
Software
Architectures
for Enterprises
SA4E
Peter Sturm
Universität Trier
Winter 2024

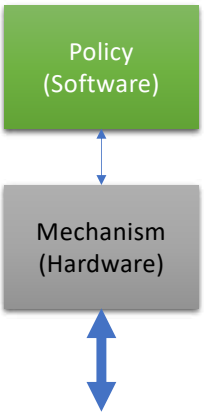

6. Architecting
2

Plumbing & Porcelain

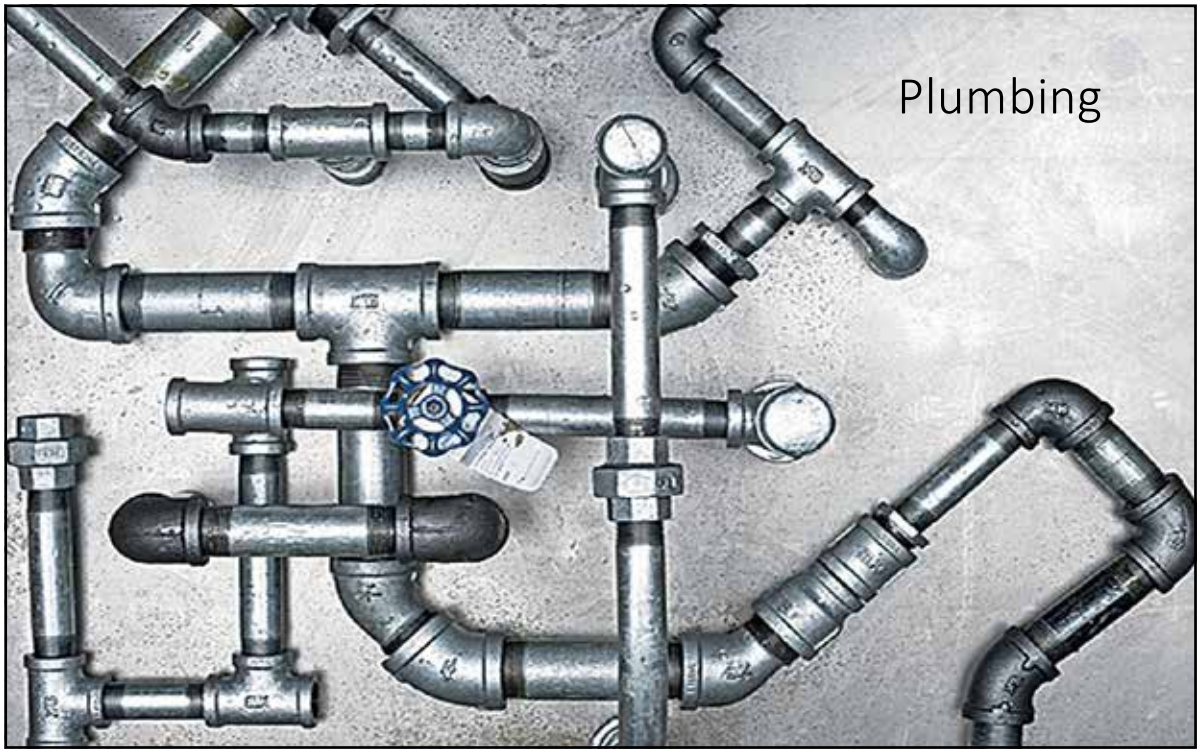
Plumbing & Porcelain

Policy & Mechanism

- Wiederkehrendes Motiv
- Software-Defined *
  - * = Network
  - * = Storage
  - Usw
- Interface & Implementierung
  - Encapsulation



