the best outcome for them.	The FDA was incentivized to maximize a different parameter than what was best for the system. They may have known that delaying the corrective action would lead to an unsafe result, but they believed that delaying the corrective action would lead to the best outcome for them.
Class One	<Controller> ignores <Input> because they are focused on improving a different metric due to their perception of the incentive structure.	FDA ignores <Input> because they are focused on improving a different metric due to their perception of the incentive structure.
Class One	<Controller> received instructions from <Superior Controller> to execute <UCA>. <Controller> may have received negative feedback from previous instances of questioning directives from <Superior Controller>.	The FDA received instructions from the Federal Government to delay the corrective action. The FDA may have received negative feedback from previous instances of questioning directives from the Federal Government.
Class One	<Controller> received instructions from <Superior Controller>to execute <UCA>. <Superior Controller> may not have sent a <UCA> request before. Therefore, <Controller> did not question the instructions. <Controller> may have access to <Input>, but did not believe that it would change their decision.	The FDA received instructions from the Federal Government to delay the corrective action. The Federal Government may not have sent an unsafe delay request before. Therefore, the FDA did not question the instructions. FDA may have access to <Input>, but did not believe that it would change their decision.
Class One	<Controller> does not check the <Input> source regularly because it rarely updates with valuable information.	The FDA does not check the <Input> source regularly because it rarely updates with valuable information.
Class One	<Controller> does not trust <Input> because it is inconsistent or has been inaccurate recently.	The FDA does not trust <Input> because it is inconsistent or has been inaccurate recently.
Class One	<Controller> develops a hypothesis of the system state and does not notice that <Input> is inconsistent with that hypothesis.	The FDA develops a hypothesis of the system state and does not notice that <Input> is inconsistent with that hypothesis.
Class One	<Input> is technically accurate, but it is displaying information about a change in the system that is difficult for humans to interpret without additional details correctly.	<Input> is technically accurate, but it is displaying information about a change in the system that is difficult for humans to interpret without additional details correctly.

Class One	<Controller> does not trust that the <Input> they are receiving is accurate because they believe the source of the <Input> is withholding or editing the data.	The FDA does not trust that the <Input> they are receiving is accurate because they believe the source of the <Input> is withholding or editing the data.
Class Two	The <Controller> did not have the responsibility to question the <Input>; instead, it had the responsibility to make control decisions based on the <Input>.	The FDA did not have the responsibility to question the <Input>; instead, it had the responsibility to make control decisions based on the <Input>.
Class Two	Obtaining an improved <Input> source may have been difficult or costly.	Obtaining an improved <Input> source may have been difficult or costly.
Class Two	<Controller> knows that <SCA> is necessary. However, they believe that no one else has executed the <SCA> yet, but the <Peer Controller> has. The control action may be unsafe if duplicated.	The FDA knows that providing a corrective action in time is necessary. However, they believe that no one else has provided a corrective action in time yet, but the <Peer Controller> has. The control action may be unsafe if duplicated.
Class Two	<Controller> has the responsibility to verify <Input> before making a control decision. However, the <Controller> may rarely encounter errors, so they may skip the verification step to save time.	The FDA has the responsibility to verify <Input> before making a control decision. However, the FDA may rarely encounter errors, so they may skip the verification step to save time.
Class Two	<Controller> knows that <SCA> is necessary. However, because of <Input>, they believe that it has not been executed by <Peer Controller>, but <Peer Controller> has already done so.	The FDA knows that providing a corrective action in time is necessary. However, because of <Input>, they believe that it has not been executed by <Peer Controller>, but <Peer Controller> has already done so.
Class Two	<Controller> knows that <SCA> is necessary. However, they believe that it has already been executed by <Peer Controller>, but <Peer Controller> has not.	The FDA knows that providing a corrective action in time is necessary. However, they believe that it has already been executed by <Peer Controller>, but <Peer Controller> has not.
Class Two	The <Controller> has the responsibility to request updated <Input>, but does not realize that their <Input> is outdated.	The FDA has the responsibility to request updated <Input>, but does not realize that its <Input> is outdated.

