Adaptable Software Architecture

ECSA 2014 - Vienna

Reto Carrara

ACTIF® SOURCE

Software Architecture allows you to implement all the features demanded by the market – even in the future

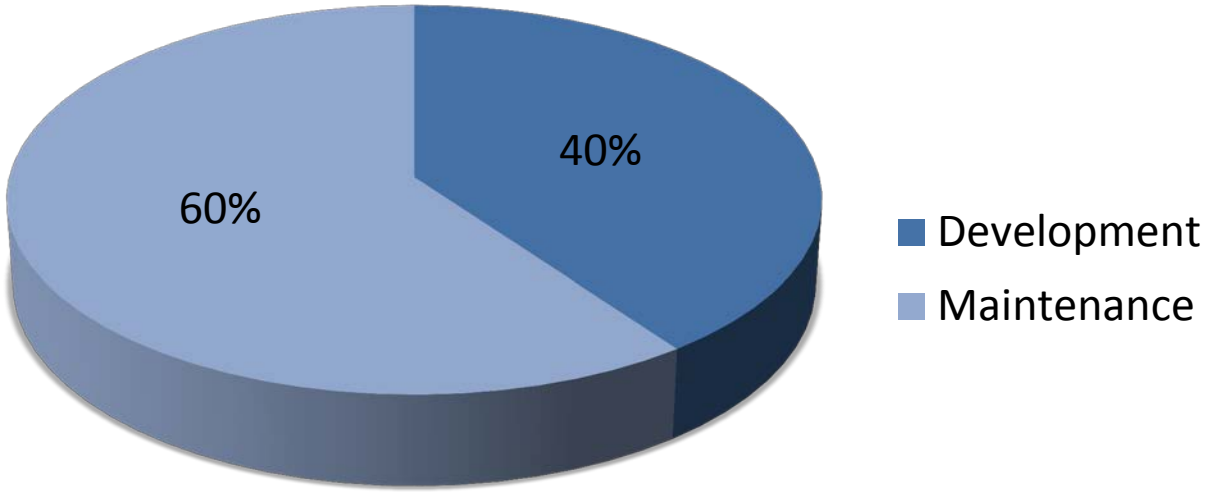
But what if this is no longer true?

Definition of Refactoring

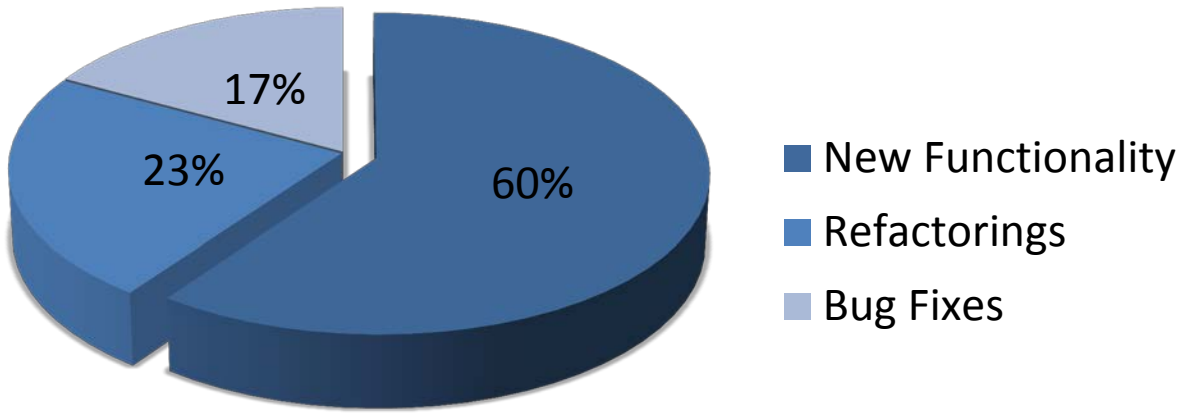
Refactoring is the process of
restructuring existing code
without changing its external behavior

Change lies at the heart of software.

«Software Aging»
David Lorge Parnas: Software Aging. Invited Plenary Talk.



