Strategic IT management - 37E00200

Discontinuance of IT/IS

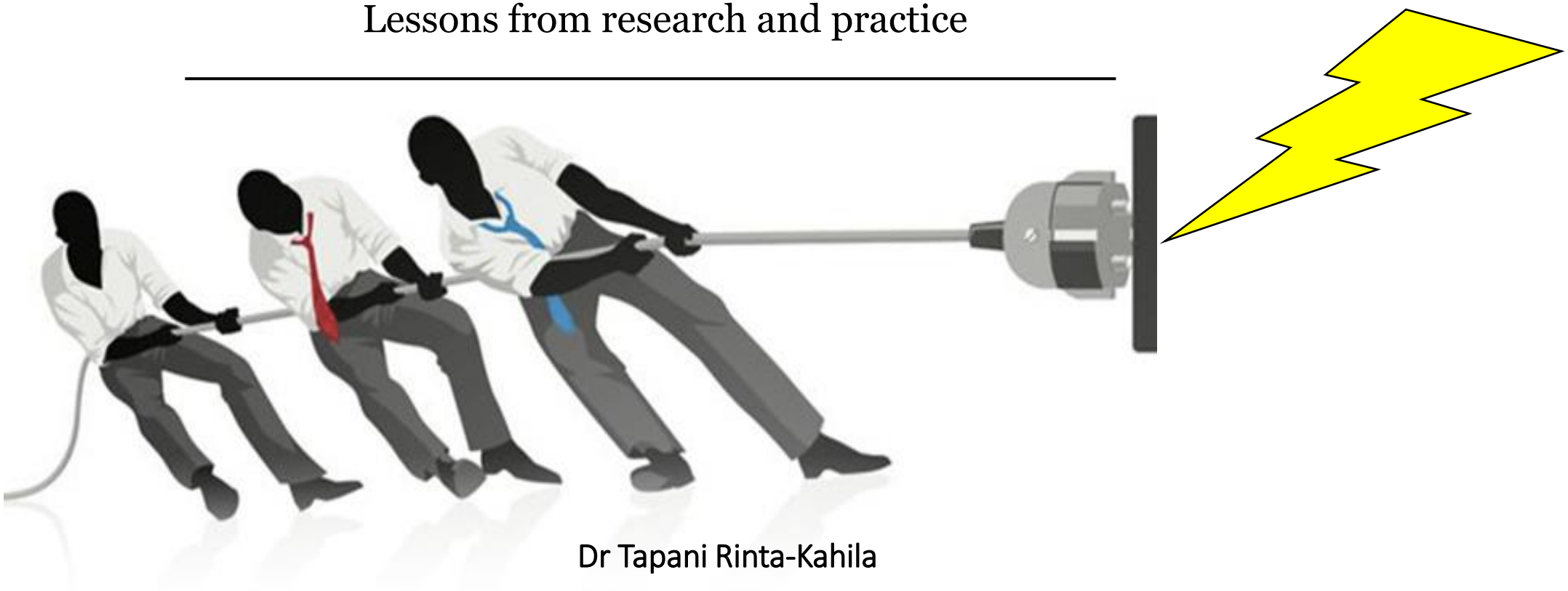
Dr Tapani Rinta-Kahila

Lecturer

The University of Queensland

Organizational Information System Discontinuance

Lessons from research and practice



Dr Tapani Rinta-Kahila

Lecturer of Business Information Systems

The University of Queensland

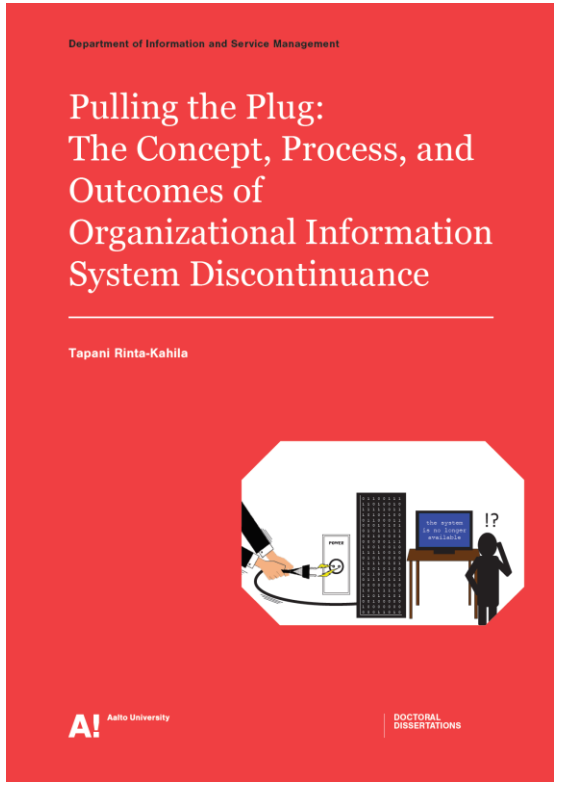
About me

Background at Aalto BIZ

- B.Sc. in Business Technology in 2011
- M.Sc. in Information and Service Management in 2013
- D.Sc. in Information Systems Science in 2018

Research interests

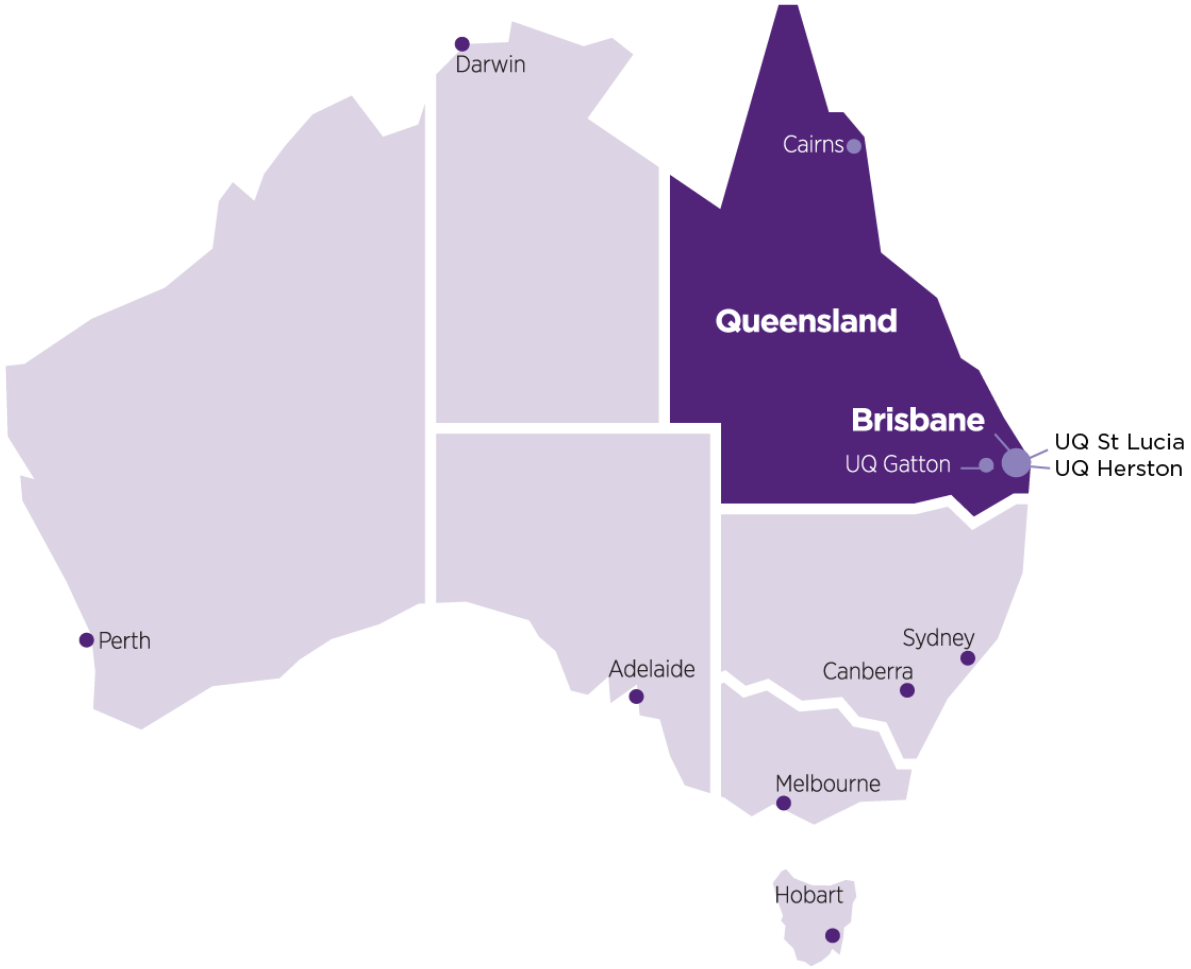
- IS/IT discontinuance
- Automation & AI in organizations
- Unintended consequences of technology, esp. algorithmic decision-making



THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

CREATE CHANGE

Where we are



3
campuses

6
faculties

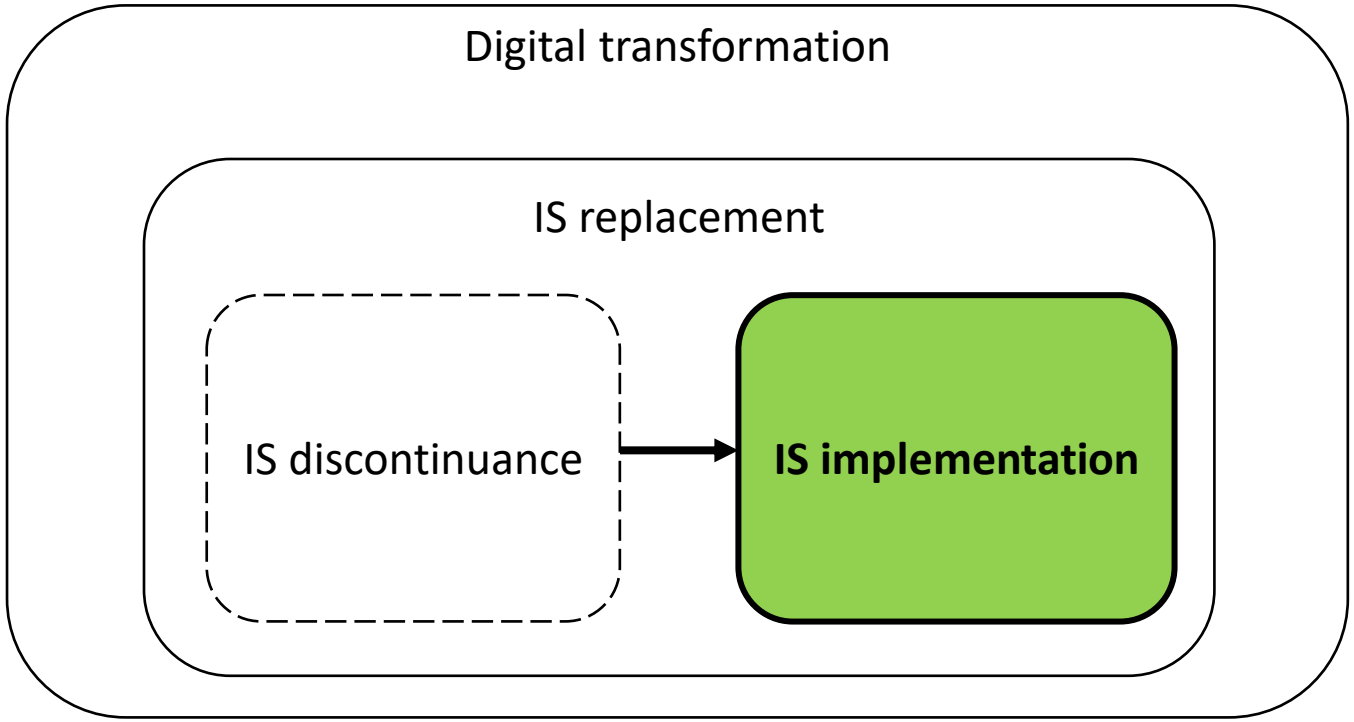
30+
teaching and research sites

8
research institutes

IS discontinuance: Replacing legacy systems

Literature on organisational IS

The focus has been on new IS implementations

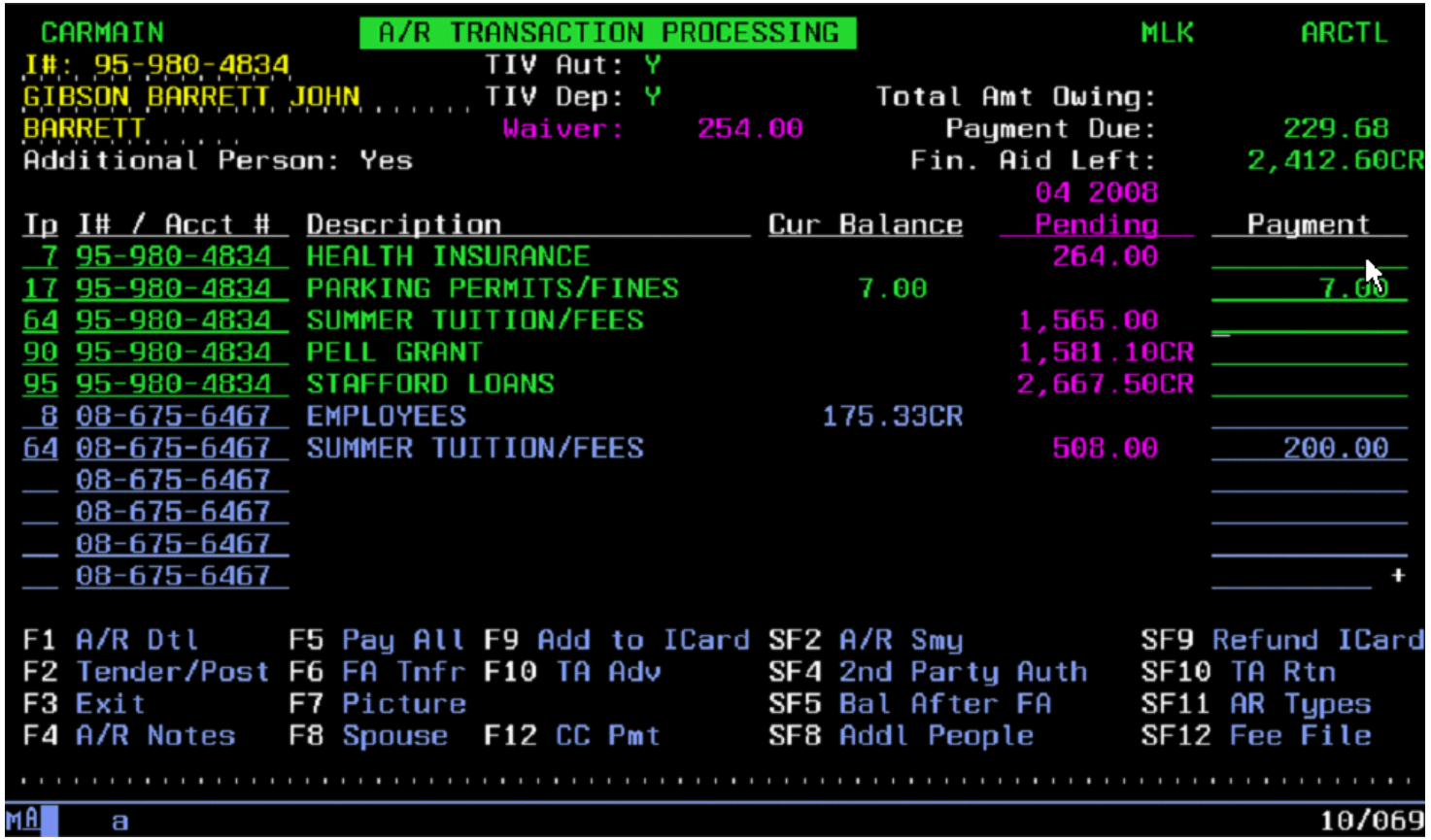


IS/IT discontinuance processes can turn out difficult in organizations…

…and they may have unexpected consequences.

Legacy systems: definition

- Information systems in place that embody the organization's business model from the time of implementation (Kelly et al. 1999)



Legacy systems: definition

- Typical characteristics
  - Old
  - Large
  - Self-made
  - Based on old or outdated coding languages
  - Run business-critical operations
  - Run in the backend

- The ”backbone” of an organization

Evolutionary design

**"I am not disorganized — I know *exactly* where everything is!
The newer stuff is on top and the older stuff is on the bottom."**

Typically developed introspectively → siloed

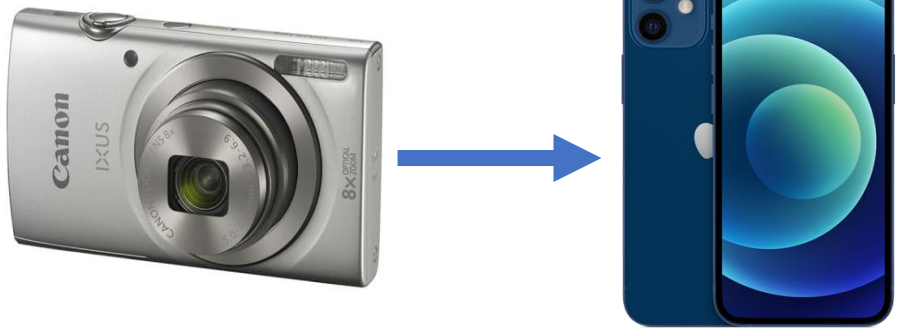
The weight of history

- Legacy systems represent “an established pattern of choices and actions that tends to continue due to historical commitments.” (Mehrizi et al. 2019, p. 144)

Issue 1: Technical obsolescence

- Old code deteriorates
- Loss of expertise – people retire
- Decreasing system support
  - Security risks

Issue 2: Functional obsolescence



- Better systems have emerged
  - Social, Mobile, Analytics, Cloud, and Internet of Things (SMACIT)
- The environment has changed
  1. Globalization
  2. Transformation of industrial societies and economies into knowledge and information-based economies
  3. Transformation of the business enterprise away from a hierachical, centralized structure toward flattened (less hierarchical) and decentralised (matrix organizations)

→ The system is no longer sufficient or relevant

POLICY

UNEMPLOYMENT CHECKS ARE BEING HELD UP BY A CODING LANGUAGE ALMOST NOBODY KNOWS

States have been starved of modernization funding for years

By Makena Kelly | @kellymakena | Apr 14, 2020, 8:00am EDT

Illustration by Alex Castro

"MODERNIZATION OF MAINFRAME COBOL IS LIKE HOPPING OFF OF YOUR BICYCLE AND JUMPING ONTO A HARLEY DAVIDSON MOTORCYCLE"

https://www.theverge.com/2020/4/14/21219561/coronavirus-pandemic-unemployment-systems-cobol-legacy-software-infrastructure

What can be done with legacy systems?

