“Develop a passion for learning.”

RISK AND RESILIENCE BOOTCAMP

TEKsystems Global Services

WORKFORCE DEVELOPMENT

SYSTEMS THINKING

This module introduces some of the basic ideas of systems thinking

- Defining systems thinking
- How it applies to risk analysis
- Cascading failures
- Managing complexity

SYSTEMS THINKING

- Systems thinking is a way of seeing and talking about reality
  - It emphasizes wholes, interrelationships and patterns of change rather than just isolated parts
  - It is a mindset that focuses on how everything is connected
  - Coupled with a set of tools/techniques that help make sense of complexity
    - For example: causal loop diagrams, behaviour-over-time graphs, stock & flow models
  - Doesn’t just look at one system or one process in isolation
  - Asks instead
    - How do the people, the process, and the systems interact?
    - What feedback loops, dependencies, delays or unintended effects arise when these interplay?

SYSTEMS THINKING

- Why systems thinking matters
  - IT systems, business processes, and human actors are deeply intertwined
  - A change in one often shifts behaviour in others, sometimes in unexpected ways
  - Without systems thinking
    - There is the risk of designing controls or modifying processes in a silo, missing side-effects in other processes and missing hidden dependencies that might cause cascading failures
  - Systems thinking helps identify emergent behaviours
    - Specifically that the whole is greater than the sum of parts
    - For example, when a process runs across multiple teams and systems, the risk is not just each part failing, it's how their interactions can produce cascading or amplifying effects

SYSTEMS THINKING

- Using systems allows an analyst to
  - Map feedback loops, both positive and negative
  - Recognize delays and time lags between action and effect
  - See non-linearities where a small change can triggers much larger changes
  - Understand dependencies among people, process and systems
  - Move the analysis of causes
    - From “What happened in system X?”
    - To “Why did the interaction across systems/processes/people cause the failure?”

SYSTEMS TRIAD

- people-process-systems triad
  - People
    - Roles, communication, culture, decision-making, skill levels, hand-offs
  - Process
    - Flow of tasks, decision points, hand-offs, controls, workflows, exceptions
  - Systems (IT / technical)
    - Software, hardware, interfaces, data flows, automation, integrations
- Systems thinking
  - Emphasizes how these three overlap and interact.
  - Example:
    - A process may call for a manual hand-off (people), but if the system interface is slow (systems), it may cause delay (process) which leads to human workaround, raising risk (people) of human error

APPLYING SYSTEMS THINKING

- Define the boundary of your system
  - Identify what "system" you are analyzing
    - For example: "vendor on-boarding process + IT systems + stakeholders"
  - Clarify what's inside, what's outside, and how it interacts with the environment.
- Map the elements and their relationships
  - Use diagrams (causal loop diagrams, stock/flow diagrams, behaviour-over-time) to visualise how people, processes and systems interact
- Identify feedback loops and delays:
  - For example
    - "System slows → manual work increases → backlog grows → errors increase → system redesign becomes urgent"

APPLYING SYSTEMS THINKING

- Look for unintended consequences
  - When you change one part what are impacts elsewhere
  - For example:
    - If we automate a manual approval, what happens elsewhere?
    - Are there other processes that are affected downstream? Does this impact regulatory compliance in other processes?
- Analyze the dynamic behaviour over time
  - Risk is not always at a single moment
  - Delays, accumulation of backlog, system degradation, human fatigue are all time-based phenomena
- Use systems thinking to inform mitigation
  - Once you see the interplay, you can design interventions that take into account people, process and systems together (not just patch one)

