



Stabilitätspakt für Südosteuropa
Gefördert durch Deutschland
Stability Pact for South Eastern Europe
Sponsored by Germany

Delivery of the JCSE as an Intensive Course for Master's Students at Polytechnic University Tirana

Klaus Bothe
Zoran Putnik

7th Workshop
"Software Engineering Education and Reverse Engineering"

Risan, Montenegro
9 – 14 September 2007

Additional documents:
Tirana-Agenda.doc, Tirana-Handouts.doc, FeedbackTirana.doc

Agenda

- Background and history
- Planning the course
- Delivery of the course: lectures
- Students feedback
- Students from the group:
from the viewpoint of the lecturers



History

- April 2006, DAAD conference at Ohrid lake:
B. Cico and K. Bothe: first vague ideas
- September 2006, Workshop in Ravda:
Outlines of issues: ECTS, preknowledge,
organisational problems
- December 2006: DAAD funded our project also for 2007
with 39.000 EUR, restricted to 27.000 as first rate
- January 2007: Outline organisational plan,
cooperation with Zoran Putnik
(selected lectures, assignments, exams, course website)
- 19 – 24 March 2007: intensive course SE (JCSE),
17 students from 4th semester of the master studies
followed by assignments and examinations

3



Main building of Polytechnic University Tirana



4

Determining the place of the SE course at the the Master's studies at Polytechnic University Tirana

- Master in "Computer Engineering":
4 semesters,
→ curriculum: *next slide*
including **Software Engineering 2** with 6 ECTS
- Bachelor of "Informatics":
including **Software Engineering 1**,
→ contents description (module template):
after next slide

5

Master in "Computer Engineering": Curriculum

I YEAR	Moduls	Done	Credits
	Statistic and Stochastic Processes	YES	each 6
	Digital Design	YES	
	Operating Systems II	YES	
	Data Base II	YES	
	Networking	YES	
	Electronic for Computer Engin.	YES	
	Languages and Compilers	YES	
	Foreign Language II	YES	
	Security of Information Systems	YES	
	Architecture of Control automatic Systems	YES	
II YEAR Moduls			
	Advanced Computer Architecture	YES	6
	JAVA II in Eclipse Platform	YES	6
	Project Management	YES	4
	Distributed Systems II	NO	6
	→ Software Engineering II	NO	6
	WEB Application	NO	6
	Artificial Intelligence	NO	6
	Diploma	NO	20

6

Bachelor of "Informatics", Software Engineering 1: contents description

Objectives:

The course covers the principles, methods and main tools used in modern software engineering. The main topics include: introduction to software engineering concepts, lifecycle of the software, core software engineering activities such as specification, design, development, verification and validation and management, graphical system models in standard UML, introduction to management issues such as project planning all through the analysis of different practical case studies.

This course, being introductory, aims to build key concepts of software engineering and some project planning techniques that will help the students to complete given project assignments. The more specific and in-depth coverage of these topics is planned on the Software Engineering 2 course further in the master level.

The Program

1. Introduction

- Software engineering vs. classical system engineering
- Software engineering processes and management

2. The software lifecycle :

- Requirements; software requirements, system models, formal specification
- Design phase: architectural design, object-oriented design, real-time systems, user interface design
- Software development; iterative development, reuse, software evolution
- Verification and validation; software testing
- Management; exploitation and maintenance of software, quality management

Literature:

1. Ian Sommerville: Software Engineering, 7th Edition, 2001

7

Conclusions from Master in "Computer Engineering" and "Software Engineering 1"

1 st YEAR	Modules	Done	Credits
	Statistic and Stochastic Processes	YES	each 6
	Digital Design	YES	
	Operating Systems II	YES	
	Data Base II	YES	
	Networking	YES	
	Electronic for Computer Engin.	YES	
	Languages and Compilers	YES	
	Foreign Language III	YES	
	Faculty of Information Systems	YES	

Objectives:

The course covers the principles, methods and main tools used in modern software engineering. The main topics include: introduction to software engineering concepts, lifecycle of the software, core software engineering activities such as specification, design, development, verification and validation and management, graphical system models in standard UML, introduction to management issues such as project planning all through the analysis of different practical case studies.

This course, being introductory, aims to build key concepts of software engineering and some project planning techniques that will help the students to complete given project assignments. The more specific and

Conclusions:

(Our knowledge about the educational environment)

- Not quite sure about contents and depth of SE1:
 - How deeply handled UML?
 - Practical exercises?
- What about other parts of the BA?
 - Mathematics: algebra, logics ...?
- What about other sources of software development skills (professional activities in companies)?

