

Aufgabe 1a: Enthaltene Informationen

- Die Grafik zeigt die Beschreibung der Architektur eines Datenbank-Interfaces
- Beschreibung der statischen Struktur mit den Bestandteilen Workspace, Query, Datastore, ...
- Eine informelle Notation, mit Pfeilen und Kästen (Box-and-Arrow) wird verwendet.
- Zusätzlich wird Farbe eingesetzt, um die Kästen näher zu klassifizieren (A-Software, T-Software, ...).
- Die Grafik unterscheidet zwischen Schnittstellen und Implementierungen.

Aufgabe 1b: Fehlende Informationen

- Kontextinformationen:
 - Rolle im Entwicklungsprozess
 - Bearbeiter, Rolle, Zielgruppe, ...
- Semantik:
 - Was bedeutet die Notation genau?
 - Was ist A-Software, T-Software, ...?
- Rolle der Objekte:
 - Wozu ist das dargestellte Datenbank-Interface nützlich?
 - Welche Aufgaben hat es und wie wird es verwendet?
- Separation of Concerns:
 - Striktere Klassifikation der Bestandteile.
 - Aufteilung in unterschiedliche Sichten

Aufgabe 1c: Vorteile der UML-Darstellung

- Semiformale Modellierung der Architektur
 - Genaue Vorgabe der Syntax
 - Vielzahl an Diagrammtypen
 - Werkzeugunterstützung
 - ...allerdings fehlende Semantikdefinition
- UML-Beschreibungstechniken (Ver. 2.0)
 - Statische Struktur: Klassen-/Komponentendiagramme
 - Schnittstellen: Klassen-/Komponentendiagramme (importiert, exportiert)
 - Verhalten: Statecharts, Aktivitätsdiagramme
 - Ablauf, Interaktion: Sequenzdiagramme, Kollaborationsdiagramme
 - Verteilung: Deploymentdiagramme
- Erkannte Verbesserungspotenziale:
 - Schnittstellenverhalten (Partialität)
 - Softwarespezifikationen (Threads, Schedules, ...)
 - Formulierung nicht-funktionaler Eigenschaften

