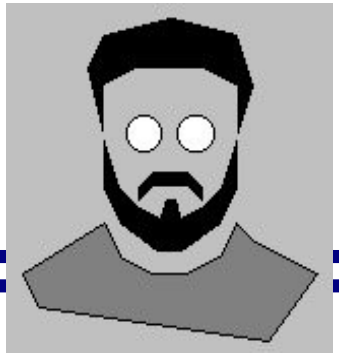
Software (Systems) Architecture Foundations

Introductory Lectures

Alar Raabe

Alar Raabe



- Nearly 40 years in IT
  - held various roles from programmer to a software architect and to enterprise business architect
- 15 years in insurance and last 10 years in banking domain
  - developed model-driven technology for insurance applications product-line (incl. models, method/process, platform/framework and tools)
  - developing/implementing business architecture framework and methods for a banking group
- Interests
  - software engineering (tools and technologies)
  - software architectures
  - model-driven software development
  - industry reference models (e.g. IBM IAA, IFW, ...)
  - domain specific languages

Course Purpose

Architecture is about:
- ❖ Durability (*firmitas*)
- ❖ Utility (*utilitas*)
- ❖ Beauty (*venustas*)

Marcus Vitruvius Pollio
(Rome, 1st century BC)

- Purpose – to provide understanding of
  - the core concepts in the discipline of software (systems) architecture
  - overview of different architecture styles
  - how software architecture affects quality attributes of the software systems
  - the value of software architecture and the architecture decisions
- Results
  - General understanding of the related concepts and techniques
  - Basic skills to
    - describe the architecture of software systems
    - evaluate the architecture of software systems
    - reason upon architectural decisions
    - organize the architecture work and governance

(Some) Sources

Mostly CMU SEI

[1] Mary Shaw, David Garlan, **Software Architecture, Perspectives on an Emerging Discipline**, Prentice Hall,1996

[2] Len Bass, Paul Clements, Rick Kazman, **Software Architecture in Practice**, 2nd ed., Addison-Wesley, 2003

[3] Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Robert Nord, Judith Stafford, **Documenting Software Architectures, Views and Beyond**, Addison-Wesley, 2003

[4] PaulClements, Rick Kazman, Mark Klein, **Evaluating Software Architectures, Methods and Case Studies**, Addison-Wesley, 2001

[5] R. T. Fielding, **Architectural Styles and the Design of Network-based Software Architectures**, UCI, 2000

[6] ISO/IEC/IEEE 42010, **Systems and software engineering — Architecture description** (IEEE 1471)

[7] Open Group, **TOGAF** 9

[8] Open Group, **ArchiMate** 3

- … Google “Software Architecture” …

Why to bother with Architecture

If a builder builds a house for someone, and does not construct it properly, and the house falls and kills its owner, then that builder shall be put to death

Hammurabi, King of Babylon (1780 BC)

- Architecture is **important**
  - as a **cause of certain properties → designing architecture allows us to address concerns** and to achieve required and desirable properties of software systems
  - as **fundamental conception** of software system → **architecture allows us to reason** (i.e. answer questions) about the software system and its properties, and foresee those properties without building and testing the actual system
  - because it **creates choices/options**, which have value → designing and building an **architecture is an investment** activity

Content of the Course

Software Architecture and Software Architecture Styles

Software architecture is what software architects do

Kent Beck

- What we call Architecture and why we need to bother with it
  - Design vs. Architecture and early views on Software Architecture
- Concepts and Terminology related to Software Architecture Description (ISO/IEC 42010)
- Other related Concepts (Abstraction, Complexity, Modularity, Model)
- Different Levels of Commonality in Software and Software Architecture Style
- Classifications of Software Architecture Styles and Analysis of some Software Architecture Styles
  - Main styles (e.g. data flow, data centered, …)
  - “Modern” styles (e.g. micro-services, map-reduce, …)
  - Derived (Complex) architecture styles
  - Designing an Architecture Style (on example of REST) and Using Software Architecture Styles

Content of the Course

Documenting and Evaluating Software Architecture

Designing an architecture without documenting it, is like winking at a girl in the dark – you know what you´re doing, but nobody else does

E. Woods

- Why to Document Software Architecture
- CMU SEI “Views & Beyond” Method and some other Documentation Methods
- Other Architecture Documentation Practices
  - Architecture Description Languages
  - Documenting Architecture in Code

Quality mean doing it right when no one is looking

Henry Ford

- Architecture and Requirements, Software Quality Models and Software Quality Attributes
- Measuring Software Quality and Evaluation of Software Architectures
  - Quality Attribute Scenarios, Architecture Trade-off Analysis Method (ATAM) and Software Metrics
- Cost and Value of Architecture
  - Valuation of Architecture Decisions (Option Value of Architecture Decisions)

Larger Context and from One System to Many

The significant problems we face cannot be solved by the same level of thinking that created them

Jeanne W. Ross

- Larger context
  - Hierarchy of Systems and Systems of Systems
- Enterprise Architecture
  - Need for Larger Context and Structured Approach
  - Some Approaches to Enterprise Architecture
  - Standard for Enterprise Architecture (on example of TOGAF)
  - Reference Architectures and their Use (on example of BIAN and IBM Industry Models)
  - Language for Enterprise Architecture (on example of ArchiMate)
- From one system to many – System Families and Product-Line Architectures
- Model-driven development
  - Model as Primary Artifact
  - Generative Programming
  - Feature Modeling

Content of the Course

Architect Role, Architecture Work and to the Future ...

