Universität Siegen

wir Wirtschaftswissenschaften Wirtschaftsinformatik Wirtschaftsrecht

Einheit 6:

Systementwicklung

Modellierungsmethoden

Willkommen zur Vorlesung: **Wirtschaftsinformatik II**

uni-siegen.de

Prof. Dr. Rainer Wieching (vertr.)

5. April 2024



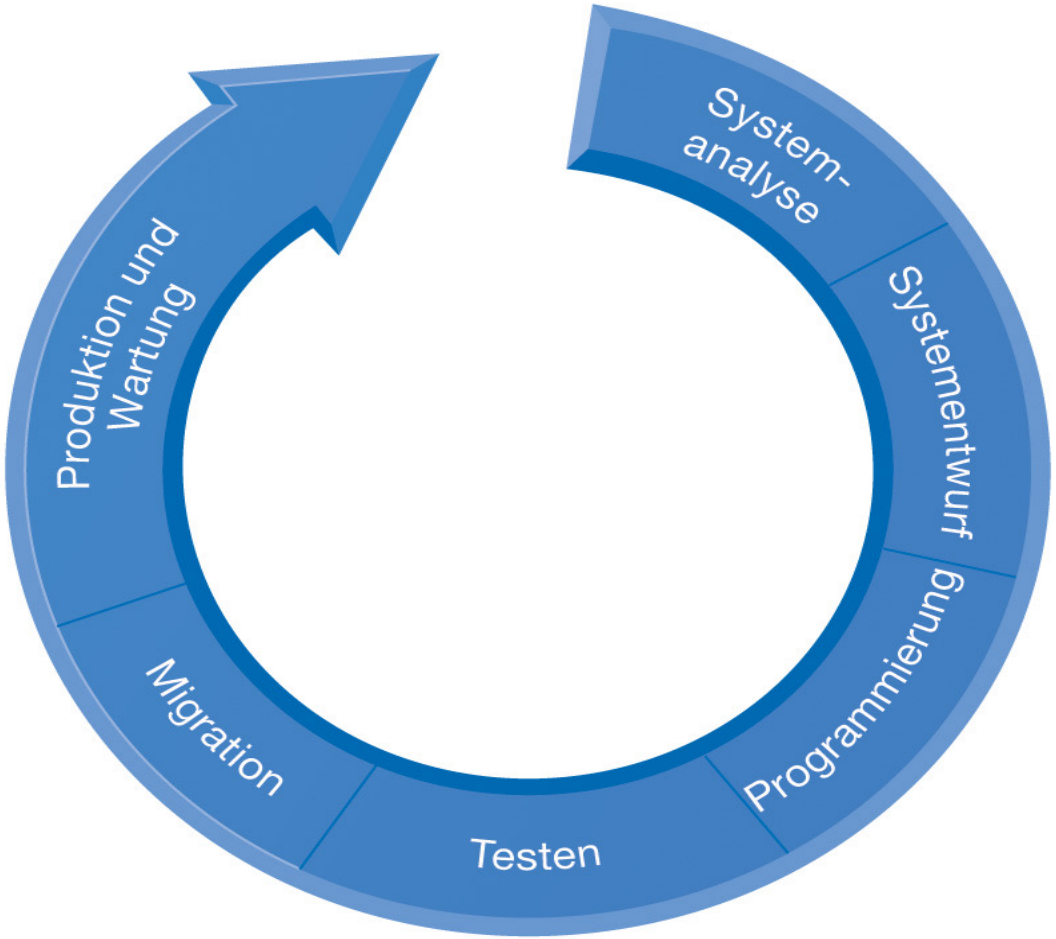
Wirtschaftsinformatik II
Roadmap

- Ethische und Soziale Fragen
- Gestaltungsrelevante Aspekte
- Organisationsstrukturen und –entwicklung
- Systementwicklung: Vorgehensweise
- **Systementwicklung: Modellierungsmethoden**
- Kooperations- und Koordinationssysteme
- ~~Kommunikationssysteme~~
- Wissensmanagement
- Informationsmanagement
- Evaluation von Informationssystemen
- Wrap-Up

Lernziele

1. Die Bedeutung der Modellierung im Systementwicklungsprozess verstehen
2. Kenntnisse über die Modellierungsmethoden zur Unterstützung der Anwendungsentwicklung erwerben
3. Einige Modellierungssprachen kennen lernen

Systementwicklungsprozess



Modelle im Systementwicklungsprozess

Gestaltung von Informationssystemen im Rahmen des Systementwicklungsprozesses wird durch Modelle unterstützt
Modelle sind Repräsentationen

Sie repräsentieren ihre Originale aber in der Regel immer nur für bestimmte Subjekte (die Modellbenutzer) innerhalb bestimmter Zeitspannen sowie für bestimmte Zwecke

Modelle im Systementwicklungsprozess

Sichten:

Teilmodelle, welche jeweils nur bestimmte Arten von Elementen, einen Teil ihrer Merkmale und bestimmte Arten von Beziehungen in einem System betrachten, z.B.

1. Teilfunktionen für Sachbearbeiter,
2. Zugriffsrechte und Daten für Datenschutzbeauftragte,
3. Datenflüsse, Funktionen, Aufgabenträger für Geschäftsprozessverwalter.

