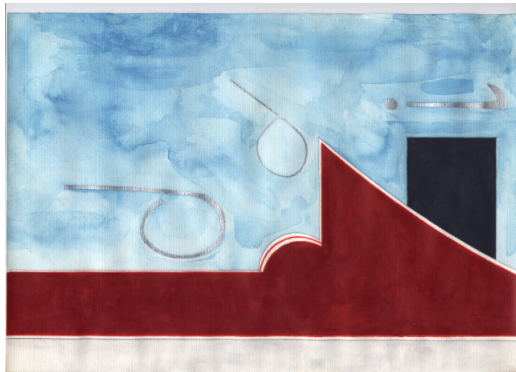


Didaktik der Informatik – Vorlesung

8. Vorlesung: Informatikunterrichtsplanung – Vorgehensmodelle

StD Dipl.-Inform. Dr. rer. nat. L. Humbert

Fachgebiet Didaktik der Informatik
Fachbereich C – Universität Wuppertal



Gliederung der Präsentation

1 Übersicht

- Gliederung der Präsentation
- Themen der Vorlesung im Sommersemester 2009

2 Informatikunterrichtsplanung – Vorgehensmodelle

- Veranstaltungsziele – Kompetenzen
- Fachlich begründetes Vorgehen
⇒ Vorgehensmodelle
- Wasserfallmodell

• STEPS

- Extreme Programming (XP) – eine »agile« Methode
- Zwischenresüme – Vorgehensmodelle
- Bezug zu didaktischen Fragen

3 A Pedagogical Pattern Language

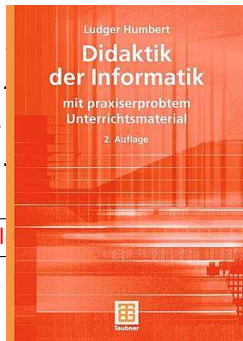
4 Unterrichtsplanungsmodelle

- Allgemein: König/Riedel – W. Schulz – W. Klafki
- Fachdidaktik – Hartmann
- Weiterentwicklung?

Themen der Vorlesung (Stand 14. Juni 2009)

Kapitel in [Humbert, 2006]

1	Organisatorisches – Einführung	2, 4
2	Informatik – geschichtliche Aspekte	2
3	Genderdiskussion	9
4	Grundfragen des Lernens	3
5	Schulinformatik – Entwicklungslinien	4
6	Schulinformatik – Normierung	4
7	Informatikunterricht – besondere Arbeitsweisen ..	5
8	Informatikunterrichtsplanung – Vorgehensmodell	6
9	Informatikunterrichtsplanung	6, 7
10	Informatikunterricht – Beispielszenarien	7
11	Informatikunterricht – Leistungsmessung	8
12	Moralisch-ethische Aspekte – Professionalisierung	9, 10



Kompetenzen 8. Vorlesung: Informatikunterrichtsplanung – Vorgehensmodelle

Kompetenzen 8. Vorlesung: Informatikunterrichtsplanung – Vorgehensmodelle

- Fachlich begründetes Vorgehen zur Planung von Vermittlungsprozessen darlegen und im Hinblick auf ihre Eignung für die Unterrichtsplanung einschätzen
- Mindestens drei Planungs-/Vorgehensmodelle angeben, darstellen und hinsichtlich der Vor- und Nachteile beurteilen
- Eignung der »Pedagogical Pattern Language« für Vermittlungsprozesse einordnen
- Bekannte allgemeine Unterrichtsplanungsinstrumente einordnen

Vorgehensmodelle zur Erstellung von Informatiksystemen

Zielmaßgaben aller Vorgehensmodelle (zur Softwareentwicklung)

- Effiziente Entwicklung . . .
- qualitativ hochwertiger Software . . .
- unter Einhaltung von Zeit- und Kostenbudgets

Auswahl und Kurzdarstellung einiger Vorgehensmodelle

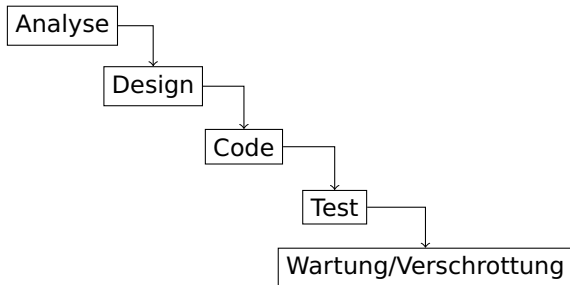
- Wasserfallmodell (70er Jahre)
- STEPS (90er Jahre)
- XP (aktuell)

nicht dargestellt V-Modell, aktuelle Ansätze zur theoretischen Fassung, RUP, Metamodellierung, allgemeine Modelltheorie, . . .

