THM | CAMPUS GIESSEN | MNI
Mathematik, Naturwissenschaften und Informatik
TECHNISCHE HOCHSCHULE MITTELHESSEN

Software Architektur & Anwendungsentwicklung

Wintersemester 2021/22

Prof. Dr. Uwe Meyer

Organisatorisches

- Beide Termine finden als SMU statt. Wir werden also nicht zwischen Vorlesung und Übung/Praktika unterscheiden.
- Beide Termine finden als online PILOS-Konferenz statt.

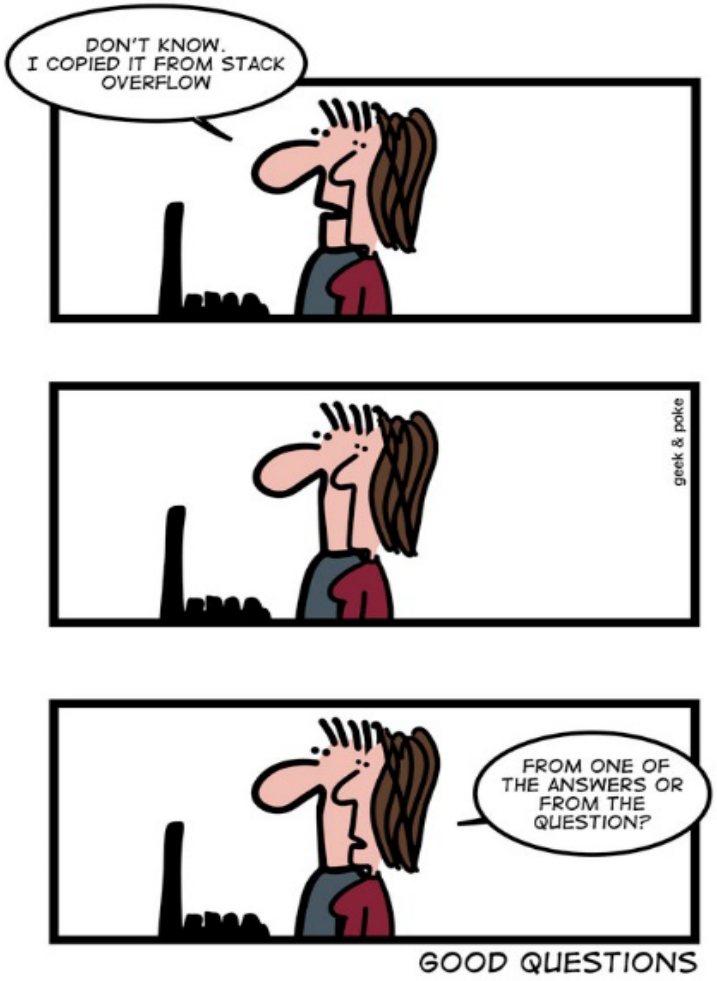
Prüfung

- Jede Gruppe (à 3 Studierende) bereitet jeweils eine halbe Doppelstunde (45 Minuten) zu vorher angekündigten Themen vor:

  Vortrag, Einbeziehung Ihrer Kommilitonen, Material (z.B. Folien)
- Mündliche Prüfung

- Erwartet wird Mitarbeit in der Veranstaltung und Bearbeitung der Übungs- / Hausaufgaben

Selbstständiges Arbeiten



Inhalte und Ziele (aus Modulhandbuch)

- Die Architektur von Softwaresystemen spielt eine entscheidende Rolle für die Anwendungsentwicklung. In der Architektur bereits enthaltene Entscheidungen bestimmen den Entwurfsprozess und sind Grundlage für die Erfüllung qualitativer Anforderungen.
- Wie kann man Softwarearchitektur entwickeln und dokumentieren? Qualitäts- und risiko-getriebene Entwicklung, Dokumentation u.a. mit Fundamental Modeling Concepts (FMC)
- Evaluierung von Softwarearchitekturen z.B. mit ATAM
- Architektur-Stile: z.B. Datenfluss-Systeme, Kontrollfluss-Systeme, Ereignisbasierte Systeme, Virtuelle Maschinen, Datenzentrierte Systeme
- Architektur-Prinzipien
- Architektur- und Entwurfsmuster: z.B. Verteilte Systeme (Broker, Interceptor), Interaktive Systeme (Model-View-Controller, Presentation-Abstraction-Control, Chain of Responsibility), Adaptierbare Systeme (Microkernel, Reflection), Metalevel-Architekturen und domänenspezifische Sprachen
- Fallstudien: z.B. Webanwendungen, mobile Anwendungen, Software-Produktlinien, verteilte Systeme...
- Die Rolle des Software-Architekten
- Zusätzlich werden Exkurse zu Themen im Umfeld des Themas angeboten.