Modelle im Systementwicklungsprozess

Sichten können mit unterschiedlichen Begriffen, Modellierungsansätzen und unterschiedlichen Abstraktionen beschrieben werden.

- **Welche verschiedenen Sichten sollten für eine gegebenenfalls vollständige Systembeschreibung berücksichtigt werden?**
- **Mit welchen Begriffen sollten bestimmte Sichten beschrieben werden?**
- **Welche Modellierung ist zur Beschreibung einer bestimmten Sicht (Datensicht, Funktionssicht, Vorgangssicht) besonders geeignet?**

Modellierungsansätze

- Datenmodellierung
- Datenflussansatz
- Objektorientierter Ansatz
- Geschäftsprozessorientierter Ansatz

Datenmodellierung

- Beschreibung der Struktur der Datenbasis eines Informationssystems
- Kernkomponenten sind Datenobjekttypen mit zugehörigen Attributen und Beziehungen
- Die Struktur der Datenbasis dient häufig als Grundlage für die Definition der Funktionen des Informationssystems
- Bekanntes Metamodell zur Datenmodellierung ist das Entity-Relationship-Modell (ERM)

Datenflussansatz

- Ein Informationssystem wird als eine Menge von Datenflüssen definiert, die durch Aktivitäten transformiert werden.

- Beispiel aus der Familie „Strukturierte Ansätze“ ist die „Strukturierte Analyse“ (SA)

  - Strukturierte Entwicklungsverfahren konzentrieren sich vorwiegend auf die Modellierung der Prozesse/ Aktionen, die Daten während des Datenflusses durch das System erfassen, speichern, verarbeiten und verteilen

