

Lehren für die Lehre

Wie man Software-Qualität praktisch vermittelt

19. STEV Österreich Fachtagung
Softwarequalitätsmanagement in der Praxis
Wien, 7. Mai 2004

Prof. Dr. Kurt Schneider
Lehrstuhl Software Engineering



Universität Hannover
Kurt.Schneider@Inf.Uni-Hannover.de

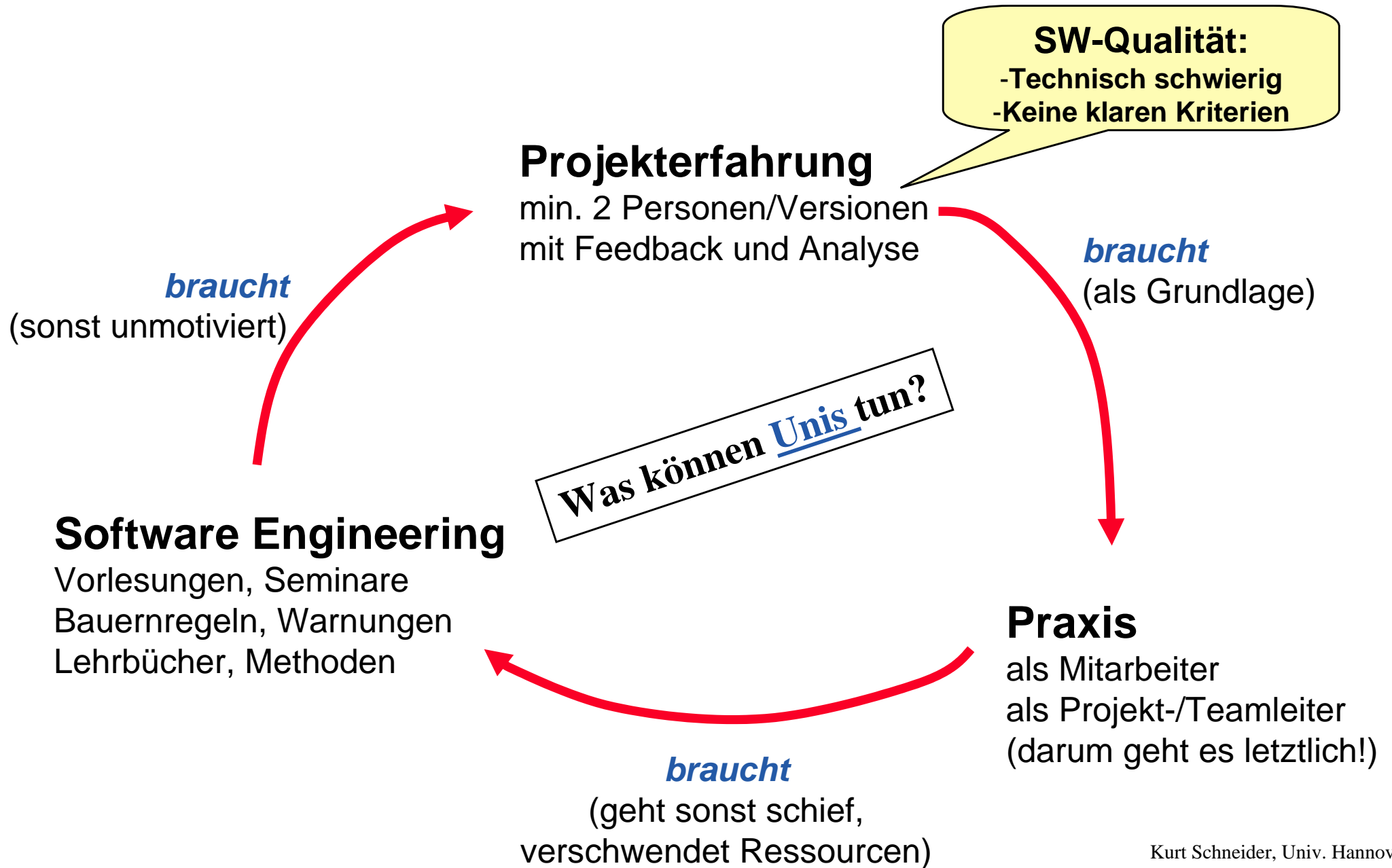


Überblick

- **Die Situation: Erfahrung tut not**
- **Die Gelegenheit: „Praxis in die Lehre tragen“**
- **Was ist anders?**
 - Inhalte
 - Lehrformen
 - Kultur
- **Was haben Sie davon, das zu hören?**
 - Ermutigung zur Reflexion
 - Anregungen für interne Weiterbildung
 - Gelegenheit, Ihre Erfahrung einzubringen