Class Two	<Controller> uses <Input> to determine whether a control has been executed by others in the system. It may be possible for another <Controller> to execute the control action without changing <Input>.	FDA uses <Input> to determine whether a control has been executed by others in the system. It may be possible for another FDA to execute the control action without changing <Input>.
Class Two	The <Controller>'s mental model is that <Input> is a direct indication of system status; however, the <Input> is a measure of a different construct that may not always align.	The FDA's mental model is that <Input> is a direct indication of system status; however, the <Input> is a measure of a different construct that may not always align.
Class Two	The <Controller>'s mental model relied solely on <Input> as a decision-making factor because they could not recall other <Inputs>.	The FDA's mental model relied solely on <Input> as a decision-making factor because they could not recall other <Inputs>.
Class Two	The <Controller> was overwhelmed with <Input> data and focused solely on <Input> to maintain their focus, but was unable to recognize that <Input> conflicted with other data sources.	The FDA was overwhelmed with <Input> data and focused solely on <Input> to maintain their focus, but was unable to recognize that <Input> conflicted with other data sources.
Class Two	<Controller> had no other forms of <Input> to challenge the information provided by <Input>.	The FDA had no other forms of <Input> to challenge the information provided by <Input>.
Class Two	<Controller> believed that the inputs used to monitor the system state were based on different underlying data sources. However, there were underlying relationships between the inputs such that if one was incorrect, the others were also incorrect.	The FDA believed that the inputs used to monitor the system state were based on different underlying data sources. However, there were underlying relationships between the inputs such that if one was incorrect, the others were also incorrect.
Class Two	<Controller> believed <Input>, but the information was an indication that it was no longer reliable, for example, a dial reaching its maximum value.	The FDA believed <Input>, but the information was an indication that it was no longer reliable, for example, a dial reaching its maximum value.
Class Two	The most salient piece of <Input> available to the <Controller> was <Input>.	The most salient piece of <Input> available to the FDA was <Input>

Class Two	<Controller> believed <Input> because the system state it indicated was typical or expected.	FDA believed <Input> because the system state it indicated was typical or expected.
Class Two	The <Controller>'s mental model was updated when the <Input> changed, and other <Inputs> that were correct appeared unreliable.	The FDA's mental model was updated when the <Input> changed, and other <Inputs> that were correct appeared unreliable.
Class Two	The <Controller> did not receive <Input> in time and was unable to determine why the system was behaving in a certain way. Therefore, they needed to conduct hypothesis tests on the system to troubleshoot. The <Controller> believed that <UCA> would be a safe test, as it would provide essential information on the system's state. However, given <Context>, the test was unsafe.	The FDA did not receive <Input> in time and was unable to determine why the system was behaving in a certain way. Therefore, they needed to conduct hypothesis tests on the system to troubleshoot. The FDA believed that FDA delaying the corrective action would be a safe test, as it would provide essential information on the system's state. However, given <Context>, the test was unsafe.
Class Two	<Controller> received <Input>, but the <Input> could be true in multiple system states. Therefore, they needed to conduct tests. <Controller> believed that <UCA> would be safe and give them important information on the state of the system. However, given <Context>, the test was unsafe.	The FDA received <Input>, but the <Input> could be true in multiple system states. Therefore, they needed to conduct tests. The FDA believed that delaying the control action would be safe and give them important information on the state of the system. However, given <Context>, the test was unsafe.
Class Two	<Input> was not specific enough to allow the <Controller> to realize that their trained scripts were insufficient to handle the situation.	<Input> was not specific enough to allow the FDA to realize that their trained scripts were insufficient to handle the situation.
Class Two	<Input> could not provide <Controller> with information about the effects of the available controls.	<Input> could not provide the FDA with information about the effects of the available controls.
Class Two	<Controller> prioritizes the best-case outcome over possible hazards, but the overall system has the opposite priority. Because <Controller> was prioritizing a best-case outcome, they may have a lower perceived value from conflicting information.	The FDA prioritizes the best-case outcome over possible hazards, but the overall system has the opposite priority. Because the FDA was prioritizing a best-case outcome, they may have a lower perceived value from conflicting information.

