



# *Das Software-Projekt im Hochschulunterricht*

*- Lernziele, Einflußgrößen, Vorschläge -*

---

Informatik-Kolloquium, Universität Potsdam

19. Juni 2003

Dr. Dirk Draheim

Institut für Informatik, FU Berlin

[draheim@inf.fu-berlin.de](mailto:draheim@inf.fu-berlin.de)

# EASE und PEASE

<http://www.inf.fu-berlin.de/projects/ease/>

- EASE – Education for Actual Software-Engineering:
  - leichtgewichtiger Software-Prozeß
  - speziell auf Risiken und Chancen des Software-Projekts im Hochschulunterricht abgestimmt
- PEASE – Platform for EASE:
  - kombiniertes CSCW/Projektmanagement-Werkzeug
  - speziell abgestimmt auf den Software-Prozeß EASE

**Dirk Draheim. Learning Software Engineering with EASE. In: Informatics and the Digital Society. Kluwer Academic Publishers, 2003.**

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

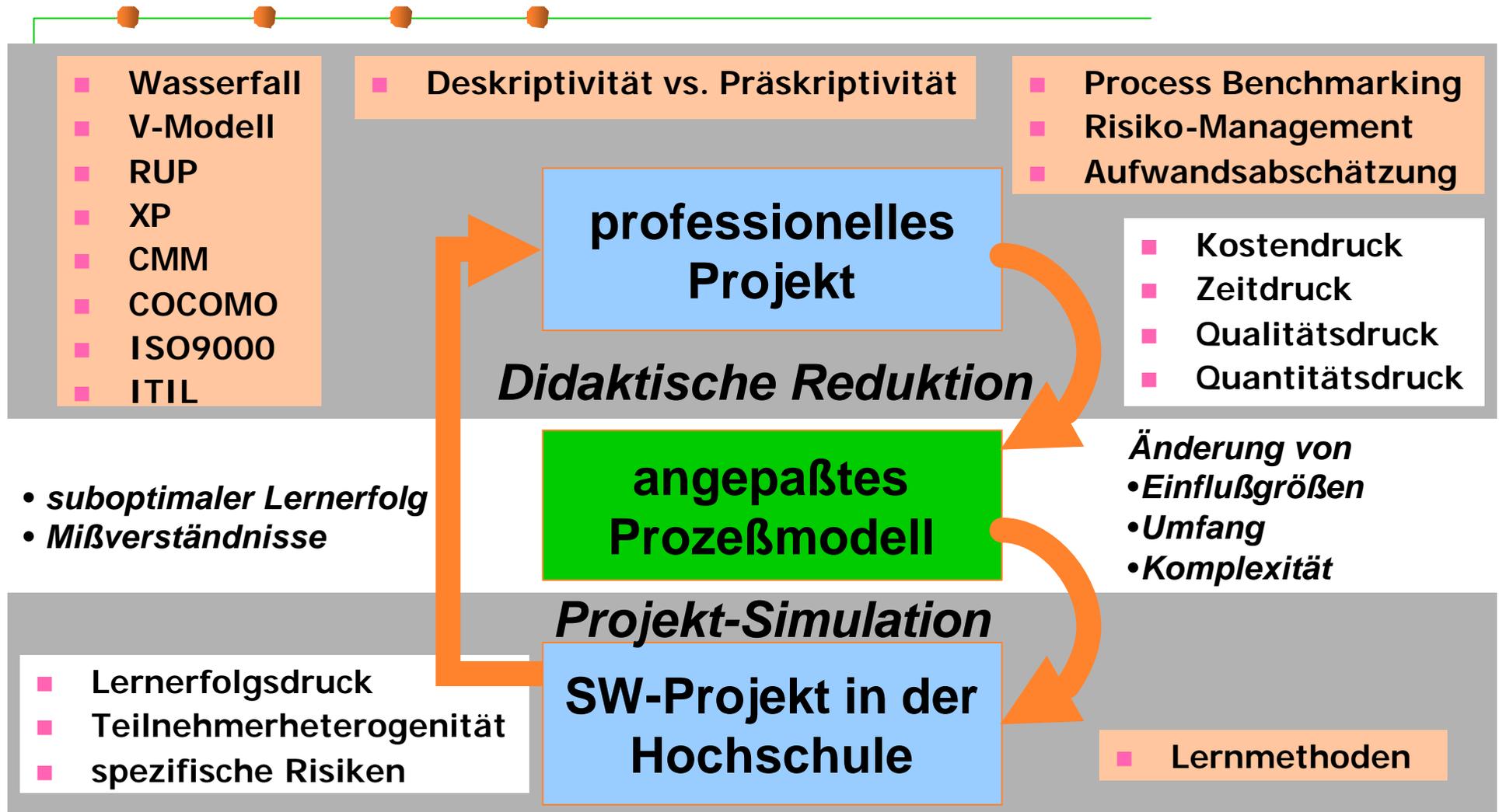
**Dirk Draheim. A CSCW and Project Management Tool for Learning Software Engineering. FIE 2003 - Frontiers in Education: Engineering as a Human Endeavor. IEEE Press, to appear.**



