UiO : University of Oslo

IN5431 – spring 2022

IT Architecture

Note: All images in the presentation are copyrighted and cannot be used or redistributed separately

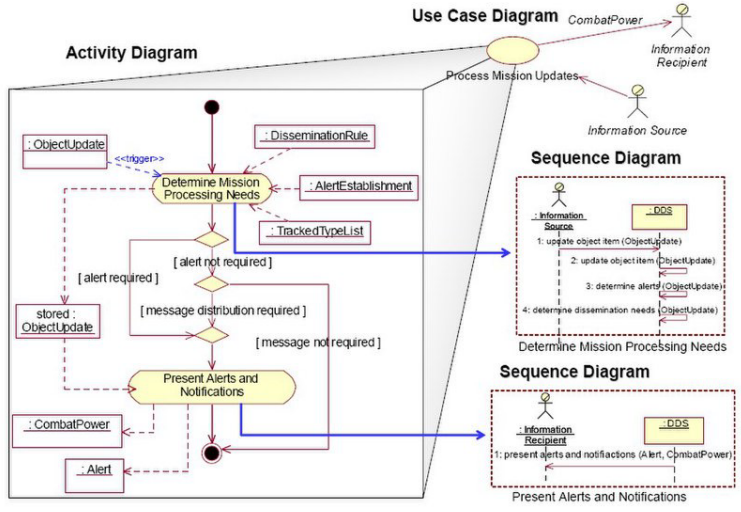
Agenda

- Introduction
- Architecture work – with examples
- One architecture approach: enterprise architecture
  - In theory
  - In practice (case study)
- On abstractions – and generic advice
- Summary

Introduction

> As the size and complexity of software systems increase, the design and specification of overall system structure become more significant issues than the choice of algorithms and data structures of computation.
>
> Structural issues include the organization of a system as a composition of components; global control structures; the protocols for communication, synchronization, and data access; the assignment of functionality to design elements; the composition of design elements; physical distribution; scaling and performance; dimensions of evolution; and selection among design alternatives. This is the software architecture level of design.

From *Mary, Shaw, and Garlan David. "Software architecture: perspectives on an emerging discipline." Prentice-Hall (1996).*



What is (IT) architecture?

No universal definition. Two important variants:

1) "The software architecture of a computing system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both."
(Carnegie Mellon University)

"System" depends entirely on the chosen perspective – can be nationwide or one single component within a deployable application

2) "Architecture is the set of design decisions that must be made early in the project"
(Martin Fowler and Ralph Johnson)

Sources:
(1) https://resources.sei.cmu.edu/asset_files/FactSheet/2010_010_001_513810.pdf
(2) https://ieeexplore.ieee.org/document/1231144

This distinction is essential

The descriptive alternative

"The software architecture of a computing system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both."

Here, architecture is about **understanding** and **documenting** (implicit or explicit) decisions – not making them.

Consequences for architecture work:

- Focuses on modeling, documentation and presentation
- Not really about (technology) management

This distinction is essential

The descriptive alternative

"The software architecture of a computing system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both."

Here, architecture is about **understanding** and **documenting** (implicit or explicit) decisions – not making them.

Consequences for architecture work:

- Focuses on modeling, documentation and presentation
- Not really about (technology) management

The normative alternative

"Architecture is the set of design decisions that must be made early in the project"

Here, architecture is about **decisions**.

Consequences for architecture work:

- Focus is on decision making – which involves a broad set of skills
- Largely about (technology) management

This distinction is essential

The descriptive alternative

"The software architecture of a computing system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both."

Here, architecture is about **understanding** and **documenting** (implicit or explicit) decisions – not making them.

Consequences for architecture work:

- Focuses on modeling, documentation and presentation
- Not really about (technology) management

The normative alternative

"Architecture is the set of design decisions that must be made early in the project"

Here, architecture is about **decisions**.

Consequences for architecture work:

- Focus is on decision making – which involves a broad set of skills
- Largely about (technology) management

So what does *an architect* do? And what is *good architecture*?

A fundamental discussion





What are the forces affecting the construction?