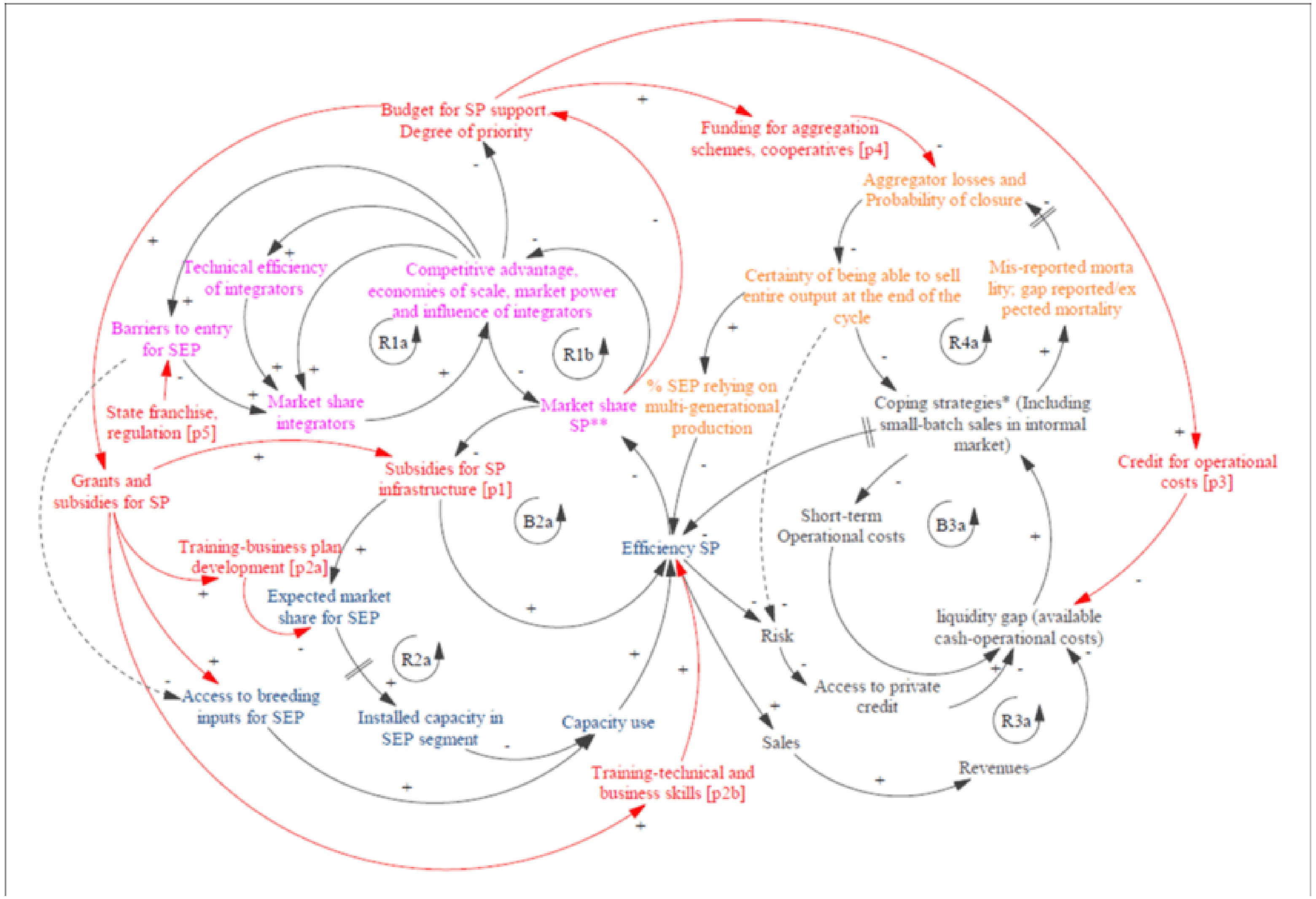
SYSTEMS THINKING TOOLS

- Causal loop diagram
  - Visualizes cause-and-effect chains and feedback loops
  - Useful to show how a process flaw or system delay may loop back and increase risk
- Behaviour over time graph
  - Chart of how a key variable changes over time
  - For example: backlog size, error rate, system performance

SYSTEMS THINKING TOOLS

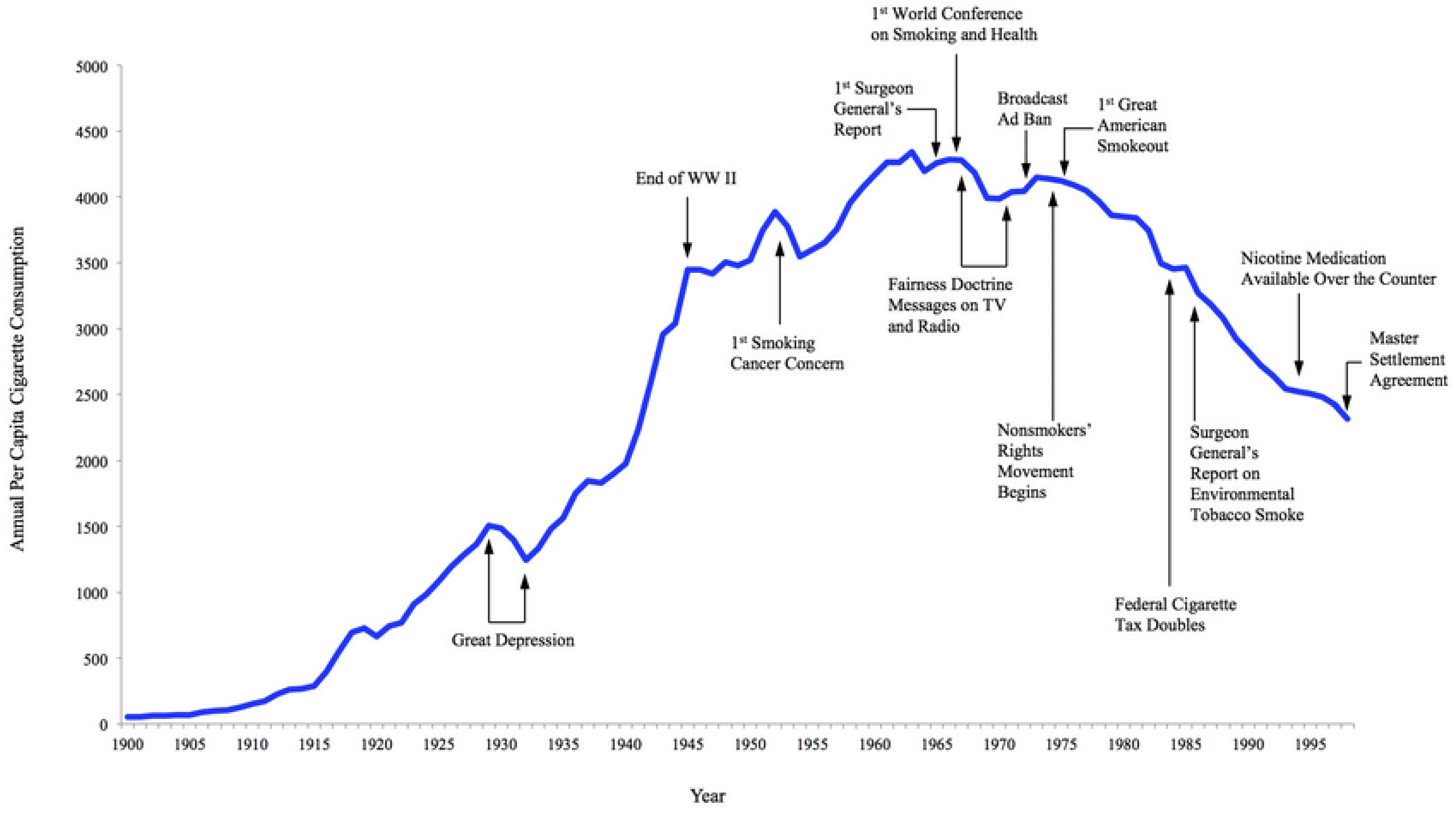
- Stock and flow diagram
  - Shows accumulation (stock) and rates (flows) in processes
    - For example: tasks waiting in queues for processing
- Rich pictures
  - Qualitative depiction of system relationships, useful early in workshops to surface how people perceive the system
- Systems archetypes
  - Generic patterns like “Shifting the Burden”, “Fixes that Fail”, “Drifting Goals”
  - These help recognize recurring risk patterns in people-process-system interaction
  - Sort of like design patterns for identifying problematic systems architectures