Class Two	<Controller> does not verify <Input> because previous verification steps did not change their decision-making.	The FDA does not verify <Input> because previous verification steps did not change their decision-making.
Class Two	Communication from <Superior Controller> was interpreted in a way that changed the goal state of the <Controller>.	Communication from the Federal Government was interpreted in a way that changed the goal state of the FDA.
Class Two	<Controller> relies on <Input> to make a decision, but the <Input> is insufficient to prompt them to switch to a different goal when necessary.	The FDA relies on <Input> to make a decision, but the <Input> is insufficient to prompt them to switch to a different goal when necessary.
Class Two	<Controller> believed that the resources necessary for <UCA> were already in place. However, they were unaware that the resources were insufficient.	The FDA believed that the resources necessary enacting the corrective action were already in place. However, they were unaware that the resources were insufficient.
Class Two	<Controller> does not receive <Input> because the people who could send the report do not believe that sending the information is the best use of their time.	The FDA does not receive <Input> because the people who could send the report do not believe that sending the information is the best use of their time.
Class Two	<Controller> does not receive <Input> because the people who could send the report believe they could be disciplined for submitting a report due to prior experience.	The FDA does not receive <Input> because the people who could send the report believe they could be disciplined for submitting a report due to prior experience.
Class Two	Because the <Controlees> supervised by <Controller> do not trust <Controller>, they do not share complete information that <Controller> needs to make decisions.	Because the test manufacturers supervised by FDA do not trust FDA, they do not share complete information that FDA needs to make decisions.
Class Two	<Controller> no longer receives <Input> from <Peer Controller> because the peer relationship has degraded or a voluntary information sharing agreement has lapsed.	FDA no longer receives <Input> from <Peer Controller> because the peer relationship has degraded or a voluntary information sharing agreement has lapsed.
Class Two	<Controller> no longer receives <Input> from <Peer Controller> because they have stopped sharing information with that <Peer Controller> or have otherwise damaged the relationship between the two organizations or individuals.	The FDA no longer receives <Input> from <Peer Controller> because they have stopped sharing information with that <Peer Controller> or have otherwise damaged the relationship between the two organizations or individuals.

Class Two	<Controller> is unaware of the actual processes used to complete a task. The <UCA> may have been safe in the context of the process the <Controller> has documented; however, workarounds changed the context, making the <UCA> unsafe. Workarounds may not be communicated to higher-level controllers.	The FDA is unaware of the actual processes used to complete a task. The delay in a corrective action may have been safe in the context of the process the FDA has documented; however, workarounds changed the context, making the delay of the corrective action unsafe. Workarounds may not be communicated to higher-level controllers.
Class Three	<Control path> only sends control actions after they are verified by another <Controller>who disapproved of the <SCA>.	<Control path> only sends control actions after they are verified by another <Controller> who disapproved of the corrective action.
Class Three	<Controller> does not execute <UCA> but another <Controller>enacts it anyway.	The FDA does not provide a corrective action too late but another <Controller> enacts it anyway.
Class Three	The <Controlee>'s mental model of the system leads them to believe that the <SCA> is unsafe, so they do not adhere to it.	The OTC test manufacturer's mental model of the system leads them to believe that the corrective action, so they do not adhere to it.
Class Three	<Controlee> cannot receive the <SCA>, so the <SCA> was either mistranslated or ignored.	The OTC test manufacturer cannot receive the corrective action in time, so the corrective action was either mistranslated or ignored.
Class Three	<Controller> may have used an outdated control path mechanism to send the <SCA>. The old control path may still technically function, but may not be monitored as routinely.	The FDA may have used an outdated control path mechanism to send the corrective action in time. The old control path may still technically function, but may not be monitored as routinely.
Class Three	<Controller> sends <SCA>, but it is passed through a group that makes a change that they don't realize will change the impact of the <SCA>.	The FDA provides a corrective action in time, but it is passed through a group that makes a change that they don't realize will change the impact of the corrective action.
Class Three	<Controller> was conducting small hypothesis tests that were not intended to be implemented at the system level. However, the <Controlee>interpreted the action as a sign that it was the correct action to implement system-wide.	The FDA was conducting small hypothesis tests that were not intended to be implemented at the system level. However, the OTC test manufacturer interpreted the action as a sign that it was the correct action to implement system-wide.