Reference: Robert C. Glass, 2003



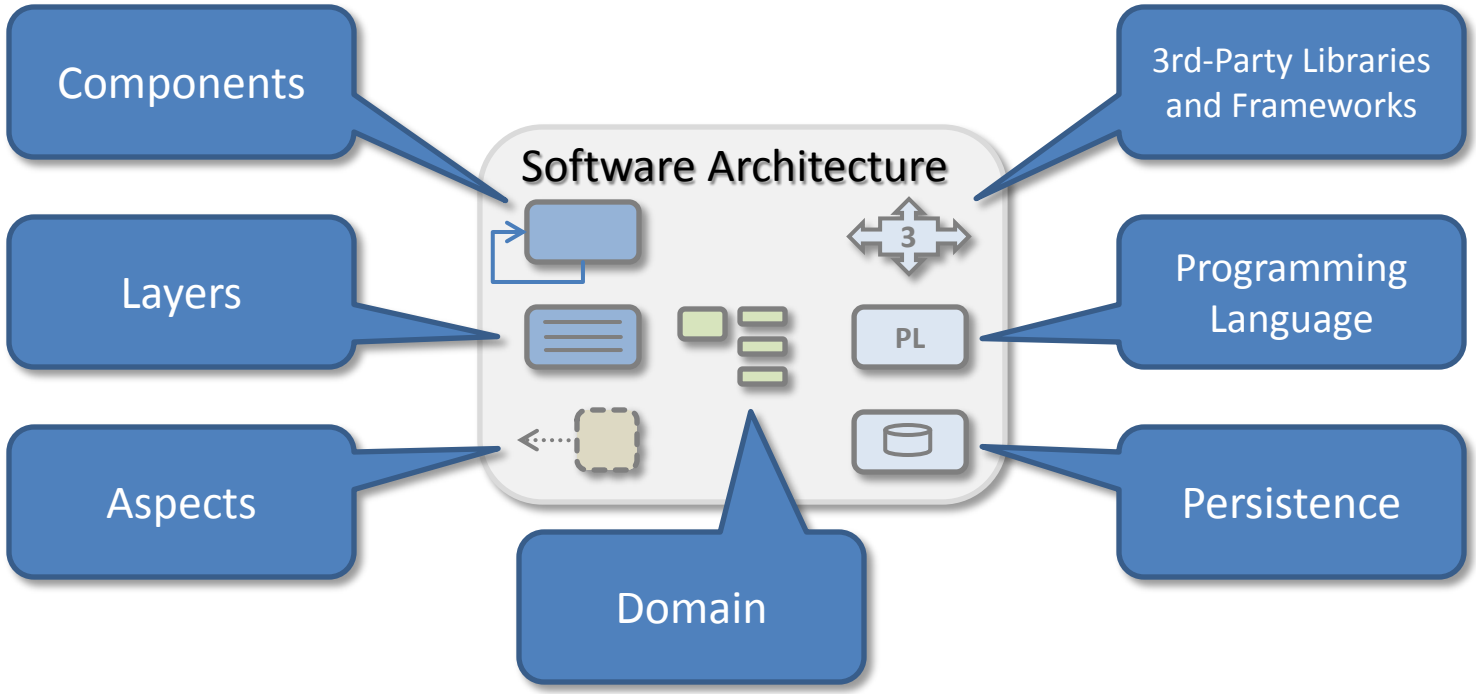
Reference: Robert C. Glass, 2003

Maintenance is a solution, not a problem

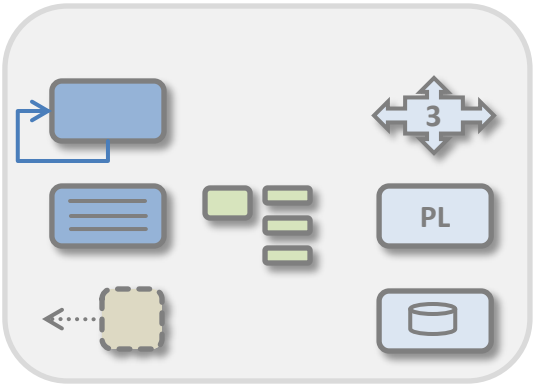
Robert C. Glass: Facts and Fallacies

Why is it so hard to
adapt software and its architecture
when business requirements change?

Every Software Architecture consists of numerous concepts.