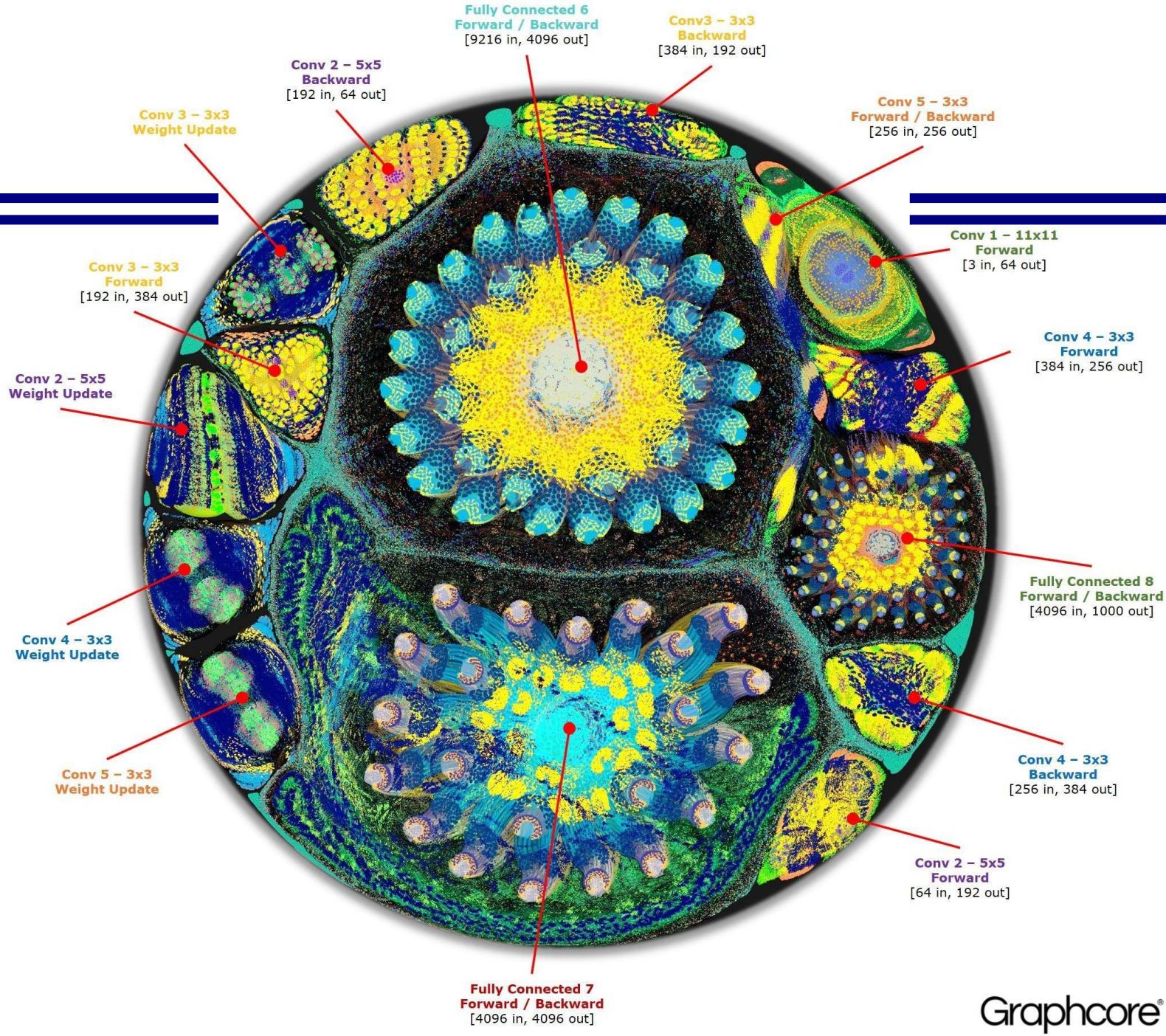
If architecture is the important stuff, then the architect is the person who worries about the important stuff