# EASE - These

- Professionelle Software-Entwicklung:  
produktorientiertes Risiko-Management
- Studentisches Software-Projekt:  
Maximierung des Lernerfolgs
- ***Ein Prozeßmodell, das sich konsequent an Risiken und Chancen des Hochschulunterrichts orientiert, lohnt sich, auch wenn es im Ergebnis aus professioneller Sicht kontraproduktiv ist.***

# Einsatz eines professionellen Prozeßmodells als Risiko





# Risikofaktoren des Hochschulunterrichts

---

- Studenten, allgemein:
  - Unterschiedliche Anfangsqualifikation
  - Unterschiedliche Talente
  - Unterschiedliche Zielsetzungen
  - Unterschiedliche Wahrnehmung
- Studenten, Informatik:
  - Real Programmers
  - Baby Duck Syndrome
- Dozent:
  - Zu starke Vereinfachung von Lehrinhalten
  - Zu große Präskriptivität



# Chancen des Hochschulunterrichts

- Studenten sind aufnahmefähig
- Studenten sind hochmotiviert

## *Grundannahmen !*

- Teilnehmerheterogenität bzgl. Anfangsvoraussetzungen kann sich positiv auf Gruppenarbeit und gemeinschaftliches Lernen auswirken



# 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)

**EASE**



# EASE - Vorgehensmodell

- Identifikation von Risiken und Chancen
- Prozeß-Beispielablauf: Makroprozeß, stark iterativer Mikroprozeß, Tracking
- Katalog von Vorgehensmustern
- Lernmethoden mit Einfluß auf EASE
  - Collaborative Learning (Bruffee)
  - Action Learning (Revans)
  - Entraînement Mental (Chosson)
  - Erwachsenenbildung aus konstruktivistischer Sicht (Siebert)
  - Jigsaw Classroom (Aronson)