Easy

- Wrapping
  - Refresh with a modern interface for easy accessibility and use
  - No change to the underlying systems

- Redeveloping
  - Expand, rewrite, bring more modern functionalities

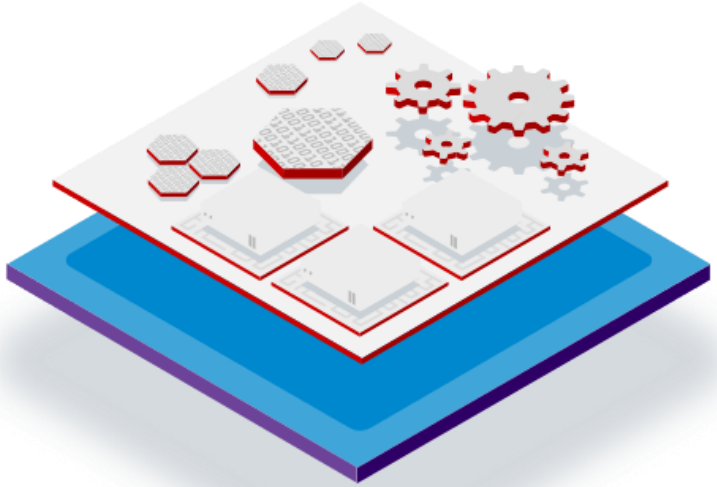
- Pulling the plug
  - Replace with a new system
  - Often considered desirable but is difficult

Hard

Organizations are replacing their legacy systems with commercial-off-the-shelf (COTS) systems

COTS systems provide “digital options”

- A digital platform - offers access to apps, products, and services produced by actors in the platform ecosystem or available in the organization’s infrastructure
- Digital options represent “the transformative capabilities that digital platform evolution affords a user organization” (Rolland et al. 2018)

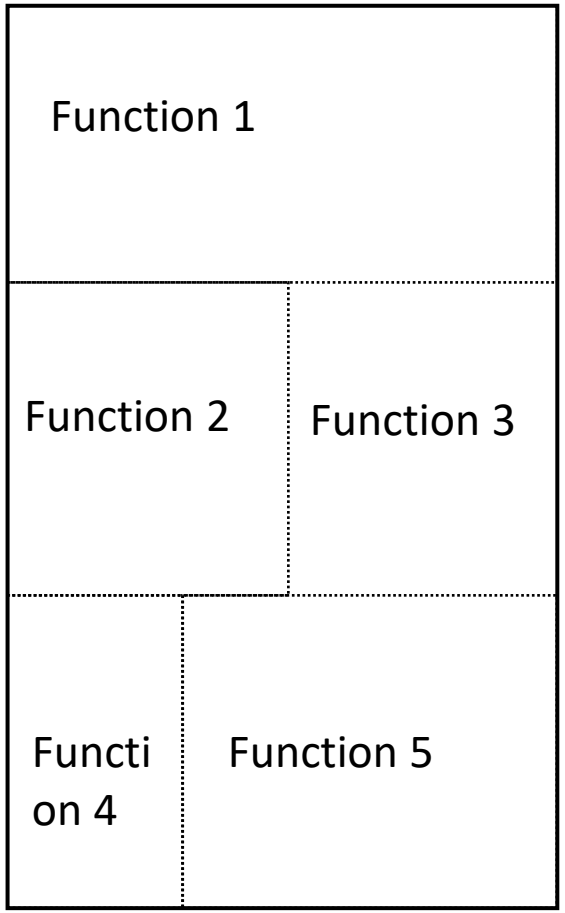


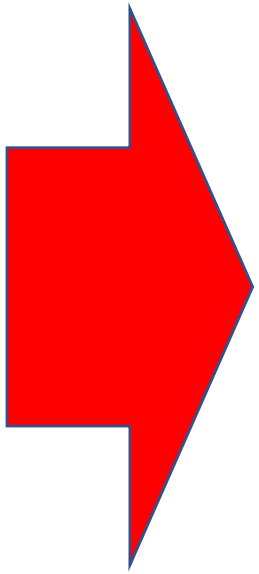
Especially in global, multi-site organizations, COTS systems decrease **technical inertia**
(i.e., rigidity caused by technology)

- Harmonized systems across different operational sites
- Externalized system maintenance

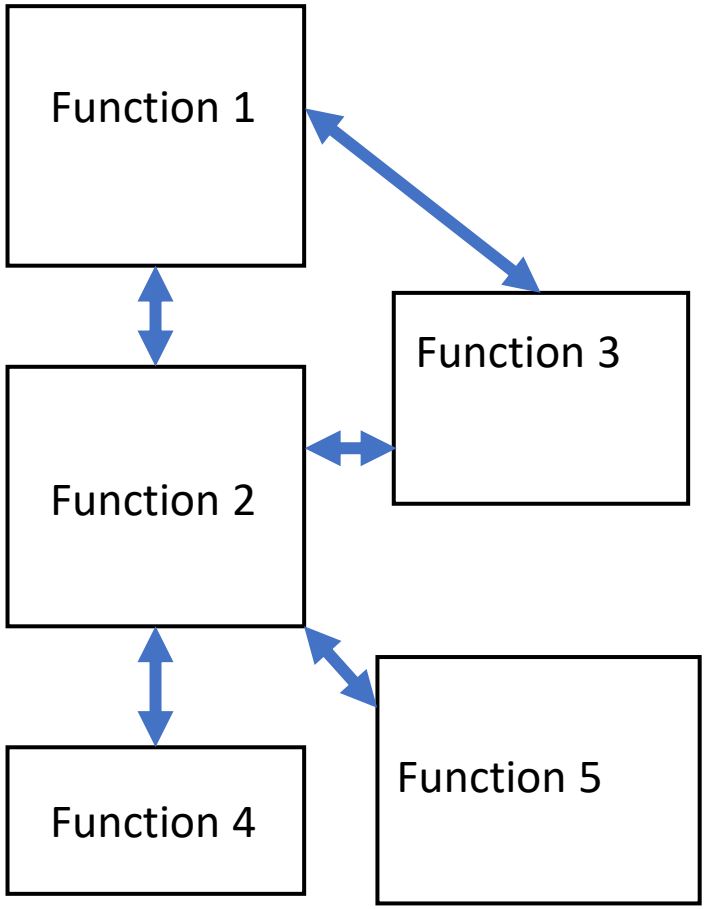
Tightly coupled vs. loosely coupled system

Legacy architecture -
monolithic





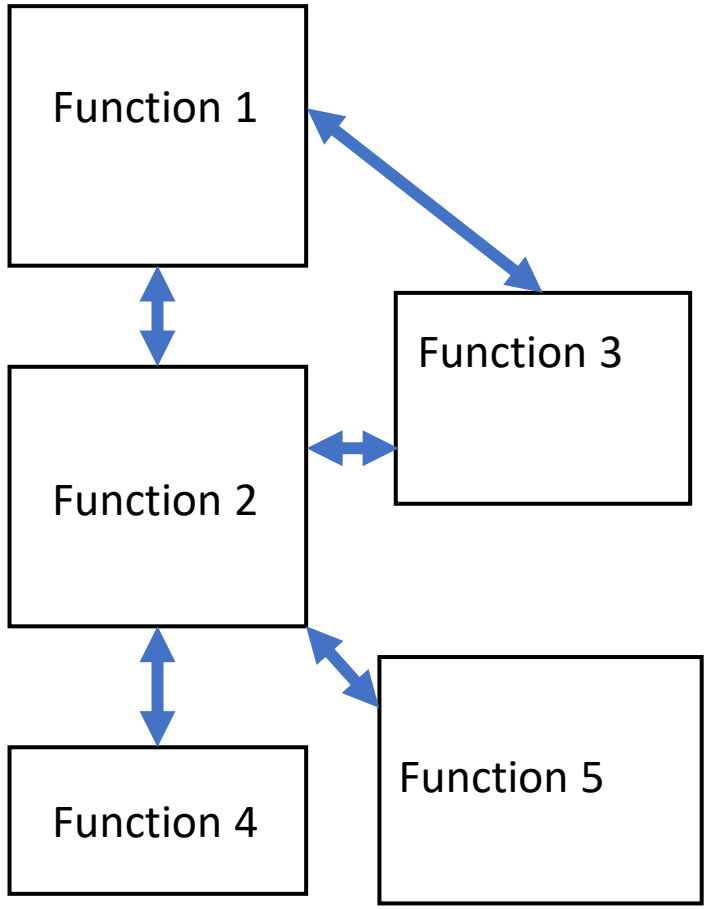
Service-oriented architecture -
modular



From tightly coupled to loosely coupled system architecture

- Gain agility
- Able to realize new business opportunities

Service-oriented architecture - *modular*



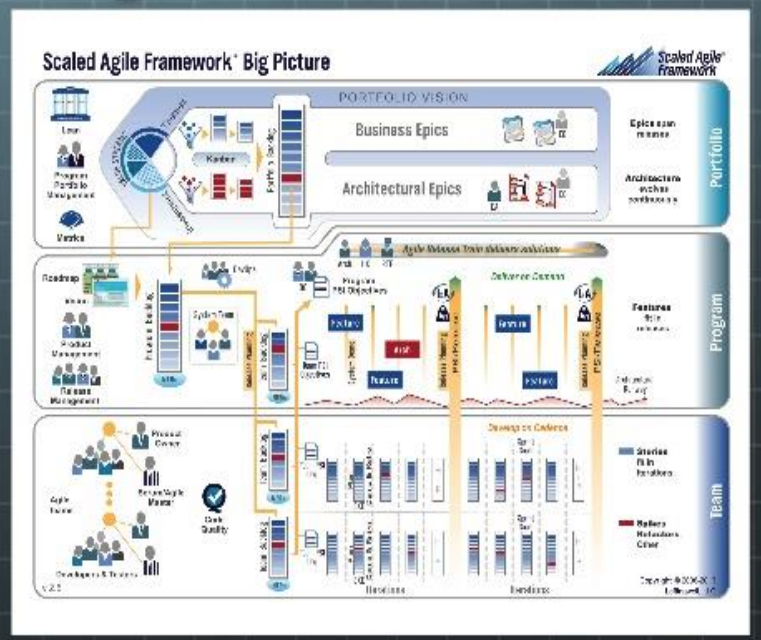
Stuck with legacy systems – why does this happen?

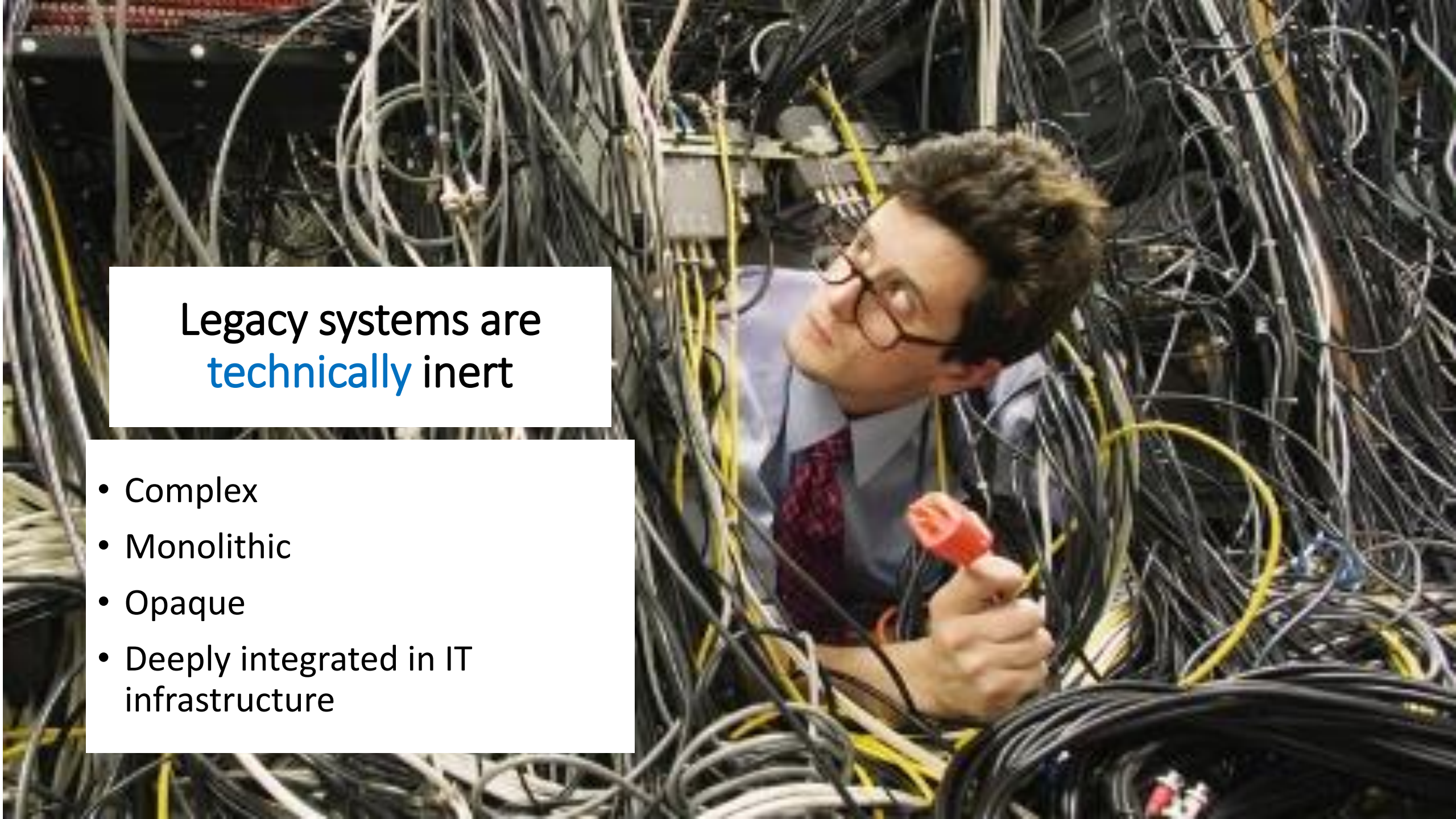
Monolithic legacy system

Organization

Inertia!

Modular and agile new IS architecture





Legacy systems are technically inert

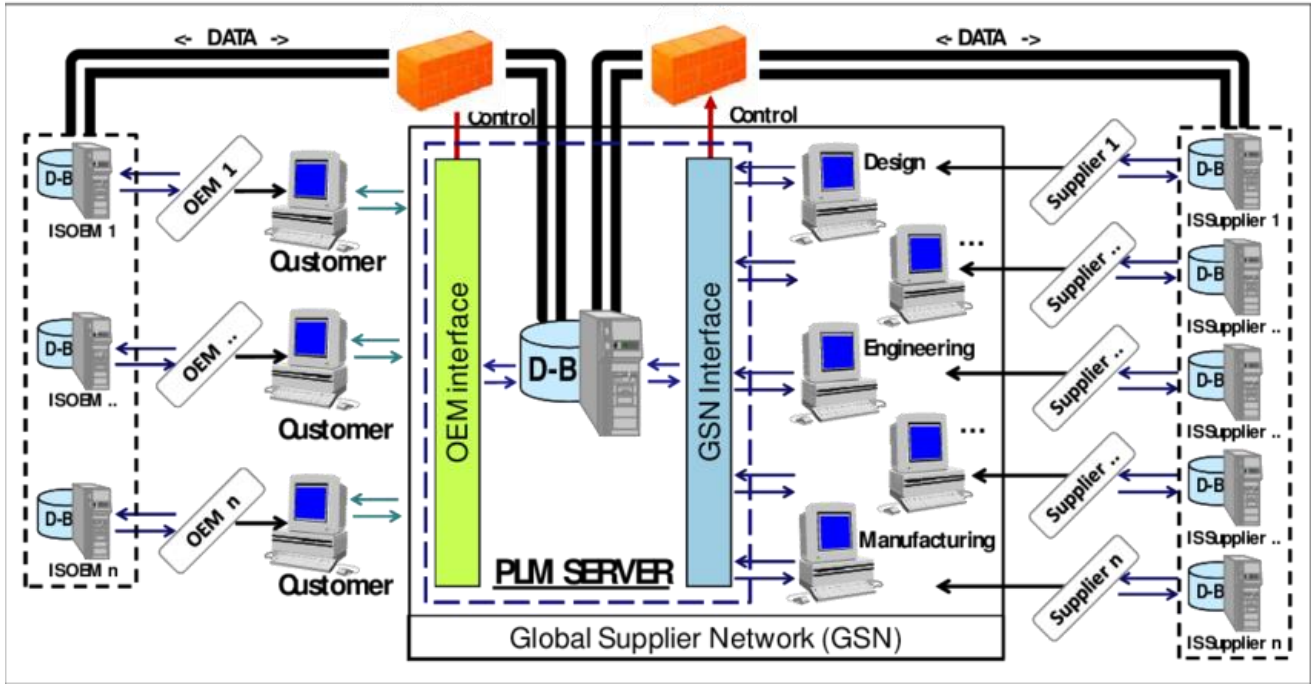
- Complex
- Monolithic
- Opaque
- Deeply integrated in IT infrastructure

Technical debt – a measure of technical inertia

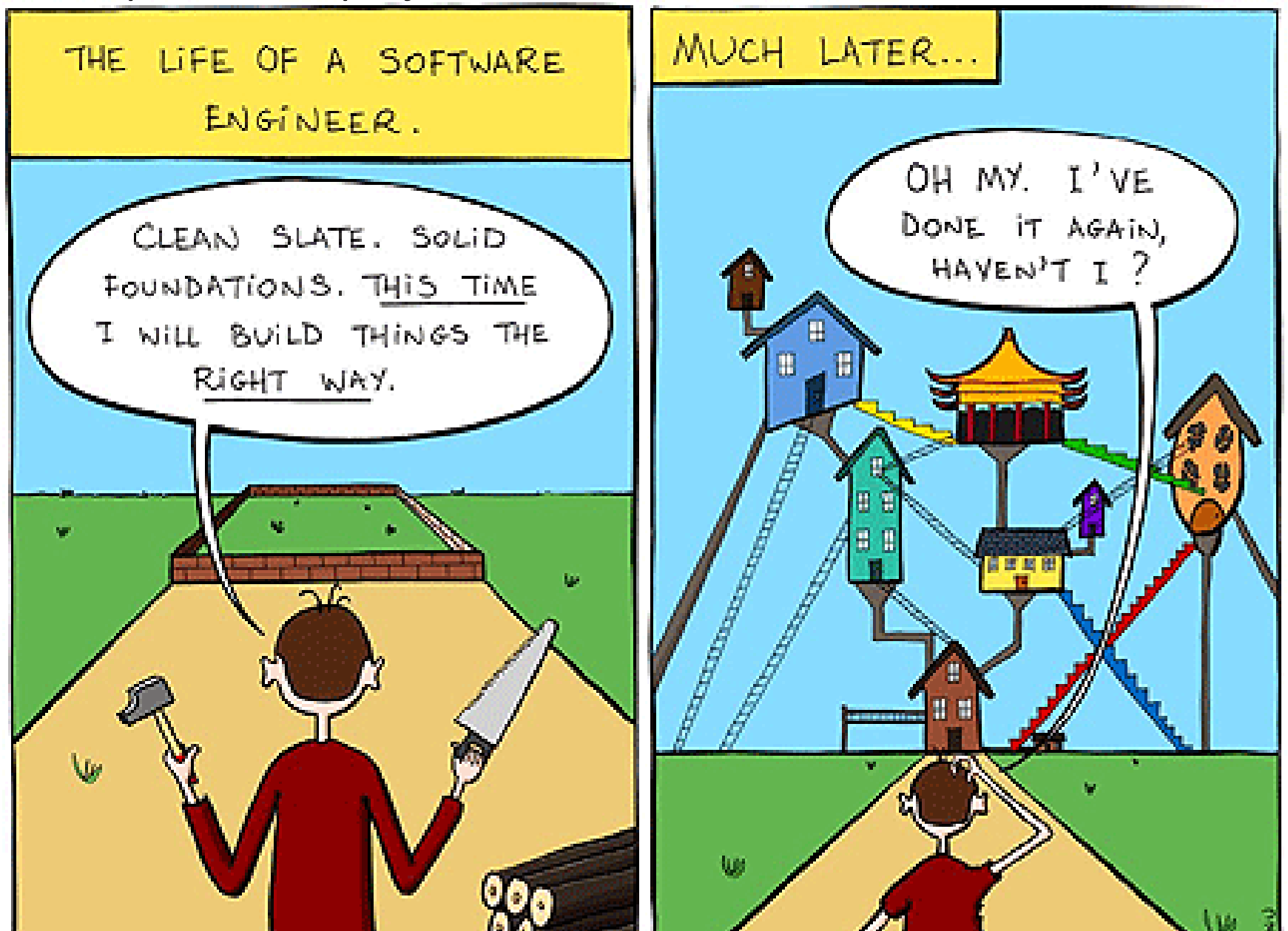
- IT maintenance obligations that render making technical changes difficult and risky
- Often a result of evolutionary or careless development practices (cutting corners)
- Manifests at different levels
- Resolving technical debt:
  - Rewriting code
  - Producing documentation
  - Replacing old components
  - Optimizing system architecture
- If technical debt is not repaid, it can accumulate 'interest', making it harder to implement changes later on
- Increases **software entropy**, i.e., the IT architecture becomes increasingly disorganized and unstable

Architectural technical debt

- Unsystematic dependencies, violation of modularity, technological gaps
- Causes complexity, maintenance costs, and difficulty of maintenance
- Give rise to other types of technical debt (e.g., when poor architectural solutions are left undocumented)



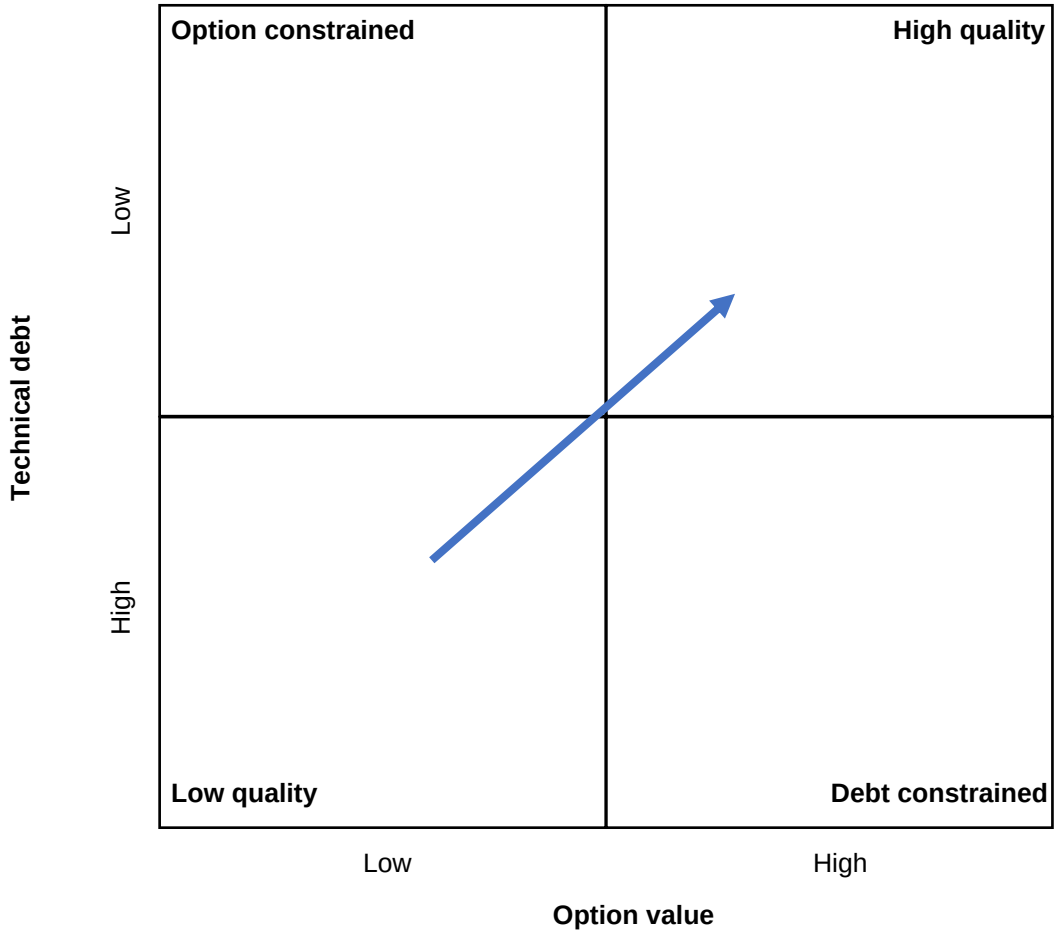
Architectural debt is especially relevant for legacy system replacement projects



