

# Softwareprojekte im Informatikstudium

Karsten Weicker

HTWK Leipzig, Fakultät für Informatik, Mathematik und Naturwissenschaften

24. Oktober 2014

# Motivation

## Gewünschte Kompetenzen

Wie viele Firmen erwarten von Informatik-Absolventen Zusatzqualifikationen

- ... in Form von Praktika? 93%
- ... in Form von Berufspraxis? 62%

(Job-Trends-Studie, Staufenbiel, 2008)

# Motivation

## Gewünschte Kompetenzen

Wie viele Firmen erwarten von Informatik-Absolventen Zusatzqualifikationen

- ... in Form von Praktika? 93%
- ... in Form von Berufspraxis? 62%

(Job-Trends-Studie, Staufenbiel, 2008)

## Rolle der Hochschule

- Softwareprojekte legen die Grundlage
- in Hochschulprojekten darf man schadlos scheitern

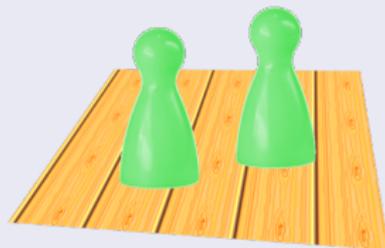
# Überblick

- 1 Entwicklung der Softwarepraktika
- 2 Kompetenzvermittlung
- 3 Agile Projekte
- 4 Fazit

# Entwicklung der Softwarepraktika

## 1991: Kleinst-Teams

- 2- oder 3-Personen-Teams, manchmal: 1 Person
- selbstgewähltes Thema
- eigentlich nur größere Programmieraufgabe
- Wasserfall
- Entwurfsdokumentation
- noch 2005 an HTWK



# Entwicklung der Softwarepraktika

## 1994: Projektgruppe

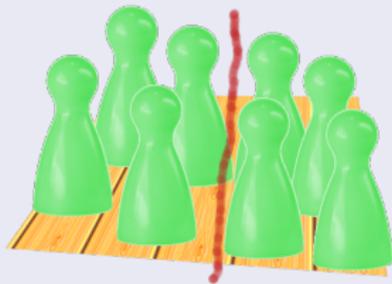
- größeres Team (8–12 Studierende)
- selbstorganisiertes Projektmanagement
- große, häufig offen formulierte Aufgabe
- Dauer: zwei Semester
- begleitend: Vorlesung, Seminar
- seit 1972 (Uni Dortmund)



# Entwicklung der Softwarepraktika

## 2000: Studienprojekte

- Weiterentwicklung der Projektgruppen für den Studiengang “Softwaretechnik” an der Uni Stuttgart
- mehr Vorgaben hinsichtlich Management
- z.T. gemeinsam mit anderen Fakultäten
- konkurrierende Angebote





# Entwicklung der Softwarepraktika

## 2006: gemischte Teams

- Übergang an der HTWK zu größeren Teams
- vorgegebene Projektthemen mit Kunde
- gemischte Teams aus den Bachelor-Studiengängen
  - Informatik
  - Medieninformatik



# Entwicklung der Softwarepraktika

## Beobachtungen

- viele Projekte verlaufen eher unorganisiert
- Projektleiter wird als *primus inter pares* nicht ernst genommen
- Wasserfall-Denken
- einzelne engagierte Studierende “retten”  
Projekt im Alleingang

# Entwicklung der Softwarepraktika

## 2007: realistische Hierarchien

- Kopplung zweier Lehrveranstaltungen:
  - Softwareprojekt für Bachelorstudiengänge
  - Projektmanagementpraktikum für den Masterstudiengang Informatik
- Master übernehmen die Projektleitung
- aus dem Unified Process abgeleitete konkrete Meilensteine



# Bewertung des Kompetenzerwerbs

## Grundlage

- Fragebögen zur Selbsteinschätzung
  - des eigenen Lernerfolgs und
  - der Kompetenz vor dem Projekt (erfragt vor Beginn und nach Abschluss des Projekts)
- 2008: 50 Bachelor, 10 Master
- 2007: 36 Bachelor, 12 Master
- (2006: 44 Bachelor)

# Ergebnisse – Lernerfolg (2006)

## Fachlich-methodische Kompetenzen I

- gut gelernt – gleichermaßen Geübtes:
  - Java
  - projektbegleitende Dokumentation
  - Schnittstellengestaltung
  - sichere Programmierkenntnisse
- gut gelernt – große projektinterne Auseinandersetzung:
  - Projektmanagement
  - GUI-Design
  - Schnittstellenarchitektur
  - Entwurfsmuster
  - ...

