

UML in der Hochschullehre: Eine kritische Reflexion

Jürgen Anke

Hochschule für Telekommunikation Leipzig

anke@hft-leipzig.de

Stefan Bente

Technische Hochschule Köln

Cologne Institute for Digital Ecosystems (CIDE)

stefan.bente@th-koeln.de



Hochschule für Telekommunikation Leipzig
University of Applied Sciences

Technology
Arts Sciences
TH Köln

Ausgangsfrage „Braucht man UML in der Lehre noch?“

■ Beobachtungen

- Hohe Komplexität der Sprache (und Tools) überfordern Studierende
- Agile Entwicklung, Microservices, ... erfordern keine komplexen Modelle
- Unklare Relevanz von UML in der Praxis

■ Forschungsfragen

- Wie und wofür wird die UML in der Praxis tatsächlich eingesetzt?
- Welche Kompetenzen sollten dafür durch die Lehre entwickelt werden?

■ Vorgehen

- Analyse empirischer Studien zum Einsatz von UML in der Praxis
- Analyse des Einsatzes von UML in studentischen Arbeiten

■ Ergebnisse

- Sieben Thesen zur Verwendung von UML in der Lehre
- Impulse zur Entwicklung neuer didaktischer Formate

Agenda

- UML-Nutzung durch Praktiker
- Tendenzen bei der UML-Nutzung
- Untersuchung von Hochschul-Abschlussarbeiten
- Sieben Thesen zur Verwendung von UML in der Lehre

UML-Nutzung durch Praktiker

Wofür wird modelliert?

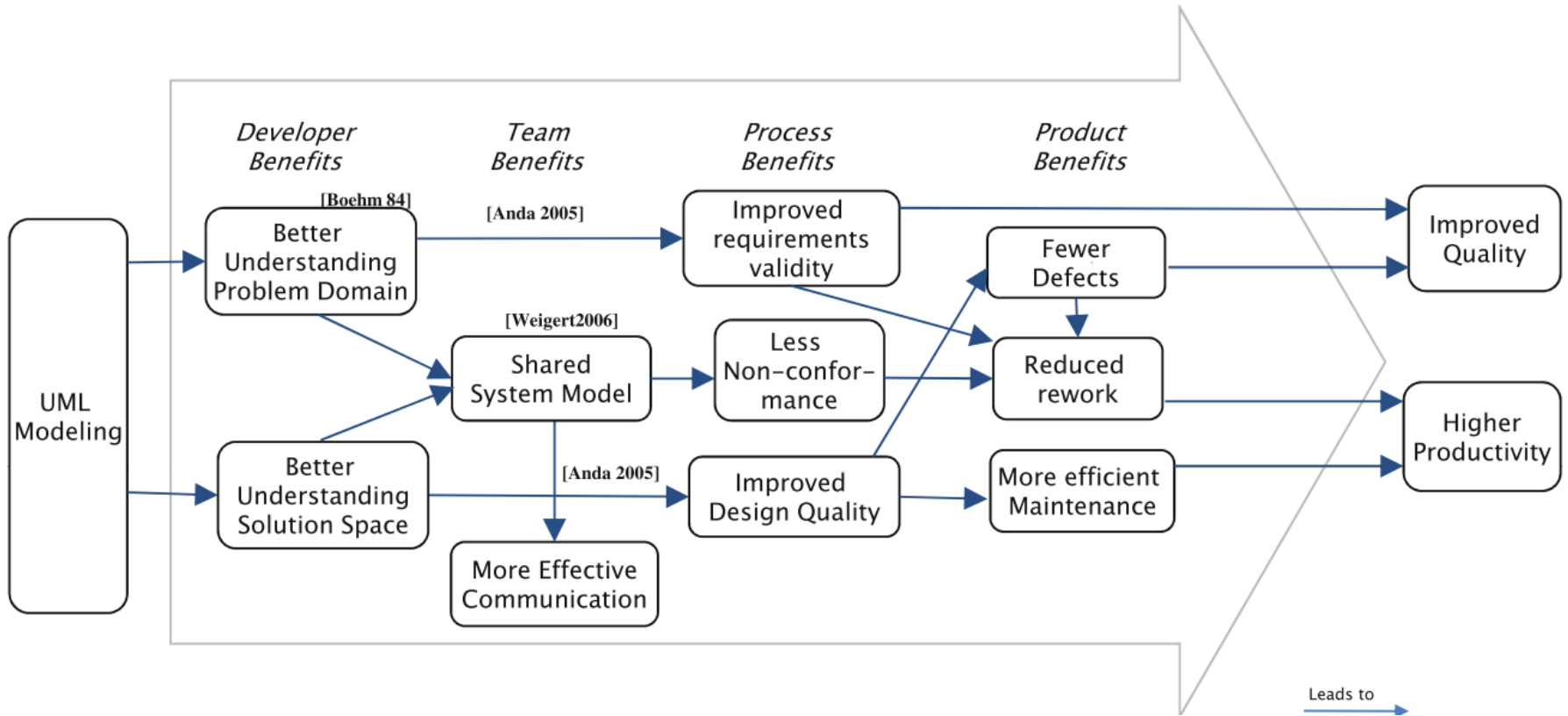
Einsatzzwecke von Modellen im SE (Chaudron u. a., 2012)