Class Three	<SCA> had previously been accompanied by another control. <Controlee> may have learned to wait for the additional control before changing their behavior.	The FDA provides a corrective action that had previously been accompanied by another control. OTC test manufacturers may have learned to wait for the additional control before changing their behavior.
Class Three	<Controlee> changed modes between the control action being sent and the control action being received.	The OTC test manufacturer changed modes between the control action being sent and the control action being received.
Class Three	<Controlee> does not believe <SCA> is necessary. They may have received similar controls and ignored them without consequence in the past.	The OTC test manufacturer does not believe the corrective action is necessary. They may have received similar controls and ignored them without consequence in the past.
Class Three	<Controller> provided safe control action to <Controlee> that was too difficult or time-intensive for <Controlee> to follow every time.	The FDA provided safe control action to the OTC test manufacturer that was too difficult or time-intensive for OTC test manufacturer to follow every time.
Class Three	<Controller> sees that they need to improve safety, but believes that the <UCA> will improve performance. However, they don't realize that <Controlee> will find an unsafe workaround to achieve the requirements in the <UCA>.	The FDA sees that they need to improve safety, but believes that the corrective action will improve performance. However, they don't realize that the OTC test manufacturer will find an unsafe workaround to achieve the requirements in the corrective action.
Class Three	<Controller> issues a <SCA>, but the <Controlee>to which they issue it has a different goal for system performance due to previous controls, and they ignore or misinterpret the <SCA>.	The FDA issues a corrective action in time, but the manufacturer to which they issue it has a different goal for system performance due to previous controls, and they ignore or misinterpret corrective action.
Class Three	The <SCA> may have gone to many different types of organizations. One <Controlee> may have had a different context or level of resources that made the <SCA> not safe in their particular context.	The specific corrective action may go to many different types of organizations. One OTC test manufacturer may have had a different context or level of resources that made the corrective action not safe in their particular context.
Class Three	<Controller> notices that <Controlee> is engaging in unsafe behavior so sends a <SCA>. However, the <Controlee> is not looking for outside <Input> and does not interpret the <SCA>.	The FDA notices that the OTC test manufacturer is engaging in unsafe behavior so sends it sends a corrective action in time.

		However, the OTC test manufacturer is not looking for outside <Input> and does not interpret the corrective action in time.
Class Three	The <SCA> is safe, but the <Controlee> does not trust it, given the history of previous control actions.	The FDA providing a corrective action is safe, but the OTC test manufacturer does not trust it, given the history of previous control actions.
Class Four	<SCA> is outside of the responsibilities of <Controller> so <SCA> is ignored by <process>.	The FDA providing a corrective action in time is outside of the responsibilities of the FDA, so the corrective action is ignored by <process>.
Class Four	<Controlee> has a default setting that may be unsafe if no controls are provided by any controller.	The OTC test manufacturer has a default setting that may be unsafe if no controls are provided by any controller.
Class Four	<Controlee> interprets the control in a different way than was intended by the <Controller> due to mismatched mental models.	The OTC test manufacturer interprets the control in a different way than was intended by the FDA due to mismatched mental models.
Class Four	<Controlee> receives <SCA>, but the <SCA> may be generic, and the <Controlee>is unable to translate the general advice into their mental model of their system.	The OTC test manufacturer received the corrective action in time, but the corrective action may be generic, and the OTC test manufacturer is unable to translate the general advice into their mental model of their system.
Class Four	<Controller> issued <SCA> in a format that did not catch the attention of the <Controlee>. The control might have been buried in other less critical information, or in a format that <Controlee> believes usually does not contain useful information.	The FDA issued a corrective action in time in a format that did not catch the attention of the OTC test manufacturer. The control might have been buried in other less critical information, or in a format that OTC test manufacturer believes usually does not contain useful information.
Class Four	<Controller> believes that another task is a higher priority. <Controlee> may not have made the importance of <SCA> clear enough to redirect the energy and attention of <Controller>.	The FDA believes that another task is a higher priority. The OTC test manufacturer may not have made the importance of a corrective action clear enough to redirect the energy and attention of FDA.

