NTNU

SafeScrum

SINTEF

Agility and Resilience

Tor Stålhane NTNU / IDI

Stig Ole Johnsen NTNU / IDI, SINTEF / Safety and Mobility

What is resilience

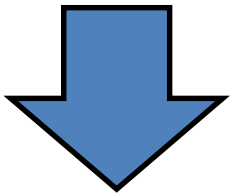
The two key attributes are dependability and robustness. A computing system can be said to be robust if it retains its ability to deliver service in conditions which are beyond its normal domain of operation".

A shorthand definition of resilience:
The persistence of dependability and adaptability when facing changes.

Why do we need resilience

We live in a continuously changing world =>

New challenges appear “all the time”



When developing safety critical systems we need to handle the safety challenges that are

- Known – safety analysis, e.g., FMEA, PHA or HazOp
- Unknown – resilience

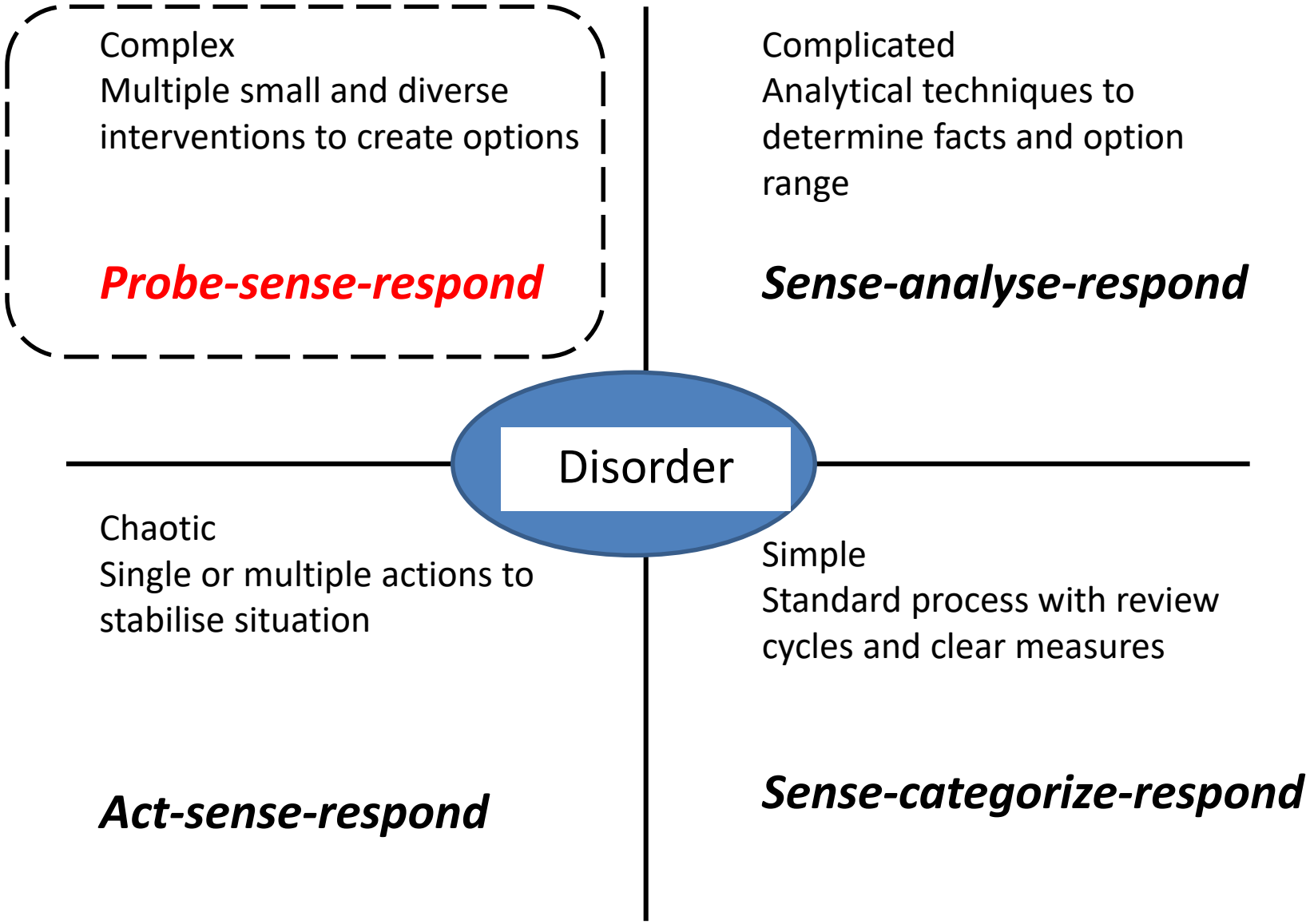
How can we achieve resilience

In order to realize resilience, we need to understand what goes wrong and what does not