# Ergebnisse – Lernerfolg (2006)

## Fachlich-methodische Kompetenzen II

- schlecht gelernt – spezielle Aufgaben oder Vernachlässigtes:
  - Risikomanagement
  - Kostenschätzung
  - Kundengespräche
  - Unit-Tests
  - Konfigurationsmanagement
  - XML

# Ergebnisse – Lernerfolg (2006)

## Selbst-/Sozialkompetenzen

- sehr gut gelernt
  - selbständiges Arbeiten
  - Einarbeitungsbereitschaft
  - Teamfähigkeit
- schlecht gelernt
  - Präsentationskompetenz

# Korrektur der Selbsteinschätzung

## Bachelor (mit < 1% Fehler)

- GUI-Design
- Kundengespräche
- andere Programmiersprachen als Java
- Programmierrichtlinien
- Unit-Tests
- Abstraktionsvermögen
- sichere Programmierkenntnisse
- die Bereitschaft Werkzeuge zu nutzen
- selbständiges Arbeiten
- Neugier
- Bereitschaft, sich in Neues einzuarbeiten

# Korrektur der Selbsteinschätzung

## Master (mit $< 5\%$ Fehler)

- Projektmanagement
- Programmierrichtlinien
- Einfühlungsvermögen

# Änderungen zum alten Lehrkonzept (2006)

## Rahmenbedingungen ändern den Lernschwerpunkt

	2006	2007	2008
UML-Modellierung	2,62±0,85	2,33±0,89	2,31±0,83
Schichtenarchitektur	2,38±0,85	2,08±0,69	2,14±0,86
Versionskontrolle	2,50±0,90	2,06±0,95	2,33±0,94
Komm. mit Fachfremden	2,34±0,92	2,91±0,85	2,58±0,92

# Hauptprobleme im Projekt

	Befragung SS07		Befragung SS08	
	Bachelor	Master	Bachelor	Master
zu komplexe Aufgabe	33%	33%	14%	40%
Komm. mit dem Kunden	3%	8%	70%	60%
Komm. im Team	50%	67%	30%	20%
Schnittstellenprobleme	17%	33%	36%	40%
fehlendes techn. Know-How	75%	50%	50%	60%
fehlende Motivation im Team	33%	42%	22%	20%
mangelhaftes Projektmanag.	3%	33%	20%	20%

# Bewertung des Konzepts

## Rollenverhalten der Masterstudenten

- Prototypen: “freundlicher Kollege”, “Choleriker”, “formaler Prozessler” (Vischgenow/Schneider)
- Selbsteinschätzung des Führungsstils über Eigenschaften (ohne Kenntnis der Prototypen)
- Gesamtergebnis:
  - “freundlicher Kollege”: 44,2%
  - “Choleriker”: 31,8%
  - “formaler Prozessler”: 25,3%
- nur 9 der 22 Masterstudenten als “reiner” freundlicher Kollege

# Fazit zur aktuellen Form

## Zusammenfassung

- Sopra bildet Hierarchien aus der Arbeitswelt ab
- Akzeptanz bei den Studierenden
- professionelle Haltung bei Mastern
- praktische Projektmanagementausbildung, bei der alle Master aktiv sind
- Befragung stützt didaktischen Ansatz

# Agile Projekte

## Prinzipien

- inkrementelle Entwicklung mit kurzen Iterationen
- z.B. Scrum
- Einbindung des Kunden statt aufwendiger Anforderungsspezifikation als Vertragsgrundlage
- Programmieretechniken:
  - Test-First
  - Continuous Integration
  - Refactoring
  - "Keep it simple and stupid"

