Strategic IT management - 37E00200

Technical debt and digital options

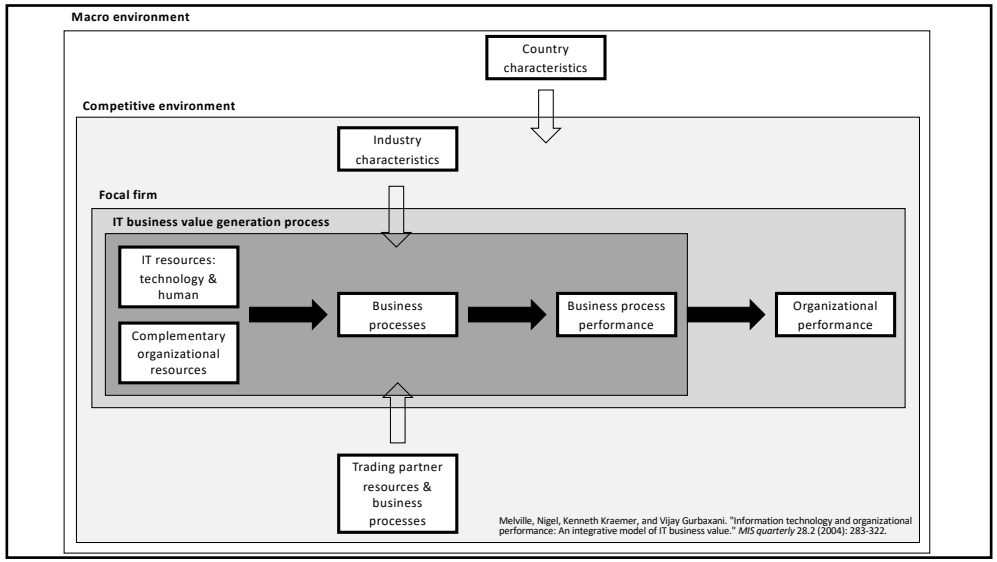
Esko Penttinen

Associate Professor, Information Systems Science, Aalto University School of Business

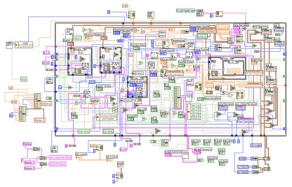
Chairman, XBRL Finland

Director, Real-Time Economy Competence Center

1



2



Functional SAP ERP Modules:

- Human Resource Management (SAP HRM), also known as Human Resource (HR)
- Production Planning (SAP PP)
- Material Management (SAP MM)
- Financial Supply Chain Management (SAP FSCM)
- Sales and Distribution (SAP SD)
- Project System (SAP PS)
- Financial Accounting and Controlling (SAP FICO)

Source: SAP

Software components can include:

- Content management systems (CMS)
- Customer relationship management (CRM)
- Enterprise resource planning (ERP)
- Operating systems
- Web servers

Source: IBM

3

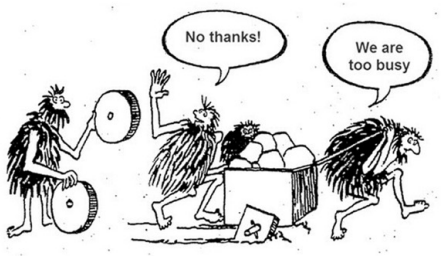
Agenda – learning outcomes

1. Understand the principles of technical debt and digital options; grasping the Design Capital framework
2. Through the presentation of two case studies…
   - … learn to identify the potential for interdependencies between types of technical debt
   - … understand the effects of Robotic Process Automation on Design Capital

4

Defining technical debt

- Technical debt refers to future information technology (IT) maintenance obligations, most often through suboptimal shortcuts taken in information systems' development
- Slows down operations, compromises reliability, and stifles innovation if left unmanaged



5

Technical debt – a measure of technical inertia

- Manifests at various levels (code, architecture, documentation)
- Often a result of evolutionary development practices
- 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