8



Challenges – from our side

- English: slides and oral presentation
- Intensive course: lectures in 6 days followed by assignments
- Which topics to present, how long will they take?
1 lecture hour in German = 1 lecture hour in English?
- Unknown educational environment:
pre-knowledge, educational style ...
- Assignments:
HU: 8 with 4 tools (3 are commercial),
Novi Sad: 6 with 2 tools
- Assessment of assignments and examinations ...?

9



Agenda

- Background and history
- **Planning the course**
- Delivery of the course: lectures
- Students feedback
- Students from the group:
from the viewpoint of the lecturers

10

Planning the contents of the course: lecture hours of JCSE

in minutes	Z.B. (W 02)	K.B. (S 03)			
Part I: Introduction			Part III: Software Design		
• 1. What is software engineering	80	120	• 15. Overview of design activities	--	90
• 2. Quality criteria ...	40	45	• 16. Structured design	--	15
• 3. Software process models	120	90	• 17. Object-oriented design	--	45
• 4. Basic concepts ...	60	40	Part IV: Implementation and testing		
Part II: Requirements engineering			• 18. Implementation	--	90
• 5. Results of the ... phase	(70)	100	• 19. Systematic testing	--	180
• 6. Cost estimation	60	100	• 20. Functional testing	--	150
• 7. Function-oriented view	60	50	Part V: Advanced problems		
• 8. Data-oriented view	50	35	• 21. Software metrics	--	180
• 9. Rule-oriented view	50	40	• 22. Maintenance	--	-
• 10. Structured analysis	80	65	• 23. Reverse engineering	--	90
• 11. State-oriented view	(45)	80	• 24. Quality of software development ...	--	90
• 12. Scenario-oriented view	30	25	• 25. Software ergonomics	--	180
• 13. Object-oriented analysis	(60)	210	• 26. User manuals	-	-
• 14. Formal software specification ...	--	190	• 27. Project management	?	90
			• 28. Configuration ... management	-	45
			Sum:	2425	
Tirana: 6 days * 5 lecture hours = 30 lh			2425 : 45 = 53 lecture hours → practical 58 lh		

Selected topics for Tirana

in minutes	Z.B. (W 02)	K.B. (S 03)			
Part I: Introduction			Part III: Software Design		
• 1. What is software engineering	80	120	• 15. Overview of design activities	--	90
• 2. Quality criteria ...	40	45	• 16. Structured design	--	15
• 3. Software process models	120	90	• 17. Object-oriented design	--	45
• 4. Basic concepts ...	60	40	Part IV: Implementation and testing		
Part II: Requirements engineering			• 18. Implementation	--	90
• 5. Results of the ... phase	(70)	100	• 19. Systematic testing	--	180
• 6. Cost estimation	60	100	• 20. Functional testing	--	150
• 7. Function-oriented view	60	50	Part V: Advanced problems		
• 8. Data-oriented view	50	35	• 21. Software metrics	--	180
• 9. Rule-oriented view	50	40	• 22. Maintenance	--	-
• 10. Structured analysis	80	65	• 23. Reverse engineering	--	90
• 11. State-oriented view	(45)	80	• 24. Quality of software development ...	--	90
• 12. Scenario-oriented view	30	25	• 25. Software ergonomics	--	180
• 13. Object-oriented analysis	(60)	210	• 26. User manuals	-	-
• 14. Formal software specification ...	--	190	• 27. Project management	?	90
			• 28. Configuration ... management	-	45
			Sum:	33 lecture hours (out of 58 lh)	

Selected topics for Tirana: Principles

in minutes	Z.B. (W 02)	K.B. (S 03)			
Part I: Introduction			Part III: Software Design		
• 1 What is software engineering	80	120	• 15 Overview of design activities	--	90
• 2 Quality criteria ...	40	45	• 16 Structured design	--	15
• 3 Software process models	120	90	• 17 Object-oriented design	--	45
• 4 Basic concepts ...	60	40	Part IV: Implementation and testing		
Part II: Requirements engineering			• 18 Implementation	--	90
• 5 Results of the ... phase	(70)	100	• 19 Systematic testing	--	180
• 6 Cost estimation	60	100	• 20 Functional testing	--	150
• 7 Function-oriented view	60	50	Part V: Advanced problems		
• 8 Data-oriented			• 21 Software metrics	--	180
• 9 Rule-oriented				--	-
• 10 Structure-oriented				--	90
• 11 State-oriented				--	180
• 12 Scenario-oriented				-	-
• 13 Object-oriented				?	90
• 14 Formal specification				-	45
			33 lecture hours		
			13		