- Die Studierenden können Konzepte von Software-Architektur beschreiben und konkreten Beispielen zuordnen
- Sie können den Aufbau eines Anwendungssystems aus konzeptioneller Sicht verständlich beschreiben, erklären und visuell darstellen.
- Sie können die Prinzipien, auf deren Basis heute Software entwickelt wird, abgrenzen.
- Die Studierenden können Architektur- und Entwurfsmuster beschreiben und für den jeweiligen Anwendungsfall ein geeignetes Muster auswählen und anwenden.
- Sie können wenden diese Kenntnisse in einer Projektarbeit im Team selbstständig anwenden.
- Im Rahmen von Fallstudien können die Studierenden einen eigenen Standpunkt zu
- Architekturentscheidungen entwickeln und diesen in Diskussionen mit fundierten Argumenten überzeugend vertreten, ihn aber auch kritisch hinterfragen

Literatur

- Richards, M., Ford, N. (2021), Handbuch moderner Softwarearchitektur, Architekturstile, Patterns und Best Practices, O‘Reilly.
- Knöpfel, A.; Gröne, B.; Tabeling, P. (2005), Fundamental Modeling Concepts: Effective Communication of IT Systems, Wiley.
- Martin, R.C. (2018), Clean Architecture - Gute Softwarearchitekturen: Das Praxis-Handbuch für professionelles Softwaredesign. Regeln und Paradigmen für effiziente Softwarestrukturierung, mitp.