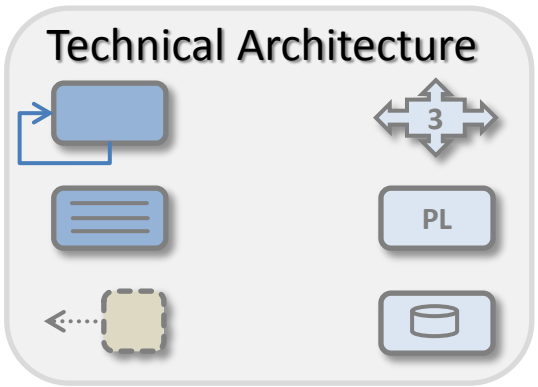


There are two different types of Software Architectures



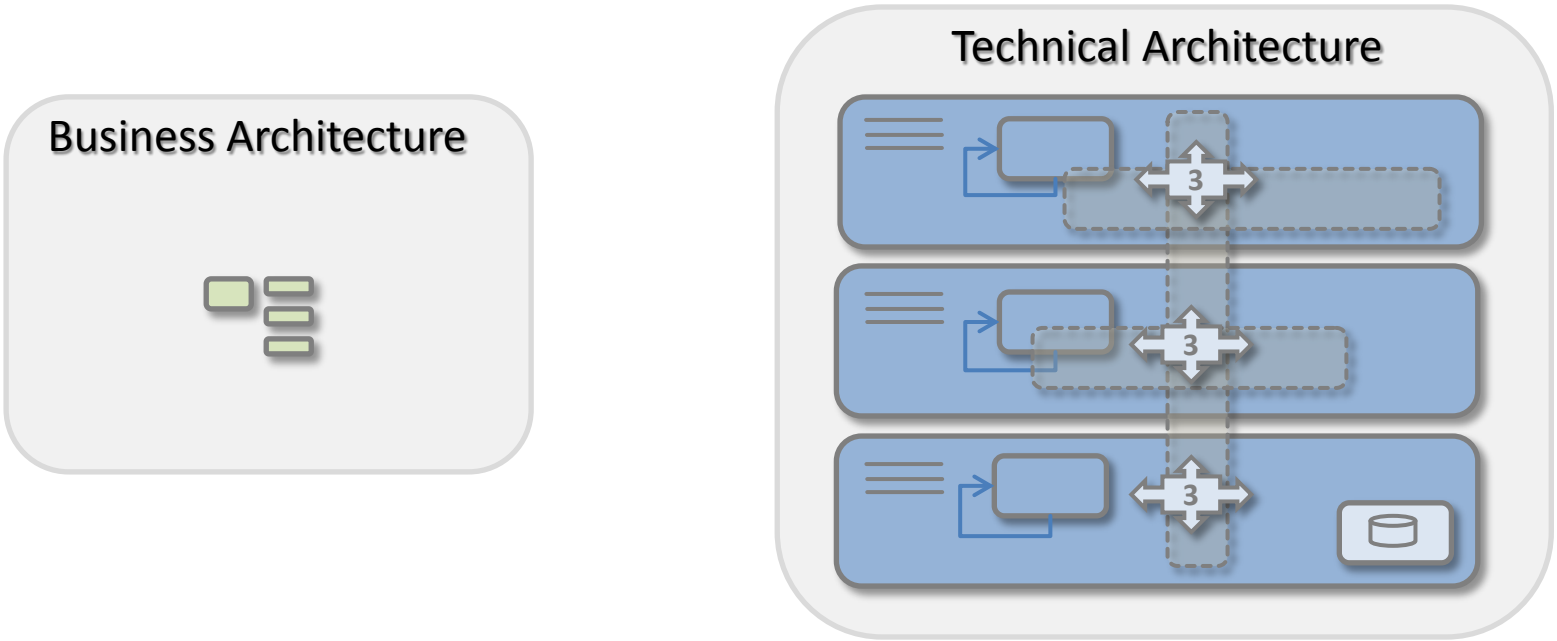
Let’s distinguish between Business Architecture and Technical Architecture

The **Business Architecture** comprises **Business Objects**, the externally visible **Properties** of those Objects, and the **Relationships** among them.



The **Technical Architecture** realizes the **Business Architecture** by implementing **Business Objects** using **Technical Concepts.**