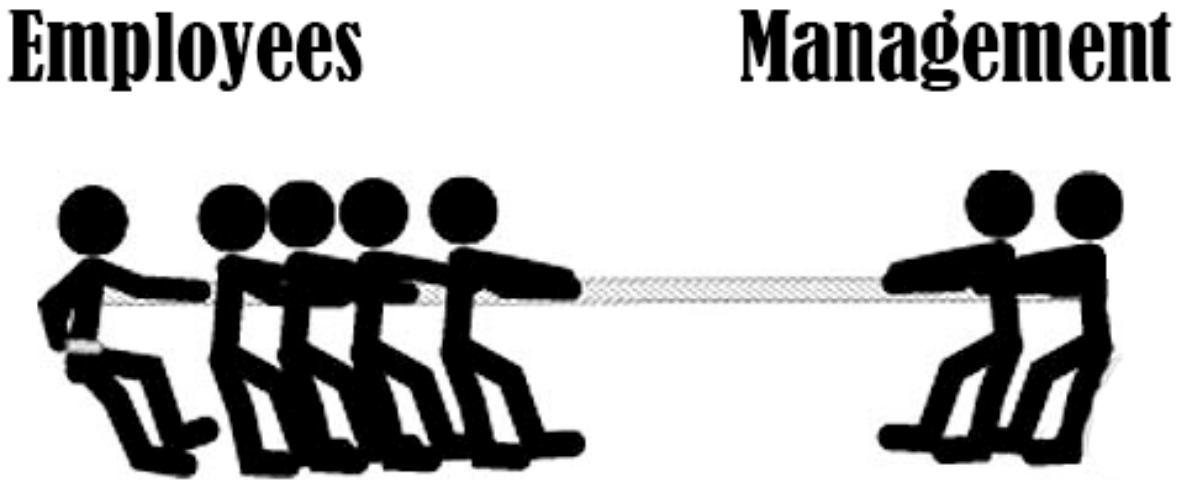
The dynamics between digital options and technical debt: Design moves (Woodard et al. 2013)

Technical Debt	Low	High
Low	**I: Option Constrained** Low debt, but few options to fuel innovation or development of complementary assets	**IV: High Quality** Low debt and many options; strongly positioned for innovation and platform leadership
High	**II: Low Quality** High debt and few options; weak position saps resources with little strategic benefit	**III: Debt Constrained** Many options, but high debt impairs the firm's ability to exploit them effectively

Option Value

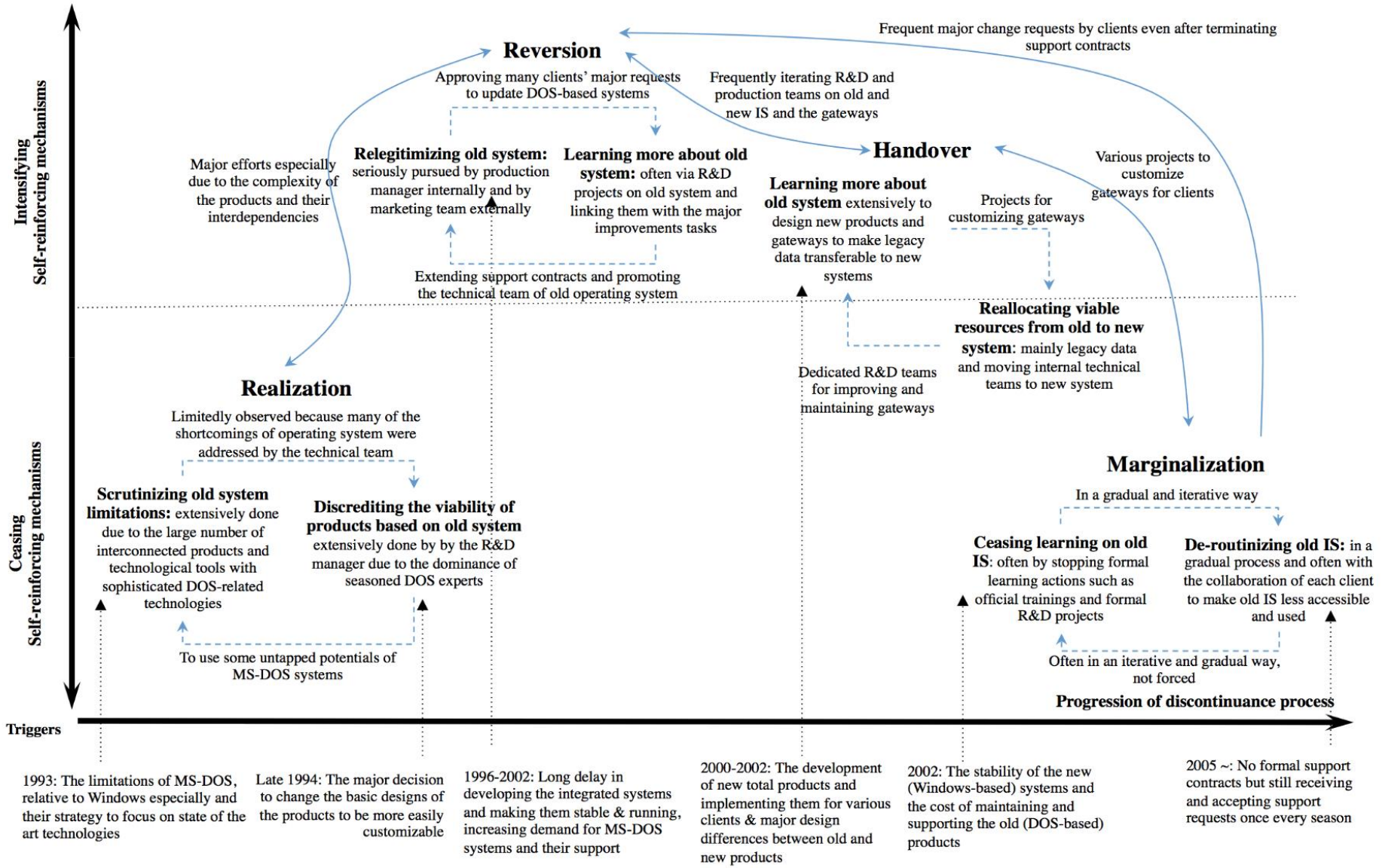


COTS implementation and legacy system discontinuance efforts face social inertia

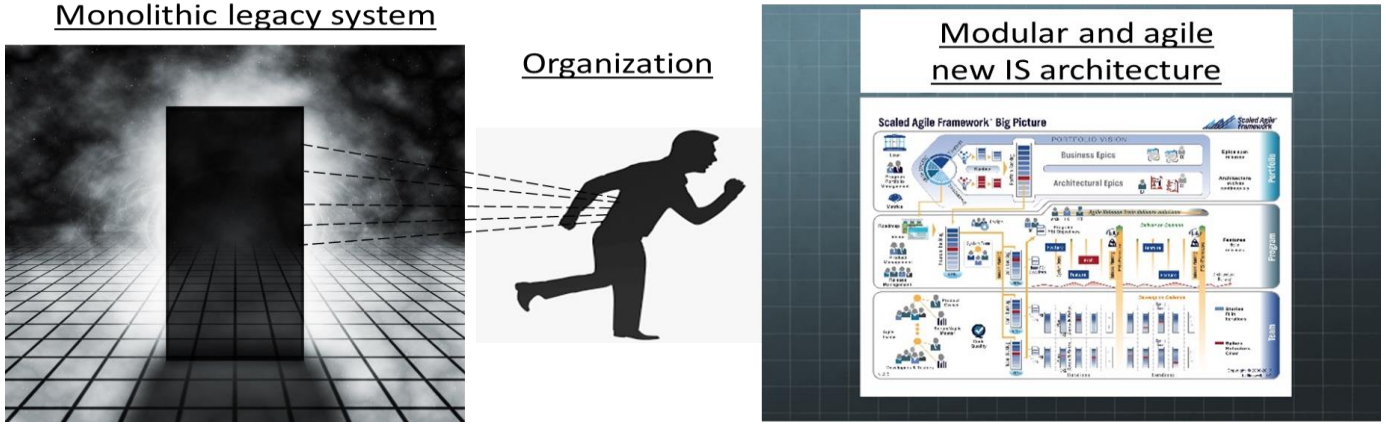


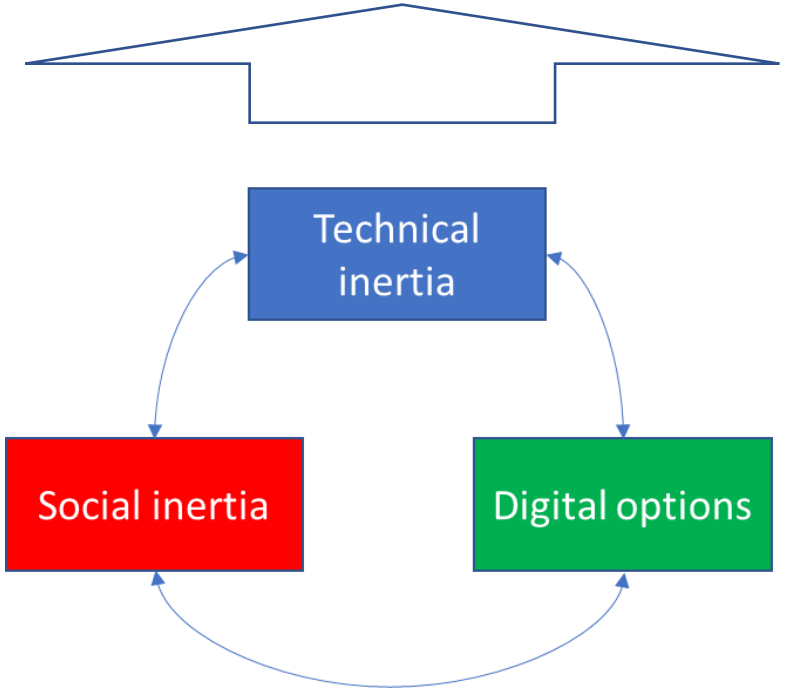
Overcoming legacy systems' social inertia

(Mehrizi et al. 2019)



Caught between – a product of technical inertia, social inertia, and digital options during a change process





Research questions

1. How do social inertia, technical inertia, and digital options interact in replacement of legacy systems with COTS ones (especially in organizations with multiple sites)?
2. How and under what conditions does the state of being caught between emerge and stabilize during this process?

Case EngineShop (Rinta-Kahila et al., 2023)

- A factory in Helsinki producing electrical engines, part of the multinational “EngineGroup” corporation
- A pioneer in system development
- The factory got caught between – has not been able to fully remove their legacy system called ”Driving Glove” (DG)

Legacy system: DG

“It was so tailored, fit-for-purpose, and made precisely for these operations. You could not find corresponding functionality from any commercial system or their combinations.”

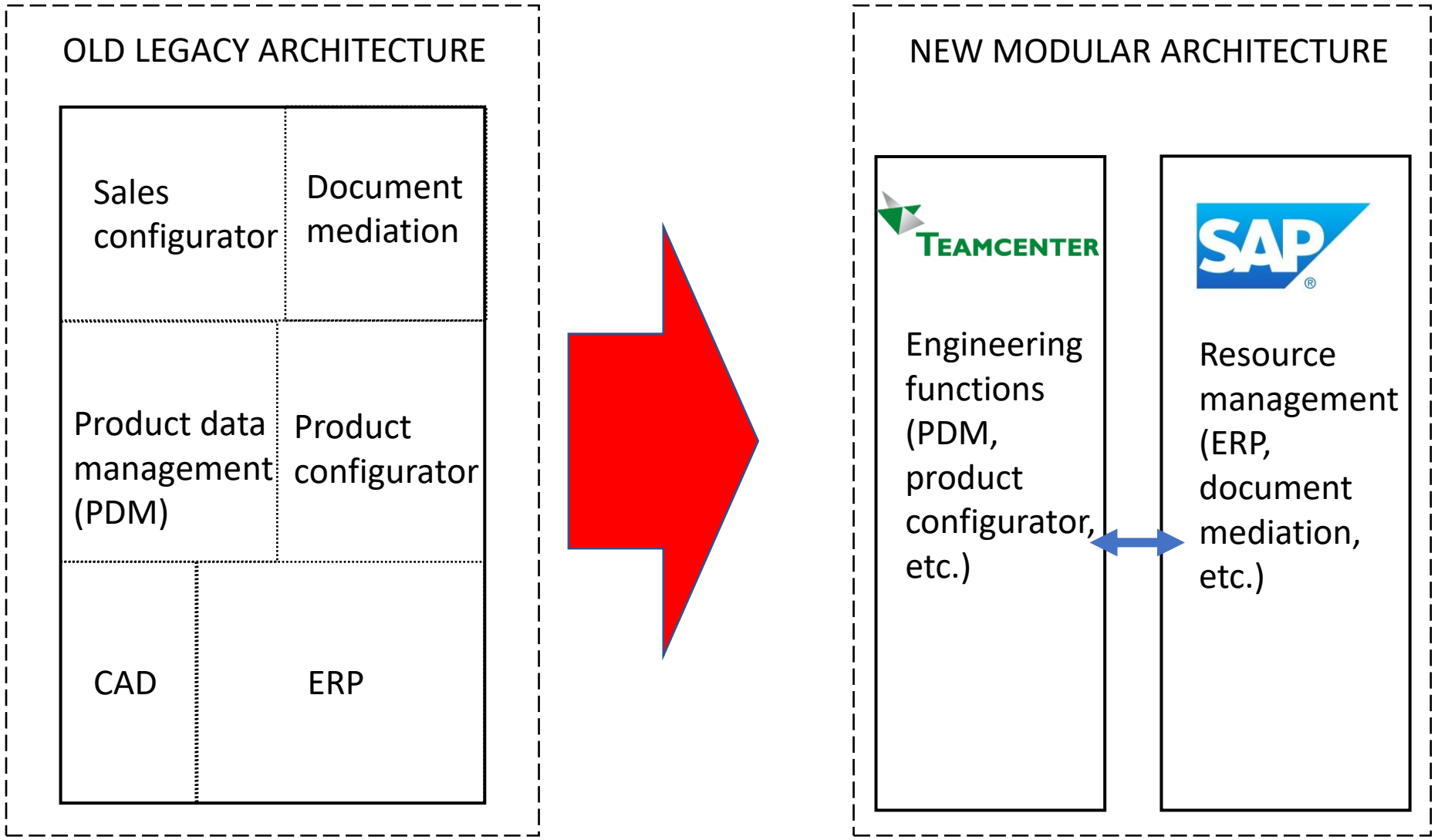
→ DG worked like a dream for the local purposes. However, it was getting old and the organization had adopted a global strategy which demanded **digital options:** need to be able to collaborate globally

Legacy system: DG

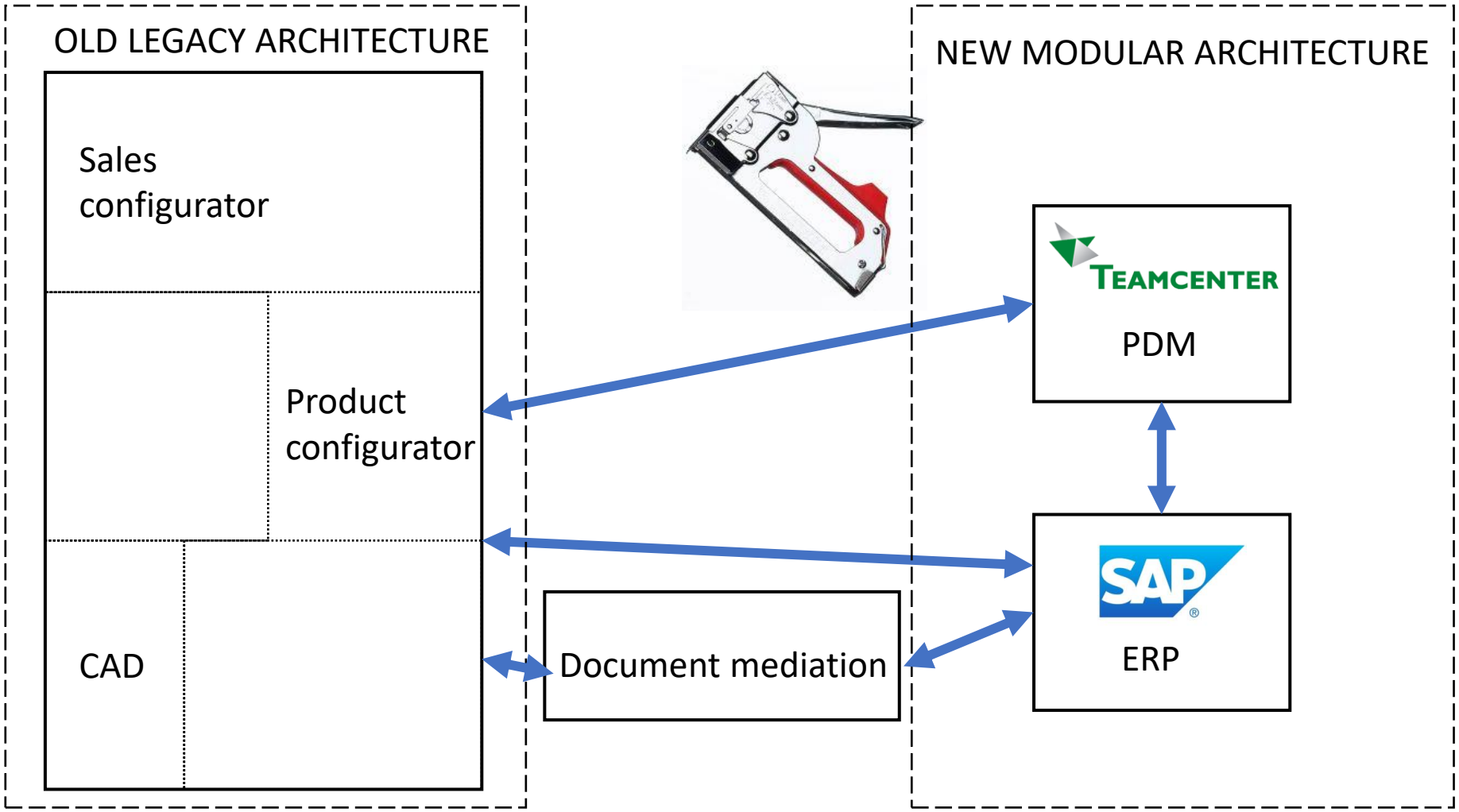
Development had been evolutionary/introspective, accumulating **technical debt**:

"...instead of rethinking our processes, it has been more about identifying inefficiencies and then developing the system to decrease the lead times. So, if an engineer does not have certain data and has to spend a long time to retrieve it, instead of thinking whether s/he even needs that data, we have just developed DG..."

