From presentation slides to self-study slides

Klaus Bothe

Institute of Informatics, Humboldt University – Berlin, Germany,
bothe@informatik.hu-berlin.de

Workshop

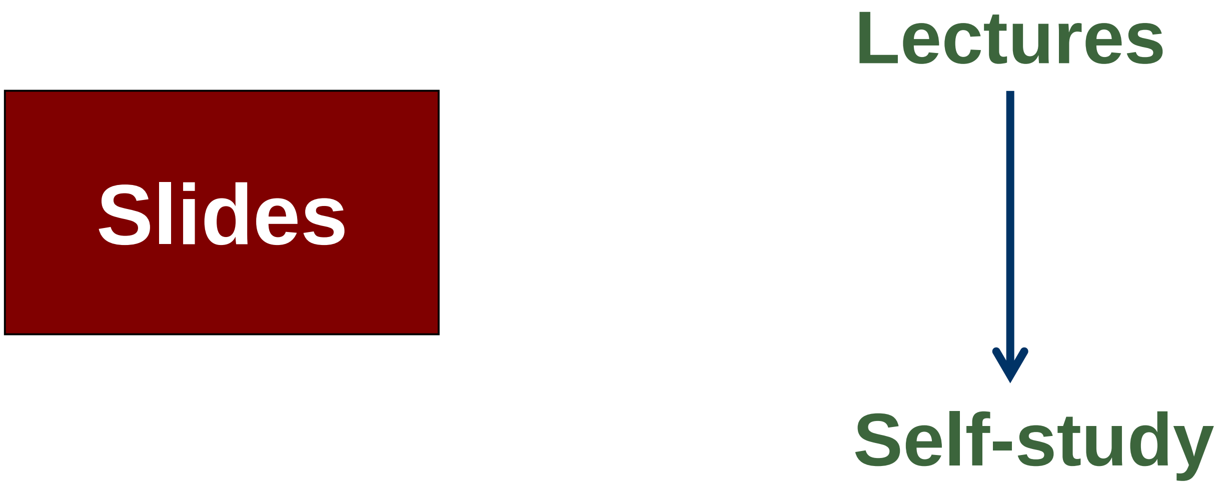
The impact of pandemic years to informatics education: review and next steps

Shkodër/Shkodra, Albania, 4th - 8th September 2023

Contents

- Introduction
- Lectures at Tirana
- JCSE presentations
- Self-Study Slides
- Summary

Basic problem: Are slides used in lectures also convenient for self-study?



Slides

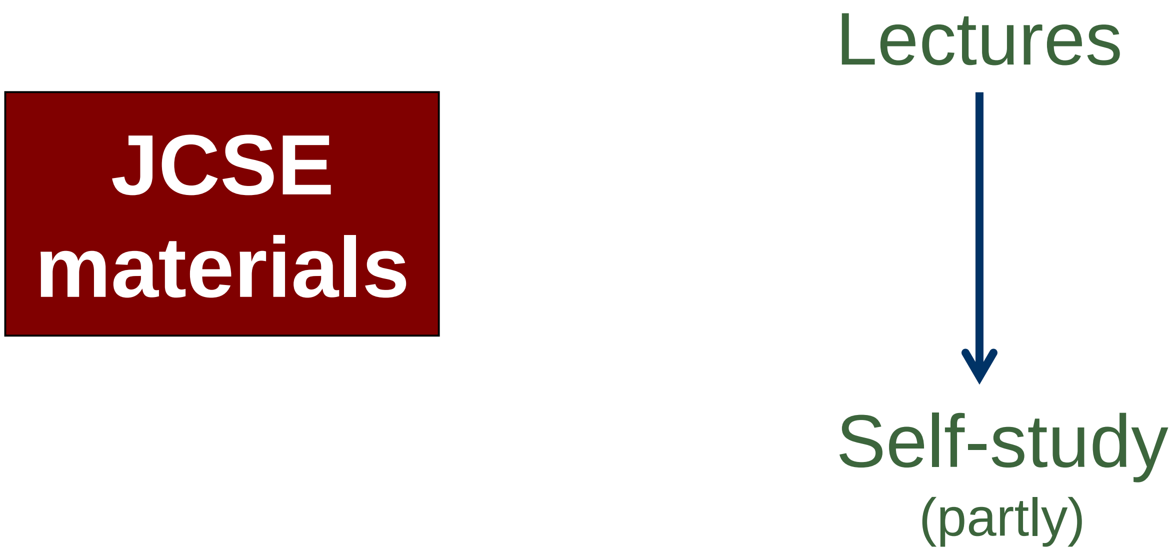
Lectures

Self-study

Example: JCSE
(Joint Course on Software Engineering)