Principles:

- Each phase included – without implementation
- Both paradigms: OO and structured approach
- Structured approach not so in detail (old one)
- Select some important advanced problems

Agenda

- Background and history
- Planning the course
- Delivery of the course: lectures
- Students feedback
- Students from the group: from the viewpoint of the lecturers

Delivery of the course: 19 – 24 March 2007

- 6 days, 5 – 7 hours per day
- **Schedule (agenda)** delivered before → next slide
- **Handouts** before: 48 important key slides → extra file
→ new part of JCSE teaching materials
- All slides published: after the lectures (the whole week)
as 1sided, 4sided pdf
- No course website (17 students): all materials copied to a PC

15

Schedule (agenda) of the lectures

DAY 1		7 lecture hours a 45 minutes	
Introduction: DAAD, JCSE, Tempus, concept of the course 45			
PART I Introduction to Software engineering		6 h	
		Duration normal	planned
1*	What is Software engineering? Motivation, Areas, Definition, History	100 90	90
2*	Quality criteria for software products Classifications, definitions, ISO 9126	45 45	45 Z
3*	Software process models – Introduction Activities of software development, overview of models, Waterfall model, Prototyping (other models are introduced in various topics)	90 120	90
4*	Basic concepts and software development documents Overview and cross-analysis	40 60	45
DAY 2		6 h	
PART II Requirements engineering (analysis and definition)			
		Duration normal	planned
5*	Results of the "analysis and definition" phase Feasibility study, Product model, Requirement document	90 70	60
6*	Cost estimation Costs, factors, function point analysis	90 60	60 Z
7*	Basic concepts of the function-oriented view Function trees, Data flow diagrams	60 60	60
8*	Basic concepts of data-oriented view data dictionary, Entity relationship	35 45	-
9*	Basic concepts of rule-oriented view Rules, Decision tables and trees	45 50	-
10*	Structured analysis Context diagram, DFD-Hierarchy, Mini-specification, Implicit function tree	90 90	90
DAY 3		4 h	
11*	Basic concepts of state-oriented view Petri-Nets, State automata, Activity diagrams	90	60
12*	Basic concepts of scenario-based view Collaboration diagrams, Sequence diagrams	30	30 Z
13*	Object-Oriented analysis Class-diagrams, use-cases, UML, demonstration of a CASE Tool	90 +120	90

DAY 4		5 1/2 h	
14*	Formal software specifications and program verification Z, Algebraic, Hoare	130	130
PART III Design			
15*	Overview of design activities Software architecture, Specification of components, Quality assurance, Overview of some software architectures		60
16*	Structured design Structure charts		15
17*	Object-oriented design Architecture design, user-interface, performance, implementation design		45 Z
DAY 5		5 h	
PART IV Implementation and Testing			
18*	Implementation Principles, methods, guidelines	60	-
19*	Systematic testing Classification, review/audit, control-flow, data-flow oriented	130	130
20*	Functional testing incl. testing tools	90	90 Z
DAY 6		5 h	
PART V Advanced problems			
		22 h	
21*	Software metrics McCabe, Halstead, LOC, OO, CAME-Tools, Demo of MC-Tools	180	130 Z
22*	Maintenance Types, requests, costs, planning		
23*	Reverse engineering Software repair, Reengineering, Restructuring, CARE-Tools	90	90
24*	Quality of software development process and its standardization ISO 9000, Capability assessment models		
25*	Introduction to software ergonomics Graphical user interfaces, Standards, Guidelines		
26*	User manuals Principles and guidelines for writing user-manuals		
27*	Project management Planning, organization, people management, control		
28*	Configuration management Motivation, activities, CVS		

16

Photos from the course: students



17



18



19

Agenda

- Background and history
- Planning the course
- Delivery of the course: lectures
- Students feedback
- Students from the group:
from the viewpoint of the lecturers

20

Feedback from students ("Questionnaire")

Our *standard feedback form with additional questions*:

- Was it a big problem that the slides and presentations were in English?
- Have the handout materials been sufficient to follow the lecture?
- Was it a big problem to get the slides only after the lectures?

