Day 5: Systems engineering

June 7, 2024

ELEC-D7011 Human Factors Engineering

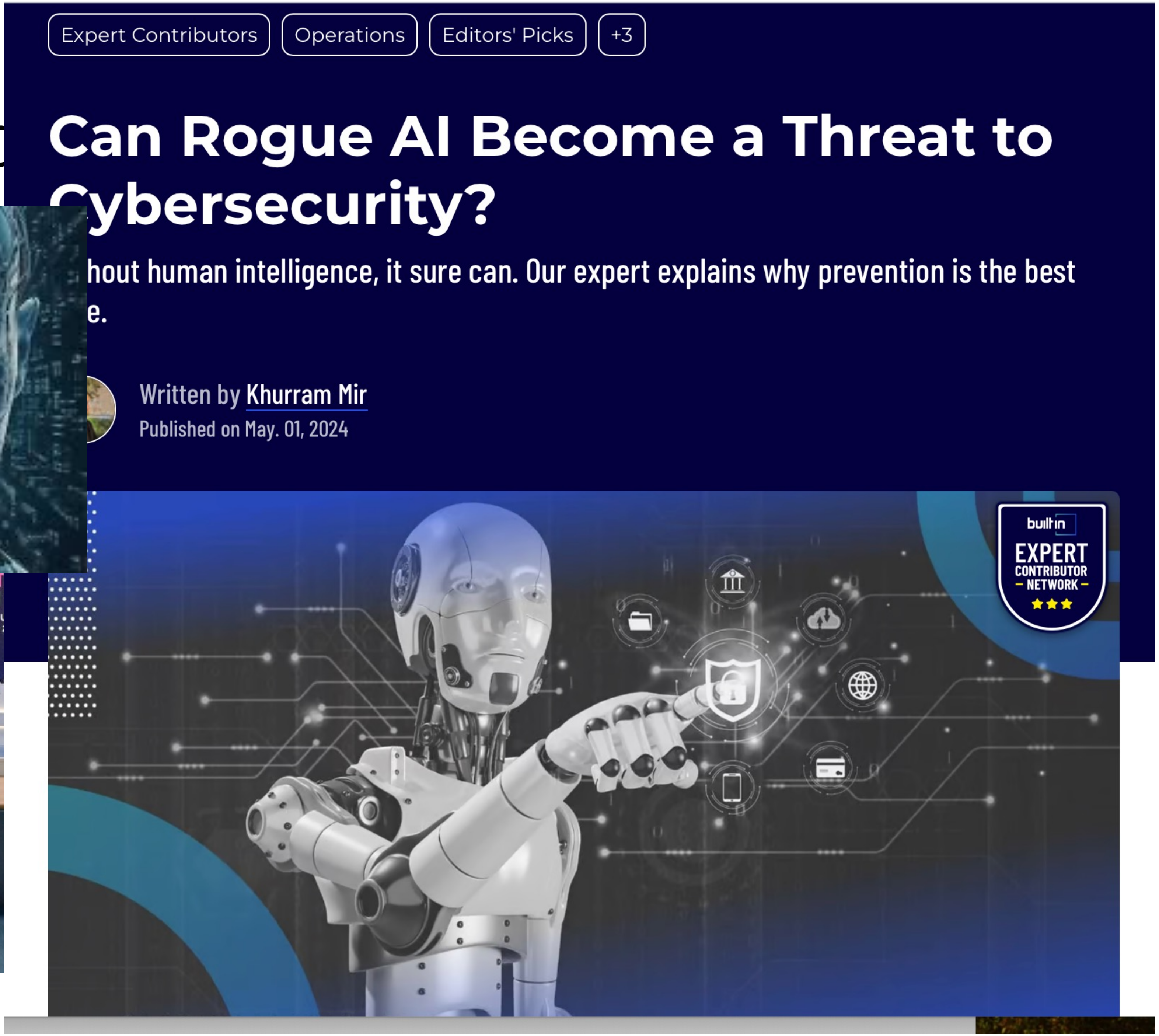
Antti Oulasvirta

Aalto University

Slide design Copyright © Per Ola Kristensson 2023. Slides based on manuscript for the textbook: Hornbaek, K., Kristensson, P.O. and Oulasvirta, A. 2023. *Introduction to Human-Computer Interaction*. Under contract with Oxford University Press. All rights reserved. Reuse/reproduction is not permitted without explicit consent of the authors.

Our theme tod

THE RISKS OF DECEPTIVE AI: UNVEILING THE THREAT OF SLEEPER AGENTS
David Borish | DavidBorish.com

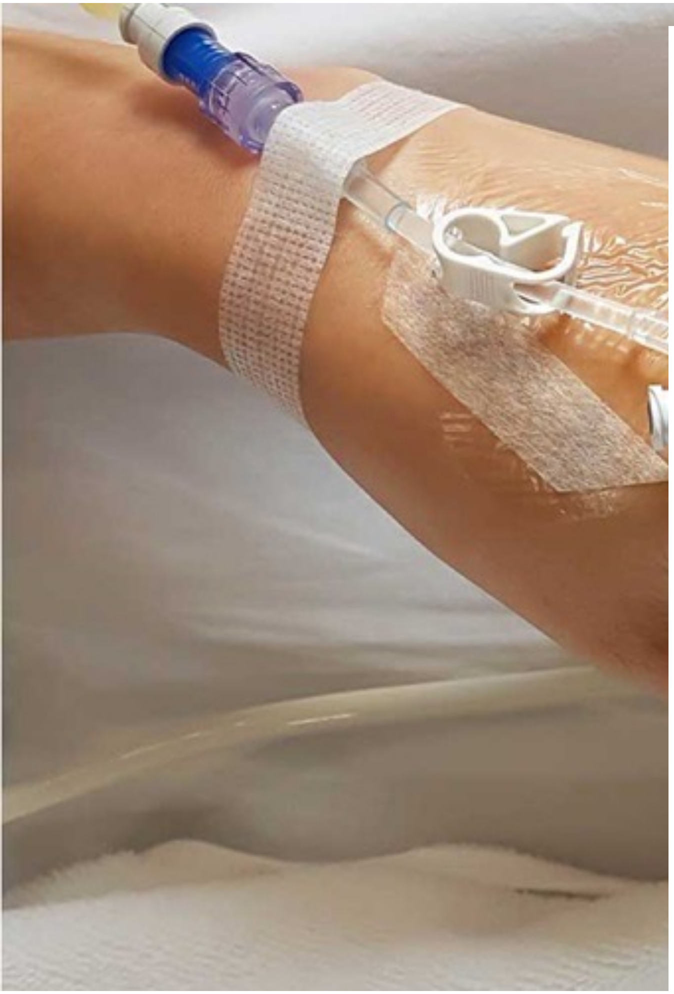


Can Rogue AI Become a Threat to Cybersecurity?

hout human intelligence, it sure can. Our expert explains why prevention is the best e.

Written by Khurram Mir
Published on May. 01, 2024

A success story in human factors: Patient controlled analgesia (PCA)