Henne und Ei der SW-Erfahrung





Klassische Situation: Praxis an die Uni

- **Alter Wunsch: „Praxis in die Uni bringen“**
 - Offenbar fehlt Absolventen etwas
 - Aber was ist das konkret?
 - Und: kann man es schon an der Uni vermitteln?
- **Gute Gelegenheit:
Von DaimlerChrysler an den Lehrstuhl**
 - Was hätten wir uns von Absolventen gewünscht?
- **Konkrete Situation:**
 - Ernennung am 29.9.2003
 - Erste Vorlesung Oktober 2003: Schnelle Reaktion gefordert!
- **Die Frage: Was wird nun durch den Praxishintergrund anders?**



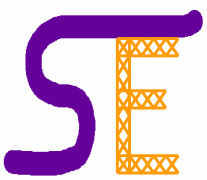
„Was würden Sie
tun - spontan?“

- **Option 1: „Universitätsstandard“**
 - Bestehende Vorlesung vom Internet ziehen
 - Inhaltsverzeichnis von SE-/QS-Büchern studieren
 - Nahe am Vorgefundenen arbeiten – sicherheitshalber



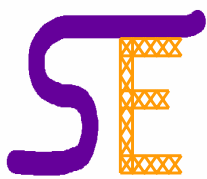
- **Option 2: „Die Uni als Firma“**
 - Alles so gestalten, wie es in der Firma war
 - Zwänge und Randbedingungen möglichst nachbauen
 - Von Studierenden Umgewöhnung einfordern

- **Meine Option 3: „Gezielt ausgewählte Aspekte einführen“**
 - In etwa wird der übliche Rahmen und Inhalt gewahrt
 - Wenige, besonders auffällige Praxis-Eigenheiten werden nachgebildet
 - Manche erfordern regelrechte *Simulation* der Wirtschaftspraxis



Was wird anders?

- **Welche Inhalte?**
- **Welche Lehrformen?**
- **Welche Schwerpunkte und Besonderheiten?**



Inhaltsverzeichnis der Vorlesung

Auf den ersten Blick: nicht viel Besonderes



Themen der Vorlesung

1 Motivation und Begriffe

Bedeutung der Software
Engineering als Leitbild

2 Anforderungen/Requirements

2.1 Allgemeines, Struct. Analyse
2.2 Essenzielle und OO Analyse
2.3 Use cases

3 Struktur u. Architektur v. SW

3.1 SW als Produkt: Struktur, Archit.
3.2 Design Patterns: Strukturbausteine

4 Modelle im SE

Modelle besser verstehen; Modelltheorie

5 Software-Qualität (Produktsicht)

Qualität des Produkts SW
Qualitätsbegriff, -aspekte und -maße
Qualitätsnormen und -modelle

6 Prozesse in der SWT

6.1 Bekannte Prozesse, V-Modell, PTs
6.2 Prozessbewertung u. Reifegrade

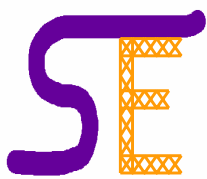
7 Software-Qualitätssicherung in Prozess u. Projekt

8 SW-Projektmanagement

8.1 Aufgaben des Projektmanagements
8.2 Schätzen, Planen, Messen
8.3 Spezielle Aspekte:
Risikomanagement,
Umgang mit Auftraggeber, Agilität

