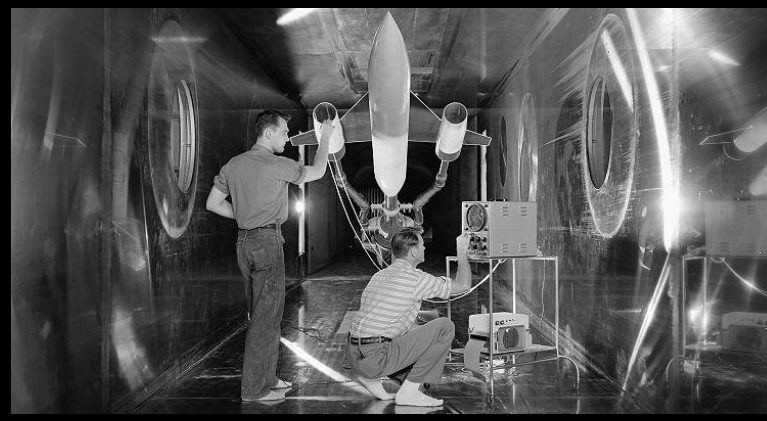
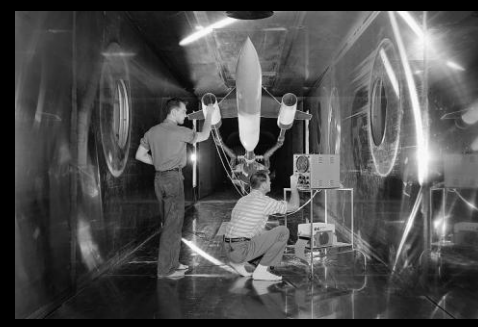
Pillars and Trends in Software Architecture

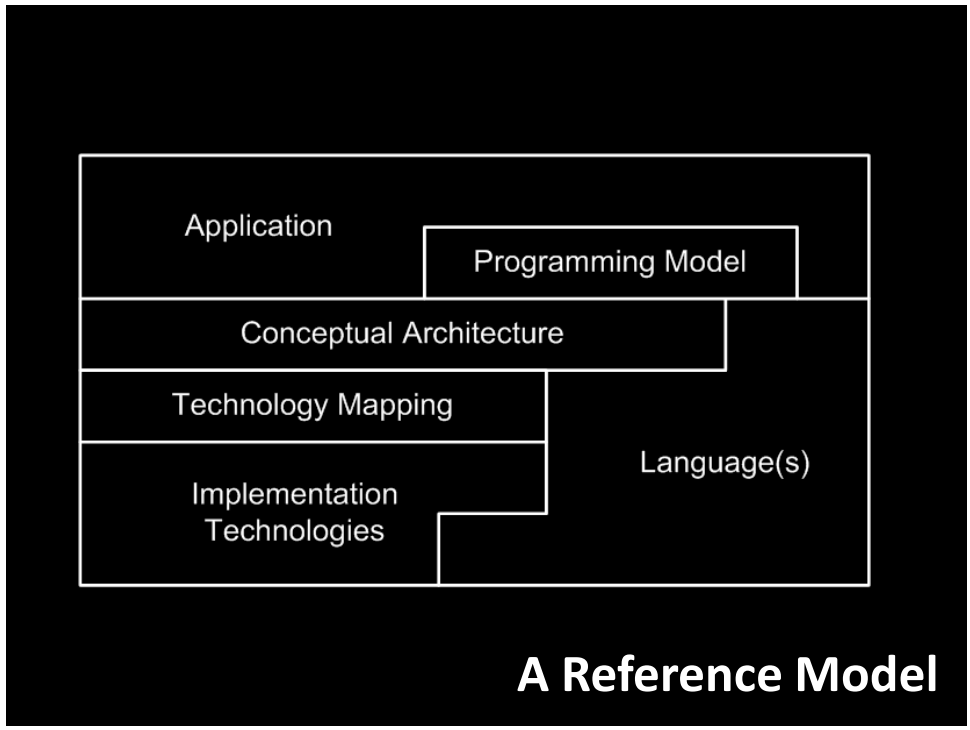


itemis

Markus Voelter
www.voelter.de
voelter@acm.org

What is Architecture?