A single Business Object is implemented by several different Technical Concepts

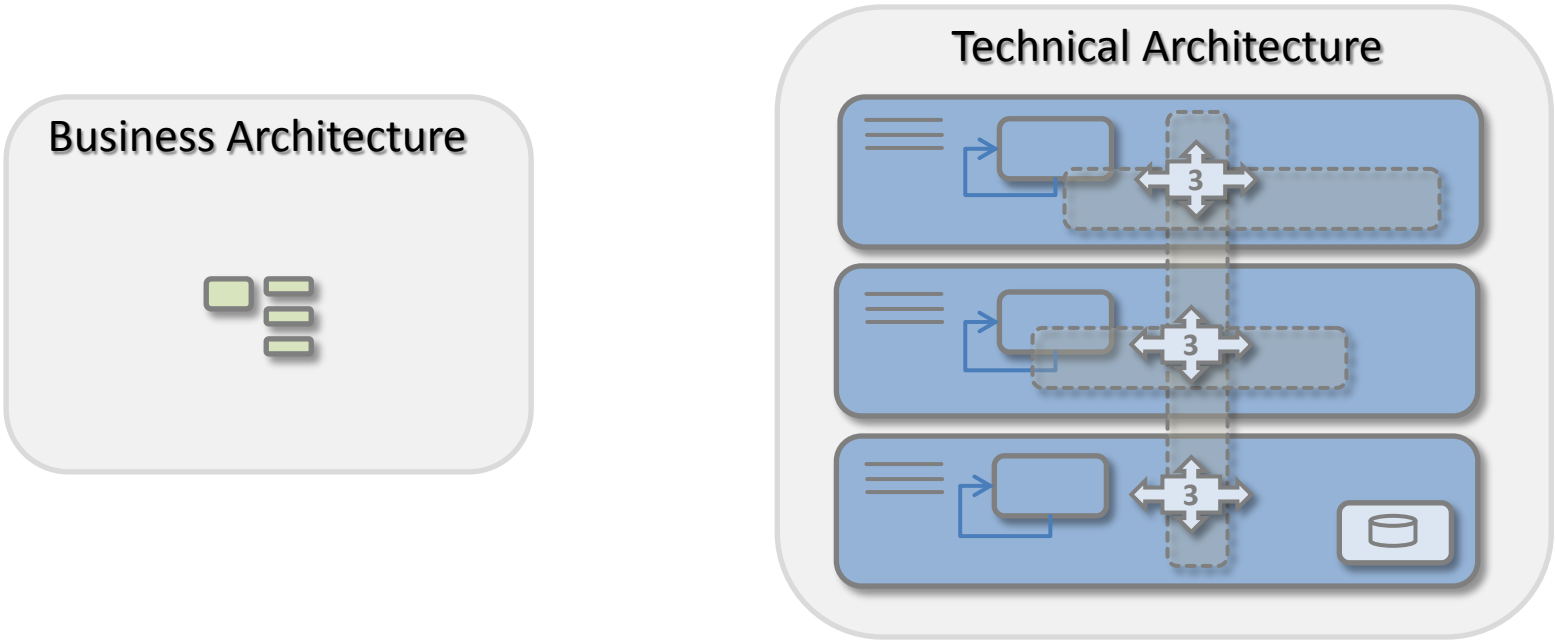


Quantity Structure:

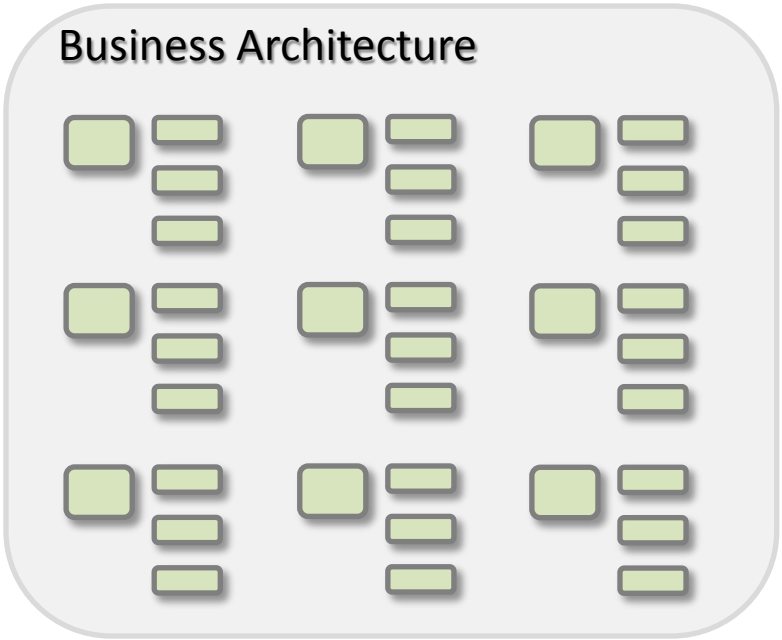
N Business Objects lead to N x M Technical Artifacts

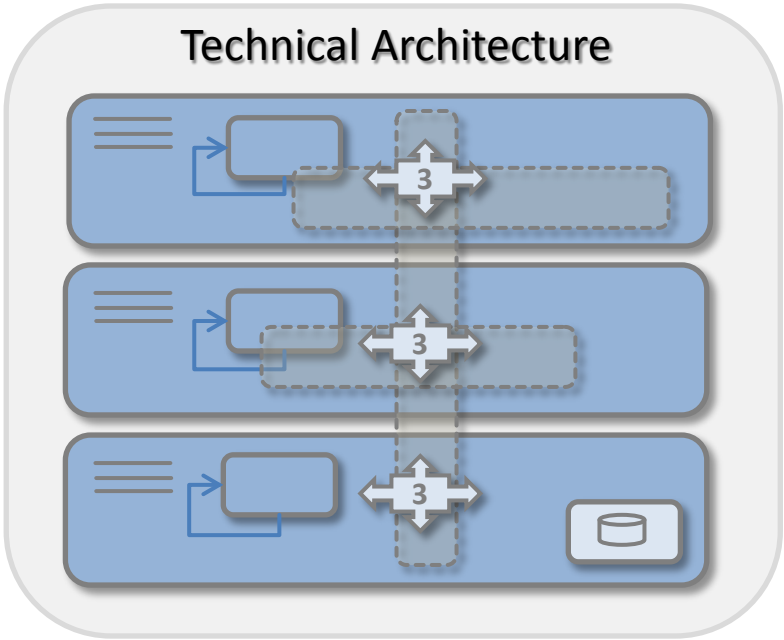
Complexity by Quantity

Over the years the Business Architecture grows big



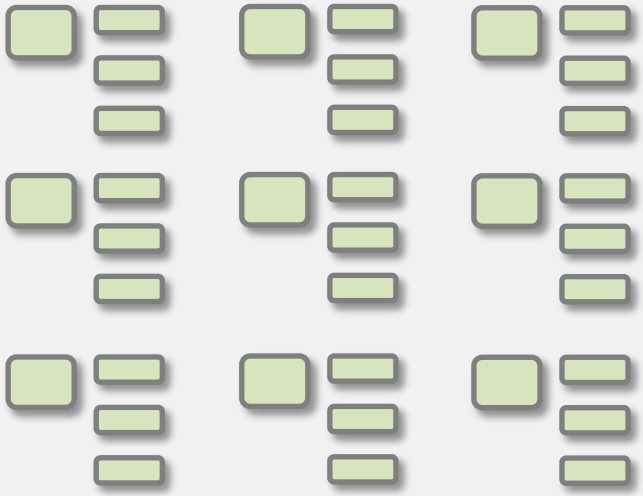
Over the years the Business Architecture grows big