Background

Background: JCSE at Polytechnic University Tirana



JCSE materials

Lectures

Self-study

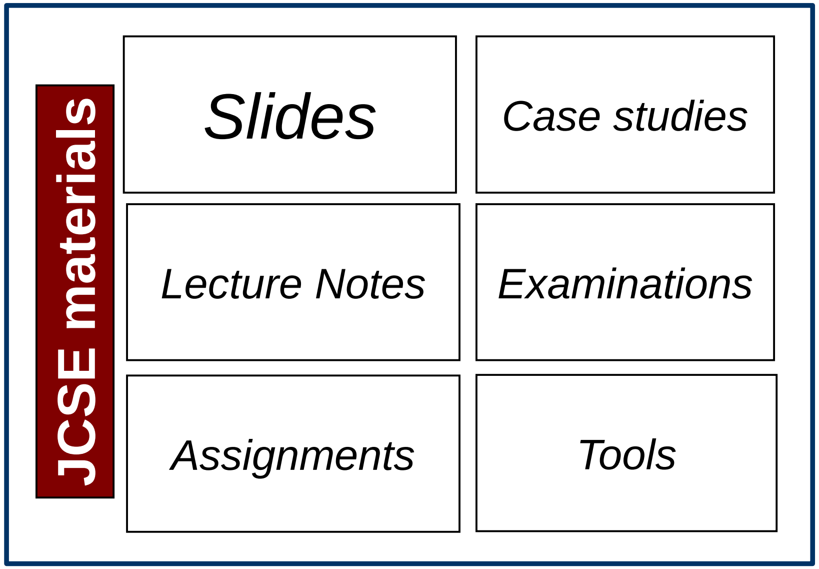
(partly)

Question:

Are JCSE materials also convenient to self-study?

Background

Background: JCSE at Polytechnic University Tirana



JCSE materials

Slides

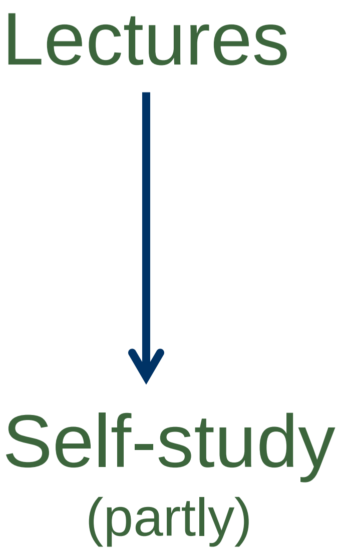
Case studies

Lecture Notes

Examinations

Assignments

Tools



Lectures

Self-study

(partly)

Question:

Are JCSE materials also convenient to self-study?

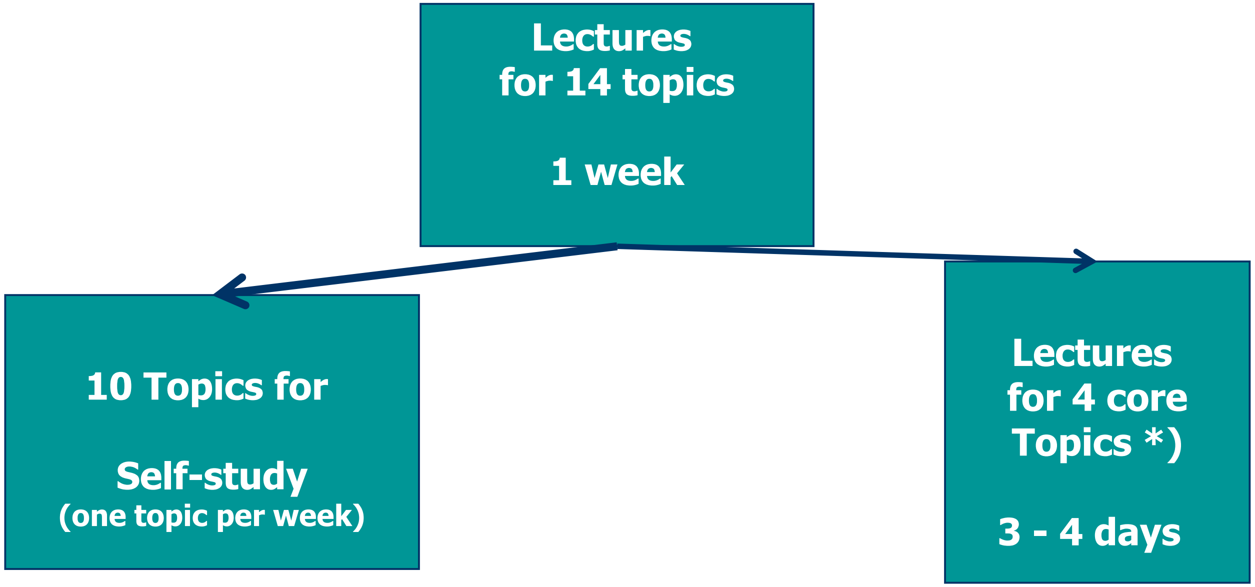
Why?