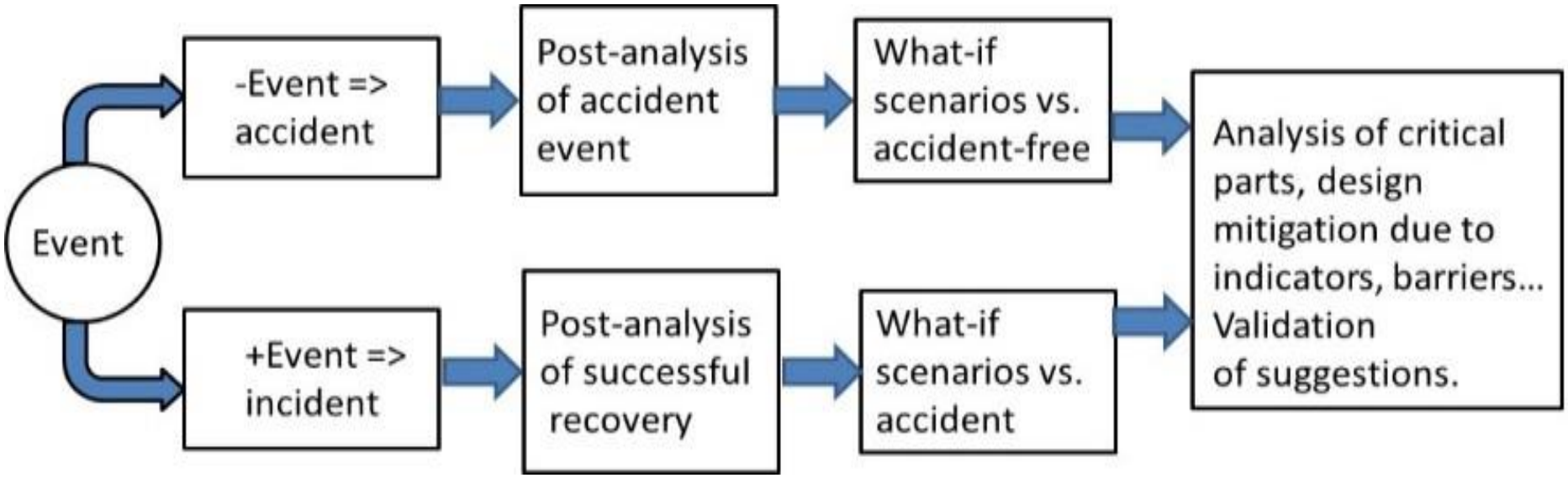
We will use

- The Cynefin model as a starting point
- STEP analysis and resilience engineering to specify resilience requirements
- An agile development process

The Cynefin model



Probe – sense – respond



Both known accidents and successful recoveries should be explored in such a process. The diagram illustrates how they should be explored.

How to obtain resilience

We need the following steps

1. STEP analysis – including scenario development. We need to understand how technology, humans and organizations interacts.
2. Resilience engineering – how to detect, handle and recover
3. Resilience requirements, based on resilience patterns

The STEP analysis

- Identify all relevant stakeholders – actors
- Define one or more ***scenarios*** – what do the actors do?
- Analyse how the actors in the scenarios interact to create or prevent an accident