6

Types of technical debt

Type of technical debt	Description
Code debt	… covers problems that make source code hard to understand, maintain, and read
Test debt	… relates to lack of testing but also poorly executed tests and inappropriate testing
Requirements debt	… accumulates from trade-offs in what requirements are met, when, and how; it can arise if some needs are not fully addressed in the final system
Infrastructure debt	… refers to use of out-dated technologies and negligence of transition from old systems to new ones
Architecture debt	… accrues through suboptimal choices (e.g., spaghetti-like architectures) in the software architecture that increase complexity and decrease stability
Data debt	… refers to data that is incorrectly formatted, inaccurate, redundant, or absent altogether
Documentation debt	… accrues when developers neglect clear indications to others of their decision processes or the changes made
People debt	… manifests in insufficient training and hiring practices, and gaps in knowledge distribution

7

Origins of technical debt

- Intentional vs. unintentional
  - *Intentional debt* can be seen as investment in an asset that should yield gains, usually short-term benefits (e.g., competitive edge)
  - *Unintentional debt* results from poor management, lack of understanding, and/or accidental mistakes related to the type of technical debt in question
- Debt's internal vs. external sources
  - *Internal technical debt* stems from the organization's internal actions (e.g., resorting to compromises)
  - *External technical debt* originates from passing of time and general technological progress as information systems become obsolete and wieldy

8

Managing technical debt (Rolland et al. 2018)

- Planting technical debt
  - Taking shortcuts in development
  - Implementing new systems
  - Customizing systems

- Evaluating technical debt
  - Charting IT architecture
  - Estimating debt
  - Identifying interdependencies
  - Making plans

- Resolving technical debt
  - Rewriting code
  - Removing customizations
  - Producing documentation
  - Replacing/discontinuing old systems

9

You can't escape technical debt!

- Anything you do accumulates technical debt: e.g., removing special IT functionality resolves debt but implementing new one plants debt

→ key is to be aware of it and actively manage it!

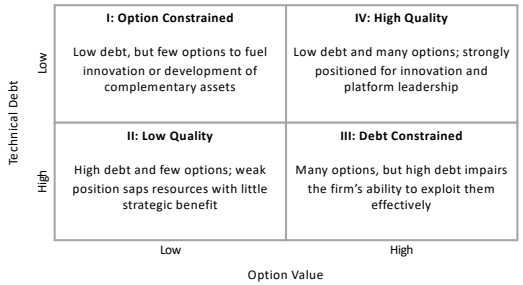
10

Pairing technical debt with digital options

- Technical debt
  - Technical debt refers to future information technology (IT) maintenance obligations, most often through suboptimal shortcuts taken in information systems' development
- Digital options
  - IT option value is a measure of the breadth of opportunities afforded by the firm's IT and is enhanced by IT architectures that enable designers to combine components within or across layers, cultivate or attract external partners (e.g., by providing application programming interfaces or software development kits), and launch innovative digital offerings

11

Design capital

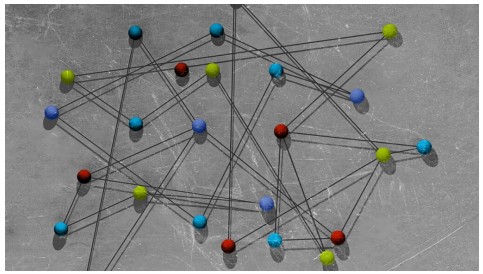


So, digital options generate technical debt…

12

…but can debt generate more debt too?

- Do you think there are interdependencies among different types of technical debt?
- If so, which types and why?



13

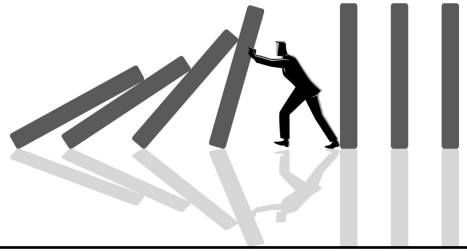
Study on types of technical debt and their interdependencies

Mäki, N., Penttinen, E. & Rinta-Kahila, T. A Domino Effect: Interdependencies among Different Types of Technical Debt, *56th Hawaii International Conference on System Sciences (HICSS), 2023*

14

Motivation & research question

- Motivated by our observations of vicious cycles of technical debt, we study the accrual of technical debt and the accrual's dynamic nature
- RQ: "What kinds of interdependencies exist among different types of technical debt?"