  - Primäres Werkzeug für die Darstellung ist das Datenflussdiagramm

Beispiel für ein Datenflussdiagramm

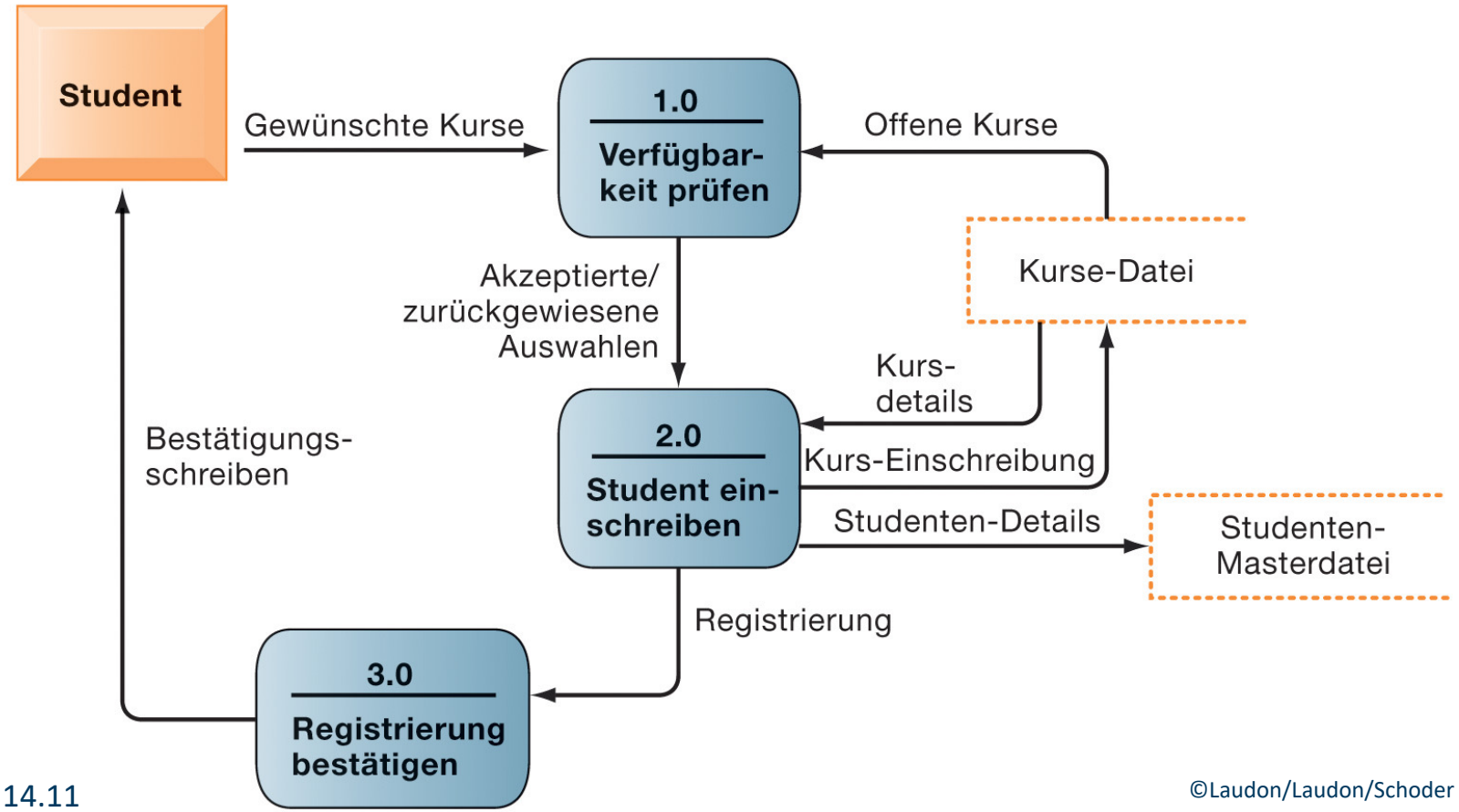


Abbildung 14.11

©Laudon/Laudon/Schoder

Objektorientierter Ansatz

- Objektorientierte Entwicklung verwendet das Objekt als Grundeinheit für Systemanalyse und -entwicklung
- Ein Objekt kombiniert Daten und die spezifischen Prozesse, die an diesen Daten ausgeführt werden.
  - Der Zugriff auf die Daten bzw. die Modifizierung der Daten ist nur über die Operationen möglich, die mit diesem Objekt verknüpft sind.
  - Statt Daten an Prozeduren zu übergeben, senden die Programme eine Nachricht, damit ein Objekt eine Operation durchführt, die bereits in ihm eingebettet ist.

Objektorientierter Ansatz

- Das System ist als eine Sammlung von Objekten und den Beziehungen zwischen Ihnen modelliert.

- Die Objektorientierte Modellierung basiert auf den Konzepten von Klasse und Vererbung
  - Klasse: eine Ansammlung gleichartiger Objekte. Objekte einer bestimmten Klasse weisen die Merkmale dieser Klasse auf und „erben“ Merkmale übergeordneter Klassen.
  - Objektklassen können die gesamte Struktur und das Verhalten einer allgemeineren Klasse übernehmen und dann Variablen und Verhalten hinzufügen, die für das Objekt eindeutig sind.

Klasse und Vererbung (*inheritance*)