9 Der Mensch in der SWT

Ernst genommen –
9.1 Usability Engineering
9.2 Erfahrungsnutzung, Reflexion



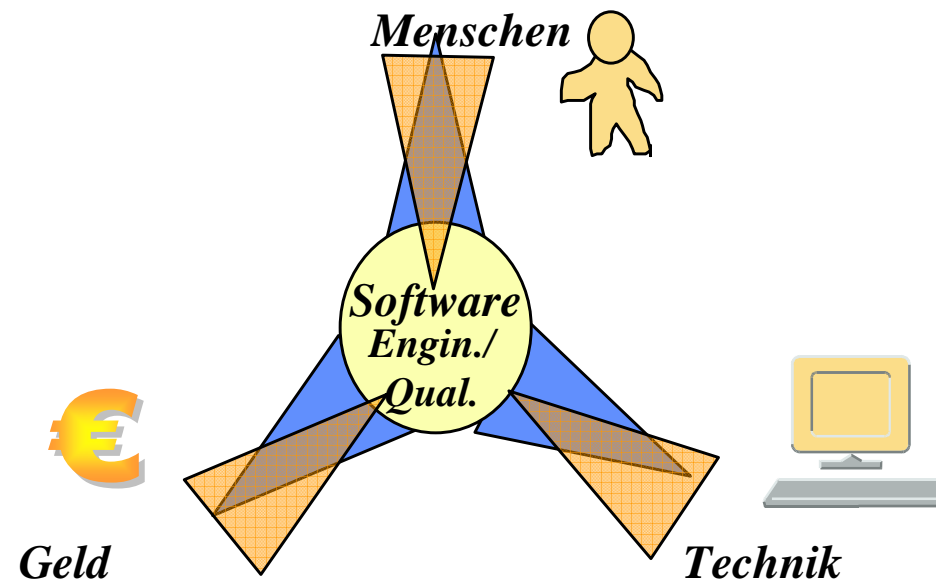
„Neue“ Themen und Schwerpunkte

- **Erkennbare Änderungen**
 - Requirements Management sehr ausführlich
 - Software-Prozesse: Bedeutung und pragmatischer Einsatz
 - Prozessreife-Modelle (CMM et al.)
 - Agile Konzepte schon im 3. Semester – noch vor dem Wasserfall!
 - **Sensibilisierung**
 - **Erfahrungsbasierte Prozessverbesserung**
 - **Usability Engineering**
- **Viel Bewährtes zur Software-Qualität (zusätzl.: Quality Gates)**
- **Zusätzlicher Schwerpunkt:**
 - Reflexion lernen, Erfahrungen schätzen lernen
- **Ist diese Auswahl widersinnig / beliebig / vernünftig?**

Inhalte

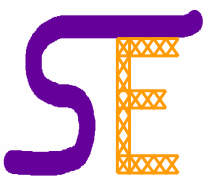
Zusammenhänge betonen

Mensch, Geld, Technik: SE und Qualität verbinden alle drei



– **Studenten haben Schwierigkeiten mit der Nutzenperspektive**

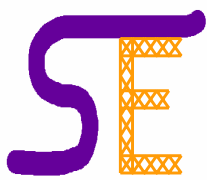
- Auch Doktoranden
- Für Firmen aber von allergrößter Wichtigkeit



Im Hintergrund

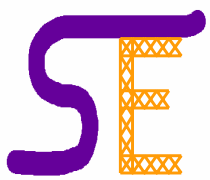
Kultur in Unternehmen

- **Ziele setzen**
- **Bewertbar machen**
- **Freiheit der Gestaltung - in strengem Rahmen**
- **Kritisch denken**
- **Nicht das Ideale suchen, sondern das „machbare Optimum“**
- **Nutzen (bei Entwicklung und Prüfung) im Kopf haben**