From complete face-to-face presentation of lectures to partly self-study on the basis of existing slides

WHY?

- Pandemic situation: Home study (2020 – 2022)
- Master students with side jobs at companies
- Special situation of intensive course at Tirana (one week, lecturer’s workload in a week)

Question: Are lecture slides convenient for self-study?



Lectures
for 14 topics

1 week

10 Topics for

Self-study
(one topic per week)

Lectures
for 4 core
Topics *)

3 - 4 days

*) used in team project
assignments,
focus in exams

Of course, not!

They have to be prepared for self-study

Contents

- Introduction
- Lectures at Tirana
- JCSE presentations
- Self-Study Slides
- Summary

Our SE course at Tirana: JCSE from 2007 to 2023

17 years

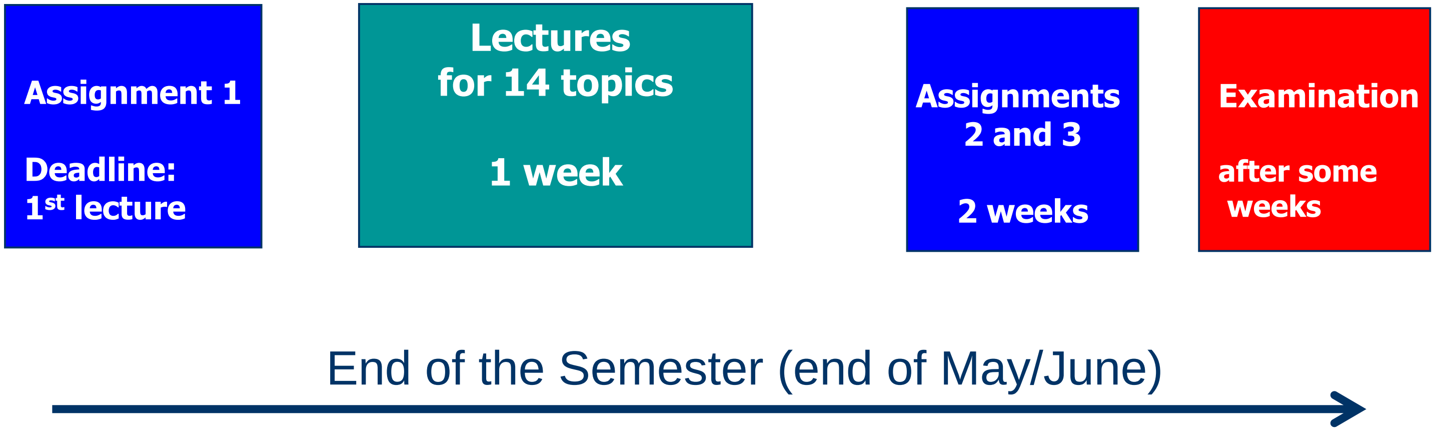
- From 2007 – 2015 HU together with Zoran Putnik (Novi Sad), full course (one week lectures, assignment, exams)
- 2016 – 2019 HU alone, full course (one week lectures)

------ Transformation due to pandemic situation ----

- 2020, 2021 Polytechnic University Tirana alone (online)
- 2022, 2023 Combined course: self study, lectures (HU) + assignments + examinations
- Self study: 10 topics (each week one topic)
- Lectures (face-to-face, one week): core areas of the course:

  - Software quality with testing and metrics
  - Subject to assignments and project work (testing tools, Testona/CTE) and core of exam questions