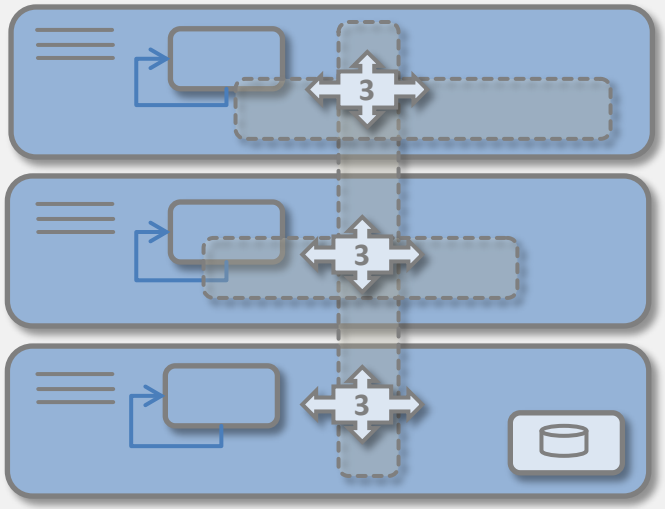


Therefore, the Technical Architecture is increasingly harder to change.

Business Architecture



Technical Architecture



Stable (over years)

- Business remains the same
- Valuable Business Knowhow

Less Stable

- Disrupting Technologies
- New Market Needs

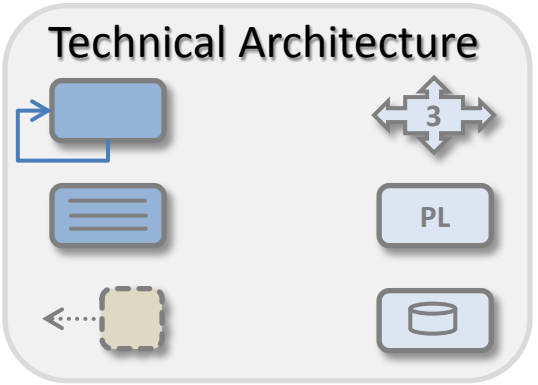
What if we could separate the highly valuable Business Architecture from the Technical Architecture?