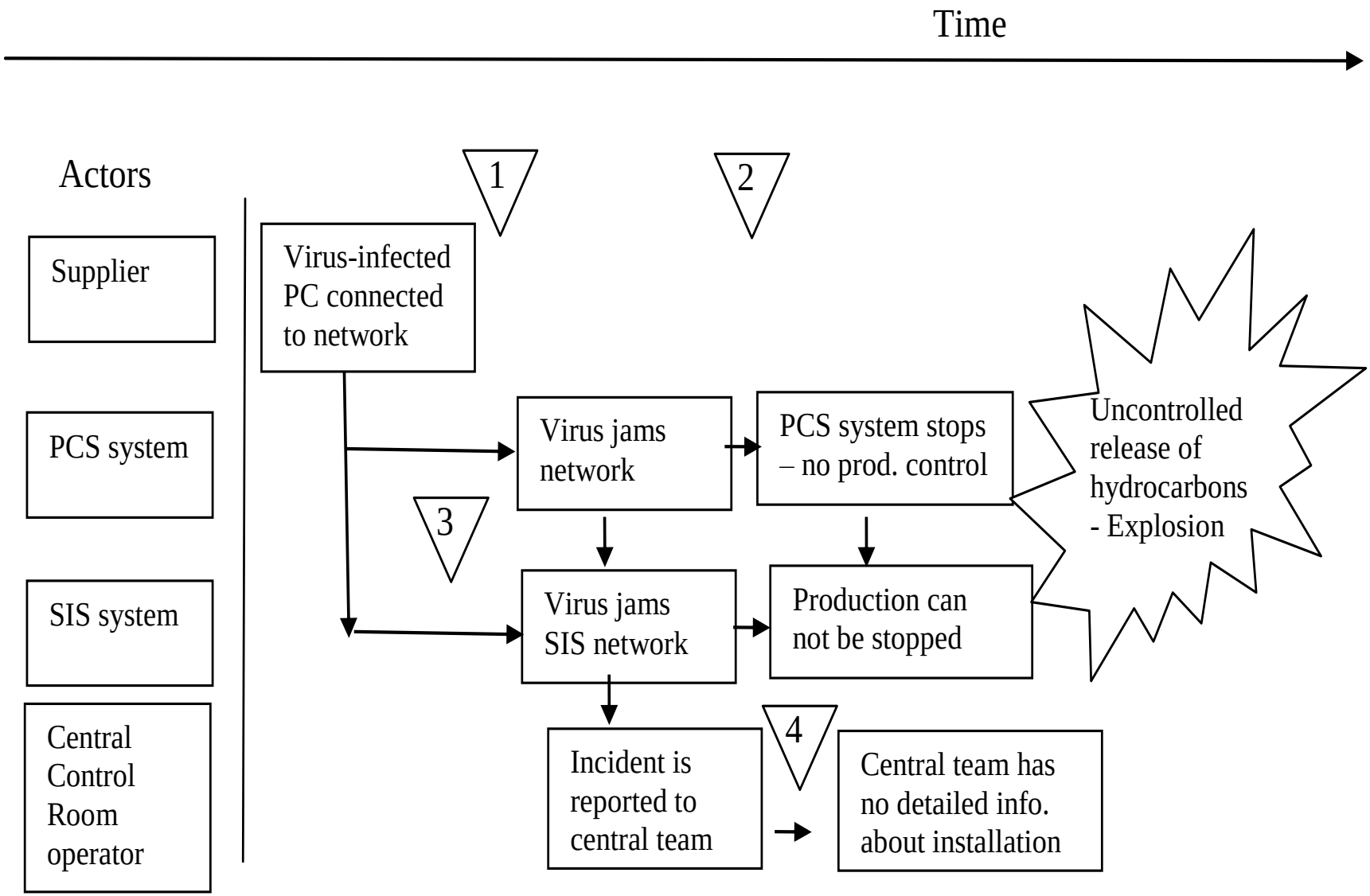
Actor identification

1. Actor – involved in the events leading up to the accidents by their own actions, decisions or omissions. The actors are drawn on the left side of the STEP diagram.
2. Identify events that influenced the accident – ”whom”, ”what” and ”how”, and place them in the diagram in the order they occurred.
3. Place events in the time-actor sheet.
4. Identify the relationship between the events and show this in the diagram by drawing arrows to illustrate the causal links.

