UNIVERSITÄT DES SAARLANDES

30 YEARS DFKI Human Centric AI.

Modeling for Architects III: Project Forces, Architectural Concerns, and Decisions

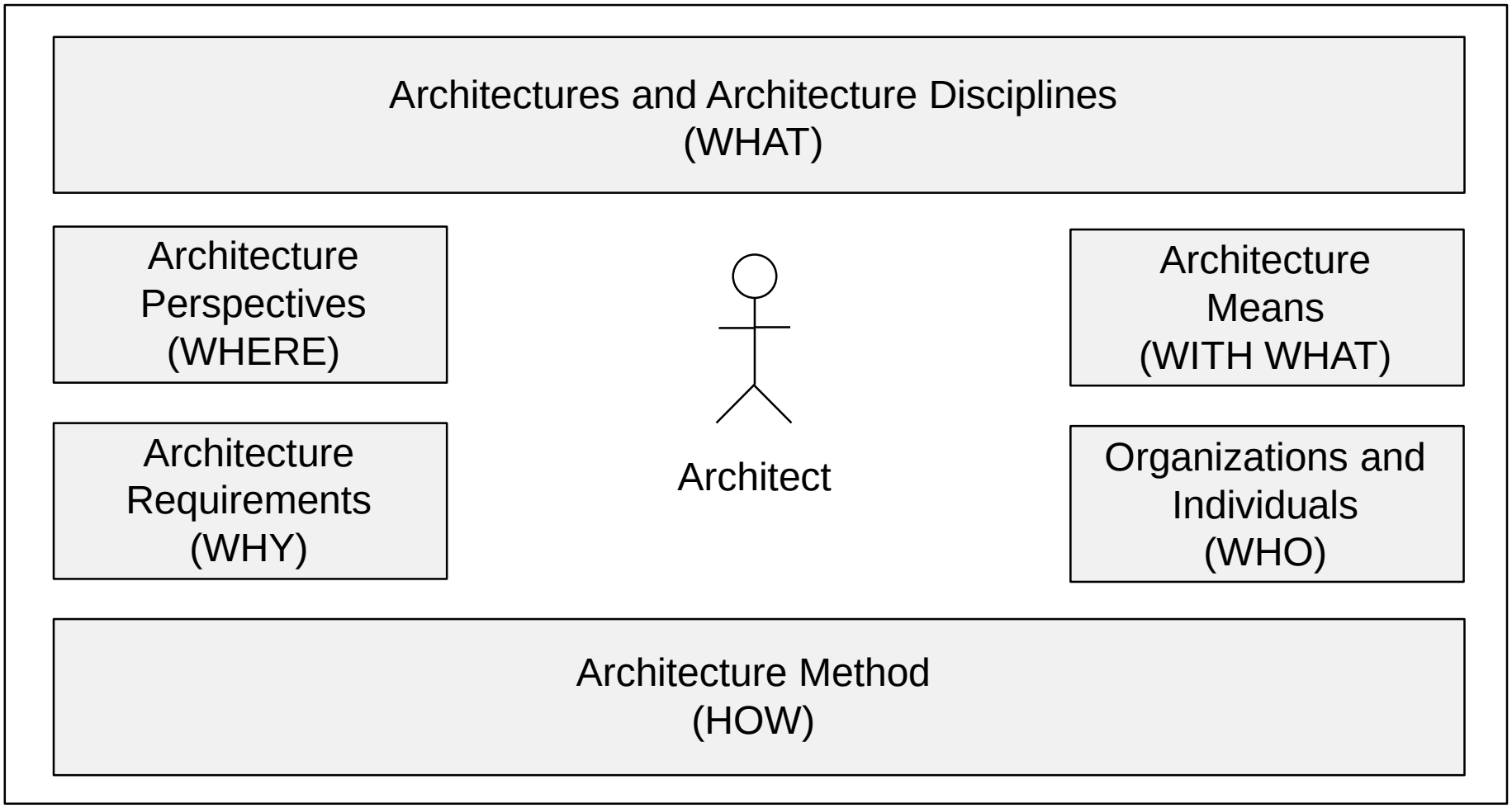
Architectural Thinking for Intelligent Systems

Winter 2019/2020

Prof. Dr. habil.Jana Koehler

Agenda

- Functional and non-functional requirements from an architectural perspective
- Controlling requirements to manage risk
- Understanding constraints
- Architectural concerns and decisions in ISO 42010
- Creating work products and documenting architectures
  - Relationship between application/solution architecture and enterprise architecture frameworks


Architectures and Architecture Disciplines
(WHAT)
Architecture
Perspectives
(WHERE)
Architecture
Means
(WITH WHAT)
Architecture
Requirements
(WHY)
Architect
Organizations and
Individuals
(WHO)
Architecture Method
(HOW)

Tutorial Assignment 3

- We begin to understand the various forces that influence our project and the concerns that we need to address.
- We create an initial list of decisions, which have to be taken and understand how they relate to each other.
- We explore potential decision alternatives and take an educated guess of the expected outcome.

Business as Usual?

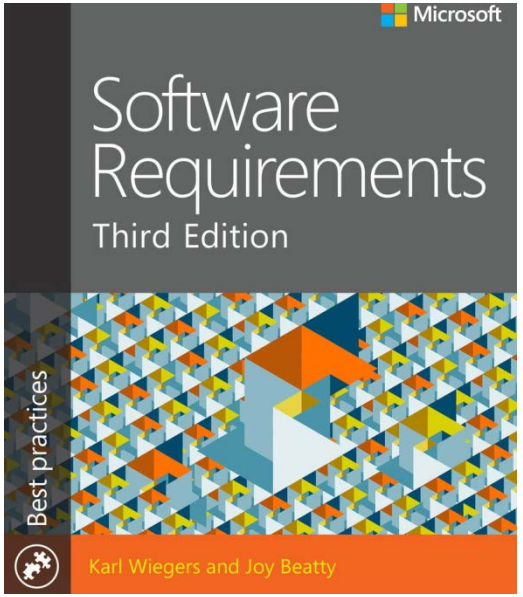
"The hardest single part of building a software system is deciding precisely what to build.

No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later."

*Brooks, Frederick P., "No Silver Bullet: Essence and Accidents of Software Engineering," Computer, Vol. 20, No. 4 (April **1987**) pp. 10-19.*

Requirements, requirements, requirements …

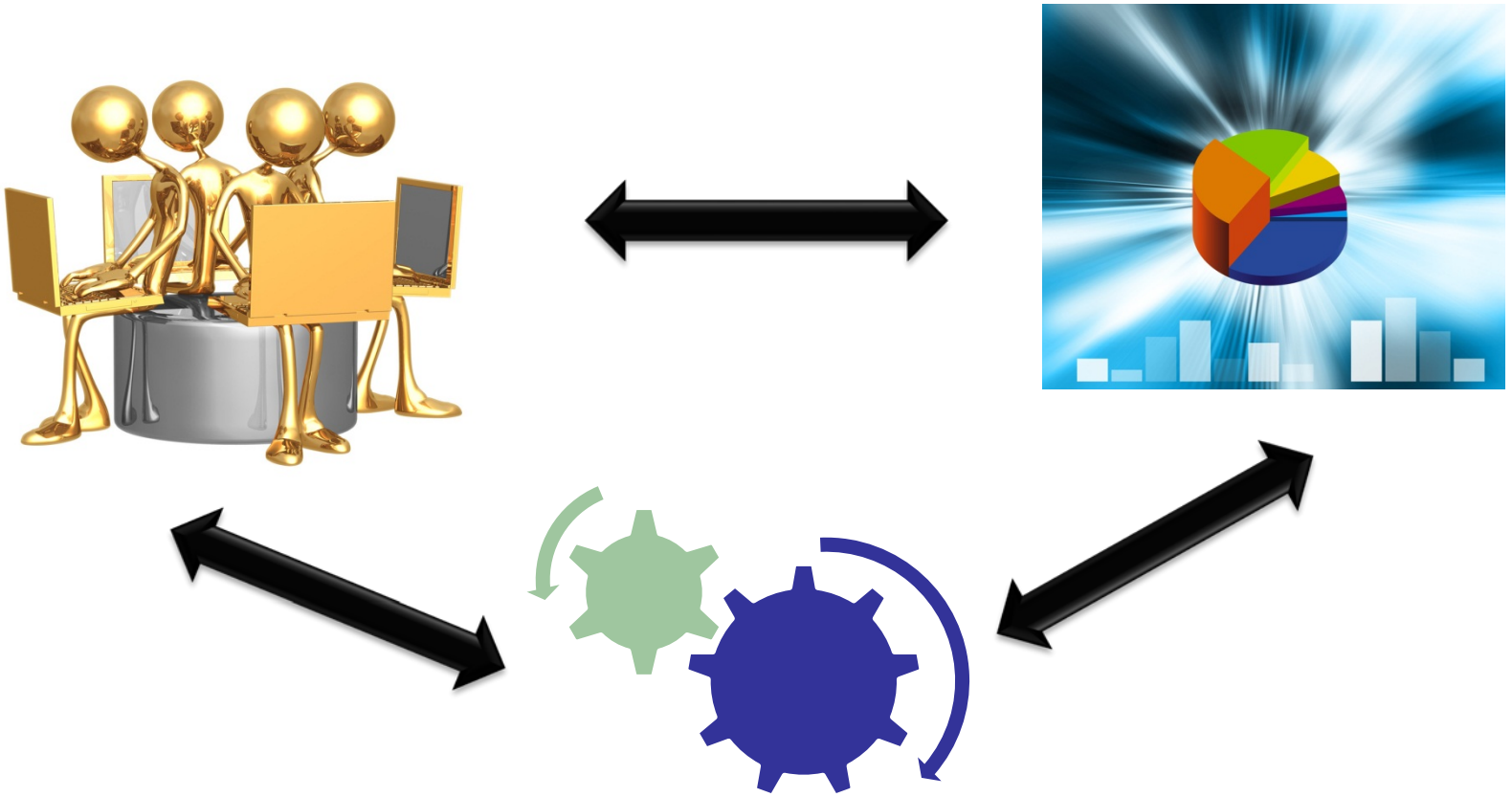
„The goal of requirements development is to accumulate a set of requirements that are good enough to allow your team to proceed with design and construction of the next portion of the product at an acceptable level of risk.“



Karl Wiegers and Joy Beatty Software Requirements

Key Requirements

- Interaction of users, data and business functions

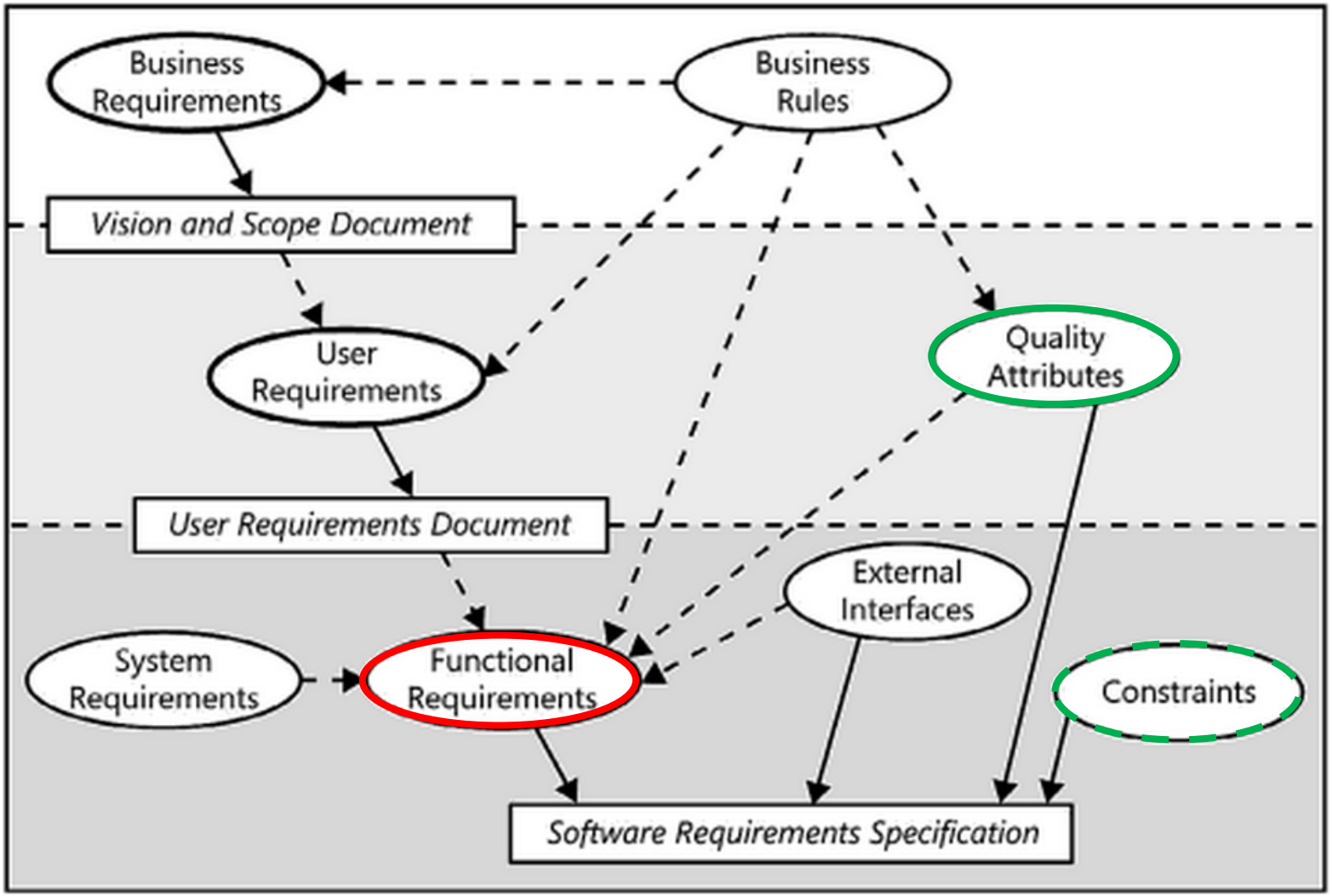


Inherent Difficulties during Requirements Elicitation

- Understanding stakeholders´needs is challenging
  - they do not really know what they want (in terms of the to-be-built system)
- Communication is complex, often unclear, suffers from the Business-IT gap
- Control of the software development process is difficult
- “Inseparable” concerns: everything seems to depend on everything