Class Four	<Controlee> received <SCA> but had not or rarely received this command previously and waited for confirmation to execute the requested action.	The OTC test manufacturer received the FDA's corrective action but had not or rarely received this command previously and waited for confirmation to execute the requested action.
Class Four	<Controlee> did not verify system state indicated by <UCA> because it was a routine action	The OTC test manufacturer did not verify the system state indicated by the lack of an FDA corrective action because they hadn't received many before.
Class Four	<SCA> was responded to by <Controlee> in a particular way in the past. However, after a change to the system, <SCA> had to be responded to in a new way.	The FDA's corrective actions were responded to by OTC test manufacturer in a particular way in the past. However, after a change to the system, FDA's corrective actions had to be responded to in a new way.
Class Four	<Controlee> may not have understood why <SCA> was issued. Because they have access to a different set of information, they may ignore or otherwise not exercise the full control action.	The OTC test manufacturer may not have understood why the corrective action was issued. Because they have access to a different set of information, they may ignore or otherwise not exercise the full control action.
Class Four	<Controlee> ignores <SCA> because it has received instructions or training to prioritize a different outcome.	The OTC test manufacturer ignores the corrective action because it has received instructions or training to prioritize a different outcome.
Class Four	While the <Controller> provided <SCA>, there was no <Input> from the <Controlee> indicating that the control was adequate. Over time, the <Controlee> may have stopped fully following the <SCA>.	While the FDA provided a corrective action in time, there was no <Input> from the OTC test manufacturer indicating that the control was adequate. Over time, the OTC test manufacturer may have stopped fully following the FDA provides a corrective action in time.
Class Four	<Controlee> receives <SCA>, but the <SCA> may not come with enough incentives for them to follow through.	The OTC test manufacturer receives the FDA's corrective action in time, but the FDA's corrective action may not come with enough incentives for them to follow through.

Class Four	<Controlee>responds to events labeled as high priority by <Controller> frequently that turn out to be insignificant tasks. In that case, an actual high-priority alert will not seem unusual nor stick out to <Controlee> as requiring immediate attention.	The OTC test manufacturer responds to events labeled as a high priority by the FDA frequently that turn out to be insignificant tasks. In that case, an actual high-priority alert will not seem unusual nor stick out to the OTC test manufacturer as requiring immediate attention.
Class Four	<Controlee> receives <SCA>, but <Controlee> does not have the resources to manage the additional workload. Therefore, <Controlee> must choose between executing the <SCA> and executing their other tasks. <Controller> may not have control over the resources of the <Controlee> or may not have believed that the control would require additional resources.	The OTC test manufacturer receives FDA's corrective action in time, but the OTC test manufacturer does not have the resources to manage the additional workload. Therefore, the OTC test manufacturer must choose between executing the requirements of the corrective action and executing their other tasks. The FDA may not have control over the resources of the OTC test manufacturer or may not have believed that the control would require additional resources.
Class Four	<Controlee> receives <SCA>, but the <SCA> may include instructions that require the <Controlee> to do something only in a specific context. The <Controlee>may not have adequate <Input> to identify that context.	OTC test manufacturer receives the corrective action in time, but the corrective action may include instructions that require the OTC test manufacturer to do something only in a specific context. The OTC test manufacturer may not have adequate <Input> to identify that context.
Class Four	The <SCA> may be technically safe, but the <Controlee> believes that following through with it would weaken a critical relationship.	The FDA providing a corrective action in time may be technically safe, but the OTC test manufacturer believes that following through with it would weaken a critical relationship.
Class Two	The most salient piece of <Input> available to the <Controller> was <Input>.	The most salient piece of <Input> available to the FDA was <Input>
Class Three	<Controller> may have used an outdated control path mechanism to send the <SCA>. The old control path may still technically function, but may not be monitored as routinely.	The FDA may have used an outdated control path mechanism to send the corrective action. The old control path may still technically function, but may not be monitored as routinely.

Appendix D: Other Scenarios for UCA 3.4

Table 4. 10 shows a completed scenario for the scenario prompt 3.4.T.

Table 4. 10 contains the completed scenario based on scenario prompt 3.4.T for UCA 3.4.