2005: Initiative to move into modular IT architecture



2010: Only some business functions get migrated…and only partially



2021: The situation has not changed, but still intentions to get rid of the legacy system

Outcome of the implementation

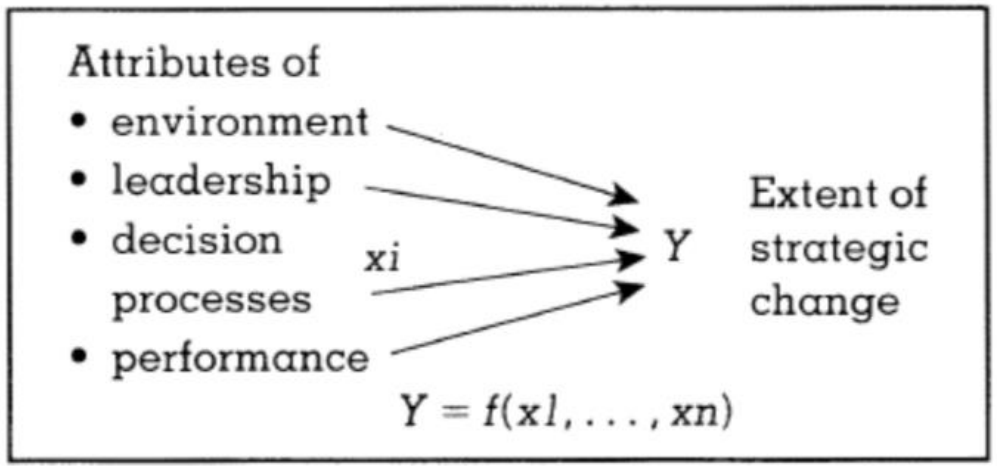
”…after engaging in the actual projects, we came to realize exactly how far thought the business logic of DG was. How hard it was to replace it with generic commercial systems. ... [now we are IS] *architecture wise pretty much in the same situation as we were after these projects in 2010.* ***Still****.”*

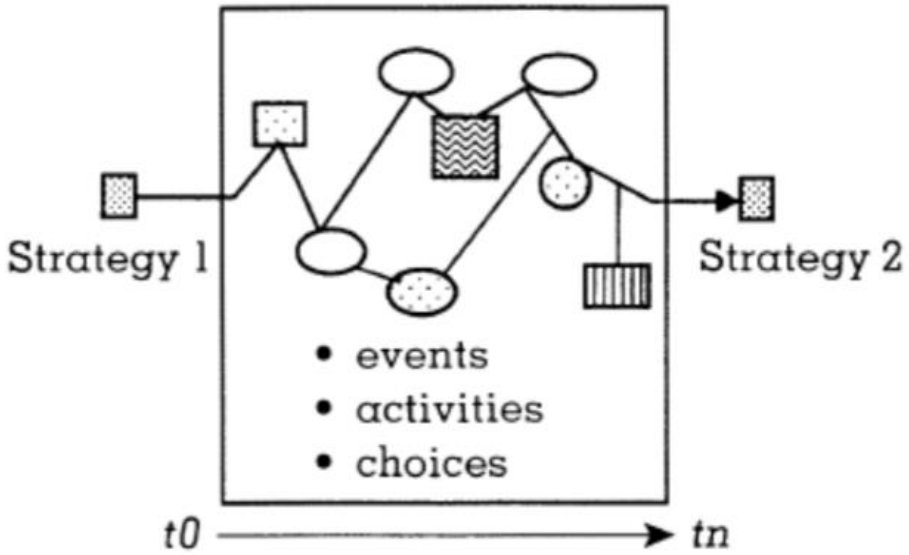
EngineShop accumulated more technical debt (especially architectural) but did not solve the problem of old systems.

How to study complex organizational change like legacy system discontinuance?

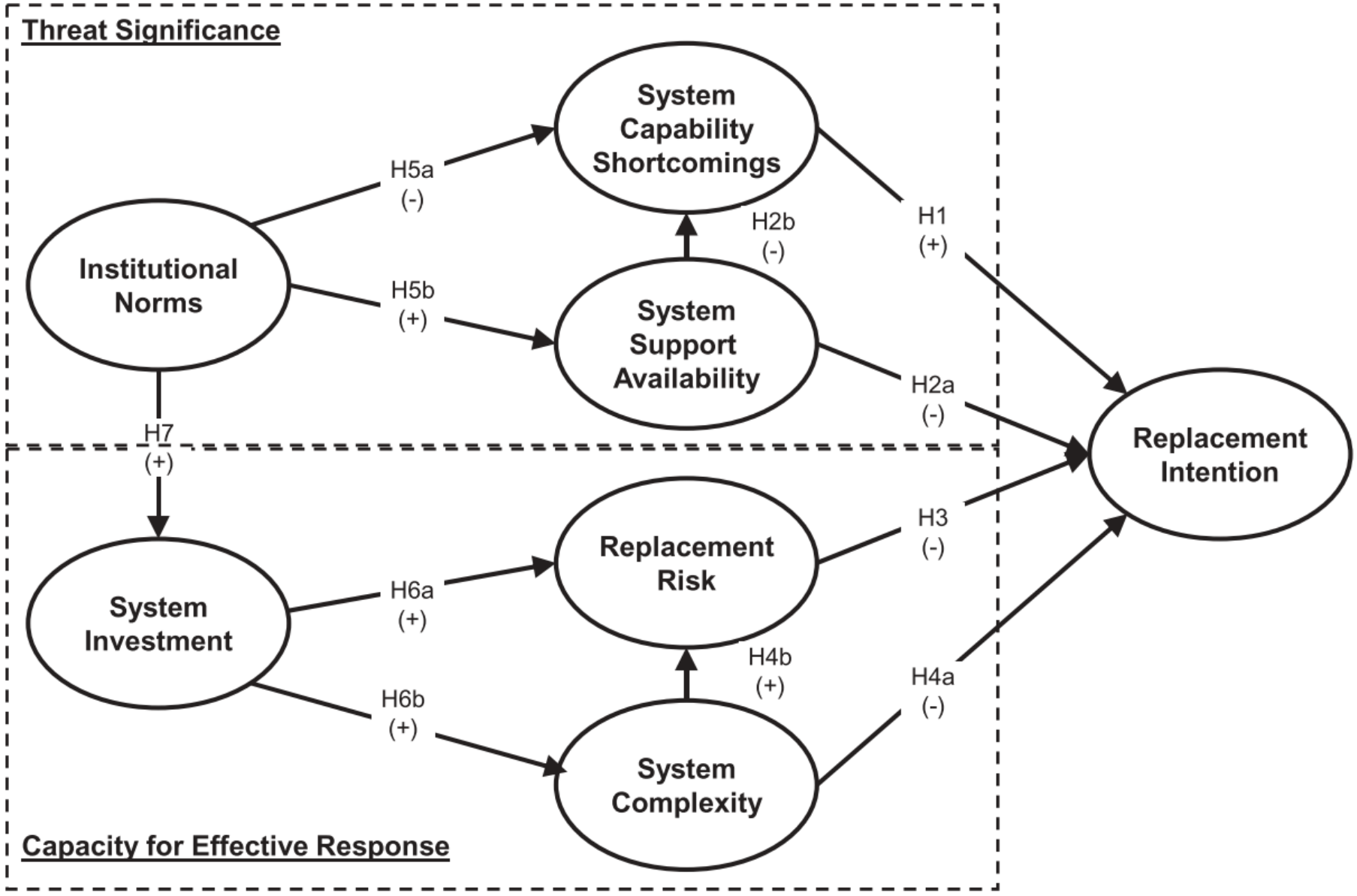
Two Approaches to Explaining Strategic Change[a]

Variance theory	Process theory
Explaining strategic change with a variance model	Explaining strategic change with a process model





Variance theory (e.g., Furneaux & Wade 2017)



Process theory (e.g., Snook 2000)

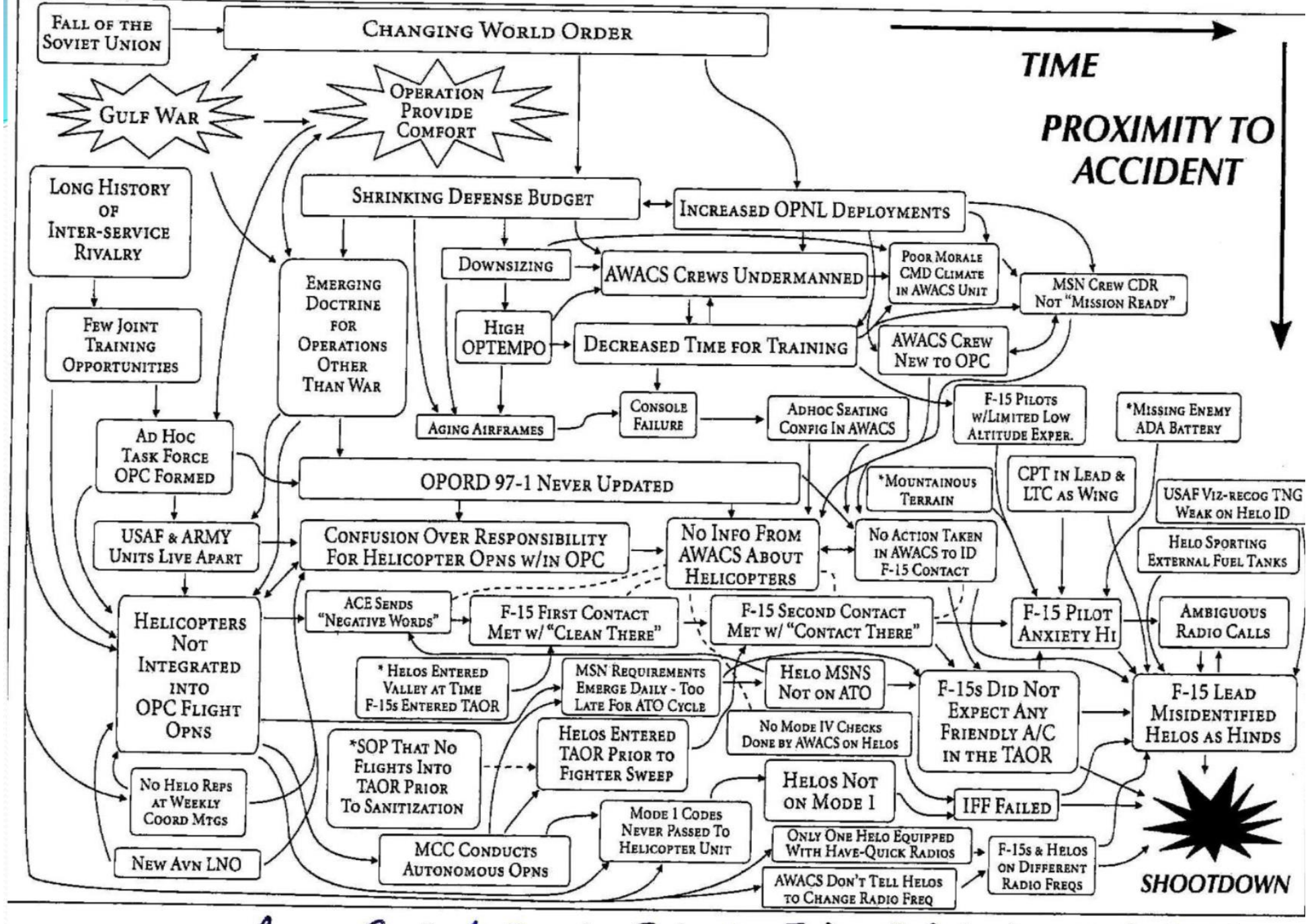
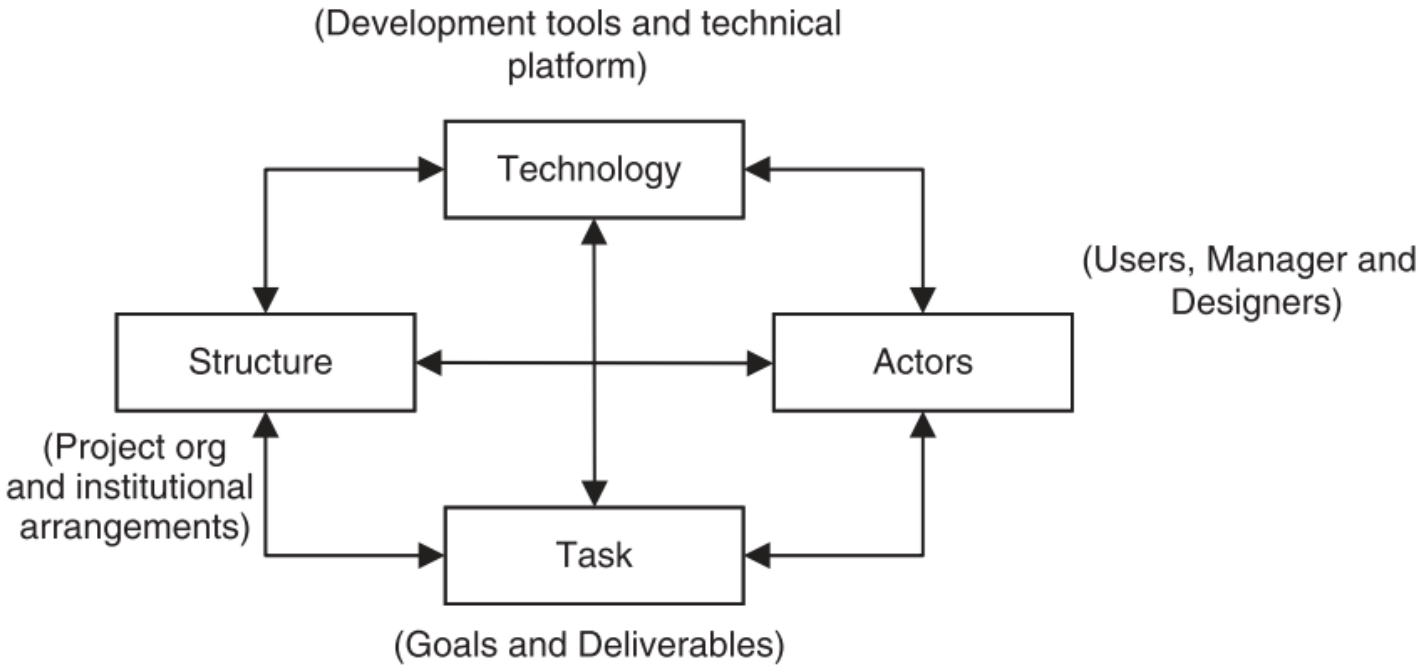


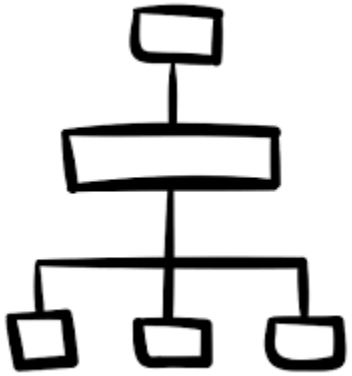
Fig. 1.3. Causal Map from Scott A. Snook, Friendly Fire, © Princeton U. P. 2000

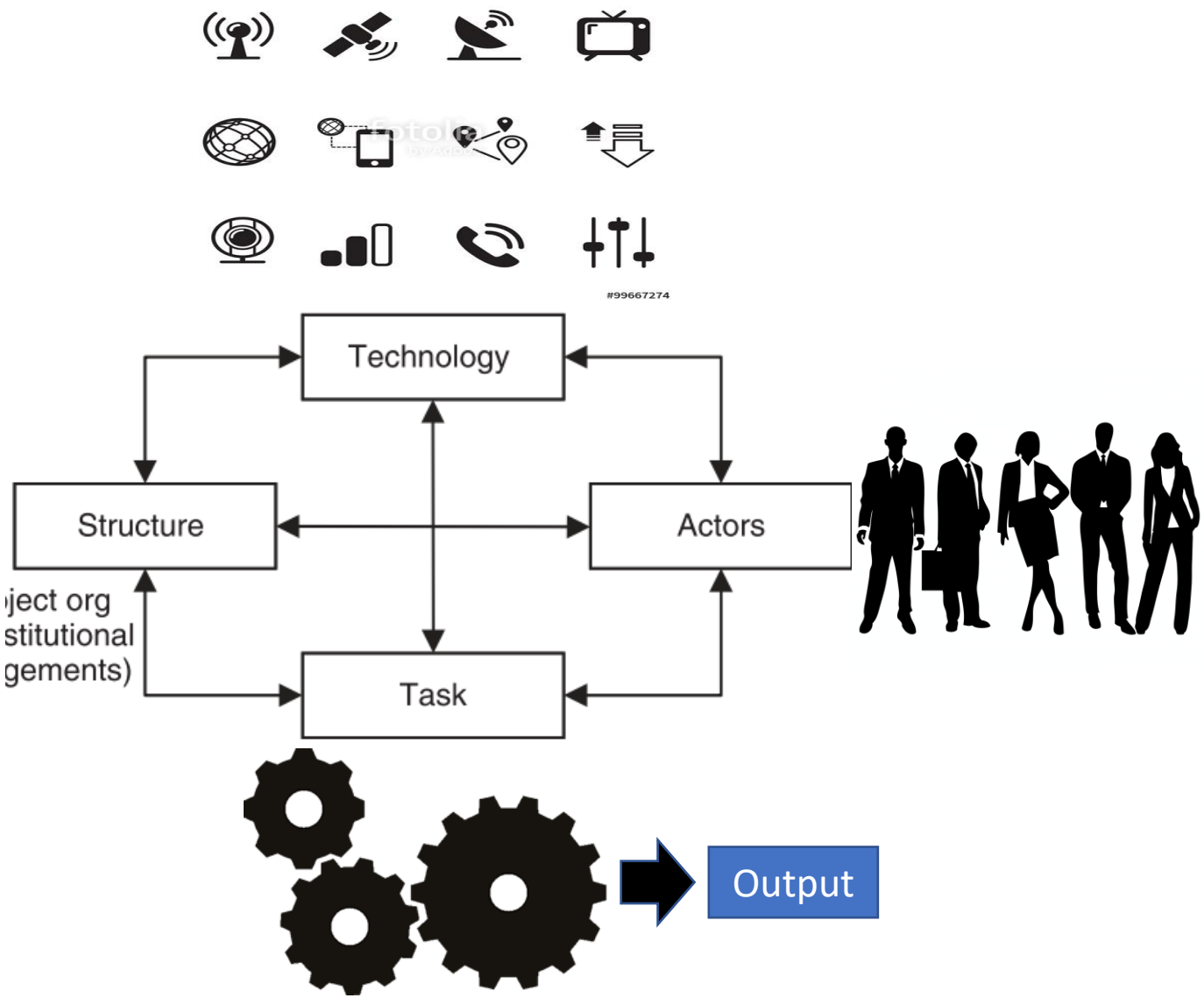
Socio-Technical Model a.k.a Leavitt's diamond (Leavitt 1964)



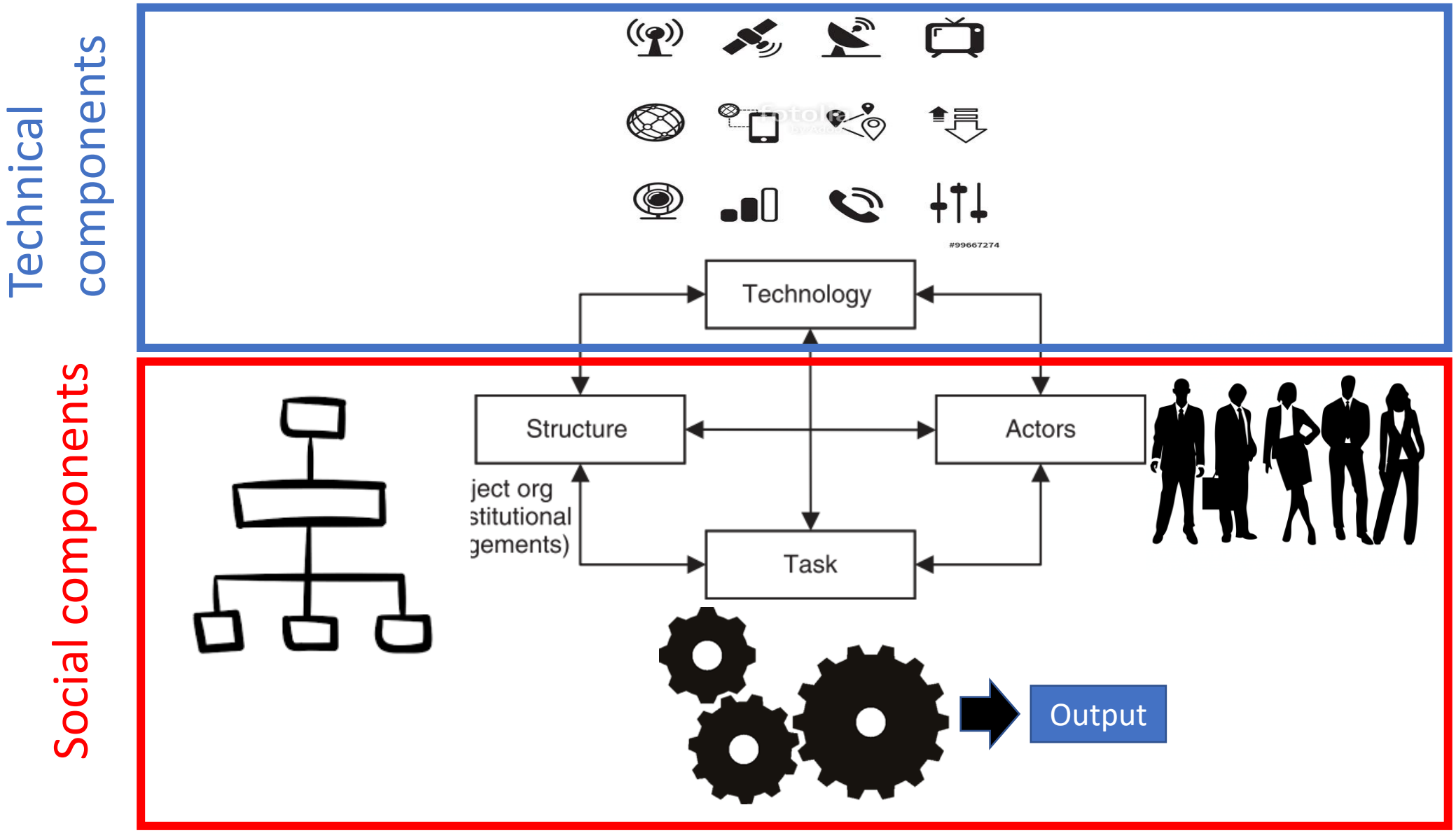
Leavitt, H. J. 1964. "Applied Organisation Change in Industry: Structural, Technical and Human Approaches," in *New Perspectives in Organization Research*, New York: Wiley.