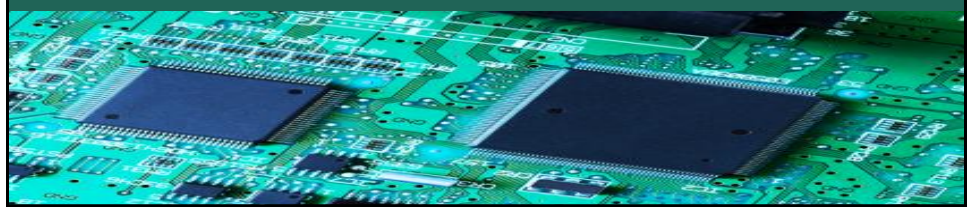




A Reference Model

Modularize

Procedures, Classes,
Components, Services, User Stories



! Modularize

Encapsulate

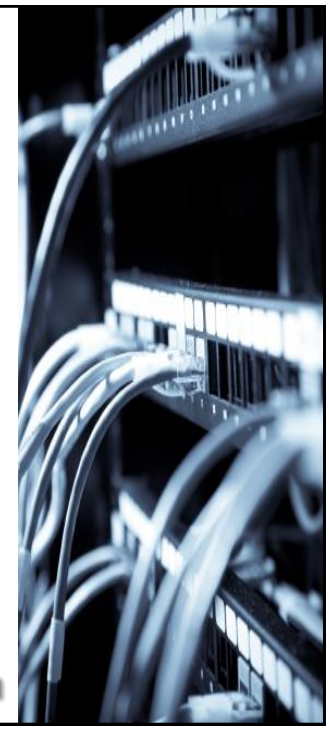
Private Members
Frameworks
Facade Pattern
Components
Layers/Rings/Levels
Packed Data Wrapper

! Modularize

Contracts

Interfaces
Pre/Post Conditions
Protocol State Machines
Message Exchange Patterns
Published APIs

Decoupling



Compensating Tx
Message Queues

! Modularization

Isolate

Pure functional vs. Impure
Safety Critical Parts
Real-Time Kernel
OS Processes

! Modularize

Abstraction
Operating Systems
High-Level Languages
Models, DSLs

Viewpoints
! Formalization
Configuration Files
4+1 Model
Blackbox/Whitebox
Types/Instances/Deployment

! Formalization

Protocols

Transctions
Locking/Synchronization
Resource Access

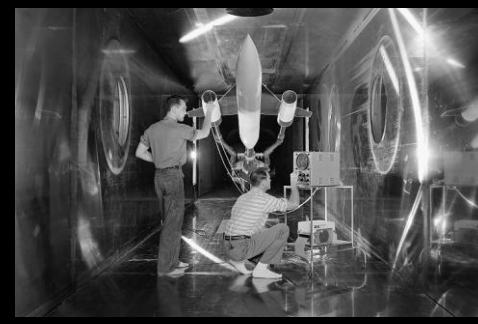
Make Explicit

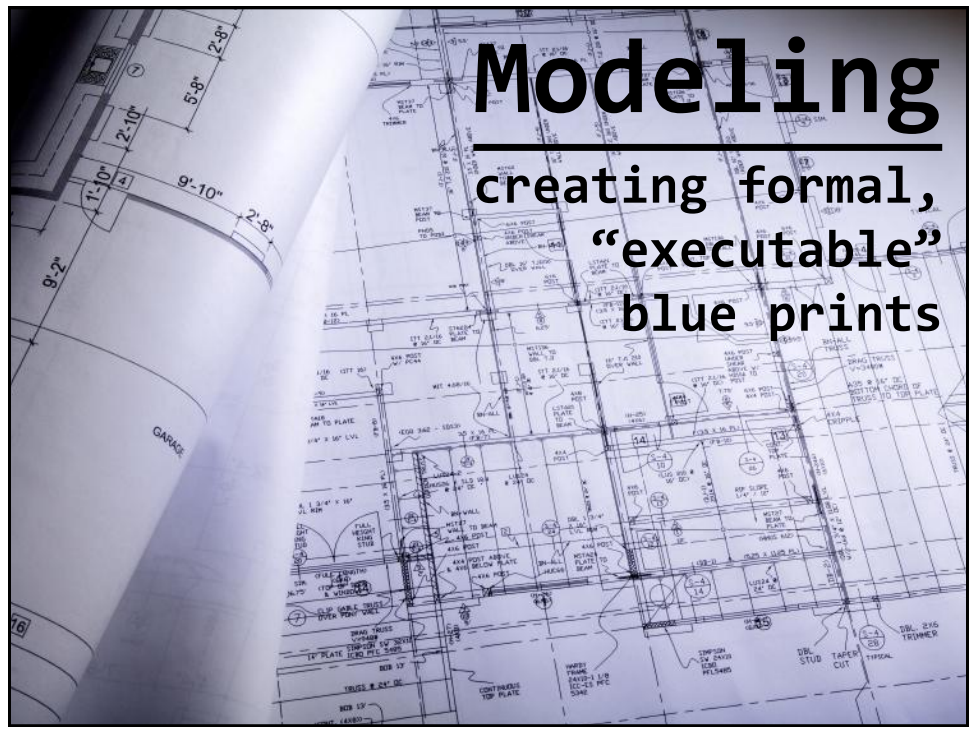
Dependencies
SOA, Messaging
Functional Programming
PLE Variabilities
Persistence: Loading Data