CAUSAL LOOP DIAGRAM

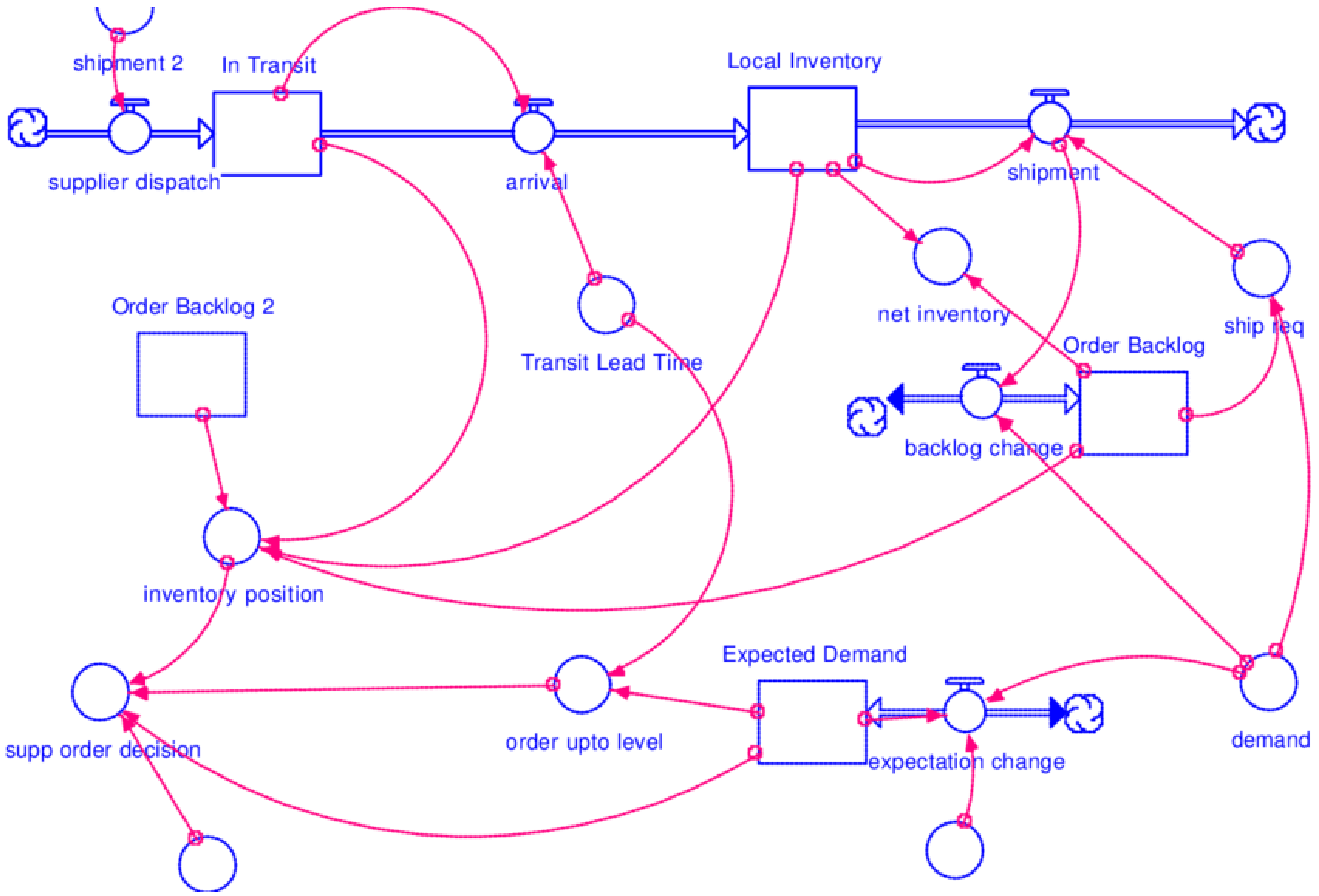


Budget for SP support
Degree of priority
Funding for aggregation schemes, cooperatives [p4]
Aggregator losses and Probability of closure
Technical efficiency of integrators
Competitive advantage, economies of scale, market power and influence of integrators
Certainty of being able to sell entire output at the end of the cycle
Mis-reported mortality; gap reported/expected mortality
Barriers to entry for SEP
R1a
R1b
R4a
% SEP relying on multi-generational production
State franchise, regulation [p5]
Market share integrators
Market share SP**
Coping strategies* (Including small-batch sales in informal market)
Grants and subsidies for SP
Subsidies for SP infrastructure [p1]
Credit for operational costs [p3]
Short-term Operational costs
B2a
B3a
Efficiency SP
Training-business plan development [p2a]
Expected market share for SEP
Risk
liquidity gap (available cash-operational costs)
R2a
Access to breeding inputs for SEP
Installed capacity in SEP segment
Capacity use
Access to private credit
Sales
R3a
Revenues
Training-technical and business skills [p2b]