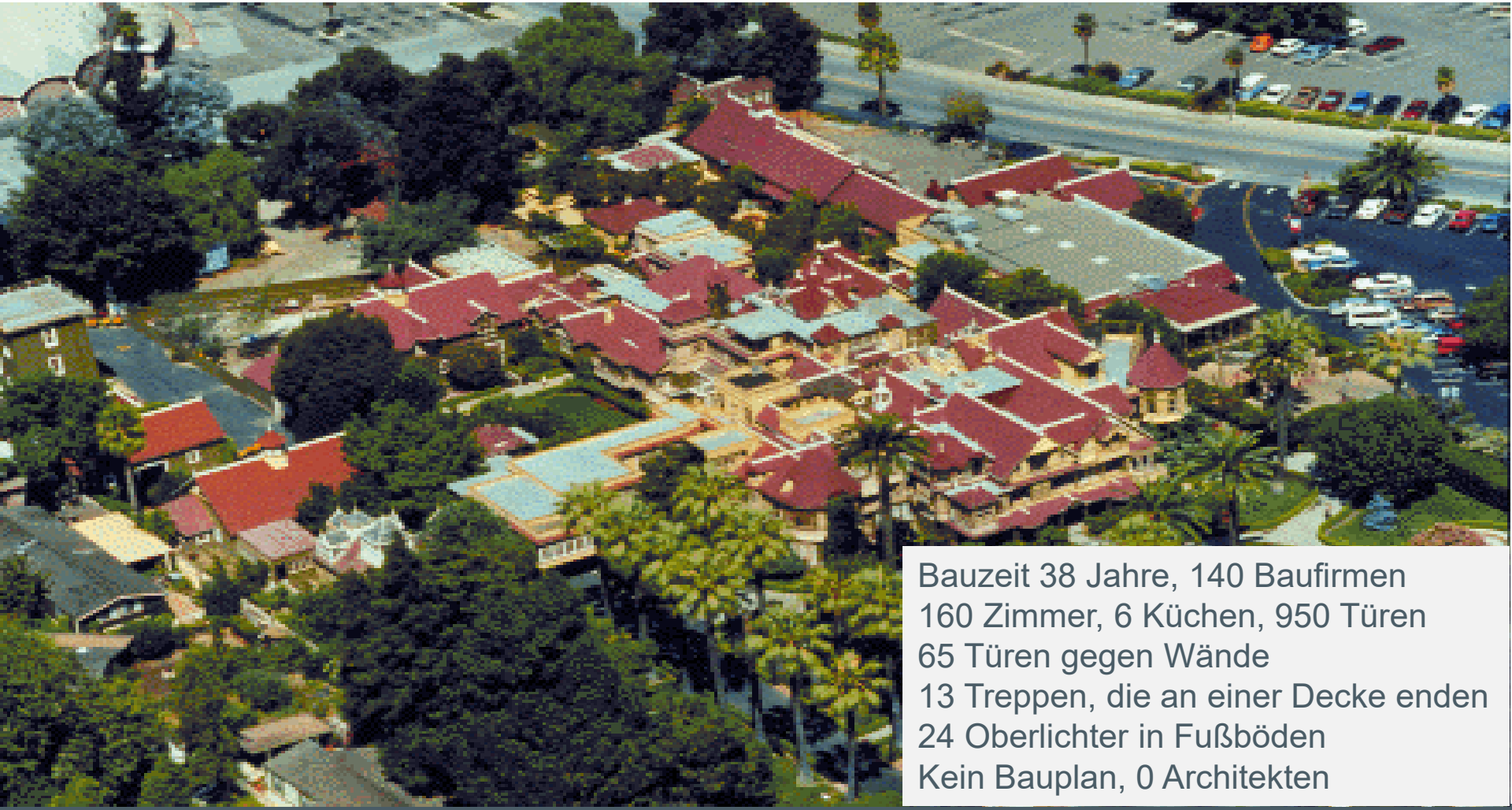
- ausgewählte Artikel

Überblick

1. Was ist Architektur?
2. Anforderungen & -ilities
3. Viewpoints…
4. Architektur Prozess
   a) Architecture Decision Records
   b) UML
   c) FMC
5. Architekturmuster (inkl. SOLID)
6. Architekturprinzipien
7. Architekturstile
8. Die Rolle des Architekten
9. Governance und Architekturevaluation

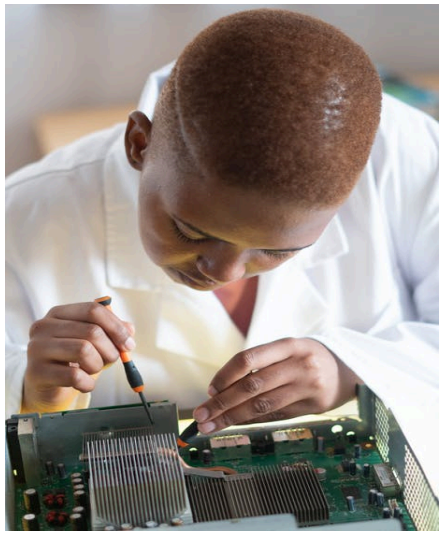
Wozu Architektur?

Winchester House, San Jose, CA



Bauzeit 38 Jahre, 140 Baufirmen
160 Zimmer, 6 Küchen, 950 Türen
65 Türen gegen Wände
13 Treppen, die an einer Decke enden
24 Oberlichter in Fußböden
Kein Bauplan, 0 Architekten

Warum ist Architektur wichtiger denn je?



Stärkerer Fokus auf Qualität
(s. Kapitel 3)

Immer größere und
komplexere Systeme

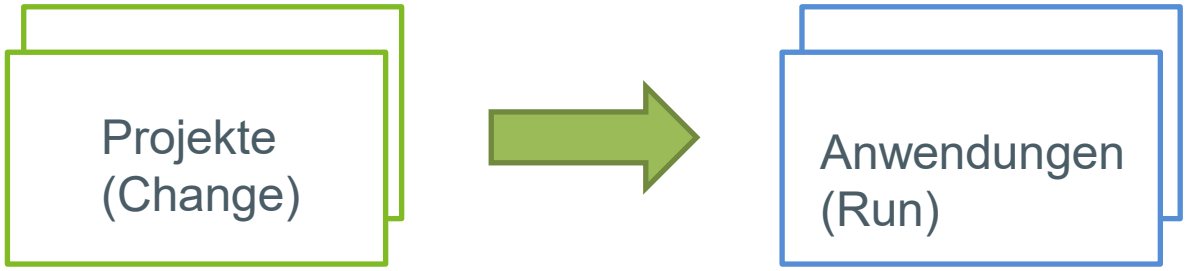
Hinweis: Paul Clement, Software Architecture Best Practices, SEI CMU
https://resources.sei.cmu.edu/asset_files/Presentation/2006_017_001_22556.pdf