Writing Code means to weave it all together

We suggest to create

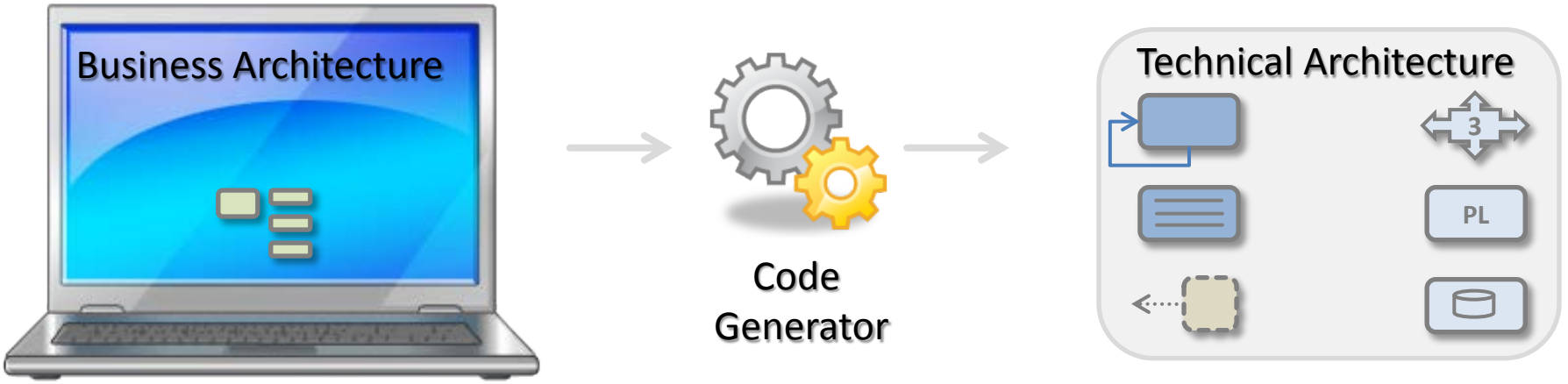
Business Specific Development Tools

to manage the Business Architecture

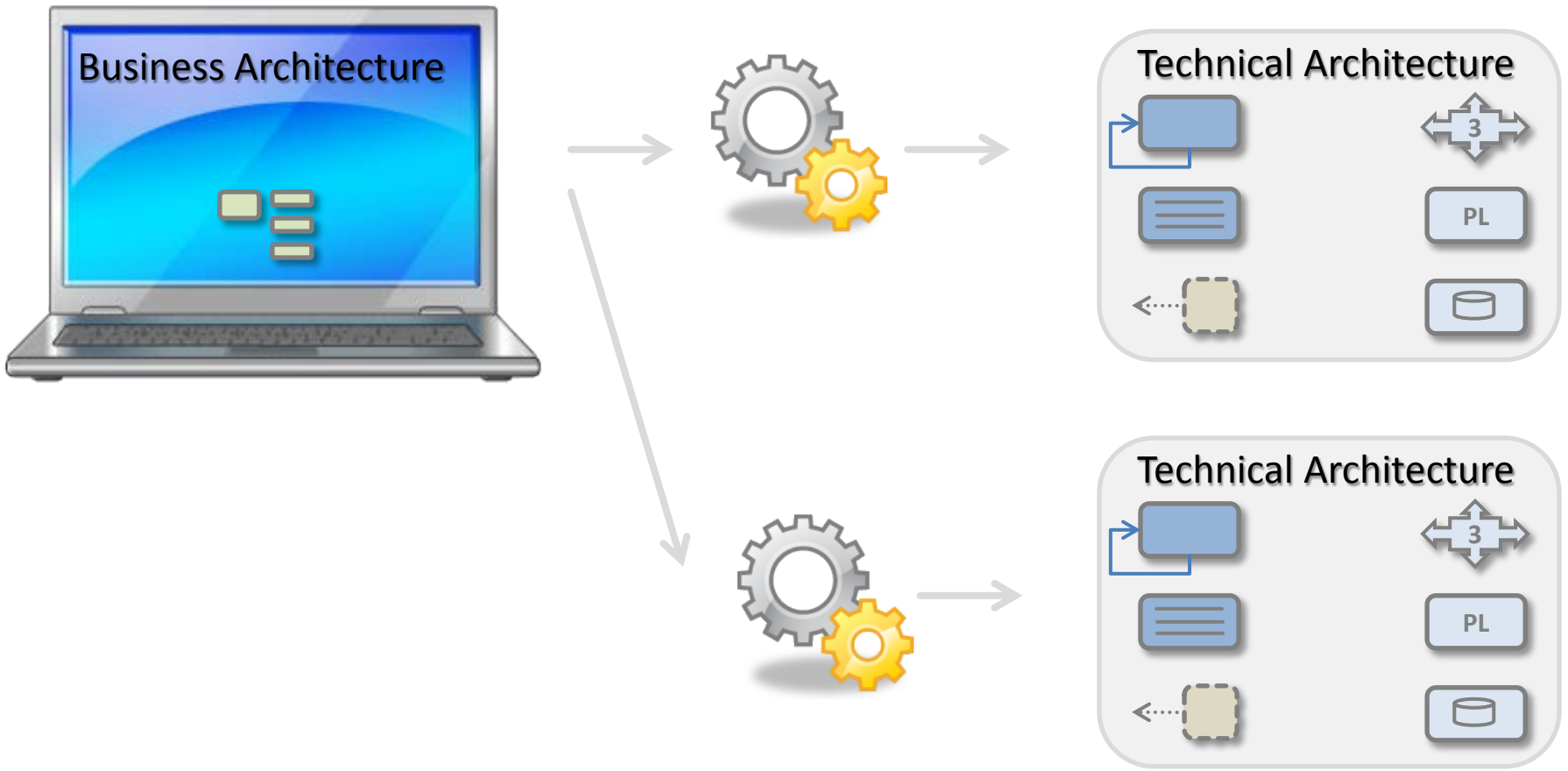


independently from the Technical Architecture.

The Technical Implementation is generated from the Business Model



Changing the Technical Architecture means to just exchange the Code Generator



Résumé

A single Business Object is implemented by several different Technical Concepts

N Business Objects lead to N x M Technical Artifacts
Complexity by Quantity

The Business Architecture remains stable over years
The Technical Architecture is less stable and may change

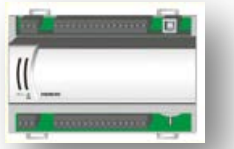
In order to manage the Business Architecture independently we suggest to use an appropriate tool