Socio-Technical Model a.k.a Leavitt's diamond (Leavitt 1964)



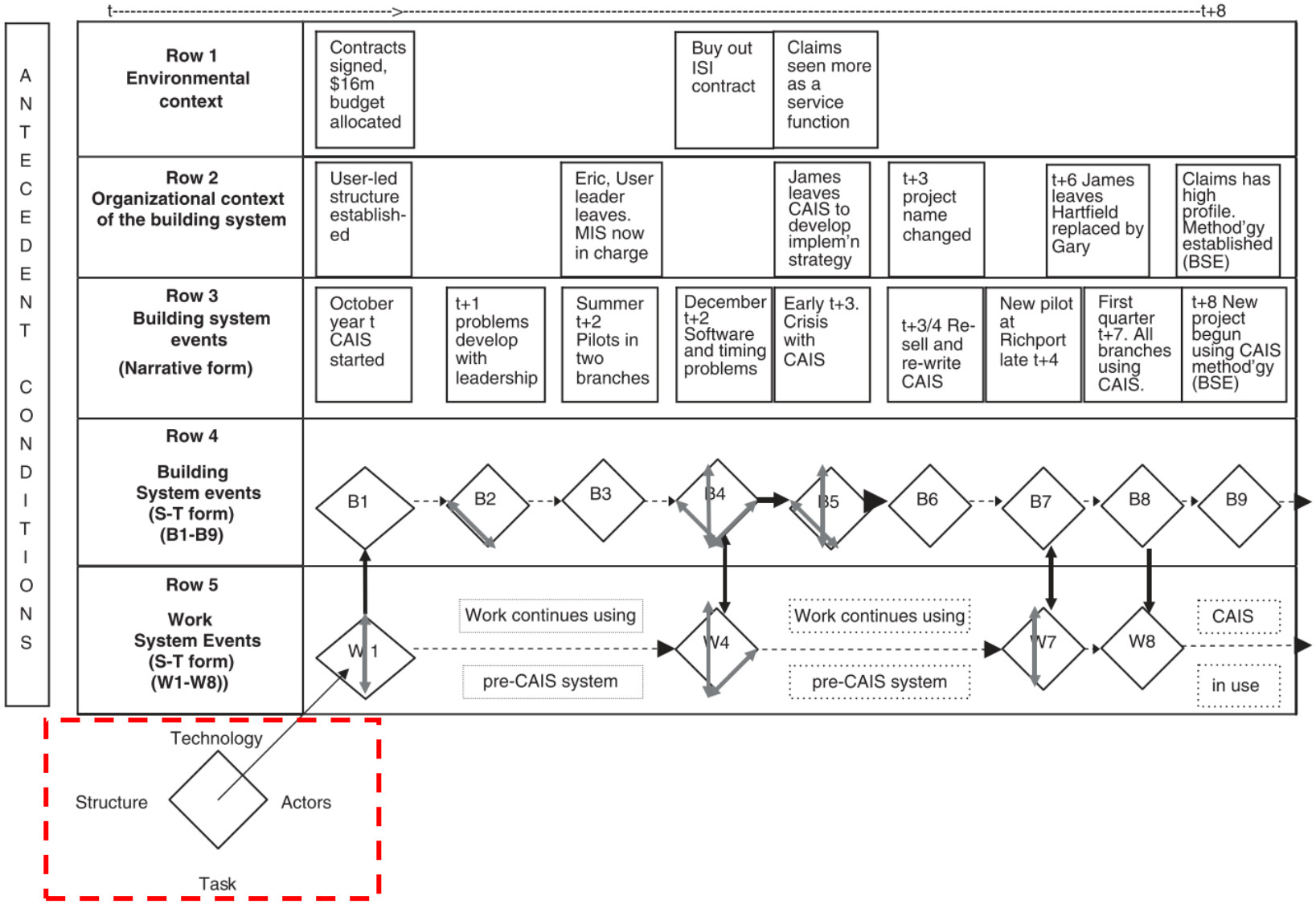


Socio-Technical Model a.k.a Leavitt's diamond (Leavitt 1964)

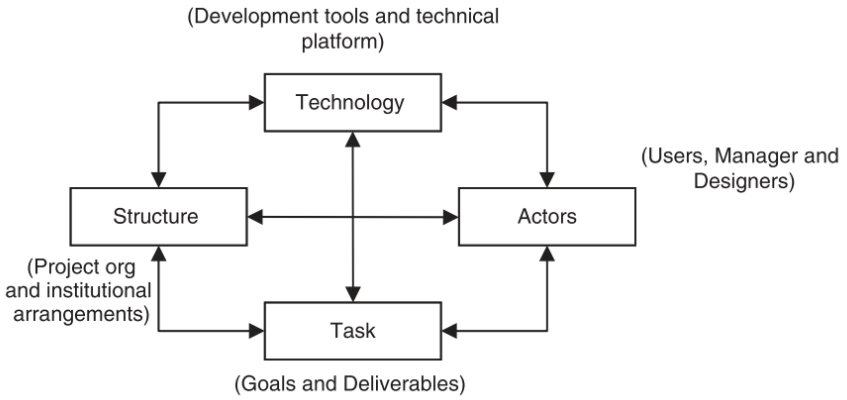


Punctuated Socio-Technical IS Change (PSIC) Model

(Lyytinen & Newman 2008)



Levels of analysis in PSIC

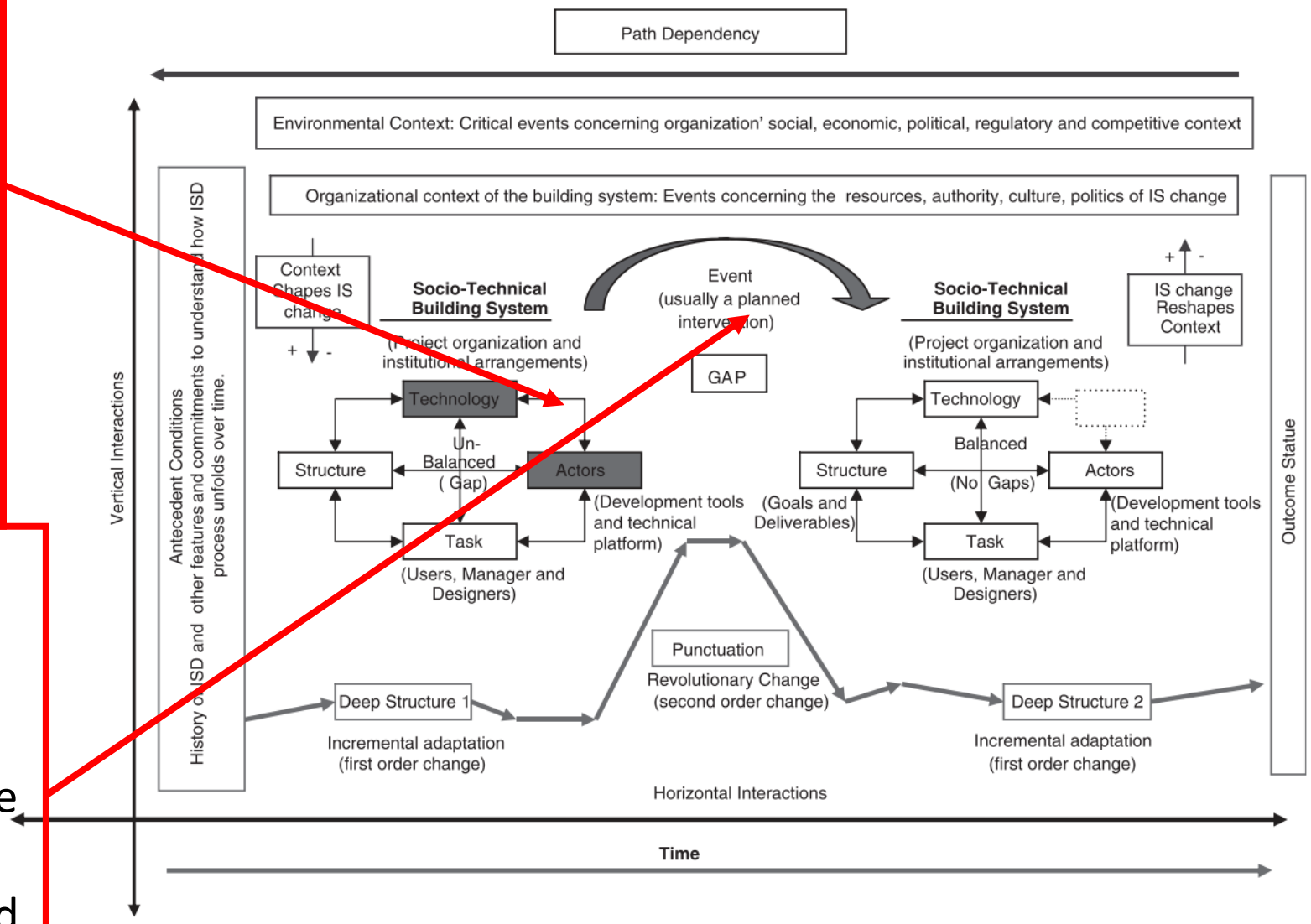


- Work system: incumbent organizational work processes
- Building system: resources and activities assembled for an implementation project
- Organizational context: e.g., top management, different departments, parent company, subsidiaries
- Environmental context: e.g., legislative and competitive environment

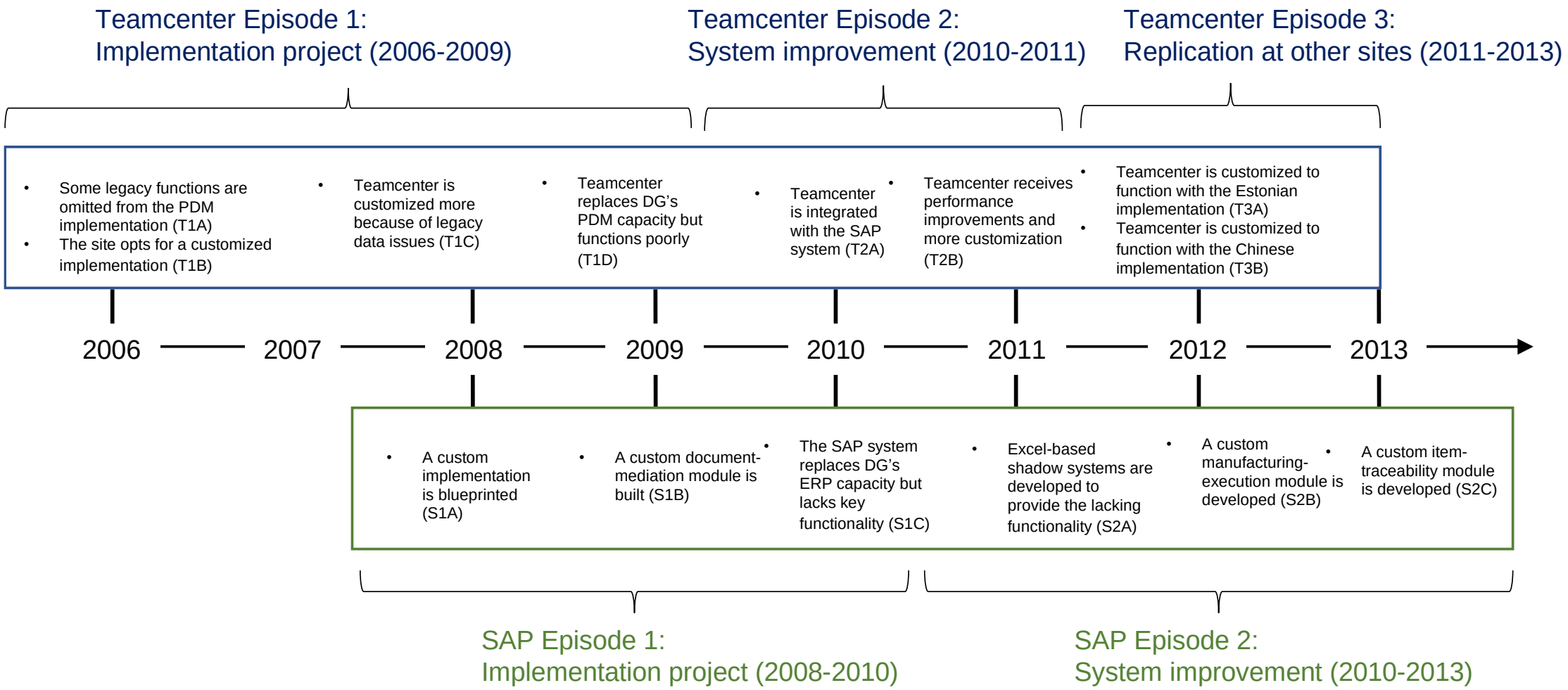
Example:

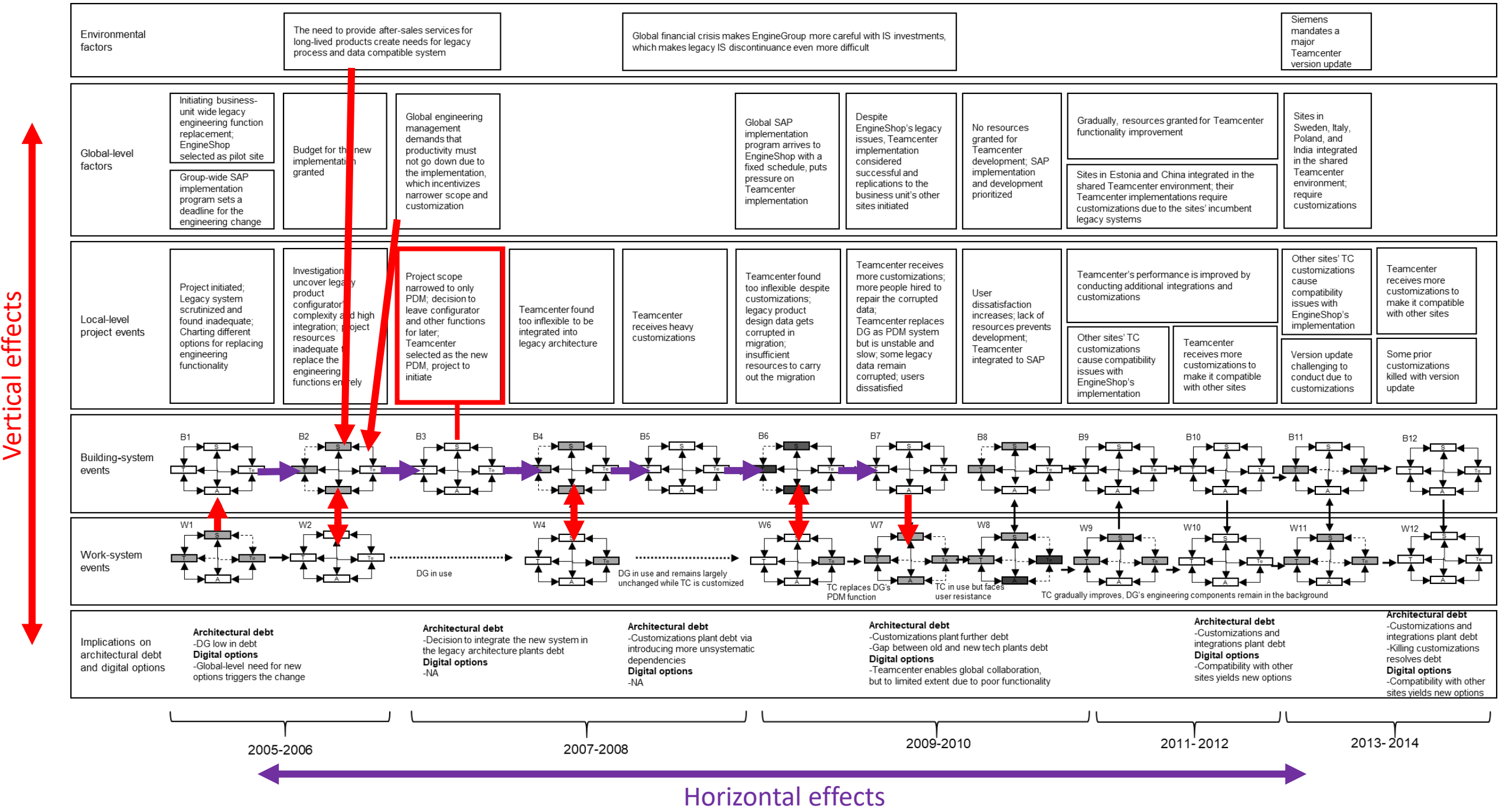
People don't have enough expertise to use the technology required to implement the new information system → a gap between actors and technology

Intervention needed to bridge the gap: hire people with more expertise / educate people / change the technology, etc. → punctuated change intended to stabilize the system



Timeline of events



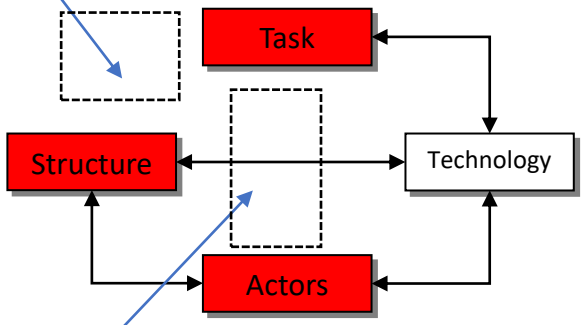

Environmental factors
The need to provide after-sales services for long-lived products create needs for legacy process and data compatible system
Global financial crisis makes EngineGroup more careful with IS investments, which makes legacy IS discontinuance even more difficult
Siemens mandates a major Teamcenter version update
Global-level factors
Initiating business-unit wide legacy engineering function replacement; EngineShop selected as pilot site
Group-wide SAP implementation program sets a deadline for the engineering change
Budget for the new implementation granted
Global engineering management demands that productivity must not go down due to the implementation, which incentivizes narrower scope and customization
Global SAP implementation program arrives to EngineShop with a fixed schedule, puts pressure on Teamcenter implementation
Despite EngineShop's legacy issues, Teamcenter implementation considered successful and replications to the business unit's other sites initiated
No resources granted for Teamcenter development; SAP implementation and development prioritized
Gradually, resources granted for Teamcenter functionality improvement
Sites in Estonia and China integrated in the shared Teamcenter environment; their Teamcenter implementations require customizations due to the sites' incumbent legacy systems
Sites in Sweden, Italy, Poland, and India integrated in the shared Teamcenter environment; require customizations
Local-level project events
Project initiated; Legacy system scrutinized and found inadequate; Charting different options for replacing engineering functionality
Investigation uncover legacy product configurator's complexity and high integration; project resources inadequate to replace the engineering functions entirely
Project scope narrowed to only PDM; decision to leave configurator and other functions for later; Teamcenter selected as the new PDM, project to initiate
Teamcenter found too inflexible to be integrated into legacy architecture
Teamcenter receives heavy customizations
Teamcenter found too inflexible despite customizations; legacy product design data gets corrupted in migration; insufficient resources to carry out the migration
Teamcenter receives more customizations; more people hired to repair the corrupted data; Teamcenter replaces DG as PDM system but is unstable and slow; some legacy data remain corrupted; users dissatisfied
User dissatisfaction increases; lack of resources prevents development; Teamcenter integrated to SAP
Teamcenter's performance is improved by conducting additional integrations and customizations
Other sites' TC customizations cause compatibility issues with EngineShop's implementation
Teamcenter receives more customizations to make it compatible with other sites
Other sites' TC customizations cause compatibility issues with EngineShop's implementation
Teamcenter receives more customizations to make it compatible with other sites
Version update challenging to conduct due to customizations
Some prior customizations killed with version update
Building-system events
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
Work-system events
W1
W2
W4
W6
W7
W8
W9
W10
W11
W12
DG in use
DG in use and remains largely unchanged while TC is customized
TC replaces DG's PDM function
TC in use but faces user resistance
TC gradually improves, DG's engineering components remain in the background
Implications on architectural debt and digital options
Architectural debt
-DG low in debt
Digital options
-Global-level need for new options triggers the change
Architectural debt
-Decision to integrate the new system in the legacy architecture plants debt
Digital options
-NA
Architectural debt
-Customizations plant debt via introducing more unsystematic dependencies
Digital options
-NA
Architectural debt
-Customizations plant further debt
-Gap between old and new tech plants debt
Digital options
-Teamcenter enables global collaboration, but to limited extent due to poor functionality
Architectural debt
-Customizations and integrations plant debt
Digital options
-Compatibility with other sites yields new options
Architectural debt
-Customizations and integrations plant debt
-Killing customizations resolves debt
Digital options
-Compatibility with other sites yields new options
2005-2006
2007-2008
2009-2010
2011- 2012
2013- 2014
Vertical effects
Horizontal effects

Building-system level (i.e., implementation project)

Gap: Task-Structure
Project resources found insufficient for implementing the system in vanilla form

Gap: Task-Actors
People have inadequate capacity to conduct a massive change

Imbalanced building system

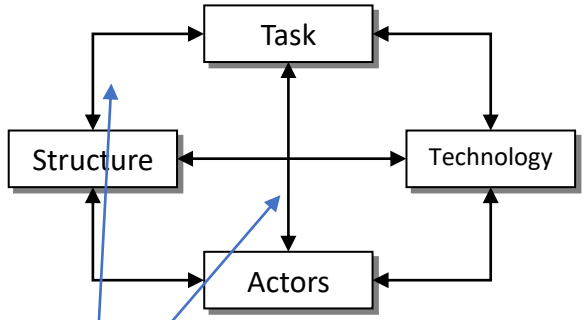


How to make the building system balanced?