Schedule of the whole course at Tirana from 2007 till 2019



Assignment 1

Deadline:
1st lecture

Lectures
for 14 topics

1 week

Assignments
2 and 3

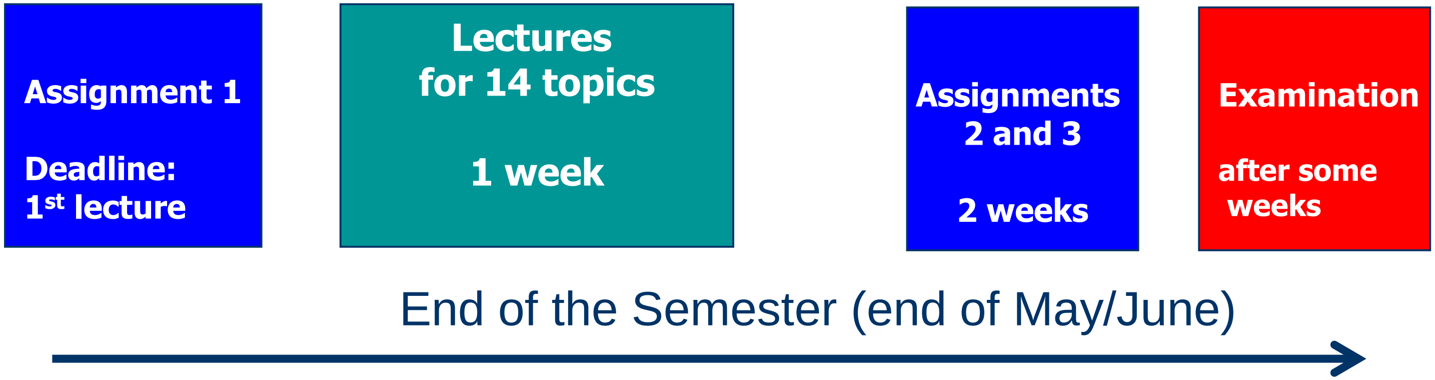
2 weeks

Examination

after some
weeks

End of the Semester (end of May/June)

Schedule of the whole course at Tirana from 2007 till 2019



Assignment 1

Deadline:
1st lecture

Lectures
for 14 topics

1 week

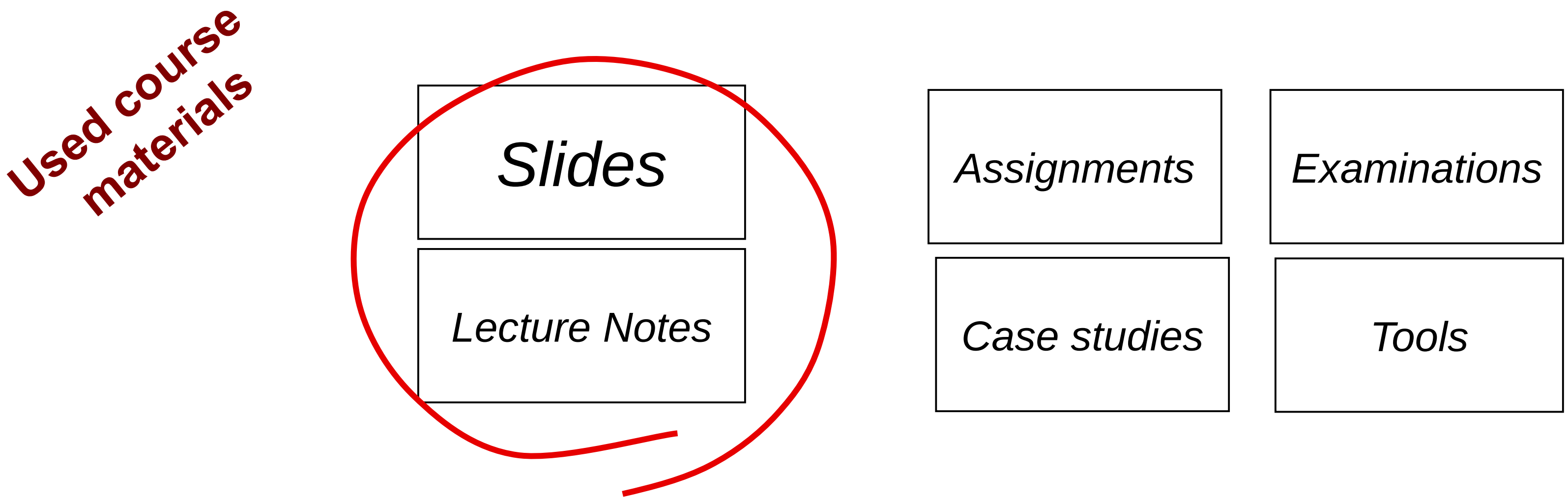
Assignments
2 and 3

2 weeks

Examination

after some
weeks

End of the Semester (end of May/June)