Scenario analysis

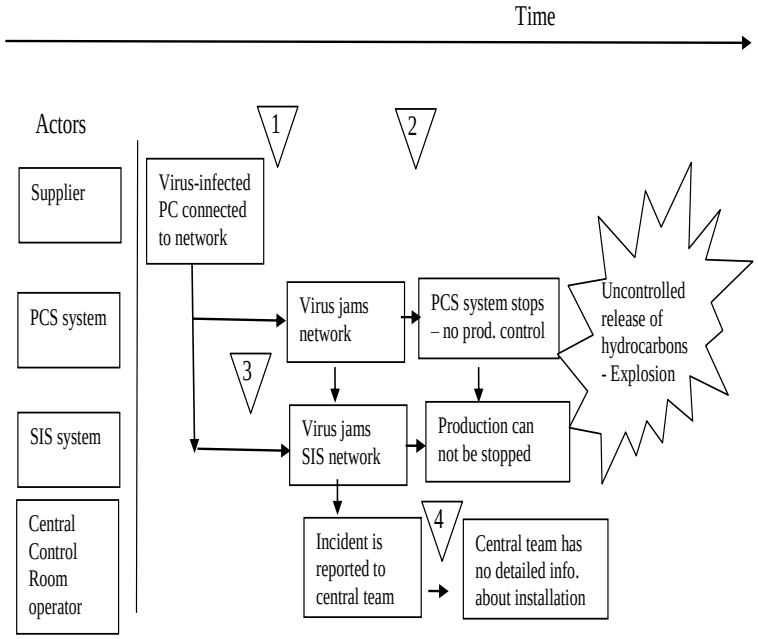
- For each event involving the actor, ask questions regarding Observation, Identification, Interpretation, Decision making, Planning, Execution.
- If an event relates to the actor
  - **receiving information**; questions regarding human-system interface may be appropriate
  - **making decisions**; questions regarding training, procedures and time available may be appropriate etc.

STEP example



Example

1. No scanning of PC prior to connection to network
2. Latest patches not deployed to network and systems connected to network, making a successful virus attack more probable.
3. SIS network integrated in PCS system, the SIS/PCS network are common, making it possible to jam the SIS through the PCS system.
4. The technical central team has not sufficient knowledge of the local complex SCADA system and does not manage to stop or shut down production.



Resilience engineering

Resilience is "the ability of a system to handle unexpected situations and recover".

Resilience Engineering focuses on how to:

- Detect early and avoid (if possible)
- Handle – ideally without disruption (but may be reduced state)
- Recover – i.e. fail fast/ get back on track and learn (agile iterative learning)

Resilience requirements – 1

Process requirements are e.g. to require solutions that

- provide loose coupling between components
- reduced code complexity
- sustain a common mental model in the project.

More resilience may lead to more complex code => lower reliability.