Choice:

Let's customize the technology!

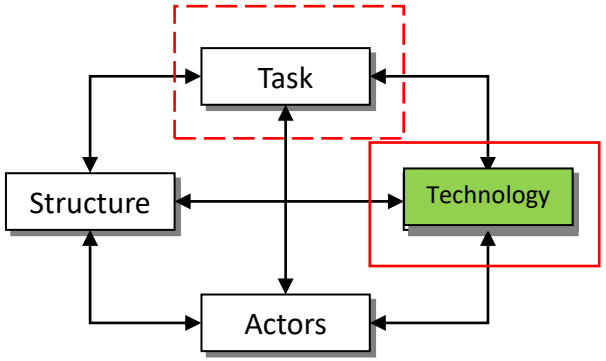
Balanced building system



System balanced by revising the project scope and objectives to align with the available resources; people now able to conduct the change

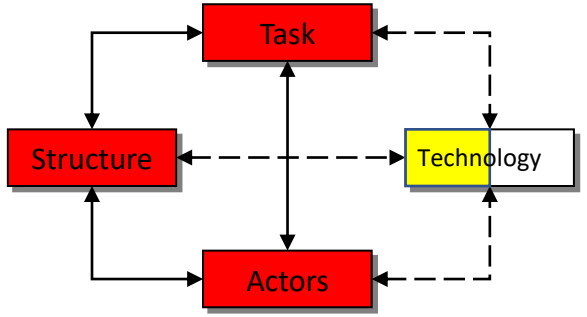
Work-system level (i.e., system for daily work)

Before: DG legacy system



Technology: 100 % malleable

Now: Teamcenter/SAP



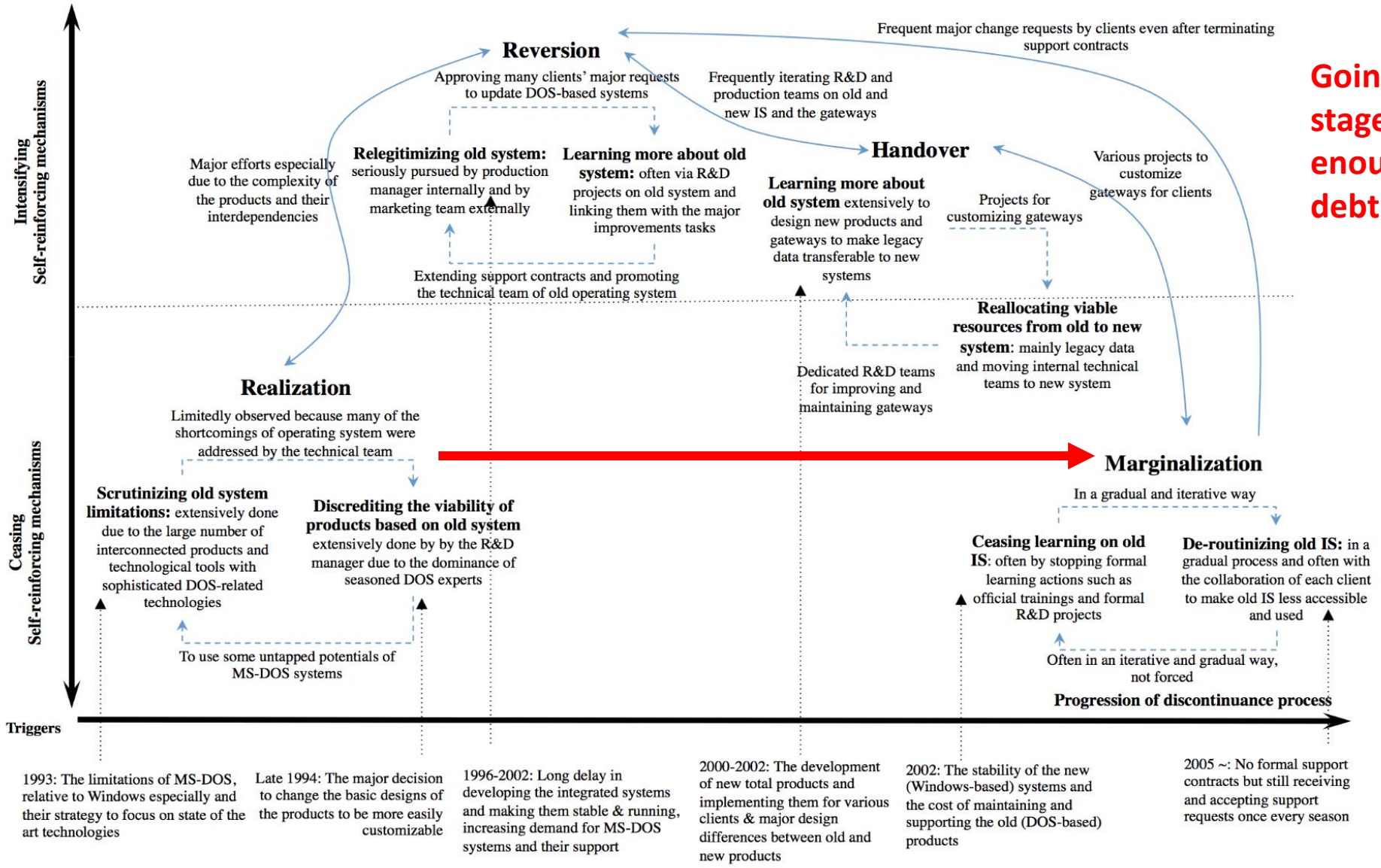
Technology: much less malleable

When technology can no longer bend, other socio-technical components have to bend!

- Even with customizations, COTS systems were poorly aligned with incumbent socio-technical systems

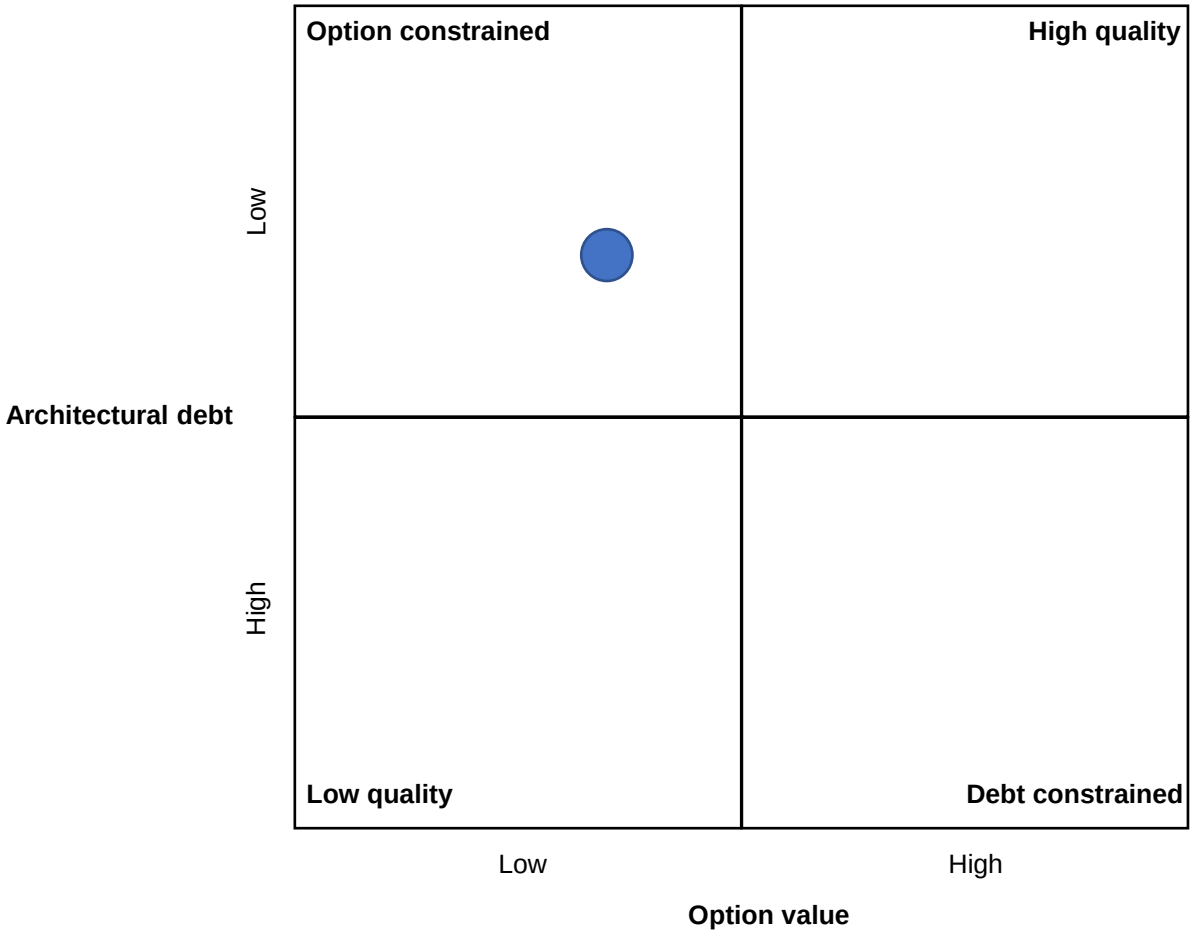
Process of legacy system discontinuance

(Mehrizi et al. 2019)

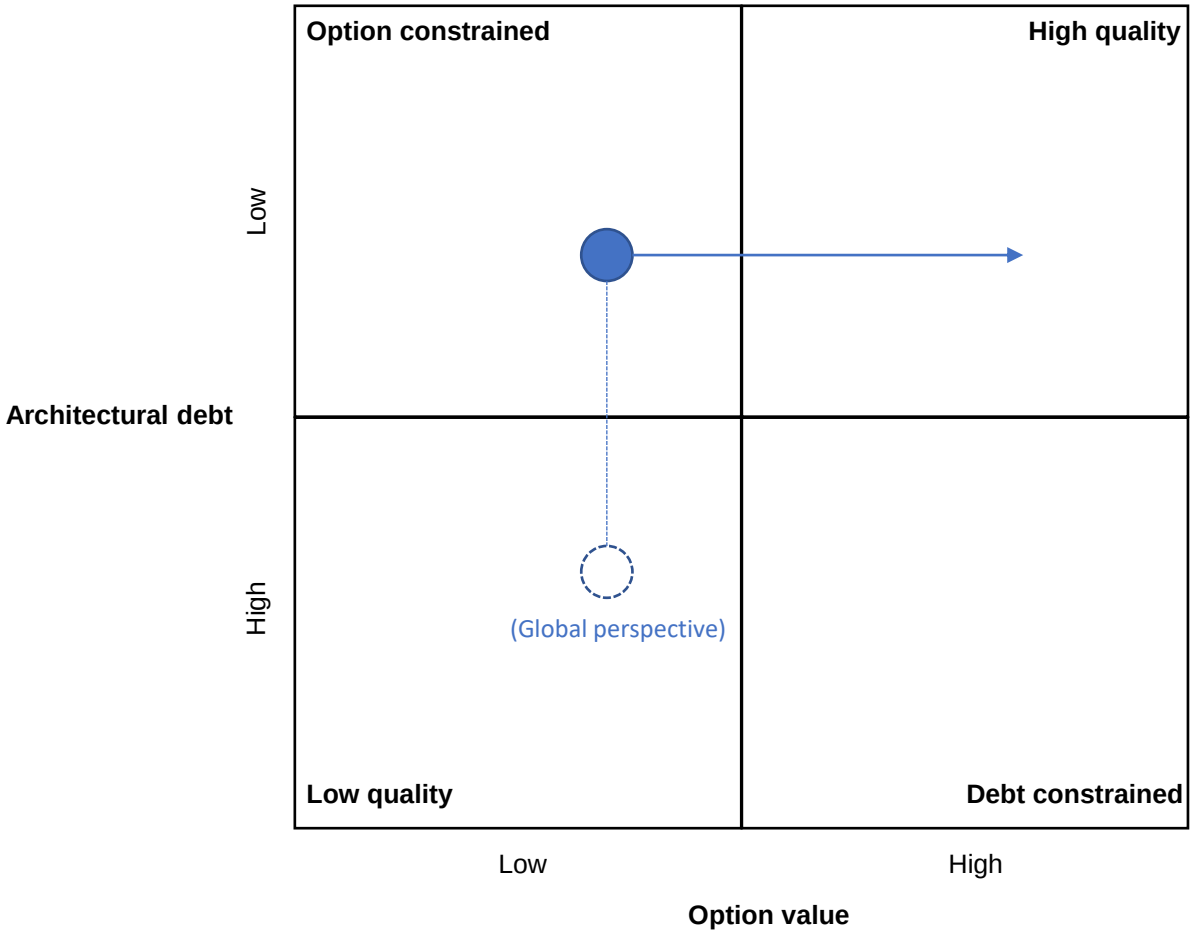


Going through these stages may not be enough if technical debt is not resolved!

Local perspective at the Helsinki factory



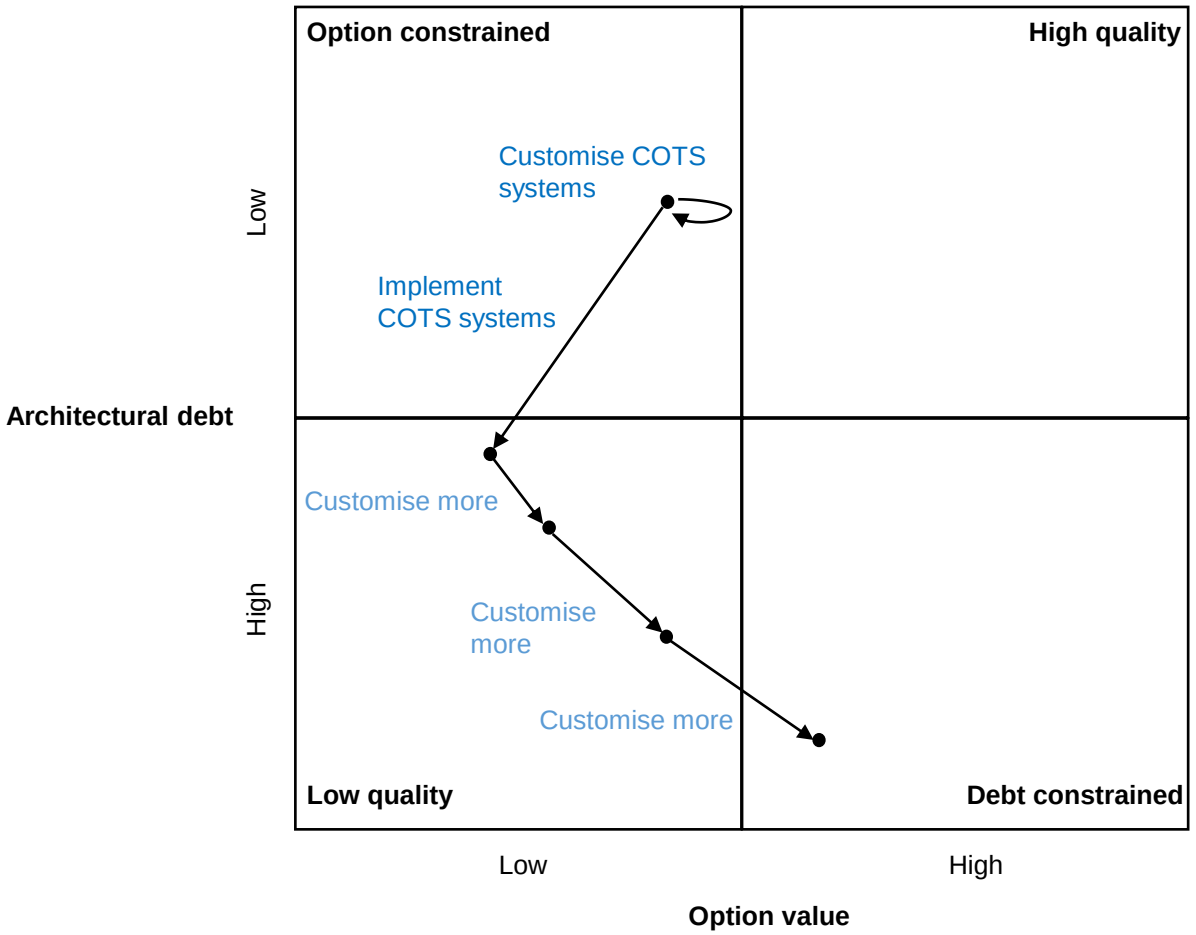
Local perspective at the Helsinki factory



Design-moves analysis

Design move	Episode and strategic intent	Socio-technical gaps	Design actions	State of design capital	Impact of design move
T1A	1: Guarantee local productivity while implementing a product-engineering system with global digital options	Building-system structure not supportive of the implementation task, because of incompatibilities within DG work-system structure	Reduce implementation scope by excluding some legacy applications from replacement	Option constrained	Increased debt
T1B		Building-system structure and actors inadequate for the implementation task, for reason of incompatibility with DG work-system structure	Customize Teamcenter for the sustained legacy environment	Option constrained	Increased debt
T1C		Building-system structure and actors not adequate for the implementation task, because of incompatibility with DG work-system structure and technology	Customize Teamcenter for the sustained legacy environment and hire people to fix the legacy-system data	Option constrained	Increased debt
T1D		Customizations balancing the building system and enabling Teamcenter implementation	Replace DG's PDM functionality with Teamcenter	Low quality	Abandoned options Increased debt
T2A	2: Create conditions for global collaboration in product engineering while maintaining acceptable levels of local productivity	Incompatibility of the Teamcenter work system's customized technology with the SAP system	Integrate Teamcenter with the SAP system by using a customized module	Low quality	Created options Increased debt
T2B		Incompatibility of the Teamcenter work system's technology with the structure, actors, and task	Customize further and fix significant bugs detected as hampering engineering work	Low quality	Created options Increased debt
T3A	3: Pursue global digital options provided by the new shared system	Incompatibility of the Teamcenter work system's technology with the structure and task, because of the Estonian site's work system	Customize Teamcenter, to create compatibility with the Estonian site	Debt constrained	Created options Increased debt
T3B		Incompatibility of the Teamcenter work system's technology with the structure and task, because of the Chinese site's work system	Customize Teamcenter, for compatibility with the Chinese site	Debt constrained	Created options Increased debt

Design moves at EngineShop



The consequences of social inertia

- Insufficient change in people, processes and structures
- Failure to decommission legacy systems
- Increased customisation of COTS systems
  - Escalation of technical debt
  - Less benefit from new systems

Local vs global trade-off

Local perspective (managers and engineers at the factory)

- Poor performance and reliability
- Increased complexity in IS architecture because legacy system still operational
- Persistent end-user dissatisfaction