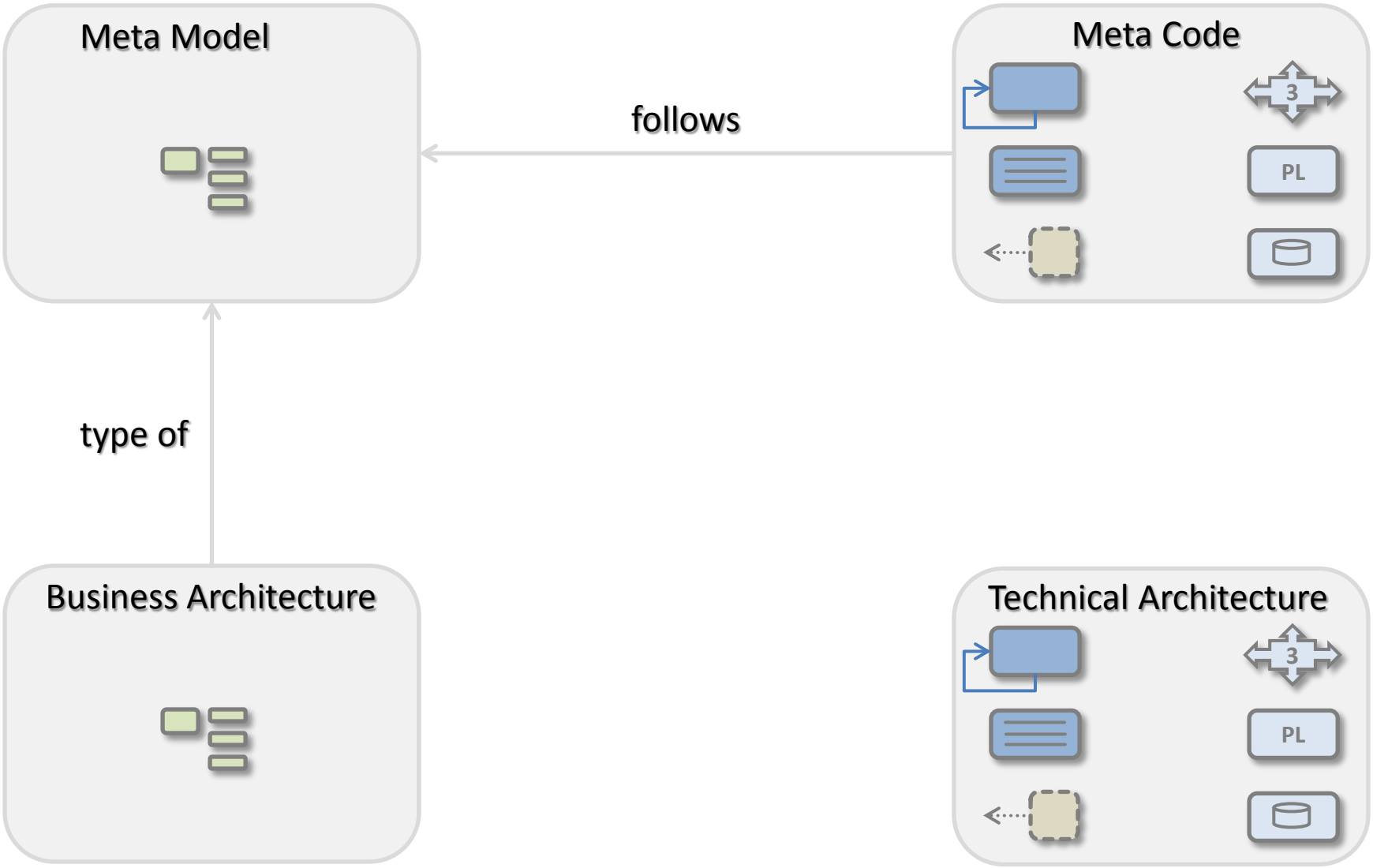
Over the last couple of years we created a tool to create