Wasserfallmodell – Produkterstellungsorientierung

- Fertigungsprozess für Produkte als erfolgreiches Vorbild für Software-Entwicklung
- starke Bürokratisierung

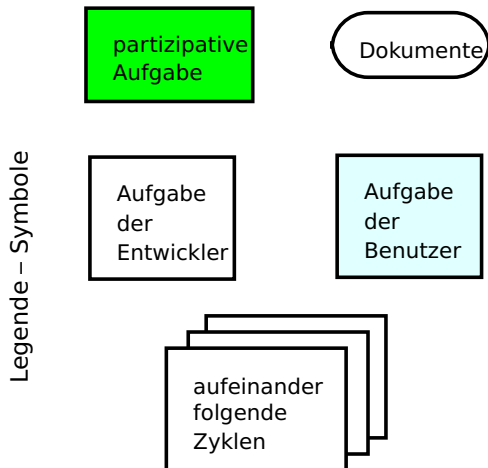
↔ »Ungeliebtes« Modell der Softwareentwicklung



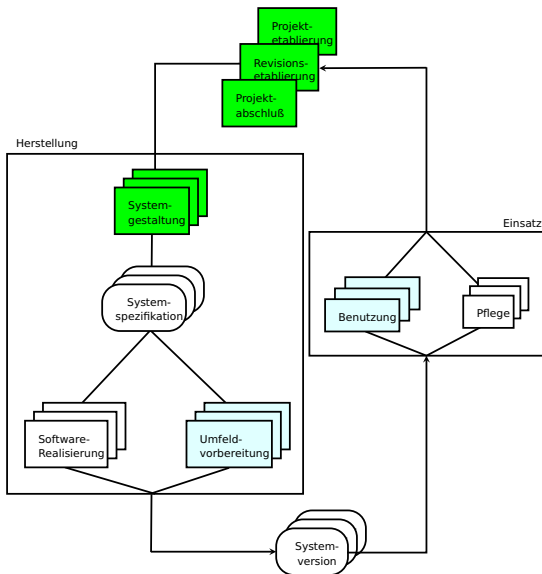
Beachte: Bereits in [Royce, 1970, 330] finden sich Hinweise auf ein iteratives Vorgehen (mit Rückwärtspfeilen).

STEPS – iterativ – [Floyd, 1993]

Software-Technik für Evolutionäre Partizipative Systementwicklung



STEPS – [Floyd, 1993]



vgl. [Pasch, 1994, S. 63]

Extreme Programming (XP) – Hacken als Modell?

Mittel

- Werte
- Rollen
- Prinzipien
- Aktivitäten

»Werte«

- Kommunikation
- Einfachheit
- Feedback
- Eigenverantwortung

Rollen

Projektleiter Management, Koordination (Ressourcen, Kosten, Zeitpläne)

Kunde wenigstens ein Kunde ist permanent ansprechbar – entwirft funktionale Tests für die Software (User-Stories)

Entwickler kodieren, testen, entwerfen und hören dem Kunden aufmerksam zu

Prinzipien

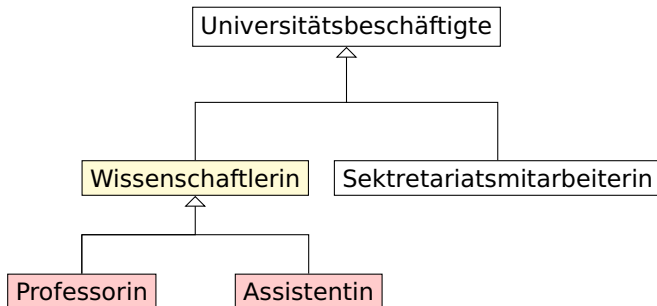
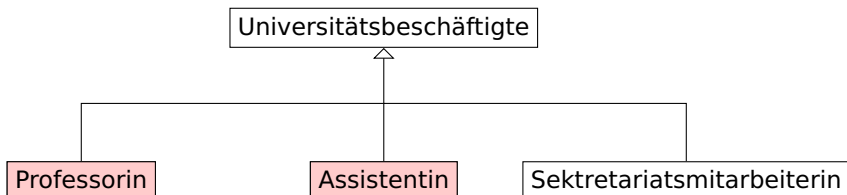


Extreme Programming – Entwicklungspraktiken

- Planspiel
- Metaphern: Unterstützung der Kommunikation
- Pair-Programming
- Testen, testen, testen
- Refactoring
- Gemeinsamer Codebesitz
- Kleine Freigaben – idealerweise im Wochentakt
- Kontinuierliche Integration
- Max. 40 Stunden Woche
- Kodierungsstandards

