technische universität
dortmund

Software-Engineering für langlebige Systeme

Thomas Ruhroth | Dortmund
SoSe 2015

Organisatorisches

Feedback Startfragebogen

Zur Person

- Geschlecht*:
  - 38 männlich
  - 1 weiblich
- Studiengang
  - Angewandte Inf: 13
  - Informatik: 15
  - Diplom: 1
- Sie/Du
  - Sie: 2
  - Du: 15
  - Egal: 17

* Durch Zählung in der Vorlesung bestimmt

- Teilnahme:
  - Sicher: 20
  - Ansehen: 11
  - Leer : 3
- Kurse:
  - SWK: 21
  - MBSE: 0

Programmiersprachen

Sprache	Exp	Mit	Grund	keine	WTF?
Java	7	23	0	0	0
C	0	4	24	4	0
C++	2	3	10	17	0
C#	1	3	7	21	0
SmallTalk	0	0	0	21	21
Perl	0	0	0	28	15
Phyton	0	4	5	23	1

- PHP (3x)
- Haskell (2x)
- VBA
- JS
- Android/Java
- JavaEE

Modellierung & Meta-Fähigkeiten

	Exp	Mit	Grund	keine	WTF?
UML	3	14	15	0	0
ERD	0	1	4	3	23
BPMN	2	2	3	3	23
Moderation	1	4	12	13	2
Proj.-Leit.	1	3	16	12	2

Vorwissen

	richtig	teilweise richtig	k.a.	falsch
Softwareerosion	1	8	18	3
Design-Pattern	7	7	13	1
Configuration Management	0	1	27	1
Code-Guidlines	8	5	15	1

Misserfolg-Frage:

- Nur graue Theorie ohne praktische Anwendung (4x)
- Kein praktischer Bezug (2x)
- Eine andere Programmiersprache als Java verwenden – Ich kann nur Java
- Zu hohe Anforderungen die zeitlich nicht erfüllbar sind
- Wenn „schwächere“ Studenten (bzw. Studenten mit wenig Vorkenntnissen) auf dem Weg zur Klausur verloren gehen.
- Meteoriteneinschlag
- Schlechte Organisation
- Zu viel vorausgesetztes Wissen
- Persönliches Desinteresse
- Nichtbestehen!
- Übungen sind mit VL-Stoff nicht lösbar
- Wenn die Übung nur an einem Termin stattfindet
- Zu abstrakt
- Angespannte Übungssituation
- Unangenehmer Leistungsdruck
- Extreme Faulheit/Unorganisiertheit meinerseits
- Übungstermin überschneidet sich mit anderen Veranstaltungen
- Ich denke, dass der Übungsstil nicht gut ist

Inhalt der Vorlesung

- Lernen wie man Fehler beim Entwickeln und warten vermeidet
- Weiterentwicklung der Kenntnisse im Bereich SE
- Anschluss an den Stoff verlieren
- Diese Begriffe auch nach der Vorlesung nicht definieren zu können
- Verständnis und Ideen bzw Best-Practices um wiederkehrende Probleme zu vermeiden und dadurch die Möglichkeit zu gewinnen gute Software zu kreieren (3x)
- Grundlage/Praxis/Besonderheiten/Tips&Tricks zur Entwicklung von langlebigen Software-Systemen vermittelt zu bekommen (6x)
- Gegenmaßnahmen im Code/Entwurf
- Spannendes, berufsnahes und interessant vermitteltes Wissen (2x)
- Software, Modellierung, Programmierung,... (3x)

Dozent

- Tempo angemessen (2x)
- Vorkenntnis bzw. Kompetenz bzgl. Thema (2x)
- Gute Erklärungen
- Anschauliche Beispiele
- Kein Monolog
- Netter Umgang mit den Studenten
- Gewissen Humor
- Verständliche, anschauliche, interessante und strukturiere Vorlesung/Inhalte (4x)
- Vom Stoff begeistert sein
- Kompetenz und gute Organisation
- Sollte auf Fragen eingehen
- Prüfung an die Credit-Points angepasst
- Brückenschlag Praxis-Theorie
- Gute Didaktik
- Hilfsbereitschaft
- Auch mal 5 Min stehenbleiben
- Angekündigte Interaktion in Übungen nicht wie angekündigt durchführen
- Das er genauso gut ist, wie in SFL im WiSe 14/15

Schlussfolgerungen… bezüglich Stoff/Aufgaben

- Design-Patter wiederholen (ist Thema von SWT)
- Aufgaben können in Java und C gestellt werden.
  - Zwei Sprachen für Konnektivität
  - C ist nah an der Hardware
  - Java hat virtuelle Maschine
- Praxis-induzierte Aufgaben
  - Praxisnah ist schrecklich kompliziert!
  - Mehr Code und Modelle als Theorie

Organisatorische Fragen?