+ *Some additional questions with open (free) answers*:

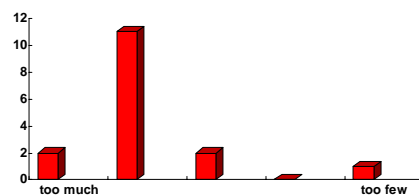
- What should be remained next time?
- What should be changed next time?

Assignments not included (after that week)

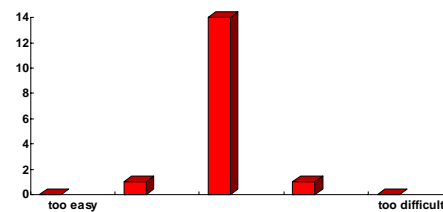
21

Questionnaire results (1)

- Do you consider the amount of knowledge offered in the lectures?



- How do you consider the contents of the lecture?

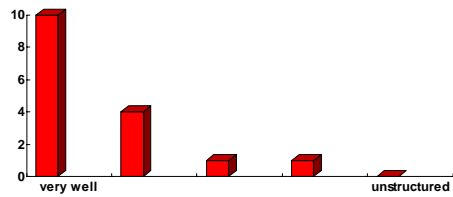


22

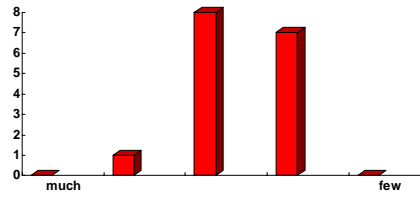


Questionnaire results (2)

- Is the course well-structured?



- Are there any special requirements (pre-knowledge) to be able to understand the course?



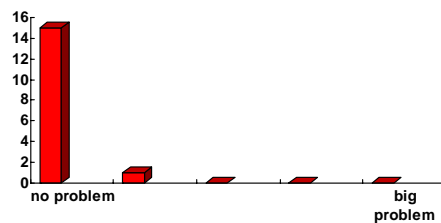
**Preknowledge:
programming languages,
Algebra, logics, UML**

23



Questionnaire results (3)

- Was it a problem that slides and presentation were in English language?

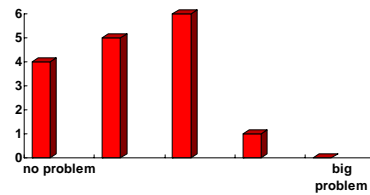
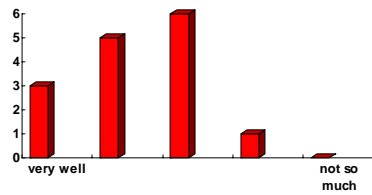


24



Questionnaire results (4)

- Have the handouts materials been sufficient to follow the lecture?
- Was it a big problem to get the slides only after the lectures?

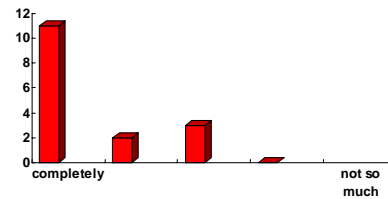
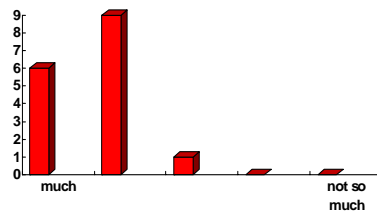


25



Questionnaire results (5)

- Did you learn a lot of new things?
- Do you think the contents of the lecture is useful?

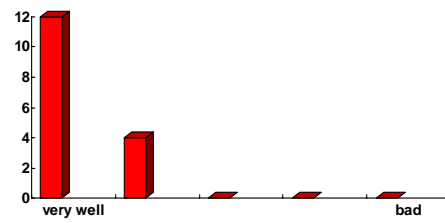


26



Questionnaire results (6)

- What is your overall ranking of the lecture?




27



Feedback from the students: open questions

- What should be remained next time?
 - What should be changed next time?
- Answers next slide

28



What should be changed next time: remarks from students (1)

- Nothing
- Don't change anything
- All should be remained as it is
- Nothing should be changed
- ...
- ...