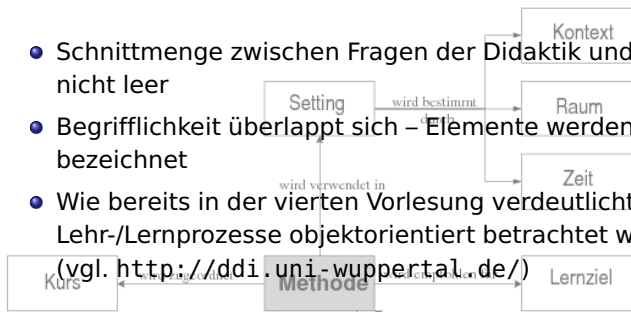
Grundlage für die Darstellung von XP – [Rumpe, 2001]

XP: Refactoring – Verbesserung der Qualität



Zwischenresümee – Vorgehensmodelle

- Schnittmenge zwischen Fragen der Didaktik und Vorgehensmodellen ist nicht leer
- Begrifflichkeit überlappt sich – Elemente werden gleich oder ähnlich bezeichnet
- Wie bereits in der vierten Vorlesung verdeutlicht, können Lehr-/Lernprozesse objektorientiert betrachtet werden



(vgl. <http://ddi.uni-wuppertal.de/>)

Erklärungsansatz

In beiden Bereichen (also Software-Entwicklung und organisierten Lehr-/Lernprozessen) geht es darum, mit Menschen komplexe Situationen zu bewältigen – dabei kommt nicht formalisierbaren Elementen häufig eine Schlüsselfunktion zu – ob das die Organisatoren nun wollen oder nicht

Hinweise: Wasserfall – [Boehm, 2002], [Boehm, 1984]

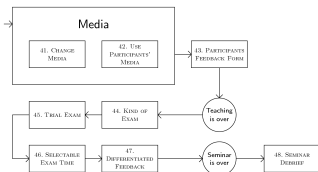
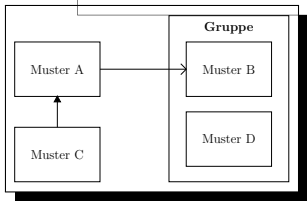
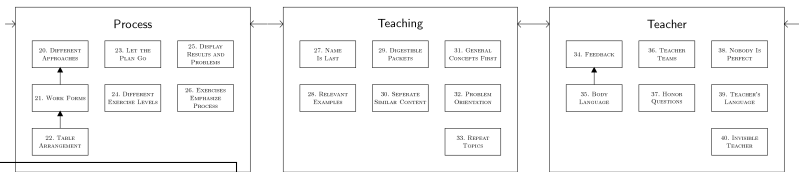
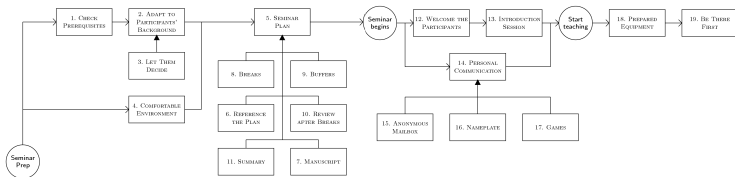
Bezug zu didaktischen Fragen

- Organisierte Lehr-/Lernprozesse finden nicht nur in der Schule statt
- Einige Entwicklerinnen und Entwicklern aus der OO-Szene haben ihre Modellierungskompetenz zur
 - Vorbereitung
 - Durchführung
 - Nachbereitung

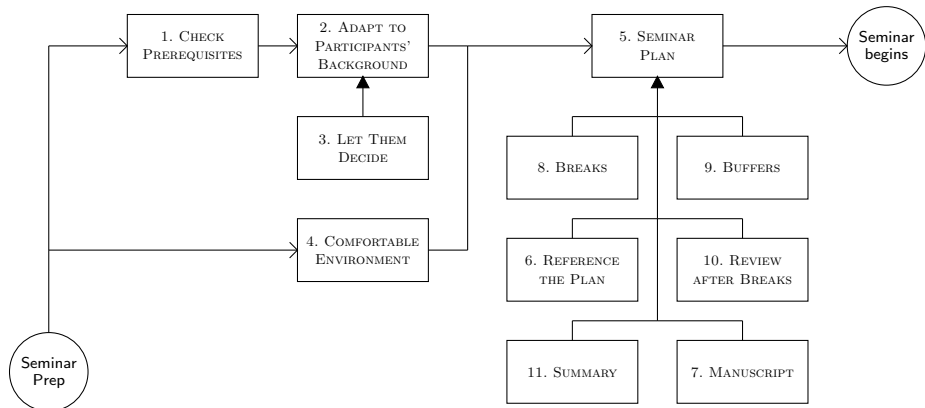
von Seminaren eingesetzt (vgl. [Fricke u. Völter, 2000])