Martin Fowler

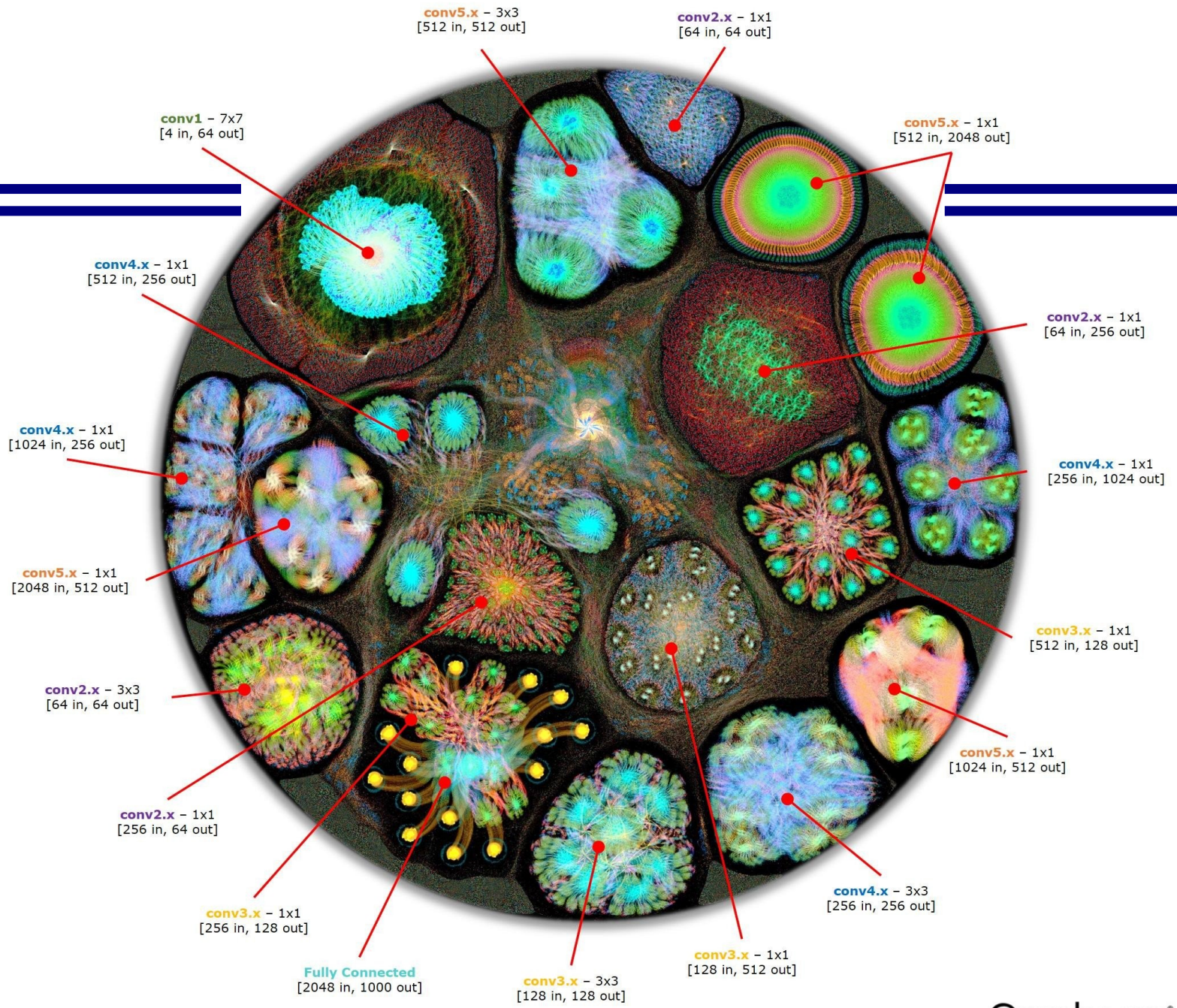
- Role of Architect and Architecture work
  - Architecture Process
  - Architecture Governance
- Architecture in the context of agile development
  - Scaled Agile Framework (SAFe)
  - Disciplined Agile (DA)
- Architecting for Cloud
  - Principles and Design Patterns
- Architecture of Adaptive Systems
- Architecture of AI
  - Machine Learning Systems
  - Cognitive Systems Architecture
  - Neural Networks (Architecture Styles)

The best way to predict the future is to invent it

Alan Kay, 1971



Graphcore®

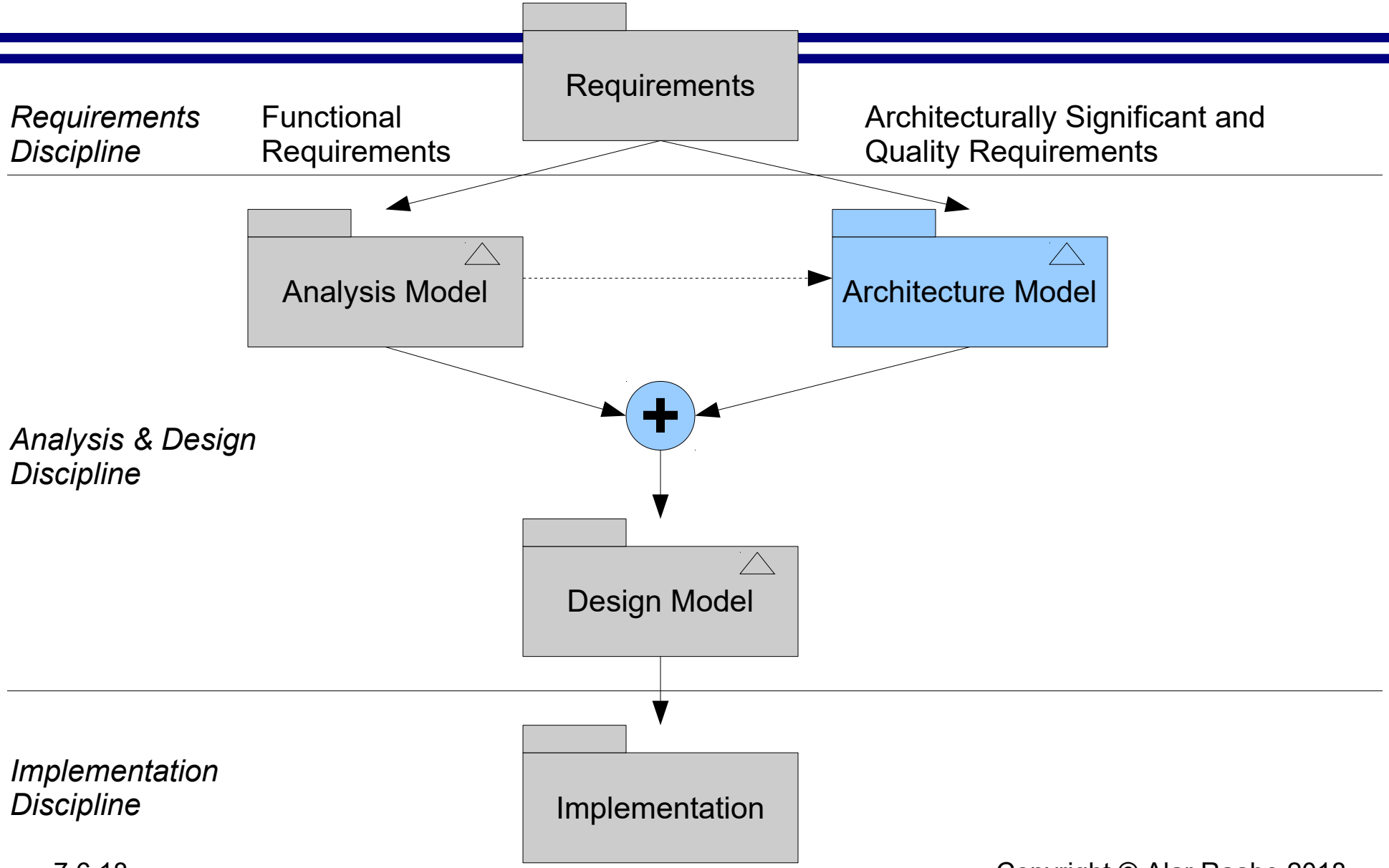


55. The architect lets all things come and go effortlessly, without desire. He never expect results; thus he is never disappointed. He is never disappointed, thus his spirit never grows old.

