



Vorlesung Integrierte Modellierung komplexer Systeme

Ganzheitliche Modellierung

Bachelor Studiengang Informatik
WS 2017-2018

Hugo Colceag | MHP

© 2018 MHP Management- und IT-Beratung GmbH

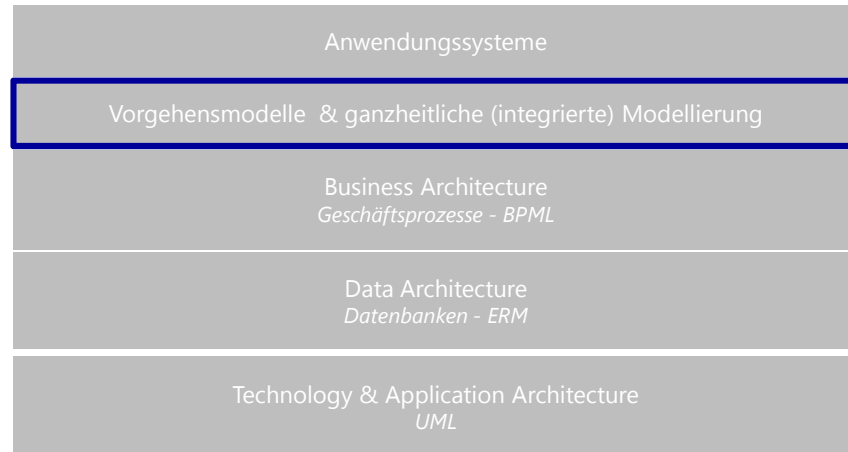


UNIVERSITATEA
BABEŞ-BOLYAI

Einleitung - Integrierte Modellierung komplexer Systeme



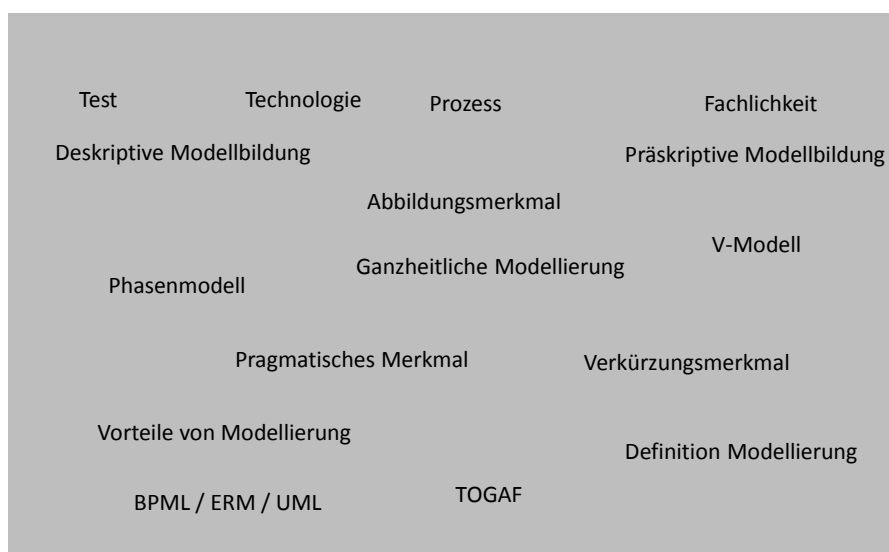
Vorlesungsinhalte und Aufbau



© 2018 MHP Management- und IT-Beratung GmbH

3

Mindmap - Ganzheitliche Modellierung



© 2018 MHP Management- und IT-Beratung GmbH

4

Agenda



- 1 Grundlagen der Modellierung
- 2 Einleitung Vorgehensweise ganzheitliche Modellierung

Grundlagen der Modellierung



Motivation - Vorteile von Modellierung in Projekten

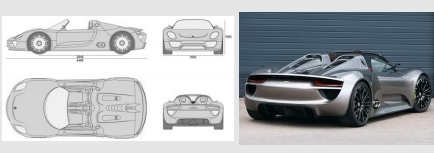
- ✓ Verständnis der fachlichen Anforderungen, der technischen Umgebung und der Randbedingungen der zu erstellenden Lösung
- ✓ Verbesserung der Kommunikation durch einfache und präzise Dokumentation, die zu jedem Zeitpunkt auf dem aktuellsten Stand ist
- ✓ Schaffung eines Überblicks und Steigerung der Effizienz und Qualität im Entwicklungsprozess
- ✓ Standardisierte Dokumentation
- ✓ Langfristige Kostenersparnis

Grundlagen der Modellierung

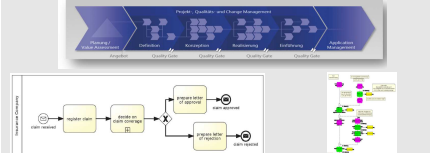
Definition Modellierung

- Modellierung ist die **vereinfachte Abbildung der Realität** mit dem Ziel eine **spezielle Eigenschaft** genauer darzustellen.

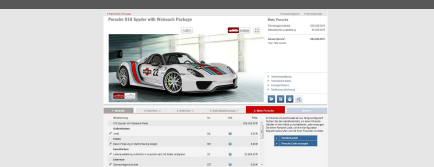
Alltägliche Modelle



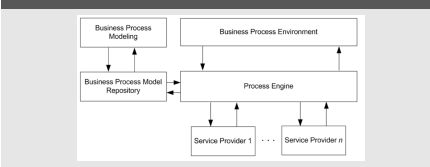
Geschäftsprozessmodelle



Modelle im Web



Architekturmodelle



© 2018 MHP Management- und IT-Beratung GmbH

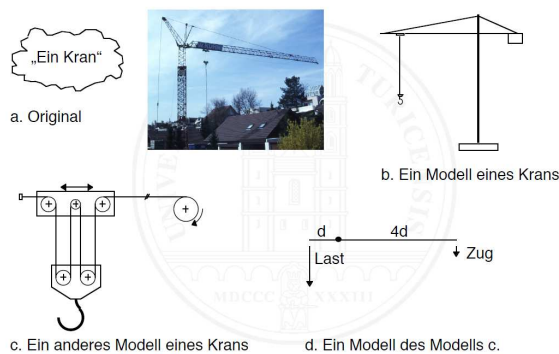
7

Grundlagen der Modellierung

Hauptmerkmale des Modellbegriffs nach H. Stachowiak [1973]