# EASE - Vorgehensmuster

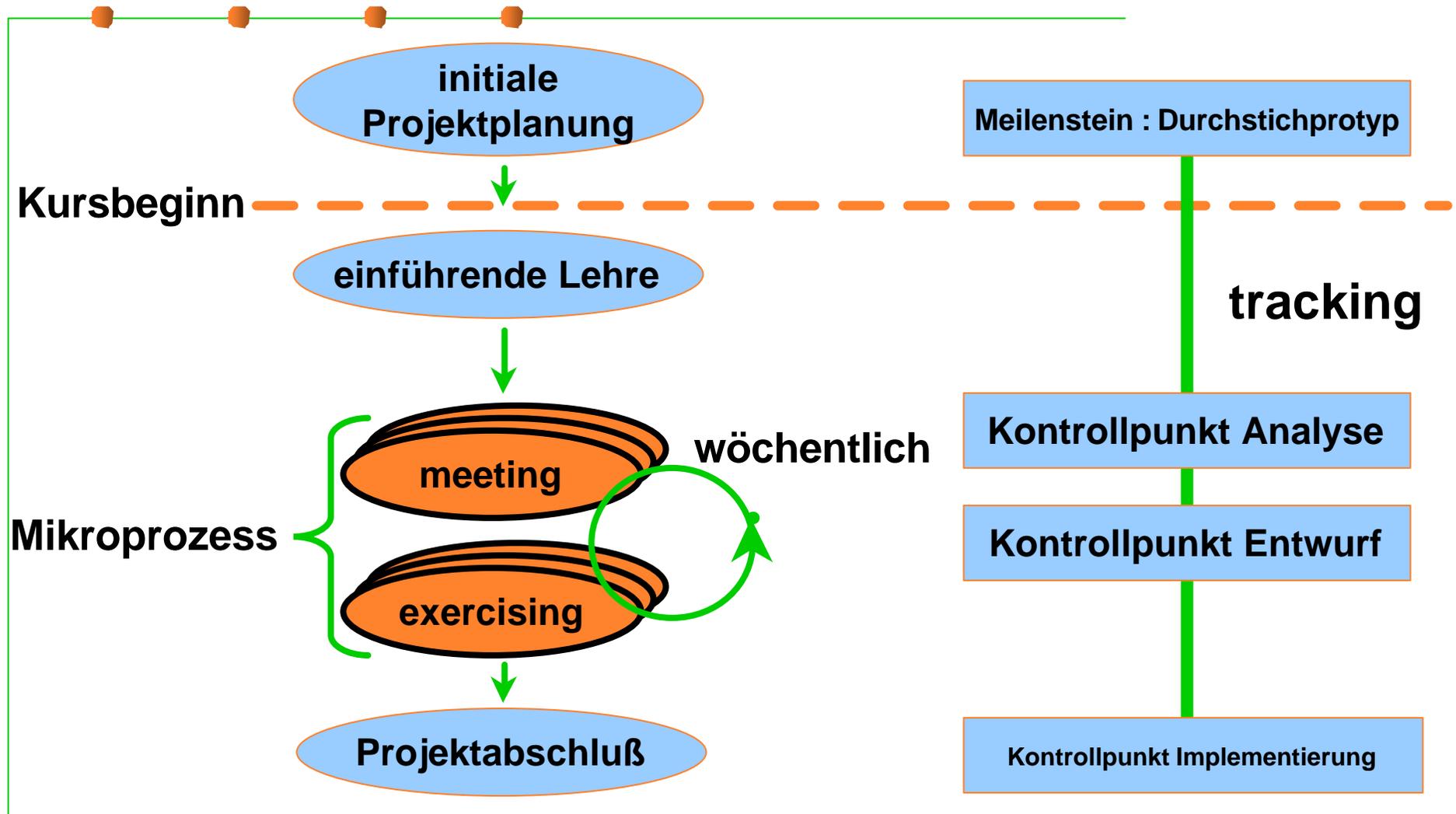
- Fest vorgegebene Gesamtstundenanzahl
- Erzeugnisorientierung und simultane Bearbeitung aller Erzeugnisse
- Gemeinschaftliche Aufgabenbearbeitung
- Wechselnde Kleingruppen
- Gemeinschaftliche Verantwortung
- Kontrollpunkte
- Sonstige Vorgehensmuster: Rolle des Dozenten, Rolle des Tutors, Durchstichprototyp, Experten, Gruppenarbeit, Lehrer-Lerner-Vertrag



# EASE - Anspruch

- EASE schreibt **kein Modellierungsparadigma** vor.
- EASE schreibt **kein Programmiersprachenparadigma** vor.
- EASE zielt auf funktionale Qualifikationen auf hohem Niveau und extrafunktionale Qualifikation, aber schreibt **keine Lernziele** vor.
- *Best Practices ? Private, recommended, good, proven, best patterns/practices*
- EASE macht Vorschläge:
  - EASE Vorgehensmuster sind **Vorschläge/Diskussionsbeiträge**.
  - Der vorgeschlagene typische EASE-Beispielablauf ist eine **Erklärung** der Vorgehensmuster und ihres Zusammenwirkens.