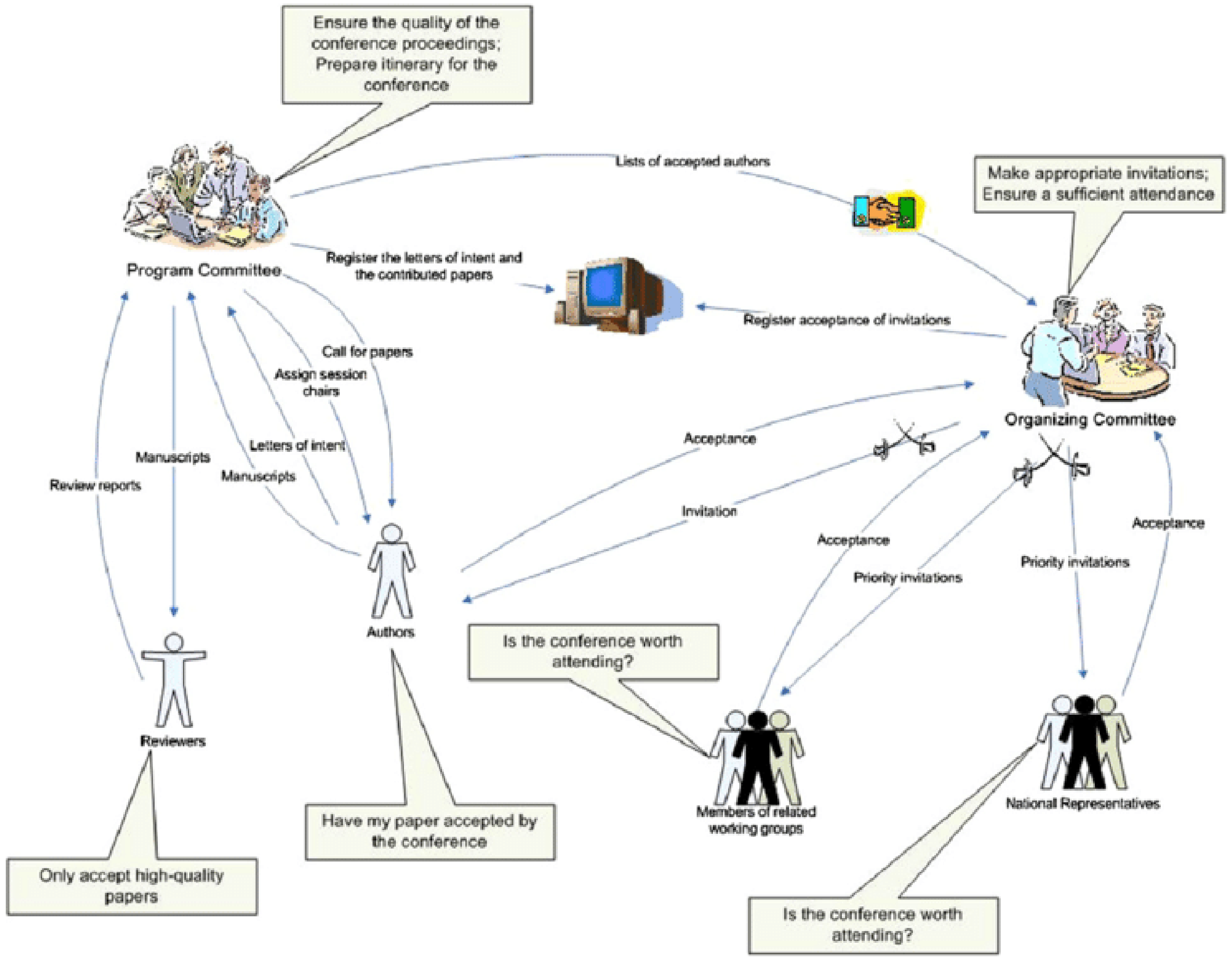
BEHAVIOUR OVER TIME GRAPH



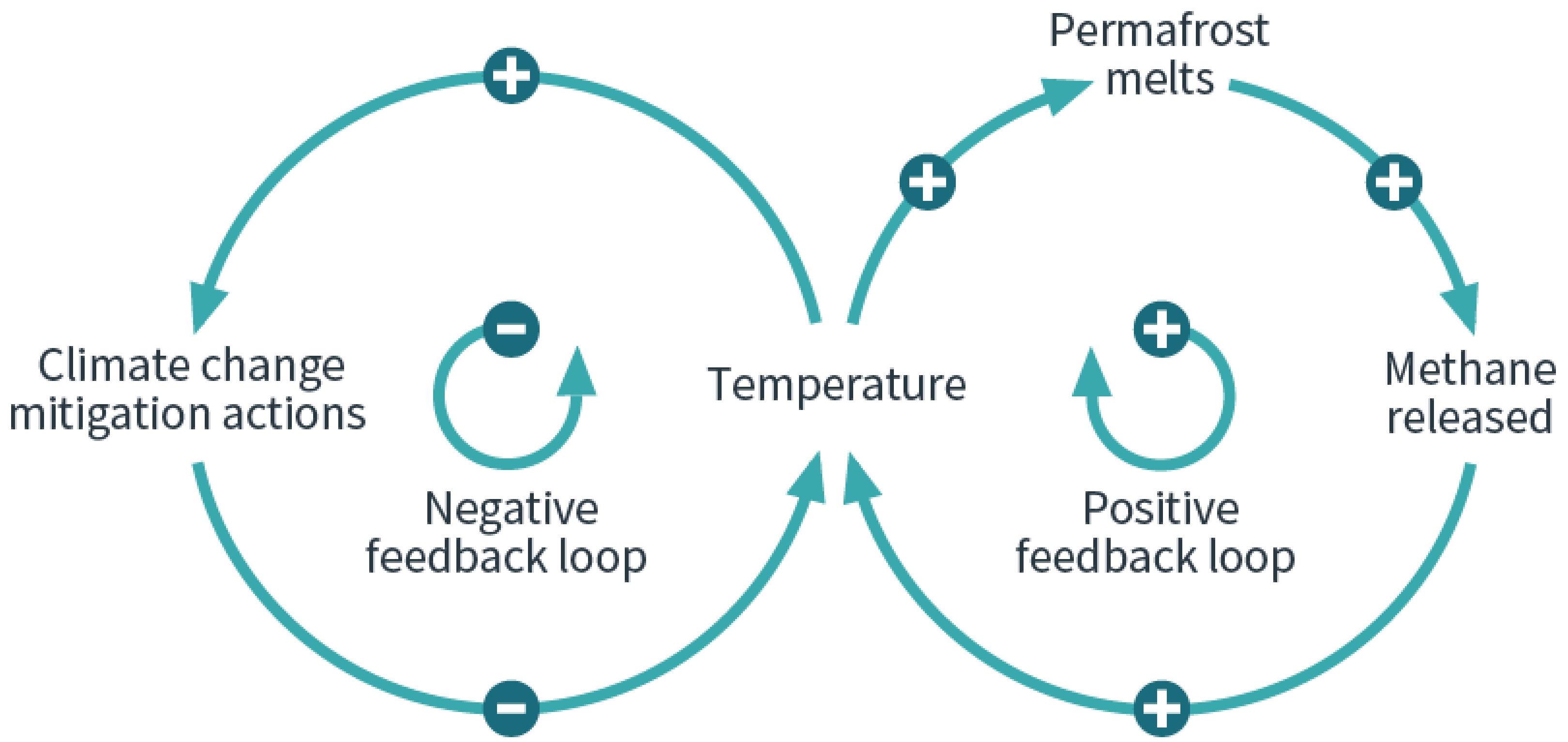
STOCK AND FLOW DIAGRAM



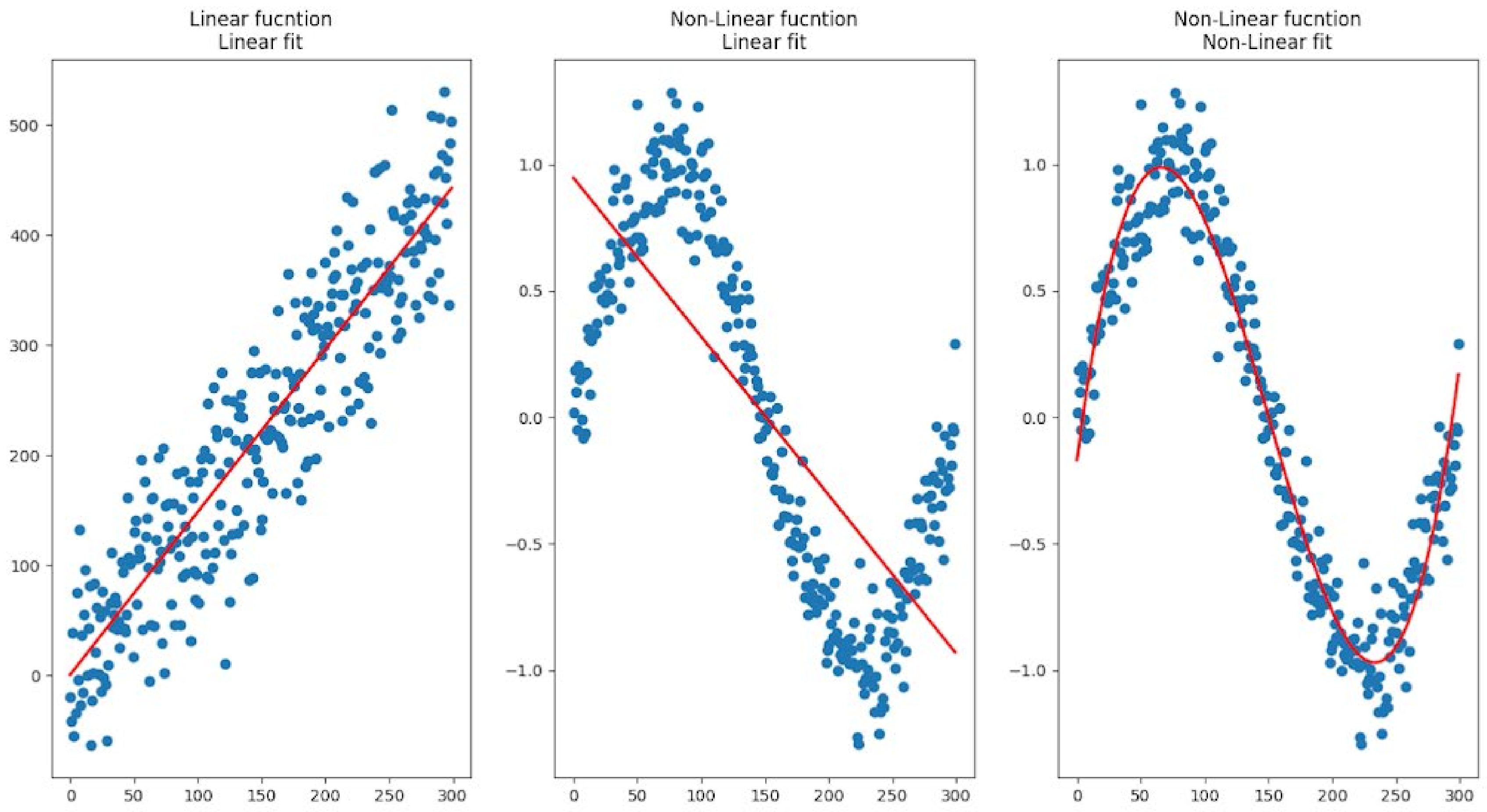
RICH PICTURE


Ensure the quality of the conference proceedings; Prepare itinerary for the conference
Lists of accepted authors
Make appropriate invitations; Ensure a sufficient attendance
Program Committee
Register the letters of intent and the contributed papers
Register acceptance of invitations
Organizing Committee
Call for papers
Assign session chairs
Letters of intent
Manuscripts
Manuscripts
Review reports
Acceptance
Invitation
Acceptance
Priority invitations
Priority invitations
Acceptance
Authors
Reviewers
Is the conference worth attending?
Members of related working groups
National Representatives
Have my paper accepted by the conference
Only accept high-quality papers
Is the conference worth attending?

