The Problem

Dark Waters of Legacy IT

Agile! Agile! Agile!
Innovation!

Vision?
Strategic Management?
Architecture?
Reengineering?
Prologue:

What obviously does not work*

*when used in isolation
Hardcore Agility
Let’s create yet another unrealizable point solution!
Software/System Architecture only
Enterprise Architecture Management as Connector?

An isolated, bloated, Ivory Tower Discipline as Connector?
Separated Disciplines

Design Thinking!

Use Microservices!

Cloud is the Solution!

I am the Leader!

Let’s Architect!

Be Agile!
Agenda

1. Architectural Thinking
   1.1 Customer centric over technocratic
   1.2 Lean over bloated Frameworks
   1.3 Collaborative & Iterative over autocratic

4. Architectural Thinking and DDD
1. Architectural Thinking

1.1 Customer centric over technocratic
1.2 Lean over bloated Frameworks
1.3 Collaborative & Iterative over autocratic

4. Architectural Thinking and DDD
Vision Statement

„We bring disciplines together to build the missing bridges that enable organizations to adapt and thrive.“
Architectural Thinking Association®

› Non Profit Association
› “Förderung des architekturellen Denkens in Unternehmen”
› Truly open Architectural Thinking Framework®
› architectural-thinking.com/participate
What we do

“We form an interdisciplinary leadership team that creates wisdom how to connect disciplines.”

“We bring in the thought leaders of relevant disciplines to participate in our vision.”

“We define a compelling, simple AT framework where everybody wants to connect to.”
Clusters of Disciplines to be connected
1.1 Customer centric over technocratic

1. Architectural Thinking

1.1 Customer centric over technocratic
1.2 Lean over bloated Frameworks
1.3 Collaborative & Iterative over autocratic

2. Architectural Thinking and DDD
Who is the primary Customer of Architectural Thinking?

C-level Executives
Business Unit Leads

- CIO / CDO
- IT Solution Architects
- Development teams
- Innovation-management
- Process-management
- Computing Center
What's the Biggest Pain of Executives?

Too intransparent!
Too expensive!
Too slow!
Strategy implementation?
How can Architectural Thinking help?

- Intransparent!
  - Business Architecture

- Expensive!
  - Strategic Budgeting based on Capabilities

- Slow!
  - Prio by Strategic Fields of Action

- Strategy implement?
  - Governance based of Business Architecture
Business Architecture
Business Architecture
Makes Business People Accountable for Architecture and Drives IT
Capabilities represent the basic building blocks, or DNA, of a business.
Strategic Budgeting
Per Capability

2019

- Sales & Service
- Giro
- Investment
- Loan
- Cross Product
- Management & Support

- Innovate
- Optimize
- Run
Governance via Strategic Fields of Action
1.2 Lean Maps over bloated Frameworks

1. Architectural Thinking

1.1 Customer centric over technocratic

1.2 Lean over bloated Frameworks

1.3 Collaborative & Iterative over autocratic

2. Architectural Thinking and DDD
Lean Maps – Enterprise Level
Lean Maps – Solution Level
Connect Solution Requirements with Enterprise Level

Assign Requirements to Capabilities
1.3 Collaborative & Iterative over autocratic

1. Architectural Thinking

1.1 Customer centric over technocratic
1.2 Lean over bloated Frameworks
1.3 Collaborative & Iterative over autocratic

2. Architectural Thinking and DDD
Grassroots Democracy?
Autocratic Leadership?
What’s the Best Way to Make Strategic Decisions?
...its Always a Sound Combination!

Democratic, Bottom-up    Management, Top-down
Connect “Managing Disciplines to Enterprise Maps”

Vision Strategy

Budget

Governance

Project

Solution

Requirement

Code
“Architecture Coordinator” im Entwicklungsteam

› 80% of architectural Work happens in Solution Team
› Everybody is architect on micro level
› “Architecture Coordinator” accountable for
   › Solution Architecture as a whole
   › Integration with Enterprise Architecture
“Architecture Coordinator” Enterprise Level

› 20% of Architecture Work
› Optimizes across Solutions
› Enge Abstimmung mit AC Solution
Architectural Map as Core of Company wide Governance
1.3 Collaborative & Iterative over autocratic

1. Architectural Thinking

1.1 Customer centric over technocratic
1.2 Lean over bloated Frameworks
1.3 Collaborative & Iterative over autocratic

2. Architectural Thinking and DDD
Domain-driven design (DDD) is an approach to software development for complex needs by connecting the implementation to an evolving model.

Domain-driven design - Wikipedia
“Model” Domain Driven Design vs Business Architecture

Domain = Enterprise
A sphere of knowledge (ontology), influence, or activity. The subject area to which the user applies a program is the domain of the software;

Model = Capabilities, Value Streams, Information Objects
A system of abstractions that describes selected aspects of a domain and can be used to solve problems related to that domain;

Ubiquitous Language = Nope...Focus on strategic Level
A language structured around the domain model and used by all team members to connect all the activities of the team with the software.
Positioning Domain Driven Design in Discipline’s Cluster
Dr. Wolfgang Goebl
wolfgang.goebbl@architectural-thinking.com
architectural-thinking.com
Recommended Reading

[Ambler10]: S. Ambler: ‘Disciplined Agile Delivery’
[Eckstein18]: J. Eckstein: ‘Company-wide Agility’
[Westerman15]: ‘Leading Digital Turning Technology into Business Transformation’