# EASE – Beispielablauf





# Initiale Projektplanung

- Dozent definiert die Lernziele, berücksichtigt werden:
  - Curriculum
  - Eigene Überzeugungen
- Dozent definiert angestrebtes System als Produktvision.
- Dozent wählt Technologien/Methoden.
- Dozent definiert zu erstellende Erzeugnisse.
- Dozent entwickelt Durchstichprototypen.
- Dozent setzt Termine für Kontrollpunkte fest.



# Einführende Lehre

- 
- Erstes Plenum: Das angestrebte System wird erklärt.
  - Studenten werden für die Dauer des Projekts in Übungsgruppen aufgeteilt (jeweils 10-40 Personen).
  - Erstes entspanntes Gruppentreffen



# Fest vorgegebene Gesamtstundenanzahl

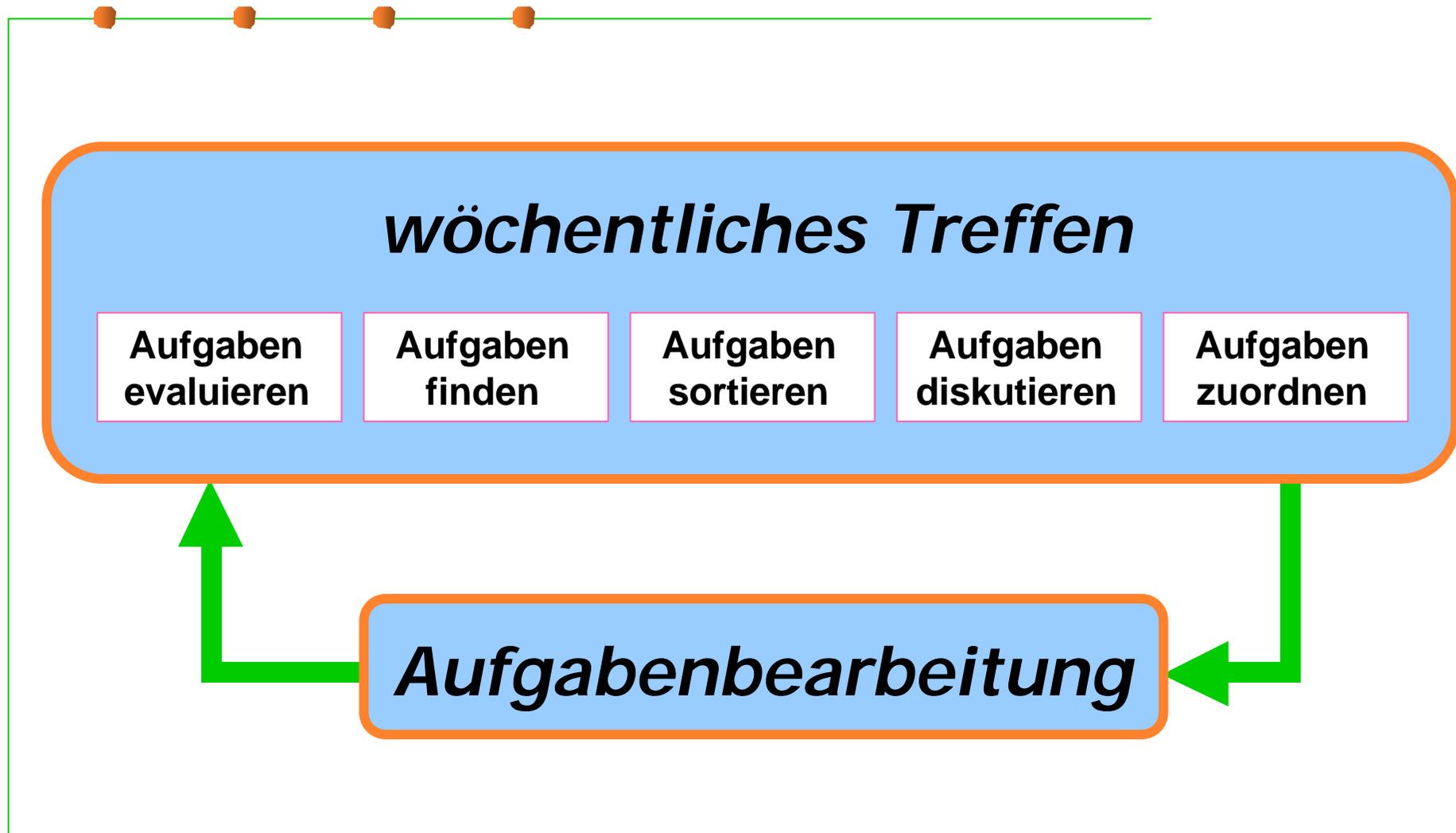
- Aufgabenstellung als Produktvision
- Feste Gesamtarbeitsstundenanzahl statt feste Gesamtaufgabe
- Aufgabe: durch Gruppenarbeit von informeller Beschreibung zu möglichst umfangreichem hochwertigem System
- Selbstorganisation
  - Permanente Planung und Kontrolle
  - Übung von Aufwandsabschätzung und Anforderungsdefinition
  - Einfluß auf Systemfunktionalität und Realisierung