Abbildungsmerkmal

- „Modelle sind stets Modelle von etwas, nämlich Abbildungen, Repräsentationen natürlicher oder künstlicher Originale, die selbst wieder Modelle sein können.“
- Jedes Modell ist Abbild oder Vorbild



Quelle: M. Glinz 2005

© 2018 MHP Management- und IT-Beratung GmbH

8

Hauptmerkmale des Modellbegriffs nach H. Stachowiak [1973]

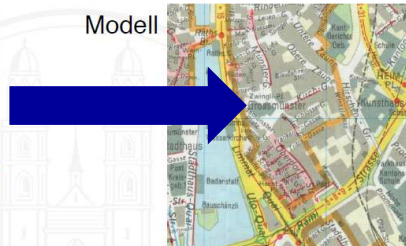
Verkürzungsmerkmal

- „Modelle erfassen im allgemeinen nicht alle Attribute des durch sie repräsentierten Originals, sondern nur solche, die den jeweiligen Modellerschaffern und/oder Modellbenutzern relevant erscheinen.“
- Jedes Modell abstrahiert

Original



Modell



Quelle: M. Glinz 2005

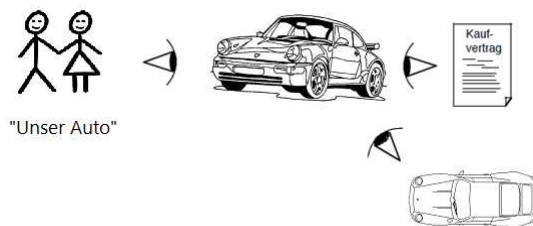
© 2018 MHP Management- und IT-Beratung GmbH

9

Hauptmerkmale des Modellbegriffs nach H. Stachowiak [1973]

Pragmatisches Merkmal

- „Modelle sind ihren Originalen nicht per se eindeutig zugeordnet. Sie erfüllen ihre Ersetzungsfunktion
 - Für bestimmte – erkennende und/oder handelnde, modellbenutzende – Subjekte
 - Innerhalb bestimmter Zeitintervalle und
 - Unter Einschränkung auf bestimmte gedankliche und tatsächliche Operationen.“
- Jedes Modell wird im Hinblick auf einen Verwendungszweck geschaffen



Quelle: M. Glinz 2005

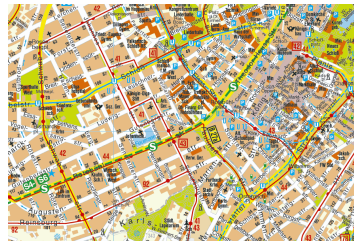
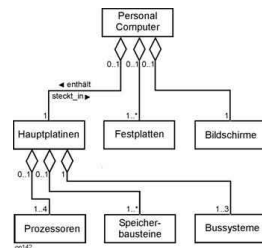
© 2018 MHP Management- und IT-Beratung GmbH

10

Grundlagen der Modellierung

Deskriptive Modellbildung

- Modellierung eines existierenden Originals
- Modellierung eines zukünftigen, aber nicht gestaltbaren Originals
- muss sich streng an der Realität orientieren

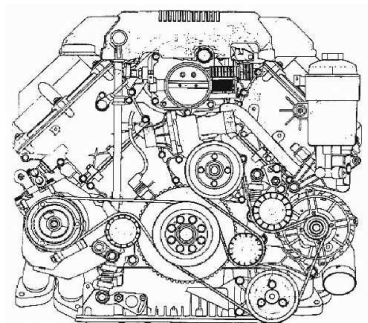


Beispiele: Stadtplan, Wettervorhersage, Komponentenstruktur eines im Einsatz befindlichen Informatiksystems

Grundlagen der Modellierung

Präskriptive Modellbildung

- Modellierung eines zu schaffenden, gestaltbaren Originals
- darf zukünftige Realität gestalten



NVH-Merkmale Porsche Cayenne V8 Motor

Ölwanne als tragendes, dynamisch versteifendes Element	Abgasanlage dynamisch abgekoppelt
Oberflächen ohne dominierende Eigenfrequenzen	Motorfeste Sekundärluftpumpen
Harmonischer Steifigkeitsverlauf	Integrierte Motorenlagernahmen
Kurbelgehäuseoberteil in Closed Deck Ausführung	Abgekoppeltes und strömungsgeräuschoptimiertes Saugmodul
Akustisch optimierter Steuertrieb mit Steuerkastendeckel	Schwingungstechnisch optimierte Nebenaggregateanordnung
Sehr steifer Kurbeltrieb mit Kurbelgehäuseventil in Leiterrahmenbauweise	

Beispiele: Konstruktionszeichnung, Anforderungsspezifikation für zu entwickelnde Teile

1 Prinzipien der SW-Technik

Einführung Prinzipien



Was ist ein Prinzip?

Nach welchen Prinzipien handeln Sie?

Welche Prinzipien fallen Ihnen zur Softwaretechnik ein?



Man sollte nicht mit dem Kopf durch die Wand rennen!



Man sollte nicht den Ast absägen, auf dem man sitzt!

Prinzipien der SW-Technik:

- Prinzip der Abstraktion
- Prinzip der Bindung und Kopplung
- Prinzip der Hierarchisierung
- Prinzip der Modularisierung
- Geheimnisprinzip



Prinzipien

- sind Grundsätze, die man seinem Handeln zugrunde legt,
- sind allgemeingültig, abstrakt, allgemeinsten Art,
- bilden eine theoretische Grundlage,
- werden aus Erfahrung gewonnen.

Prinzipien der SW-Technik



Motivation



Warum Prinzipien für die SW-Technik?



Alle Prinzipien der Softwaretechnik haben das Ziel, **Komplexität** zu **reduzieren** und **beherrschbar** zu machen.



Welches Ergebnis erhält man durch die Anwendung der Prinzipien?



Als Ergebnis der Anwendung von Prinzipien entstehen **einfache und verständliche Systemdokumentationen und Entwürfe**, sowie **wartungs- und änderungsfreundliche Systeme**.