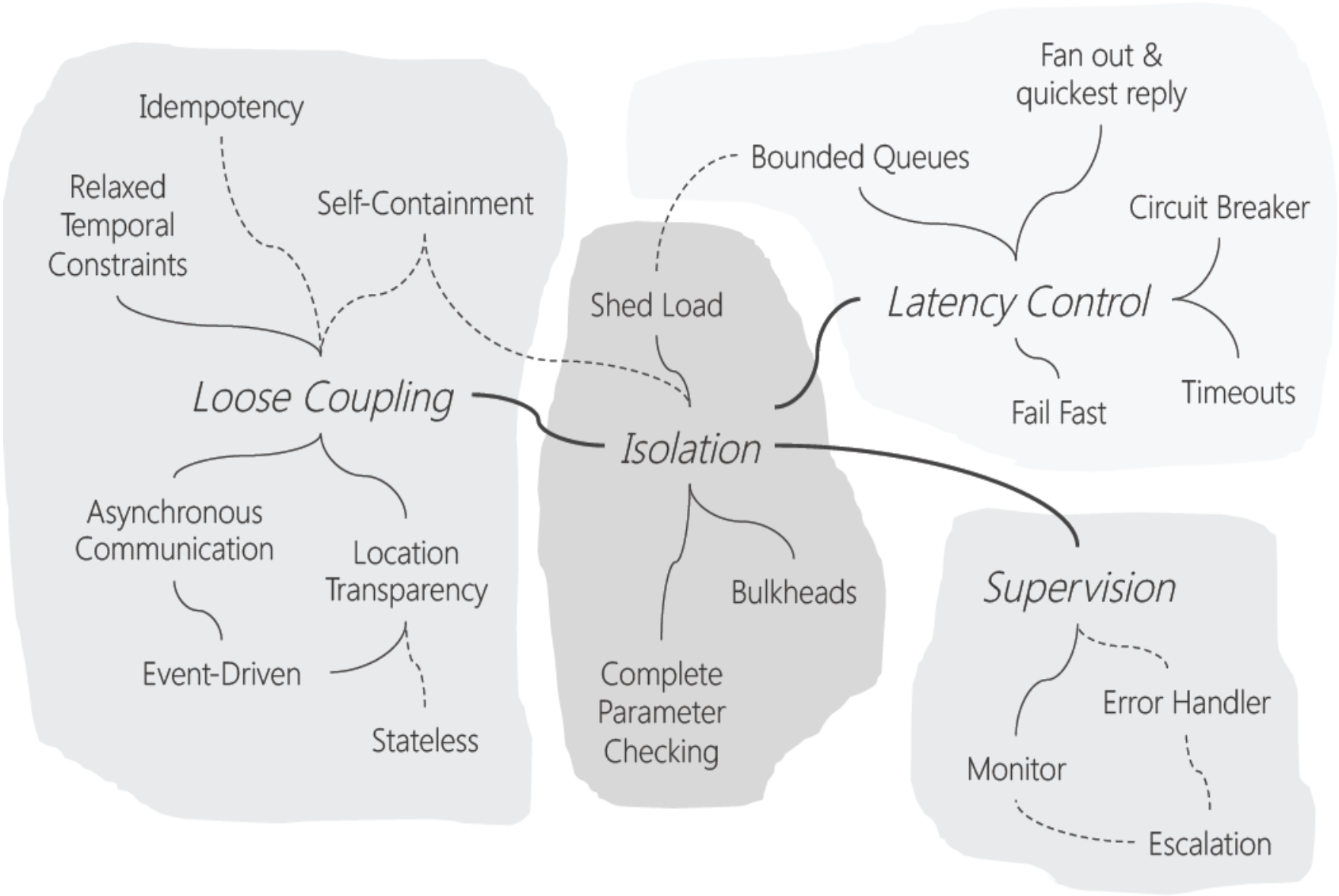
Over time an organizations should

- build up a library of known threats.
- search the net for reported events and add own experiences.