Große und komplexe Systeme

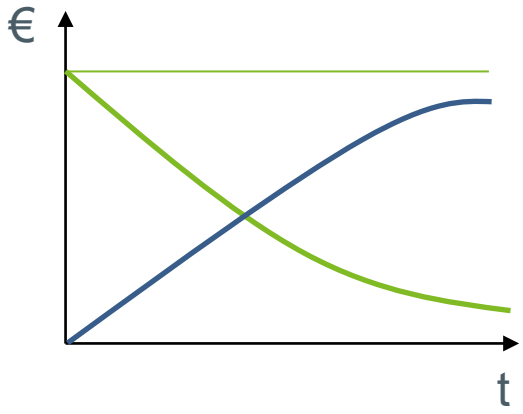
- Ein großes System wie z.B. Netflix kann nicht von einer Gruppe von Programmierern erschaffen werden, weil
  - Fokus auf Einzelteile statt auf das Ganze
  - Tiefes Wissen in wenigen Technologien
- Teams aus Business Analysts, Designern, Architekten, Entwicklern, Testern, Betrieb können einzelne Aufgaben besser

- Der Projektleiter muss das Team „leiten“
- Der Architekt muss das Team inhaltlich führen

Das IT-Dilemma



Führt zu …



Summe = konstant
Run the Business

Change the Business

Diskussion

Wie kommt man aus dem Dilemma heraus?

