Transcript of the Swiss cheese model of system accidents

We can adapt James Reason's (2000) Swiss cheese model of system accidents to visualise how multiple conditions must align in order for an IT incident to transpire.

A complex system, like an IT organisation, has multiple layers designed to prevent hazards from causing an incident.

These defences might include engineering safeguards, like code reviews and design practices.

And quality safeguards, like user inspection and automated testing.

And operational safeguards, like platform health and monitoring procedures.

However, these defences are not perfect, and like a slice of Swiss cheese, each layer has inherent weaknesses or "holes."

These holes can allow hazards to pass through.

A design flaw in the software might get caught in code review.

A software bug might be missed in code review, but caught in testing.

And occasionally, a combination of factors can line up and allow a hazard to pass through all of the system's defences.

By understanding how these layers can fail, organisations can work to identify and address the weaknesses in their systems.

References

Reason, J. (2000) 'Human error: models and management', *BMJ*, 320(7237), 768–770, available: https://doi.org/10.1136/bmj.320.7237.768.