A fundamental discussion



The forces affecting a bridge are:

- Wind
- Weight
- Temperature
- Water
- Erosion
- Corrosion
- Space constraints
- Human cognitive capabilities
- Human behavior
- Human aesthetical preferences

A fundamental discussion



The forces affecting computer software are

- ~~Wind~~
- ~~Weight~~
- ~~Temperature~~
- ~~Water~~
- ~~Erosion~~
- ~~Corrosion~~
- Network speed and capacity
- CPU and memory speed and capacity
- Human cognitive capabilities
- Human behavior
- Human aesthetical preferences

A fundamental discussion





The "engineering" of software engineering is very different from other engineering disciplines. The forces affecting computer systems are dominated by human cognitive capability, human behavior and human aesthetical preferences.

A fundamental discussion





This again means that the "quality" in software engineering – and in software architecture - is mainly given by human judgment. So what is **good** architecture?

> **I have sometimes seen software developers design systems with beautiful architectures, then voice their resistance to software architecture. This resistance may stem from resistance to bureaucracy-intensive up-front design processes , to pompous architects, or to having been forced to waste time making diagrams instead of systems.**
>
> **Fortunately, none of these problems need to be a part of creating software architecture. Job titles, development processes , and engineering artifacts are separable, so it is important to avoid conflating the job title “architect,” the process of architecting a system, and the engineering artifact that is the software architecture.**

"Just Enough Software Architecture", av George Fairbanks

Architecture work

Architecture taxonomy (according to TOGAF)

- **Business Architecture** defines the business strategy, governance, organization, and key business processes
- The **Data Architecture** describes the structure of an organization's logical and physical data assets and data management resources
- The **Application Architecture** provides a blueprint for the individual applications to be deployed, their interactions, and their relationships to the core business processes of the organization
- The **Technology Architecture** describes the logical software and hardware capabilities that are required to support the deployment of business, data, and application services; this includes IT infrastructure, middleware, networks, communications, processing, standards, etc.

Enterprise Architecture

Source: https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap02.html



Example 1: Development

Thale is an application architect working in the IT department of her employer, Goodumbrellas Inc. She is currently engaged in a project working to integrate the logistics system of Goodumbrellas with their customer database, to allow customers to track their orders through a web interface. This is expected both to increase customer satisfaction and reduce the number of support requests.

Thales role is to assist the hired developers with technical questions, and ensure the quality of their work. The developers are friendly and energic, but they understand way to little of Goodumbrellas complex logistics system to work independently. She also helps designing REST-based service interfaces according to the current standards of Goodumbrellas.

Thale will work on...

- Systems design
- Problem solving
- Professional guidance
- Security
- Documentation





Example 2: Standardization

Finn is an enterprise architect at Goodumbrellas Inc. After being harshly criticised in the executive meeting, the IT director has initiated a programme (i.e. a set of projects) named "Economies of scale". The goal is to reduce complexity and consolidate the plethora of small, independent applications into a set of shared, standardized and integrated systems.

To ensure correct implementation, all exceptions from the standards must be validated by an Architecture Governance Board led by Finn. The number of applications for exceptions is unexpectedly high, and the waiting time for a decision soon exceeds 8 weeks.



Example 2: Standardization

Luke is the new project manager of www.umbrella-heaven.com They plan to – finally - launch the first complete version to the public in 6 weeks.

The only important feature missing is an integration between the web site and the logistics system. Luke has found a local provider developing the necessary integration with a custom, Java-based component to integrate with the logistics system. The problem is that the "Economies of scale" program has decided that all internal systems should communicate through the enterprise service bus. The necessary integrations require specific certifications on the designated service bus product, and all available personell are booked for the next three months. Luke is offered the option of applying for an exception, where waiting times are only eight weeks...

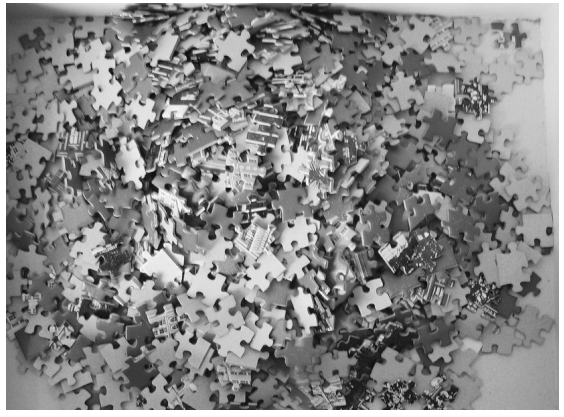


Luke's response

Finn will work on...