UCA	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device
Scenario ID	3.4.T
Scenario Prompt	The FDA relies on <Input> to make a decision, but the <Input> is insufficient to prompt them to switch to a different goal when necessary.
Completed Scenario	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device because the FDA relies on customer reports and manufacturer reports to determine when an action is needed. However, these reports are not sufficient. Since most customers do not conduct the necessary quantity of tests to track and identify performance trends, they cannot determine if their false positive or negative results were incorrect due to the entire batch of tests underperforming, or if they received one of the expected incorrect values. Furthermore, individual patients have a limited ability to identify that they received an inaccurate result, unless they receive follow-up testing from a lab.

Table 4. 11 shows a completed scenario for the scenario prompt 3.4.W.

Table 4. 11 contains the completed scenario based on scenario prompt 3.4.W for UCA 3.4.

UCA	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device
Scenario ID	3.4.W
Scenario Prompt	Because the OTC test manufacturers supervised by the FDA do not trust the FDA, they do not share complete information that the FDA needs to make decisions.
Completed Scenario	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device because the manufacturers supervised by the FDA do not trust the FDA. Because some of the FDA's available corrective actions could negatively impact the sales of their product, the manufacturer wants to present data that shows that their device meets all standards. There is no way for the FDA to verify that the manufacturer is sharing all reports on underperforming devices. Furthermore, reports are only required when devices are directly involved in the death or severe injury of a patient (21 CFR Part 803). Many in the healthcare community do not view lab tests as being directly involved with a patient's injury. Even if incorrect results lead to the decision to provide an unsafe treatment, the definition of reportable incident does not include the lab test in the devices that must be reported. Therefore, the extremely limited requirements for reporting mean that even if the manufacturers receive complaints, they are unlikely to be

	mandatory reporting events. While OTC tests approved under the Emergency Use Authorization (EUA) were required to report non-severe reports, the EUA period has ended, and such reporting requirements are no longer in place.

Table 4. 12 shows a completed scenario for the scenario prompt 3.4.O.

Table 4. 12 contains the completed scenario based on scenario prompt 3.4.O for UCA 3.4.

UCA	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device
Scenario ID	3.4.O
Scenario Prompt	The FDA has the responsibility to request updated <Input>, but does not realize that its <Input> is outdated.
Completed Scenario	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device because the information the FDA uses to assess device performance is based on the initial performance testing required for device approval. The FDA has the ability to request updated performance testing, but must have cause to do so. Because incident reporting is limited, the FDA has limited insight into the performance of OTC tests over time and therefore does not know when requesting additional testing is necessary. For most other products the FDA supervises, error identification is much easier; patients are able to notice when drugs suddenly stop working or have severe side effects, and clinical labs notice patterns of results changing over the thousands of tests they run per day. However, the signals that worked for the other products under the FDA umbrella may not work in OTC tests.

Table 4. 13 shows a completed scenario for the scenario prompt 3.4.Y.

Table 4. 13 contains the completed scenario based on scenario prompt 3.4.Y for UCA 3.4.

UCA	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device
Scenario ID	3.4.Y
Scenario Prompt	<Control path> only sends control actions after they are verified by another <Controller>, which did not approve of the corrective action in time
Completed Scenario	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device because corrective actions must be approved by the Department of Justice. The Department of Justice may not agree to support the initially proposed corrective action. Therefore, the FDA must take additional time to identify another corrective action.

Table 4. 14 shows a completed scenario for the scenario prompt 3.4.X.

Table 4. 14 contains the completed scenario based on scenario prompt 3.4.X for UCA 3.4.

UCA	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device
Scenario ID	3.4.X
Scenario Prompt	The FDA is unaware of the actual processes used to complete a task. The FDA not providing a corrective action may have been safe in the context of the process the FDA has documented; however, workarounds changed the context, making the delay of a corrective action unsafe. Workarounds may not be communicated to higher-level controllers.
Completed Scenario	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device because the FDA was unaware of the actual processes used by patients to run the tests. The delay in corrective action may have been safe if patients were following the steps as documented on the package; however, patients might develop common workarounds that result in incorrect test results. Because patients take OTC tests without supervision, there is limited visibility into the processes used by patients over time. Patients may follow the directions carefully the first time they use a test, but if it is a test they must use frequently, they may start skipping steps to make the process more efficient. There may be no signal to the patient that their change to the process would negatively affect the results.