# Agile Projekte

## In Unternehmen

- 49% der schweizer IT-Unternehmen arbeitet u.a. agil
- in Deutschland: 25,3% aller Projekte

# Agile Projekte

## In Unternehmen

- 49% der schweizer IT-Unternehmen arbeitet u.a. agil
- in Deutschland: 25,3% aller Projekte

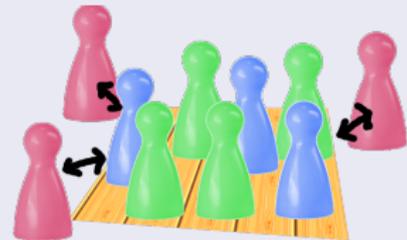
## An Hochschulen

- schwer in Stundenplan integrierbar
- allgemein vertretene Meinung:  
räumliche und zeitliche Nähe notwendig

# Agile Projekte

## Rollen der Master

- **Product-Owner:** vertritt den Kunden im Projekt – reine Anwendungssicht
- **Scrum-Master:** überwacht den Prozess, räumt Hindernisse aus dem Weg
- **Software-Architekt:** technischer Berater als Teil des Teams – Micromanagement und Software-Qualität



# Agile Projekte

## Benotung der Bachelorstudenten

Punkte	Tätigkeit	Indikator/Metrik
55	Programmierung	SVN-Auswertung, LOC, Umfang der bearbeiteten Tasks, Zeitaufwand, Inspektion einzelner Klassen
30	Dokumente	individuelle Beiträge zur Produktvision und zur Technologierecherche, Abschlussbericht, Dokumentation eines Sprints
30	Engagement	Beobachtung bei den Meetings, gesamter Zeitaufwand, Peer-Bewertung
10	Präsentationen	Umfang und Qualität der Präsentation der Zwischenstände
125	Summe	