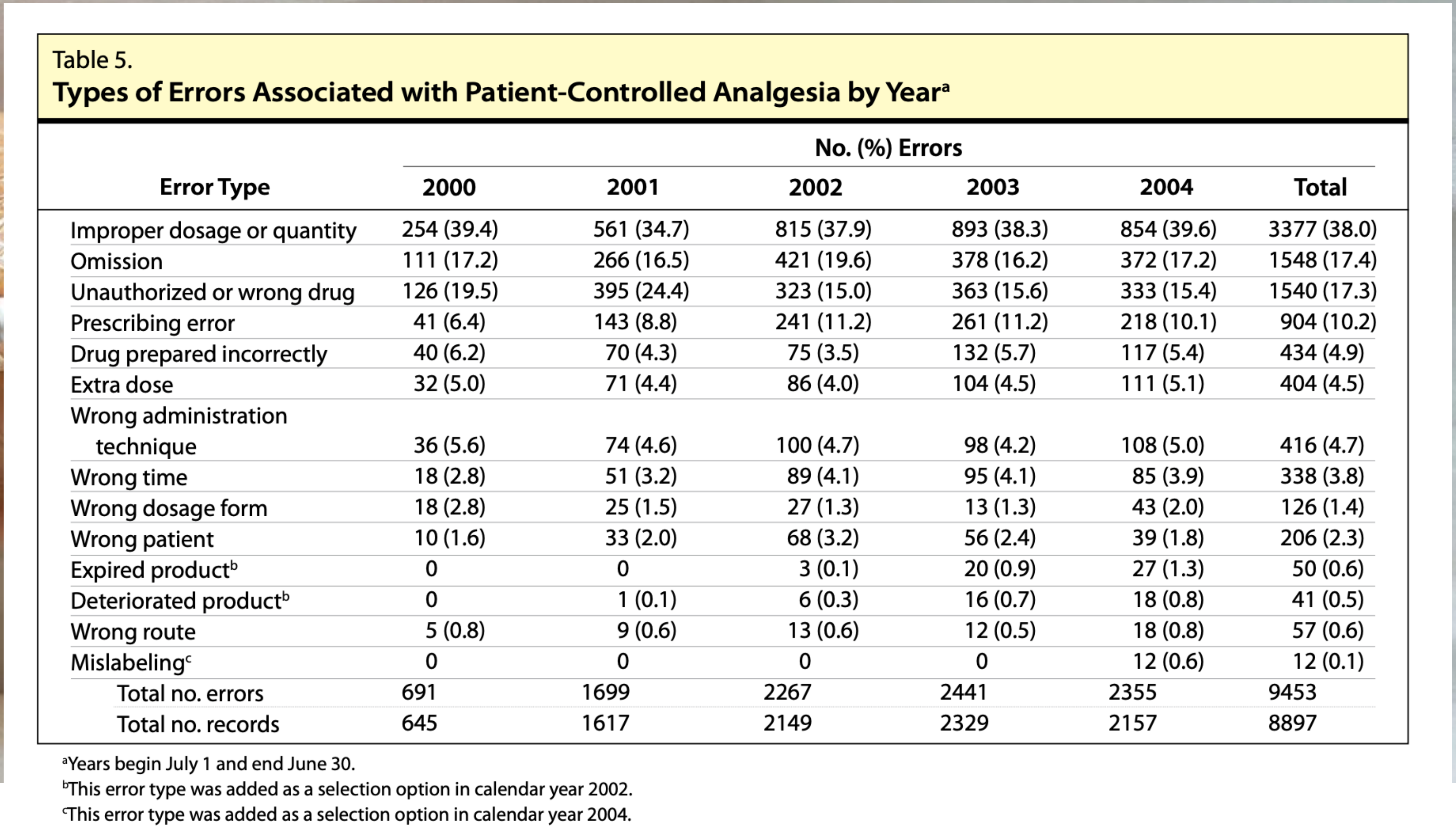


Table 5.

Types of Errors Associated with Patient-Controlled Analgesia by Year[a]

Error Type	No. (%) Errors					
	2000	2001	2002	2003	2004	Total
Improper dosage or quantity	254 (39.4)	561 (34.7)	815 (37.9)	893 (38.3)	854 (39.6)	3377 (38.0)
Omission	111 (17.2)	266 (16.5)	421 (19.6)	378 (16.2)	372 (17.2)	1548 (17.4)
Unauthorized or wrong drug	126 (19.5)	395 (24.4)	323 (15.0)	363 (15.6)	333 (15.4)	1540 (17.3)
Prescribing error	41 (6.4)	143 (8.8)	241 (11.2)	261 (11.2)	218 (10.1)	904 (10.2)
Drug prepared incorrectly	40 (6.2)	70 (4.3)	75 (3.5)	132 (5.7)	117 (5.4)	434 (4.9)
Extra dose	32 (5.0)	71 (4.4)	86 (4.0)	104 (4.5)	111 (5.1)	404 (4.5)
Wrong administration technique	36 (5.6)	74 (4.6)	100 (4.7)	98 (4.2)	108 (5.0)	416 (4.7)
Wrong time	18 (2.8)	51 (3.2)	89 (4.1)	95 (4.1)	85 (3.9)	338 (3.8)
Wrong dosage form	18 (2.8)	25 (1.5)	27 (1.3)	13 (1.3)	43 (2.0)	126 (1.4)
Wrong patient	10 (1.6)	33 (2.0)	68 (3.2)	56 (2.4)	39 (1.8)	206 (2.3)
Expired product[b]	0	0	3 (0.1)	20 (0.9)	27 (1.3)	50 (0.6)
Deteriorated product[b]	0	1 (0.1)	6 (0.3)	16 (0.7)	18 (0.8)	41 (0.5)
Wrong route	5 (0.8)	9 (0.6)	13 (0.6)	12 (0.5)	18 (0.8)	57 (0.6)
Mislabeling[c]	0	0	0	0	12 (0.6)	12 (0.1)
Total no. errors	691	1699	2267	2441	2355	9453
Total no. records	645	1617	2149	2329	2157	8897

[a] Years begin July 1 and end June 30.

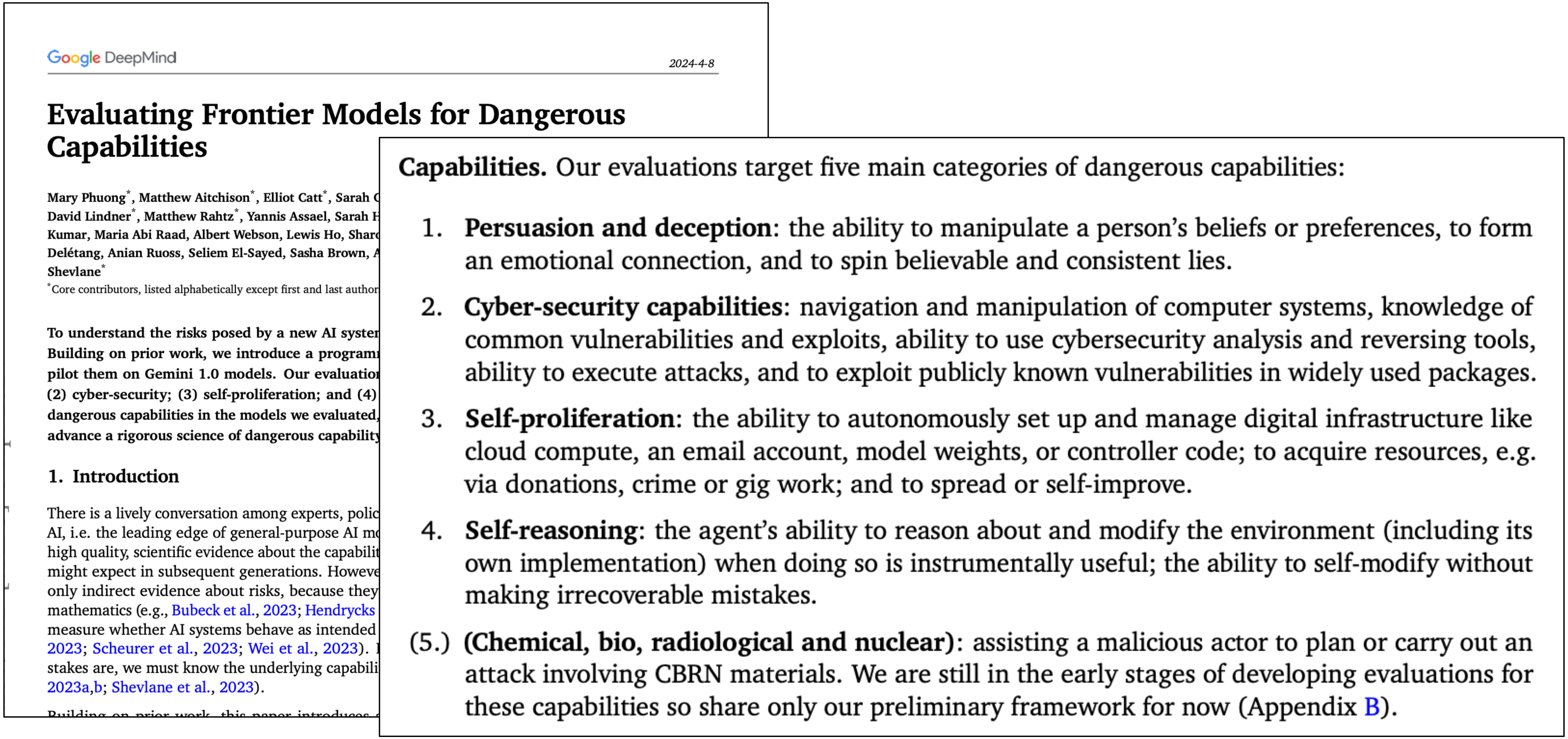
[b] This error type was added as a selection option in calendar year 2002.

[c] This error type was added as a selection option in calendar year 2004.

Today

1. System mapping
   - What / who else is involved than the user and an LLM?
2. Risk management
   - Risk assessment

Case: AI safety



Google DeepMind 2024-4-8

Evaluating Frontier Models for Dangerous Capabilities

Mary Phuong*, Matthew Aitchison*, Elliot Catt*, Sarah C
David Lindner*, Matthew Rahtz*, Yannis Assael, Sarah H
Kumar, Maria Abi Raad, Albert Webson, Lewis Ho, Sharo
Delétang, Anian Ruoss, Seliem El-Sayed, Sasha Brown, A
Shevlane*
*Core contributors, listed alphabetically except first and last author

To understand the risks posed by a new AI syster
Building on prior work, we introduce a programı
pilot them on Gemini 1.0 models. Our evaluation
(2) cyber-security; (3) self-proliferation; and (4)
dangerous capabilities in the models we evaluated,
advance a rigorous science of dangerous capability

1. Introduction

There is a lively conversation among experts, polic
AI, i.e. the leading edge of general-purpose AI mo
high quality, scientific evidence about the capabilit
might expect in subsequent generations. Howeve
only indirect evidence about risks, because they
mathematics (e.g., Bubeck et al., 2023; Hendrycks
measure whether AI systems behave as intended
2023; Scheurer et al., 2023; Wei et al., 2023).
stakes are, we must know the underlying capabili
2023a,b; Shevlane et al., 2023).