- Conflict resolution
- Difficult dilemmas
- Politics
- Systems design



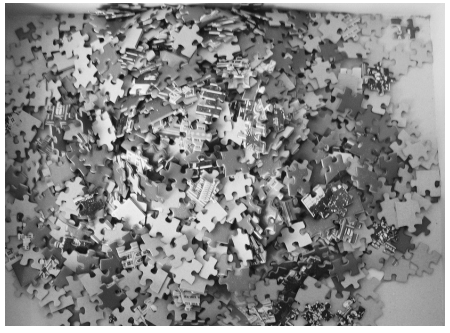


Example 3: Big decisions

After the launch of Umbrella Heaven, Goodumbrella establishes two divisions: Business Customers and Personal Customers. The Business division targets physical retailers, while the Personal division is responsible for direct sales through www.umbrella-heaven.com.

The core systems of the Business Customers division is the CRM (Customer Relational Management) system, which is used also for order management, and the billing system to ensure invoices are processed correctly. The CRM system is working well, while the billing system is obsolete and troublesome. Mia is responsible for choosing a new, custom billing system to replace the old.

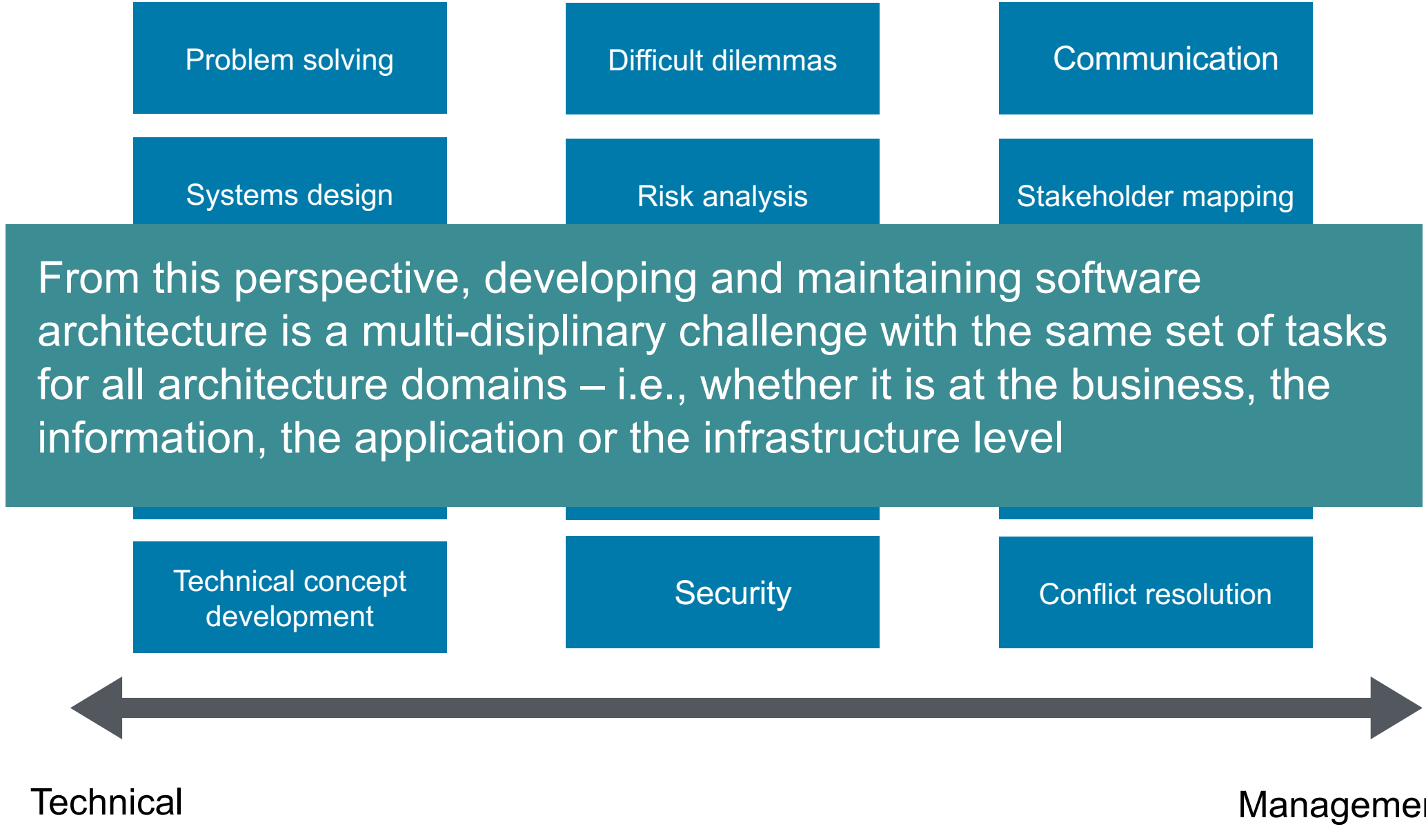
However, the Personal Customers-division have their own billing system which also contains an order management module, and the new director of Personal Customers, Luke, is strongly advocating establishing their system as a company standard to ensure an integrated view of the order status. The challenge is: the CRM module of Business is strongly integrated with the ordering system - how to proceed?