# Erzeugnisorientierung

- 
- Sicht der Studenten : keine Phasen
  - Kursbeginn: Definition zu erstellender Erzeugnisse und Erzeugnismengen liegt vor
  - Projektablauf: simultane, kontrollierte Bearbeitung aller Erzeugnismengen

# Mikroprozess





# Gemeinsame Aufgabenplanung

- Gemeinsame Planung des Projektfortschritts
- Gemeinsame Kontrolle des Projektfortschritts
- Rolle des Tutors:
  - Kein Lehrer
  - Motivation, Moderation, Experte



# Wechselnde Kleingruppen

- Wöchentlich neue Aufteilung der Gesamtgruppe in Teilgruppen
- Studenten lernen voneinander
- Verhinderung zu starker Spezialisierung durch Umgang mit
  - verschiedenen Erzeugnisarten
  - verschiedenen Technologien
  - verschiedenen Einflußgrößen auf Software-Architektur



# Gemeinsame Verantwortung

- 
- Gemeinsame Verantwortung für alle Dokumente (gemeinsamer Besitz)
  - Kontrollpunkte statt Meilensteine
    - Inspektion der Erzeugnisse
    - Überprüfung: Einhaltung der Prinzipien von EASE



# EASE - Fallstudie

- Softwarepraktikum 2001 (3 Gruppen je 15-20 Teilnehmer)
- Produktvision: Webshop
- Teilnehmer: 80% Informatik-Studenten, 10% Erfahrung mit netzbasierten Systemen, 10% Erfahrung mit Datenbanken
- Erzeugnisse: Form Storyboard, UML Datenmodell, textuelle Anforderungsspezifikation, Architekturdiagramme, kommentierter Code, automatisierte Tests
- Lernerfolge (**Hard Skills**) : Sourceforge, CVS, Visual Age IDE und GUI-Builder, JTogether, Multi-Tier-Architecture, Load-Balancing, DB-Caching, Connection-Pooling, DB-Trigger für Constraint Checking, JavaScript, UNIX-Calls, JSP, Servlets, JDBC, PL/SQL