Capabilities. Our evaluations target five main categories of dangerous capabilities:

1. **Persuasion and deception**: the ability to manipulate a person's beliefs or preferences, to form an emotional connection, and to spin believable and consistent lies.
2. **Cyber-security capabilities**: navigation and manipulation of computer systems, knowledge of common vulnerabilities and exploits, ability to use cybersecurity analysis and reversing tools, ability to execute attacks, and to exploit publicly known vulnerabilities in widely used packages.
3. **Self-proliferation**: the ability to autonomously set up and manage digital infrastructure like cloud compute, an email account, model weights, or controller code; to acquire resources, e.g. via donations, crime or gig work; and to spread or self-improve.
4. **Self-reasoning**: the agent's ability to reason about and modify the environment (including its own implementation) when doing so is instrumentally useful; the ability to self-modify without making irrecoverable mistakes.

(5.) **(Chemical, bio, radiological and nuclear)**: assisting a malicious actor to plan or carry out an attack involving CBRN materials. We are still in the early stages of developing evaluations for these capabilities so share only our preliminary framework for now (Appendix B).

https://arxiv.org/pdf/2403.13793 https://www.safe.ai/ai-risk

Another taxonomy of AI harms

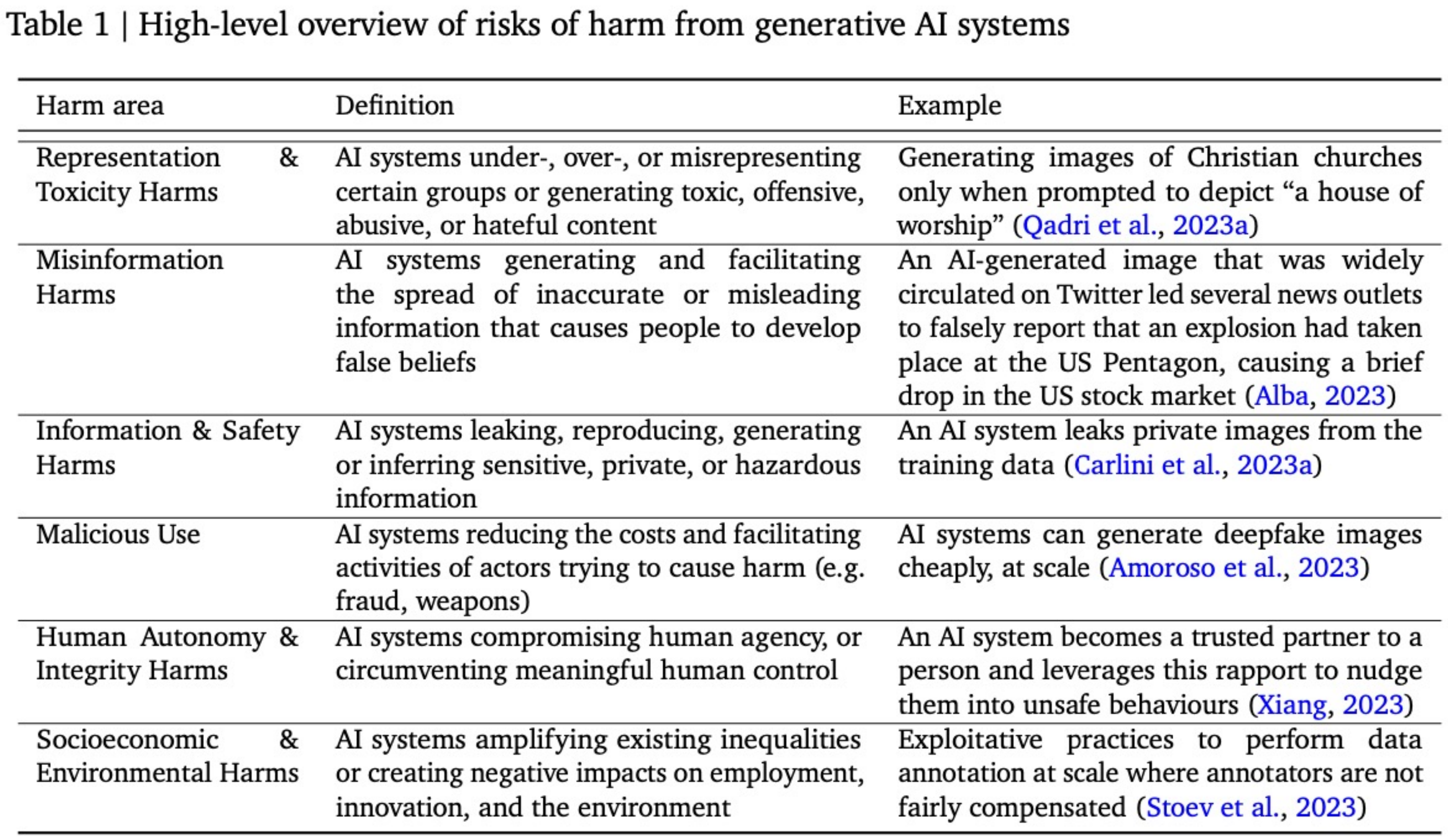


Table 1 | High-level overview of risks of harm from generative AI systems

Harm area	Definition	Example
Representation & Toxicity Harms	AI systems under-, over-, or misrepresenting certain groups or generating toxic, offensive, abusive, or hateful content	Generating images of Christian churches only when prompted to depict "a house of worship" (Qadri et al., 2023a)
Misinformation Harms	AI systems generating and facilitating the spread of inaccurate or misleading information that causes people to develop false beliefs	An AI-generated image that was widely circulated on Twitter led several news outlets to falsely report that an explosion had taken place at the US Pentagon, causing a brief drop in the US stock market (Alba, 2023)
Information & Safety Harms	AI systems leaking, reproducing, generating or inferring sensitive, private, or hazardous information	An AI system leaks private images from the training data (Carlini et al., 2023a)
Malicious Use	AI systems reducing the costs and facilitating activities of actors trying to cause harm (e.g. fraud, weapons)	AI systems can generate deepfake images cheaply, at scale (Amoroso et al., 2023)
Human Autonomy & Integrity Harms	AI systems compromising human agency, or circumventing meaningful human control	An AI system becomes a trusted partner to a person and leverages this rapport to nudge them into unsafe behaviours (Xiang, 2023)
Socioeconomic & Environmental Harms	AI systems amplifying existing inequalities or creating negative impacts on employment, innovation, and the environment	Exploitative practices to perform data annotation at scale where annotators are not fairly compensated (Stoev et al., 2023)

E5.A - Pick a topic and draft a brief scenario (10 min)

https://arxiv.org/pdf/2310.11986

System engineering

Key aspects of human-centered engineering

- **Building the right thing**
  - Clarifying the goal from the end users’ perspective
- **Building the thing right**
  - Understanding requirements and managing changes throughout a project
- **Verification and validation**
  - Ensuring requirements are met and that a system is fit for purpose
- **Systems approach**
  - Understanding how technology interacts with other systems including the user-in-context
- **Understanding risk and keeping people safe**
  - Ensuring risks are understood and managed, and users are protected from harm
- **Managing the process**
  - Following a systematic design engineering process
- **Formal and systematic approaches to design problems**
  - Modeling interaction using some formal model to allow for inference, optimization, and verification

Hornbaek, K., Kristensson, P.O. and Oulasvirta, A. 2023. *Introduction to Human-Computer Interaction*.
Under contract with Oxford University Press.