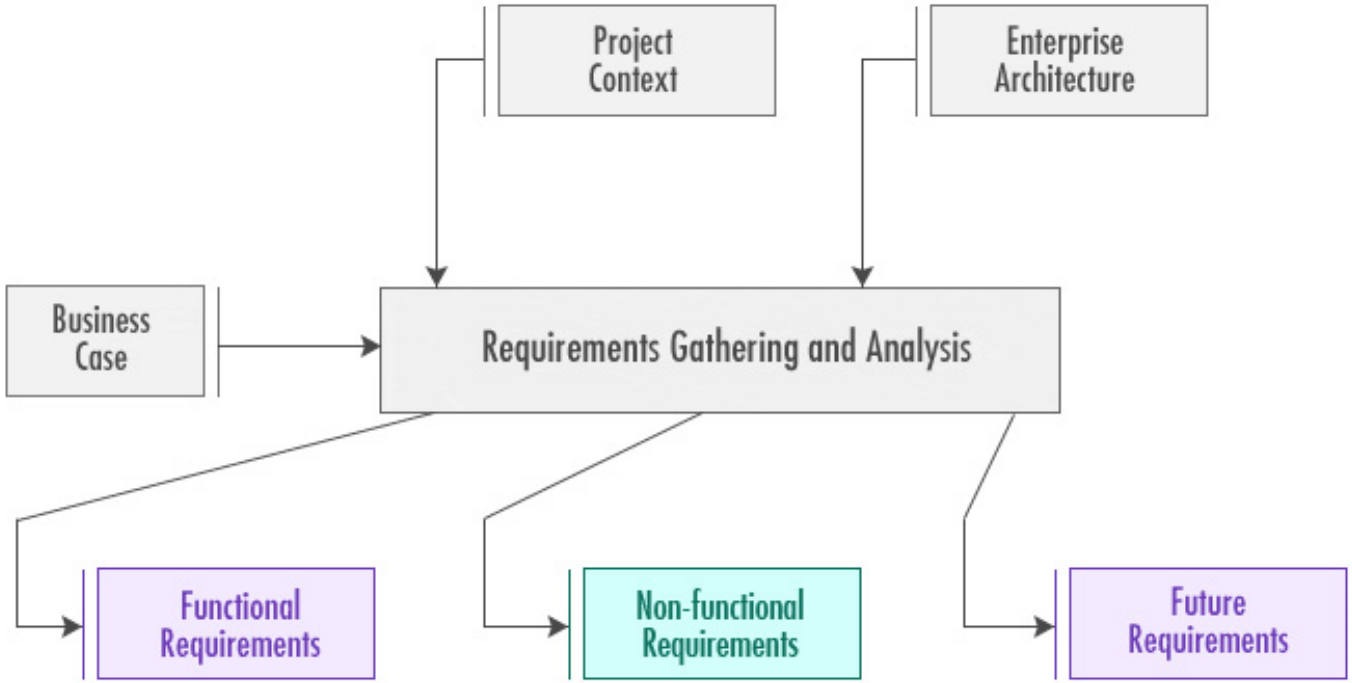
Random Difficulties during Requirements Elicitation

- Writing requirements at a later point in times when things become more clear
  - not helpful for developers (but: rapid prototyping)
- Contradictory interests when elicitating and documenting requirements:
  - Selling the future product (meet a marketing hype)
  - Do some general documentation based on software development practice and methods used
  - Writing contracts
- Insufficient efforts
- Undetected …



Wiegers/Beatty: Software Requirements

Types of Requirements



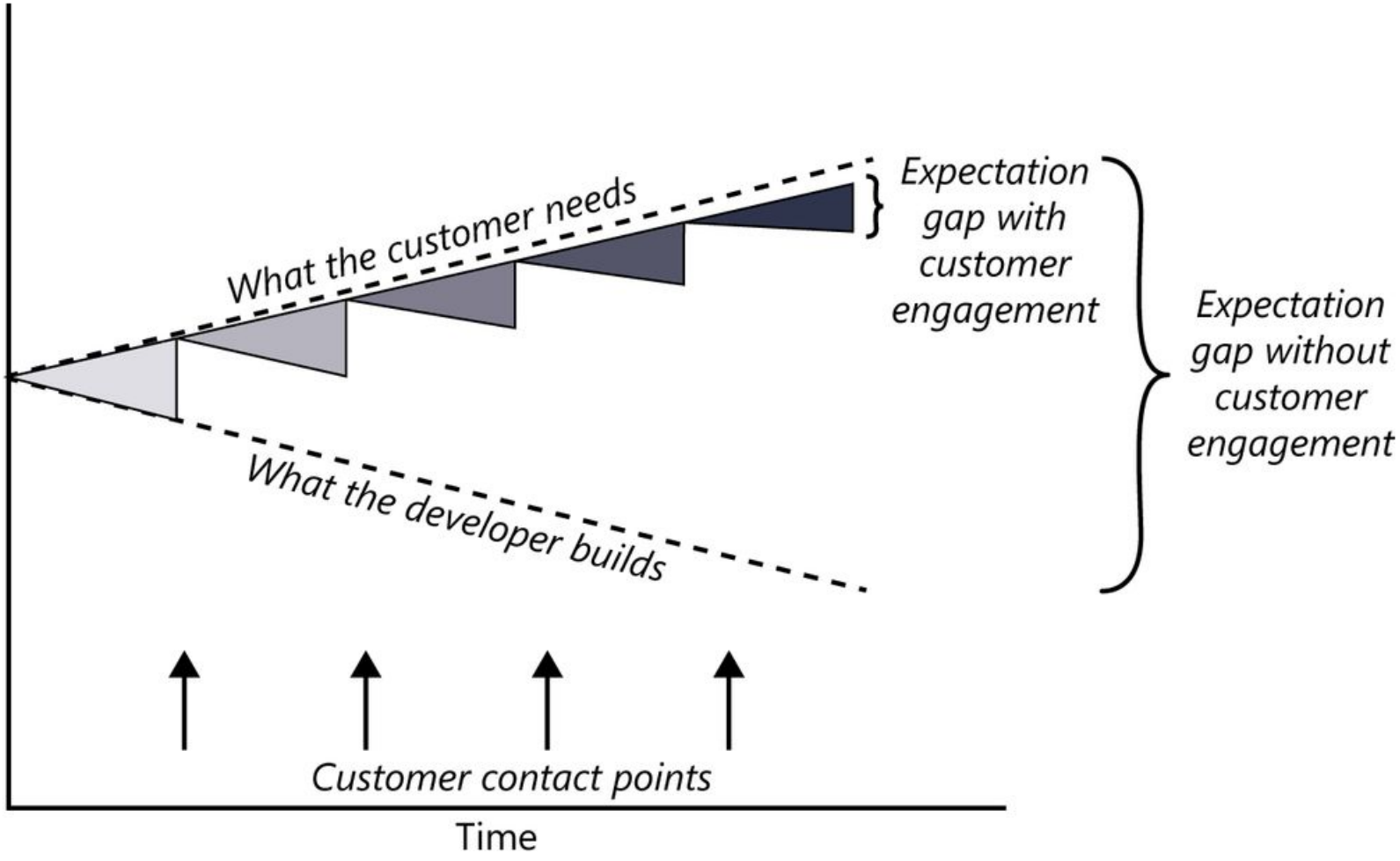
- Availability, security, usability
- Reliability, maintainability, expandability
- Time, resources, legal and organizational constraints
- Legacy systems, existing infra-structure, standards, state of the art

Source: IBM Architectural Thinking

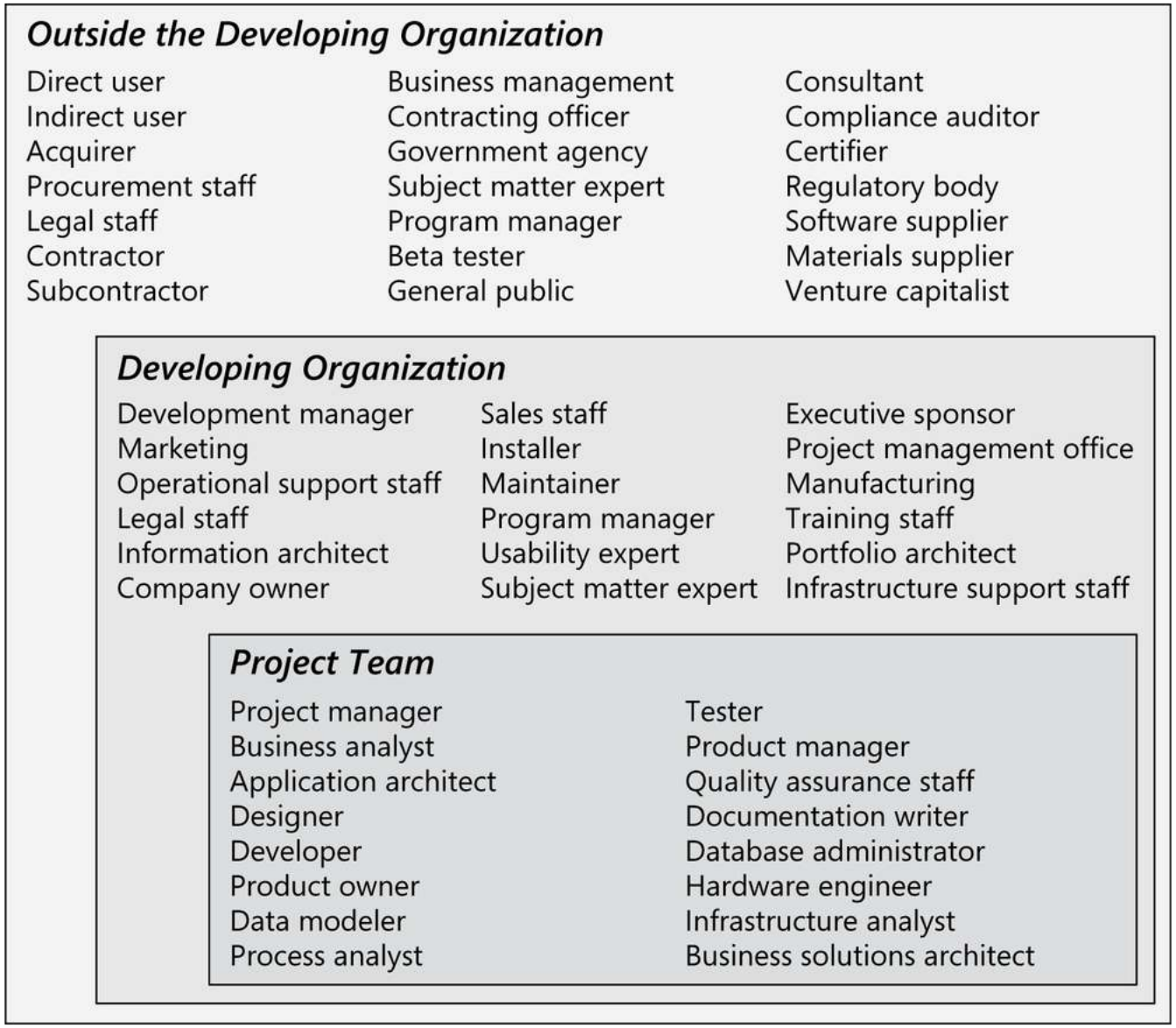
Smart Requirements

- **S**pecific: unambiguous and consistent
  - described at appropriate level of detail
- **M**easurable: objectively and independent of human interests
  - how can we decide if the requirement has been satisfied?
- **A**ttainable: it is in principle possible to build it
  - technically feasible with the current state of the art
- **R**ealizable: *we* can build it
  - under the given constraints (available resources)
- **T**raceable: track work items through the development process
  - conception - specification - design - implementation - test