Realität



Customer Review

Wario

★★★★☆ **Schweizer Ingenieur im Wenger gefunden**

Reviewed in Germany on 27 February 2012

Grundsätzlich bin ich mit den Funktionen des Wenger Giant sehr zu zufrieden. Allerdings scheinen mir die Produktionsstandards etwas mangelhaft zu sein. So habe ich zwischen den Funktionen #721 (Abrissbirne) und #722 (Skisprungschanze) zufällig einen Schweizer Ingenieur (Herr Ing. Meier) gefunden. Dieser ist anscheinend bei der Montage des Wenger Giant vergessen und eingeschlossen worden. Neulich beim Ausklappen der Abrissbirne (wir reissen mit dem Wenger derzeit eine alte Lagerhalle ab -> funktioniert 1a, das Wenger macht alles platt :)) schallte mir plötzlich ein "Grüezi" entgegen. Ich habe mit dem schwenkbaren Fluchtlichtmasten (#433) gleich mal den Zwischenraum ausgeleuchtet und tief unten den winkenden Herrn Ing. Meier entdeckt. Mithilfe des Lastenzugs (#1011) konnte ich Herrn Ing. Meier zwar problemlos bergen aber der gute Mann war doch sichtlich erschöpft und leicht unterkühlt. Also ab in die Infrarot-Kabine (#73) mit ihm und erstmal wieder aufwärmen. Nach einem leckeren Käsefondue (#973) und einem frischen Glas Wasser von der haus- bzw. messereigenen Bergquelle (#244) war er dann wieder bei Kräften. Die Geschichte ist zwar zum Glück nochmal glimpflich ausgegangen aber ärgerlich ist es trotzdem und sollte bei einem Qualitätsprodukt wie dem Wenger Giant eigentlich nicht passieren. Dafür gibt's von mir und Herrn Ing. Meier einen Punkt Abzug.

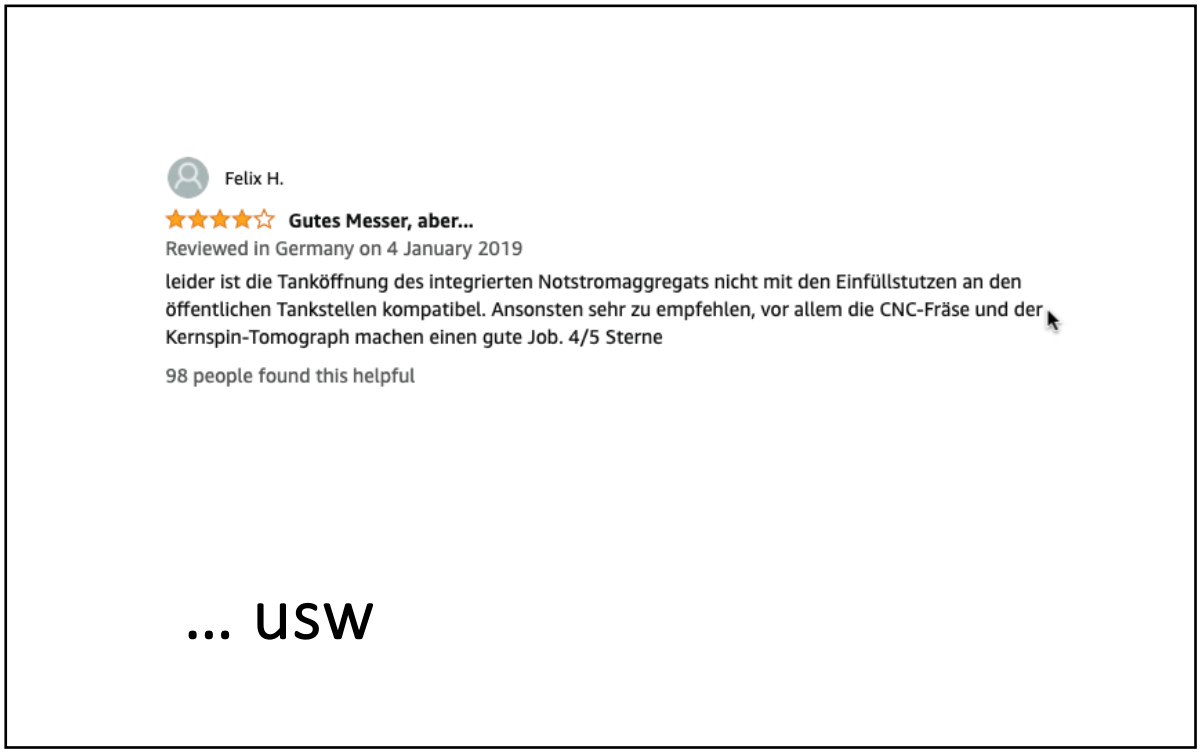
PS: Herr Ing. Meier und ich werden sicherheitshalber mit dem Radar (#768) nochmal alle Zwischenräume absuchen, nicht dass hier noch weitere Personen eingeschlossen sind. Ich möchte allen anderen Besitzern des Wenger Giant ans Herz legen, das gleiche zu tun. Denn eines muss einem klar sein, mit einem mächtigen Werkzeug wie dem Wenger Giant geht auch viel Verantwortung einher!