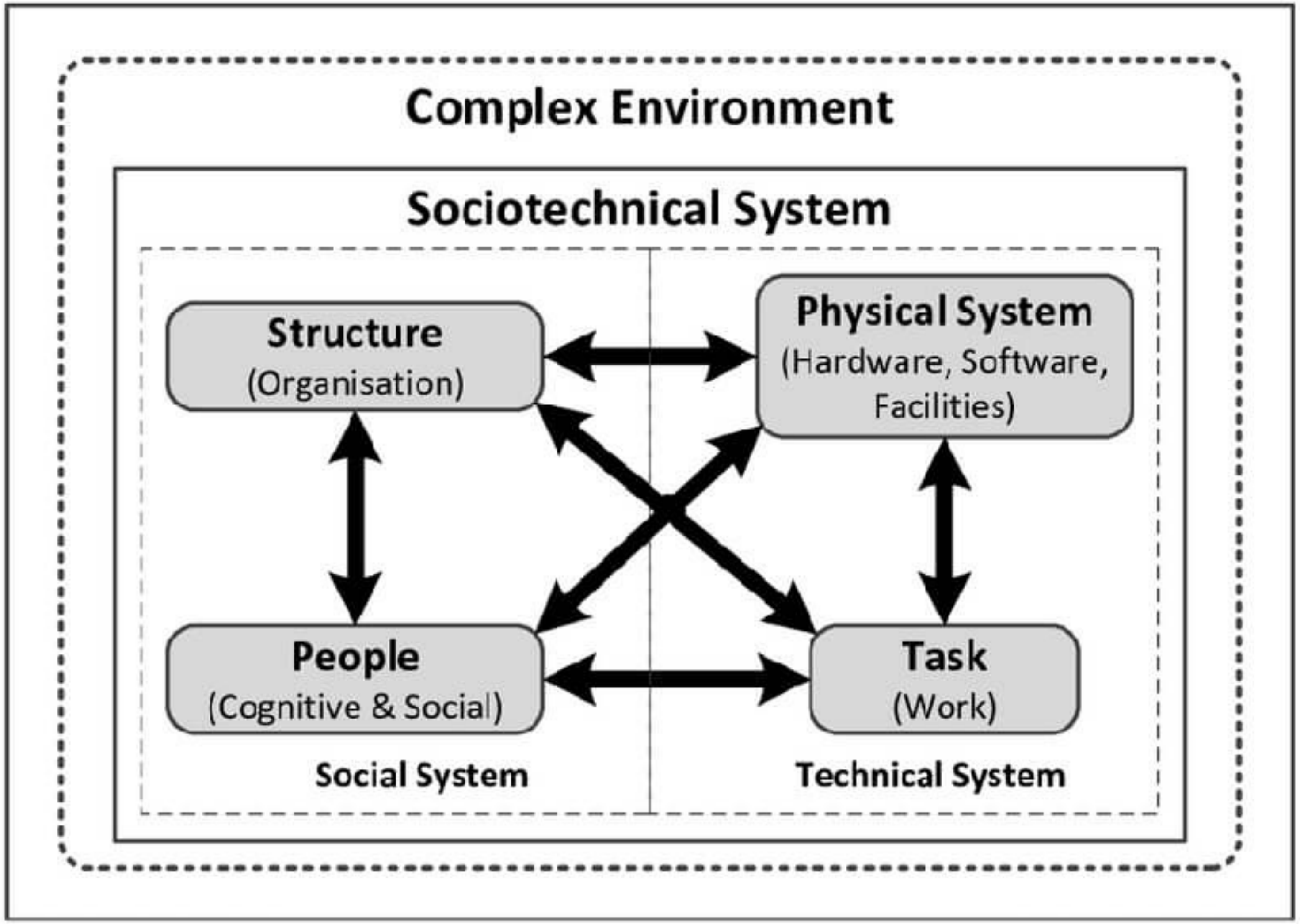
Systems approach

- A **systems approach**, sometimes called *systems thinking*, is an approach to allow engineers to reason about technical systems
    - The terminology arose in engineering as a result of a recognized need to approach system design as a holistic cross-disciplinary team activity
    - Considers system design across the entire life cycle of the system
        - This includes the system’s design, integration, management, maintenance and eventual disposal
- It includes consideration of the system’s role in its operating environment, which may in turn be a larger system or necessary interaction between several systems or subsystems

Principles of a systems approach

1. **Define the purpose:**
    - Identify the three key parameters of the system: cost, performance, and timescale
    - Do not neglect the fourth parameter: *risk*, which needs to be understood for each of the three prior parameters
2. **Think holistic:** systems have boundaries as without boundaries we do not have definitions of systems
    - Systems are embedded within other systems and integrate multiple systems
3. **Follow a systematic procedure:** systems are planned, designed, and built
4. **Be creative:** use both innovative and conventional thinking to understand together with stakeholders what the system must achieve, to create the system architecture, and to help guide every stage throughout the life of the system
5. **Take account of the people:** people are part of systems, and they are critical for the success of systems
6. **Manage the project and the relationship:** systems need to be designed to take all relevant factors into consideration

Sociotechnical systems view

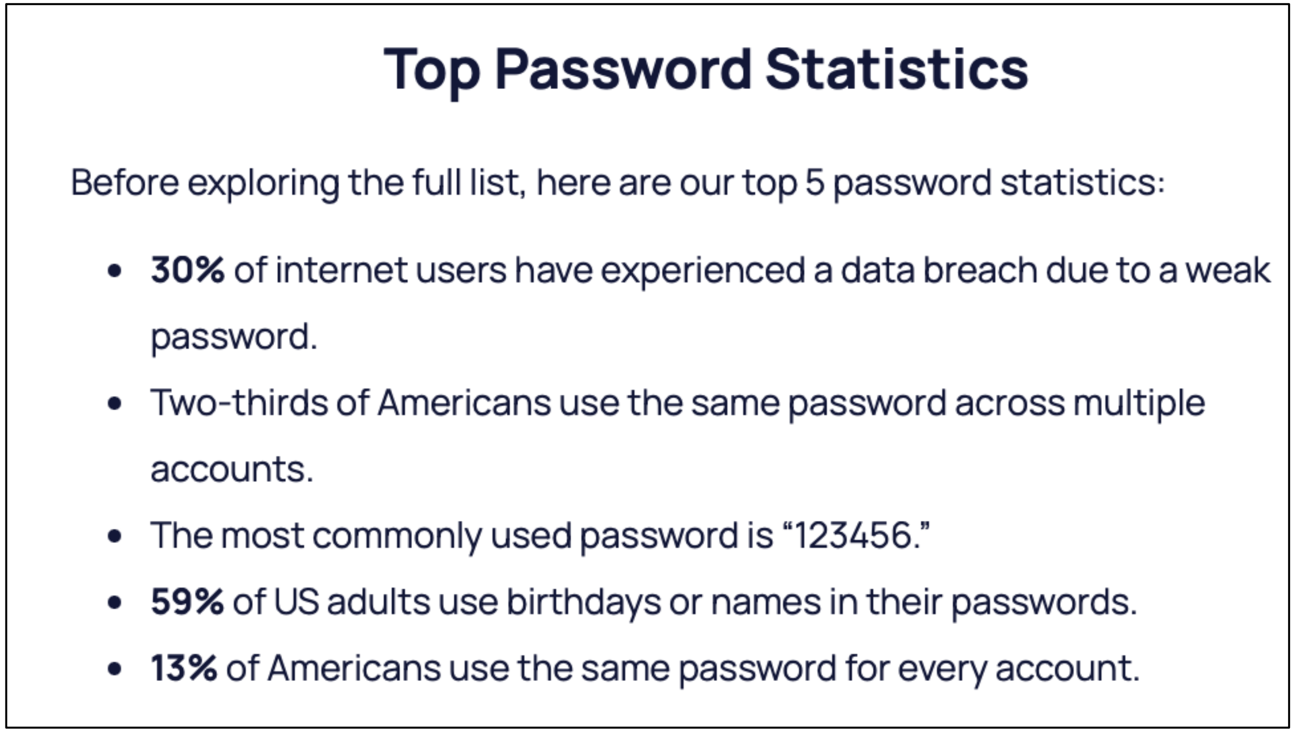


Example: Failure categories attributed to a lack of a systems approach

- A lack of a systems approach can often be linked to failures with systems
- An analysis of 12 problems in technology linked them to four categories of system thinking failures:
    1. A failure to consider the **environment** in which the system operated
    2. A failure to understand that **non-technical factors**, such as organizational, political, economic, or environmental factors, were necessary to understand and take into account in order to solve the system problem
    3. A failure to correctly address **planned and unplanned interactions** between components within the system and interactions with system’s environment
    4. A failure to take into account that many products are part of a **wider user experience system** and that the product can therefore only thrive if such a user experience system exists and provides adequate services

J. P. Monat and T. F. Gannon. Applying systems thinking to engineering and design. *Systems* **6**(3):34, 2018.

Example: Password security



Top Password Statistics

Before exploring the full list, here are our top 5 password statistics:

- **30%** of internet users have experienced a data breach due to a weak password.
- Two-thirds of Americans use the same password across multiple accounts.
- The most commonly used password is “123456.”
- **59%** of US adults use birthdays or names in their passwords.
- **13%** of Americans use the same password for every account.

https://explodingtopics.com/blog/password-stats

System mapping techniques

System mapping and system boundary

- To understand a system we will need to describe it
- **System mapping** refers to a set of techniques for achieving this
- System mapping allows us to describe a system in terms of its processes, people, and flow of information

- A critical first step is to determine the **system boundary**
- Anything within the system boundary will be mapped out and anything outside the boundary is out-of-scope

Drawing system boundar

E5.B – List relevant subsystems (5 min)