Mia will work with...

- Stakeholder mapping
- Technical concept development
- Risk analysis
- Business case
- Politics
- Systems design
- Procurement
- Communication
- Test and deployment


Problem solving
Difficult dilemmas
Communication
Systems design
Risk analysis
Stakeholder mapping
Test and deployment
Guidance
Politics
Documentation
Business case
Procurement
Technical concept development
Security
Conflict resolution
Technical
Management


Problem solving
Difficult dilemmas
Communication
Systems design
Risk analysis
Stakeholder mapping
From this perspective, developing and maintaining software architecture is a multi-disiplinary challenge with the same set of tasks for all architecture domains – i.e., whether it is at the business, the information, the application or the infrastructure level
Technical concept development
Security
Conflict resolution
Technical
Management

How you will meet architecture

Which **systems** or **components** do we need?
How should we structure our **knowledge**?
What should we **generalize** and what should be allowed to be **specific?**
How should our systems **communicate** - and with whom?
How should we **document** our systems?
How should we **structure** our systems to ensure maintainability and security?

Enterprise architecture

Enterprise Architecture

"The purpose of enterprise architecture is to optimize across the enterprise the often fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy"

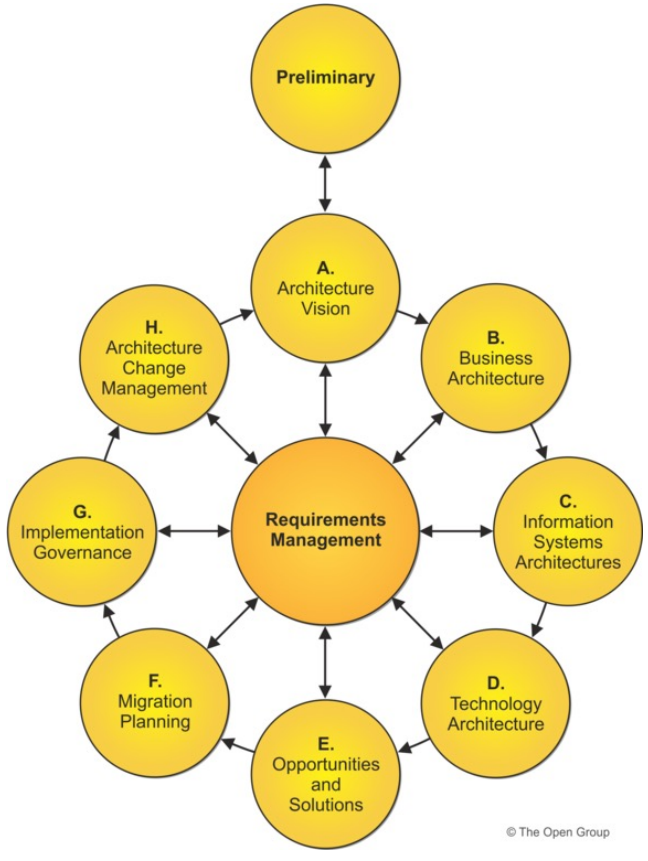
Harrison, Rachel. *TOGAF 9 Foundation Study Guide*. Van Haren, 2013.

"In the years since John Zachman originated the field of Enterprise Architecture (EA) in his seminal 1987 article for *IBM Systems Journal*, EA has achieved a surprisingly paltry level of success.

Yes, Enterprise Architects have used various frameworks and other tools to document how their organization operates, often with meticulous detail. But to what end? The cost savings and responsiveness benefits that EA has purported to deliver have been few and far between. Stories of stalled or misdirected EA initiatives vastly outnumber bona fide examples of EA efforts leading to measurable business value."

https://forbes.com/sites/jasonbloomberg/2014/07/11/is-enterprise-architecture-completely-broken/?sh=2b181b633710

TOGAF: The Architecture Development Method (ADM)



Excerpt from step C "Information Systems Architecture"

Develop a Target Description for the Application Architecture, to the extent necessary to support the Architecture Vision, Target Business Architecture, and Target Data Architecture.