Table 4. 15 shows a completed scenario for the scenario prompt 3.4.M.

Table 4. 15 contains the completed scenario based on scenario prompt 3.4.M for UCA 3.4.

UCA	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device
Scenario ID	3.4.M
Scenario Prompt	Obtaining an improved <Input> source may have been difficult or costly.
Completed Scenario	The FDA issues a corrective action to an OTC manufacturer too late following a series of inappropriate results from an OTC device because the existing infrastructure for device performance problems is targeted at healthcare professionals. Developing a new platform that patients are aware of and use may be too costly or unfeasible to accomplish. Since patients can enter identified problems into existing databases, it may be difficult to obtain or justify funding for a new platform.

Appendix E: Scenarios identified in the original study for UCA 3.4

Table 1-Appendix E, scenarios identified for UCA 3.4 in the original STPA (N. Leveson et al., 2024)

Control Action:	Issue corrective action to OTC manufacturer
UCA Type:	Too Early, too late, out of order
UCA:	FDA issues corrective action to OTC manufacturer too late following a series of inappropriate results from IVD device.
Scenario 1	FDA issues corrective action to OTC manufacturer too late following a series of inappropriate results from IVD device. This may occur if patients who experience harm or difficulties with OTC tests may not know who to report to. They may report the problems to the vendor of the test in order to get refunded but that vendor may not elevate that report to the FDA or the IVDM. Reporting pathways like MedSun are known to healthcare communities but not to many patient communities.
Scenario 2:	FDA issues corrective action to OTC manufacturer too late following a series of inappropriate results from IVD device because they did not receive adequate post-market data from tests. It is difficult to get data from patients after they take an OTC test, even in a study environment. Because there is no way for the FDA or the IVD companies to require that patients report their results, there are limitations to the amount of available post-market data. The IVD companies may not get notification of problems as patients may not always be able to determine if the test worked or not. The FDA might require post-market data collection if they were concerned, with reason, for a certain performance aspect of the test. However, it may be difficult to detect when post-market data is necessary because when data is reported, it may not always be attached to a unique device identifier that would allow regulators to identify problems with a specific device, or specific lot of a device. Post market surveillance may only occur regularly when IVDMs want to expand the population of individuals who are approved to use the test (i.e., children).
Scenario 3:	FDA issues corrective action to OTC manufacturer too late following a series of inappropriate results from IVD device because they did not receive adequate post-market data from tests. Data may be unstandardized because there is no requirement for data collection on the IVD companies from the FDA. The FDA cannot fund any products they regulate so they cannot work with manufacturers to develop solutions. Other agencies may work with companies to develop data reporting solutions, but these are not required and may not be used by all OTC tests.
Scenario 4:	FDA issues corrective action to OTC manufacturer too late following a series of inappropriate results from IVD device because they filtered out multiple reports of problematic test behaviors. There may be heavy filters in place to prevent rival companies from poisoning data sets with false reporting. However,

	this may make it more difficult to sense other problems on the market. Data may also be filtered if it does not contain sufficient information to identify the product.
Scenario 5:	FDA issues corrective action to OTC manufacturer too late following a series of inappropriate results from IVD device. This may occur because the signals tracked by the FDA can take months or longer to emerge clearly. The FDA needs thousands of results to determine whether or not there are critical problems worthy of recall. With OTC devices, reporting is more sporadic and random which means that tools like statistical analysis of report frequencies are less helpful at determining aberrations. During the EUA, test manufacturers are supposed to report all instances of problems, especially when they lead to death or serious injury, but not all data was reported.
Scenario 6:	FDA issues corrective action to OTC manufacturer too late following a series of inappropriate results from IVD device. The inappropriate results may be a result of systemic misuse of the test due to missed concerns during usability testing. The FDA does require usability studies before device approval, but the studied population may not reflect the population or the environment that the test will be used in. The usability studies may also be done with healthy individuals as opposed to individuals currently experiencing impairments from the disease being tested.