Seminar "Ausgew. Beiträge zum SW-Engineering"

Parameter für Softwarepraktika

Dr. Dirk Draheim, Prof. Dr. Lutz Prechelt
Freie Universität Berlin, Institut für Informatik
http://www.inf.fu-berlin.de/inst/ag-se/

- Lernziele
  - Hard skills, soft skills
- Chancen und Risiken
- Mögliche Ziele:
  - Methodik
  - Technologie
  - Meta-Lernen
- Parameter:
  - Organisationsform
  - Inhalt
  - Betreuung

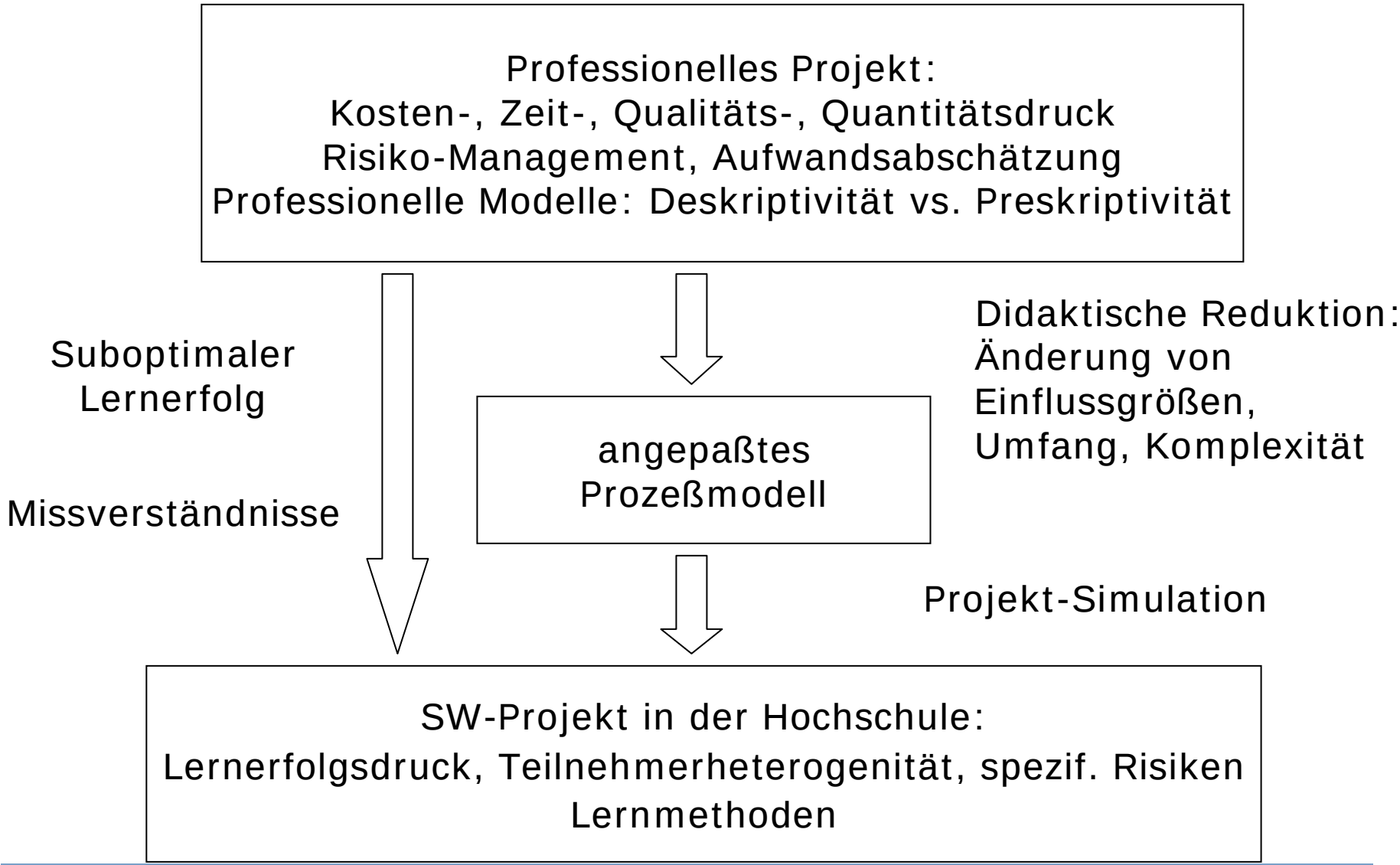
Zweidimensionale Lernziel-Taxonomie des Software-Engineering