Die Ergebnisse bieten zur Bewältigung dieser Art von Planungsaufgaben strukturierte Unterstützung

Übersicht – Legende



Teaching seminars effectively



Zieldimension

Die Modellierung erfüllt verschiedene Ziele, **ein** Ziel besteht darin, Fragen beantworten zu können, die sich im Zusammenhang mit Problemen ergeben. Ein Beispiel wird von den Autorinnen/Autoren folgendermaßen formuliert (aus [Fricke u. Völter, 2000, S. 8]):

typical problem

My sessions are boring, I do not feel I can engage the participants.

patterns in this language

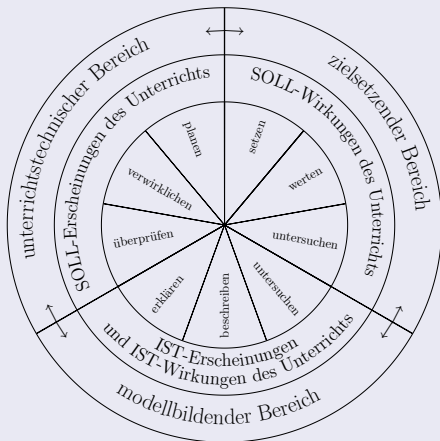
change media (41), body language (35), problem orientation (32), relevant examples (28), adapt to participants' background (2), reference the plan (6)

- Aus didaktischer Sicht ist zu bemerken, dass die Ziele des Vermittlungsprozesses nicht berücksichtigt werden.

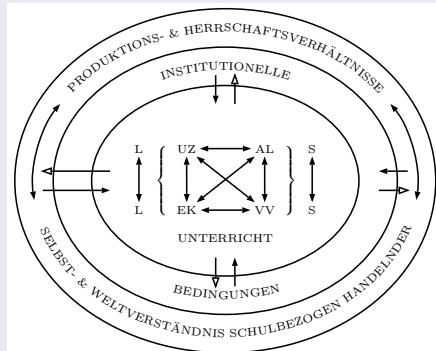
Muster zur didaktischen Planung

Didaktische Planungsmodelle – Beispiele ...

Systemtheoretische Didaktik



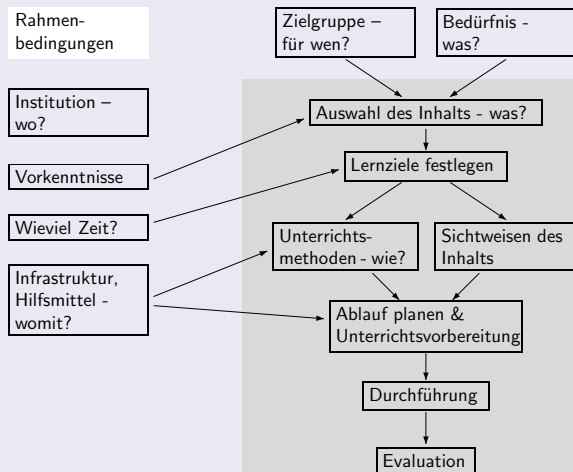
Berliner/Hamburger Modell



[Schulz, 1981, S. 82]

Fachdidaktik – Planungsmodell[e]

Bedingungsgefüge Informatikunterrichtsplanung – Werner Hartmann



aus: [Humbert, 2006, S. 97]

Planungsmodelle – Kritik und Weiterentwicklung

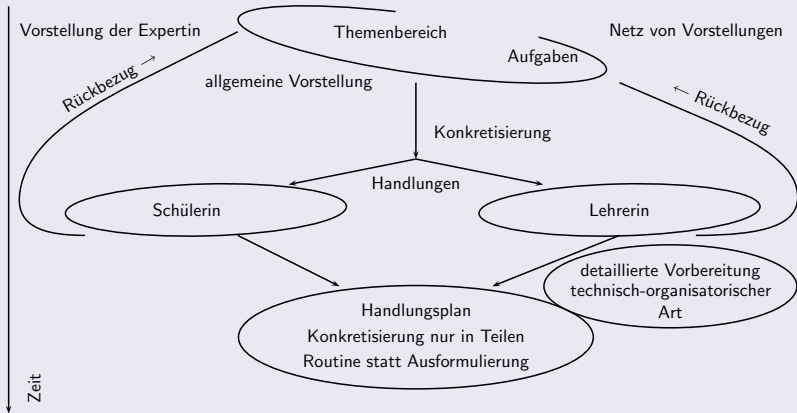
- Planungsmodelle für den Unterricht – Inputorientierung
- Vollständige Planbarkeit des Unterrichts ist eine Chimäre
- Berücksichtigung der professionellen Unterrichtsvorbereitung findet kaum statt
- Aktuell werden eher Elemente benannt, die [nur] ausgewählte Planungsmomente betreffen