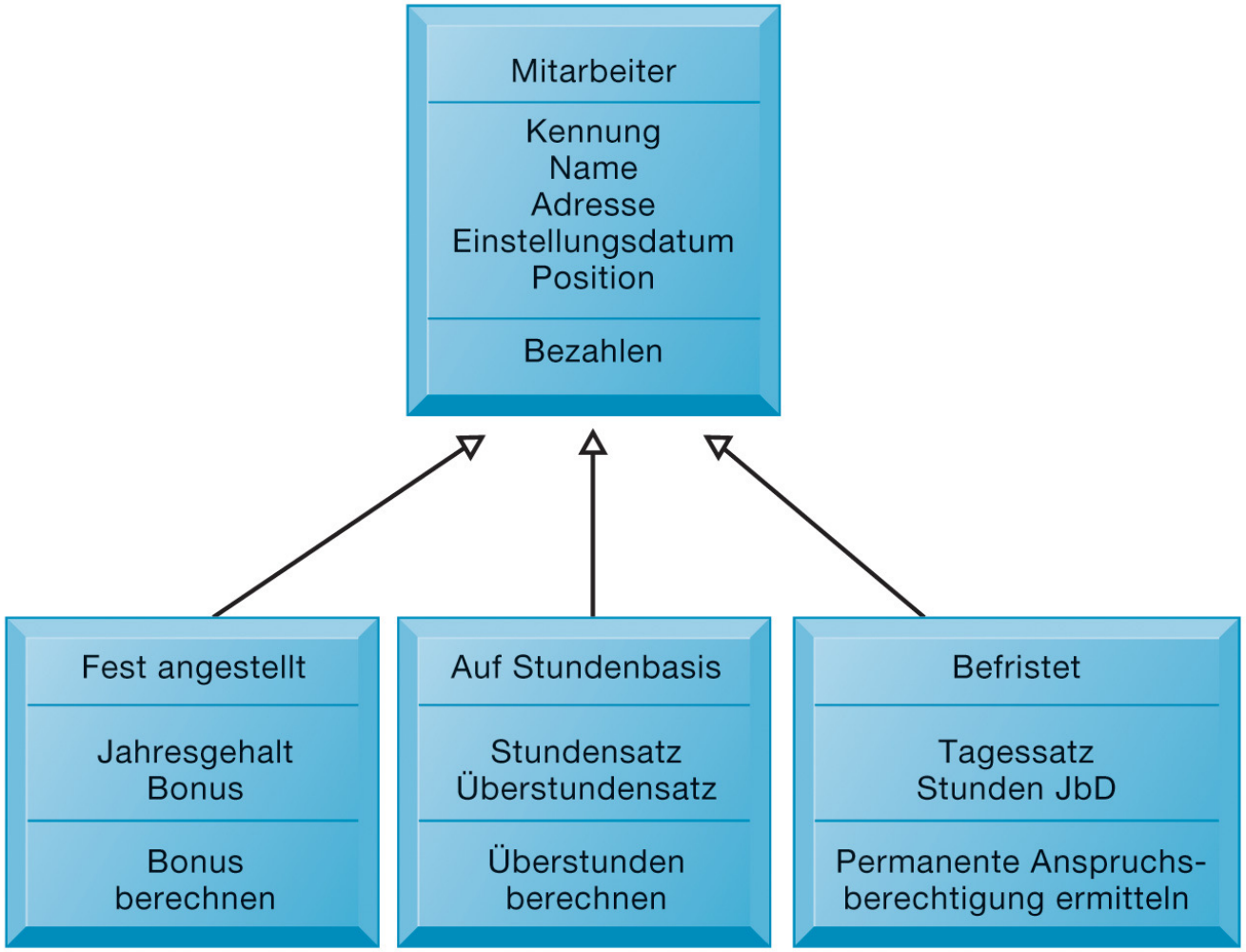


Abbildung 14.13

©

Laudon/Laudon/Schoder

Unified Modeling Language (UML) als Modellierungssprache

- Bietet eine Fülle von Diagrammtypen zur Beschreibung eines Informationssystems
- „Things“ sind Objekte, „Structural Things“ Beziehungen der Objekte zueinander
- Hauptsächlich zwei Arten von Diagrammen:
  - strukturelle Diagramme: beschreiben die Beziehungen zwischen Klassen
  - Verhaltensdiagramme: beschreiben Interaktionen in einem objektorientierten System

Überblick über UML und ihre Komponenten

Tabelle 14.6

UML-Kategorie	UML-Elemente	Spezifizische UML-Details
Things (Dinge)	Structural Things (strukturelle Dinge)	Classes (Klassen)
		Interfaces (Schnittstellen)
		Collaborations (Kollaborationen)
		Use Cases (Anwendungsfälle)
		Active Classes (aktive Klassen)
		Components (Komponenten)
		Nodes (Knoten)
	Behavioral Things (Verhaltensdinge)	Interactions (Interaktionen)
		State Machines (Automaten)
	Grouping Things (gruppierende Dinge)	Packages (Pakete)
	Annotational Things (Anmerkungsdinge)	Notes (Anmerkungen)
Relationships (Beziehungen)	Structural Relationships (strukturelle Beziehungen)	Dependencies (Abhängigkeiten)
		Aggregations (Aggregationen)
		Associations (Assoziationen)
		Generalizations (Verallgemeinerungen)
	Behavioral Relationships (Verhaltensbeziehungen)	Communicates (kommuniziert)

Fortsetzung auf nächster Folie

© Laudon/Laudon/Schoder

Überblick über UML und ihre Komponenten

		Includes (beinhaltet)
		Extends (erweitert)
		Generalizes (verallgemeinert)
Diagrams (Diagramme)	Structural Diagrams (strukturelle Diagramme)	Class Diagrams (Klassendiagramme)
		Object Diagrams (Objektdiagramme)
		Component Diagrams (Komponentendiagramme)
		Deployment Diagrams (Einsatzdiagramme)
	Behavioral Diagrams (Verhaltensdiagramme)	Use Case Diagrams (Anwendungsfalldiagramme)
		Sequence Diagrams (Sequenzdiagramme)
		Collaboration Diagrams (Kollaborationsdiagramme)
		Statechart Diagrams (Statusdiagramme)
		Activity Diagrams (Aktivitätsdiagramme)

Tabelle 14.6

© Laudon/Laudon/Schoder

Ein UML-Anwendungsfalldiagramm

Anwendungsfalldiagramme modellieren die Funktionen eines Systems und zeigen dabei, wie die Objekte miteinander und mit den Benutzern des Systems interagieren.