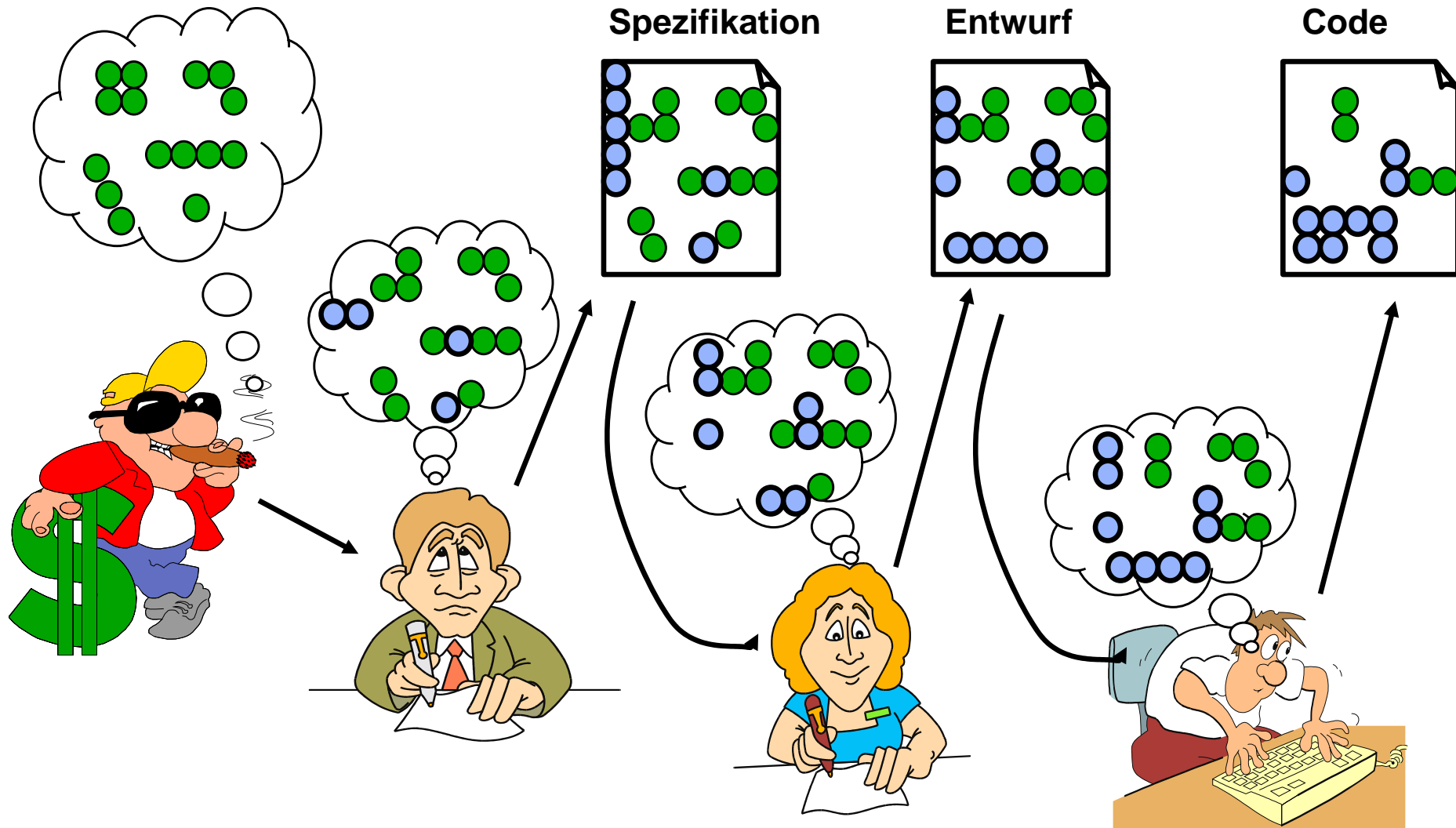


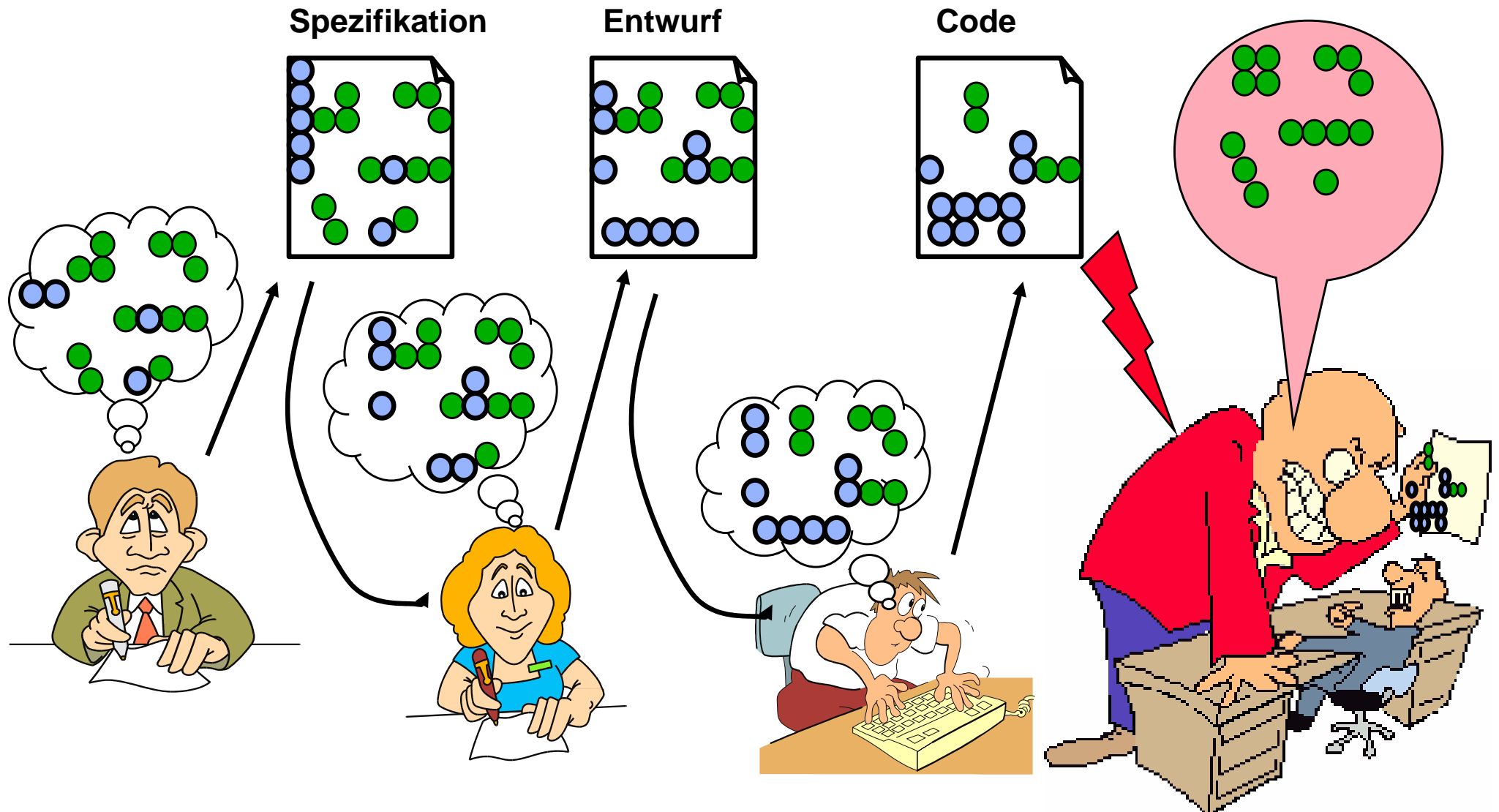
Vorteile von Universitäten ausspielen

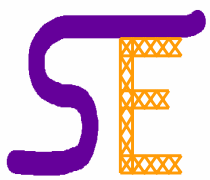
- **Gelegenheiten und Freiheiten**
 - etwas mehrfach tun und vergleichen
 - Zeit für Erfahrungssammlung und Reflexion einplanen
 - Produkte nutzen – oder einfach wegzuwerfen
- **Wer lernt, darf ...**
 - Mehr fragen
 - Etwas noch nicht wissen
 - Sich mit anderen absprechen
 - Noch einmal neu beginnen
- **Wer lehrt, darf ...**
 - Qualitätsanforderungen festschreiben
 - Rahmenbedingungen setzen – echte oder künstliche
 - Sich etwa ausdenken und es einfach einmal ausprobieren!