Gesendet mit meinem Wenger

8,026 people found this helpful

Helpful | Report abuse | Permalink

Translate review to English



Felix H.

★★★★☆ **Gutes Messer, aber...**

Reviewed in Germany on 4 January 2019

leider ist die Tanköffnung des integrierten Notstromaggregats nicht mit den Einfüllstutzen an den öffentlichen Tankstellen kompatibel. Ansonsten sehr zu empfehlen, vor allem die CNC-Fräse und der Kernspin-Tomograph machen einen gute Job. 4/5 Sterne

98 people found this helpful

... usw

Porcelain

Architecting

Literatur

2002 2020

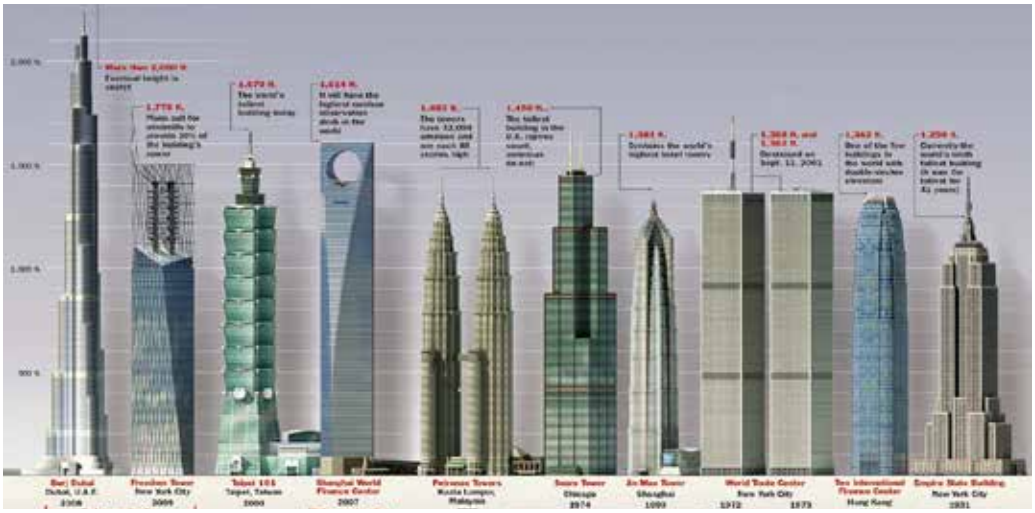
Decompose Existing

Create New

Studium fertig!

- Es bleibt stehen!
- Detailprobleme?

Scale!



Software Architecting

- Architecting is the profession and the know-how to achieve successful systems
- Dedicated role in a software project
  - Not anticipated by many companies yet
- Close similiarity with architecting of buildings
  - … but!

What's different?

- Building houses
  - Rooms with clear and limited interfaces
  - Window, Door, Electricity, Water, Heating
  - Computer Access!!!
- Building software
  - Functional requirements (Anything)
  - Non-Functional requirements (Anything)
  - Interdependencies (Anything)
  - Many approaches

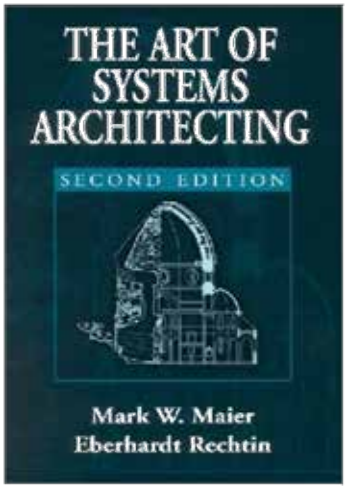
First Law of Software Architecture

Everything in Software Architecture is a tradeoff!