FEEDBACK LOOPS



LINEAR VS NON-LINEAR



TIPPING POINTS

- A tipping point
  - The critical threshold at which a small change in one part of a system causes a sudden and often irreversible shift in the system’s overall state or behaviour
- In complex systems
  - Feedback loops, inter-dependencies, and accumulated pressures can make a system appear stable right up until it reaches a tipping point
  - Beyond that threshold, the system rapidly reorganizes itself into a new pattern, often with disruptive consequences
    - Often seen in ecosystems, economies, and IT infrastructures
  - Basically a tipping point is where “more of the same” suddenly becomes “no longer the same”

TIPPING POINTS - KEY CHARACTERISTICS

- Non-linearity
  - Cause and effect are not proportional
  - Incremental stress builds quietly, but the response suddenly accelerates once the threshold is crossed
- Feedback-driven change
  - Reinforcing feedback loops amplify disturbances once they start
    - For example, error impacts workload performance that produces more error resulting in a loss of control
- Irreversibility or hysteresis
  - Returning conditions to their pre-tipping-point state doesn’t always restore the system
  - New equilibria may form
    - For example: being locked into a crash recovery loop

TIPPING POINTS - KEY CHARACTERISTICS

- Sensitivity to small triggers
  - Minor perturbations like a delayed patch, missed communication can tip a system already under hidden stress into collapse
- Early-warning signals
  - Systems nearing tipping points often show growing volatility, slower recovery from small disturbances, or accumulating backlogs
  - Each warning signal on its own may not be enough to trigger a control
  - But when these signals together may be symptomatic of an emerging system instability

TIPPING POINTS - EXAMPLES

- Infrastructure overload
  - A system's performance declines gradually as load increases
  - But beyond a tipping point, latency spikes, processes hang, and cascading service outages occur
- Operational fatigue
  - Staff under constant high workload start making small mistakes
  - Error rates build until quality collapses
  - A human-process tipping point.
- Security risk escalation
  - Patch delays accumulate
  - Dependency vulnerabilities multiply
  - One exploit triggers a chain reaction that brings systems offline
- Cultural or governance shifts
  - Persistent blame culture leads to suppressed reporting until a major incident forces a governance overhaul

TIPPING POINTS - RELEVANCE TO RISK

- Monitoring leading indicators is as important as tracking lagging events
  - Traditional analysis assumes proportional change
  - Systems thinking reveals that risk often behaves non-linearly
  - Understanding tipping behaviour helps detect when an environment is approaching instability before catastrophic change occurs
- Early detection allows for preventive resilience
  - Adjusting load, redistributing responsibilities, or introducing dampening feedbacks before the threshold is crossed