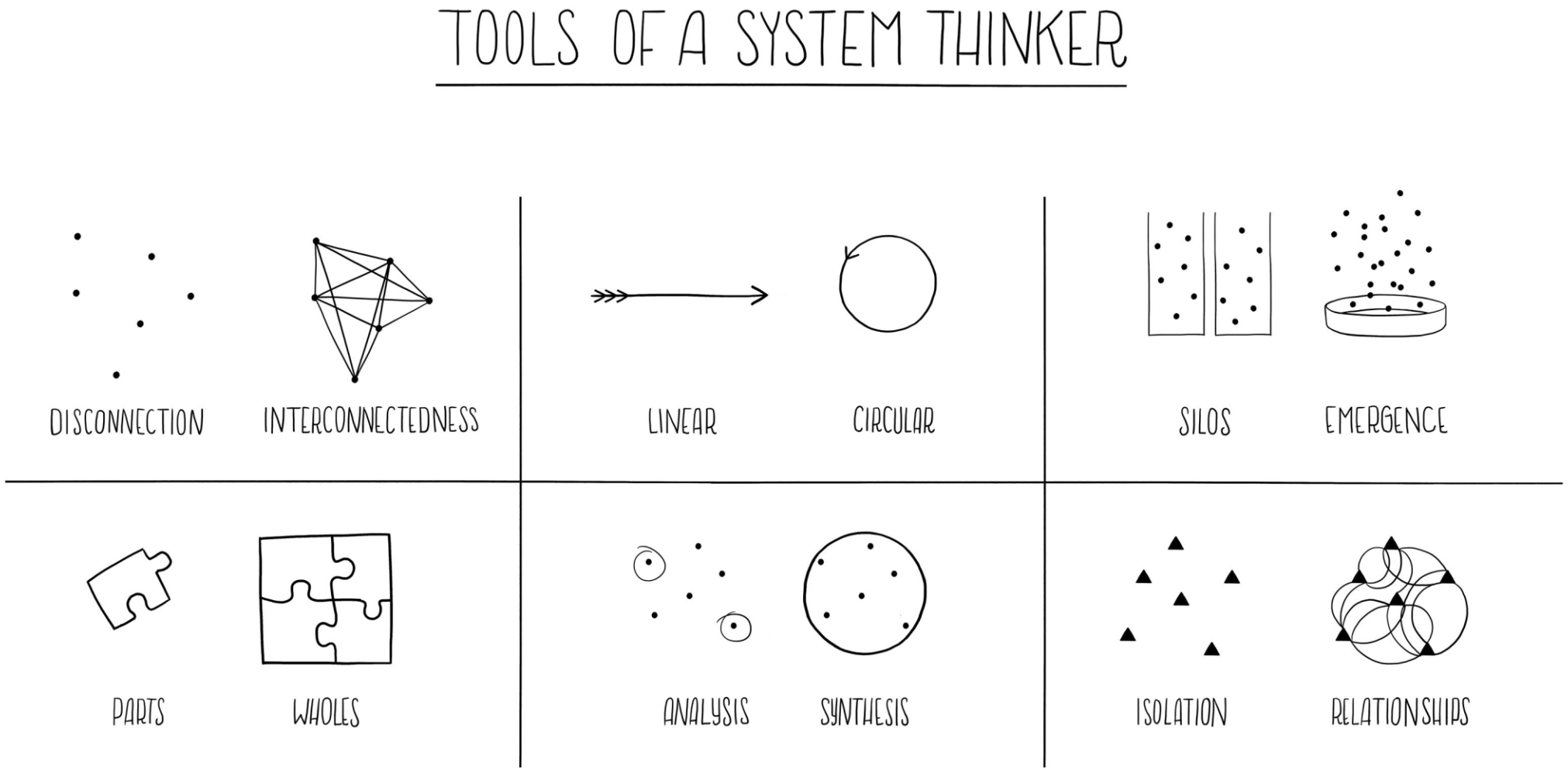
Global perspective (top management)

- Scalable modular system
- Enables global operation strategy
- Externally maintained

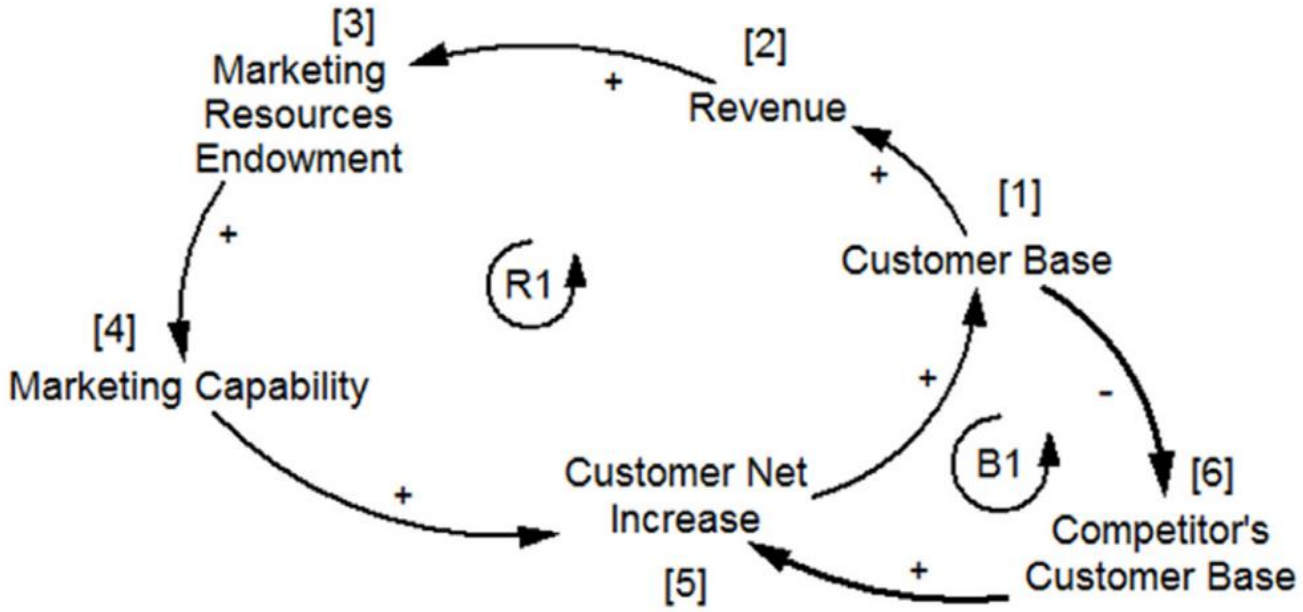
→ Hard to justify a business case for discontinuing the remnants of DG

- Decentralized matrix organization makes change complex

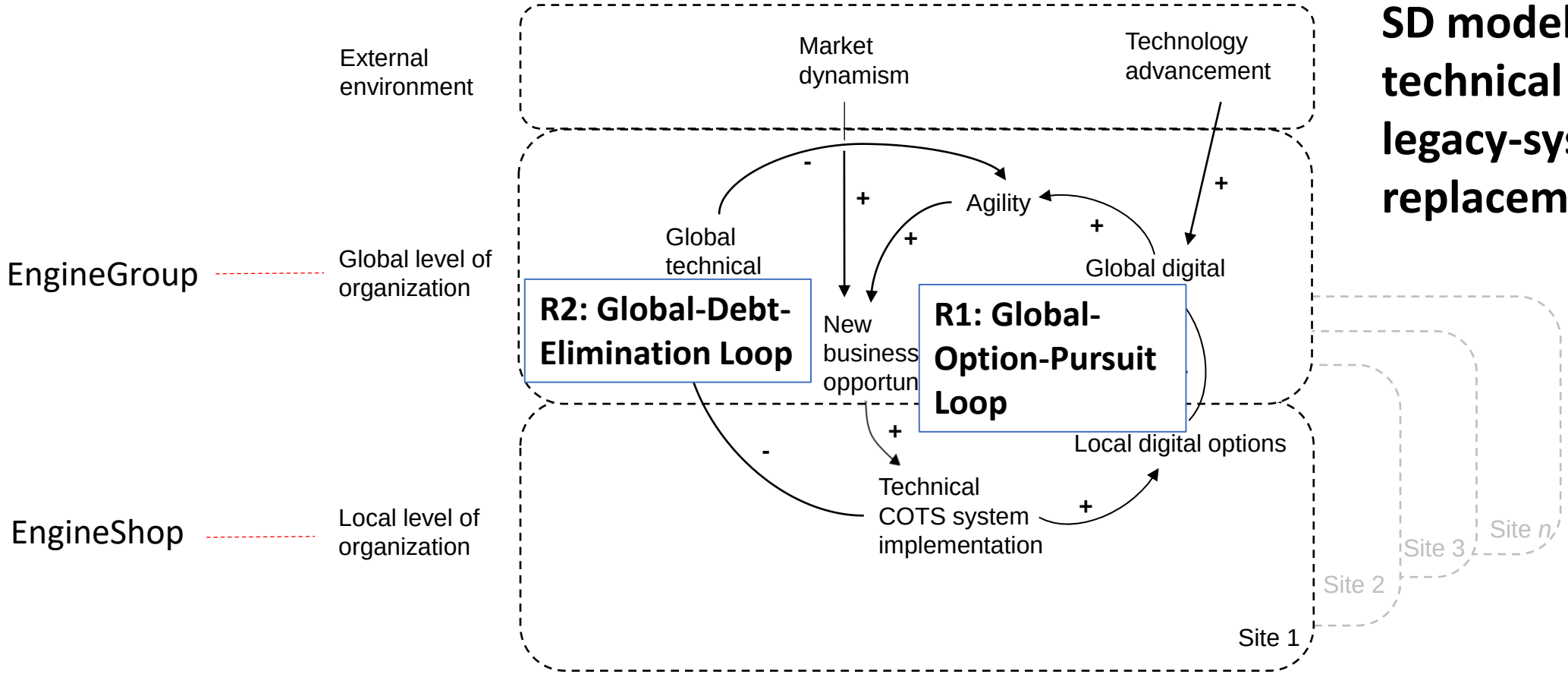
Theoretical synthesis: systems dynamics (SD) approach



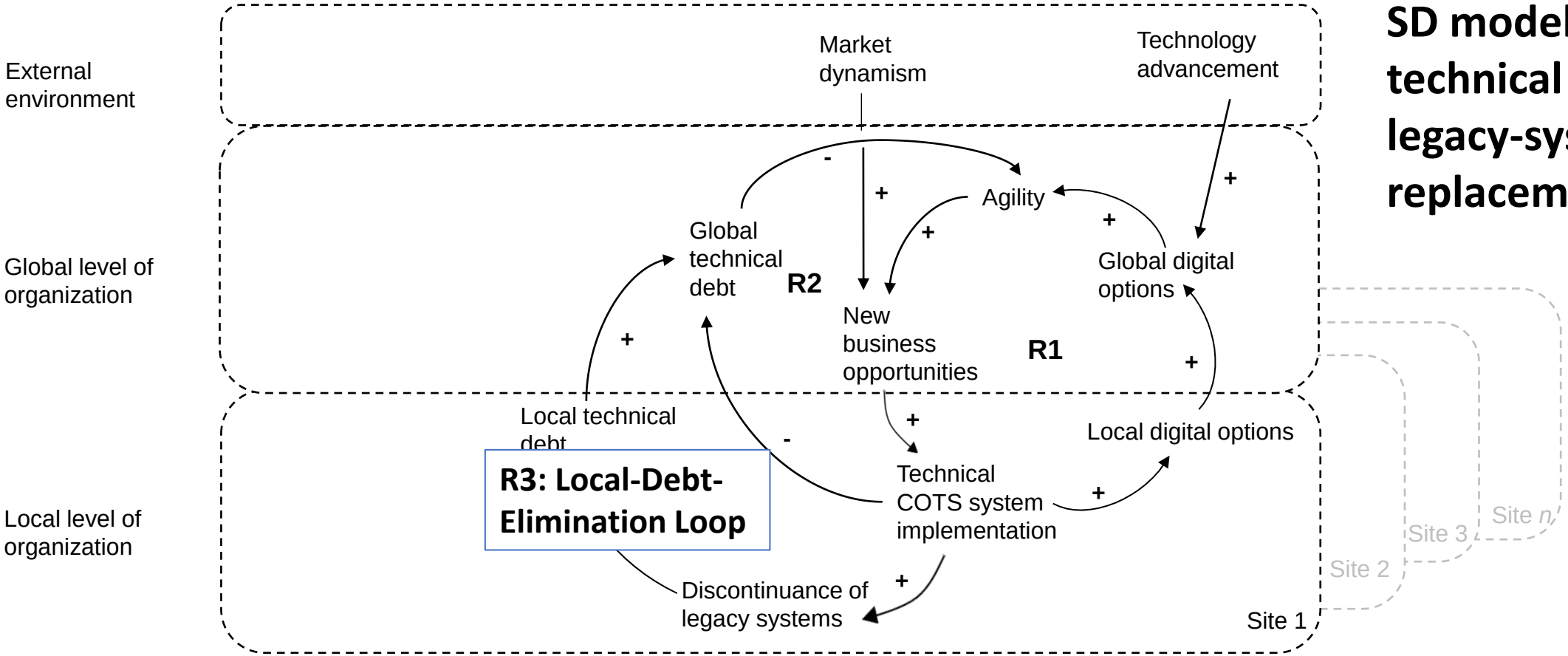
SD modelling: example



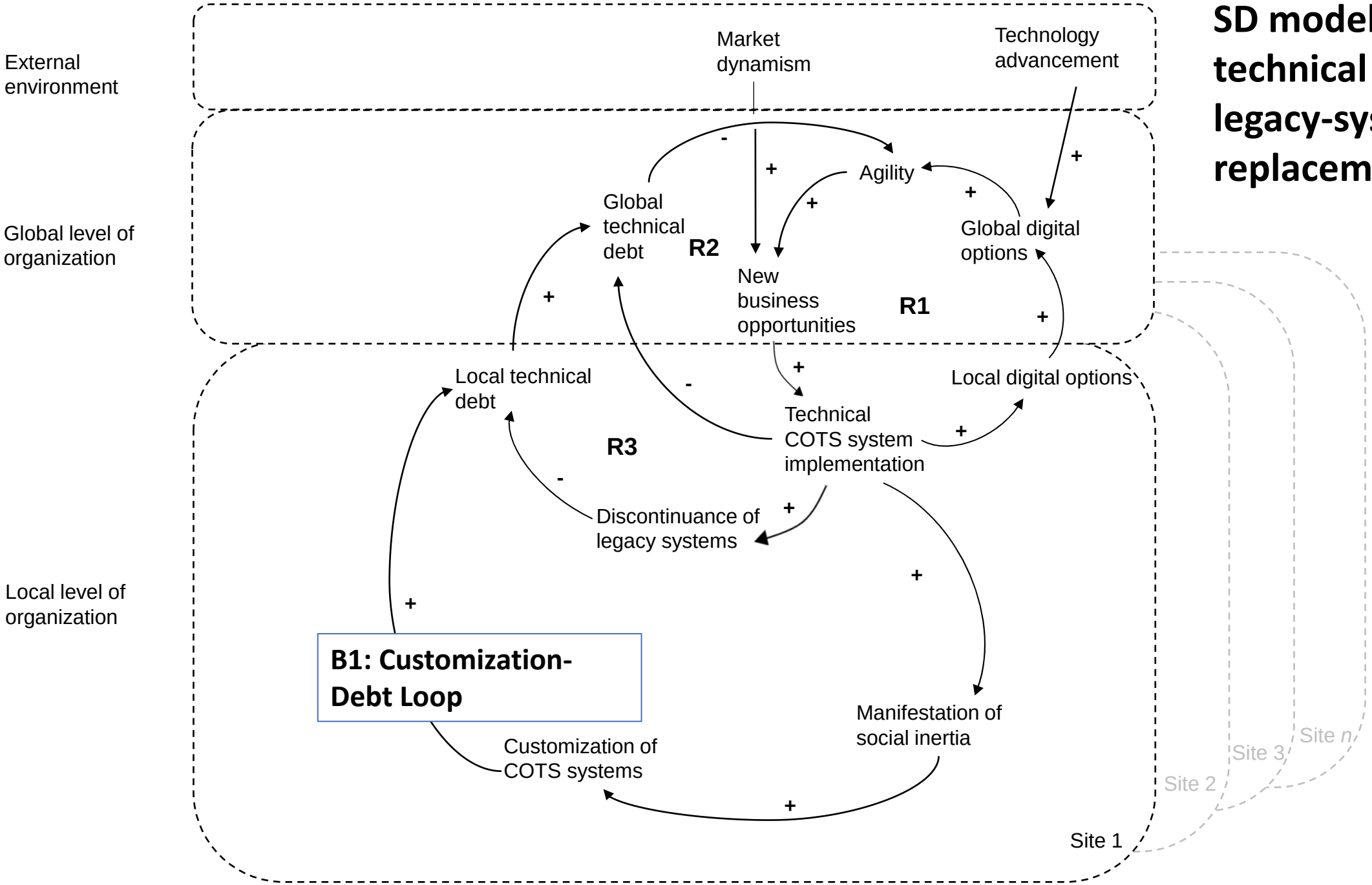
SD model of technical debt and legacy-system replacement



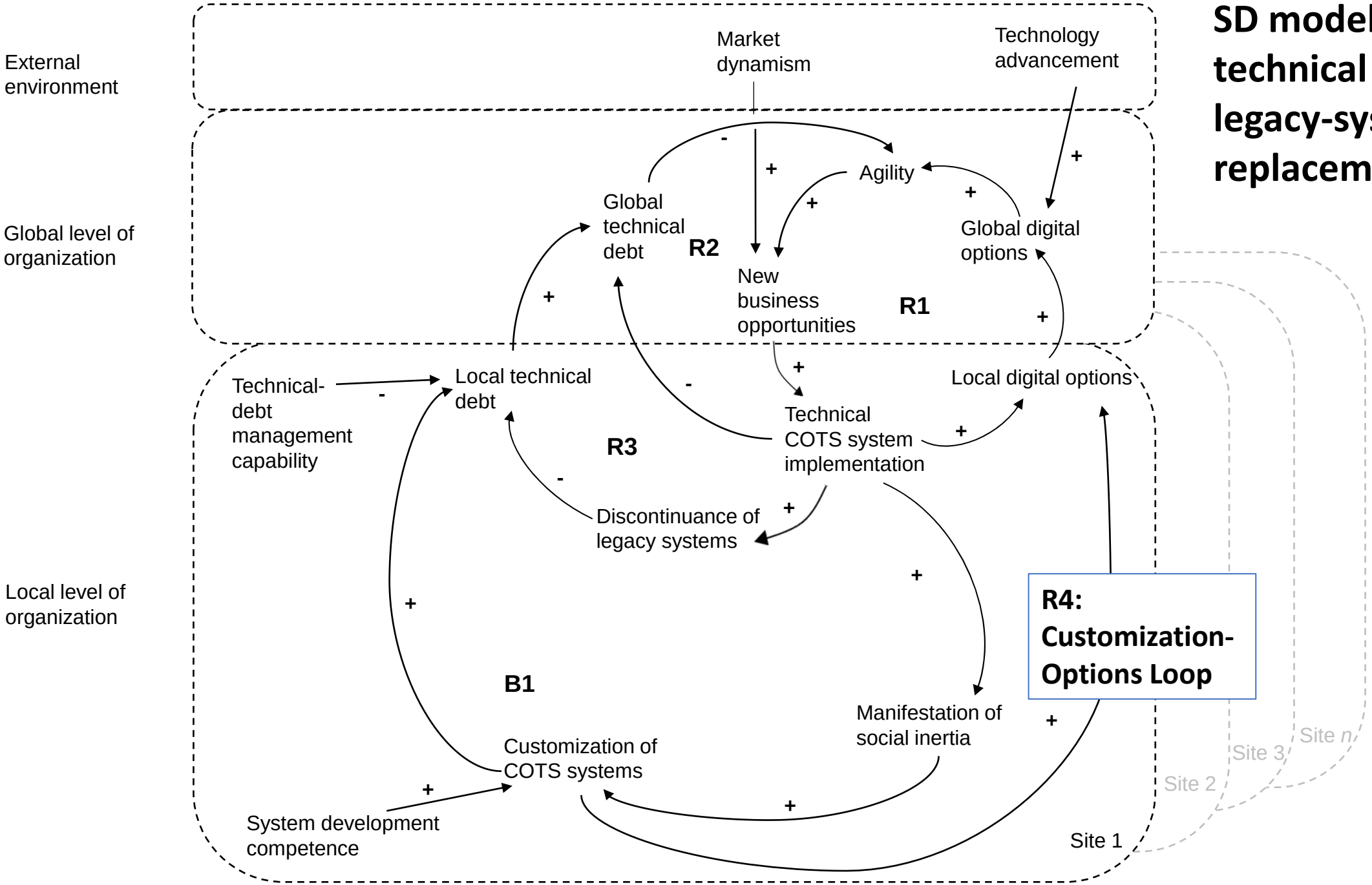
SD model of technical debt and legacy-system replacement