Aus “Fundamentals of Software Architecture”

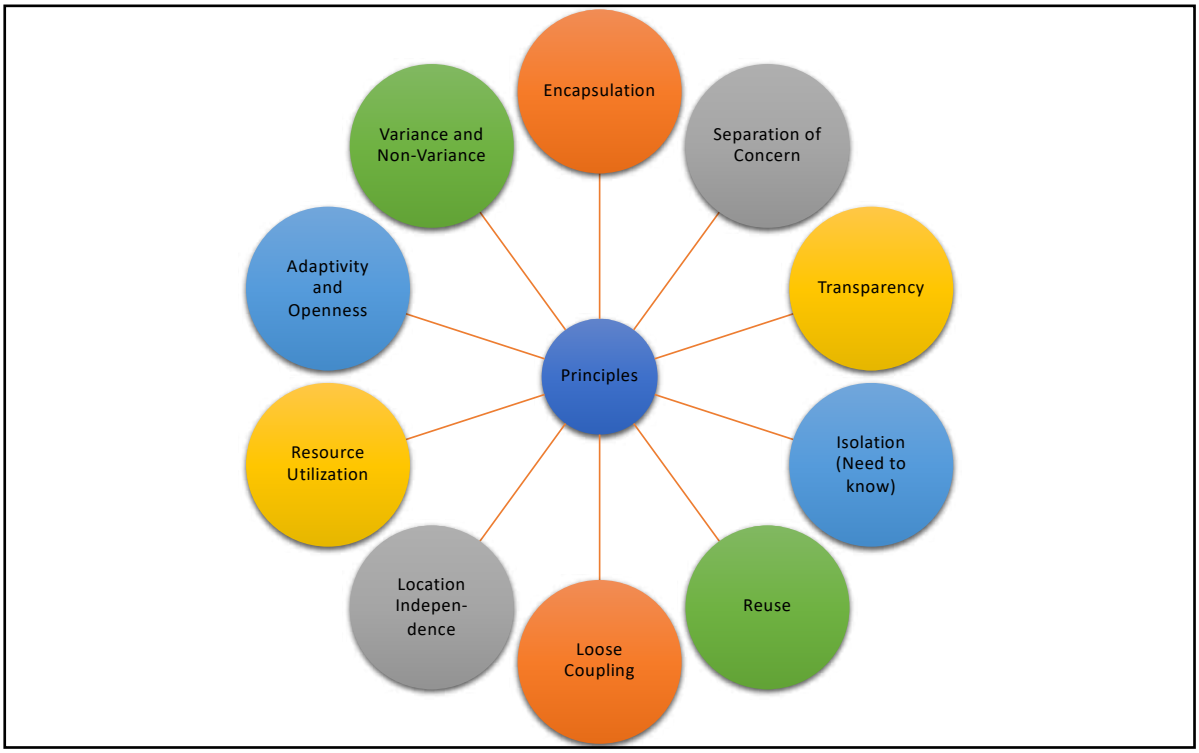
The Architect!