Agenda



1 Prinzip der Abstraktion

Bekanntes Ölgemälde von René Magritte (1898–1967)



© 2018 MHP Management- und IT-Beratung GmbH

17

Was bedeutet Abstraktion

- Oft spricht man anstelle von Abstraktion auch von Modellbildung.
- Durch Abstrahieren vom Konkreten erstellt man ein Modell der realen Welt.



Ein Modell ist die Abstraktion von Konkretem.

- Das Gegenteil von Abstrahieren ist Konkretisieren.
- Unter Abstrahieren versteht man
 - das Herausheben des Wesentlichen,
 - das Beiseite lassen von Unwesentlichem,
 - das Erkennen gleicher Merkmale.



Abstraktion bedeutet immer Konzentration auf das Wesentliche.

© 2018 MHP Management- und IT-Beratung GmbH

18

Die Bedeutung der Abstraktion für das SW-Engineering

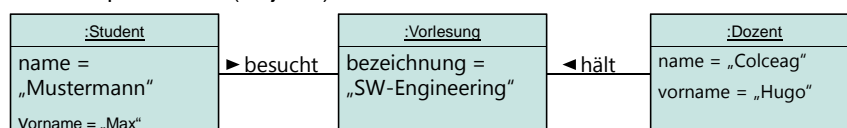
- Kennen Sie die Aussagen:
 - „Das ist mir zu abstrakt!“
 - „Das ist mir zu abgehoben!“.
- Beim SW-Engineering geht es darum, die geeigneten Abstraktionen zu finden (für Kunden, für Vorgesetzte, für SW-Entwickler, etc.)
- Bei der Objektorientierten Modellierung werden Modelle bzw. Sichten auf das zu realisierende System erstellt. Es muss also ständig geeignet abstrahiert werden.
 - Abstraktion zur Abgrenzung des Problembereiches.
 - Abstraktion zum Finden der Objekte.
 - Die Klasse ist eine Abstraktion (ein Modell) von konkreten Objekten.
 - Abstraktion zum Erkennen der Datenfelder und Methoden.
 - ...



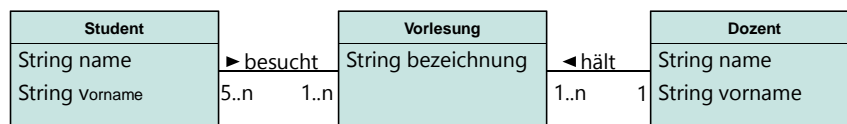
„Abstrahieren ist eine äußerst anspruchsvolle Tätigkeit, denn es ist meist sehr schwierig, geeignete Abstraktionen zu finden und aus vielen konkreten Tatsachen das Wesentliche zu isolieren.“ [Balzert]

Abstraktionsebenen – Exemplare und Typen

■ Exemplar-Ebene (Objekte)



■ Typ-Ebene (Klassen)



Exemplar-Ebene entspricht in der Objektorientierung dem Objekt-Diagramm.
 Typ-Ebene entspricht in der Objektorientierung dem Klassendiagramm.

1 Prinzip der Bindung und Kopplung

Low Coupling and High Cohesion im Alltag



Warum ordnen wir die Dinge wie dargestellt in die Räume ein?



Kinderzimmer

Arbeitszimmer

Garderobe

Küche



Besonders effizient sind wir beim Erledigen der täglichen Aufgaben, wenn die Gegenstände, die zum Erledigen einer Aufgabe benötigt werden, nach dem Prinzip Low Coupling und High Cohesion angeordnet sind.

- Low Coupling: Ich muss möglichst wenig zwischen den Räumen umherlaufen, um eine Aufgabe (kochen, Brief schreiben, anziehen) zu erledigen.
- High Cohesion: Gleichartiges wird in einem Zimmer untergebracht (leicht aufzufinden).

Bindung und Kopplung in der SW-Technik

- Bindung (cohesion)
 - Qualitatives Maß für die Kompaktheit einer Komponente.
 - Es wird geprüft, ob die Bestandteile einer Komponente ein hohes Maß an Zusammengehörigkeit aufweisen und wie viele Aufgaben in der Komponente erledigt werden.
 - **High Cohesion** (stark gebundenes System): Jede Komponente ist für eine definierte Aufgabe verantwortlich.
- Kopplung (coupling)
 - Gegenstück zur Bindung.
 - Qualitatives Maß für die Schnittstellen zwischen den Komponenten.
 - Der Kopplungsgrad bestimmt, wie einfach oder schwierig es ist, Änderungen an einem System vorzunehmen.
 - **Low Coupling** (schwach gekoppeltes System): Änderungen beschränken sich optimalerweise nur auf eine Komponente.



Optimierung erfolgt immer nach Low Coupling und High Cohesion.

Praktische Beispiele für Low Coupling und High Cohesion

- Client-/Server-System
 - Low Coupling: möglichst wenig Kommunikation zwischen Client und Server.
 - High Cohesion: Berechnungsalgorithmen, die kontinuierlich auf Daten zugreifen, werden auf dem Server implementiert.
- Schichtenmodelle für Informationssysteme
 - High Cohesion: Zusammengehöriges wird in eine Schicht gepackt.
 - Schicht für die grafische Bedienoberfläche, Separierung der Business Logik in einer Schicht, Schicht für den Zugriff auf die Daten.
 - Grund: Die Business Logik ist sehr stabil, während sich die Technologie für die grafische Bedienoberfläche und Datenhaltung häufig ändert.
- Klasse
 - High Cohesion nach Innen: Daten und Methoden, die auf den Daten arbeiten, werden in einer Einheit untergebracht.
 - Low Coupling nach außen: Kommunikation zwischen den Klassen wird minimiert.



Low Coupling und High Cohesion bedingen sich meistens gegenseitig.

1 Prinzip der Hierarchisierung

Ein weiteres Mittel zur Komplexitätsbeherrschung ist die Bildung von Hierarchien.