Used course materials

Slides

Lecture Notes

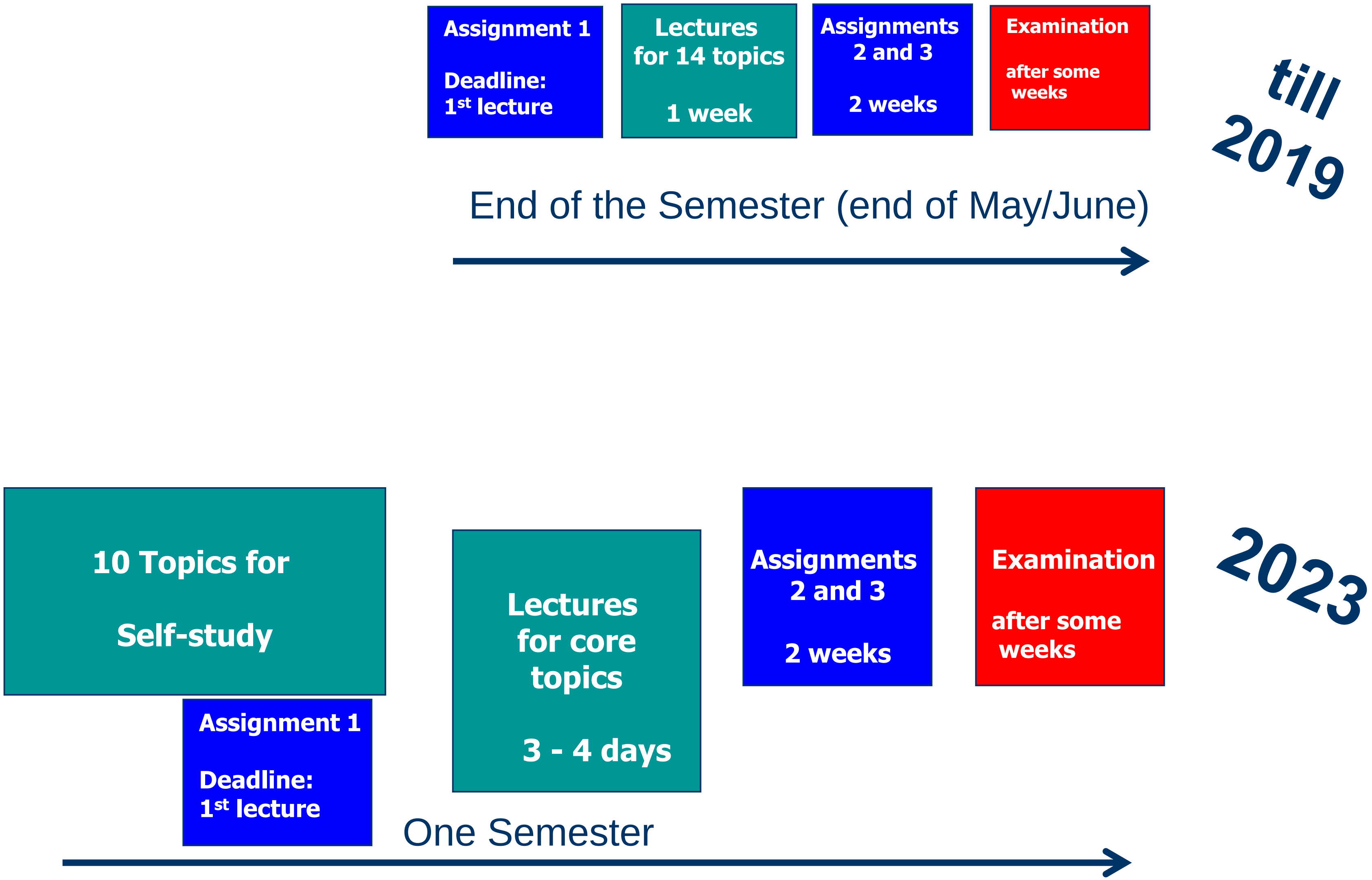
Assignments

Examinations

Case studies

Tools

Schedule of the whole course at Tirana till 2019 and in 2023



Assignment 1

Deadline:
1st lecture

Lectures for 14 topics

1 week

Assignments 2 and 3

2 weeks

Examination

after some weeks

till 2019

End of the Semester (end of May/June)