Customer Needs and Software Development



The Universe of Stakeholders



A stakeholder is a party who has some interest in the system under consideration

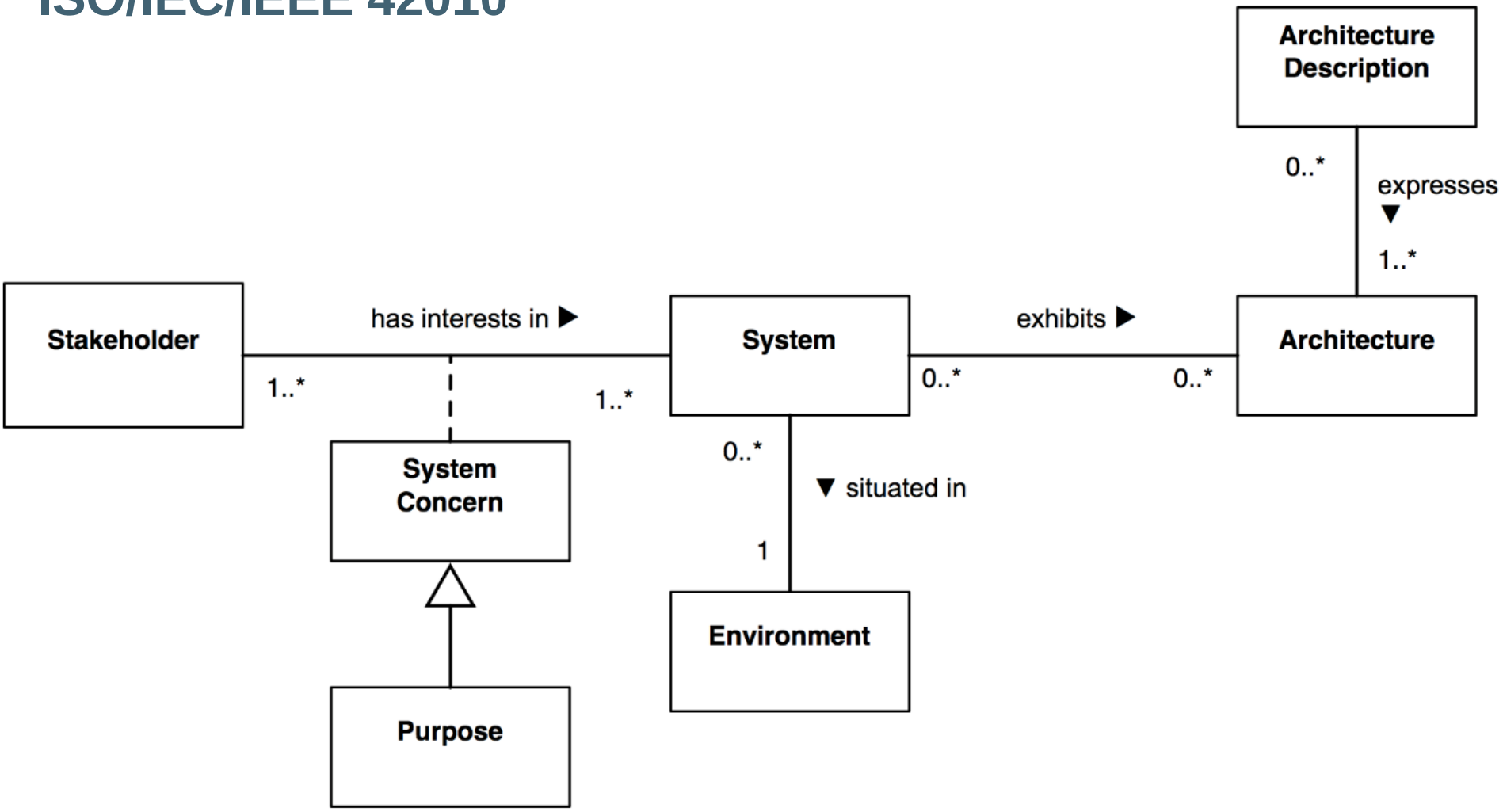
ISO/IEC/IEEE 42010 *Systems and software engineering — Architecture description* is an international standard for architecture descriptions of systems and software

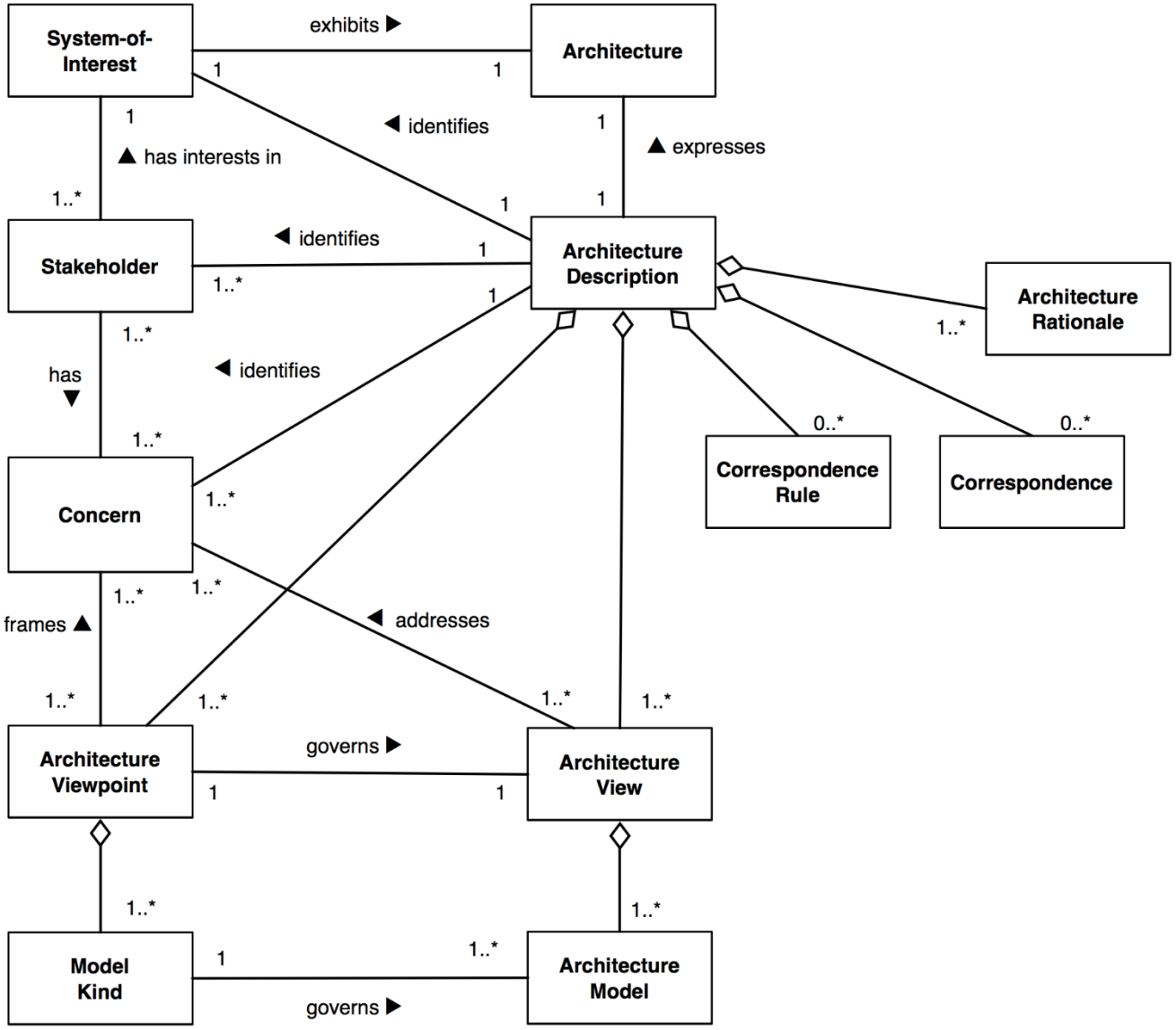
- ***Concern*** is an interest in a system relevant to one or more of its stakeholders. A concern pertains to any influence on a system in its environment, including developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological and social influences.

- ***Architecture Decision*** affects AD Elements and pertains to one or more Concerns. By making a Decision, new Concerns may be raised.

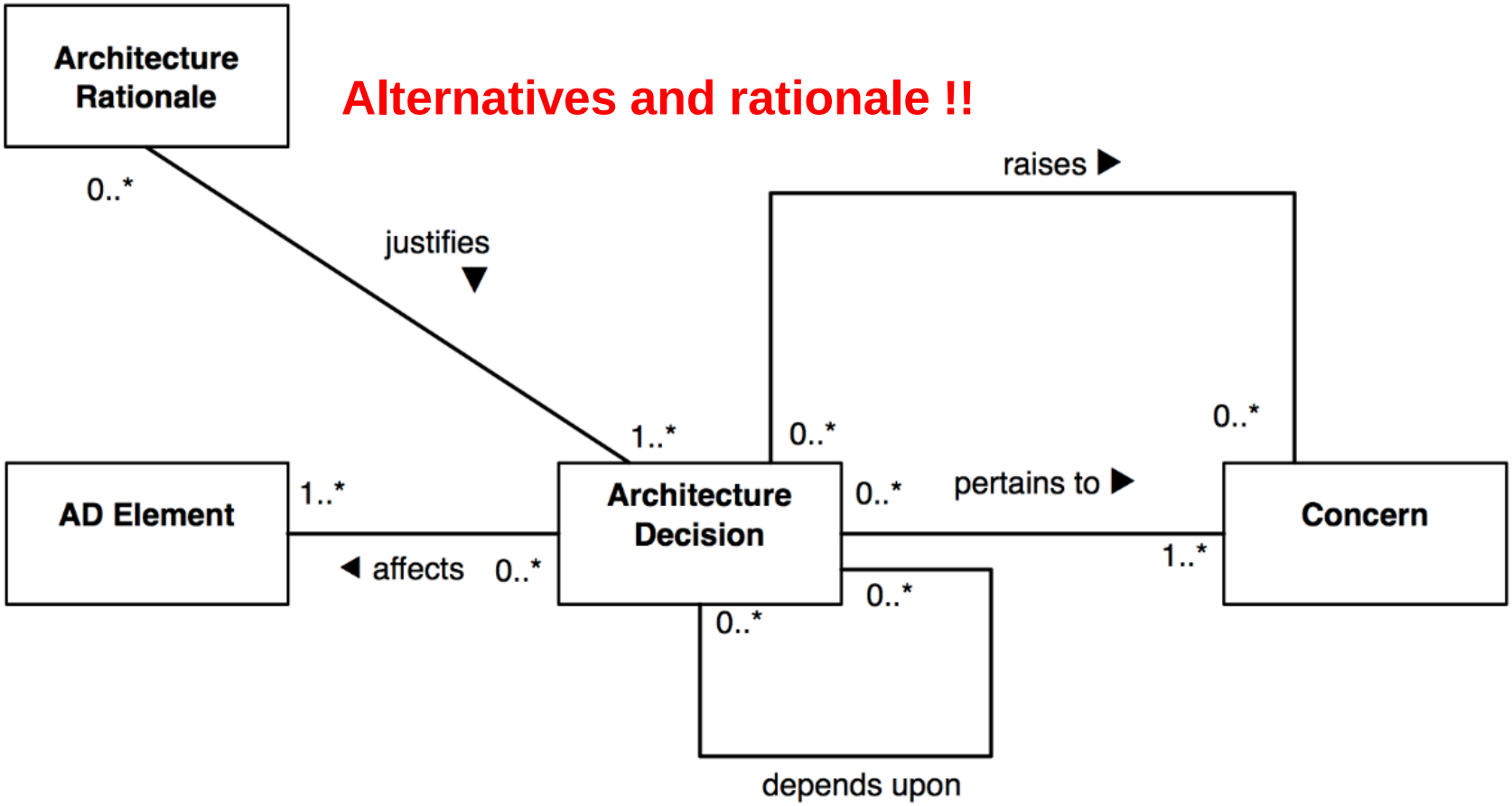
- ***Architecture Rationale*** records the explanation, justification or reasoning about Architecture Decisions that have been made and architectural alternatives not chosen.