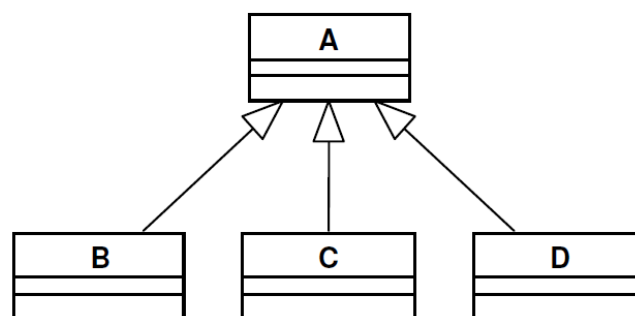


Bild 1-1 Vererbungshierarchie

Typisch für solche Vererbungshierarchien sind die Klassifikationsschemen¹ der Tiere und Pflanzen in der Biologie (siehe Bild 1-2).

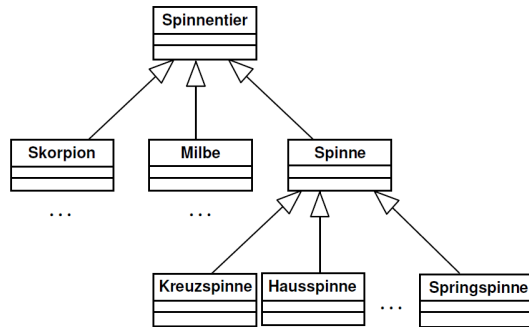


Bild 1-2 Klassifikation von Spinnentieren

¹ Während man in der Objektorientierung nur von Klassen redet, werden in der Biologie die Namen Stamm, Klasse, Unterklasse, Ordnung, Familie, Gattung, Art und Rasse verwendet.

Komposition und Aggregation unterscheiden sich bezüglich der **Lebensdauer** des zusammengesetzten Objektes und seiner Komponenten.

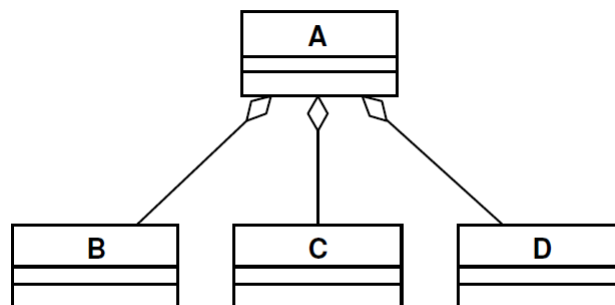


Bild 1-3 Aggregationshierarchie

Agenda

1 Geheimnisprinzip

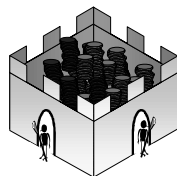
Prinzipien der SW-Technik - Geheimnisprinzip

Geheimnisprinzip im Alltag und in der SW-Technik



Geheimnisse im Alltag

- zum Überraschen.
- zum Verbergen und Vertuschen (wenn was schief gelaufen ist).
- zum Informationen vorenthalten (Macht ausüben).



Geheimnisprinzip in der SW-Technik

- nur die externen Schnittstellen sind sichtbar.
- Interna werden vollständig verborgen.



Warum ist es so wichtig, die Interna zu verbergen?

Geheimnisprinzip (Information Hiding)

- Geheimnisprinzip ist eine Verschärfung des Prinzips der Modularisierung.
- Anwendung des Geheimnisprinzips auf ein Modul bedeutet, dass nur die Schnittstelle des Moduls nach außen sichtbar ist.
- Vorteile:
 - Die Verwendung eines Moduls wird **zuverlässiger**, da nur über die definierten Schnittstellen kommuniziert werden kann.
 - Der Anwender eines Moduls wird nicht mit unnötigen Informationen belastet.
 - Die **Datenkonsistenz** innerhalb des Moduls kann besser sichergestellt werden, da direkte, unkontrollierbare Manipulationen nicht möglich sind.
 - **Fehlersuche** wird **erheblich vereinfacht**, da alle Methoden, welche die Daten manipulieren, im Modul implementiert sind.



Änderungen im Inneren eines Moduls haben keine Auswirkungen auf andere Module, so lange die **Schnittstellen** stabil sind.

Geheimnisprinzip in der Objektorientierung

- Verstecken von Daten und der Implementierung von zugehörigen Methoden in einer Kapsel (Objekt/Klasse).
- Kommunikation mit der Außenwelt nur über wohldefinierte Schnittstellenmethoden.
- Vorteil für Entwickler:
 - Bei der Implementierung der Algorithmen innerhalb der Methoden bestehen hohe Freiheitsgrade.
 - Innerer Aufbau kann optimiert werden, ohne dass sich an den Schnittstellen etwas ändert.
- Vorteil für Benutzer:
 - Muss sich nicht um die internen Details kümmern und muss diese auch nicht verstehen.
 - Kann immer ohne Änderung die neueste Version verwenden, so lange die Schnittstellen identisch bleiben.



Wie unterstützen Objektorientierte Programmiersprachen das Geheimnisprinzip?

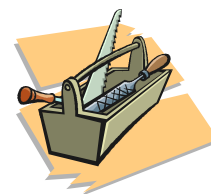
1 (Softwareentwicklungs-) Werkzeuge

(Softwareentwicklungs-) Werkzeuge

(Softwareentwicklungs-)Werkzeuge



Was ist ein Werkzeug in der Softwaretechnik?
Welche Werkzeuge für welchen Zweck kennen Sie?



- sollen Effizienz und Effektivität von Entwicklern erhöhen
 - **Effektiv:** am Ergebnis/Ziel orientiert.
 - **Effizient:** ein Ergebnis/Ziel mit möglichst geringem Mitteleinsatz zu erreichen.
 - Beispiel: Baugrube ausheben.
Effektiv aber nicht effizient: Baugrube mit der Schaufel ausheben.
Effektiv und effizient: Baugrube mit dem Bagger ausheben.
- (benötigtes) Hilfsmittel zur SW-Entwicklung
 - Das beste Werkzeug bringt ihnen nichts, wenn Sie keinen talentierten Handwerker haben, der weiß, wie man das Werkzeug richtig einsetzt.
 - Ohne rudimentäre Werkzeuge kann auch der beste Handwerker kein Haus bauen.

erwarteter Nutzen nur dann, wenn

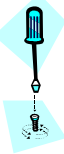
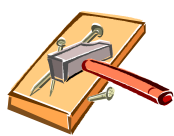
- Anwender die erforderliche Eignung hat:
Wenn der Arbeiter nur eine Schaufel, aber keinen Bagger bedienen kann, nützt ihm der Bagger nichts!
- das Werkzeug für die Aufgabe geeignet ist:
Wenn der Handwerker einen Balken absägen will, nützt ihm der Hammer nichts!
- die eingesetzte Methode zum Werkzeug passt:
Beim Einsatz einer objektorientierten Methode braucht man ein UML-Werkzeug.