29



What should be changed next time: remarks from students (2)

- Course within a week too intensive: two weeks much better
- More exercises needed after or within the lectures
- More examples, e.g. from real projects (including experience) should be added to some topics
- More questions to students during the lecture (interactivity)
- Slides before the lectures → notes during the lecture
- Best topic was "Testing": we saw a real testing tool CTE
- Only lectures connected with assignments should be presented; since assignments are more important
- Testing should be included as an assignment (in particular: testing tools)

30



What should be changed next time: remarks from students (3)

- Market requirements like job offers were interesting
- Course better to presented as “Introduction to SE”:
Thus, some parts should be offered in specialized courses
- Some topics should be handled in more detail, not only
as an overview
- Also some more topics should be included: concerning new
approaches to SE (e.g. “Domain engineering”)
- A lot of information was connected with German industry:
include also other European countries
- Some references are a bit old: job offers, cited literature ...

In general: rather satisfied with the course

31



Official feedback from Tirana

- Poster
- Letter to DAAD
- Letter to the head of Department of Mathematics and
Informatics at University of Novi Sad

**→ Rather positive answer by DAAD
to this kind of intensive course**

32



Agenda

- Background and history
- Planning the course
- Delivery of the course: lectures
- Students feedback
- Students from the group:
from the viewpoint of the lecturers

33



Students from the Master's group from the viewpoint of lecturers

- Highly motivated, rather interested
- Working in software companies
- Engaged in lessons by giving feedback to questions rather quickly (they like interactivity)
- Not so much own questions
(Stack with algebraic specifications: "What about the methodology to find the proper axioms?")
- Do not make notices during the lectures (remarks from lecturers) like German students
- Disciplined: attendance, in-time present, no eating and drinking, no leaving the room and return with a cafe ☺
- Open-minded

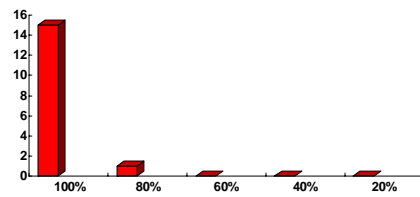
34



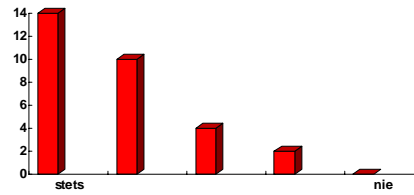
Questionnaire results (11)

- How many lectures did you attend (percentage)

Tirana



HU Berlin



35



Lecturers invited to a café after the week of lessons



36



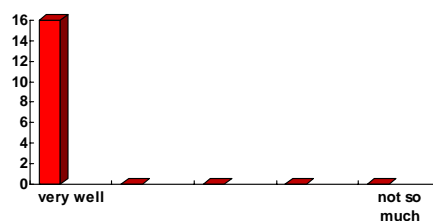
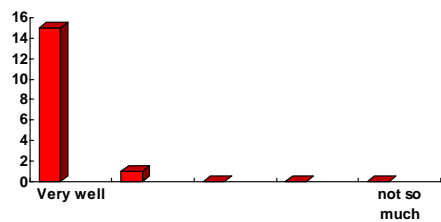
Appendix: Some more questionnaire results

37



Questionnaire results (4)

- Is the lecturer familiar with the contents of the lecture?
- Do the lectures seem to be well-prepared?

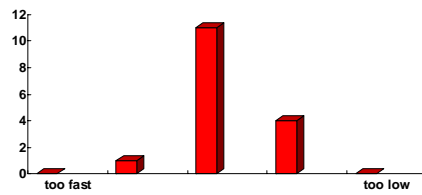


38

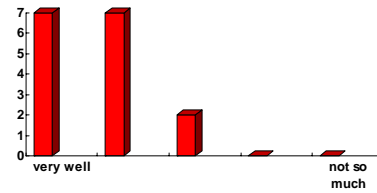


Questionnaire results (5)

- Is the presentation of the lecture



- Does the presentation style encourage you to follow the lecture?

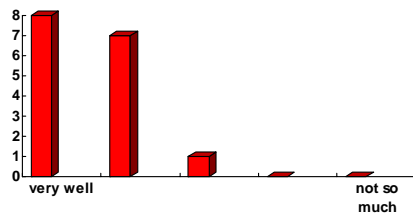


39

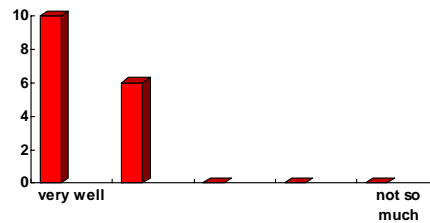


Questionnaire results (6)

- Is the amount of information on the slides adequate?



- Are the slides well-structured and clearly-arranged?



40