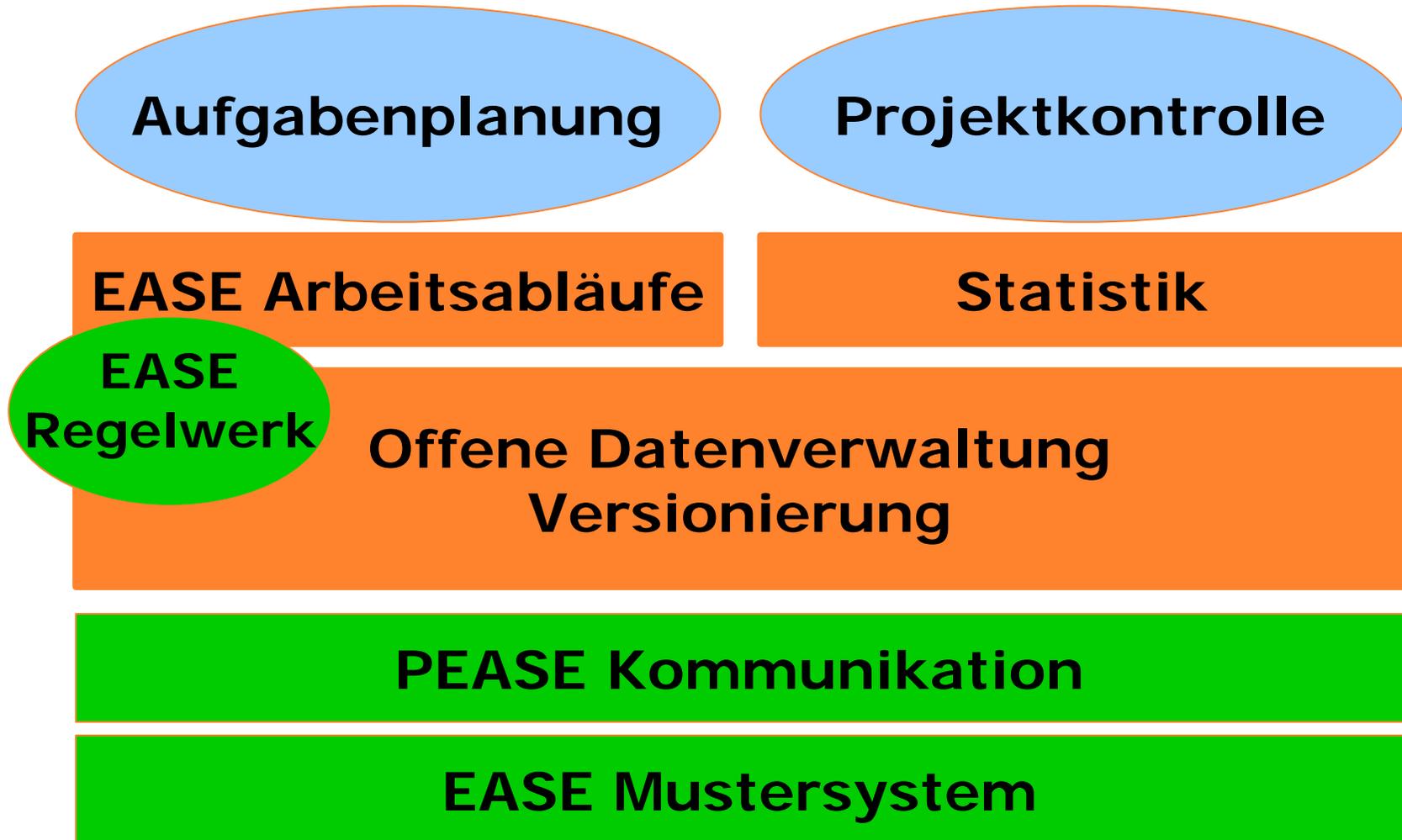
***Dirk Draheim and Gerald Weber. Storyboarding Form-Based Interfaces. INTERACT 2003 - Ninth IFIP TC13 International Conference on Human-Computer Interaction.***