- Funktionale Qualifikation (Hard Skills)
  - Niedriges Niveau: konkretes Detailwissen
    - bzgl. Technologie: Programmiersprachen, APIs, Betriebssysteme, CASE Tools, usw.
    - bzgl. Methoden: Spezifikationssprachen, Modellierungssprachen, Prozesse, Entwurfsmuster,usw.
  - Hohes Niveau: Einschätzung von Vor- und Nachteilen verschiedener Technologien, relativ zu konkretem Projekt, ohne Wissensangst
- Extrafunktionale Qualifikation (Soft Skills)
    - Soziale Kompetenz, Führungsstärke, Entscheidungsfreude Arbeitsdisziplin, Motivation, Moderation
- Schlüsselqualifikation (Key Skills)

Risiken und Chancen der SE Ausbildung

- Risiken
  - Studenten, allgemein: Unterschiedliche Anfangsqualifikation, Talente, Zielsetzungen und Wahrnehmung
  - Studenten, Informatik: Real Programmers, Baby Duck Syndrome
  - Dozent: Zu starke Vereinfachung von Lehrinhalten, zu große Präskriptivität
- Chancen
  - Grundannahme: Studenten sind aufnahmefähig und motiviert
  - Teilnehmerheterogenität bzgl. Anfangsvoraussetzungen kann sich positiv auf Gruppenarbeit und gemeinschaftliches Lernen auswirken

Einsatz eines professionellen Prozeßmodells als Risiko



Thesen

- Lernmethode follows Lernzielbestimmung
- Methode muss Risiken/Chancen von Erwachsenenbildung und SE Ausbildung berücksichtigen

Dirk Draheim and Gerald Weber. Co-Knowledge Acquisition of Software Organizations and Academy. In: Proceedings of LSO 2004 - 6th International Workshop on Learning Software Organisations.

Dirk Draheim. A CSCW and Project Management Tool for Learning Software Engineering. In: Proceedings of FIE 2003 - Frontiers in Education: Engineering as a Human Endeavor. IEEE Press, 2003.

Dirk Draheim. Ein kombiniertes CSCW-/Projektmanagementwerkzeug für den Softwareprozeß EASE. In: GML 2003 - 1. Workshop "Grundfragen multimedialer Lehre"

Raymond Morel, Dirk Draheim, Marc Pilloud and Muddassar Farooq. Recommendations of the Working Group on Social Issues and Powershifts at SECIII. In (Tom J. van Weert and Robert K. Munro, Editors): Informatics and the Digital Society. Kluwer Academic Publishers, January 2003.

Dirk Draheim. Learning Software Engineering with EASE. In (Tom J. van Weert and Robert K. Munro, Editors): Informatics and the Digital Society. Kluwer Academic Publishers, January 2003.

Mögliche Ziele für SW-Praktika

- Methodik:
  - Anforderungsbestimmung erlernen
  - Entwurf und Schnittstellendefinition erlernen
  - Programmierpraxis erhöhen
  - Testmethodik/Qualitätssicherung erlernen
  - Programmverstehen einüben
  - Teamarbeit/Projektmanagement erlernen/einüben
  - Prozess-/Qualitätsmanagement erlernen
- Technologie:
  - Technologiewissen vergrößern
- Meta-Lernen:
  - Erlernen von Technologie erlernen
  - Erlernen von Methoden erlernen

Parameter zur Konzeption von SW-Praktika

- Organisationsform
  - Gruppengröße
  - Organisationsform/Rollen (Ausdifferenzierung)
  - Aufgabenbreite Einzelperson
- Inhalt
  - Art der Aufgabenstellung
  - Umfang der Aufgabenstellung
  - Katalog der Ergebnisansprüche
  - Niveau der Ergebnisansprüche
- Betreuung
  - Vorstrukturierung der Aufgabe
  - Art und Intensität der Betreuung unterwegs

Parametergruppe Organisationsform

Gruppengröße

- Wie viele Personen bilden ein Team?
  - meist 4 bis 20
- Vorteile großer Gruppen:
  - Umfangreichere Projekte möglich
  - Stärkere Differenzierung der Rollen möglich
  - Personalausfälle leichter zu verkraften
- Nachteile großer Gruppen:
  - Koordination ist schwierig und verschlingt viel Kapazität

Organisationsform/Rollen (Ausdifferenzierung)

Jeweils von außen (durch die Veranstalter):

- Wie stark werden Rollen wie Entwerfer, Programmierer, Tester, Projektmanager usw. ausdifferenziert?
- Wie viel formale Kompetenzen (Autorität) werden diesen Rollen zugewiesen?
  - Wie werden diese durchgesetzt?