Lao Tsu (by Philippe Kruchten)

Thank You!

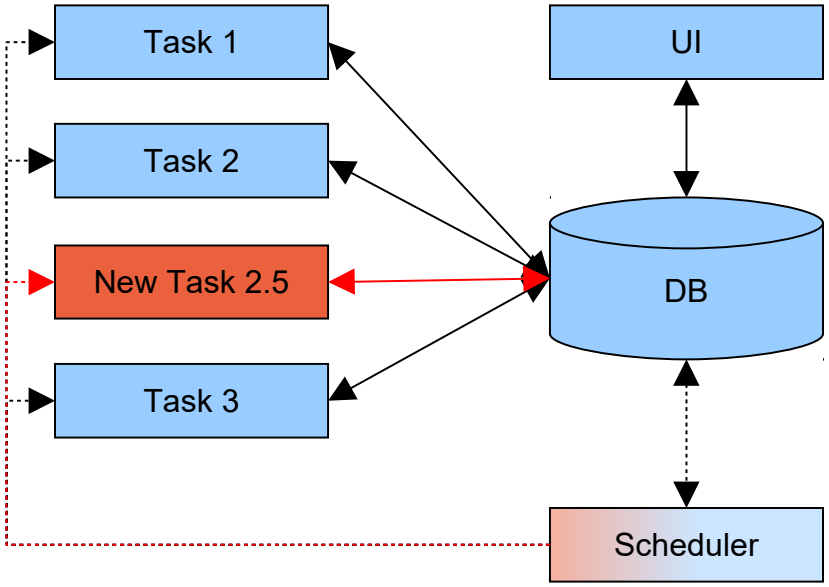
Creating Architecture

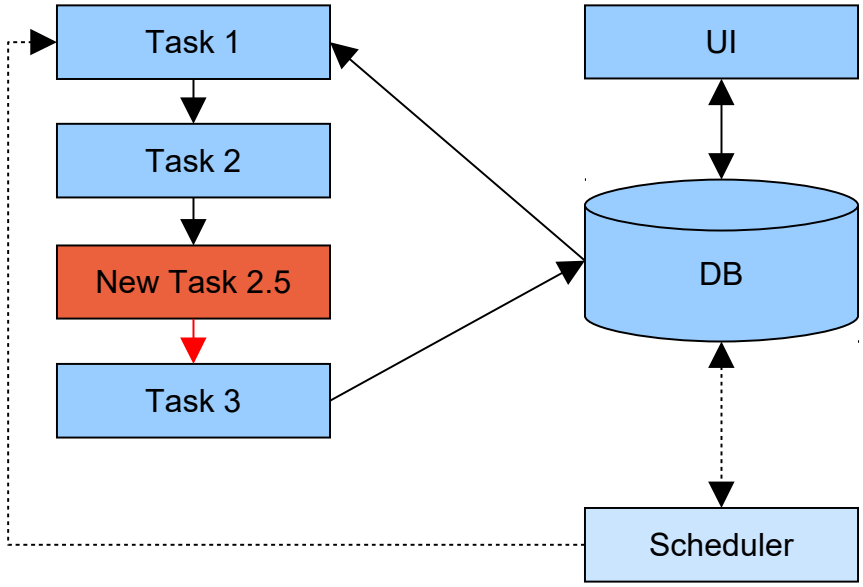


Example

Different Architecture Styles → Different Properties