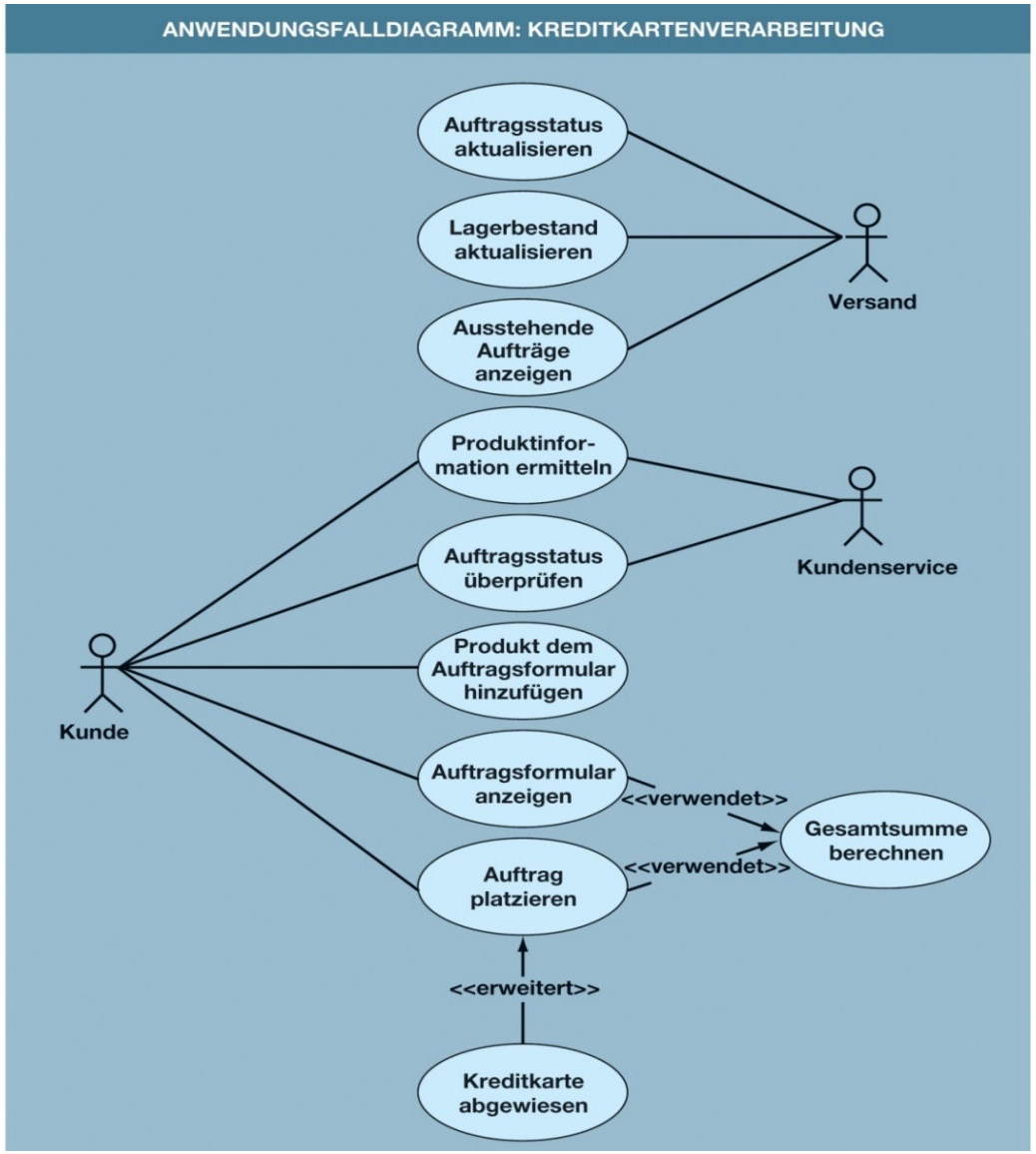


Abbildung 14.14

© Laudon/Laudon/Schoder

Ein UML-Sequenzdiagramm

Sequenzdiagramme beschreiben Interaktionen zwischen Klassen in Hinblick auf die Kommunikation zwischen Objekten (Nachrichten) während eines bestimmten Zeitintervalls. Hier ist ein Sequenzdiagramm für eine Webseite gezeigt, die mit einer Datenbank interagiert.