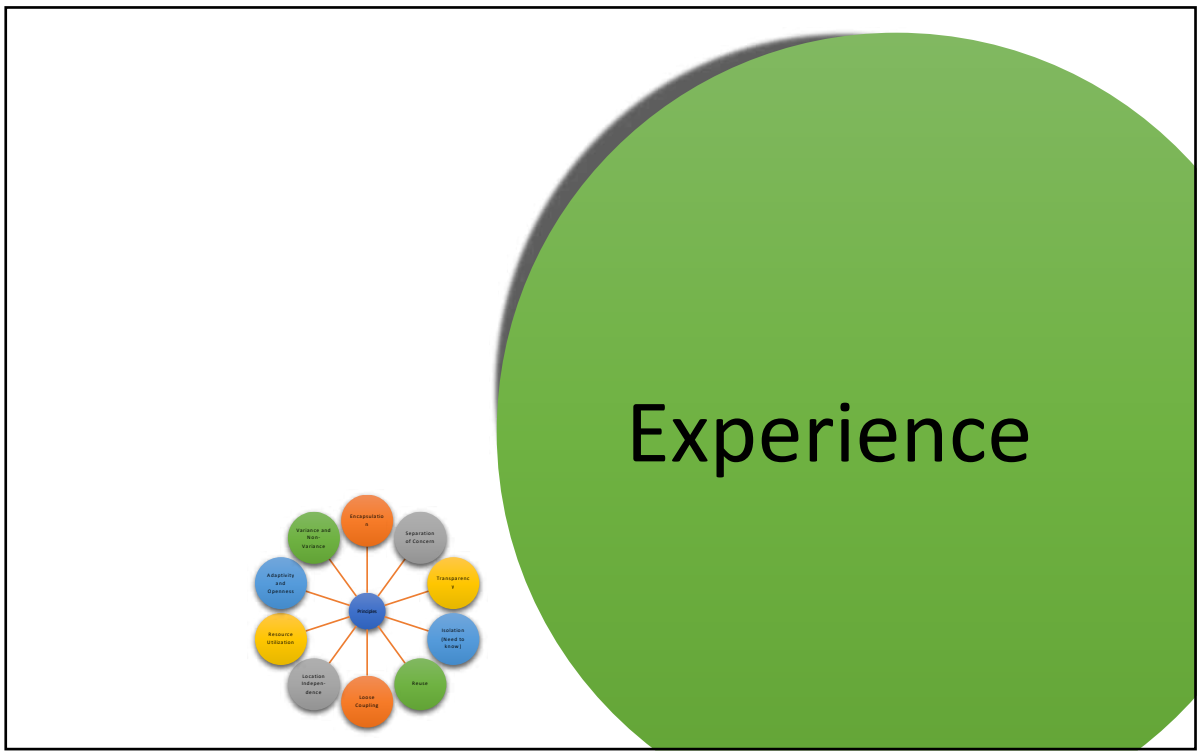
Reference

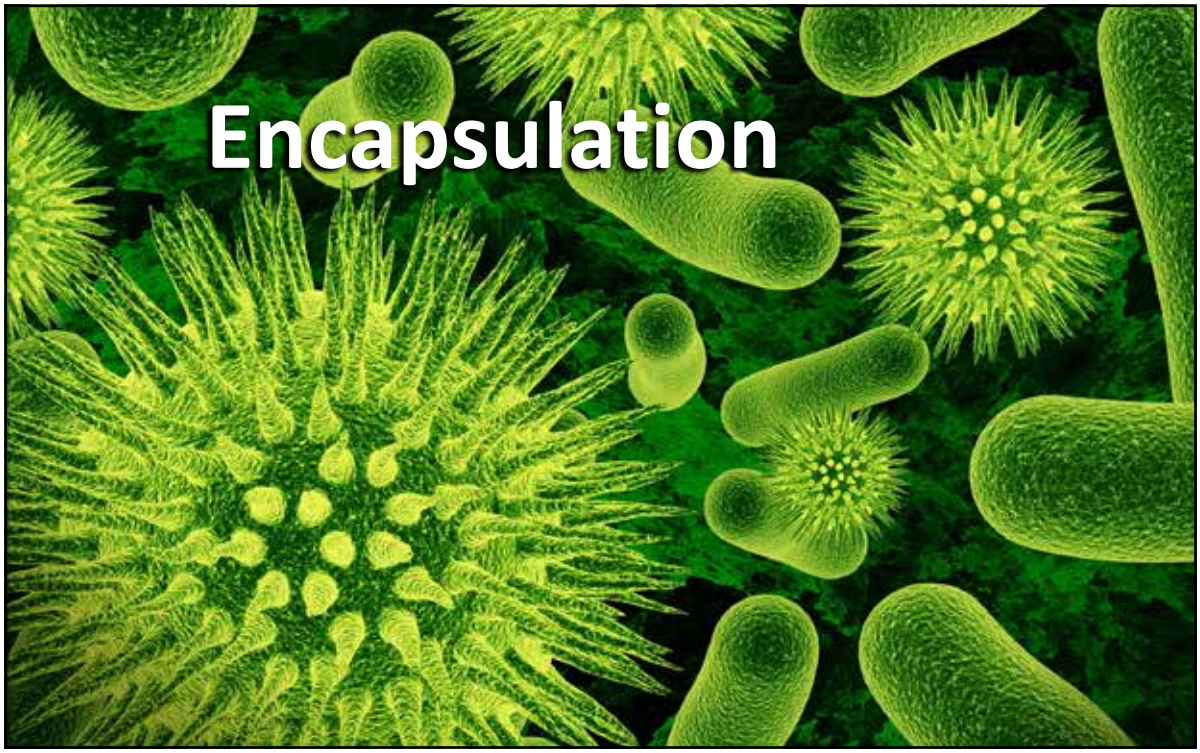

THE ART OF SYSTEMS ARCHITECTING
SECOND EDITION
Mark W. Maier
Eberhardt Rechtin


SUCCESS
IDEA
PRODUCTION
PRODUCT
Result
PROCESS
TEAM
MEDIA
COMMUNITY
GROUP
IDEA

Architectural Principles




Experience
Encapsulation
Separation of Concern
Transparency
Isolation (Need to know)
Reuse
Loose Coupling
Location Indepen-dence
Resource Utilization
Adaptivity and Openness
Variance and Non-Variance
Principles


Encapsulation

Encapsulation

- Software is divided into independent modules
  - internal black box structure
  - communication only via a publicly available interface
- Examples: components, objects, services

Objectives

- Independence
  - Internal modification of a specific module does not affect other modules
- Flexibility
  - Reconfiguration of the process flow does not require internal module modifications
- Manageability
  - Development of the application can be divided into meaningful and digestible pieces
- Reuse
  - Ready modules can be used in different configurations
  - Productized modules are especially cost-efficient

Challenge

- Granularity level
  - Large, complex modules are hard to divide into workstreams and can quickly become too specific for reuse. Very small modules do not constitute reasonable work entities and have little capacity to encapsulate reusable functions

Challenge

- Business alignment
  - Defined application landscape has to be optimally covered with possibly cost-efficient, productized modules

Challenge

- Maintenance of interface stability
  - During development of the module additional features expanding the interface cannot interfere with the basic ones

Separation

of Concerns

Separation of Concerns