Ein Werkzeug muss Teil einer Methode sein, die beschreibt, wann und wie man das Werkzeug sinnvoll einsetzen kann.

Motivation

Aus Erfahrung weiß man, ...



... das Werkzeug muss zur Aufgabe passen!

Für die SW-Entwicklung bedeutet dies:

Ist eine geeignete Methode für ein Projekt gefunden,
die von den Projektmitarbeitern verstanden und akzeptiert wird,
so können die Arbeitsschritte der Methode
durch den Einsatz eines geeigneten Werkzeuges unterstützt werden,
um effizienter zu werden!

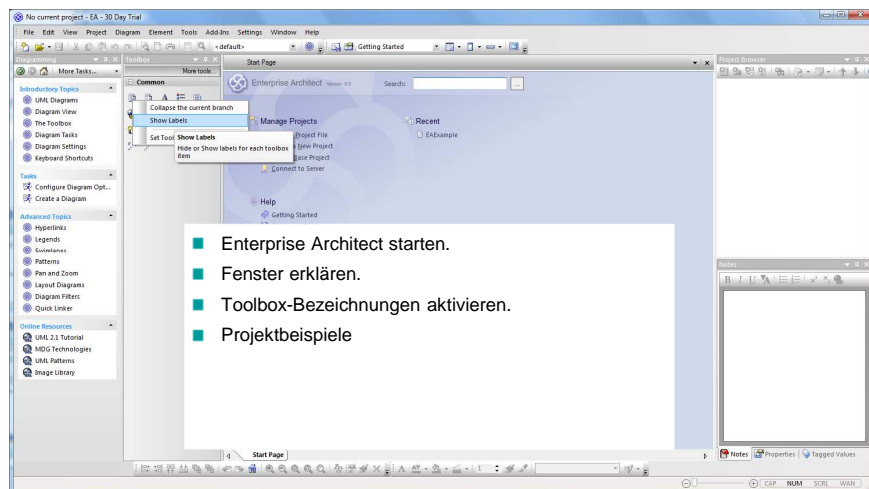
Was ist Enterprise Architect?

- Objektorientiertes Software-Entwicklungswerkzeug für den gesamten Entwicklungsprozess.



- Enterprise Architect unterstützt bei folgenden Tätigkeiten:
 - Geschäftsprozessmodellierung,
 - Anforderungsmanagement,
 - Systemanalyse,
 - Systementwurf,
 - Test
 - und Änderungsmanagement.
- Unterstützung der Modellierung von Systemen mit der UML 2.3.
- Umfassende Informationen unter <http://www.sparxsystems.de>.

Online Demonstration



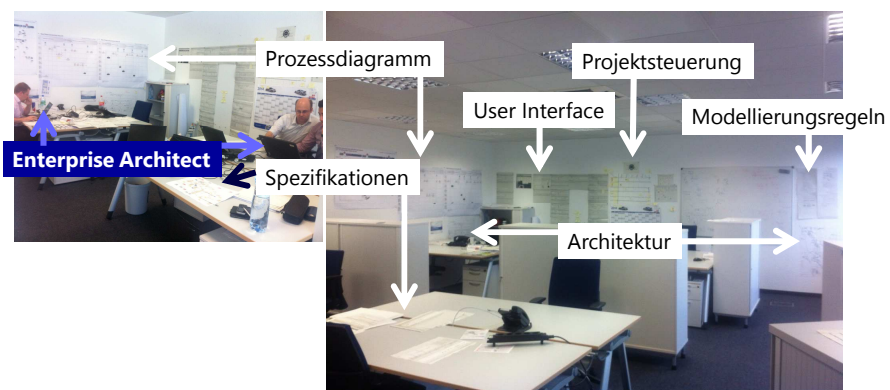
Agenda

- 1 Grundlagen der Modellierung
- 2 Einleitung Vorgehensweise ganzheitliche Modellierung

Einleitung Vorgehensweise ganzheitliche Modellierung

Zielgruppe

▪ Beispiel aus dem Projekt

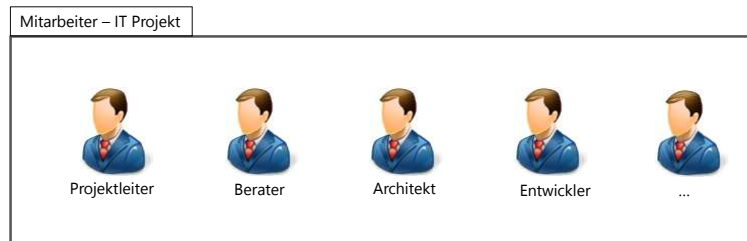


Einleitung Vorgehensweise ganzheitliche Modellierung



Zielgruppe

- **Alle Mitarbeiter** die in IT Projekten (Vor allem Individualentwicklung) involviert sind



© 2018 MHP Management- und IT-Beratung GmbH

41

Einleitung Vorgehensweise ganzheitliche Modellierung



Exkurs: Prozess, Prozessmodell und Vorgehensmodell

Ein **Prozess** ist ein **Arbeitsablauf zur Erledigung einer Aufgabe**.

Ein **Prozessmodell** beschreibt in abstrakter und idealisierter Form die **zeitlich-sachlogische Abfolge von Aktivitäten für einen Prozess**.

Ein Prozessmodell, das die **Abfolge von Aktivitäten** für die **SW-Entwicklung** beschreibt, wird auch **Vorgehensmodell** genannt.

Bei bestimmten Vorhaben (Projekten) haben sich bestimmte Vorgehen als besonders zielführend erwiesen.