ISO/IEC/IEEE 42010



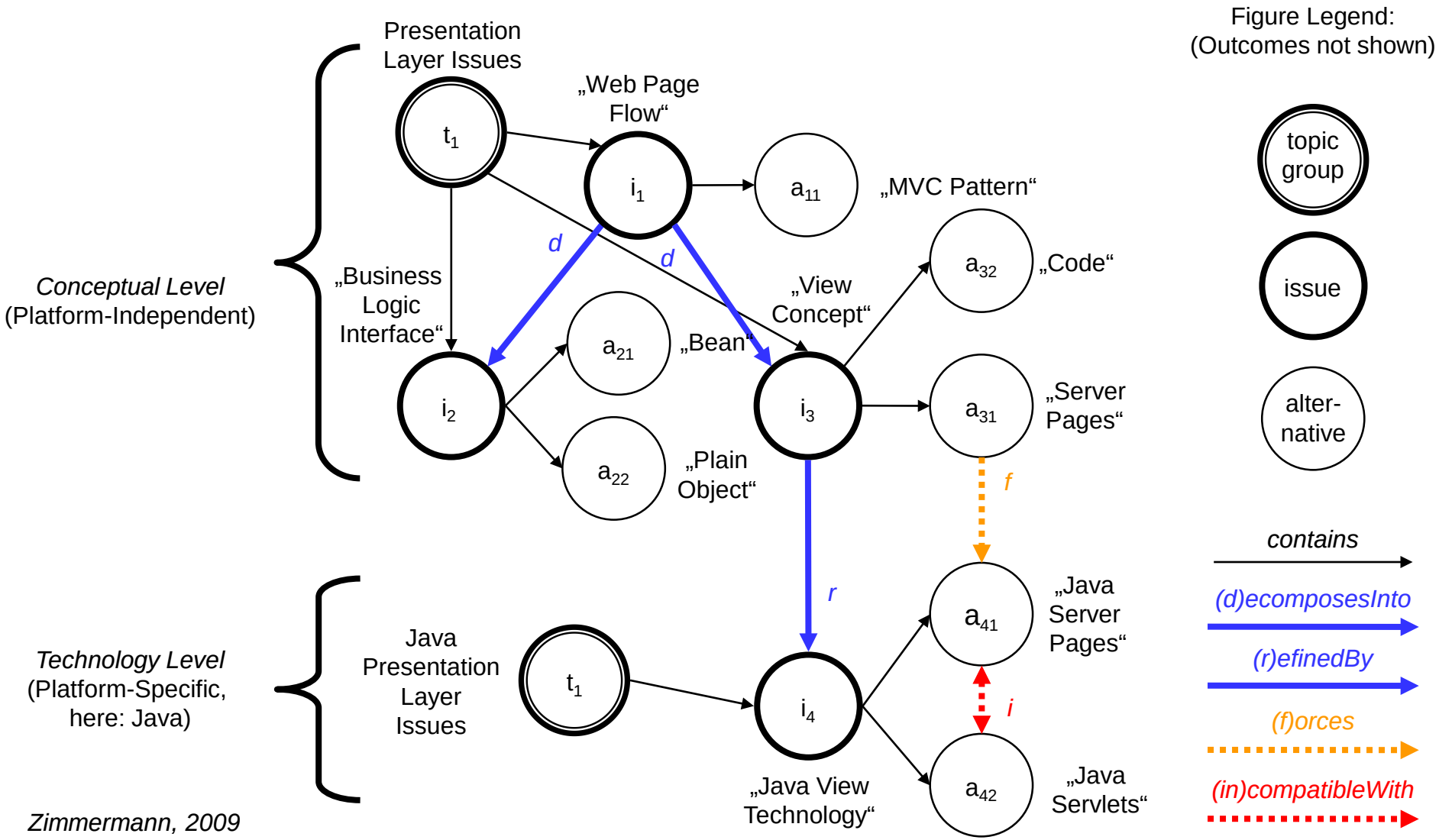

System-of-Interest
exhibits
Architecture
1
1
identifies
1
has interests in
1..*
1
expresses
1
Stakeholder
identifies
1
Architecture Description
1..*
1
Architecture Rationale
1..*
has
identifies
1..*
1
1..*
Concern
1..*
Correspondence Rule
0..*
Correspondence
0..*
1..*
1..*
addresses
frames
1..*
1..*
1..*
1..*
Architecture Viewpoint
governs
Architecture View
1
1
1..*
1..*
Model Kind
1
1..*
governs
Architecture Model

Documentation of Architectural Decisions



Understand the decision problem, possible decision alternatives, result of your decision process and **why** you arrived at the particular decision result

Decision Dependencies



Recommendations for the Documentation of Decisions

- Unique identifier for the decision
- Precise description of the decision (statement)
- Concerns to which it pertains
- Owner of the decision
- Affected AD elements
- Rationale linked to the decision
- Forces and constraints on the decision
- Assumptions influencing the decision
- Considered alternatives and their potential consequences

7 Types of Decisions (Bass,Clements,Kazman)

1. Assignment of Responsibilities
   - Which components in the system are responsible for specific functions and quality attributes?
2. Coordination Model
   - How do components interact?
   - Which ones are not allowed to interact?
3. Data Model (incl. data management)
4. Resource Management
   - Which resources are required?
   - How are resources managed or shared?

7 Types of decisions (cont.)

5. Mapping of architectural elements
   - Inside system architecture: components and operational model
   - To external systems: interfaces and dependencies

6. Binding time Decisions to achieve variability in architecture

7. Choice of Technology
   - Given as constraint or decision of the architect

How do we organize our decisions?

- According to concern ….
- According to AD Element
- According to architectural levels

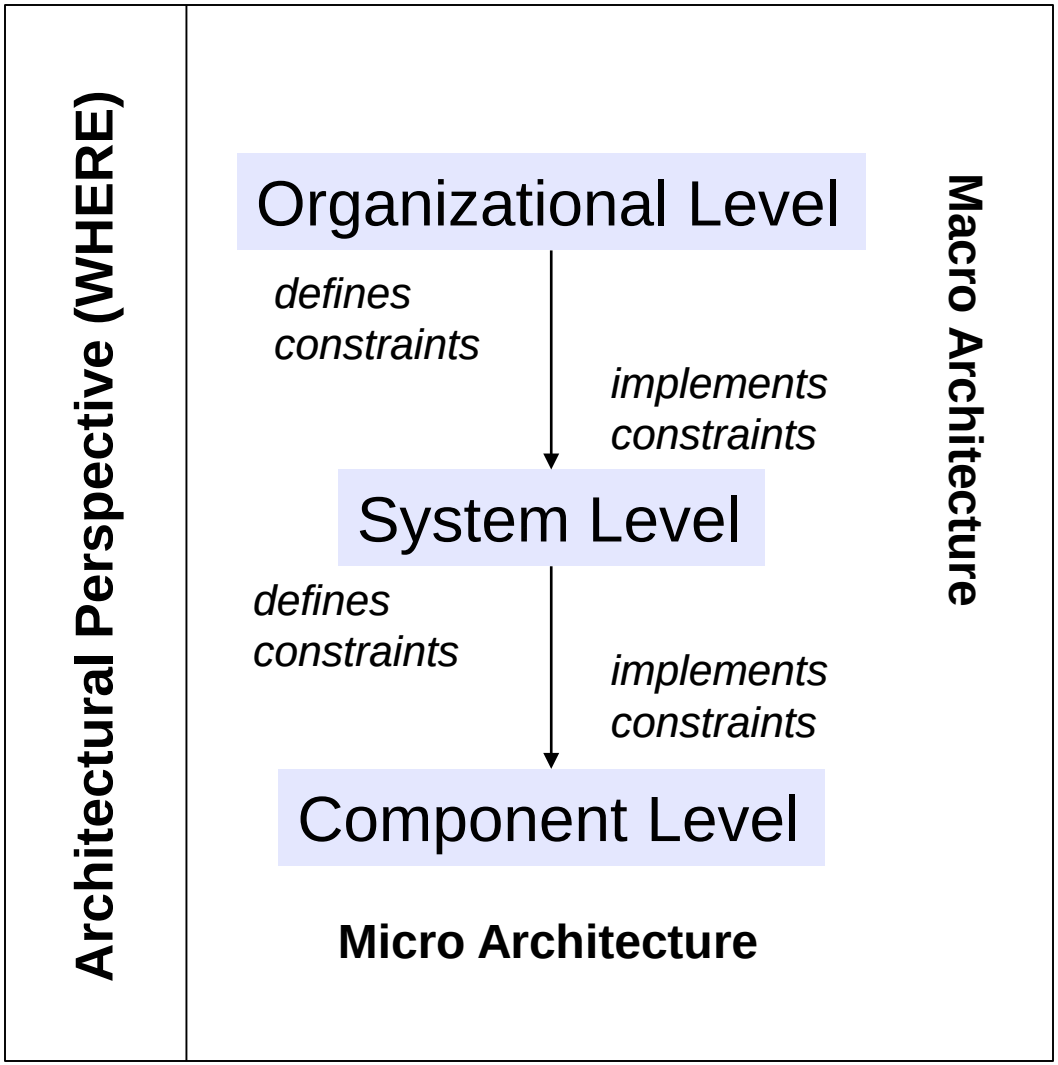
Perspectives and Levels in an Architecture

- On which levels of abstraction does an architect move within the framework of his activity? At which levels does he make decisions?
- How does architecture manifest itself on these levels of abstraction?

"With the help of architectural levels, as an architect one is more aware of the powers and their origins, which always have a fundamental effect on an architecture. At the module level, forces from the system level and at the system level forces from the organizational level have an effect on an architecture.
By being aware of these facts, the problems and questions arising during the creation of an architecture are treated in a more uniform way and the mixing of different aspects is avoided."

Levels

- Assign problems to the appropriate levels and thus handle them more easily and uniformly
- Separate different concerns
- ➢ Influential forces on an architecture are explicitly present, can be better understood and taken into account



Vogel et al: Software Architecture

Dimensions and Levels

Dimension \ Level	Organization	System	Components
What	Enterprise Architecture, Business Processes, …	Software Architecture, …	Software Architecture, …
Where	Requirements View, …	Requirements View, Logical View, …	Requirements View, Logical View, …
Why	Organisational Requirements, IT Standards, IT Guidelines, …	System Requirements, …	Component Requirements, …
With what	Business Process Descriptions, Business Use Case, …	System Context Diagram, Styles, Reference Architectures, Frameworks, …	Architectural Pattern, Design Pattern, Frameworks, …
Who	Enterprise Architect, …	Software Architect, …	Software Architect, …
How	Creating the System Vision, Context	Creating the System Idea, Developing the Architecture, …	Developing the Architecture, …

Architectural Decisions and Levels

Levels (according to Zimmermann)	Basic Decision
Executive level	Architectural style
Conceptual level	Architectural pattern
Technology level	Java EE or .NET?
Manufacturer level	Oracle or IBM? Microsoft?

- Decisions are not independent of each other!

O. Zimmermann et al.: Managing architectural decision models with dependency relations, integrity constraints, and production rules. Journal of Systems and Software (2009)

Examples of Architectural Decisions

An AI System for Cable Tree Wiring

Architecture Documentation

Software Architecture Documentation

- Also architecture master document, architecture reference or architecture state-of-the-art

- Includes various *work products*, in particular
  - Tasks, requirements and goals, stakeholders
  - Boundary conditions and context view
  - Other views, glossaries of terms
  - Design decisions and patterns used
  - Architecture assessment scenarios
  - Risks, change requests, project aspects

➢ No really satisfying tool support currently available

Documentation Templates

- Companies & organizations define their own frameworks
  - Specifications for uniform terminology, roles, work phases, work products, templates
  - e.g. IBM Unified Method Framework (UMF)
- Enterprise Architecture Frameworks: Open Group TOGAF
- Hruschka/Starke: http://arc42.de/
- Standards
  - **ISO 10746** Reference Model for Open Distributed Processing
    - http:// www.rm-odp.net
  - **ISO 42010** Recommended Practice for Architectural Description of Software-intensive Systems

Enterprise Architecture (EA) defines Application Architecture Documentation - John Zachman 1987

A framework for information systems architecture

by J. A. Zachman

IBM SYSTEMS JOURNAL, VOL26, NO 3, 1987; © 1987, 1999

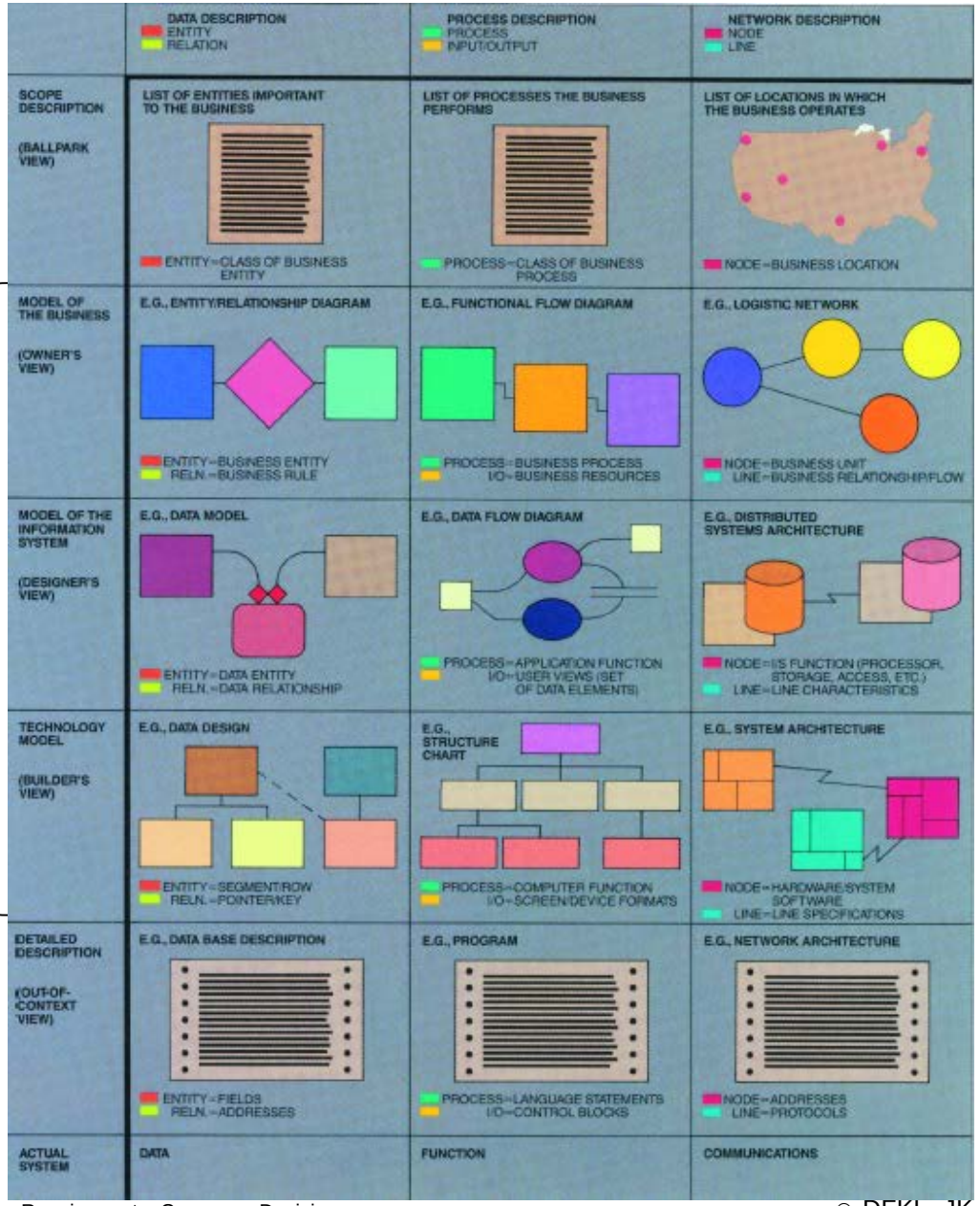
"With increasing size and complexity of the implementations of information systems, it is necessary to use some logical construct (or architecture) for defining and controlling the interfaces and the integration of all the components of a system."