Beispiele

- TPS – Think, Pair, Share (kooperatives Lernen)
- Klippert

Planung – professionell

Unterrichtsplanung von Expertinnen des Unterrichts (schematisch)



aus: [Humbert, 2006, S. 96]

Literatur

- [Boehm 1984] Boehm, Barry: Software Engineering Economics. In: **[Broy u. Denert, 2002]**, S. 641–686. – zuerst veröffentlicht in: IEEE Transactions on Software Engineering, Vol. SE-10(1), pp. 4-21, 1984
- [Boehm 2002] Boehm, Barry: Early Experiences in Software Economics. In: **[Broy u. Denert, 2002]**, S. 632–640. – sd&m Konferenz, 28., 29. Juni 2001
- [Broy u. Denert 2002] Broy, Manfred (Hrsg.) ; Denert, Ernst (Hrsg.): *Software Pioneers Contributions to Software Engineering*. Berlin : Springer, 2002 . – ISBN 3-540-43081-4. – sd&m Konferenz, 28., 29. Juni 2001
- [Floyd 1993] Floyd, Christiane: STEPS – a methodical approach to PD (Participatory Design). In: *Comm. ACM* 36 (1993), June, Nr. 6, S. pp. 83–85
- [Fricke u. Völter 2000] Fricke, Astrid ; Völter, Markus: *SEMINARS – A Pedagogical Pattern Language about teaching seminars effectively*. July 2000. – <http://www.voelter.de/data/pub/tp/tp.pdf>, <http://www.voelter.de/data/pub/tp/html/> and <http://www.coldewey.com/europlop2000/papers/voelter+fricke.zip> – last visited 29th April 2008
- [Humbert 2006] Humbert, Ludger: *Didaktik der Informatik – mit praxiserprobtem Unterrichtsmaterial*. 2., überarbeitete und erweiterte Aufl. Wiesbaden : B.G. Teubner Verlag, 2006 (Leitfäden der Informatik). – ISBN 3-8351-0112-9. – <http://humbert.in.hagen.de/ddi/> – geprüft: 8. März 2009
- [Klafki 1985] Klafki, Wolfgang: *Neue Studien zur Bildungstheorie und Didaktik: Beiträge zur kritisch-konstruktiven Didaktik*. Weinheim, Basel : Beltz Verlag, 1985. – ISBN 3-407-54148-1
- [König u. Riedel 1973] König, Ernst ; Riedel, Harald: *Systemtheoretische Didaktik*. Weinheim und Basel : Beltz, 1973. – ISBN 3-4075-4001-9
- [Pasch 1994] Pasch, Jürgen: *Software-Entwicklung im Team*. Berlin : Springer Verlag, 1994. – ISBN 3-540-57228-7

Literatur (cont.)

- [Royce 1970] Royce, Winston W.: Managing the Development of Large Software Systems. In: *Proceedings, WESCON*, TRW (The Institute of Electrical and Electronics Engineers Inc.—IEEE), August 1970, S. 1–9 (328–338). – <http://facweb.cs.depaul.edu/jhuang/is553/Royce.pdf> and <http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf> – last visited 19th June 2009
- [Rumpe 2001] Rumpe, Bernhard: Extreme Programming – Back to Basics? In: Engels, Georg (Hrsg.) ; Oberweis, Andreas (Hrsg.) ; Zündorf, Albert (Hrsg.): *Modellierung 2001, Workshop der Gesellschaft für Informatik e. V. (GI) 28.–30.3.2001, Bad Lippspringe*. Bonn : Gesellschaft für Informatik, Köllen Druck + Verlag GmbH, März 2001 (GI-Edition – Lecture Notes in Informatics – Proceedings), S. 121–131. – <http://www4.in.tum.de/~rumpe/papers/Rum01/Rum01.pdf> – geprüft: 25. Mai 2008
- [Schulz 1981] Schulz, Wolfgang: *Unterrichtsplanung. Mit Materialien aus Unterrichtsfächern*. Beltz, 1981 (Fachbuch). – ISBN 3–407–26016–4