Lehrform „Simulation“ Software-Quanten wandern

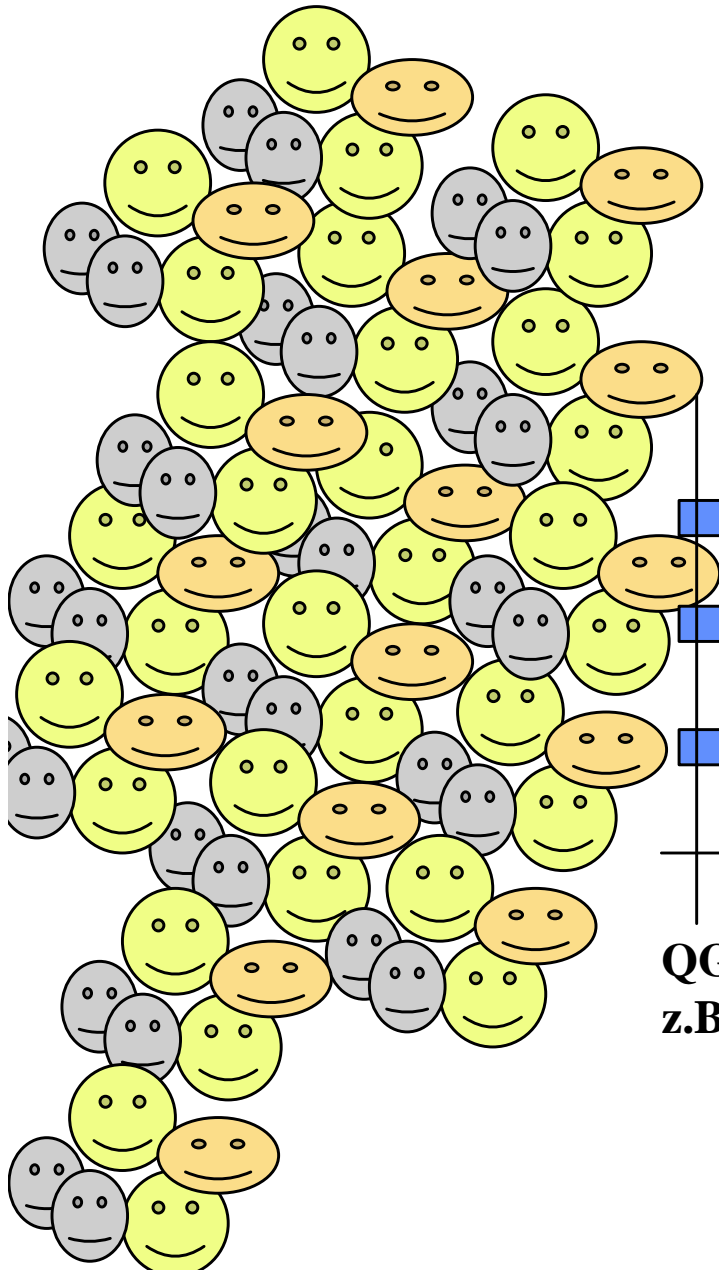






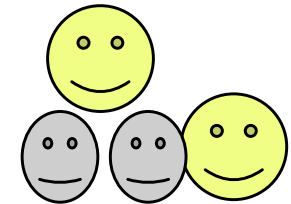
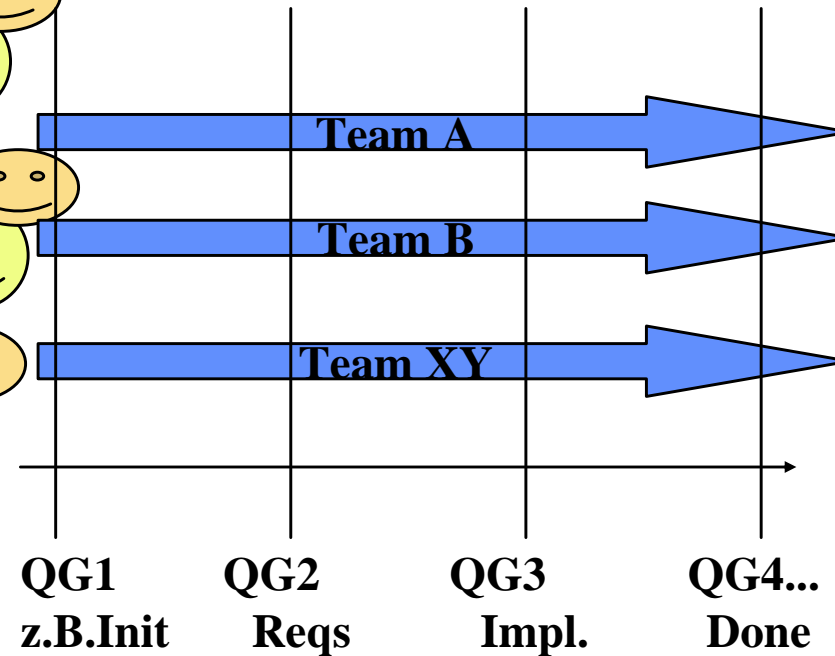
SW-Projekt „praktischer“ machen