Resilience requirements – 2

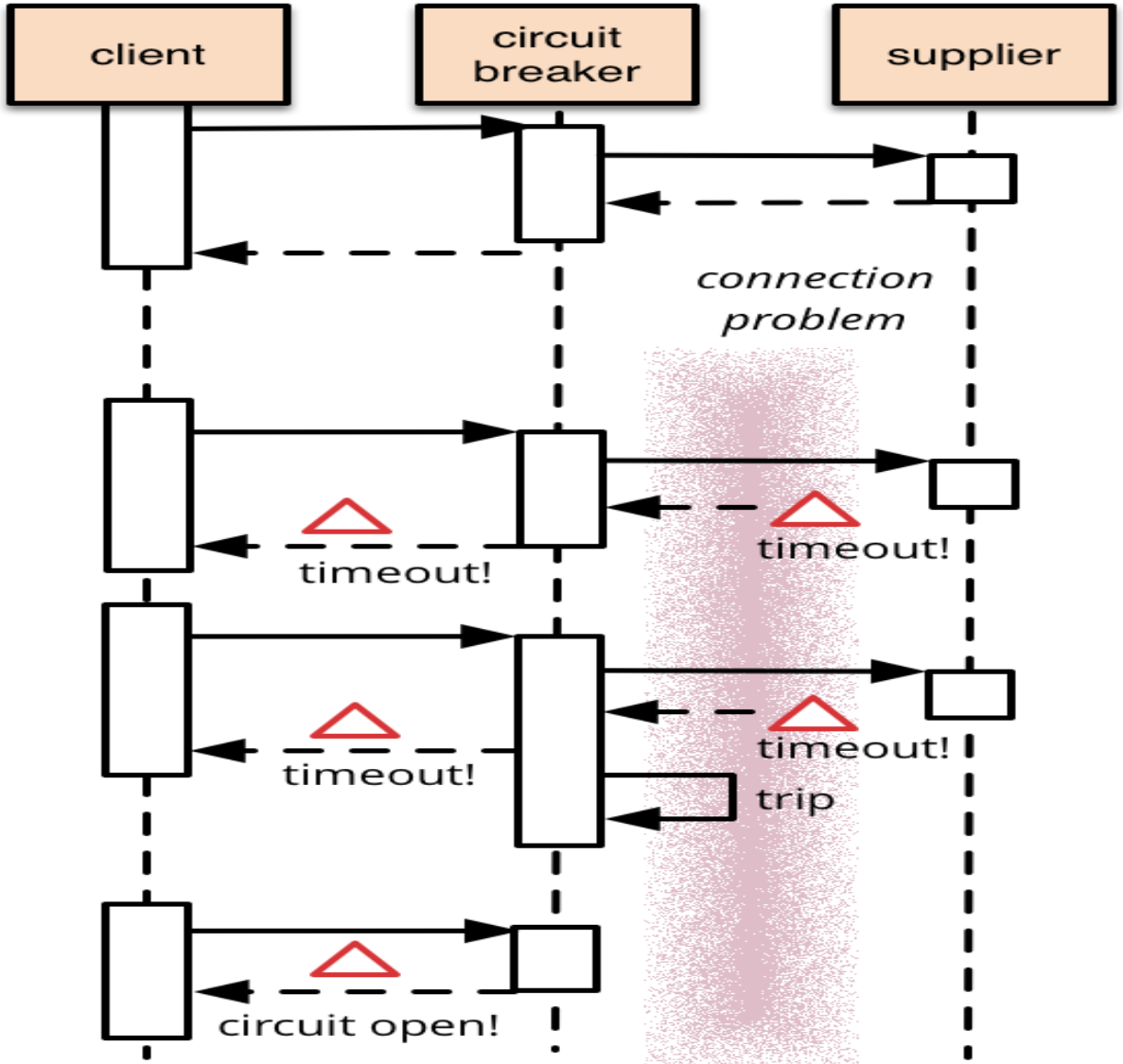
- Manage system margins – The ability to adjust before boundaries are crossed.
- Build common mental models – share Situational Awareness:
- Flexibility – Accept improvisations, support error tolerance => alternative means to perform key functionality and recovery procedures,
- Reduction of complexity
- Reduction of coupling – Reduced coupling => flexibility in of sequencing and in methods used to reach the goal.

Resilience patterns



Resilience requirement - example

If the remote sensor unit stops working it shall be disconnected

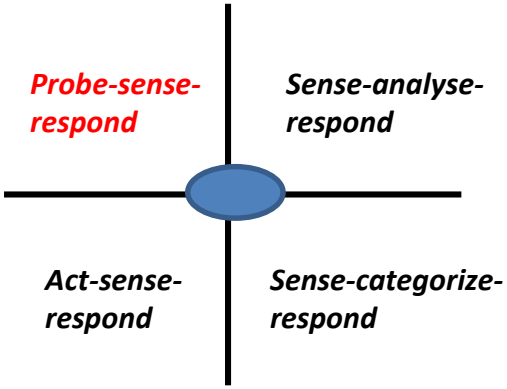


Many threats are unknown or diffuse

Resilience focuses on the assets and key processes that we want to protect.

Resilience is about

- Self-organizing systems – able to adapt to unpredictable situations
- Reactive responses – able to evolve with a changing environment
- Proactively innovative