adding new task

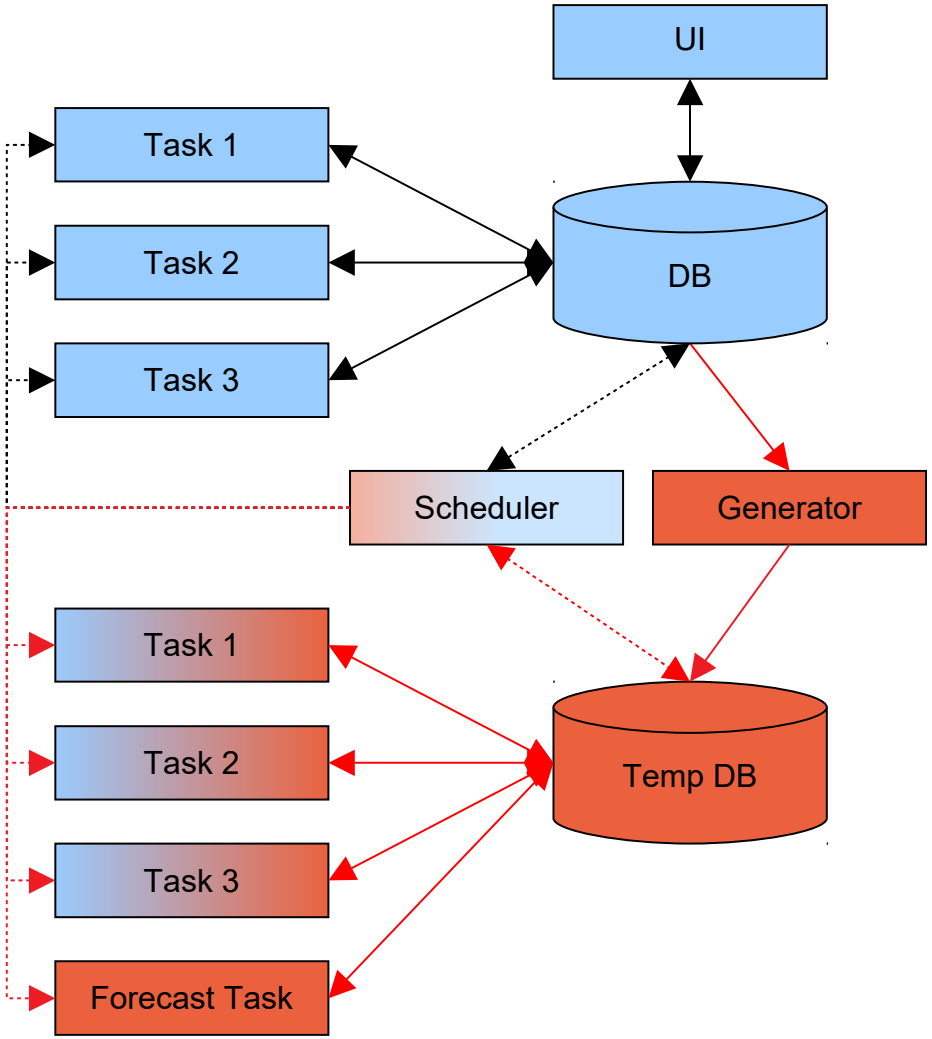


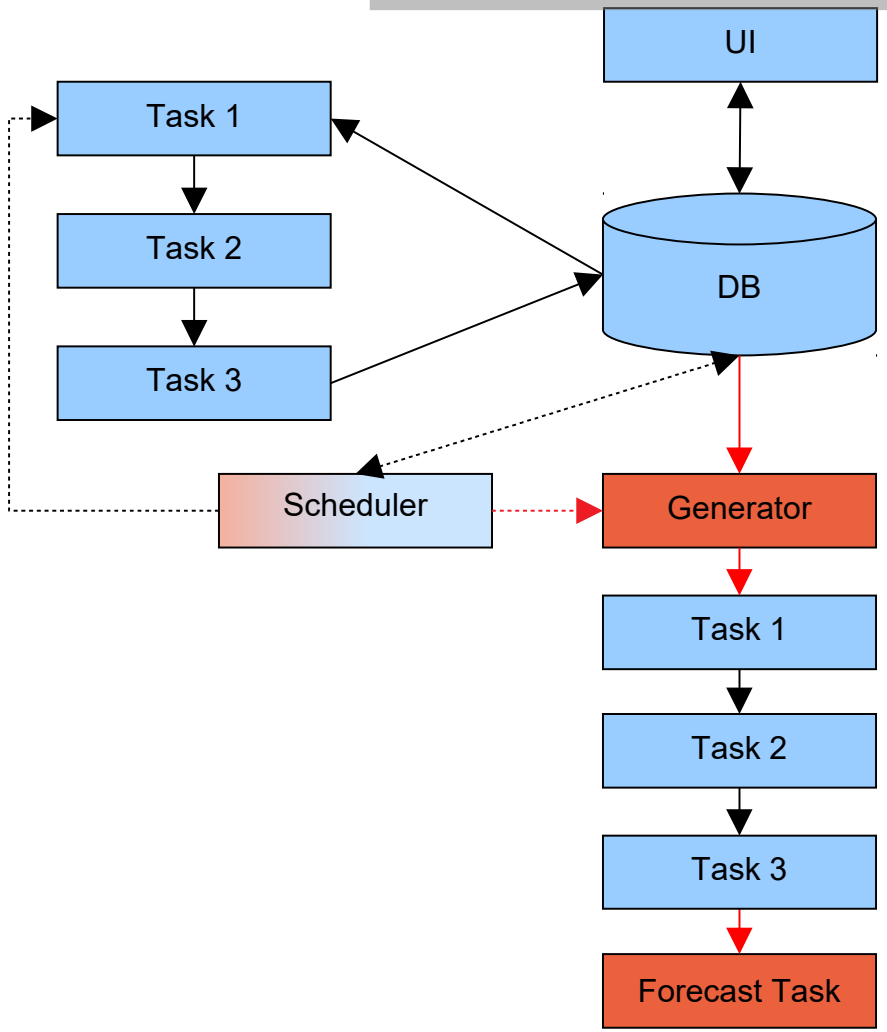


Example

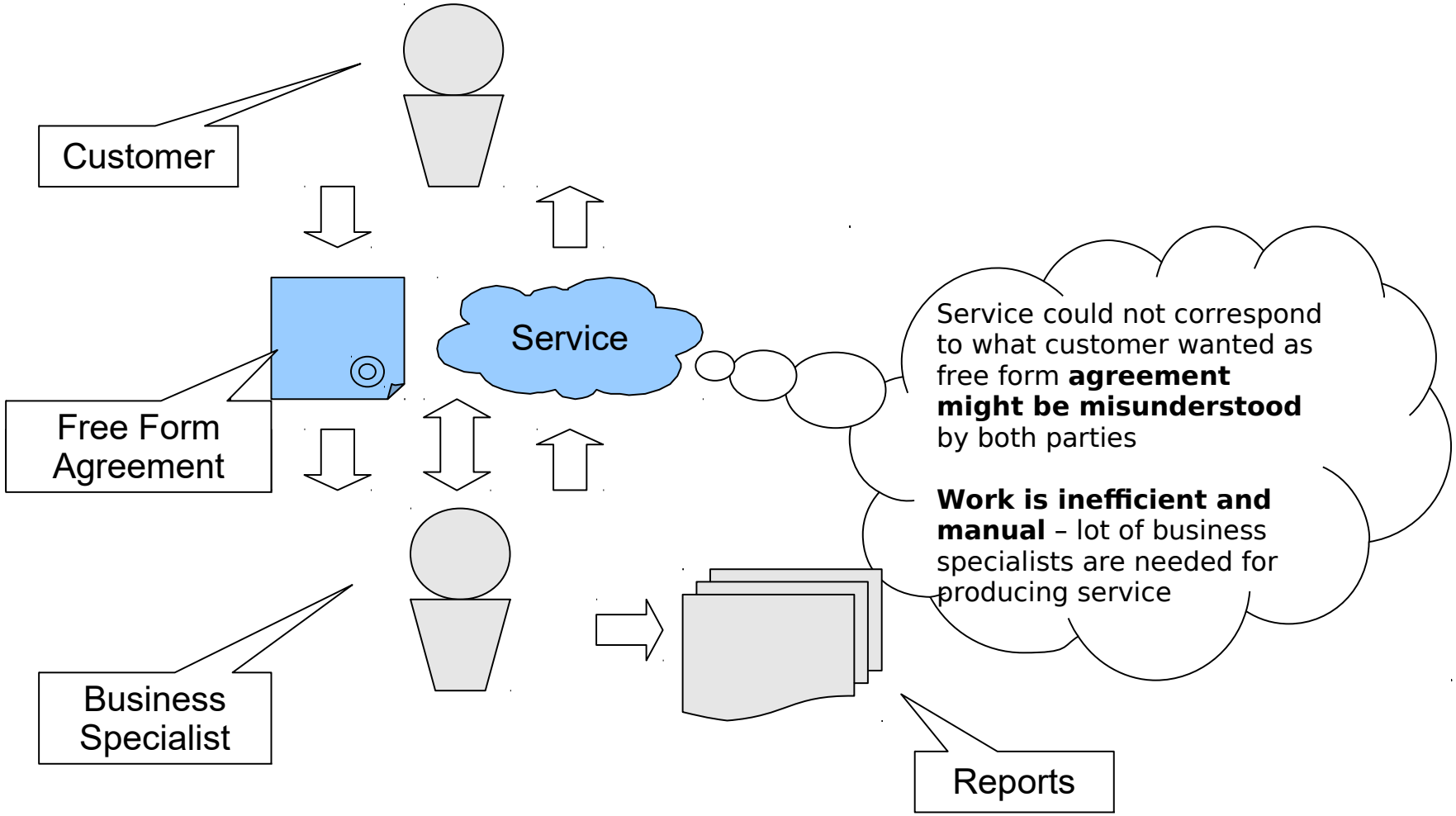
Different Architecture Styles → Different Properties