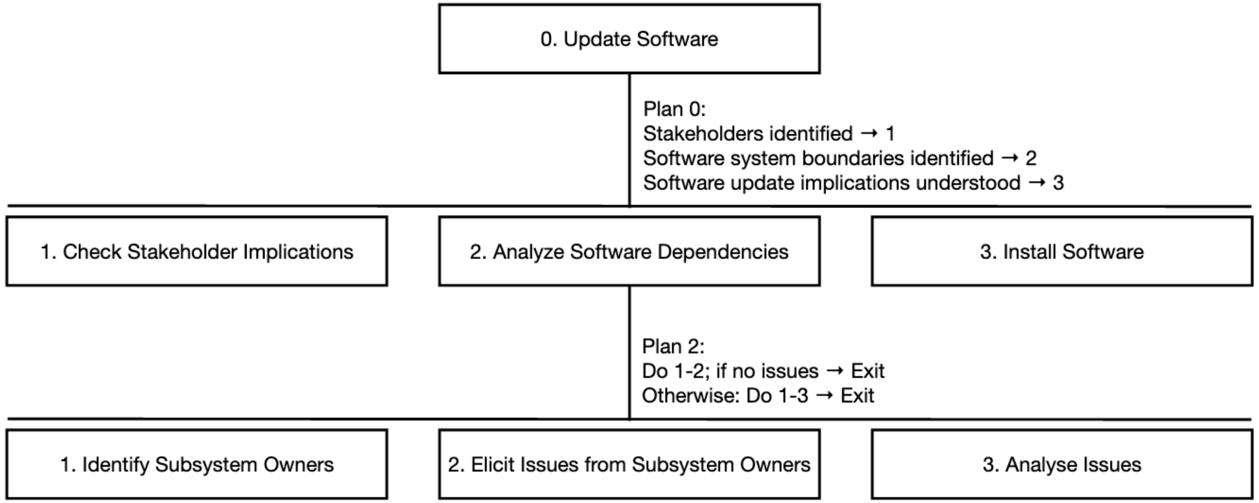
- Example 1: a wearable fall-detector that tracks patients in a hospital environment and warns a nurse of a fall

System mapping techniques
Example case: Updating software

1. Task diagram
2. Information diagram
3. Organizational diagram
4. System diagram
5. Process diagram
6. Communication diagram

Task diagram

- A hierarchical representation of tasks and conditions for carrying out the tasks
  - Tasks are nodes, relationships links
- Describe complete processes, organization of work, and user interface workflows
- Focus on processes and procedures, user behavior, and technology use

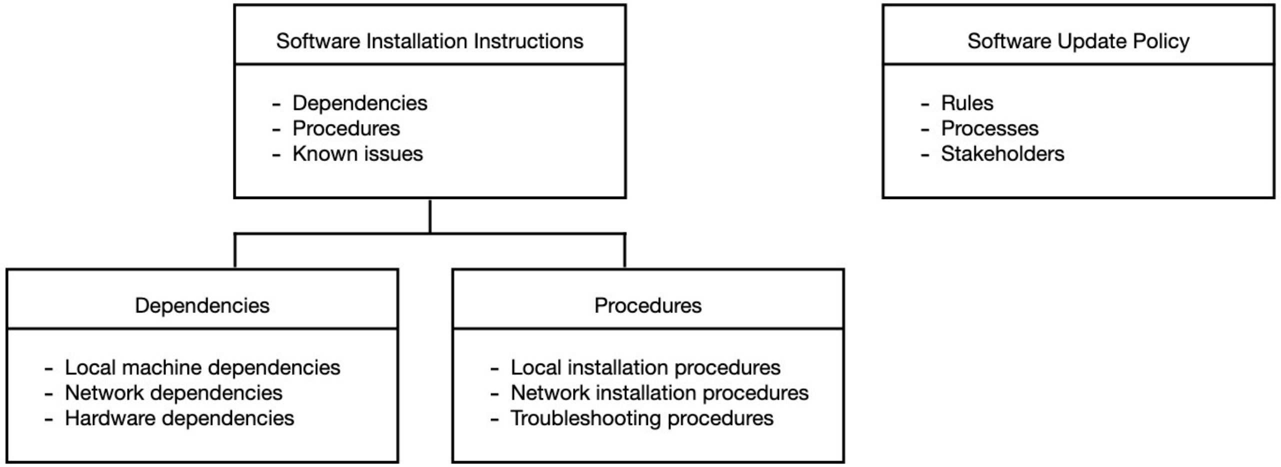


0. Update Software
Plan 0:
Stakeholders identified → 1
Software system boundaries identified → 2
Software update implications understood → 3
1. Check Stakeholder Implications
2. Analyze Software Dependencies
3. Install Software
Plan 2:
Do 1-2; if no issues → Exit
Otherwise: Do 1-3 → Exit
1. Identify Subsystem Owners
2. Elicit Issues from Subsystem Owners
3. Analyse Issues

A task diagram for updating software in an organization.

Information diagram

- A hierarchical representation of documentation
  - Documents are nodes and the relationships links
- Exposes documentation and policies within an organization relevant for some activity



Software Installation Instructions

- Dependencies
- Procedures
- Known issues

Software Update Policy

- Rules
- Processes
- Stakeholders

Dependencies

- Local machine dependencies
- Network dependencies
- Hardware dependencies

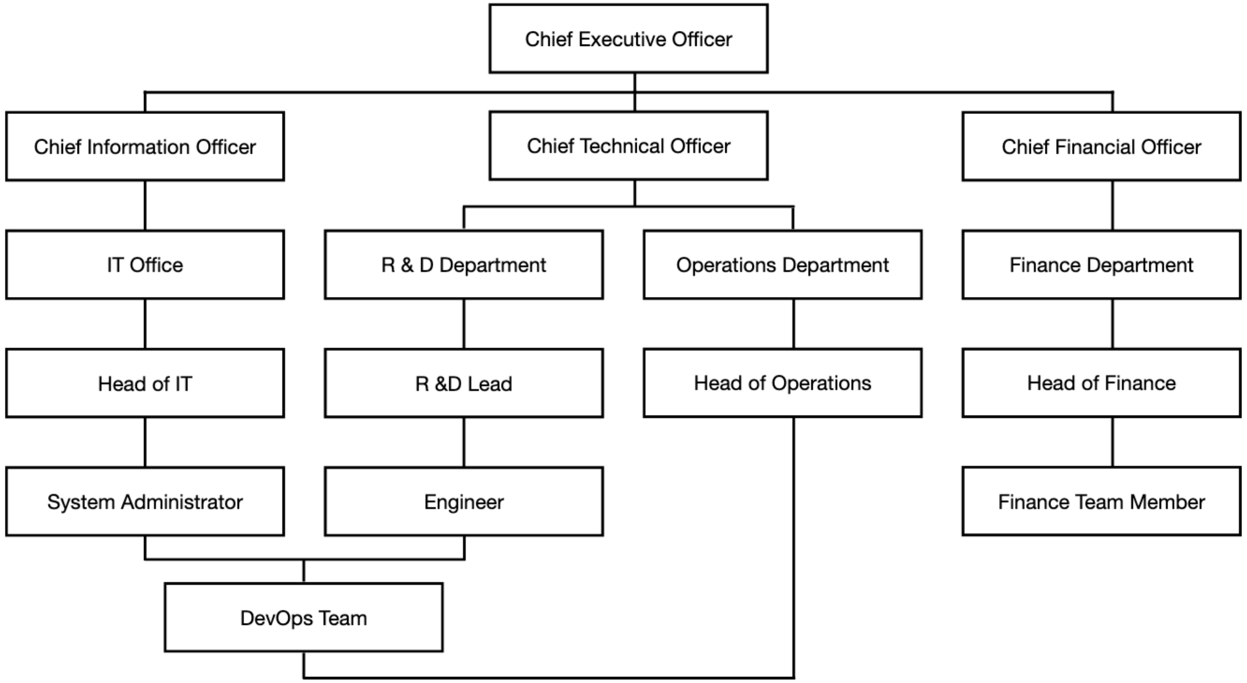
Procedures

- Local installation procedures
- Network installation procedures
- Troubleshooting procedures

Part of the hierarchy of information that a system administrator would require to do a software update in an organization.