Software Architecture
DSL: expressiveness
MDSD: skeletons
Scade/SystemC

Limit Freedom

What is Architecting?





Patterns

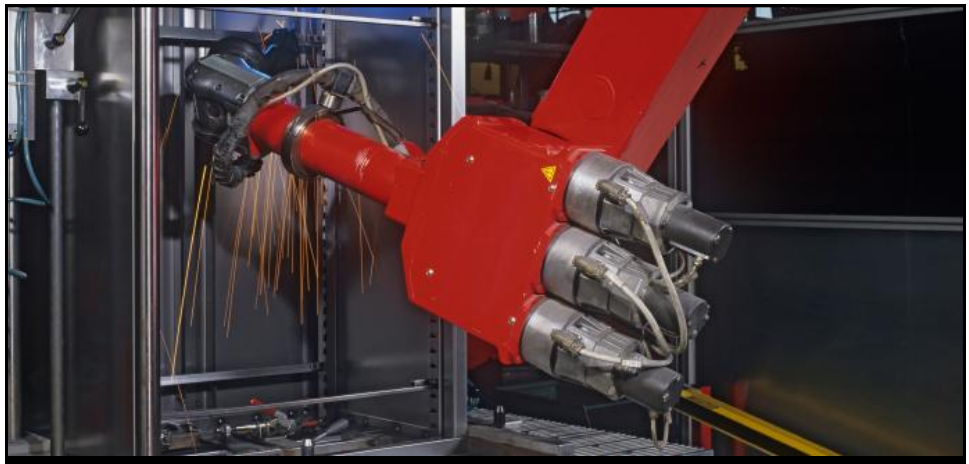
capturing and exploiting best practices

Documentation

explaining (how to use) an architecture to stakeholders

Technology

evaluating and deciding about implementation technologies



Automation

automate construction of soft-ware based on an architecture

Validation

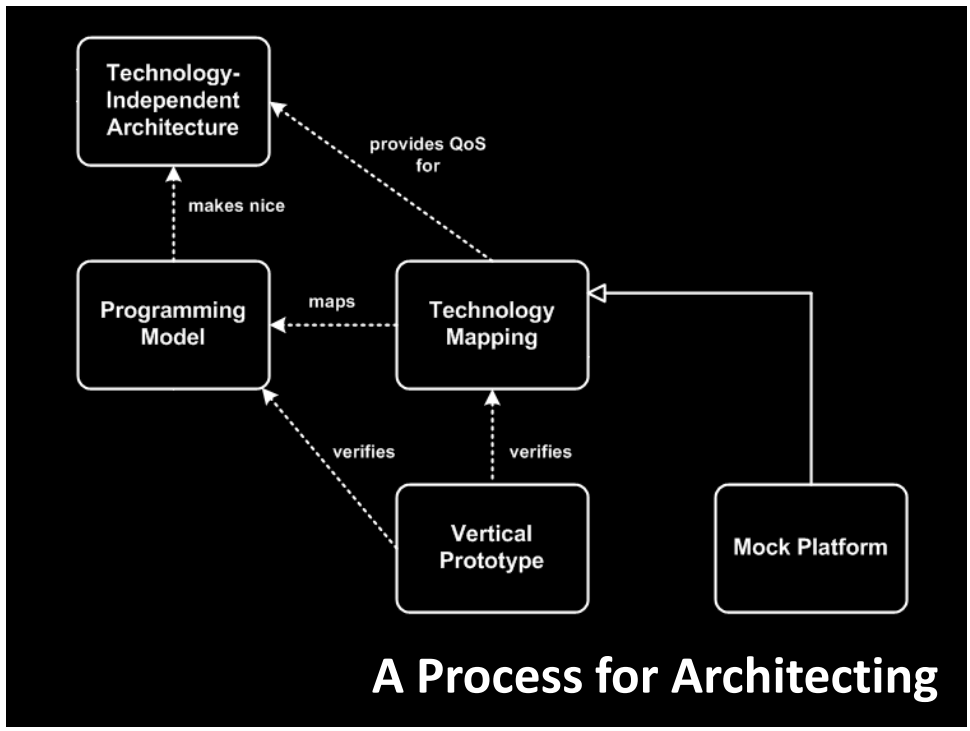
ensure that a system
conforms to an architecture