adding forecasts of portfolio

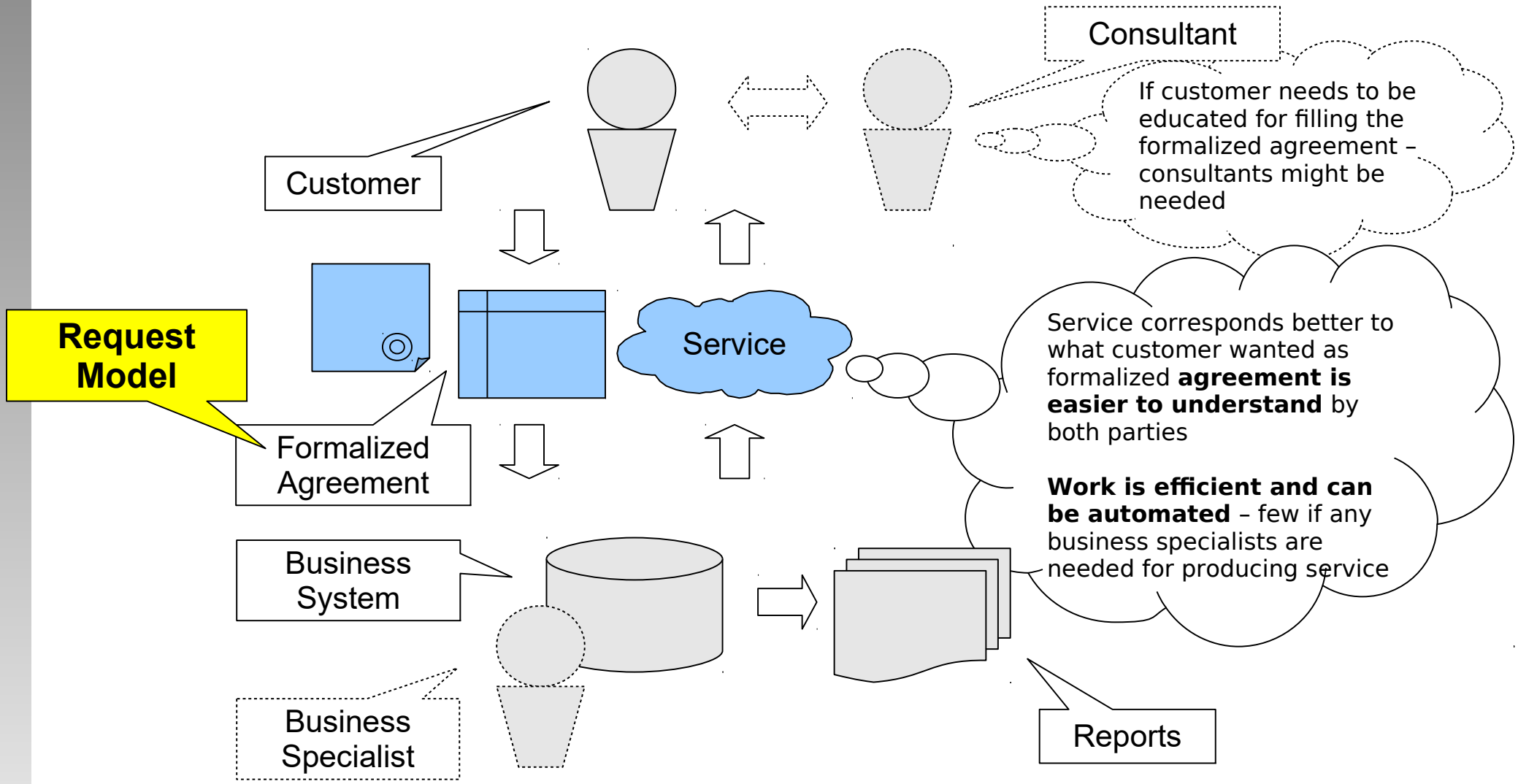




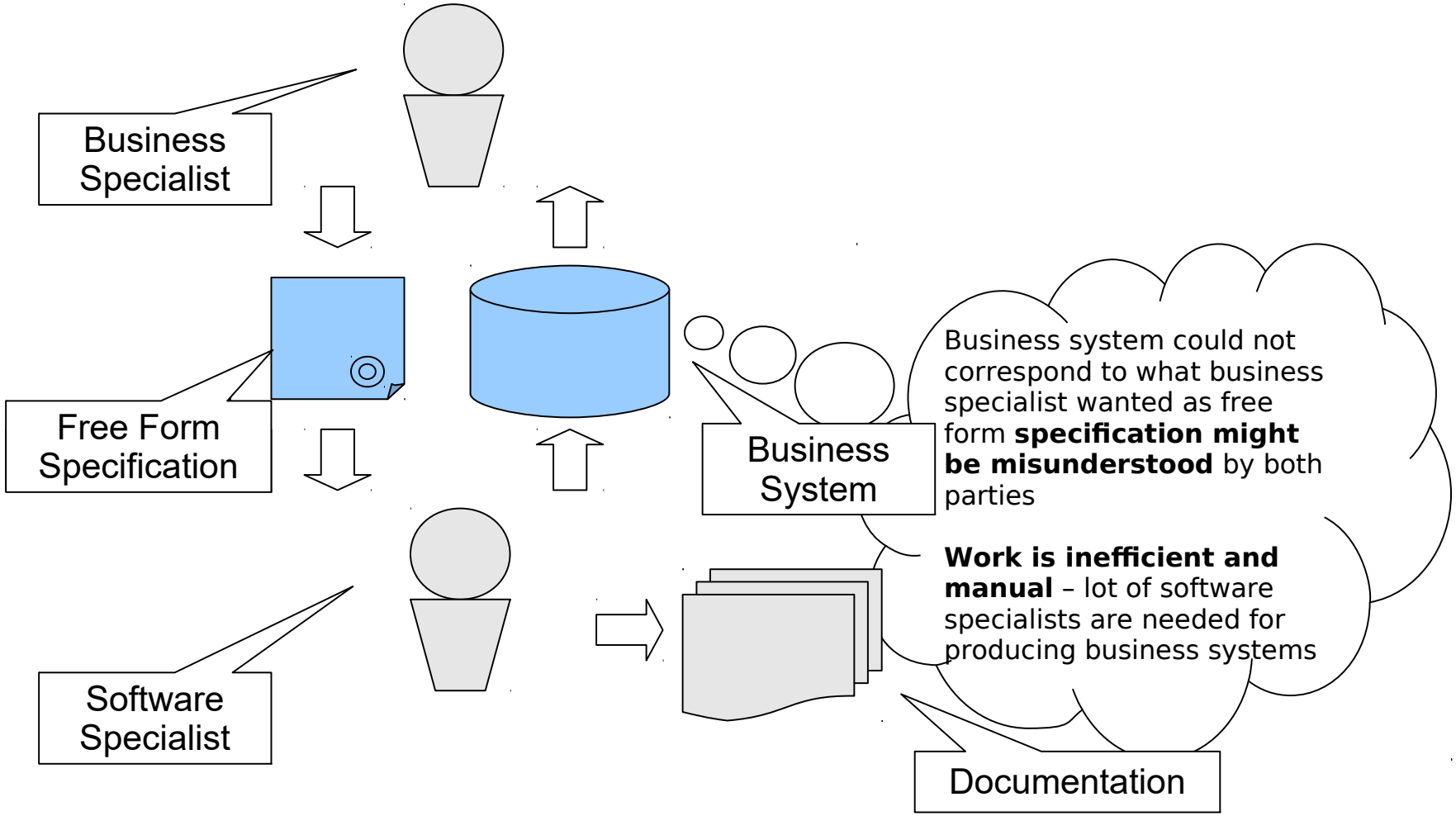
How we did Business Yesterday



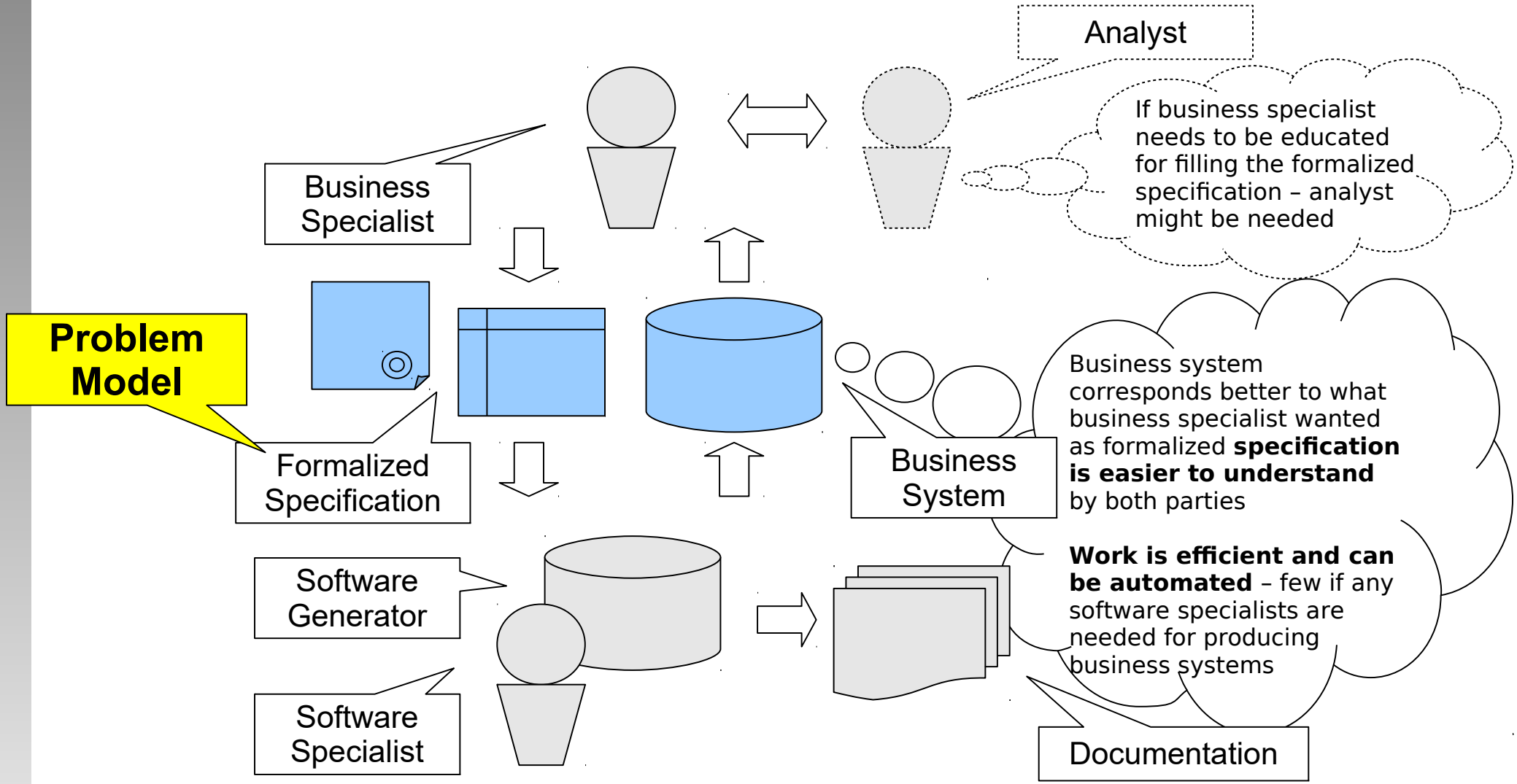
How we do Business Today