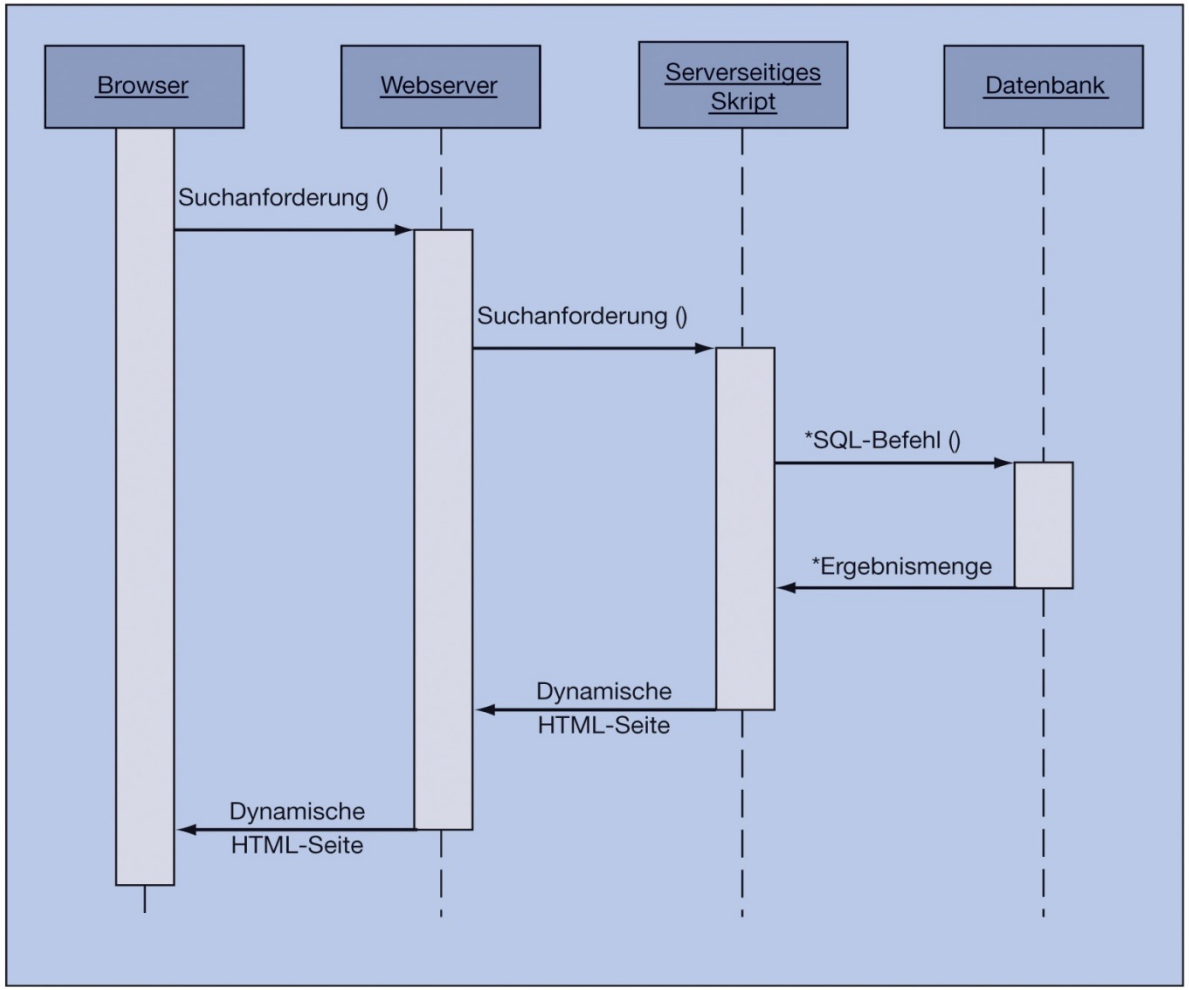


Abbildung 14.15

© Laudon/Laudon/Schoder

Geschäftsprozessorientierter Ansatz

- Betonen die Hinwendung von einer primär statischen und strukturorientierten Sicht eines Informationssystems zu einer dynamischen und verhaltenstheoretischen Sicht
- Modellierung umfasst neben Aufgabenebene auch Aufgabenträgerebene (wer oder welche Rollen)
- Weitverbreiteter geschäftsprozessorientierter Modellierungsansatz ist ARIS

Beispiel ARIS:
Architektur integrierter Informationssysteme

- **Ganzheitliche Betrachtung von Geschäftsprozessen**

- **Fünf Sichten auf ein Informationssystem:**
  - Organisationssicht (Organisationseinheiten, Standorte)
  - Funktionssicht (Ziele und Funktionen)
  - Datensicht (Daten zur Erfüllung der Funktionen)
  - Leistungssicht (Sach- und Dienstleitungen)
  - Steuerungssicht (Integration der anderen Sichten)
    - Prozesse werden als Abfolge von Funktionen verstanden
    - Abfolge von Funktionen werden ereignisorientiert definiert
    - Die Ausführung einer Funktion wird durch Ereignisse ausgelöst.