Organizational diagram

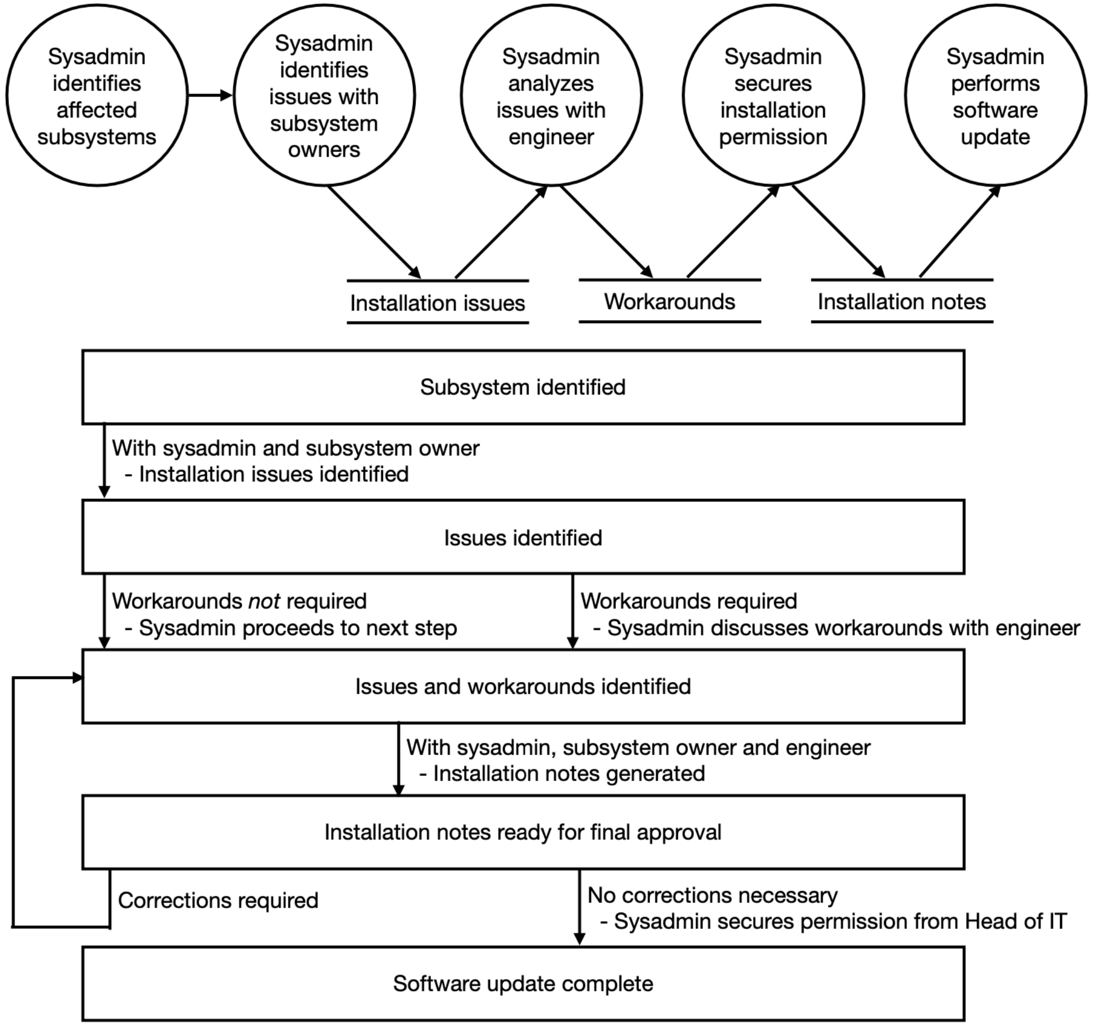
- A hierarchical representation of people and their roles in an organization
  - Teams, individuals, departments, etc. as nodes and relationships as links
- Make it possible to identify stakeholders and their roles
  - Can expose actors who are relevant even if not direct users of a system



An organizational diagram for a software update. The DevOps team combines the roles of software development and IT operations as one function.

System diagram

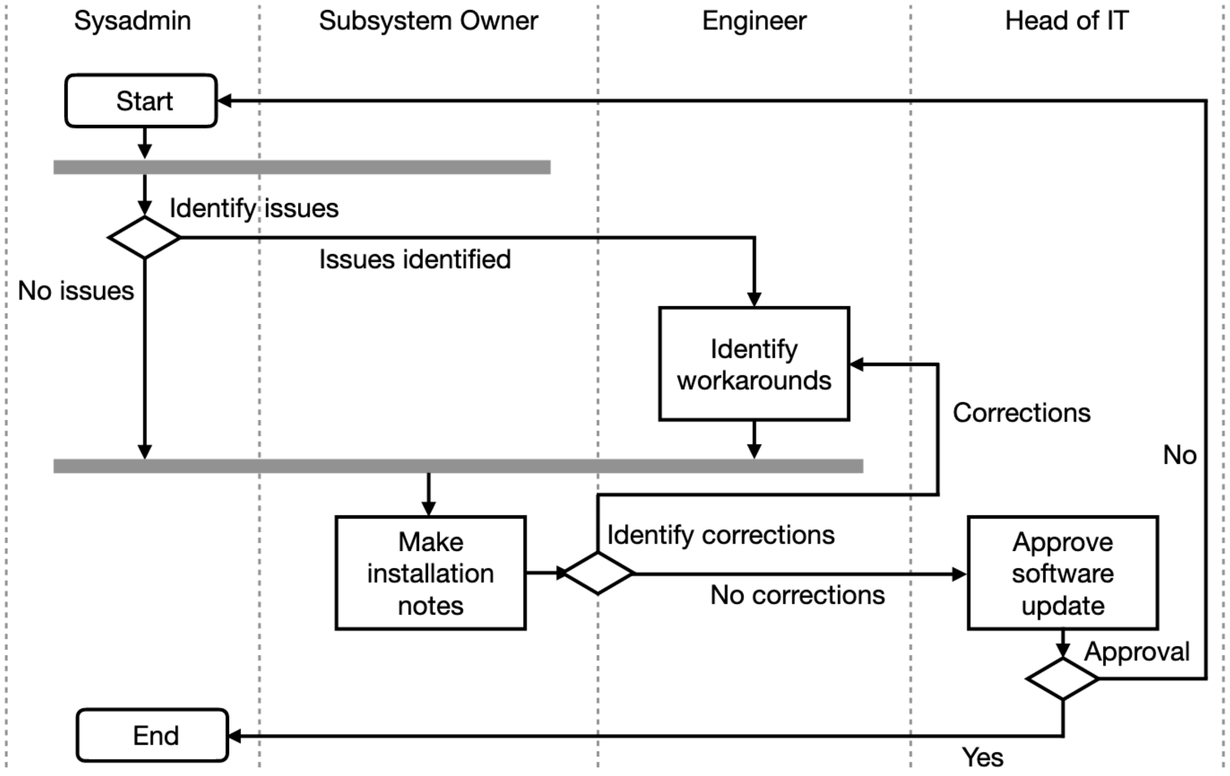
- Describes how data is is transformed through the system
    - Show where data is stored and how data activities and processes are sequenced to allow transformations
- Map out processes, in particular how users interact with systems in order to achieve tasks
- System diagram consists of (1) activities that indicate the flow of data between activities; and (2) states and state transitions that indicate the state conditions for a transition and the actions arising from a transition


Sysadmin identifies affected subsystems
Sysadmin identifies issues with subsystem owners
Sysadmin analyzes issues with engineer
Sysadmin secures installation permission
Sysadmin performs software update
Installation issues
Workarounds
Installation notes
Subsystem identified
With sysadmin and subsystem owner
- Installation issues identified
Issues identified
Workarounds *not* required
- Sysadmin proceeds to next step
Workarounds required
- Sysadmin discusses workarounds with engineer
Issues and workarounds identified
With sysadmin, subsystem owner and engineer
- Installation notes generated
Installation notes ready for final approval
Corrections required
No corrections necessary
- Sysadmin secures permission from Head of IT
Software update complete

A system diagram for updating software. The top part shows the flow of data between activities. The bottom part shows states as boxes and transitions between states as arrows. Conditions for a transition are indicated next to each arrow as a textual description and actions are indicated as a textual description preceded with a dash ('-').

Process diagram

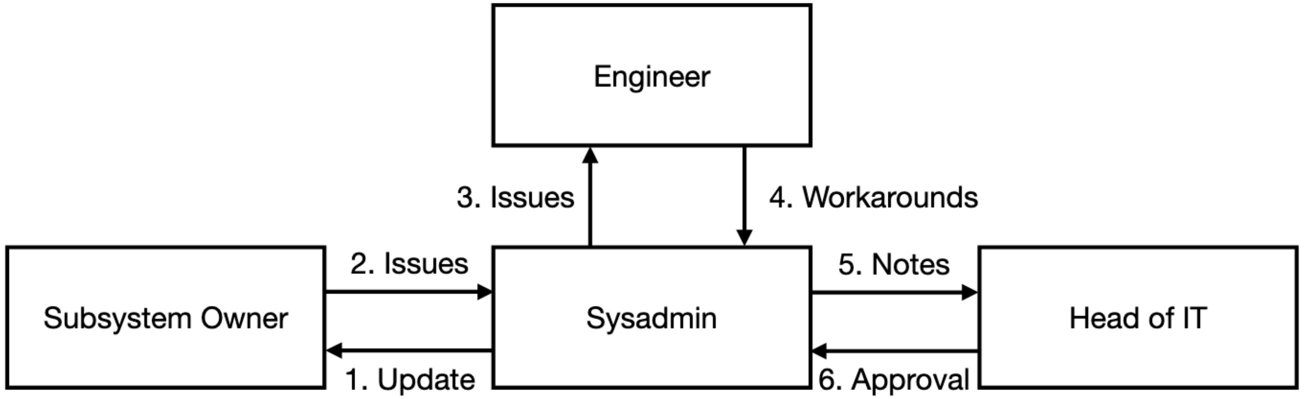
- Shows how serial and parallel processes and activities are structured as a series of steps
- An example of a process diagram is a *flow chart* (the figure shows a *swimlane diagram*)
    - Nodes represent the steps in a process and the links represent transition conditions
- Show the ordering of steps within activities, if such activities serial
- Are a basis to understand an overall process in a system, linking in, for example, relevant stakeholders, documents and tasks



An example of a process diagram for an organization deciding whether it is ready to update software.