The scope and level of detail to be defined will depend on the relevance of the application elements to attaining the Target Architecture Vision, and on whether architectural descriptions exist. To the extent possible, identify the relevant Application Architecture building blocks, drawing on the Architecture Repository

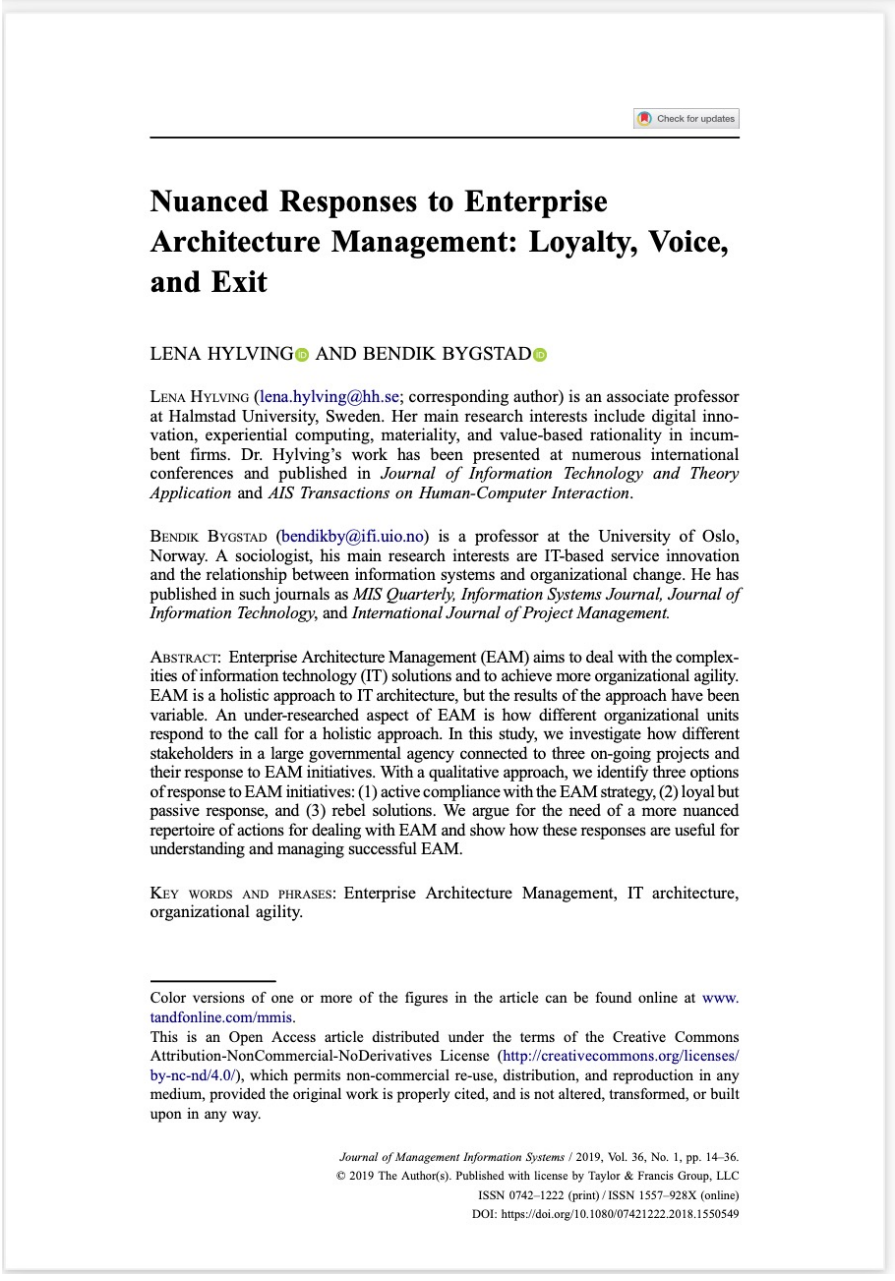


"What if we take on step back and proceed with a modular solution?"

© Børge Lund – with permission

Why is enterprise architecture hard?