Integration der vier Basissichten durch die Steuerungssicht

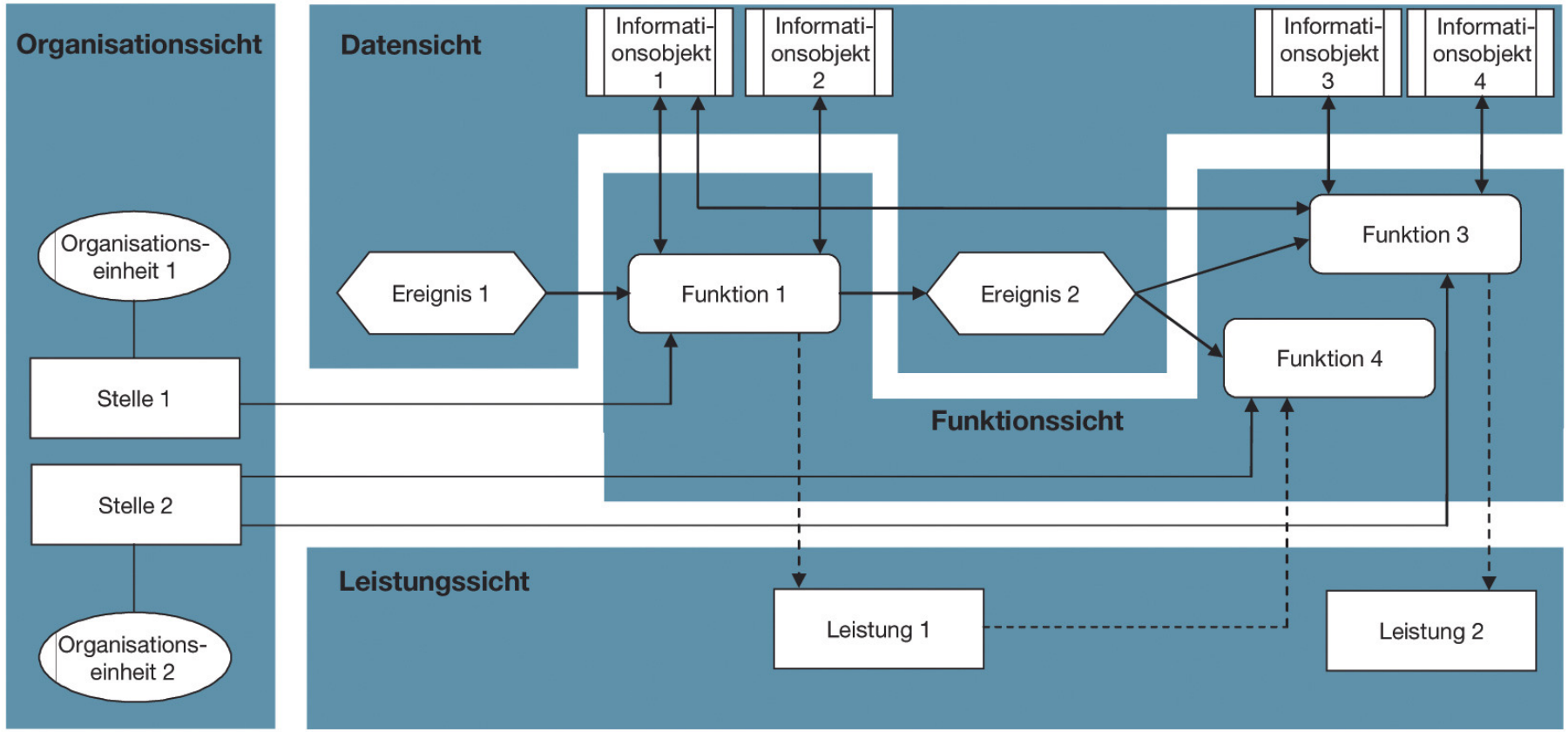


Abbildung 14.18

© Laudon/Laudon/Schoder

EXKURS: ePK-Modellierung

- Ereignisgesteuerte Prozessketten (ePK) sind semiformale grafische Darstellungen, die hauptsächlich dazu benutzt werden, um Geschäftsprozesse bzw. deren Modelle (Schemata) darzustellen

- Die ereignisgesteuerte Prozesskette beschreibt einen Geschäftsprozess als zeitlich-logische Abfolge betriebswirtschaftlicher Aufgaben

EXKURS: Einige ePK-Elemente

Notationselement	Beschreibung
Ereignis / Zustand 	Beschriftungskonvention: Bei Ereignissen wird das Objekt, welches eine Zustandsänderung erfährt, mit einem Verb im Partizip Perfekt verbunden (z. B. „Kundenauftrag (ist) angelegt"). Ein Event kann sein: ein Trigger (Auslöser) für eine bestimmte Funktion, welcher einen bestimmten zu erfüllenden Status genauer beschreibt (Input für eine Funktion); ein Status, der erreicht wurde, nachdem ein Vorgang stattgefunden hat (Output einer Funktion).
Funktion 	Zur Beschriftung von Funktionen gilt die Vorgabe Verb–Objekt. Das Verb soll immer in der Präsensform stehen. Beispielsweise: „wähle Kunde", „bestelle Ware x". Eine Funktion steht für einen bestimmten Ablauf oder Auftrag, welcher von einer bestimmten Person ausgeführt wird und einen gewissen Input braucht, um einen Zustand zu erreichen und auch um eventuell einen definierten Output zu erzeugen.
Logische Operatoren XOR, OR, AND 	Die logischen Operatoren werden benutzt, um Funktionen und Events zu verbinden oder um Folgendes abzubilden: Entscheidungen oder Optionen XOR, wenn nur EINE der angegebenen Optionen möglich sein soll; OR, wenn mehrere Optionen möglich sind; AND, wenn parallele Ausführungen von Funktionen vorgesehen sind.

Abbildung 14.7.1

© Laudon/Laudon/Schoder

EXKURS: Einige ePK-Elemente

Organisationseinheit 	Die Organisationseinheiten stehen für Rollen oder Personen, die verantwortlich für bestimmte Funktionen sind.
Informationsobjekt 	Ein Informationsobjekt kann als Input oder Output einer Funktion angesehen werden.
Prozesspfad / Prozesswegweiser 	Der Prozesspfad oder Prozesswegweiser ist mit der oben erwähnten „Funktion" vergleichbar. Er wurde eingeführt, um zu zeigen, dass ePKs hierarchisch beschrieben werden können, beginnend mit einer sehr abstrakten, allgemeinen Beschreibung und fortlaufend detaillierteren Funktionen. Des Weiteren repräsentiert ein Prozesspfad komplexe Aktivitäten (beispielsweise eine Aufgabe oder eine Unteraufgabe), die in einer separaten ePK (detaillierter) beschrieben werden. Ähnlich lässt sich verfahren, wenn aus „Platzgründen" mehrere Aktivitäten gruppiert und durch ein Prozesspfadelement ersetzt werden sollen.
	Kontrollfluss
	Ressourcenfluss
	Informationsfluss

Abbildung 14.7.2

© Laudon/Laudon/Schoder

EXKURS: ePK-Modellierung

- **Einige Regeln und Konventionen**
  - **Regel 1:** Ein ePK-Modell muss mit einem Ereignis, dem sogenannten Startevent, beginnen.
  - **Regel 2:** Ein ePK-Modell muss mit einem Ereignis, dem sogenannten Endevent, enden.
  - **Regel 3:** Funktionen und Ereignisse sollen abwechselnd vorkommen!
  - **Regel 4:** Bezüglich jeder Verbindung zwischen Ereignissen und Funktionen gilt, dass jedes Event und jede Funktion nicht mehr als einen Input- und einen Output-Konnektor haben dürfen.