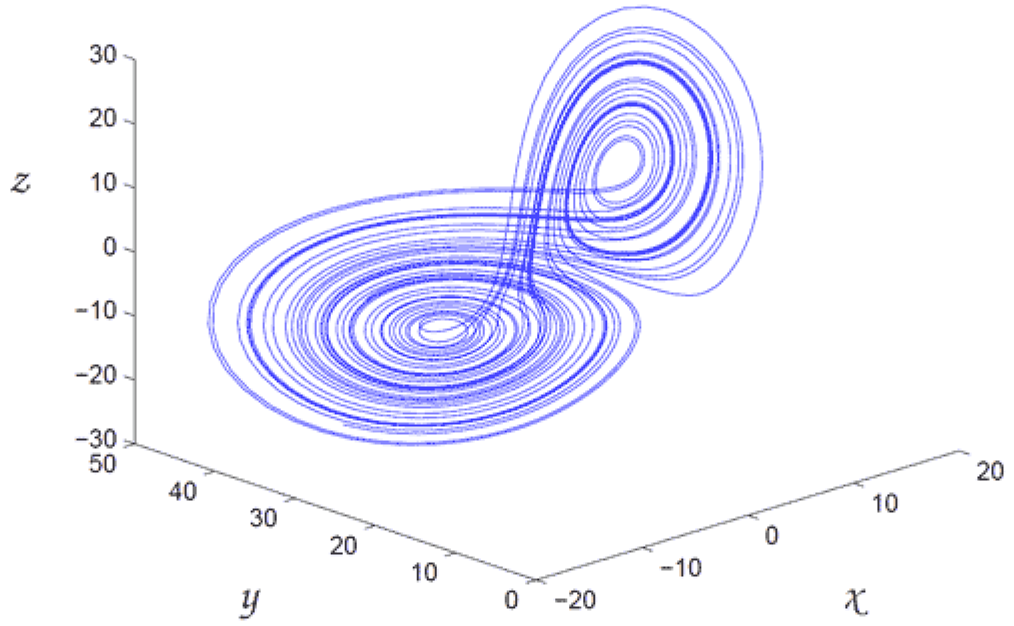
MITIGATION AND DETECTION STRATEGIES

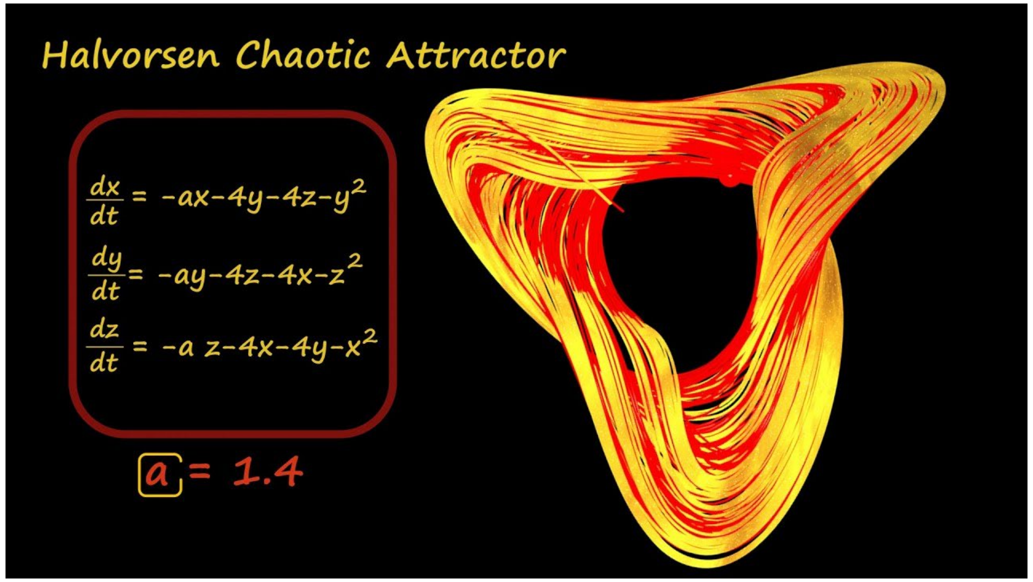
- Identify critical thresholds
  - Use metrics (e.g., latency, backlog size, error rates, turnover) to define points where performance sharply degrades
- Monitor leading indicators
  - Look for signals such as increasing variability, repeated near-misses, or slower recovery from small disruptions
- Build buffers and redundancy
  - Introduce spare capacity or modular design so that stress is absorbed gradually rather than accumulating.
- Strengthen feedback controls
  - Implement mechanisms that balance reinforcing loops
    - For example: automatic throttling, escalation alerts, adaptive load balancing.
- Encourage transparent reporting
  - A culture that surfaces weak signals early prevents invisible accumulation of risk pressure

MITIGATION AND DETECTION STRATEGIES

- Tipping points
  - Are the dynamic consequence of feedback loops, delays, and interdependencies
  - When mapping causal loops, a tipping point is often where a reinforcing feedback loop overwhelms balancing controls
  - By identifying and modelling these loops, risk teams can anticipate thresholds and design interventions that restore balance before irreversible change occurs
- The formal study of tipping points is in the mathematical domain called chaos theory
  - Studies that some systems shift into to configurations at tipping points
  - The configurations are called attractors
  - An IT system can go from stable to a number of unstable states
  - The effort required to revert back to a stable state can be very high

ATTRACTORS





$$\frac{dx}{dt} = -ax-4y-4z-y^2$$
$$\frac{dy}{dt} = -ay-4z-4x-z^2$$
$$\frac{dz}{dt} = -a\,z-4x-4y-x^2$$
$$\boxed{a} = 1.4$$