1. zur Analyse und Verstehen
2. zur Kommunikation
3. als Vorlage für die Implementierung („Entwurf“)
4. als Basis für die Codegenerierung

Dabei steigen Anforderungen an Modellqualität (Korrektheit, Genauigkeit, Vollständigkeit, ...) mit jeder Stufe

Ziel: **Entwicklerproduktivität** und **Produktqualität** erhöhen.

Nutzen der Modellierung mit UML (Chaudron u. a., 2012)



Ausgewählte Studien

Autoren	Jahr	Methode	TN/Objekte	
Erickson & Siau	2007	Delphistudie		Häufigkeit von UML-Elementen
Dobing&Parsons	2010	Befragung	284	
Reggio et al.	2013	Dokumentenanalyse (Lehrbücher, Tutorials usw.)		
Petre	2013	Experteninterviews	50	
Langner et al.	2014	Quantitative Analyse öffentlich zugänglicher UML Modelle	121	
Davis et al.	2006	Befragung zur konzeptionellen Modellierung	312	Einsatzkontexte konzeptioneller Modelle
Osman & Chaudron	2013	Analyse von Open Source Repositories	10	
Gorschek et al.	2014	Befragung zu Entwurfsmodellen	3785	
Baltes & Diehl	2014	Befragung zum Einsatz von Skizzen u. Diagrammen im SE	394	

Tendenzen bei der UML-Nutzung

Wesentliche Ergebnisse

- UML hat hohe Bekanntheit und häufige Nutzung, jedoch nur **bestimmte Diagrammtypen**. Häufigste Diagrammtypen:
 - Klassendiagramm
 - Aktivitätsdiagramm
 - Use Case-Diagramm
 - Sequenzdiagramm
- Häufiger Einsatz in **frühen Phasen** der Systementwicklung, besonders für Analyse, Kommunikation und Zusammenarbeit
- Eingesetzte Modelle sind **häufig informell** oder gemischt
- Medien: **Papier, Whiteboard** und **Tools** (klassische UML-Werkzeuge, Zeichenprogramme und web-basierte Tools)
- **Erzeugung von Code** spielt untergeordnete Rolle, etabliert v.a. für langlebige Systeme in gut verstandenen Domänen