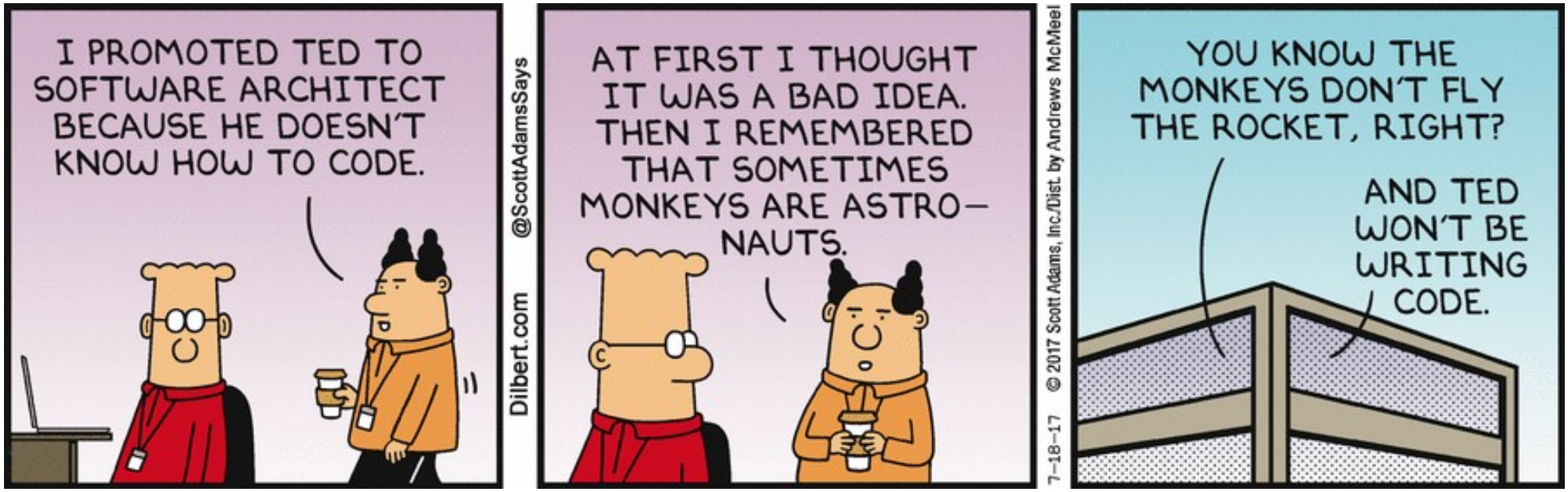


How we Develop Software Today



How we should Develop Software




I PROMOTED TED TO SOFTWARE ARCHITECT BECAUSE HE DOESN'T KNOW HOW TO CODE.
Dilbert.com @ScottAdamsSays
AT FIRST I THOUGHT IT WAS A BAD IDEA. THEN I REMEMBERED THAT SOMETIMES MONKEYS ARE ASTRO-NAUTS.
7-18-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel
YOU KNOW THE MONKEYS DON'T FLY THE ROCKET, RIGHT?
AND TED WON'T BE WRITING CODE.