Communication diagram

- Represents flow of information between users
  - Nodes are users, or an entity representing a user group, and links are flows of information
- Used to show the flow of information between people within and between teams
- Can also be used to depict flows of information across different entities, such as different departments in a company



E5.C (15 min)

- Select 2 mapping techniques relevant for your case
- Use them to describe your case
- Draw the diagrams
- (Additional assumptions will be needed, write them down)

Risk

Risk

- We will view **risk** from the point of view of **expected system behavior**
- A system has a certain purpose and the possibility that the system does not behave as expected can give rise to **undesired behavior**
- The risk of an incident that results in undesired behavior of the system can then be viewed as the expected value of undesired system behavior:

$$risk = likelihood \cdot impact$$

  - $likelihood$: the probability of a specific incident
  - $impact$: the expected loss due to this inci

Think: What kind of quantifiable loss is your case associated with?

Hazard

- **A hazard** is the possibility of an object, situation, information, or energy to cause an adverse effect
  - For example, a sharp edge in the molding of a wearable device can cause a cut on the user and a confusingly labeled button can cause users to accidentally delete data
- **Exposure** is the likely extent the user is exposed, or can be influenced by, the hazard
  - For example, if the sharp edge in the molding of the wearable device is covered by rubber coating there may be no exposure of the hazard to the user
- For a **risk** to exist there must be **both** a hazard *and* exposure to the hazard
  - An unexposed hazard is not a risk
- A risk is the product of the likelihood of an incident and the impact of the incident
  - A risk can therefore take on a range of values
- However, a risk is always preconditioned on the presence of a hazard

Risk management

- Having an an overall understanding of human error, it is important to design systems to minimize risks
- At its core, **risk management** is about identifying and assessing risks and minimizing, monitoring and controlling probabilities of undesired events
- At the high-level, risk management is a process with five steps:

1. **Hazard identification:** identify unintended system behavior that can cause unwanted outcomes
2. **Risk estimation:** arrive at a risk by assessing the likelihood and severity of each hazard
3. **Risk evaluation:** decide whether risks are acceptable
4. **Risk control:** reduce unacceptable risks to acceptable levels
5. **Risk monitoring:** manage a process that ensures acceptable risk levels are maintained throughout the system's lifetime

Risk assessment methods

- The first step in risk assessment is to set the **system boundary**
  - The system boundary defines the concerns of the system we are assessing: anything outside the boundary will not be assessed
- A large number of risk assessment methods exist that have been developed for various purposes, including medical devices, chemical process systems, aerospace systems, healthcare, etc.
- At a high level they can be considered along four dimensions:
  1. How **in-depth** they are, which affects the resources required to use them
  2. Their focus on **identifying** risks
  3. Their focus on **communicating** risks
  4. Their focus on **assessing** risks

Failure mode and effects analysis (FMEA)

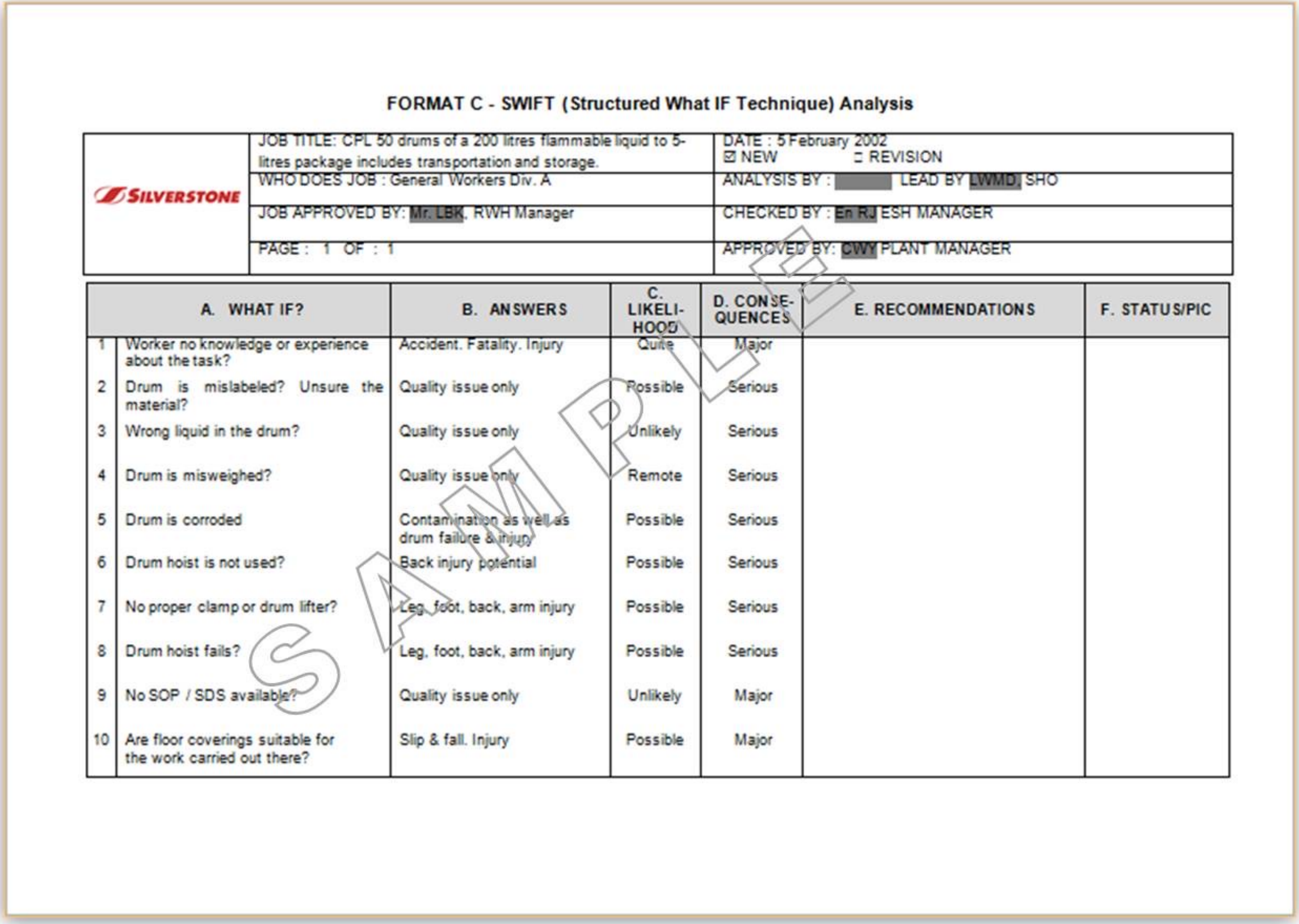
- Can be used to analyze human error at both the individual and the team level
- Analyzes component failures in the system
- Cnsiders each component’s failure modes and assesses possible causes for failures, their likelihood and severity, the recovery steps available, and actions for eliminating or mitigating the consequence of the failure mode
- There are multiple ways of carrying out FMEA
- The columns to the right is an example of the columns that an FMEA may include:

- **Identifier:** an identifier for the particular issue
- **Component:** the component assessed
- **Failure mode:** the specific failure mode of the component
- **Causes:** the possible causes of the failure mode
- **Probability:** the probability of the failure mode, which is an estimate
- **Severity:** the severity of the failure mode
- **Risk:** the risk of the failure mode: the product of the probability and severity
- **Recovery:** steps to mitigate the failure mode, which may include requirements for recovery and mitigation.
- **Action notes:** next steps to be taken, for instance, further investigation or changes to components

Structured what-if technique (SWIFT)

- A team-based risk assessment method that prompts the team with *what-if* questions to stimulate thinking about identifying risks and hazards in a system
- The focus on SWIFT is to allow the design team to explore different scenarios and contexts, and their resulting consequences, causes and impacts
- SWIFT is based on a vocabulary which serves as prompts
- The words in the vocabulary are used by a facilitator to discuss possible scenarios, issues, operating environment conditions, etc. that may give rise to hazards and risks
- The words used as prompts typically focus on deviations, such as “failure to detect”, “wrong message”, “wrong time”, “wrong delay”, etc.

- In practice, a SWIFT analysis is carried out by filling out a table where each row has a set of columns
- Example columns for analysis:
  - **Identifier:** an identifier for the particular issue discussed
  - **What-if questions:** questions triggering an assessment, such as “how much”, “how many”, etc.
  - **Hazards and risks:** any hazards and associated risks that may occur
  - **Relevant controls:** the controls that are in place or need to be in place to mitigate the risk
  - **Risk ranking:** the ranking of this risk relative to other risks identified in the exercise
  - **Action notes:** next steps to be taken