Und es kommt noch schlimmer …

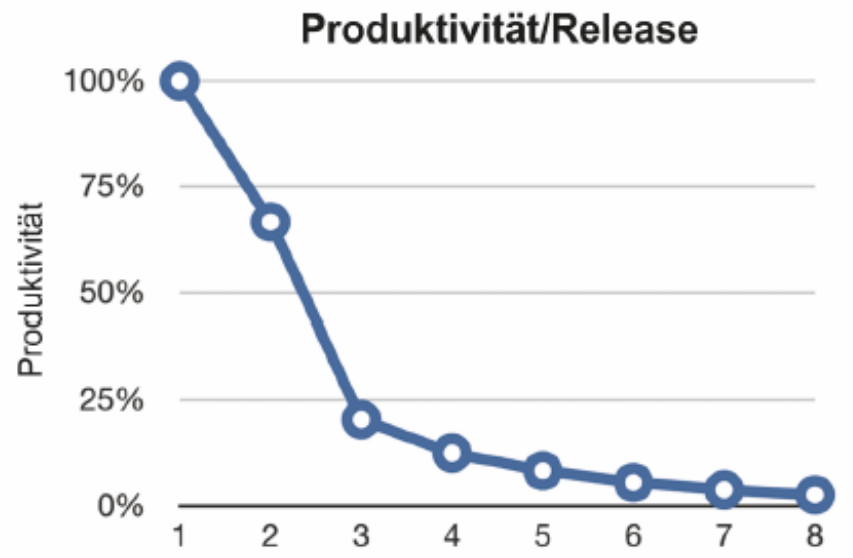


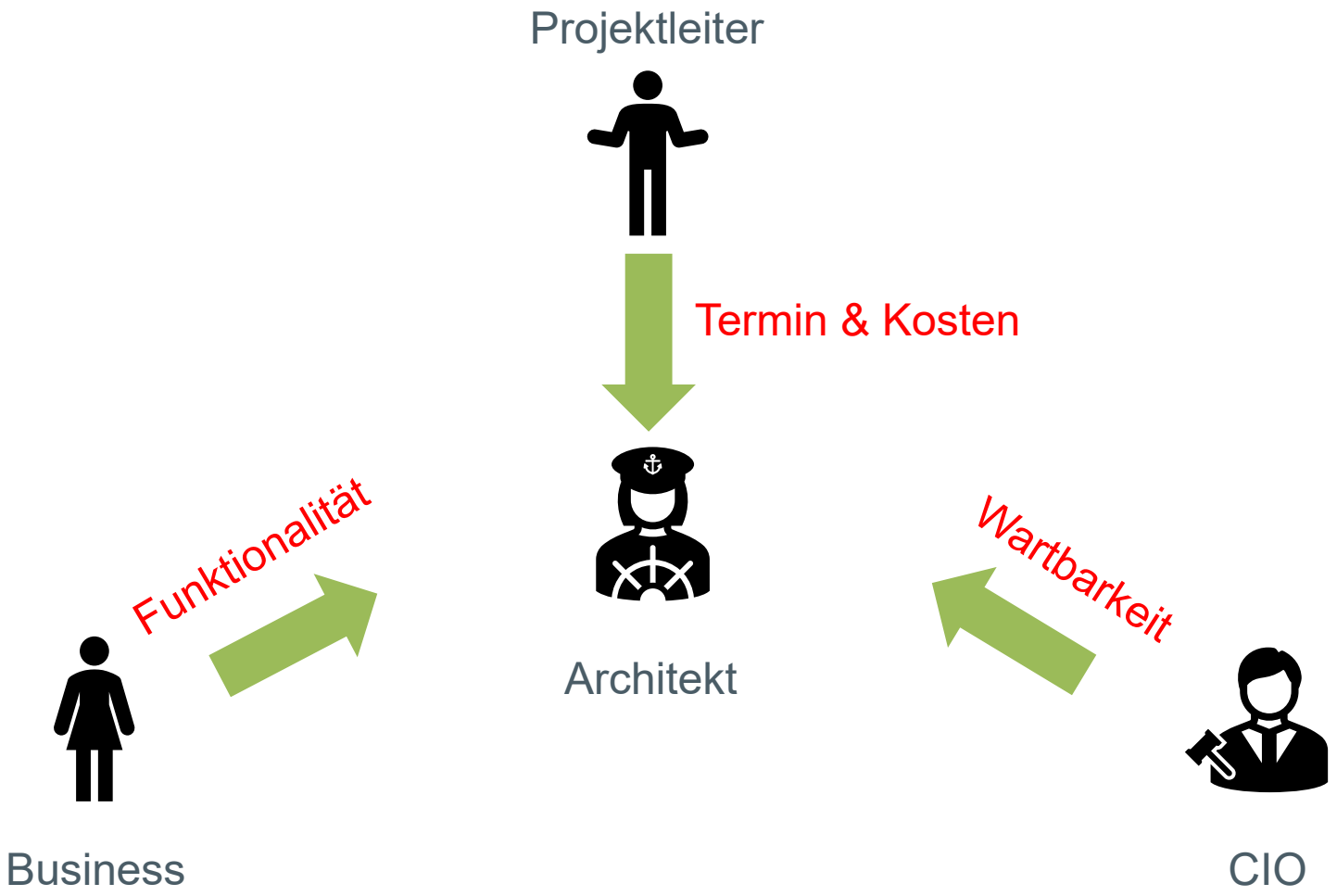
Abb. 1.4: Produktivität je Release

Im Laufe der Zeit sinkt die Produktivität und die Kosten pro Release steigen dramatisch.

Warum?

Quelle: Robert C. Martin, Clean Architecture

Das IT-Dilemma im Kleinen



Erste Definition (22 vor Christus)

Marcus Vitruvius Pollio war römischer Architekt und schrieb das Buch *De architectura libri decem,* in dem er drei Hauptanforderungen an Architektur beschreibt:

- *Firmitas* (Festigkeit),
- *Utilitas* (Nützlichkeit) und
- *Venustas* (Schönheit).

Darüber hinaus definiert Vitruv sechs Grundbegriffe des Faches Architektur: „ordinatio“, „dispositio“, „eurythmia“, „symmetria“, „decor“ und „distributio“.

Bitte finden Sie Beispiele für diese neun Begriffe in
a) Hausbau
b) Softwarearchitektur

Bildquelle: https://commons.wikimedia.org/wiki/File:Vitruvius,_Florence,_Plut._30.10.jpg

Erste Definition (22 vor Christus)

Marcus Vitruvius Pollio war römischer Architekt und schrieb das Buch *De architectura libri decem,* in dem er drei Hauptanforderungen an Architektur beschreibt:

- *Firmitas* (Festigkeit),
- *Utilitas* (Nützlichkeit) und
- *Venustas* (Schönheit).