Hylving, Lena, and Bendik Bygstad. "Nuanced responses to Enterprise architecture management: Loyalty, voice, and exit." *Journal of Management Information Systems* 36.1 (2019): 14-36.



Check for updates

Nuanced Responses to Enterprise Architecture Management: Loyalty, Voice, and Exit

LENA HYLVING AND BENDIK BYGSTAD

LENA HYLVING (lena.hylving@hh.se; corresponding author) is an associate professor at Halmstad University, Sweden. Her main research interests include digital innovation, experiential computing, materiality, and value-based rationality in incumbent firms. Dr. Hylving's work has been presented at numerous international conferences and published in *Journal of Information Technology and Theory Application* and *AIS Transactions on Human-Computer Interaction*.

BENDIK BYGSTAD (bendikby@ifi.uio.no) is a professor at the University of Oslo, Norway. A sociologist, his main research interests are IT-based service innovation and the relationship between information systems and organizational change. He has published in such journals as *MIS Quarterly, Information Systems Journal, Journal of Information Technology*, and *International Journal of Project Management*.

ABSTRACT: Enterprise Architecture Management (EAM) aims to deal with the complexities of information technology (IT) solutions and to achieve more organizational agility. EAM is a holistic approach to IT architecture, but the results of the approach have been variable. An under-researched aspect of EAM is how different organizational units respond to the call for a holistic approach. In this study, we investigate how different stakeholders in a large governmental agency connected to three on-going projects and their response to EAM initiatives. With a qualitative approach, we identify three options of response to EAM initiatives: (1) active compliance with the EAM strategy, (2) loyal but passive response, and (3) rebel solutions. We argue for the need of a more nuanced repertoire of actions for dealing with EAM and show how these responses are useful for understanding and managing successful EAM.

KEY WORDS AND PHRASES: Enterprise Architecture Management, IT architecture, organizational agility.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/mmis.
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (http://creativecommons.org/licenses/by-nc-nd/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited, and is not altered, transformed, or built upon in any way.

Journal of Management Information Systems / 2019, Vol. 36, No. 1, pp. 14–36.
© 2019 The Author(s). Published with license by Taylor & Francis Group, LLC
ISSN 0742–1222 (print) / ISSN 1557–928X (online)
DOI: https://doi.org/10.1080/07421222.2018.1550549

Enterprise Architecture Challenges

Enterprise Architecture should "*optimize across the enterprise the often fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy*" (TOGAF)

EAM initiatives are usually run in a top-down manner; a central team of enterprise architects run the process of developing and implementing the architecture, in co-operation with business man- agers and IT specialists

…

One core issue is: organizations are not "architected". They grow and change organically as they adapt to outer and inner pressures and changes.

(Hylving and Bygstad, 2019)

- EAM has been hailed as a holistic and feasible approach for organizations with complex and fragmented IT portfolios
- It has also been proposed as a means to increase organizational agility and emphasis on organizational aspects has been highlighted
- However (...), many EAM initiatives have been disappointments; they are not necessarily outright failures, but they seem to go on forever, without concrete results



(Hylving and Bygstad, 2019)

Theoretical lens: customer's or members responses to [perceived negative] change (Hirschman)

Members of customers can choose one of three responses:

- **Voice**: “any attempt at all to change, rather than to escape from ... with the intention of forcing a change in management”. This can be manifested in different ways, for example raising the voice and articulating the opinion about the change.
- **Exit**: the result of the exit option can be “revenues drop, membership declines, and management is impelled to search for ways and means to correct whatever faults have led to exit”
- **Loyalty:** passively waiting for conditions to improve.

See also: Hirschman, Albert O. *Exit, voice, and loyalty: Responses to decline in firms, organizations, and states*. Vol. 25. Harvard university press, 1970.

In EAM context

- **Voice**: actively oppose or challenge the policies. The implementation requires more work.
- **Exit** means to ignore it or even go against EAM. The exit option means that no communication exists between the initiative and the people that are affected by the initiative.
- **Loyalty**: comply with the EAM policies and blueprints. If employees are loyal to an initiative, for example an EAM initiative, the implementation of the initiative can be done with little resistance and fuzz

Case study

	Downward governance	Upward feedback	Response to EAM
Facility project	Frequent	Frequent	Critical compliance/Voice
Billing	One-way	None	Loyal
API	None	None	Rebel/Exit

Theoretical contributions from case study

- A deeper organizational understanding of EAM is required
- EAM is based on the assumption that it is analytically and managerially possible to control everything at the operative level.
- Particularly in a turbulent environment it is clear that this is too ambitious and may even harmful to innovation. Therefore, the key to successful EAM is not necessarily compliance, but flexibility. Our findings highlight that the EAM frameworks (i.e., TOGAF and the others) need to be supplemented with the insight that successful socio-technical change is the result of not only top-down implementation, but also of continuous learning and adjustment.