Reuse

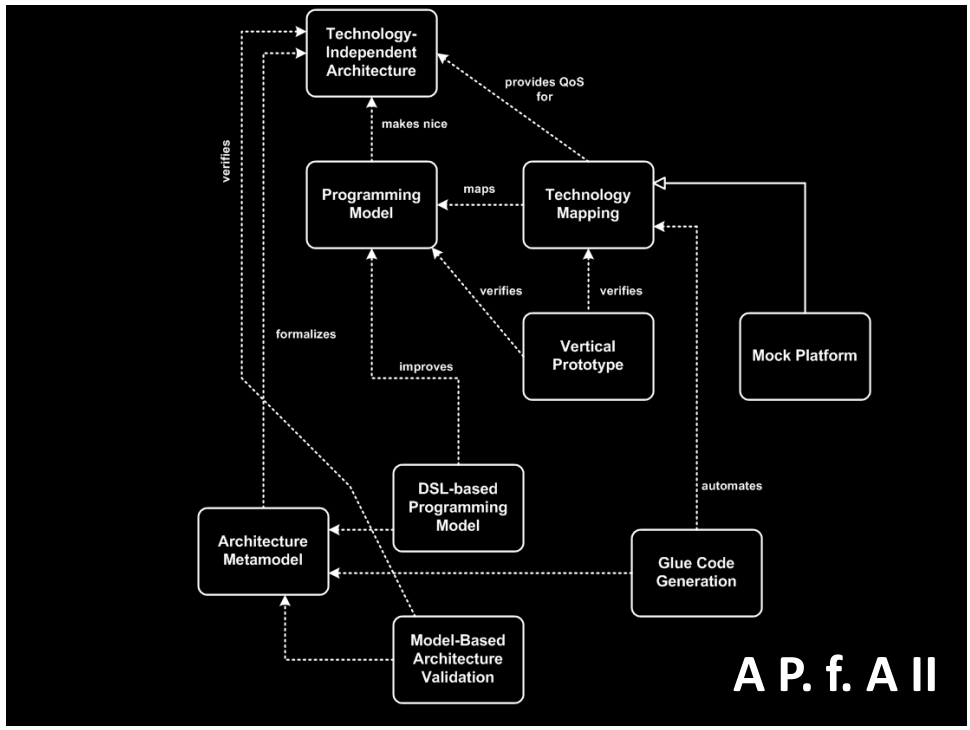
avoid wasting design
and implementation effort

Product Lines

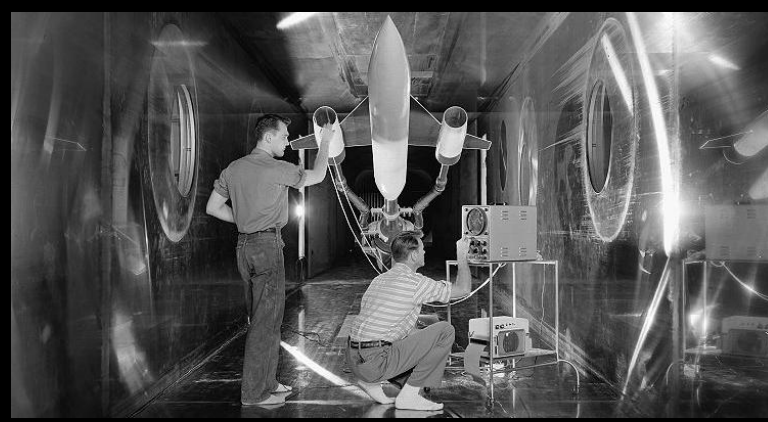
efficiently manage the variability within a family of products



A Process for Architecting



THE END.



itemis

.coordinates

web www.voelter.de
email voelter@acm.org
twitter markusvoelter

xing http://www.xing.com/profile/Markus_Voelter
linkedin http://www.linkedin.com/pub/0/377/a31