- The process of breaking a computer program into distinct features that overlap in functionality as little as possible

Details

- SoC splits a large problem into easily manageable pieces or sub-problems whose separate solutions can be combined to solve the entire problem
- SoC can mean multiple things in different programming paradigms
  - Objects (Encapsulation)
  - Aspects, Annotations, Attributes, Procedures, ...

Objectives

- Loose coupling between concerns
  - Changes in one system functionality does not have an adverse cascading effect on other functions
- Reuse
  - Ready modules can be used in different configurations. Productized modules are especially cost-efficient and reduce the overall length of code
- Portability
  - Enhanced flexibility with concerns being portable across platforms

Loose Coupling

- Loose coupling is drawn from the principle of separation of concerns where communication between systems is a separate concern
- Coupling implies the definition of two parameters
  - The relationship between two systems (defining what needs to be shared)
  - Exchange relationship i.e. how will it be shared

Coupling Degree

- The degree of loose coupling can be measured by noting the number of changes (addition/removal) in data elements that could occur in the sending or receiving systems and determining if the computers would still continue communicating correctly

Virtualization

- Virtualization is the key enabler of loose coupling between legacy systems and modern data centric applications

Transparency

Transparency

- Transparency means that a complex system should hide some of its complex nature from its users

Variations

- Network Transparency
- Syntactic Transparency
- Semantic Transparency
- Name Transparency
- Location Transparency
- Performance Transparency
- Failure Transparency
- Execution Transparency