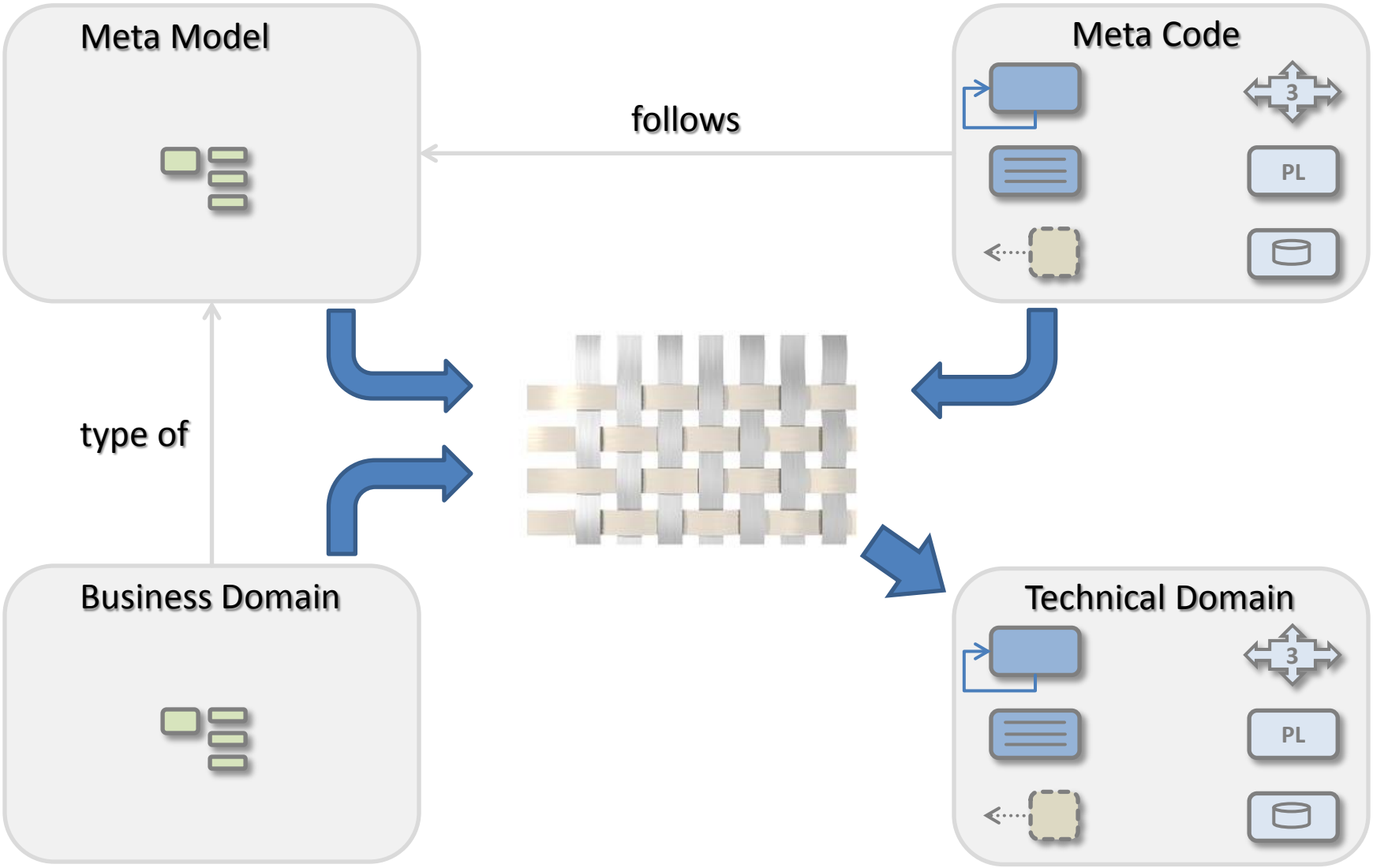
Business Specific Development Tools

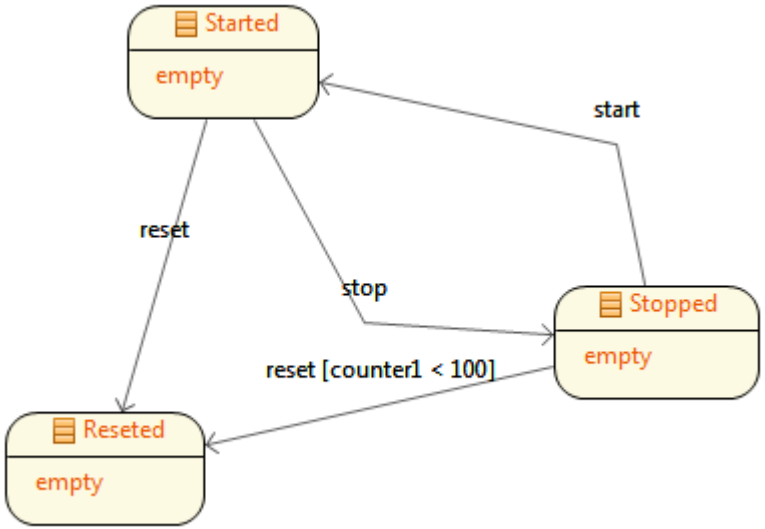
Let's Build Your **Business Specific Development Tool**
That Turns Your **Business Architecture** Into **Running Code**.

typeOf	**com.siemens.bt.ba.metamodel.generic.Object**
name	**BAAnalogInput**
description	BA-Analog-Input-Object refers to a BA-Object which represents standardized characteristics of a physical analog variable sensed by Peripheral Device.
id	4
extends	BAInput
mainProperty	com.siemens.bt.ba.model.object.BAAnalogInput.PresentValue
property[1]	PresentValue : **Property**
property[2]	TrackingValue : **Property**
property[3]	Units : **Property**
property[4]	Resolution : **Property**
property[5]	MinPresValue : **Property**
property[6]	MaxPresValue : **Property**
property[7]	CorrectionFactor : **Property**
property[8]	CorrectionOffset : **Property**
property[9]	COVIncrement : **Property**
privateProperty	




Meta Model
follows
Meta Code
3
PL
type of
Business Architecture
Technical Architecture
3
PL


Meta Model
follows
Meta Code
3
PL
type of
Business Domain
Technical Domain
3
PL



Let’s build a Development Tool