Practical contributions from case study

- EA managers should their managerial repertoire, in line with Bloomberg's[1)] argument that iterative learning and frequent feedback is a prerequisite for a successful EAM initiative
- While most of the EAM frameworks emphasize compliance as a success factor, our study highlights that other responses should not only be tolerated, but also be leveraged as opportunities for learning. The three response types should not be seen as resistance to be overcome, but should serve as input to a discipline which is in need of renewal

[1)] https://www.forbes.com/sites/jasonbloomberg/2014/07/23/agile-enterprise-architecture-finally-crosses-the-chasm/?sh=43e2a4be52a5

Abstractions

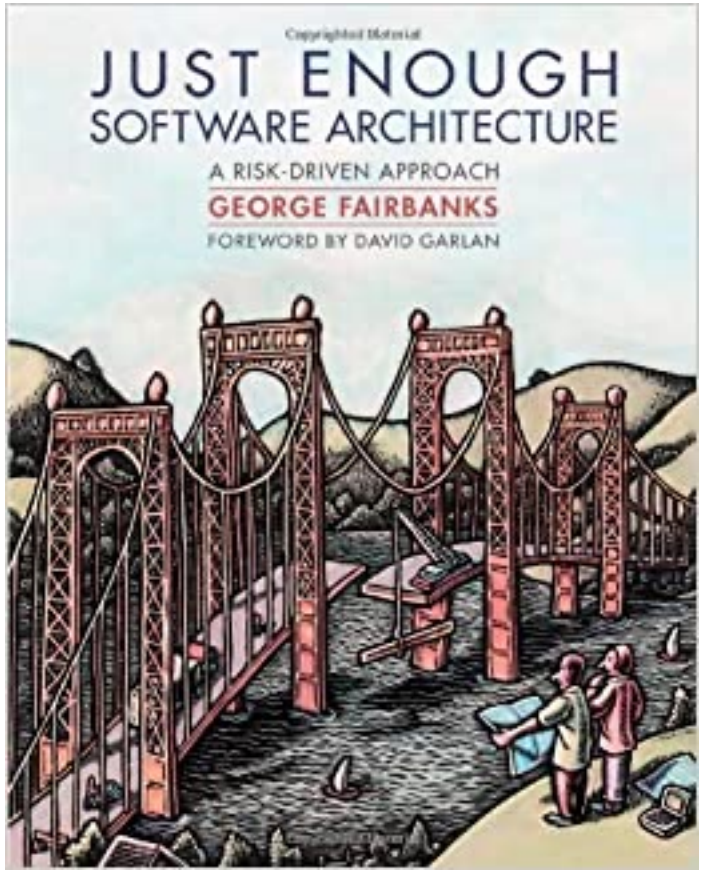
The Conceptual Model and Abstractions

- A large part of developing and maintaining Software Architecture is to find:
  - Map a part of the world to a conceptual model with the right level of generalization and abstraction
  - **Do not** aim to generalize and abstract as much as possible
  - Find the right vocabulary (terms and perspectives) to build a common understanding of business requirements
  - Align IT to the business – e.g., through "ubiquitous language" and Domain Driven Design

Note: there is a huge difference between:

- building software for "yourself" (e.g. for IT-professionals)
- entering a "greenfield" domain
- providing better tools for existing, complex domains such as healthcare or supply chain

Creating good architecture is about cooperation and compromise - **listen** to users and to other developers


JUST ENOUGH
SOFTWARE ARCHITECTURE
A RISK-DRIVEN APPROACH
GEORGE FAIRBANKS
FOREWORD BY DAVID GARLAN

Summary

Summary

- "Architecture is the set of design decisions that must be made early in the project"
- **But also:** "The software architecture of a computing system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both"
- Developing and maintaining software architecture is a multi-disciplinary challenge with the same set of tasks for all architecture domains – i.e., whether it is at the business, the information, the application or the infrastructure level
- The benefits of Enterprise Architecture are often much harder to achieve in practice than expected
- Creating good architecture is about cooperation and compromise