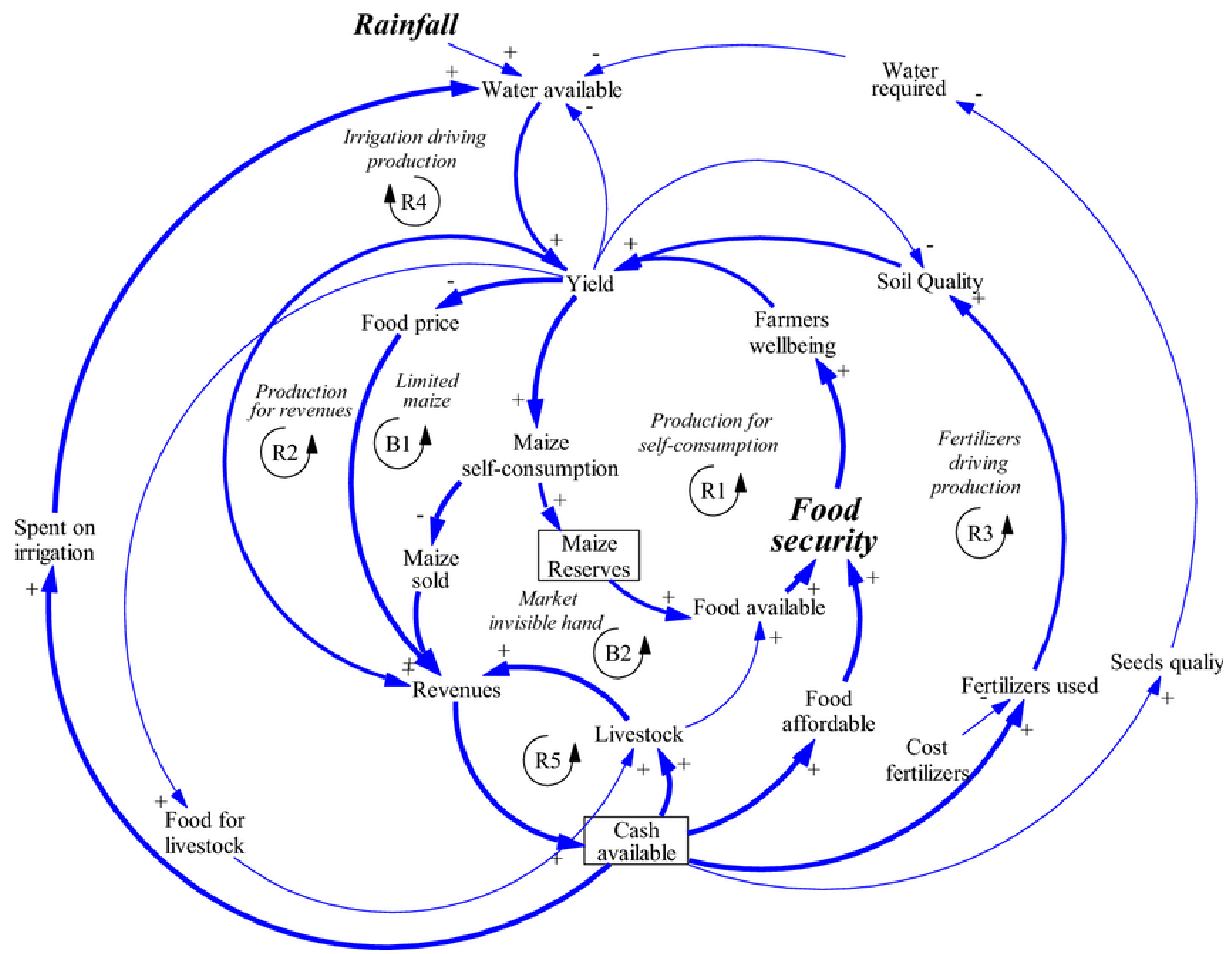
CASCADING FAILURES

- A cascading failure
  - Occurs when a failure or disruption in one component of a system triggers failures in other components
  - Often in sequence or via feedback loops, resulting in a large-scale systemic disruption.
  - A small error or delay in one process or team can propagate through downstream processes, across systems, or between teams
  - Effects are amplified by hidden dependencies or lack of redundancy
  - Example:
    - A key system goes offline
    - Manual workaround puts load onto a team
    - That team delays
    - Downstream system receives incorrect input
    - Compliance control is bypassed
    - Regulatory breach occurs.

HIDDEN DEPENDENCIES

- Dependencies are often not visible in standard process maps
  - For example, Team A’s output becomes system input for Team B, but the dependency is informal/un-documented
- Hidden dependencies increase risk because
  - They create single points of failure (if Team A fails, Team B can't proceed)
  - They produce tight coupling (the downstream team or system has no buffer)
  - They reduce visibility and monitoring (failure upstream may not be flagged until it impacts downstream)
  - Systems thinking identifies these by asking
    - Which processes/systems/teams are reliant on another?
    - What would happen if that upstream function failed or was degraded?

CAUSAL LOOP DIAGRAM



CAUSAL LOOP DIAGRAM

- Purpose and overview
  - It helps visualize how different variables in a system interact and influence one another
  - Often through feedback loops that either reinforce (amplify) or balance (stabilize) change
  - CLDs shows how causes and effects circulate through a process
  - This reveals hidden dependencies, feedbacks, and potential tipping points
  - Instead of showing just a linear "A causes B," a CLD shows networks of cause-and-effect relationships that evolve over time

KEY CONCEPTS

- Variables
  - Each node in the diagram represents a variable representing something that can increase or decrease over time
  - Examples
    - System Load
    - Team Workload
    - Error Rate
    - Backlog Size
    - Control Effectiveness
  - Each variable is connected by arrows that indicate causal influence

KEY CONCEPTS

- Causal links (arrows)
  - Arrows show direction of influence, how one variable affects another
  - Each arrow has a polarity, marked with a "+" or "–" sign
  - + (same direction)
    - When A increases, B increases
    - When A decreases, B decreases
    - For example: Workload (+) → Stress: means that workload increases stress
  - – (opposite direction)
    - When A increases, B decreases
    - When A decreases, B increases
    - For example: "Control Effectiveness (–) → Error Rate" means that better controls reduce errors.
  - Polarity does not mean good or bad, it just describes the direction of the effect

KEY CONCEPTS

- Feedback loops
  - A closed chain of causal links where a change in one variable eventually feeds back to influence itself
  - There are two major types
- Reinforcing (positive) feedback loop
  - Also called an R loop, this creates growth or amplification
  - Small changes compound over time, leading to exponential growth or runaway effects
  - Example:
    - More workload → More errors → More rework → More workload
  - Even if it starts small, the loop reinforces itself and grows until something breaks or stabilizes it or until it reaches a tipping point
    - Often labelled with R (or sometimes "+") in the loop symbol.
    - Or a small circle with the letter R at the loop center or end

KEY CONCEPTS

- Balancing (negative) feedback loop
  - Also called a B loop, this creates stability or regulation
  - Changes are counteracted over time, pushing the system back toward a target, goal, or equilibrium
    - Rather than compounding, effects dampen the original change
  - Example:
    - More workload → More prioritization → Fewer tasks accepted → Reduced workload
  - If a change occurs, the loop works to limit growth or decline, keeping the system within acceptable bounds
    - Often labelled with B (or sometimes "–") in the loop symbol
    - Or a small circle with the letter B at the loop center or end
  - They often include delays, which can cause oscillation or overshoot if not well managed

KEY CONCEPTS

- Time delays
  - Real systems rarely react instantly
  - To show this, a delay is marked with a small double slash “//” across a causal arrow
    - For example: “Incident Occurs → // → Management Awareness”
  - This means it takes time for information or consequences to propagate through the system, a crucial source of instability and surprise

HOW TO READ A CLD

- To interpret a causal loop diagram
  - Start with one variable and follow the arrows
  - Track the direction (same “+” or opposite “–”) of each causal link
  - Continue following until you return to the starting variable, that’s a loop.
    - Count the number of negative (–) signs in the loop
    - Even number → Reinforcing Loop (R)
    - Odd number → Balancing Loop (B)
- Ask
  - Does the loop amplify change (R) or dampen it (B)?
  - Are there delays that could make reactions too slow or too strong?
  - What risks or tipping points might arise if this loop becomes dominant?

EXAMPLE

- Scenario: incident backlog loop
    - Workload (+) → Stress (+) → Errors (+) → Rework (+) → Workload
    - This is a Reinforcing Loop (R1)
    - More work leads to more errors, which increases rework, which further increases workload
- To balance it, you might add a control
    - Errors (+) → Management Oversight (–) → Errors
    - Creates a Balancing Loop (B1)
    - Increased oversight reduces errors, stabilizing the system
    - If the reinforcing loop grows faster than the balancing loop can react (especially if there's a delay in oversight)
    - Then a tipping point may occur where the process collapses under workload stress

Q&A AND OPEN DISCUSSION