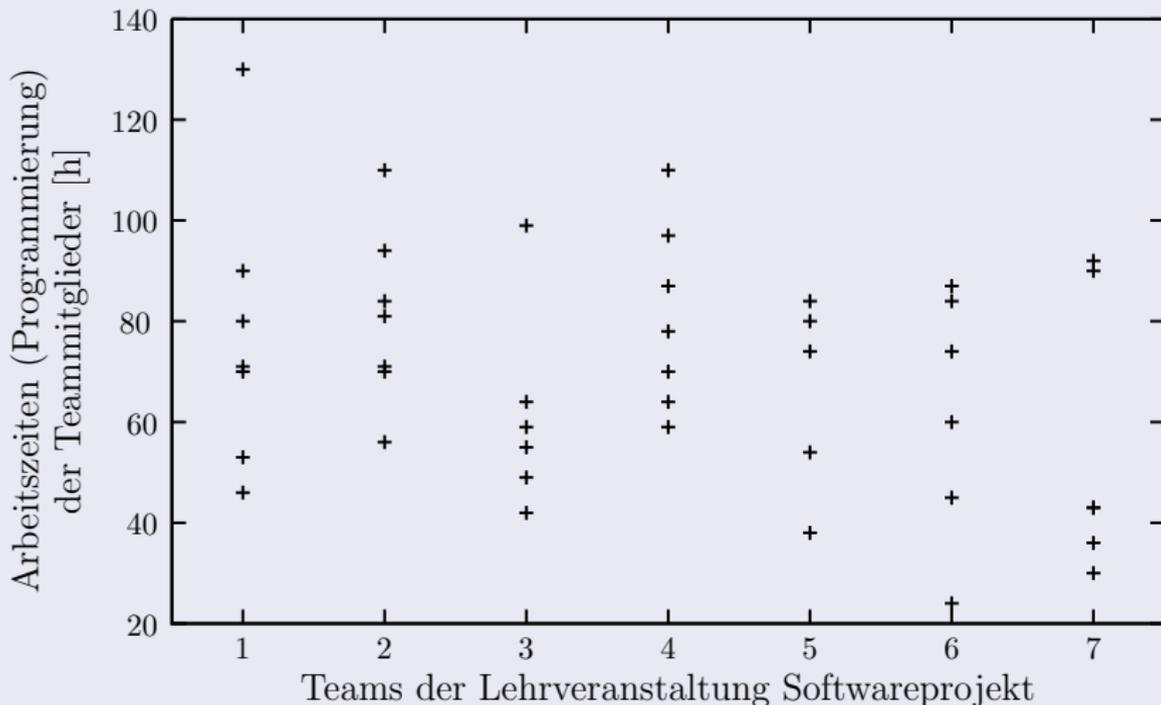
# Agile Projekte

## Benotung der Masterstudenten

50	Bericht	individueller Abschlussbericht mit Analyse des Gesamtprojekts
20	Dokumente	zusätzliche rollenspezifische Dokumente
20	Engagement	Betreuereindruck, Peer-Bewertung
15	Vortrag	Statusvortrag im Projektmanagementseminar
20	Produktbewertung	Bewertung des Kunden und des Betreuers, rollenspezifische Auswertung (Kundenzufriedenheit, Code-Qualität, Scrum-Prozess)
125	Summe	

# Agile Projekte

## Programmierarbeitszeiten



# Agile Projekte

## Zufriedenheit

	Zufriedenheit mit ...		
	der Zusam- menar- beit im Team	dem erstel- ten Produkt	dem eigenen Einfluss auf den Prozess
Bachelorstud.	1,6585	2,0731	1,9756
Masterstud.	1,6818	2,0909	2,1363

# Agile Projekte

## Wahrnehmung der Rolle "Product-Owner"

Schnittstelle zum Kunden	13
Anforderungsspezifikation	5
Leitung des Teams	5
Ansprechpartner/Helfer	5
Organisation+Kontrolle	3
sonstiges	3

# Agile Projekte

## Wahrnehmung der Rolle "Scrum-Master"

regelt Scrum-Prozess	8
leitet Meetings	7
Moderator	4
löst Probleme (Retrospektive)	4
fördert Team/Kommunikation	4
Organisation/Terminplanung	3
sonstiges	3

# Agile Projekte

## Wahrnehmung der Rolle “Software-Architekt”

Helfer bei Problemen	9
legt Architektur fest	8
negative Charakterisierung	6
wichtigster Problemlöser	4
Technologe	3
Schnittstelle Code/POs	2
Integrator	1
prüft Codqualität	1

# Fazit

## Kompetenz durch Praxis

- Untersuchungen zeigen den positiven Effekt
- Firmen bestätigen dies bei unseren Studenten im Praktikum
- Masterstudiengang durch Projektmanagement stark nachgefragt
- agile Softwareprojekte sind möglich und stärken das einzelne Teammitglied

# Fazit

## Feedback eines Masterstudenten

*“Nach diesen Bemerkungen [...] möchte ich noch einmal kurz herausstellen, dass sich das Softwareprojekt meiner Meinung nach sehr positiv im Laufe der Jahre entwickelt hat. Vor allem die Verwendung von Scrum [...] [kann] nicht genug belobigt werden.”*

# Literatur

- Nicole Weicker, Karsten Weicker: Analyse des Kompetenzerwerbs im Softwarepraktikum. In: Hochschuldidaktik der Informatik, HDI2008 – 3. Workshop des GI-Fachbereichs Ausbildung und Beruf/Didaktik der Informatik, Hrsg: Andreas Schwill, Potsdam: Universitätsverlag Potsdam, pp. 93-104, 2009.
- Karsten Weicker, Nicole Weicker: Von Häuptlingen und Indianern – Bachelor/Master als Chance. In: Software Engineering im Unterricht der Hochschulen SEUH 11 – Hannover 2009, Hrsg: Ulrike Jaeger, Kurt Schneider, Heidelberg: dpunkt.verlag, pp. 61-73, 2009.
- Christopher Hensel, Philipp Rabe, Benjamin Stahl, Karsten Weicker: Agile Softwareprojekte im Studium – ein Erfahrungsbericht. *to be submitted*