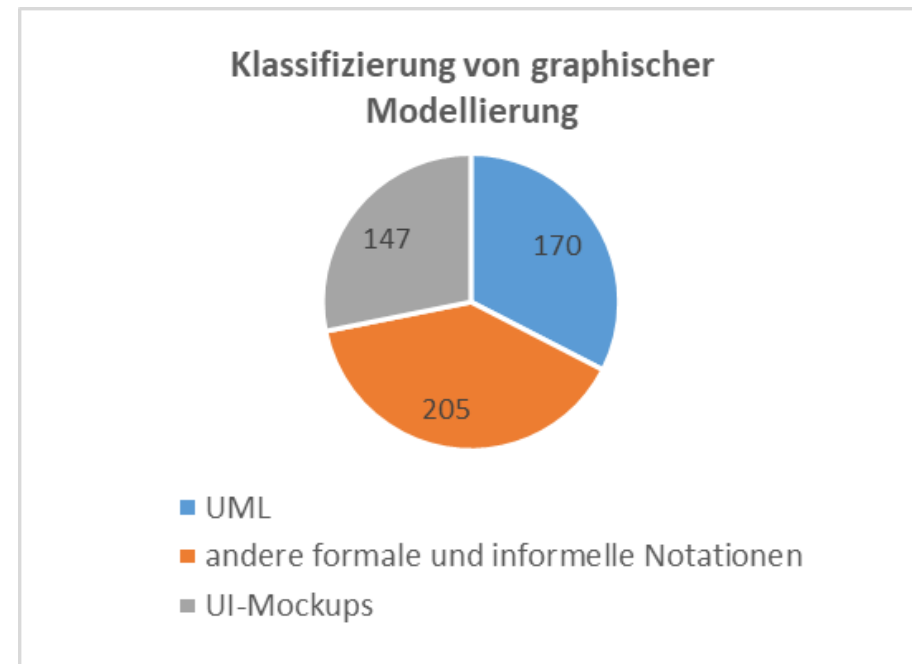
(Mögliche) Gründe für veränderte Nutzung von UML

- Diversifizierung der Softwareproduktarten
 - Webanwendungen, Apps, Enterprise Software, Embedded Software, ...
 - Unterschiede: Stabilität der Anforderungen, technische Plattformen, Compliance, Sicherheit, Umfang, Größe der Projektteams, Formalisierung des Vorgehensmodells
- Agile Entwicklung
 - reduziert Komplexität durch Aufteilung in Iterationen
 - Keine umfangreiche Spezifikation von Anforderungen („Lastenheft“) und Systemarchitektur mehr notwendig / sinnvoll
- Microservices als Architekturstil
 - lose gekoppelte, fachlich orientierte Dienste → individuelle Lebenszyklen
 - Geringere Komplexität, keine Top-Down-Modellierung des Gesamtsystems
- Domain-driven Design als Entwurfsansatz
 - Fachlich orientierte Modellierung → Analyse, Domänenverständnis
 - Einsatz informeller/textueller Notationen (Context Map, Ubiquitous Language)