EXKURS: Business Process Modelling Notation (BPMN)

- **Die Object Management Group verfolgt mit der Standardisierung der Business Process Modelling Notation (BPMN) zwei Ziele:**
  - Es soll eine standardisierte, grafisch unterstützte Modellierungssprache für Geschäftsprozesse entstehen, die sowohl von Mitarbeitern der Fachbereiche als auch von IT-Fachleuten verstanden wird
  - Diese gemeinsame Prozessmodellierungssprache soll darüber hinaus auf ausführbare Prozessmodellierungssprachen, wie z.B. die XML-basierte BPEL4WS (Business Process Execution Language for Web Services) abbildbar sein

EXKURS: BPMN-Diagramm des Testprozesses

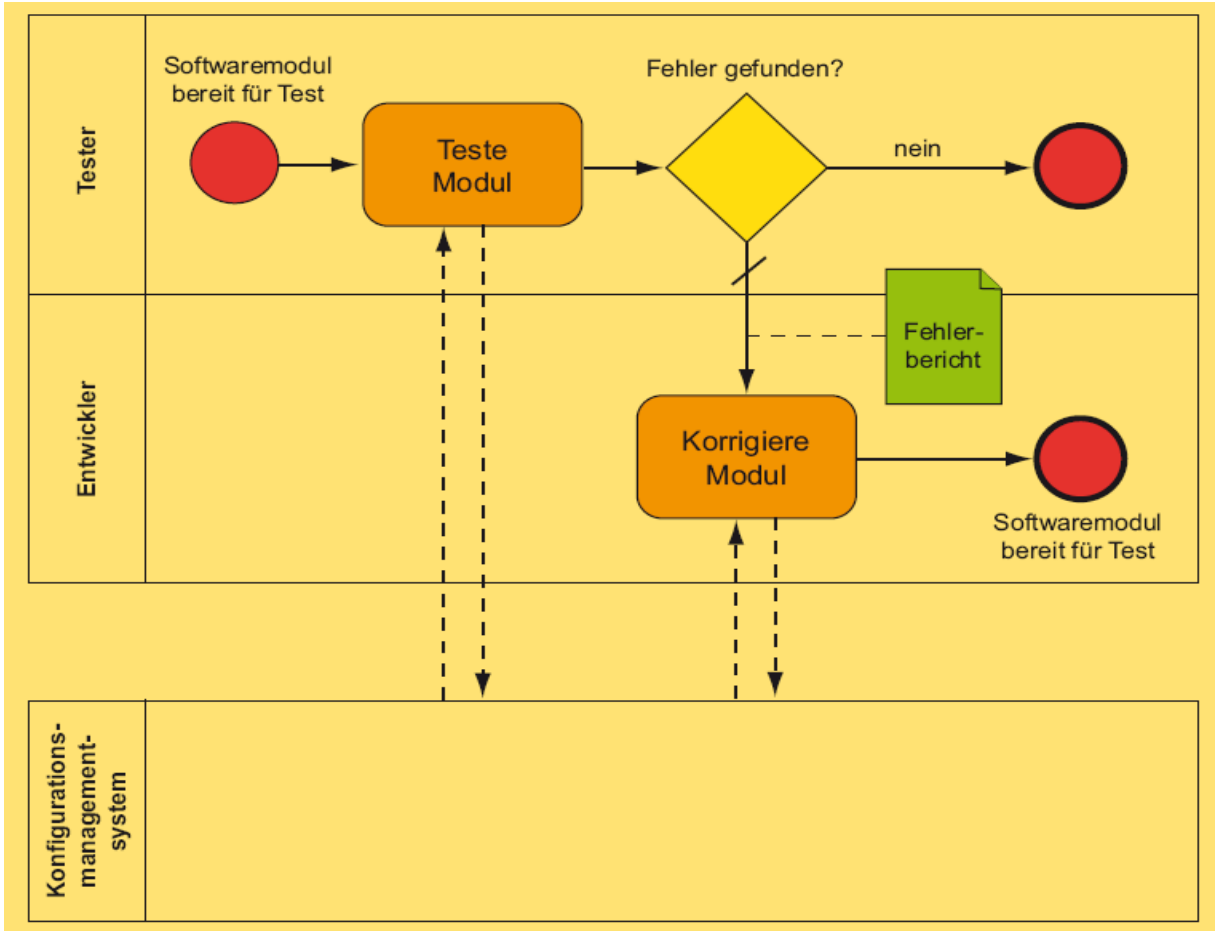


Abbildung 14.19

©Laudon/Laudon/Schoder

Literatur

Basisliteratur

- **Kapitel 14. aus: K. Laudon, J. Laudon, D. Schoder: Wirtschaftsinformatik - Eine Einführung, Pearson Studium, 2010.**

Weitere relevante Quellen

- **Balzert, H. Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering . 3. Auflage, Heidelberg: Spektrum Akademischer Verlag. 2009. Kapitel: Anforderungen modellieren.**

- **Zuser, W.; Grechening, T.; Köhle, M. Software Engineering. Mit UML und dem Unified Process. Kapitel 7: Think UML.**

Vielen Dank!

Kontakt

Fakultät III
Kohlbettstraße 15
57223 Siegen

rainer.wieching@uni-siegen.de