A SafeScrum process for resilience – 1

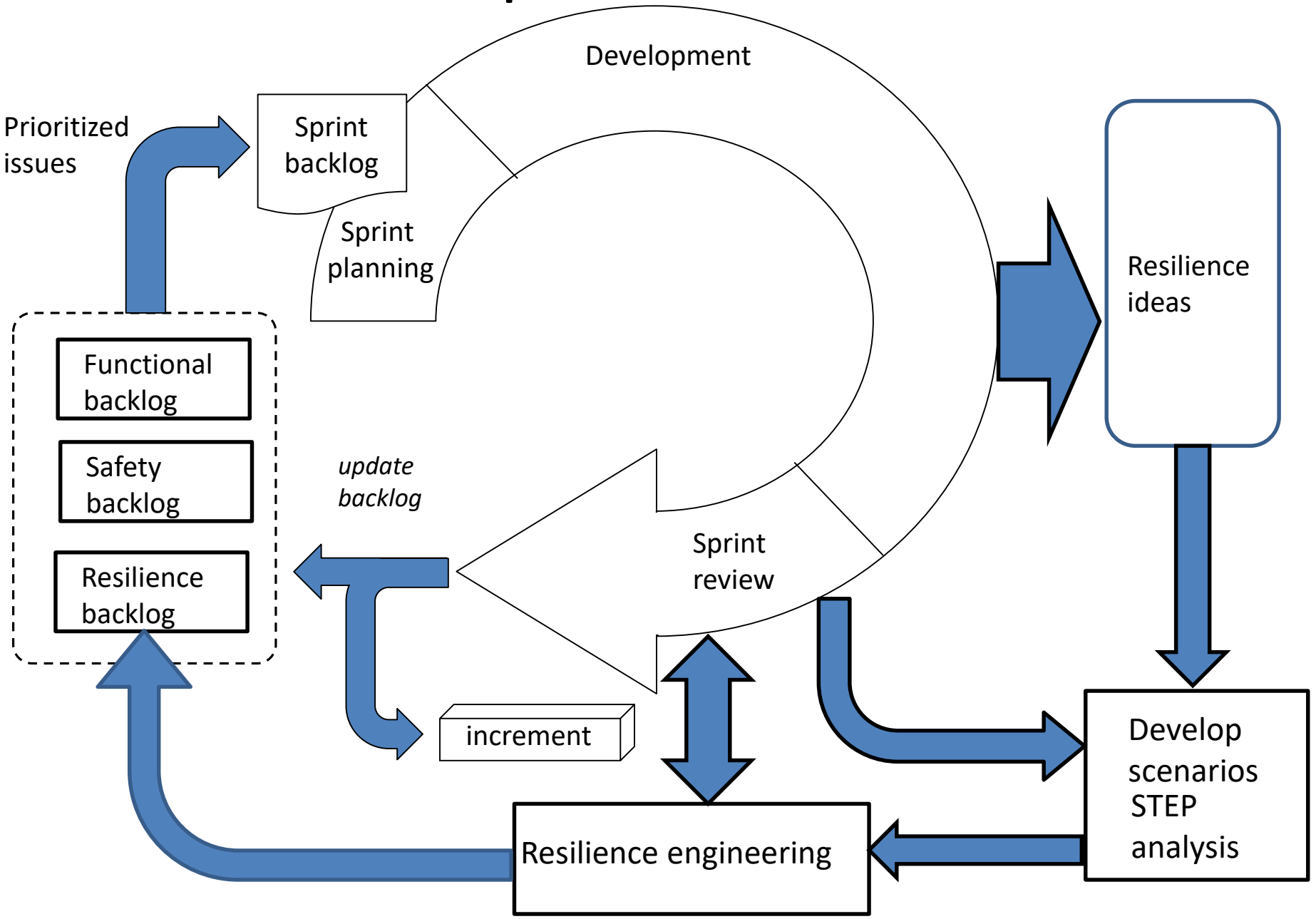
Use some minutes of each daily stand-up to brainstorm on emergent threats that can reduce safety. This will:

- Make the developers pay attention to resilience issues.
- Develop scenarios on emergent threats and vulnerabilities together with possible solutions

Sketch possible solutions. Have an in-depth discussion at the end of each sprint as part of the sprint review / retrospective.

- If necessary, involve resilience and domain experts.
- Link new requirements to the solution sketches to get an early start on the implementations.

A SafeScrum process for resilience – 2



How can agile development help

New resilience needs will be discovered throughout development. Thus, we need

- Effective communication among all stakeholders, including the customer
- Rapid response to new resilience challenges

=> Rapid, incremental delivery of software

Resilience and agility is a good match

Create a separate backlog for resilience requirements. This will help to increase the developers' resilience awareness.

Resilience requirements will be linked to other requirements =>
Remind developers that resilience requirements may influence functional requirements and safety requirements and vice versa.

Proactive innovations

Proactive innovation with “speculative assemblies for unknown needs.”

A daily focus on resilience will create resilience culture, i.e., knowledge and awareness in the SafeScrum team.

Where do we go from here

Important issues:

- Systematic collection and analysis of system failures in order to learn more
- Registration and analysis of near-misses
- Focus on resilience issues throughout development