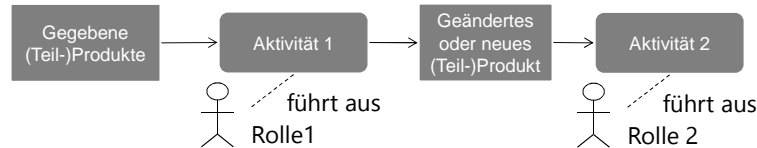
Bewährte **Vorgehensweisen für die Projektabwicklung** werden in **Vorgehensmodellen** dokumentiert.

© 2018 MHP Management- und IT-Beratung GmbH

42

Einleitung Vorgehensweise ganzheitliche Modellierung

Exkurs: Notation für Prozess- und Vorgehensmodelle



Das Basiselement eines Softwareprozesses beschreibt eine **Aktivität**,

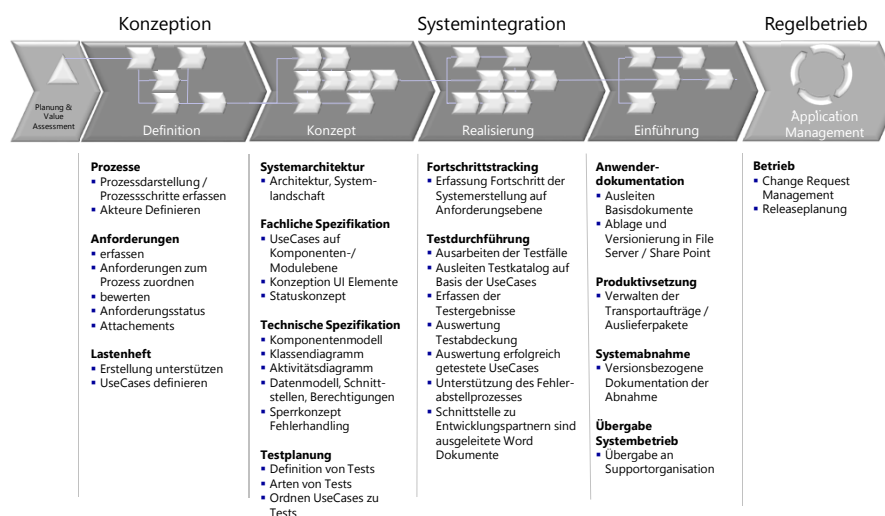
- für die eine **Rolle** (z. B. System-Analytiker, System-Architekt, Projektleiter, Entwickler, Tester, etc., keine konkreten Personen) verantwortlich ist,
- das Ziel einer Aktivität ist es, ein **Produkt fertig zu stellen**,
- für die Erstellung der Produkte können **Werkzeuge** zum Einsatz kommen,
- eine Aktivität wird **innerhalb** einer **Entwicklungsphase** durchgeführt.

Mehrere Aktivitäten werden in (Entwicklungs-)Phasen zusammengefasst:



Einleitung Vorgehensweise ganzheitliche Modellierung

Exkurs: Phasenmodell



Einleitung Vorgehensweise ganzheitliche Modellierung

Exkurs: Klassifizierung von Vorgehensmodellen

Basismodelle

Vorgehensmodelle auf Projektebene

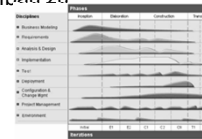
Beispiele: Sequentielles Modell, nebenläufiges Modell, V-Modell, Spiralmodell, etc.



Monumentale Modelle (schwergewichtige Prozesse)

Detaillierte Beschreibungen, wie Prozess- und Qualitätsziele zu erreichen sind – genaue Beschreibung von Prozessschritten sowie Struktur und Aufbau zu erstellender Dokumente.

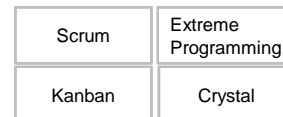
Beispiele: V-Modell XT, RUP.



Agile Modelle (leichtgewichtige Prozesse)

Gegenbewegung zu den immer schwieriger zu durchschauenden und anzuwendenden monumentalen Modellen.

Beispiele: Extreme Programming, Scrum.



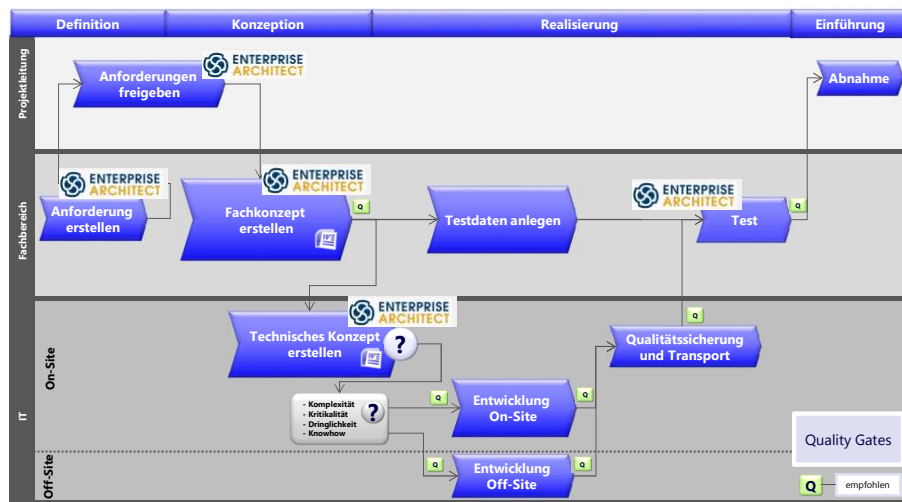
Einleitung Vorgehensweise ganzheitliche Modellierung

Exkurs: Vergleich monumentale und agile Modelle

Monumentales Modell	Agiles Modell
Prozess vorhersagbar	Prozess adaptiv
prozessorientiert	menschen- und teamorientiert
formale Kommunikation	informelle Kommunikation
umfangreiche, formale Dokumentation	minimale, informelle Dokumentation
keine festen Zeitraster	oft feste Zeitraster (time box)
mittelmäßig qualifizierte, z. T. unmotivierte Entwickler	verantwortungsvolle & motivierte Entwickler
Kunden, die wenig mit der Entwicklung zu tun haben wollen	Kunden, die in die Entwicklung einbezogen werden wollen
Teamgröße > 50	Teamgröße < 50
Festpreisauftrag	Auftrag nach Aufwand
Viele Prozesse detailliert und formal	Wenige Prozesse
Schwergewichtige Prozesse	Leichtgewichtige Prozesse