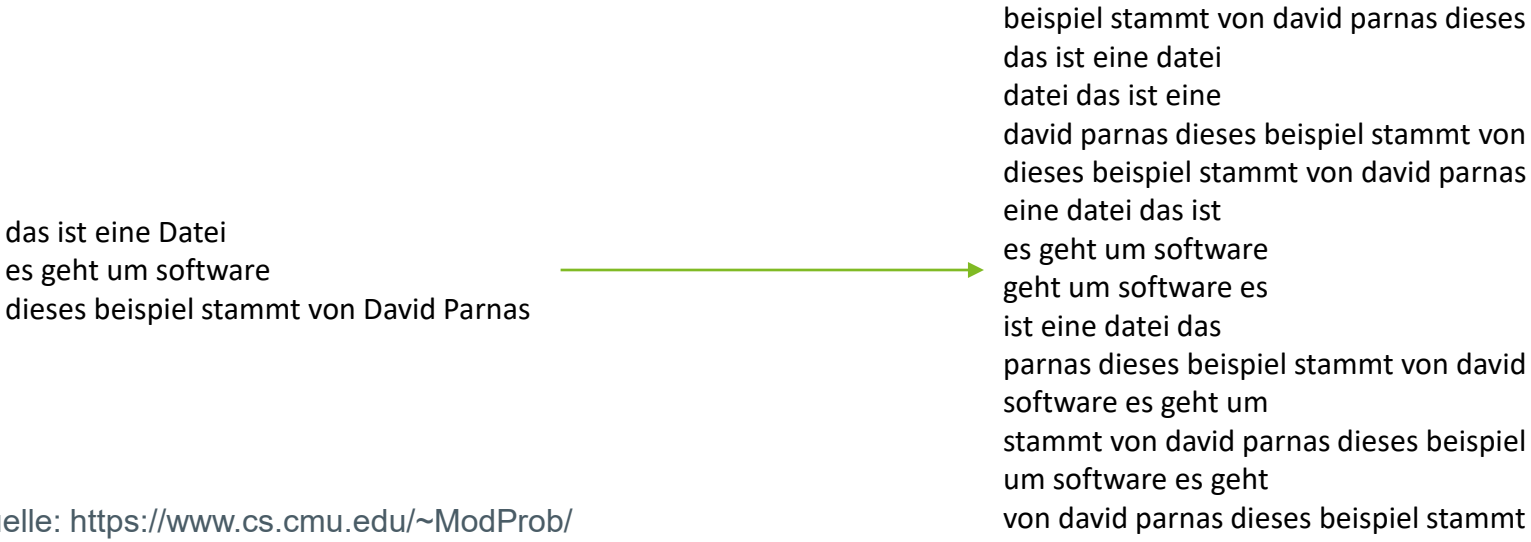
Frage: Sind diese drei immer gleichermaßen erreichbar? Gibt es **trade-offs**?

Bildquelle: https://commons.wikimedia.org/wiki/File:Vitruvius,_Florence,_Plut._30.10.jpg

Beispiel

Eines der 10 ‚Model Problems' der Software Architektur ist „Keywords in Context" (KWIC) von David Parnas. KWIC wurde früher zum Indizieren von Texten benutzt.

- Eine Datei besteht aus mehreren Zeilen
- Jede Zeile besteht aus mehreren Wörtern
- Jede Zeile wird so oft wie möglich rotiert (erstes Wort an das Ende der Zeile)
- Alle Ergebniszeilen werden alphabetisch sortiert.



Quelle: https://www.cs.cmu.edu/~ModProb/

Mögliche Architekturen

Frage: Wie würden Sie das umsetzen?

- Prozedural
  Gemeinsamer Speicher, der Schritt-für-Schritt manipuliert wird
- Funktional
  Eine Reihe von Funktionsaufrufen
- Objekt-orientiert
  Kapselung der Daten in ein Index-Objekt und Zugriff durch Methoden
- Pipe-and-Filter
  Die Erzeugung eines neues Datensatzes ist ein Ereignis, dass per Pipe an den nächsten Verarbeitungsschritt weitergeleitet wird

Hausaufgabe

- Implementieren Sie eine der Architekturen

- Wir werden Ihre Lösungen in der nächsten Stunde diskutieren.