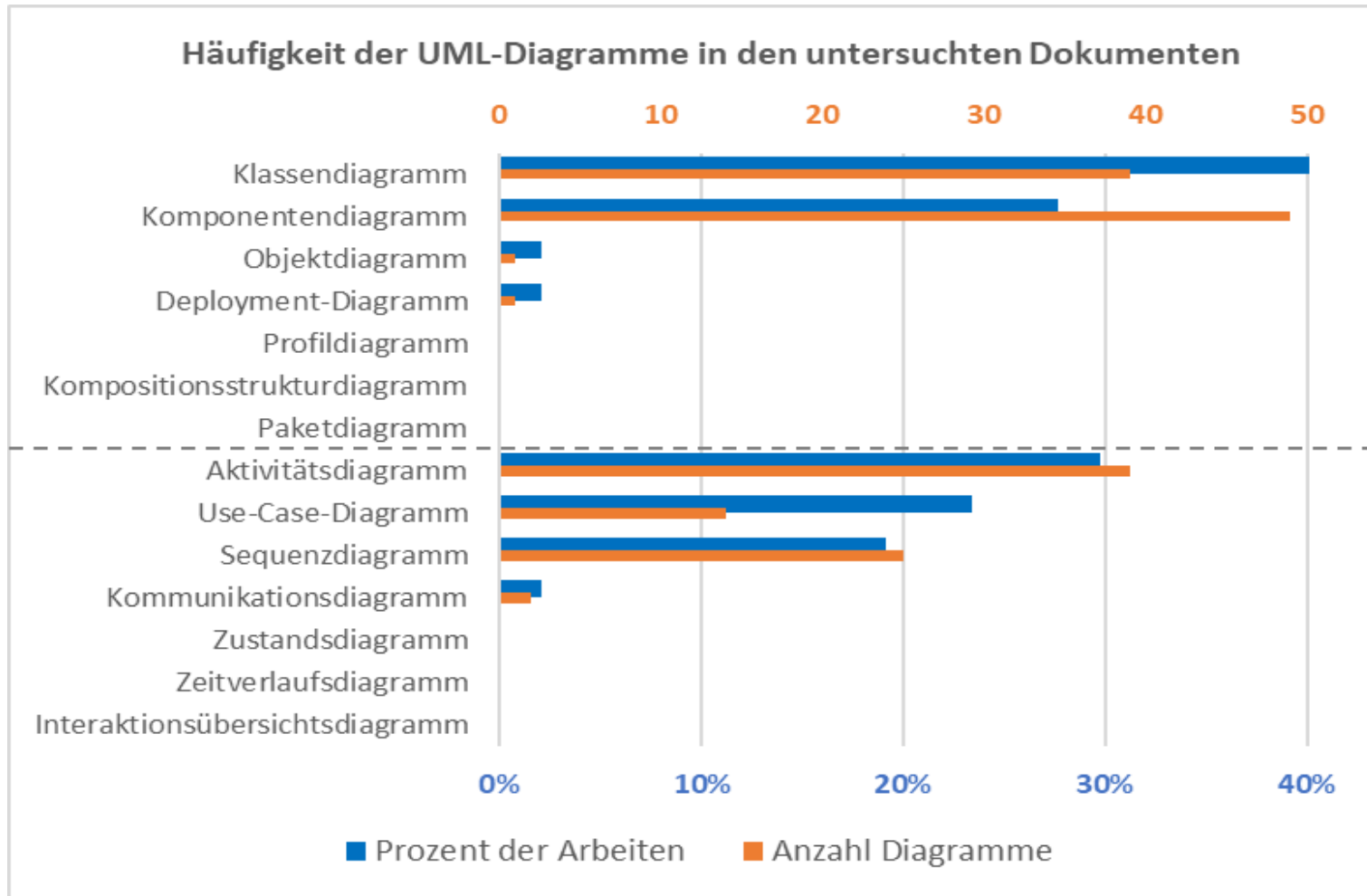
Untersuchung von Hochschul-Abschlussarbeiten

Analyse von Abschlussarbeiten / Projektberichte

- 47 Abschlussarbeiten und Projektberichte
 - Informatik, Medien- und Wirtschaftsinformatik (BA/MA) an der TH Köln
- Erstellung eines Softwaresystems als (Teil-)Schwerpunkt
- Nicht älter als fünf Jahre
- Keinerlei Vorgaben seitens der Betreuer bezüglich Modellierung
- **Verwendete Diagramme:**
 - Zählung
 - Klassifikation nach Diagrammtyp
 - Klassifikation nach Einsatzkontext