When the question is asked, "What is information systems architecture?" the answer is, "There is not *an* information systems architecture, but a *set* of them!" Architecture is relative. What you think architecture is depends on what you are doing. For an example, see Table 6.

We are having difficulties communicating with one another about information systems architecture, because a *set* of architectural representations exists, instead of a *single* architecture. One is not right and another wrong. The architectures are different. They are additive and complementary. There are reasons for electing to expend the resources for developing each architectural representation. And there are risks associated with *not* developing any one of the architectural representations.

Zachman 1987

- Notion of architectural levels
  - Business
  - Information systems
  - Technology
- Architectural views
  - Data
  - Process
  - Network
- Importance of managing risks



Data – Process – Network

Table 3 Three different types of descriptions of the same product

	Description I	Description II	Description III
Orientation	Material	Function	Location
Focus	Structure	Transform	Flow
Description	*WHAT* the thing is made of	*HOW* the thing works	*WHERE* the flows (connections) exist
Example	Bill-of-materials	Functional specifications	Drawings
Descriptive model	Part-relationship-part	Input-process-output	Site-link-site

Table 4 Information systems analogs for the different types of descriptions

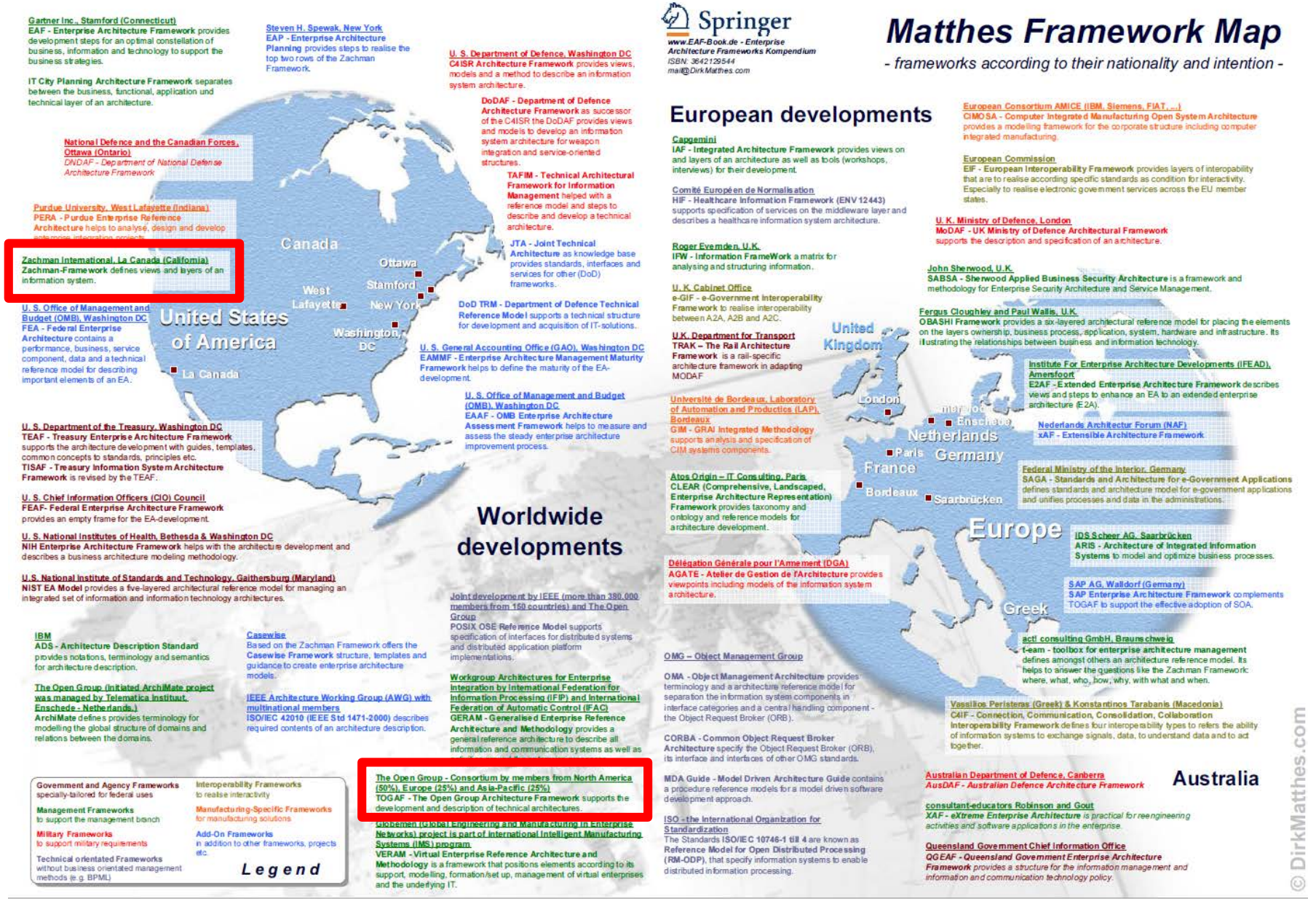
	Description I (material)	Description II (function)	Description III (location)
Information systems analog	Data model	Process model	Network model
I/S descriptive model	Entity-relationship-entity	Input-process-output	Node-line-node

Work Products in Zachman Framework

	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*
Objective/Scope (contextual) *Role: Planner*	List of things important in the business	List of Business Processes	List of Business Locations	List of important Organizations	List of Events	List of Business Goal & Strategies
Enterprise Model (conceptual) *Role: Owner*	Conceptual Data/ Object Model	Business Process Model	Business Logistics System	Work Flow Model	Master Schedule	Business Plan
System Model (logical) *Role:Designer*	Logical Data Model	System Architecture Model	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
Technology Model (physical) *Role:Builder*	Physical Data/Class Model	Technology Design Model	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
Detailed Reprentation (out of context) *Role: Programmer*	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Speculation
Functioning Enterprise *Role: User*	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy

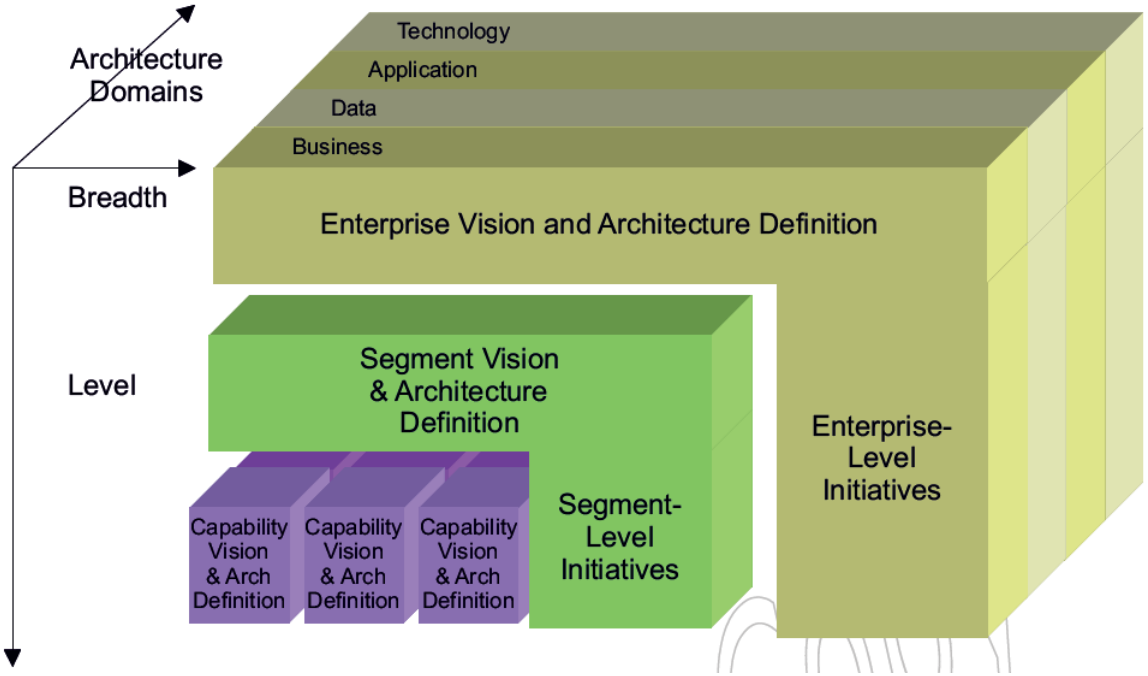
The Zachman Framework

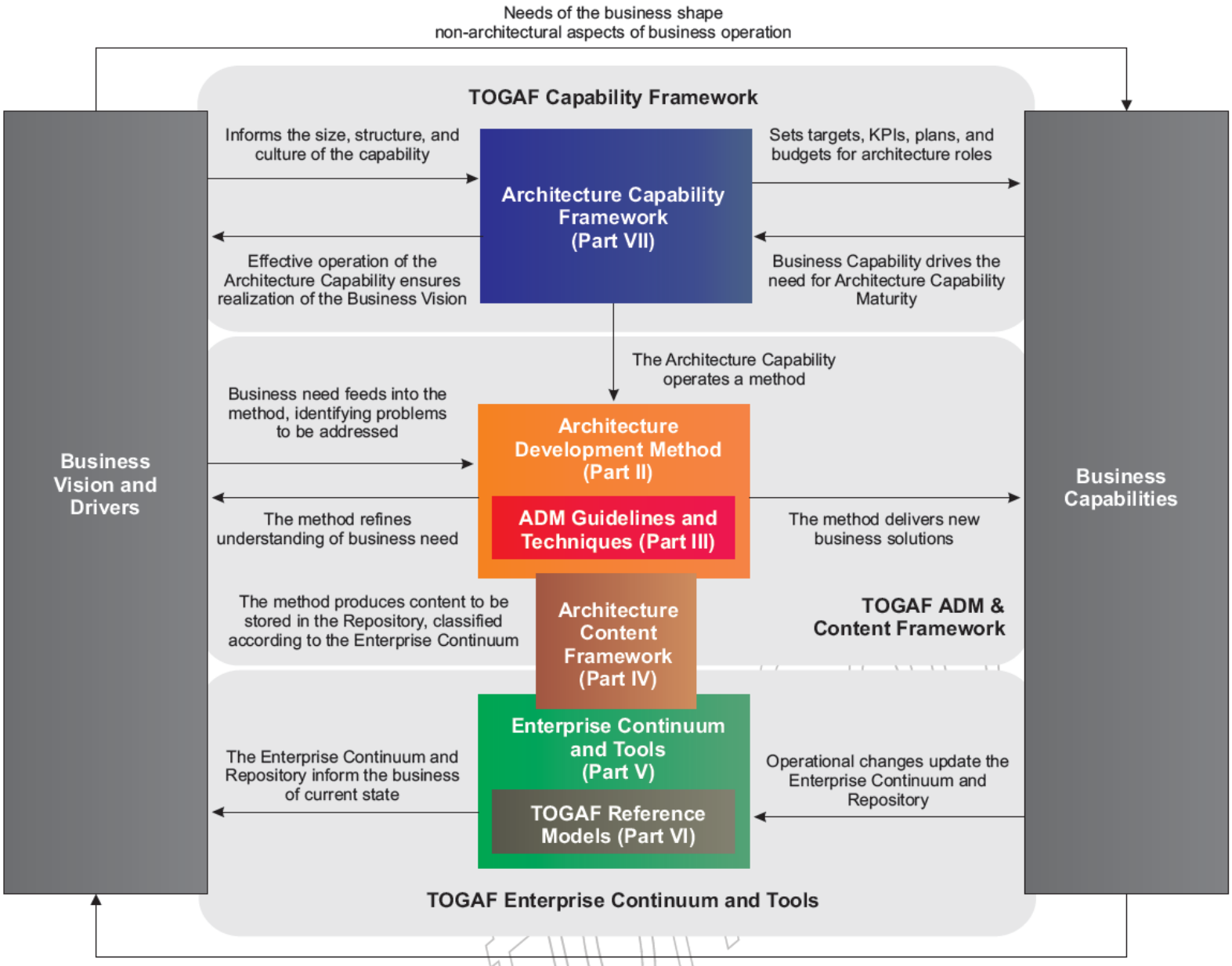
	What? Data	*How?* Function	*Where?* Network	*Who?* People	*When?* Time	*Why?* Motivation	
Planner's Viewpoint Contextual							Scope
Owner's Viewpoint Conceptual							Enterprise Models
Designer's Viewpoint Logical							Systems Models
Builder's Viewpoint Physical							Technology Models
Sub-contractor's Viewpoint Out-of-context							Detailed Representations
Functioning Enterprise							Actual Systems