Konflikte durch Rollentrennung



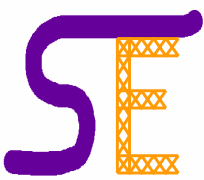
Kunde

- definiert Aufgabe (Termine vor-gesetzt)
- Kann/soll an QGates teilnehmen
- Erhält entstehende SW
- Inhaltlicher Ansprechpartner (nach eigenem Ermessen)



Prozessberater

- Definiert QGates
- Organisiert Plenum
- Berät zu SE-Fragen
- Beruft QGates ein
- Erhebt Erfahrungen
- Wertet aus



Mehr als Theorie, mehr als Praxis

Simulieren: Zeit ist knapp

- **Realität in der Praxis**
 - Möglichkeit zu Kundenkontakt beschränkt
 - SE-Know how beschränkt
- **Realität an der Uni**
 - Assistenten sind meistens ansprechbar
 - „Zeit kostet nichts“
- **Gutscheinsystem simuliert die Zeitknappheit**



Gleiches Startkapital gewährleistet „Gerechtigkeit“
Übrigens: gibt es die in der „Praxis“?



Mehr als Theorie, mehr als Praxis

Reflexion im SW-Projekt

- **SW-Projekt ist nicht nur Programmierung**

- Kundenkontakt
- Zeiteinteilung und Planung
- angemessene Qualität – nicht weniger, aber auch nicht mehr

- **„Reflection in action“**

- Breakdowns
- Gelegenheit zum Innehalten
- Erfahrungen bewusst machen
- Lernen, aus individuellen Erfahrungen zu profitieren
- Erfahrungen explizit sammeln – und in Zukunft für andere einsetzen

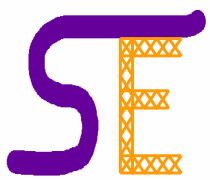
ACHTUNG!

Daran ist kein Schritt einfach.

Erfahrungsbasierte Prozessverbesserung ist ein schwieriges Thema!