# Das Werkzeug PEASE für CSCW/Projektmanagement

- Projektmanagement - Dokumentation, Kontrolle, Steuerung : MS Project
- Groupware / CSCW (Computer Supported Cooperative Work) : LOTUS Notes
- Kombinierte Funktionalität: BSCW, Sourceforge, PHProject, Tutos, Wice, Xcolla
- PEASE : „Box of Chocolates“ für EASE „Jump Start“
- Web-basierte Standard-Dreischichtarchitektur
- GNU Public License

# PEASE – Platform for EASE





# PEASE - Offene Datenverwaltung

- Verwaltung beliebig vieler Projektgruppen
- Projektdaten
  - z.B. Projektbeschreibung, Teilnehmer mit ihren Daten, Gruppen mit ihren Teilnehmern, Treffen mit ihrem Termin, Listen der gefundenen und tatsächlich zu bearbeitenden Aufgaben, Teilnehmer von Kleingruppen *usw.*
  - Ansichtsmodus, Editiermodus, und Historiemodus
- Erzeugnisse
  - Konfigurierbares baumförmigen Verzeichnis
  - Automatische Versionierung



# PEASE - Regelwerk

- Diverse Regeln bzgl. Prozeßstruktur und Prinzipien von EASE
  - z.B. Veränderung von Aufgabendaten durch nebenverantwortliche Teilnehmer, Abhängigkeit einer Aufgabe von einem nicht aktuellen Treffen *usw.*
- Warnprinzip
  - Keine Erzwingung der Regeln
  - Warnungen im Dialog
  - Mail-Benachrichtigungen



# PEASE - Kommunikationsplattform

- Mail-Adressen-Verwaltung
- Kontextabhängige Mails entlang Hauptverantwortlichkeiten
- Mail-Beantwortung über das System
- Mail-Archivierung



# PEASE Arbeitsablaufunterstützung

- Motivation: schneller Einstieg in das EASE-Projektmanagement
- Zentrale EASE Arbeitsabläufe als modale Dialoge in PEASE
  - Erläuterung der Einzelschritte
  - Aufbereitung der Daten zur Entscheidungsfindung in jedem Schritt
  - Erfassung der relevanten Daten



# PEASE - Statistikfunktionalität

- Motivation: Kontrolle des Projektfortschritts und der Einhaltung von EASE-Prinzipien
- Filterschnittstelle
- Vordefinierte typische Anfragen
- Related Work : <http://bloof.sourceforge.net/>

***Dirk Draheim and Lukasz Pekacki. Process-Centric Analytical Processing of Version Control Data. IWPSE 2003 - International Workshop on Principles of Software Evolution. IEEE Press, to appear.***



# PEASE - Vorgehensmuster

- 
- Katalog der EASE-Vorgehensmuster (process pattern language) als Hypertext
  - Integration mit den modalen Dialogen und der offenen Datenverwaltung von PEASE



# Resümee

- Software-Projekte im Hochschulunterricht unterscheiden sich von professionellen Software-Projekten in Umfang und Einflußgrößen.
- Der unreflektierte Einsatz eines professionellen Software-Prozesses im Hochschulunterricht birgt Risiken.
- EASE analysiert Risiken und Chancen des Software-Projekts im Hochschulunterricht.
- EASE zielt auf funktionale Qualifikation auf hohem Niveau und extrafunktionale Qualifikation.
- EASE schlägt Vorgehensmuster und einen typischen Projektablauf vor.
- EASE schreibt keine konkreten Lernziele vor, EASE erzwingt kein Modellierungs- oder Programmierparadigma.
- PEASE ist ein spezialisiertes CSCW/Projekt-Management-Werkzeug für EASE