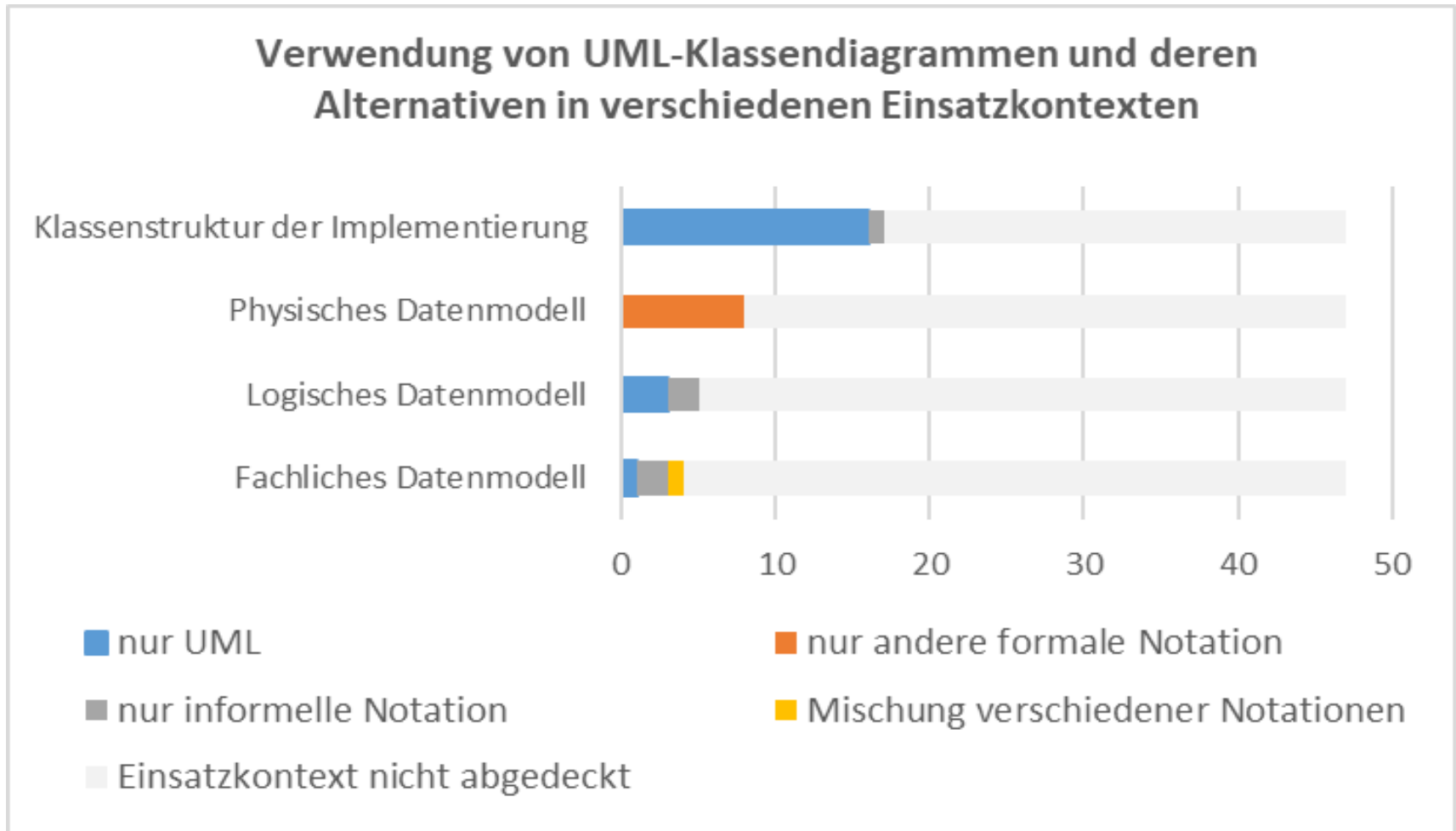


Verwendete Diagrammtypen



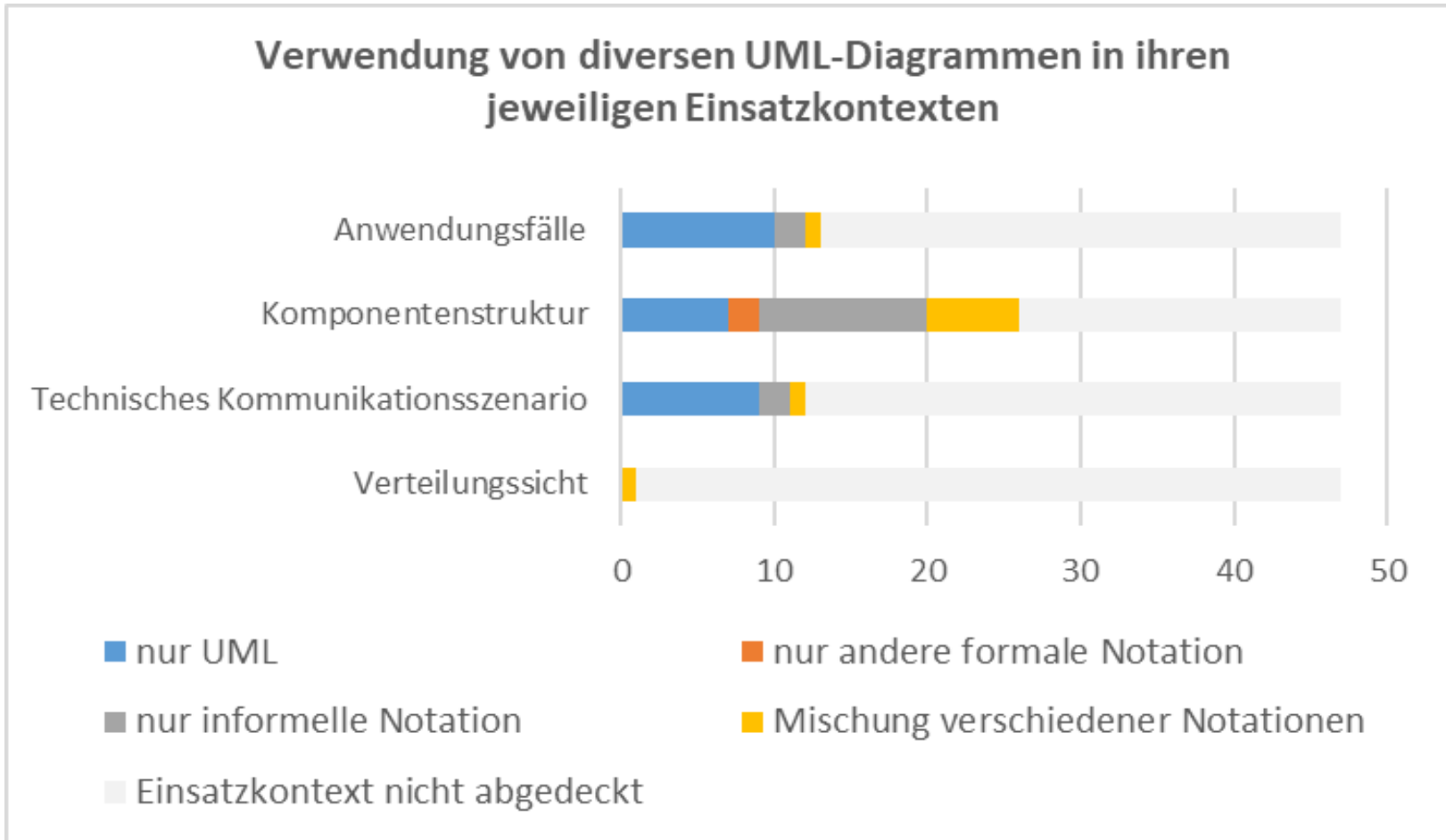
a. Nur ein Subset der UML-Spezifikation ist für studentische Arbeiten relevant.

UML vs. formale / informelle Alternativen (1)



b. Potential für Modellierung fachlicher Aspekte nicht ausgeschöpft.
c. Andere formale Notationen als UML haben ihren Platz.