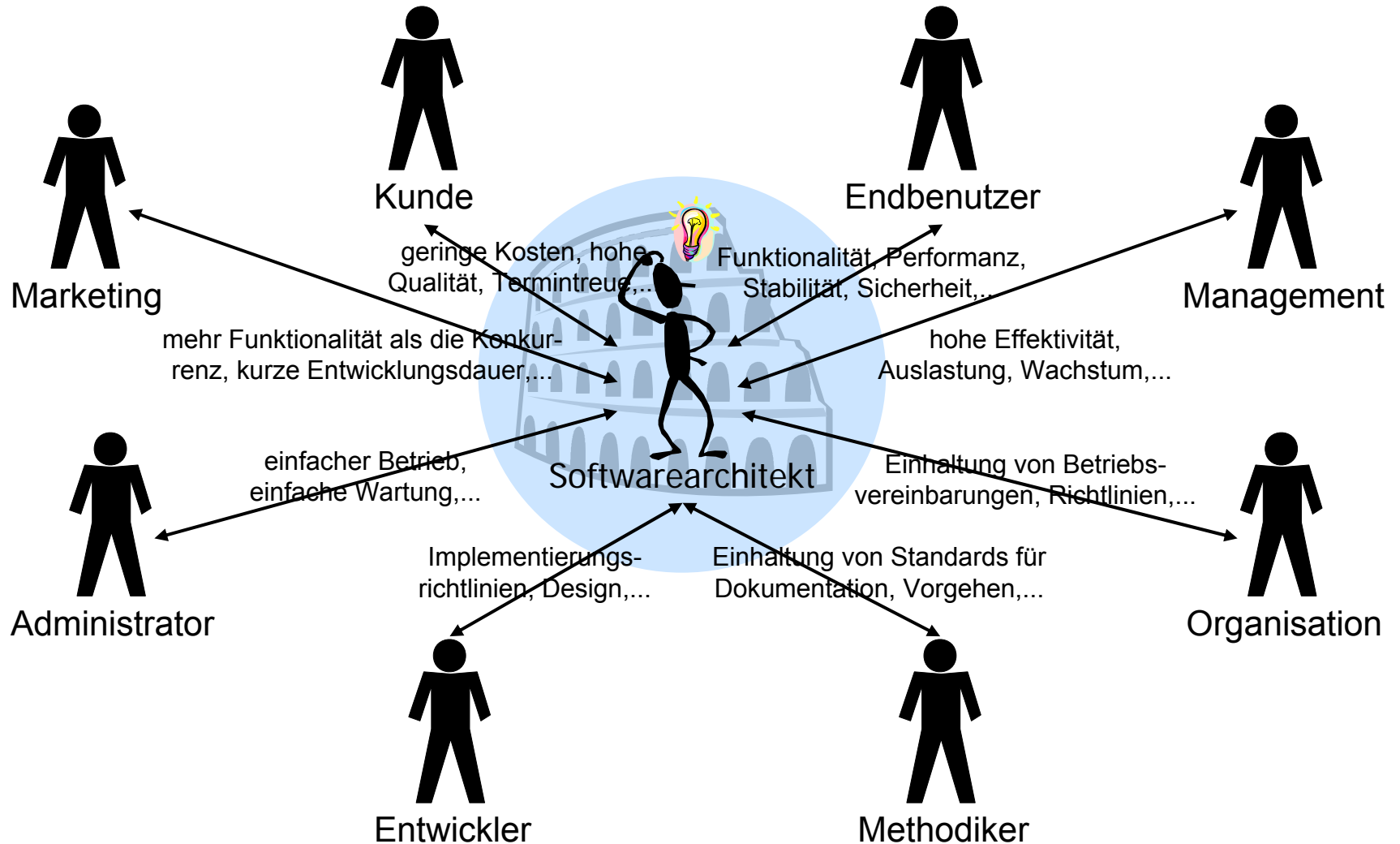
Aufgabe 1d: Vorteile formale Notation

- Die UML ist eine semiformale Notation.
- Was genau ein Klassendiagramm bedeutet ist nicht festgelegt.
- Eine Überführung beispielsweise nach Java oder C++ ist nicht eindeutig definiert.

- Die FOCUS-Theorie, welche in der Vorlesung dargestellt wurde, schließt diese Lücke.
- Die Semantik der Darstellungselemente ist exakt festgelegt.
- Struktur und zeitliches Verhalten der Architekturelemente ist präzise definiert.

- Was hat das für Vorteile?
 - Verschiedene Interpretationen und Mehrdeutigkeiten sind nicht möglich.
 - Eine formale Darstellung ist maschinell analysierbar bzw. verifizierbar.
 - Generierung der Implementierung aus der Spezifikation

Aufgabe 1e: Zielgruppe der Beschreibung

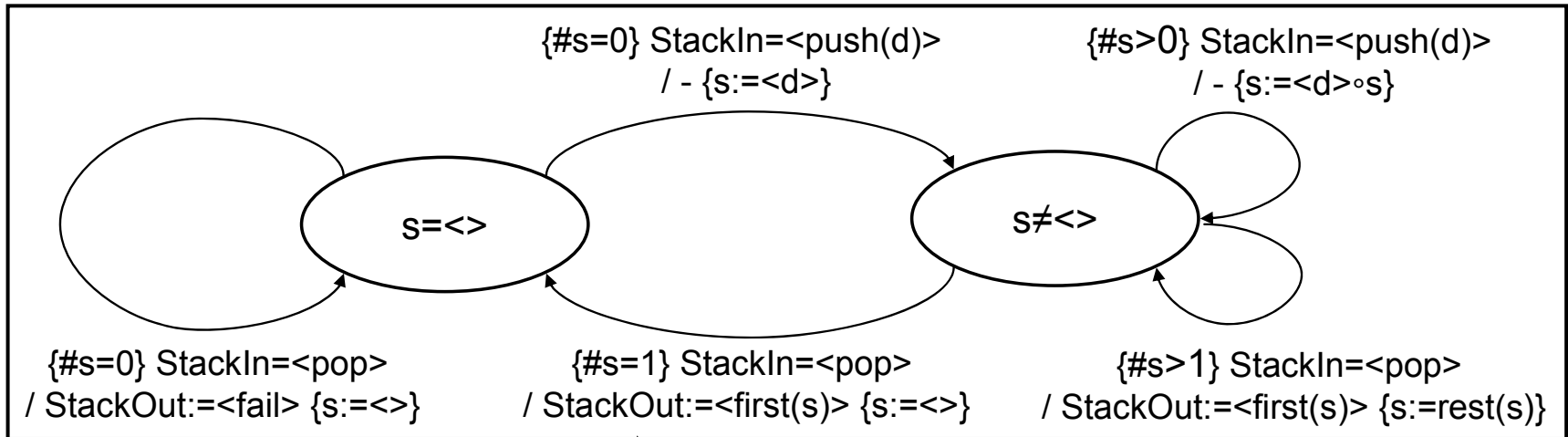


Aufgabe 1f: Eignung als ADL

- Die Frage kann nicht pauschal beantwortet werden.
- Für den Projektmanager und eventuell auch für die Auftraggeber des Datenbank-Interfaces ist die gezeigte Darstellung geeignet.
- Für die meisten Zielgruppen fehlen Informationen, etwa Details zu den Schnittstellen, Laufzeitverhalten oder nicht-funktionale Eigenschaften.
- Für einen Endanwender ist die Darstellung nicht geeignet.

Aufgabe 1g: Verfeinerung der Komponenten

DataStore



Input = $\{ \text{push}(d): d \in \text{Data} \} \cup \{ \text{pop} \}$

Output = $\{ \text{top}(d): d \in \text{Data} \} \cup \{ \text{fail} \}$