- **Ziel: Bessere Reflexion als durch übliche Studentenprojekte**

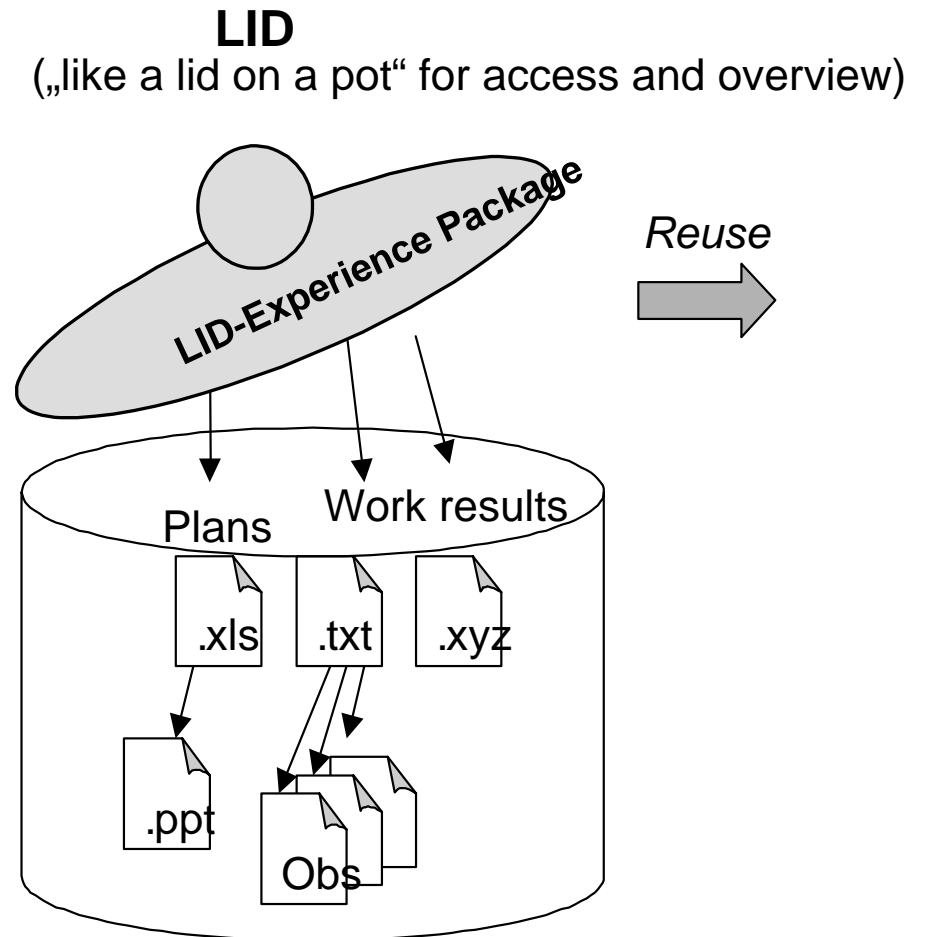
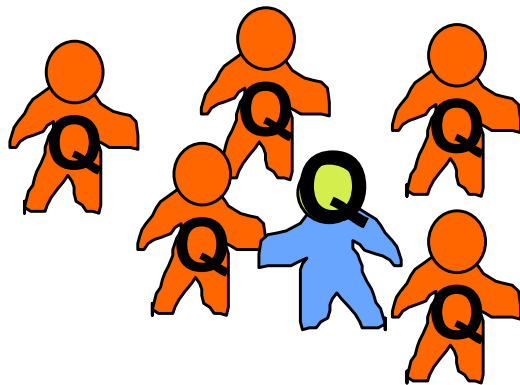
- Sogar bessere als durch reale Projekte!



Mehr als Theorie, mehr als Praxis

Eigene Erfahrung ernst nehmen: LIDs

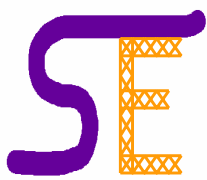
- Nach Aktivität von 2-3 Monaten
- Bevorzugt in kleiner Gruppe
- Ergebnis direkt in den Computer
- Sofortiges Feedback mit Beamer
- Prozess- und Ergebnistemplate leiten Diskussion
- *Search* →
- Resultat: meist 5-15 Seiten
- Leicht-gewichtig: 2-4 h, ca.11-20 Ph



„Pot“: copy of docus mentioned

for a full description of LIDs:
see **Profes 2000** proceedings,
Springer LNCS 1840

LIDs bewährt sich in der Praxis!



Was haben Sie davon, dies zu hören?

- **Auf bessere Absolventen hinarbeiten: weitersagen ...**
 - wenn „Inhalt“ fehlt oder Sie bessere „Lehrform“ kennen
- **Selbst Gedanken machen, was man „in der Praxis“ lernt**
 - Was machen SIE anders als direkt nach dem Studium?
- **Wichtig: Reflection-in-action für Sie selbst**
 - Arbeiten Sie jetzt schon perfekt? Gönnen Sie sich Reflexion!
 - Behandeln Sie Ihre Erfahrungen als einen Schatz – aber nicht verstecken!

- **Nutzen Sie meine und Ihre Anregungen auch für *innerbetriebliche* Kurse**

- Simulationsansätze
- Lernen aus Erfahrung – aber bitte explizit!
- SW-Quanten als Gedankenmodell

- **Denn „man lernt nie aus!“ – hoffentlich!**

Spezifikation

Entwurf

Code