Springer
www.EAF-Book.de - Enterprise Architecture Frameworks Kompendium
ISBN: 3642129544
mail@DirkMatthes.com
Matthes Framework Map
- frameworks according to their nationality and intention -
Gartner Inc., Stamford (Connecticut)
EAF - Enterprise Architecture Framework provides development steps for an optimal constellation of business, information and technology to support the business strategies.
IT City Planning Architecture Framework separates between the business, functional, application und technical layer of an architecture.
Steven H. Spewak, New York
EAP - Enterprise Architecture Planning provides steps to realise the top two rows of the Zachman Framework.
U. S. Department of Defence, Washington DC
C4ISR Architecture Framework provides views, models and a method to describe an information system architecture.
DoDAF - Department of Defence Architecture Framework as successor of the C4ISR the DoDAF provides views and models to develop an information system architecture for weapon integration and service-oriented structures.
TAFIM - Technical Architectural Framework for Information Management helped with a reference model and steps to describe and develop a technical architecture.
JTA - Joint Technical Architecture as knowledge base provides standards, interfaces and services for other (DoD) frameworks.
DoD TRM - Department of Defence Technical Reference Model supports a technical structure for development and acquisition of IT-solutions.
National Defence and the Canadian Forces, Ottawa (Ontario)
DNDAF - Department of National Defense Architecture Framework
Purdue University, West Lafayette (Indiana)
PERA - Purdue Enterprise Reference Architecture helps to analyse, design and develop
Zachman International, La Canada (California)
Zachman-Framework defines views and layers of an information system.
U. S. Office of Management and Budget (OMB), Washington DC
FEA - Federal Enterprise Architecture contains a performance, business, service component, data and a technical reference model for describing important elements of an EA.
U. S. General Accounting Office (GAO), Washington DC
EAMMF - Enterprise Architecture Management Maturity Framework helps to define the maturity of the EA-development.
U. S. Office of Management and Budget (OMB), Washington DC
EAAF - OMB Enterprise Architecture Assessment Framework helps to measure and assess the steady enterprise architecture improvement process.
U. S. Department of the Treasury, Washington DC
TEAF - Treasury Enterprise Architecture Framework supports the architecture development with guides, templates, common concepts to standards, principles etc.
TISAF - Treasury Information System Architecture Framework is revised by the TEAF.
U. S. Chief Information Officers (CIO) Council
FEAF- Federal Enterprise Architecture Framework provides an empty frame for the EA-development.
U. S. National Institutes of Health, Bethesda & Washington DC
NIH Enterprise Architecture Framework helps with the architecture development and describes a business architecture modeling methodology.
U.S. National Institute of Standards and Technology, Gaithersburg (Maryland)
NIST EA Model provides a five-layered architectural reference model for managing an integrated set of information and information technology architectures.
Canada
Ottawa
Stamford
West Lafayette
New York
Washington, DC
United States of America
La Canada
Worldwide developments
IBM
ADS - Architecture Description Standard provides notations, terminology and semantics for architecture description.
The Open Group (Initiated ArchiMate project was managed by Telematica Instituut, Enschede - Netherlands.)
ArchiMate defines provides terminology for modelling the global structure of domains and relations between the domains.
Casewise
Based on the Zachman Framework offers the Casewise Framework structure, templates and guidance to create enterprise architecture models.
IEEE Architecture Working Group (AWG) with multinational members
ISO/IEC 42010 (IEEE Std 1471-2000) describes required contents of an architecture description.
Joint development by IEEE (more than 380.000 members from 150 countries) and The Open Group
POSIX OSE Reference Model supports specification of interfaces for distributed systems and distributed application platform implementations.
Workgroup Architectures for Enterprise Integration by International Federation for Information Processing (IFIP) and International Federation of Automatic Control (IFAC)
GERAM - Generalised Enterprise Reference Architecture and Methodology provides a general reference architecture to describe all information and communication systems as well as
The Open Group - Consortium by members from North America (50%), Europe (25%) and Asia-Pacific (25%)
TOGAF - The Open Group Architecture Framework supports the development and description of technical architectures.
Globemen (Global Engineering and Manufacturing in Enterprise Networks) project is part of international Intelligent Manufacturing Systems (IMS) program
VERAM - Virtual Enterprise Reference Architecture and Methodology is a framework that positions elements according to its support, modelling, formation/set up, management of virtual enterprises and the underlying IT.
Government and Agency Frameworks specially-tailored for federal uses
Management Frameworks to support the management branch
Military Frameworks to support military requirements
Technical orientated Frameworks without business orientated management methods (e.g. BPML)
Interoperability Frameworks to realise interactivity
Manufacturing-Specific Frameworks for manufacturing solutions
Add-On Frameworks in addition to other frameworks, projects etc.
Legend
European developments
Capgemini
IAF - Integrated Architecture Framework provides views on and layers of an architecture as well as tools (workshops, interviews) for their development.
Comité Européen de Normalisation
HIF - Healthcare Information Framework (ENV 12443) supports specification of services on the middleware layer and describes a healthcare information system architecture.
Roger Evernden, U.K.
IFW - Information FrameWork a matrix for analysing and structuring information.
U. K. Cabinet Office
e-GIF - e-Government Interoperability Framework to realise interoperability between A2A, A2B and A2C.
U.K. Department for Transport
TRAK – The Rail Architecture Framework is a rail-specific architecture framework in adapting MODAF
Université de Bordeaux, Laboratory of Automation and Productics (LAP), Bordeaux
GIM - GRAI Integrated Methodology supports analysis and specification of CIM systems components.
Atos Origin – IT Consulting, Paris
CLEAR (Comprehensive, Landscaped, Enterprise Architecture Representation) Framework provides taxonomy and ontology and reference models for architecture development.
Délégation Générale pour l'Armement (DGA)
AGATE - Atelier de Gestion de l'Architecture provides viewpoints including models of the information system architecture.
OMG – Object Management Group
OMA - Object Management Architecture provides terminology and a architecture reference model for separation the information system components in interface categories and a central handling component - the Object Request Broker (ORB).
CORBA - Common Object Request Broker Architecture specify the Object Request Broker (ORB), its interface and interfaces of other OMG standards.
MDA Guide - Model Driven Architecture Guide contains a procedure reference models for a model driven software development approach.
ISO - the International Organization for Standardization
The Standards ISO/IEC 10746-1 till 4 are known as Reference Model for Open Distributed Processing (RM-ODP), that specify information systems to enable distributed information processing.
European Consortium AMICE (IBM, Siemens, FIAT, ...)
CIMOSA - Computer Integrated Manufacturing Open System Architecture provides a modelling framework for the corporate structure including computer integrated manufacturing.
European Commission
EIF - European Interoperability Framework provides layers of interopability that are to realise according specific standards as condition for interactivity. Especially to realise electronic government services across the EU member states.
U. K. Ministry of Defence, London
MoDAF - UK Ministry of Defence Architectural Framework supports the description and specification of an architecture.
John Sherwood, U.K.
SABSA - Sherwood Applied Business Security Architecture is a framework and methodology for Enterprise Security Architecture and Service Management.
Fergus Cloughley and Paul Wallis, U.K.
OBASHI Framework provides a six-layered architectural reference model for placing the elements on the layers ownership, business process, application, system, hardware and infrastructure. Its illustrating the relationships between business and information technology.
Institute For Enterprise Architecture Developments (IFEAD), Amersfoort
E2AF - Extended Enterprise Architecture Framework describes views and steps to enhance an EA to an extended enterprise architecture (E2A).
Nederlands Architectur Forum (NAF)
xAF - Extensible Architecture Framework
Federal Ministry of the Interior, Germany
SAGA - Standards and Architecture for e-Government Applications defines standards and architecture model for e-government applications and unifies processes and data in the administrations.
IDS Scheer AG, Saarbrücken
ARIS - Architecture of Integrated Information Systems to model and optimize business processes.
SAP AG, Walldorf (Germany)
SAP Enterprise Architecture Framework complements TOGAF to support the effective adoption of SOA.
actl consulting GmbH, Braunschweig
t-eam - toolbox for enterprise architecture management defines amongst others an architecture reference model. Its helps to answer the questions like the Zachman Framework: where, what, who, how, why, with what and when.
Vassilios Peristeras (Greek) & Konstantinos Tarabanis (Macedonia)
C4IF - Connection, Communication, Consolidation, Collaboration Interoperability Framework defines four interoperability types to refers the ability of information systems to exchange signals, data, to understand data and to act together.
Australian Department of Defence, Canberra
AusDAF - Australian Defence Architecture Framework
consultant-educators Robinson and Gout
XAF - eXtreme Enterprise Architecture is practical for reengineering activities and software applications in the enterprise.
Queensland Government Chief Information Office
QGEAF - Queensland Government Enterprise Architecture Framework provides a structure for the information management and information and communication technology policy.
United Kingdom
London
Netherlands
Enschede
Amsterdam
Germany
Paris
France
Bordeaux
Saarbrücken
Europe
Greek
Australia
© DirkMatthes.com

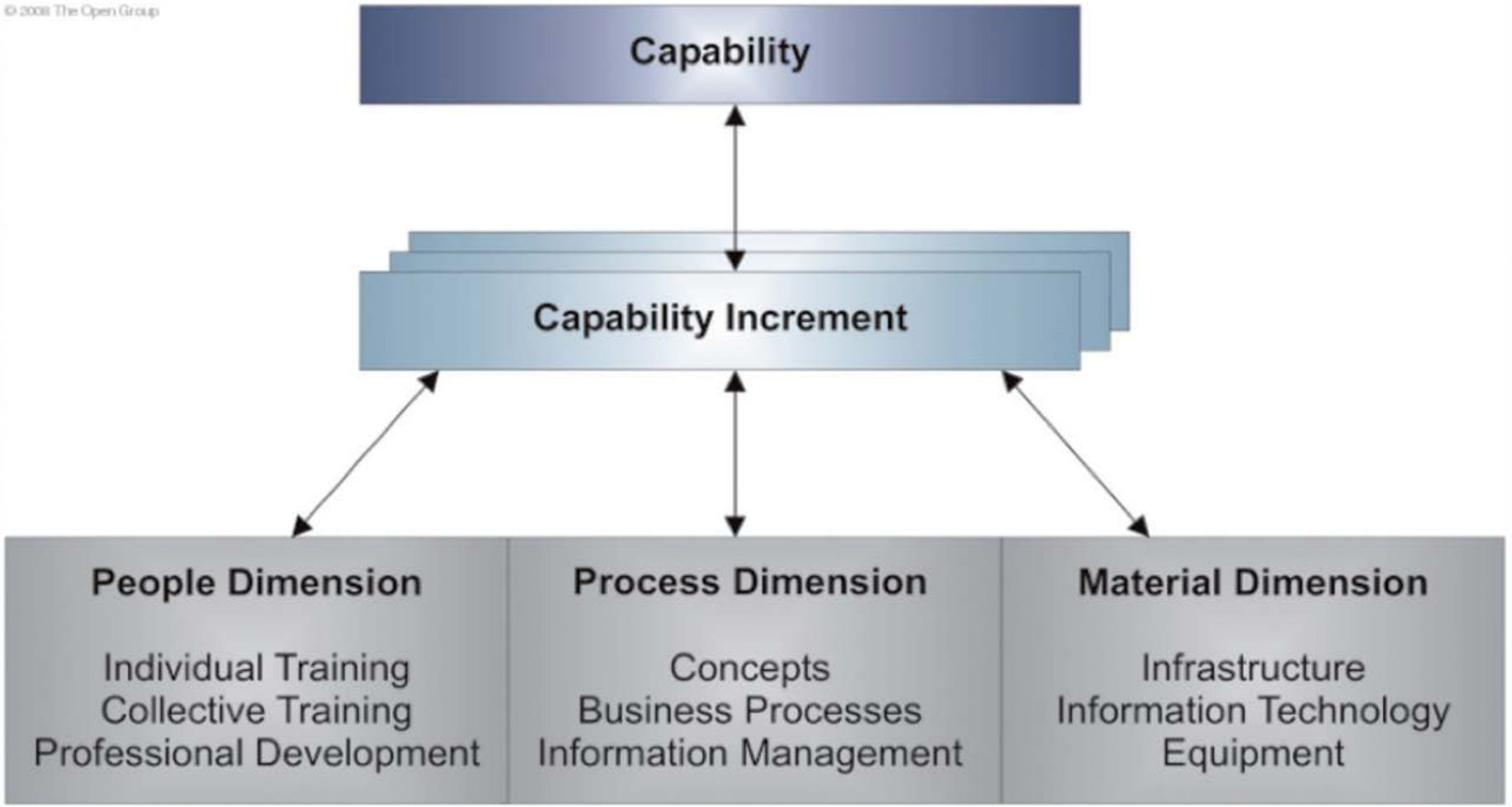
The Open Group Architecture Framework (TOGAF)

- Open, industry consensus framework for enterprise architecture, developed since 1994
- *"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."*