Grundlegende Informationen

Vorgehensmodell für IT Projekte in Beratungsfirmen (Basismodell - MHP)

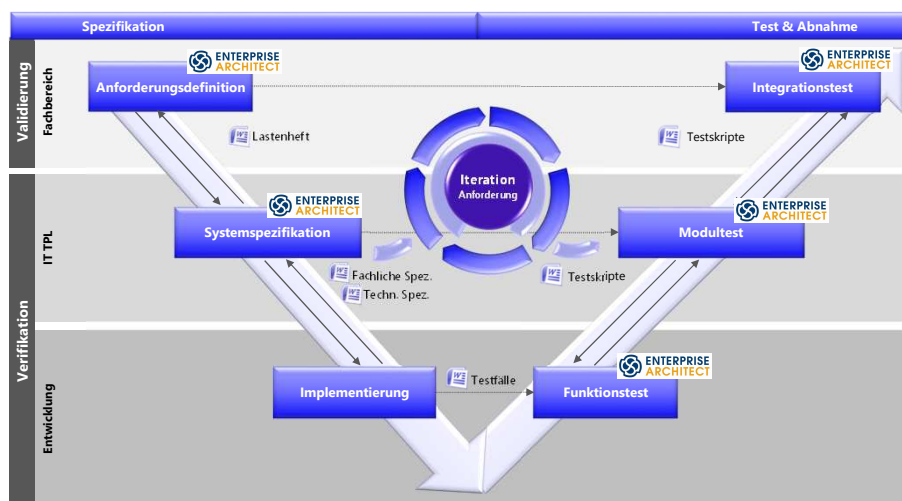


© 2018 MHP Management- und IT-Beratung GmbH

47

Einleitung Vorgehensweise ganzheitliche Modellierung

V-Modell Vorgehensmodell – Individualentwicklung (klassisch und agil)



© 2018 MHP Management- und IT-Beratung GmbH

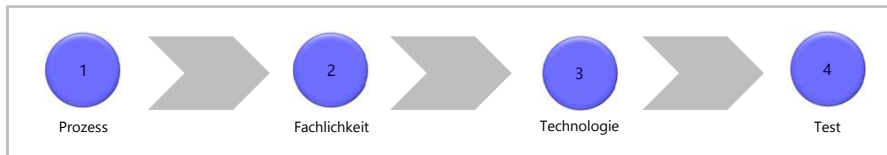
48

Einleitung ganzheitliche Modellierung

Überführung der Informationen ins Modell



Ganzheitliche Modellierung



© 2018 MHP Management- und IT-Beratung GmbH

49

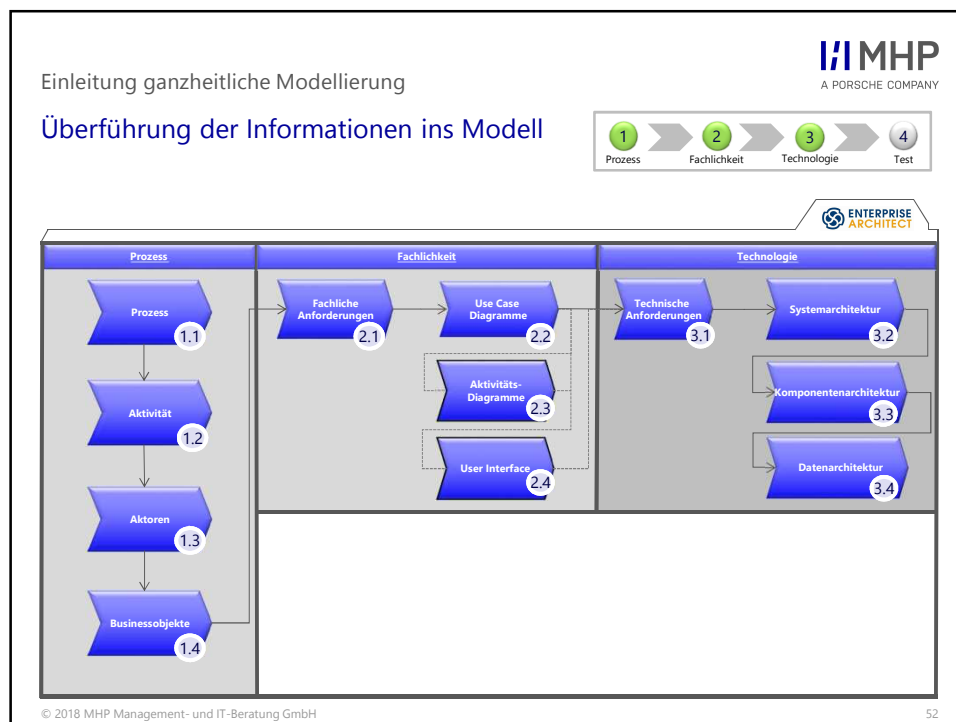
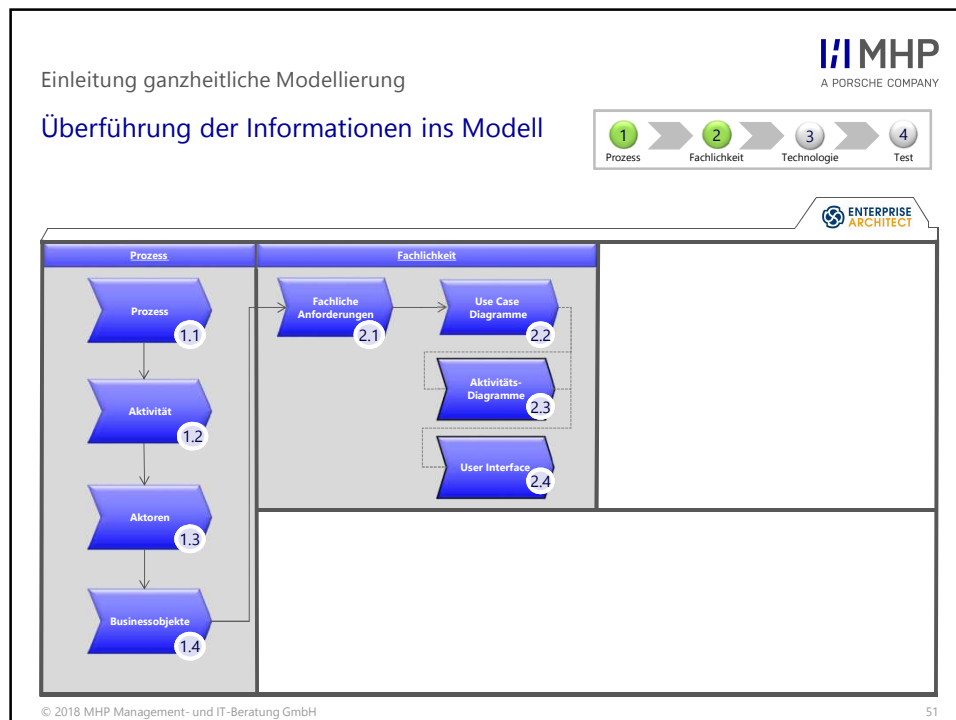
Einleitung ganzheitliche Modellierung