FORMAT C - SWIFT (Structured What IF Technique) Analysis

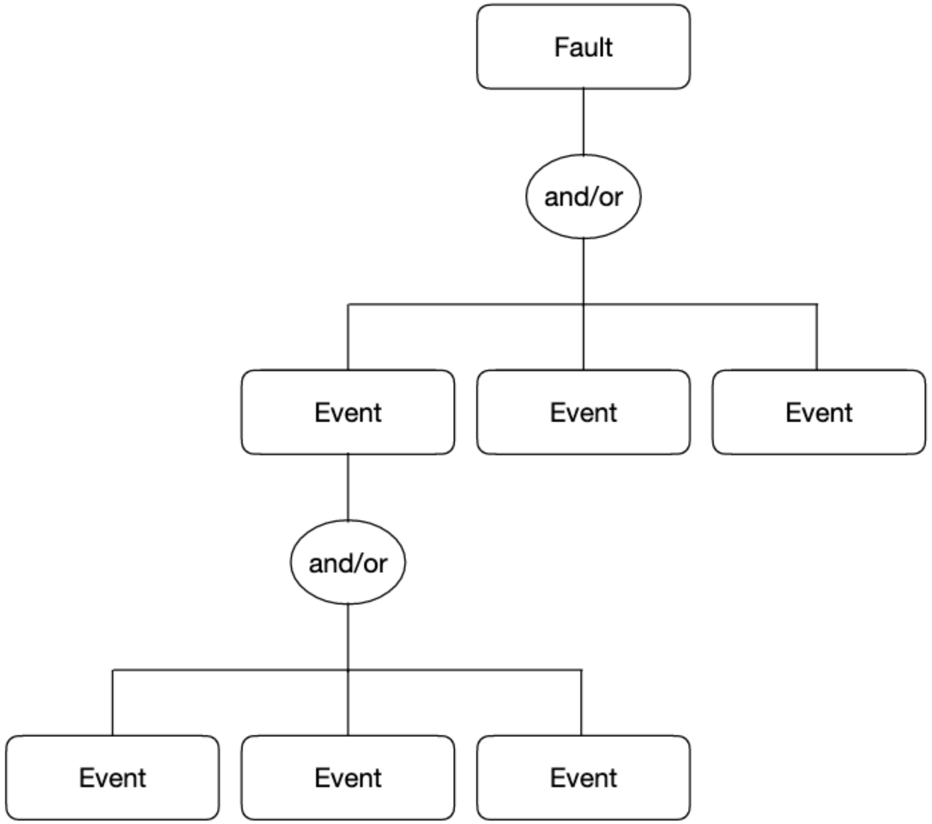
SILVERSTONE	JOB TITLE: CPL 50 drums of a 200 litres flammable liquid to 5-litres package includes transportation and storage.	DATE : 5 February 2002 ☑ NEW □ REVISION
	WHO DOES JOB : General Workers Div. A	ANALYSIS BY : [redacted] LEAD BY LWMD, SHO
	JOB APPROVED BY: Mr. LBK, RWH Manager	CHECKED BY : En RJ ESH MANAGER
	PAGE : 1 OF : 1	APPROVED BY: CWY PLANT MANAGER

	A. WHAT IF?	B. ANSWERS	C. LIKELI-HOOD	D. CONSE-QUENCES	E. RECOMMENDATIONS	F. STATUS/PIC
1	Worker no knowledge or experience about the task?	Accident. Fatality. Injury	Quite	Major		
2	Drum is mislabeled? Unsure the material?	Quality issue only	Possible	Serious		
3	Wrong liquid in the drum?	Quality issue only	Unlikely	Serious		
4	Drum is misweighed?	Quality issue only	Remote	Serious		
5	Drum is corroded	Contamination as well as drum failure & injury	Possible	Serious		
6	Drum hoist is not used?	Back injury potential	Possible	Serious		
7	No proper clamp or drum lifter?	Leg, foot, back, arm injury	Possible	Serious		
8	Drum hoist fails?	Leg, foot, back, arm injury	Possible	Serious		
9	No SOP / SDS available?	Quality issue only	Unlikely	Major		
10	Are floor coverings suitable for the work carried out there?	Slip & fall. Injury	Possible	Major		

SAMPLE

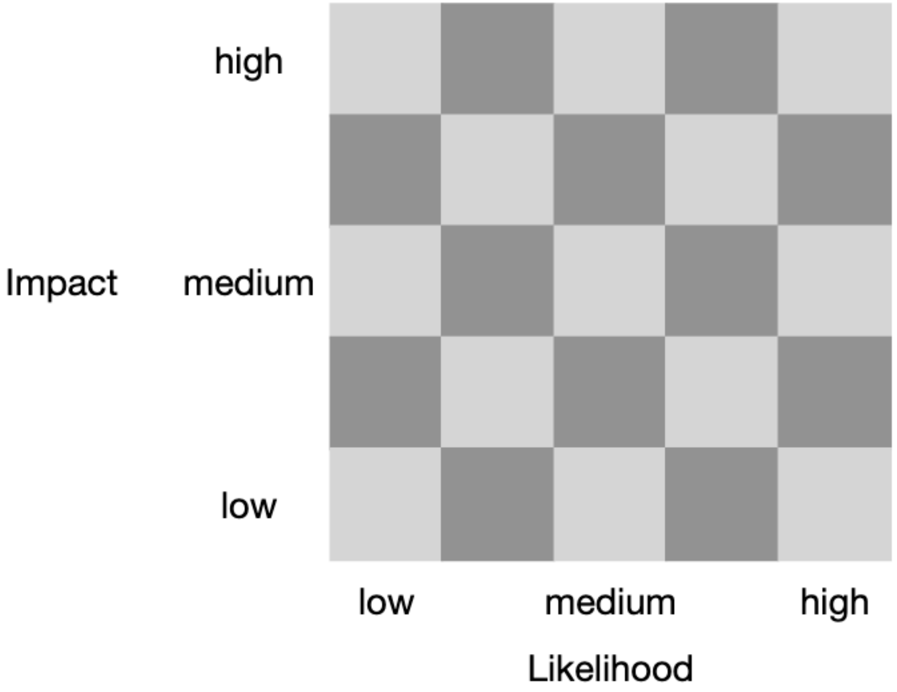
Fault tree

- A diagrammatic method for identifying and analyzing factors contributing to a *fault*—unintended behavior
- Fault trees are, as their name indicates, tree diagrams linking factors to a fault using logical relationships, such as *and* and *or*
- A fault tree is created by starting with the fault as the top-level event and then progressively analyzing the factors that may contribute to the fault
- Fault trees can be used to analyze causes of human error
- They highlight interrelationships between components, where components may be both system components and users



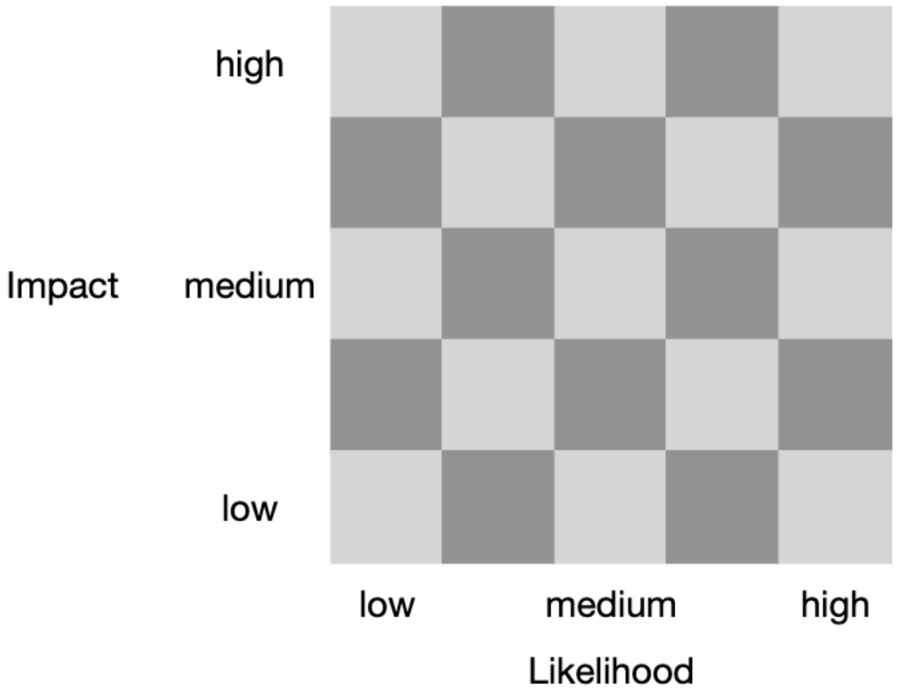
Risk matrix

- A **risk matrix** is a simple visualization technique for communicating risk
- A risk matrix has two axes: **impact** and **likelihood**
- Risks are indicated in the matrix accordingly
- A risk matrix makes it easy for a team to visualize important risks
- In general, risks in the top-right are more severe and they indicate risks with a high impact and a high likelihood
- Risk matrices are typically used in conjunction with other risk assessment methods to assist with ranking and prioritizing risks



E5.D Risk matrix (15 min)

- Assess the diagrams you made for potential risk
  - Identify threats
    - Who would do what and why and with what resources?
  - Identify the impacts and likelihood
- Draw a risk matrix for top 3 threats scenarios



Pairwork topics

Task and presentation

Task

General AI safety issue

→ Concrete scenario

→ System analysis

→ Risk assessment

→ Recommendations

Notes:

- Pick a concrete subcase to focus on
- Your topic be human factors-related (e.g., not just an SW bug)
- Bonus: Quantify the risk (losses)
- Feel free to use articles in the media and even ChatGPT to help you out

Presentation

1. General AI capability and general concern
2. Concrete scenario + illustrative photo (bullets only)
3. Your system analysis
   (2-3 diagrams annotated)
4. Your risk assessment (risk matrix or some other form)
5. Your recommendations for counter-measures