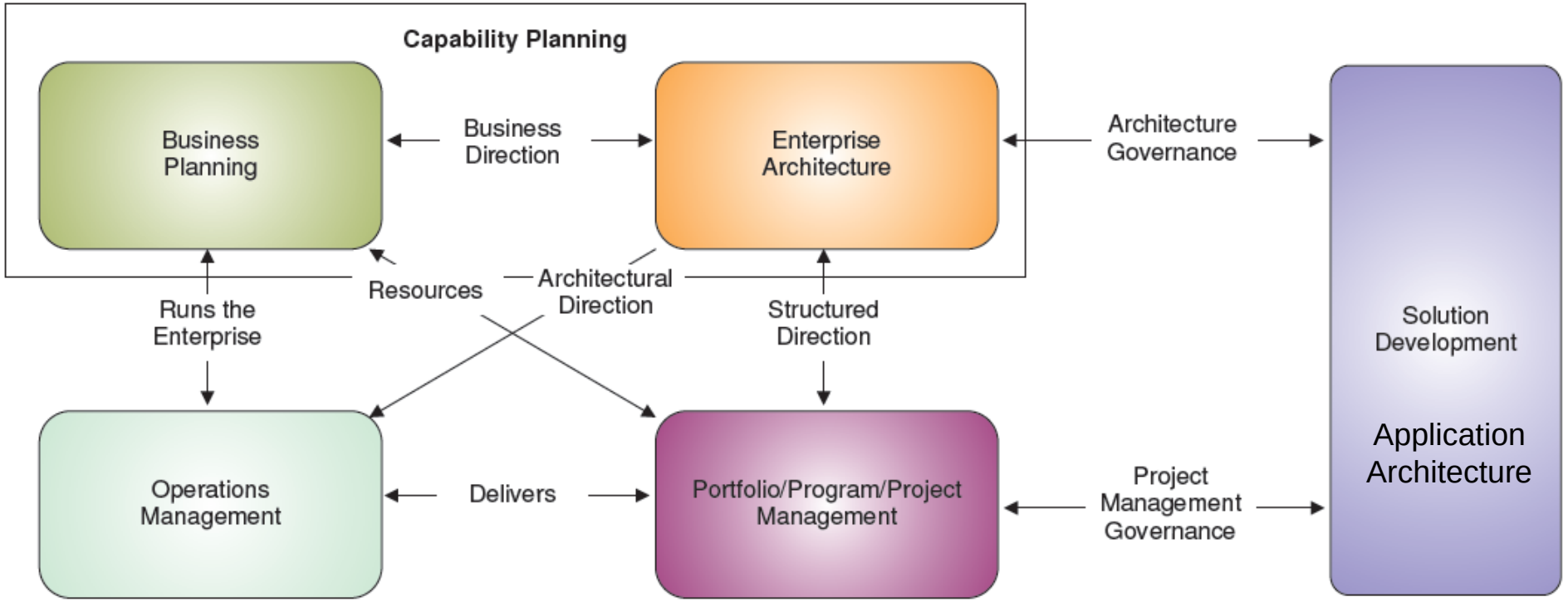



Needs of the business shape
non-architectural aspects of business operation
TOGAF Capability Framework
Informs the size, structure, and culture of the capability
Architecture Capability Framework (Part VII)
Sets targets, KPIs, plans, and budgets for architecture roles
Effective operation of the Architecture Capability ensures realization of the Business Vision
Business Capability drives the need for Architecture Capability Maturity
The Architecture Capability operates a method
Business need feeds into the method, identifying problems to be addressed
Architecture Development Method (Part II)
ADM Guidelines and Techniques (Part III)
The method refines understanding of business need
The method delivers new business solutions
Business Vision and Drivers
Business Capabilities
The method produces content to be stored in the Repository, classified according to the Enterprise Continuum
Architecture Content Framework (Part IV)
TOGAF ADM & Content Framework
Enterprise Continuum and Tools (Part V)
TOGAF Reference Models (Part VI)
The Enterprise Continuum and Repository inform the business of current state
Operational changes update the Enterprise Continuum and Repository
TOGAF Enterprise Continuum and Tools
Learning from business operation creates new business need

Capability-Based Planning

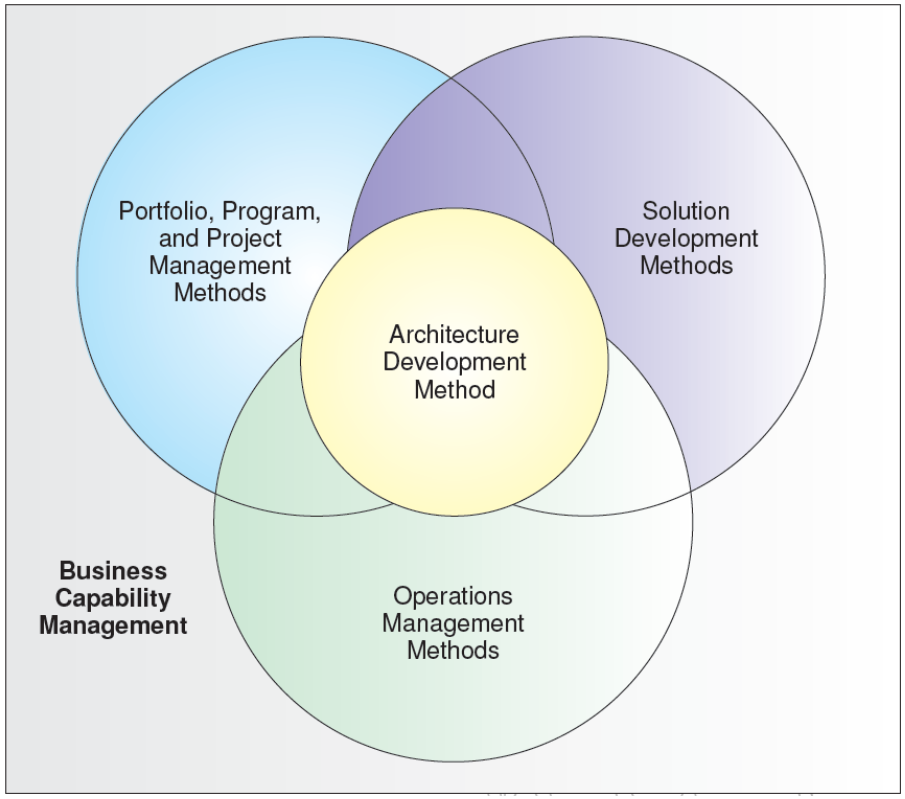


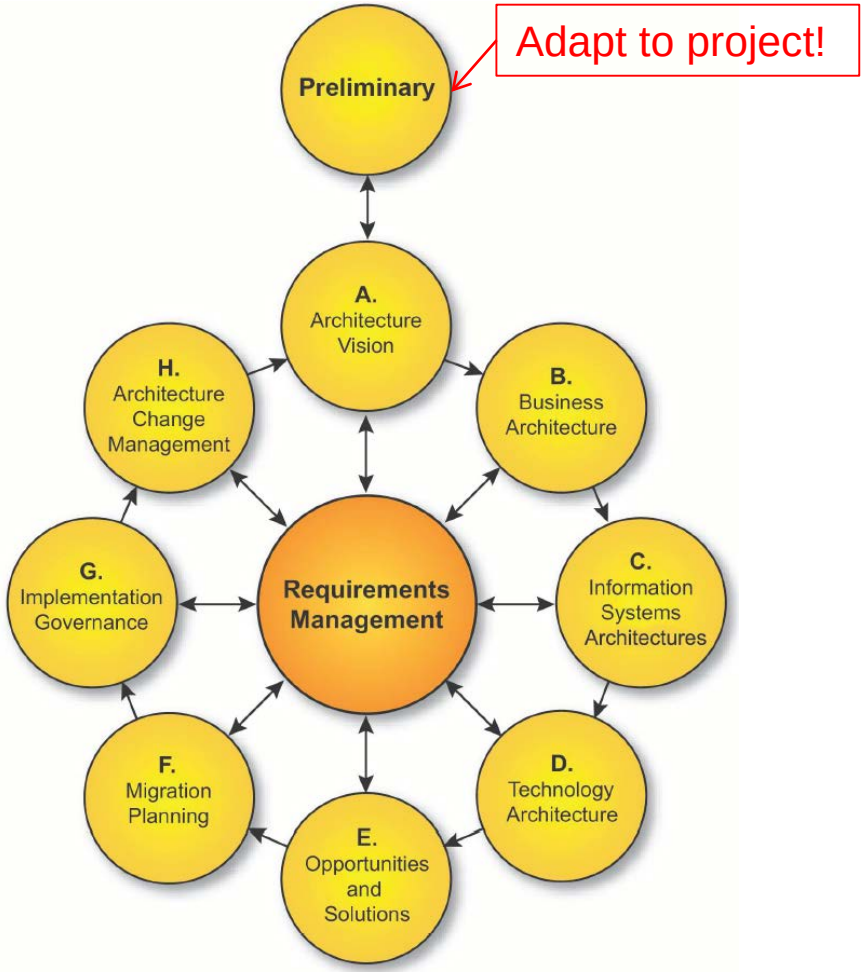
Application Architecture and Other Disciplines



TOGAF Architecture Development Method (ADM)

- Tested and repeatable process for developing architectures





Phase C: Information Systems Architectures — Application Architecture

11.1 Objectives

- Develop the Target Application Architecture that enables the Business Architecture and the Architecture Vision, while addressing the Request for Architecture Work and stakeholder concerns
- Identify candidate Architecture Roadmap components based upon gaps between the Baseline and Target Application Architectures

11.3.2 Non-Architectural Inputs

- Request for Architecture Work
- Capability Assessment
- Communications Plan

11.3.3 Architectural Inputs

- Organizational Model for Enterprise Architecture including
  - scope of organizations impacted, maturity assessment, gaps, and resolution approach, roles and responsibilities for architecture team(s), constraints on architecture work, budget requirements, governance and support strategy
- Tailored Architecture Framework
  - tailored architecture method, tailored architecture content (deliverables and artifacts), configured and deployed tools
- ….
- Draft Architecture Definition Document - Baseline and Target
  - Business Architecture
  - Data Architecture
  - Application Architecture
  - Technology Architecture
  - Target Technology Architecture
- Draft Architecture Requirements Specification
  - Gap analysis results (from Business Architecture and Data Architecture, if available)
  - Relevant technical requirements that will apply to this phase

11.4. Steps

1. Select reference models, viewpoints, and tools
2. Develop Baseline Application Architecture Description
3. Target Application Architecture Description
4. Perform gap analysis
5. Define candidate roadmap components
6. Resolve impacts across the Architecture Landscape
7. Conduct formal stakeholder review
8. Finalize the Application Architecture
9. Create Architecture Definition Document

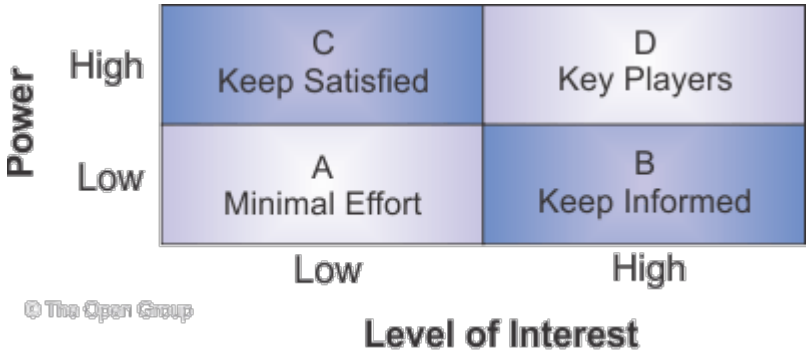
11.5 Outputs

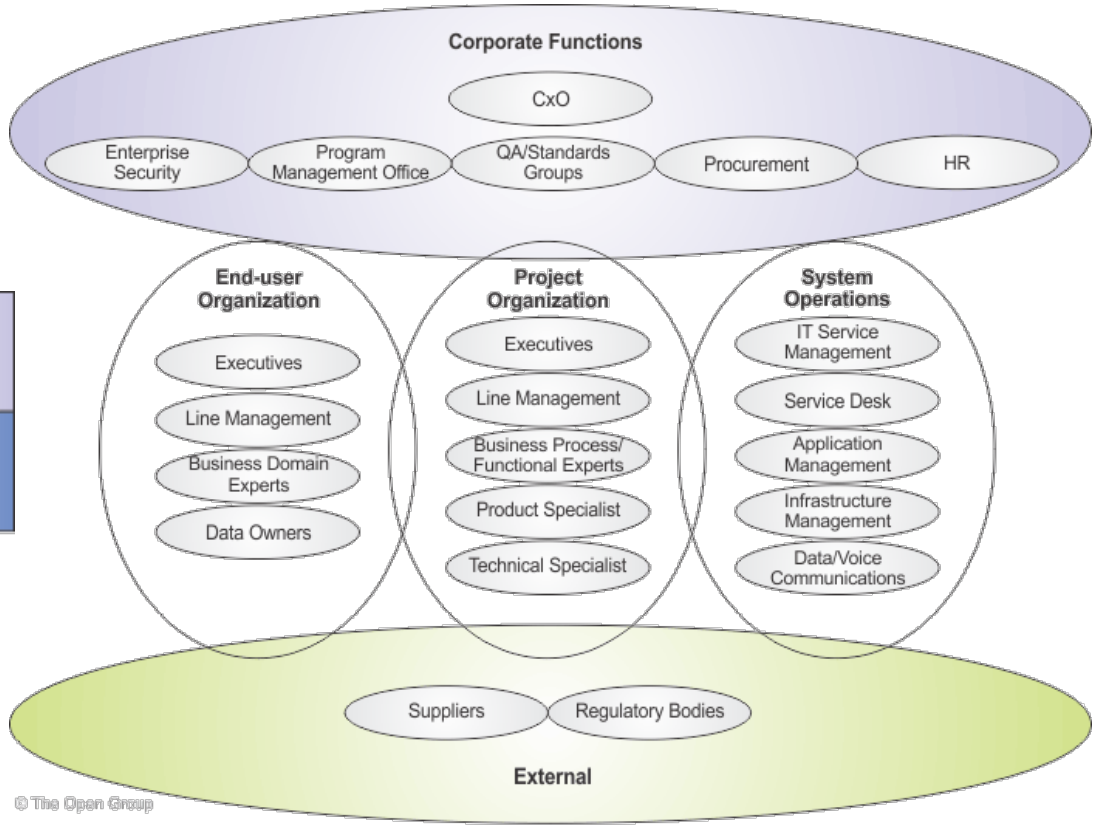
- Architecture Vision
- Draft Architecture Definition Document
- Draft Architecture Requirements Specification