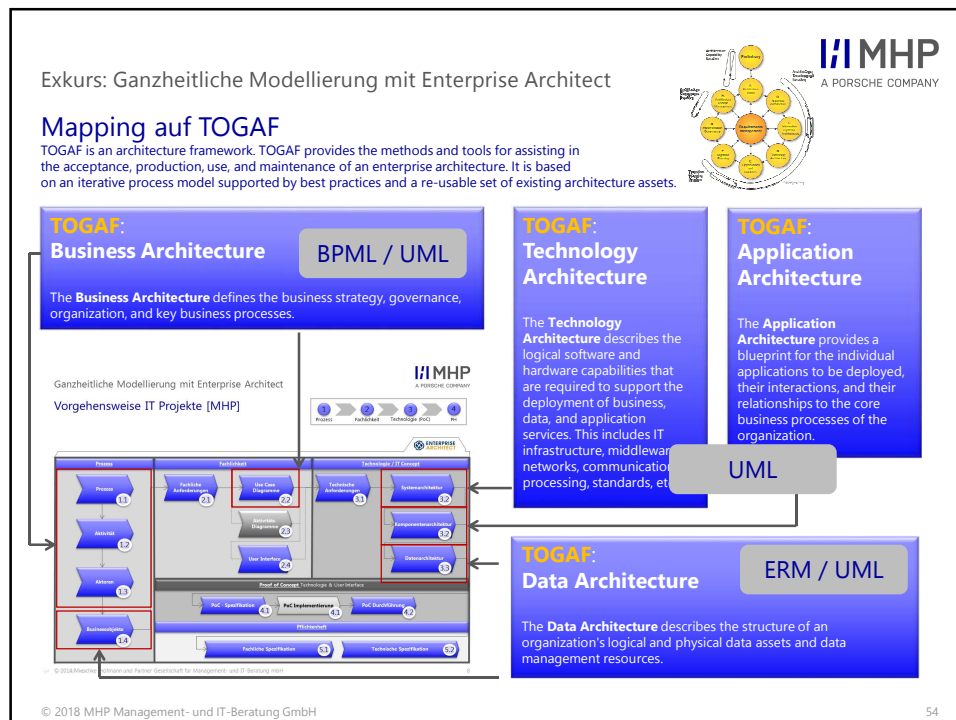
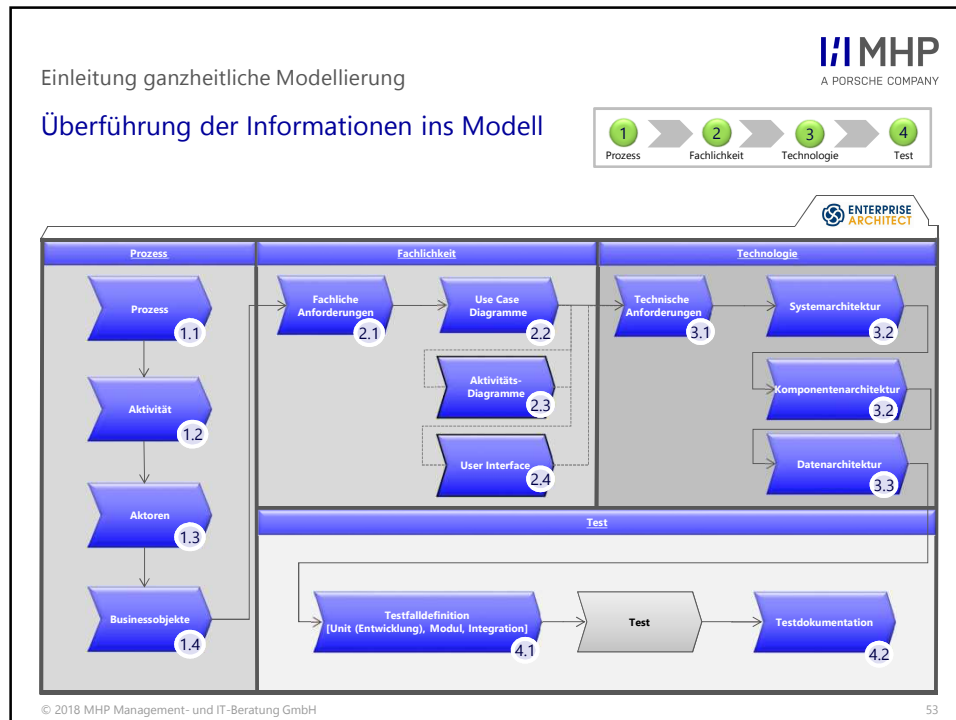
Überführung der Informationen ins Modell



© 2018 MHP Management- und IT-Beratung GmbH

50







Herzlichen Dank
für Ihre Aufmerksamkeit!

Hugo Colceag

MHP
Management- und IT-Beratung GmbH

Film- und Medienzentrums | Königsallee 49 | D-71638 Ludwigsburg
Telefon +49 (0)7141 7856-0 | Fax +49 (0)7141 7856-199
eMail info@mhp.com | Internet www.mhp.com

© 2018 MHP Management- und IT-Beratung GmbH

55