UML vs. formale / informelle Alternativen (2)



d. Informelle Notationen haben ebenfalls ihren Platz
e. In bestimmten Fällen ist UML „alternativlos“

Sieben Thesen zur Verwendung von UML in der Lehre

These 1 - 2

1. UML-Kompetenz sollte in der Softwaretechnikausbildung weiterhin vermittelt werden.
 - Modellbildende Abstraktion ist eine grundlegende Querschnittskompetenz in der Informatik
 - Einige UML-Diagrammtypen intuitiv verständlich und gut merkbar

2. Die Lehre sollte sich auf ausgewählte UML-Modellelemente fokussieren.
 - *Insbesondere:* Klassen-, Use-Case und Sequenz-Diagramme
 - *Mit weniger Fokus auf Formalitäten:*
 - Komponenten-,
 - Deployment-,
 - Aktivitäts-,
 - Zustands- und
 - Kommunikationsdiagramme

These 3

3. Die Lehre sollte sich auf die Beherrschung von Einsatzkontexten anstatt von UML-Diagrammtypen fokussieren.

UML-Diagramm (Bsp.)	Einsatzkontext (Bsp.)
Klassendiagramm	Fachliches Datenmodell / Domänenmodell
	Logisches Datenmodell
Aktivitätsdiagramm	Nutzungsszenario (Blackbox)
	Geschäftsprozess (Whitebox)
Zustandsdiagramm	Fachliche Zustände von Geschäftsobjekten
	Technische Zustandsfolge (State Machine)

These 4 - 5

4. Die Fähigkeit zur Auswahl von Modellierungsansätzen sollte gestärkt werden.
 - UML vs. andere formale Modellierungssprachen
 - Formal vs. informell

5. Die Modellierungskompetenz sollte auf das Absolventenprofil abgestimmt sein.

Studiengang	Berufsbild (Bsp.)	Kernkompetenz (Bsp.)
Reine Informatik	Software-Architektur	sicherer Umgang mit Modellierung und Metaebenen
Medien- / Wirtschafts-informatik	Anwendung	Fähigkeit, pragmatisch passende Notation zu wählen
Technische Informatik	Entwicklung von Echtzeitsystemen	Modelle für formale Verifikation und Codegenerierung

These 6 - 7

6. UML-Ausbildung sollte die Modellierung von Fachlichkeit stärker fokussieren.

- *Klassendiagramm*: Fachliches DM / Domänenmodell
 - Domain-Driven Design
- *Zustandsdiagramm*: Geschäftsobjekt-Zustände
 - fachliche Events im System
 - Design von REST-APIs gemäß CQRS

7. Fokus der Lehre sollte auf Kommunikation und Interaktivität liegen, nicht auf Tools.

- Software-Erstellung ist Teamwork
- „Whiteboard over UML-Tool“

Ausblick

- Untersuchung weiter vertiefen
 - Wofür wird UML wie eingesetzt?
 - Welche Alternativen gibt es?
- Ergebnisse stärker systematisieren
 - Zusammenhänge zwischen Softwareproduktart, Vorgehensmodell, Projektumfang und Einsatz der Modellierung herstellen
 - Untersuchungen zum Tool-Einsatz und Lebenszyklus der Modell-Artefakte
- Kompetenzziele für die Hochschullehre ableiten
 - Differenzierung nach Studiengang und Ausbildungsstufe (Bachelor/Master)
 - Berücksichtigung einschlägiger Empfehlungen (z.B. GI)
- Entwicklung von didaktischen Konzepten
 - Was sind geeignete Formen der Wissensvermittlung bzw. interaktiven Wissenerschließung für Modellierung?
 - Welche Prüfungsformen sind geeignet?

**Vielen Dank für die
Aufmerksamkeit!**

Backup

Abdeckung von Einsatzkontexten (*formal + informell*)