15

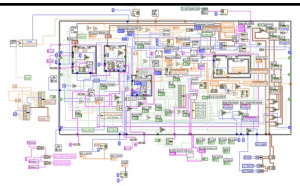
Method, Data collection and Analysis

- Exploratory single-case study at a Finnish manufacturing site
  - Follow-up on an earlier research project on system discontinuance
- Data collection anchored to the design processes of two specific types motor
  - Process involving creating and fulfilling orders, and product and design development in general
- Inductive 1st round analysis on nine semi-structured interviews (130 pages of text) to get a sense of occurrence of technical debt types
- Systematic 2nd round analysis on deliberativeness of debt accrual, and its causes and consequences
- Specific 3rd round analysis of the interdependencies by re-examining every segment of data where two or more technical debt types were indicated together

Informant	Role	Interview length
Chris	IS manager	66 min
Bob	Manager, engineering tools and processes	69 min
Mark	Head of product platforms and R&D	56 min
John	Team leader, mechanical-design engineer	74 min
Alex	System engineer	64 min
Tom	Team leader and product-owner, R&D	65 min
Matt	Senior software-development engineer	80 min
Dave	Project manager	57 min
Luke	Manager, sales tools and processes	46 min

16

Findings



- Quite unsurprisingly, evidence of various types of technical debt was found
  - High **architectural** complexity, large number of different systems, lack of smooth integration, and excessive system customization were said to slow down work processes, disrupt data flows, and make updates difficult to execute
  - Issues of data formatting signaled also of **data debt**, with some interviewees seeing problems in transferring data between separate locations, plants, and facilities
  - **Infrastructure debt** manifested mainly in terms of the issues arising because legacy systems were excessively interwoven in the process
  - **People debt** was mentioned mainly in connection with the older legacy systems as their maintenance and use require a very specific skillset, possessed by only a few people at the case company and even in the world

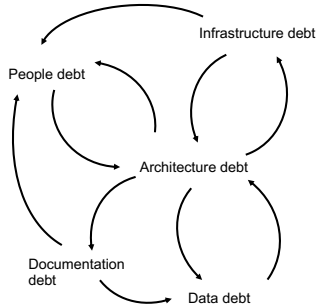
17

Findings on interdependencies

Types of technical debt and…	… their observed interdependence
Architecture debt and infrastructure debt	The reason for the ever-increasing number of systems (architecture debt) was the fact that the company was accumulating infrastructure debt: it was rarely able to fully remove its legacy systems when implementing new replacement systems.
Architecture debt, people debt, and documentation debt	Employees tended to specialize in only a few of the tools and systems and the cause for this was that, with so many tools to keep up with (and, furthermore, some of them being difficult to operate and update), it is easier to confine oneself to a single specific tool. This, in turn, results in documentation debt as minor-seeming changes were not well documented.
Architecture debt and data debt	The complex architecture demands extensive manual effort for entering data in the course of the motor-design process. Master data-entry point was sometimes unknown or inaccessible, thanks again to the information-systems architecture being so serpentine.
Documentation debt, infrastructure debt, and people debt	Individuals' choices of how thoroughly and methodically to document changes, processes, etc. is making it harder to train new employees at the case company. Prevalence of old systems further aggravates these difficulties.
Documentation debt and data debt	In the absence of clear guidelines on where and how to save data, they had established their own ways of working. With no comprehensive way of storing all the data for the design process, and with people generally operating as they see fit, it can be nearly impossible to locate particular data when needed.

22

Empirically observed interdependencies among types of technical debt



23

Contribution

- Technical debt can “infect” information systems further, in a seemingly unending loop
  - Prior research has primarily examined such relationships in case of single type of technical debt (for instance, complex architectures leading to even greater complexity)
  - Our study reveals contagious effects across types of technical debt
- Architecture debt identified as the main culprit in fueling (and being fueled by) infrastructure debt, data debt, people debt, and documentation debt
  - Our study identifies heavy customization and large number of different systems as root causes for complex architectures

24