- Vorteil starker Ausdifferenzierung:
  - Softwareprozess (und dessen Wirkung) wird besser sichtbar
  - Evtl. höhere Prozessdisziplin und bessere Ergebnisqualität
- Nachteil starker Ausdifferenzierung:
  - Erzeugt Mehraufwand und (da künstlich) Mehrkonflikt

Aufgabenbreite Einzelperson

- Wie sehr rotieren die Rollen im Laufe des Praktikums?
  - d.h. wie viele verschiedene Rollen bekomme ich nacheinander mal für gewisse Zeit zugewiesen?
- Vorteil höherer Rotation:
  - Mehr Einblick in verschiedene Rollen
- Nachteil höherer Rotation:
  - Mehraufwand
  - Erhöhte Neigung zu Chaos und Anarchie

Parametergruppe Inhalt

Art der Aufgabenstellung

Teilaspekte:

- Volle oder eingeschränkte Breite der Tätigkeiten
  - z.B. Anforderungsbestimmung, Spezifikation, Architekturentwurf, Testautomatisierung, Dokumentation, Benutzbarkeitstests, Einführung, etc.
  - Breite vs. Tiefe
- Programmierlastigkeit
  - Fragwürdig(?), aber schwer zu vermeiden
- Von Null auf oder aufbauend auf existierendes Produkt
  - von Null: hohe Kontrolle; aufbauend: größeres System
- Sex-Appeal
- Allgemeinverständliche oder spezielle Domäne
  - Allgemein: einfacher Einstieg; speziell: interessanter?

Umfang der Aufgabenstellung

- Ist die Aufgabe im Umfang groß, klein oder variabel angelegt?

- Vorteil kleiner Aufgabenstellungen:
  - Hohe Qualität(sansprüche) möglich
  - Volle Abdeckung der Tätigkeitsbreite wird realistischer
  - Frustration wird weniger wahrscheinlich
- Nachteil kleiner Aufgabenstellungen:
  - Evtl. wenig motivierend
  - Lösung durch Einzelkämpfer wird realistischer

Katalog der Ergebnisansprüche

- Werden alle Ergebnisarten, die im Prinzip zur Aufgabe gehören auch tatsächlich (gleichmäßig) eingefordert?
  - z.B. nur lauffähiger Code oder auch: sauberer Entwurf, Tests, Testberichte, Entwurfsdoku, Benutzerdoku, etc.

- Vorteil breiter Ergebnisansprüche:
  - Förderung der Qualitätsorientierung
  - Breiteres Einüben von Methoden
  - Höherer Macherstolz der Teilnehmenden
- Nachteil breiter Ergebnisansprüche:
  - Realistischer Projektumfang sinkt

Niveau der Ergebnisansprüche

- Wird für die verlangten Ergebnisse auch eine hohe Qualität gefordert?
  - z.B. bezüglich Testabdeckung, Versagen bei Spontantests, Einhaltung von Entwurfsrichtlinien, etc.
- Vorteil hoher Ergebnisansprüche:
  - Förderung der Qualitätsorientierung
  - Höherer Macherstolz der Teilnehmenden (bei Erfolg)
- Nachteil hoher Ergebnisansprüche:
  - Hoher Betreuungsaufwand
  - Hohe Durchfallquote

Parametergruppe Betreuung

Vorstrukturierung der Aufgabe

- Wie viele Hilfestellungen und Hinweise liefern die Veranstalter von vornherein mit?

- Vorteil starker Vorstrukturierung:
  - Viele Frustrationserlebnisse werden vermieden
  - Höherer Umfang und höhere Qualität werden möglich
  - Evtl. größerer Lerneffekt durch Konzentration auf das Wesentliche
- Nachteil starker Vorstrukturierung:
  - Sehr hoher Vorbereitungsaufwand
  - Unrealistische Praktikumssituation
  - Evtl. geringerer Lerneffekt

Art und Intensität der Betreuung unterwegs

- Wie intensiv und wie autoritär geben die Betreuer unterwegs Hilfestellung?

- Vorteile starker Betreuung:
  - Vermeidet evtl. Frustration
  - Höhere Qualitätsansprüche werden möglich
  - Evtl. hilfreich für Lerneffekt
- Nachteile starker Betreuung:
  - Hoher Aufwand für die Veranstalter
  - Evtl. schädlich für Selbstvertrauen
  - Evtl. schädlich für Lerneffekt

Diskussion

Fragen an das Publikum:

- Womit gibt es konkrete Erfahrungen?
  - Positive oder negative
- Welche Entscheidungen erscheinen zwingend?
  - Weil das Praktikum sonst überhaupt nicht richtig funktioniert
- Welche Präferenzen gibt es beim Rest?
  - Und womit begründen sich diese?

Danke!