- Catalogs: application portfolio, interface
- Matrices: application/organization, role/application, application/function, application interaction
- Diagrams: application communication, application and user location, application use case, enterprise manageability, process/application realization, software engineering, application migration, software distribution

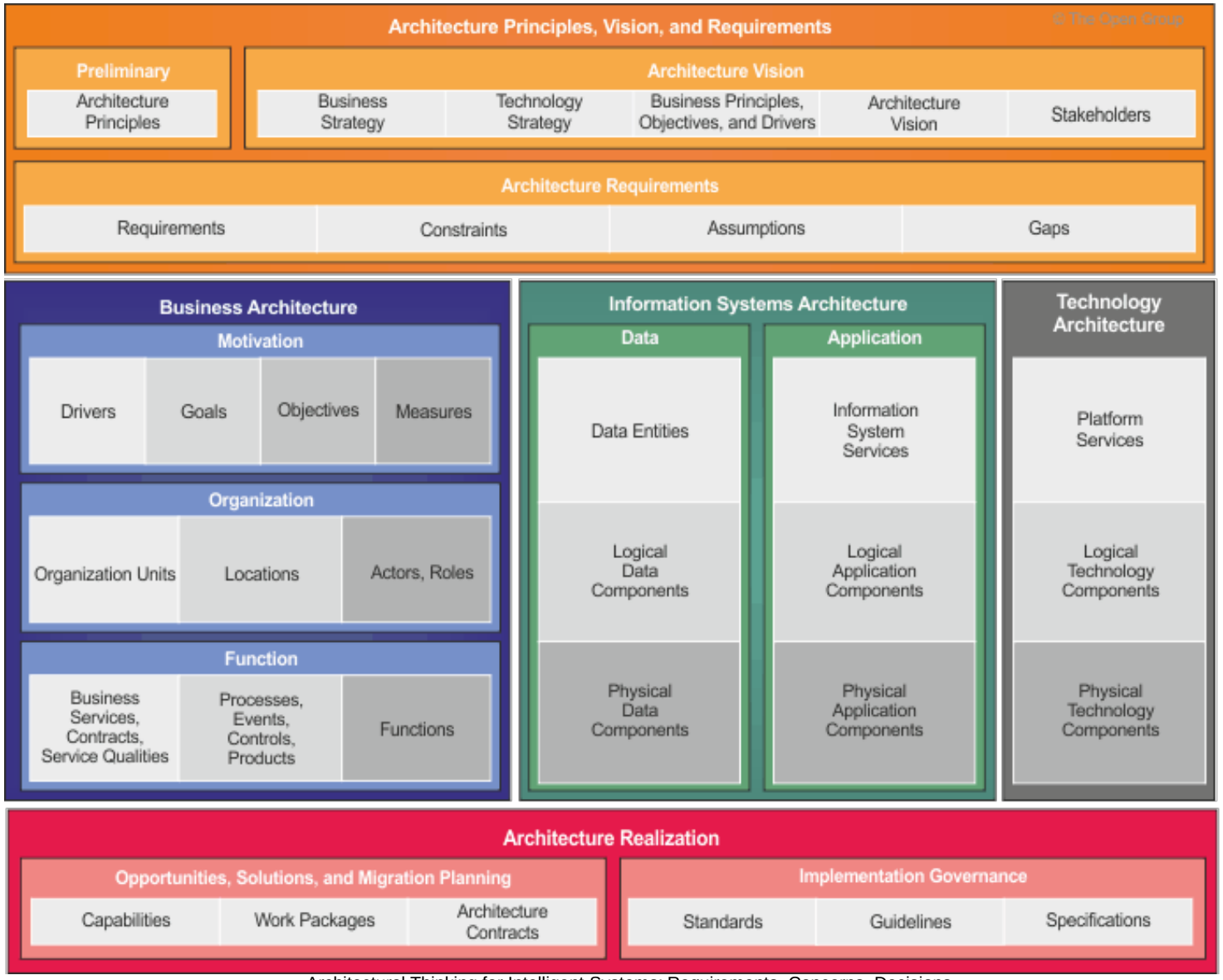
Stakeholder Management

Stakeholder Powergrid

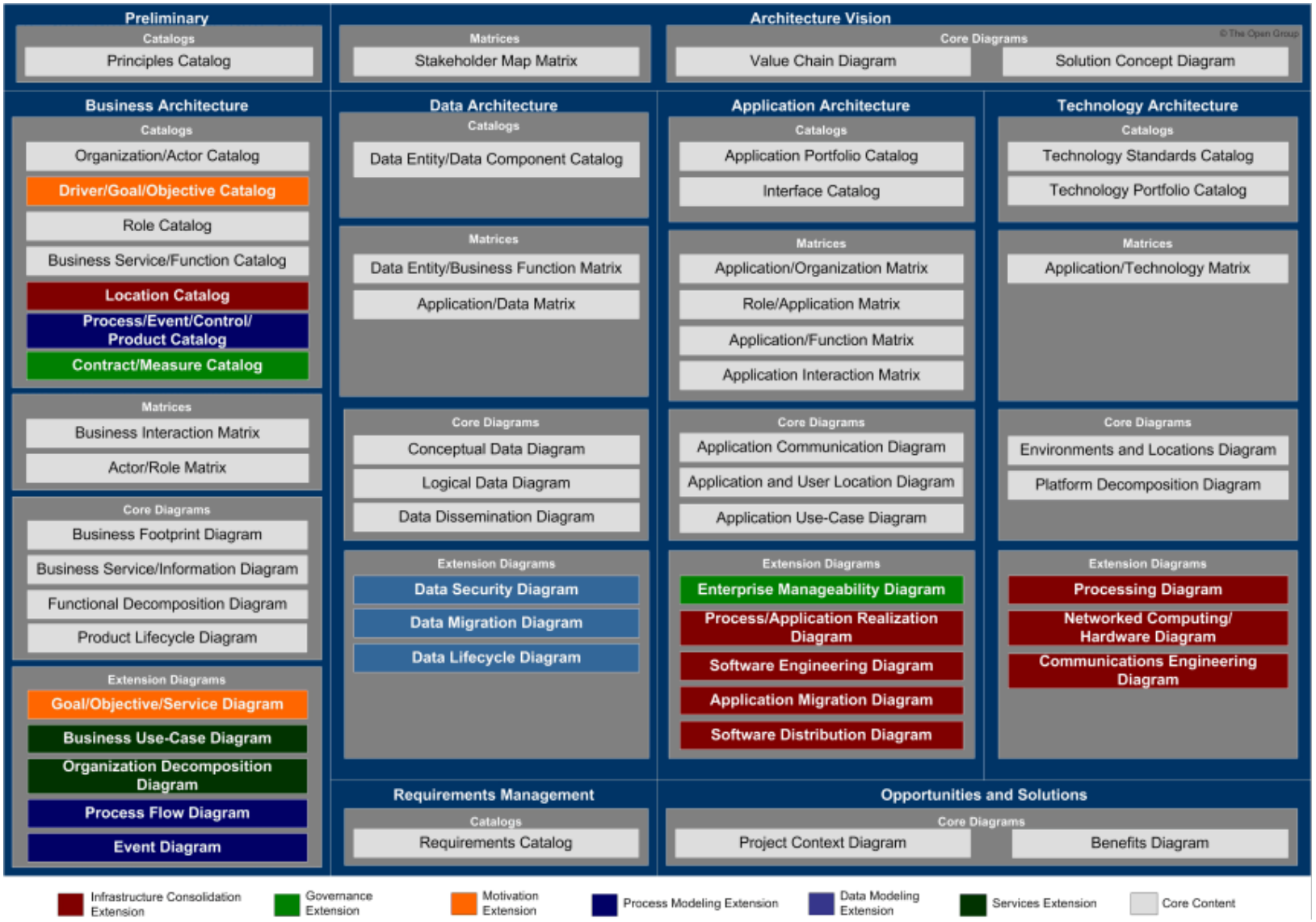




Content Metamodel



Architectural Artifacts (Work Products) by ADM Phase



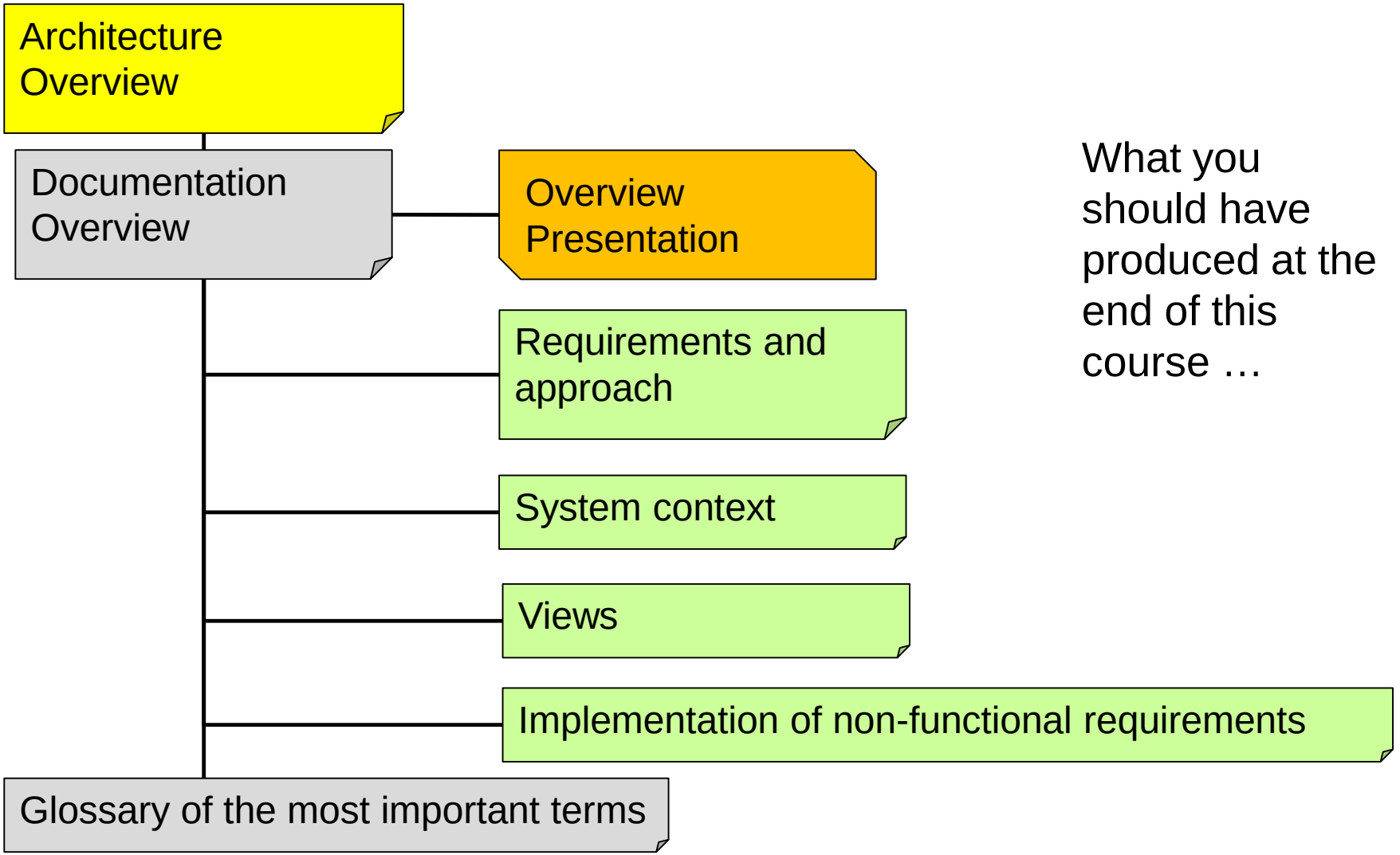
Guidelines for a good Architecture Documentation

- Provide clarity and ensure uniform understanding among stakeholders
  - Define and use terms consistently
  - Ensure consistency of diagrams and views
- Define a structure for the documentation and communicate this structure to all stakeholders
- Document «**Why» something is like it is**
  - Decisions, alternatives, result with rationale
- Avoid redundancy

Questions a good Documentation should answer

- How does the system fit into its environment, especially its technical infrastructure?
- How is the system structured as a set of implementation units and what are the relationships between them?
- How do the modules behave at runtime and how do they work together?

Possible structure of an Architectural Documentation



What you should have produced at the end of this course …

Read More

- http://www.iso-architecture.org/ieee-1471/

- Uwe van Heesch, Paris Avgeriou, and Rich Hilliard. A documentation framework for architecture decisions. The Journal of Systems & Software, 85(4):795–820, April 2012. https://dl.acm.org/citation.cfm?id=2148467

Summary

- Managing requirements as key method to control risk
- SMART requirements

- Understand constraints and stakeholders

- Enterprise Architecture provides tools for application/ solution architects – AD guidelines & work products

- Always document decisions precisely and at a detailed level
  - Insurance of the architect in the event of a crisis
  - “why” some decision was taken
    - Surprisingly frequently: “because the customer wanted it”

Working Questions

1. Which main work products should an architectural documentation include?
2. How do you document decisions according to ISO 42010?
3. What do we mean by a concern?
4. What role do forces and constraints play in a decision?
5. Why should we precisely list the affected elements of the architecture in the documentation of the decision?
6. Why do we need to list the considered alternatives in the decision?