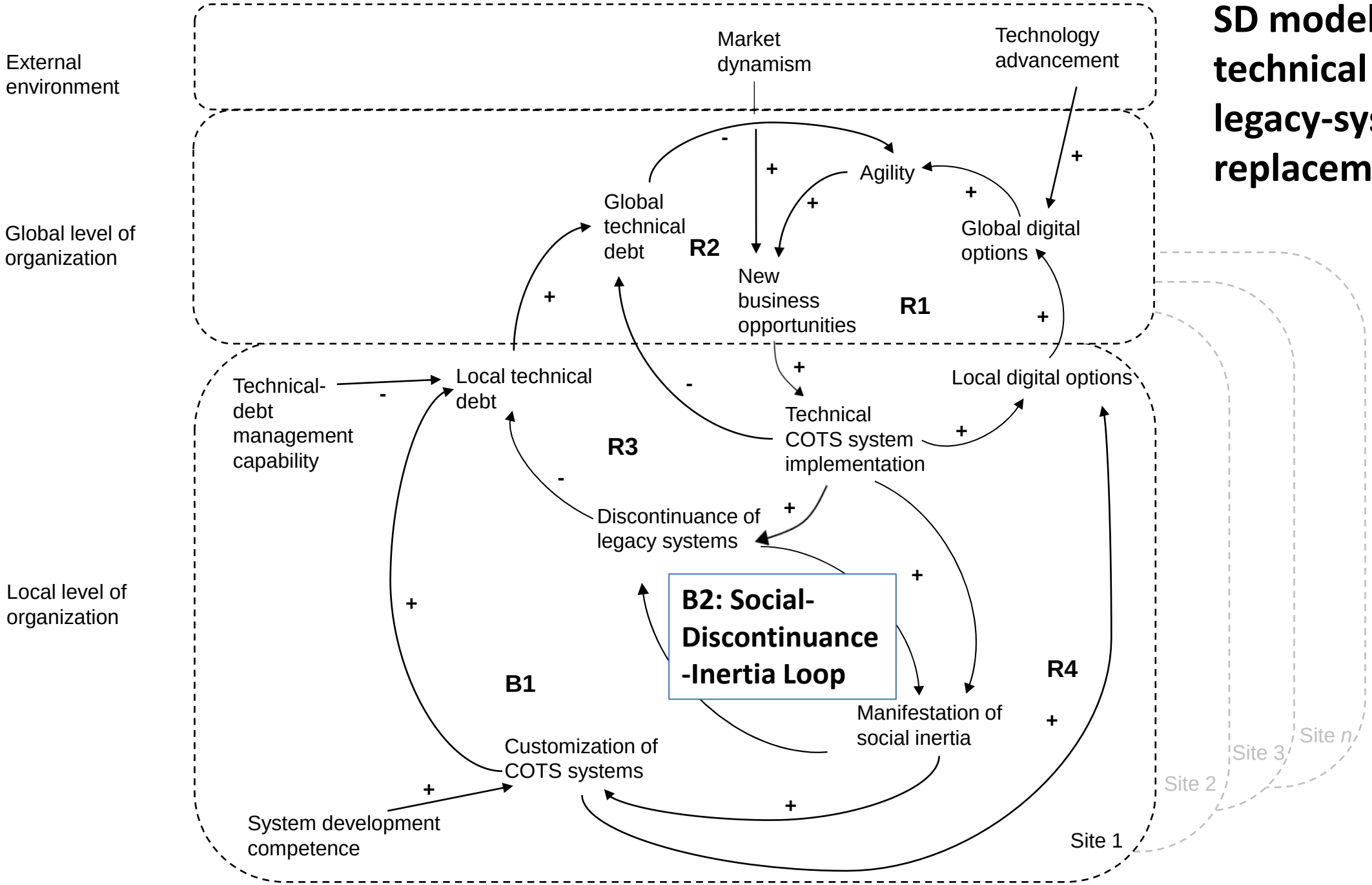
SD model of technical debt and legacy-system replacement



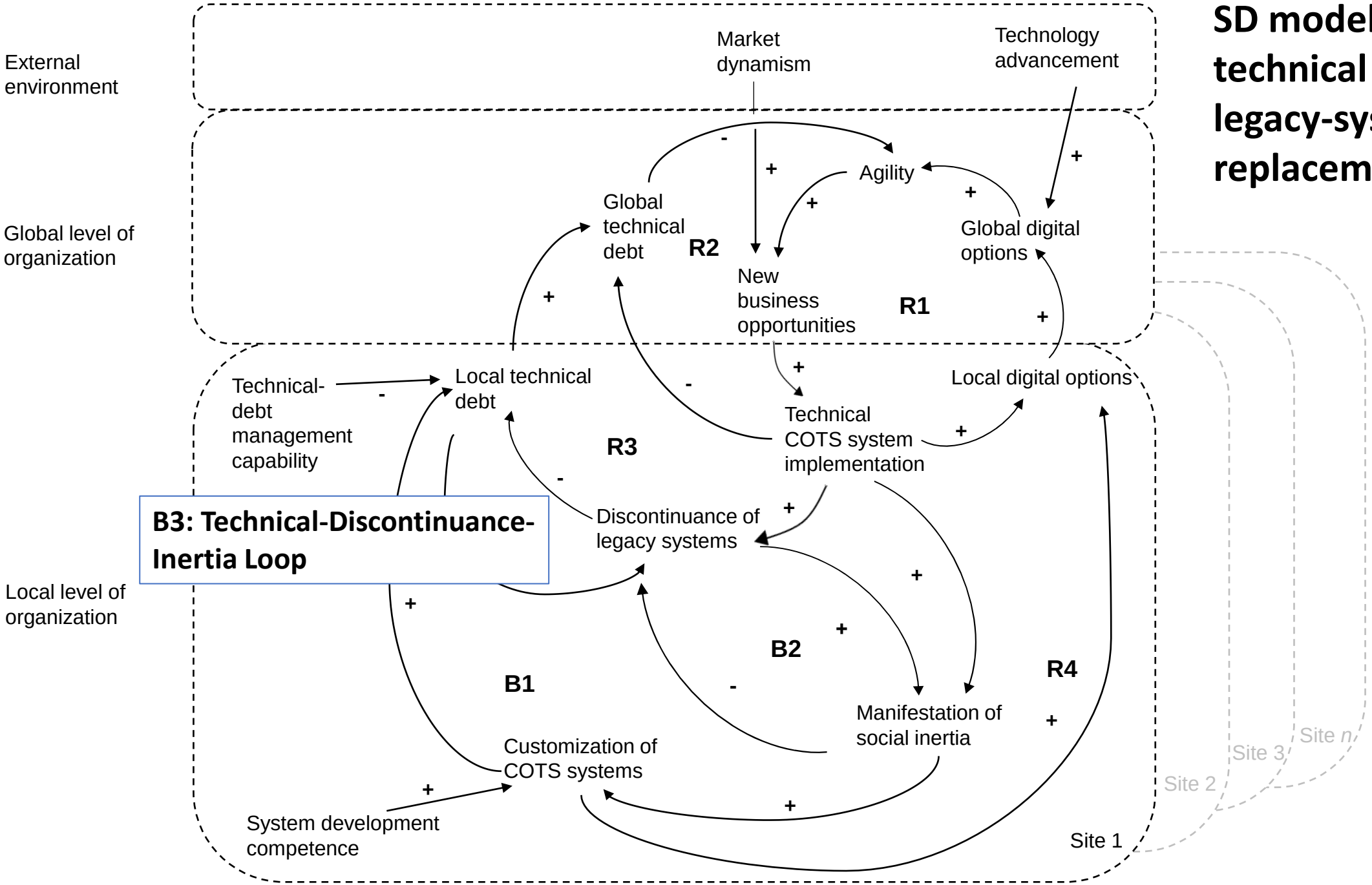
SD model of technical debt and legacy-system replacement



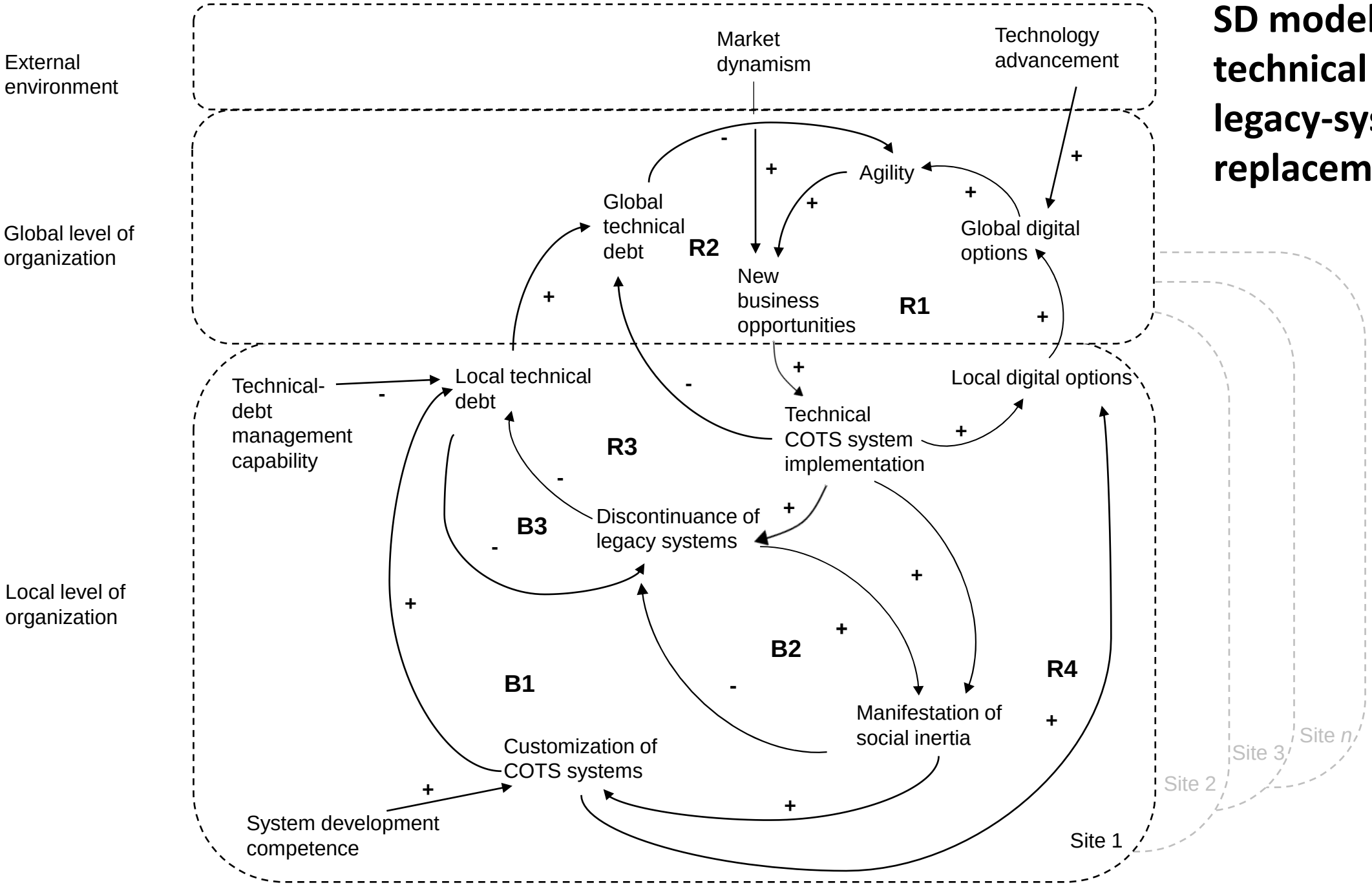
SD model of technical debt and legacy-system replacement



SD model of technical debt and legacy-system replacement



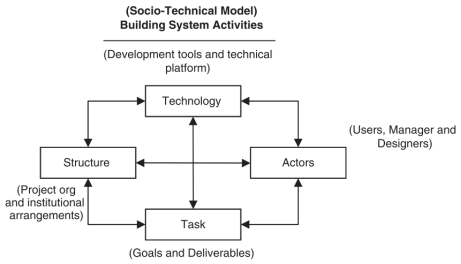
SD model of technical debt and legacy-system replacement

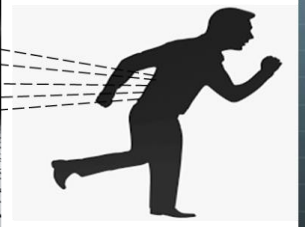


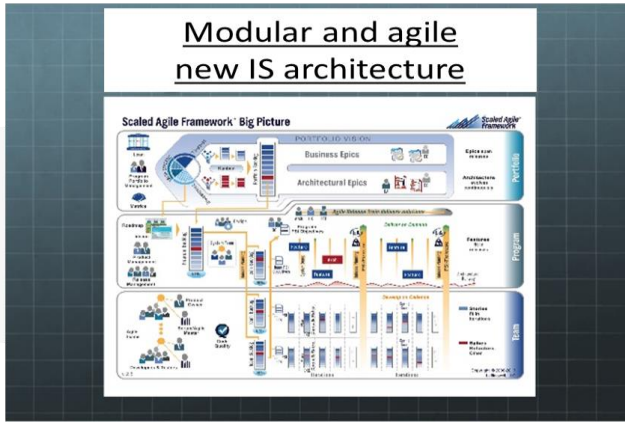
Implications

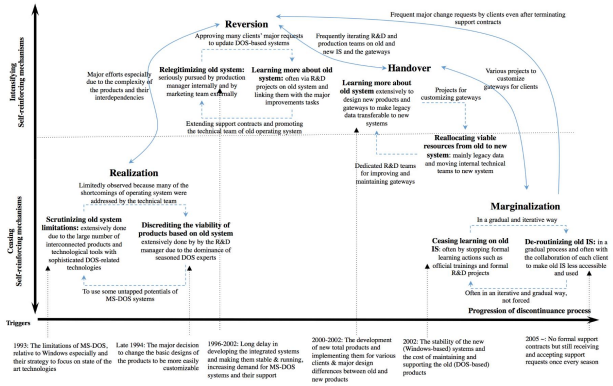
- Eternal questions about COTS system implementation
  - Vanilla vs customized – depends on the context
  - Important to consider short-term vs long-term risks
- We provide a vocabulary for a more systematic understanding of system implementation pain points familiar to many organizations
- The SD model provides a holistic overview - a tool for understanding the dynamics that are likely to ensue

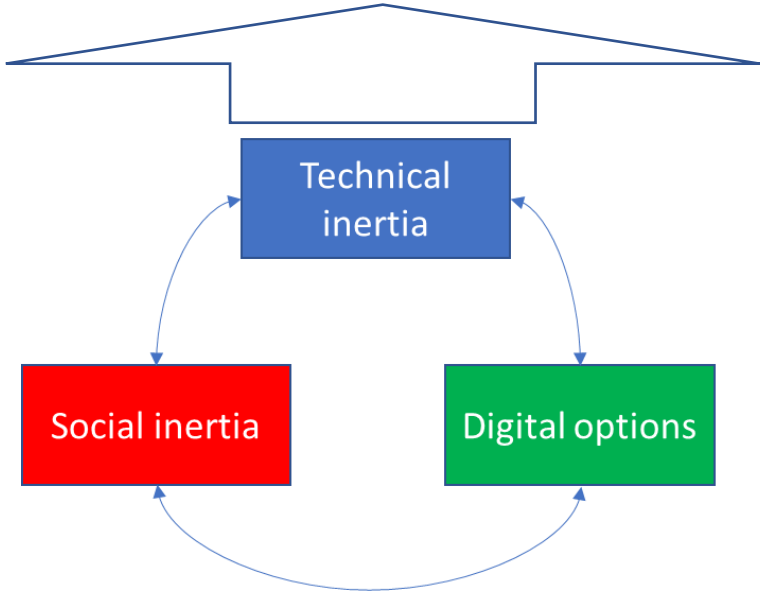
Caught in between old and new IS architectures

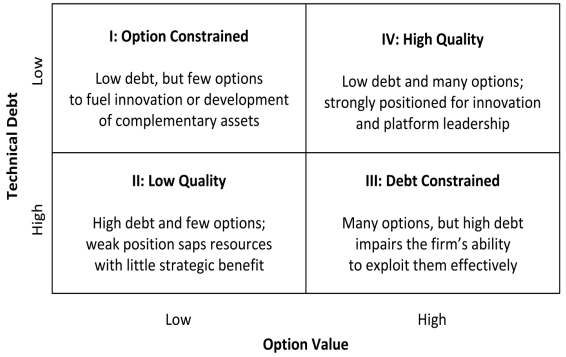


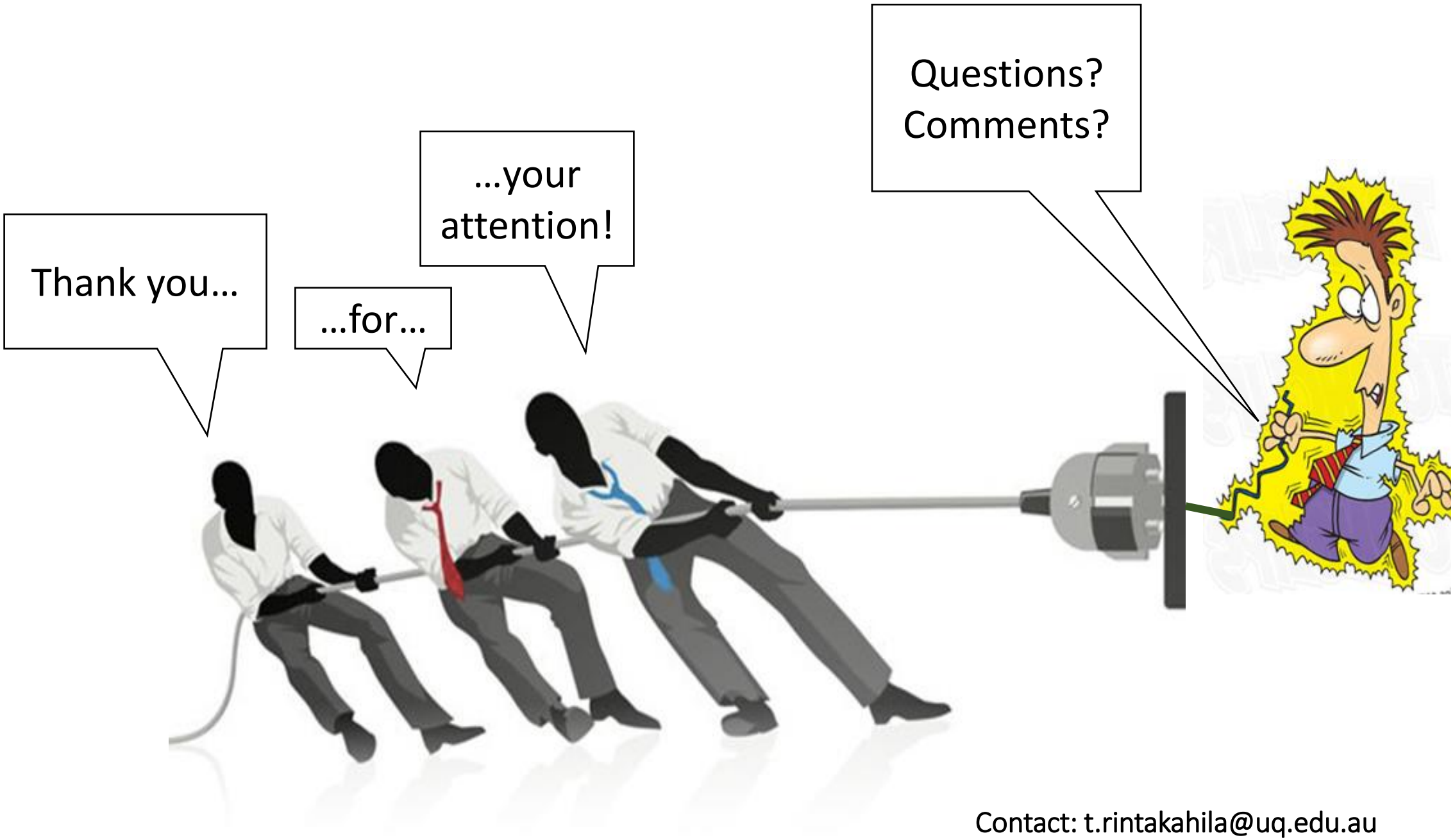












Contact: t.rintakahila@uq.edu.au

References

- Arvidsson, V., Holmström, J., and Lyytinen, K. 2014. "Information Systems Use as Strategy Practice: A Multi-Dimensional View of Strategic Information System Implementation and Use," Journal of Strategic Information Systems (23:1), Elsevier B.V., pp. 45–61.
- Furneaux, B., and Wade, M. 2011. "An Exploration of Organizational Level Information Systems Discontinuance Intentions," *MIS Quarterly* (35:3), pp. 573–598.
- Furneaux, B., and Wade, M. 2017. "Impediments to Information Systems Replacement: A Calculus of Discontinuance," *Journal of Management Information Systems* (34:3), pp. 902–932.
- Kelly, S., Gibson, N., Holland, C. P., and Light, B. 1999. "A Business Perspective of Legacy Information Systems," *Communications of AIS* (2:Focus Issue on Legacy Information Systems And Business Process Change).
- Lyytinen, K., and Newman, M. 2015. "A Tale of Two Coalitions - Marginalising the Users While Successfully Implementing an Enterprise Resource Planning System," Information Systems Journal (25:2), pp. 71–101.
- Mehrizi, M. H. R., Modol, J. R., and Nezhad, M. Z. 2019. "Intensifying to Cease: Unpacking the Process of Information Systems Discontinuance," MIS Quarterly (43:1), pp. 141–165.
- Rinta-Kahila, T, Penttinen, E. & Lyytinen, K., 'Getting Trapped in Technical Debt: Socio-Technical Analysis of a Legacy System's Replacement', (2023) MIS Quarterly, 47 (1). doi: https://doi.org/10.25300/MISQ/2022/16711
- Rolland, K. H., Mathiassen, L., and Rai, A. 2018. "Managing Digital Platforms in User Organizations: The Interactions between Digital Options and Digital Debt," Information Systems Research (29:2), pp. 419–443.