10 Topics for

Self-study

Assignment 1

Deadline:
1st lecture

Lectures for core topics

3 - 4 days

Assignments 2 and 3

2 weeks

Examination

after some weeks

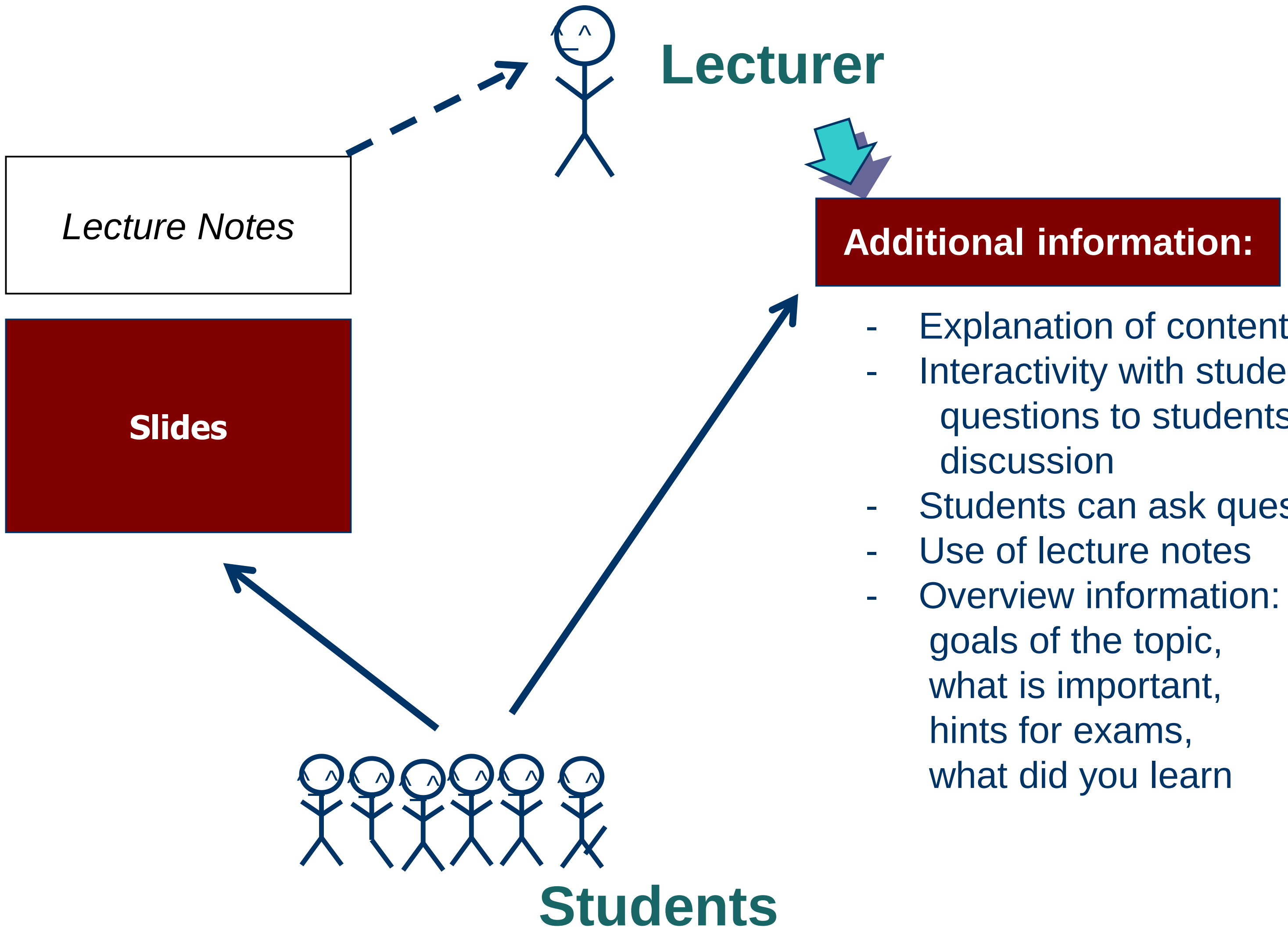
2023

One Semester

Contents

- Introduction
- Lectures at Tirana
- ➡ JCSE presentations
- Self-Study Slides
- Summary

Face-to-face lectures



Lecturer

Lecture Notes

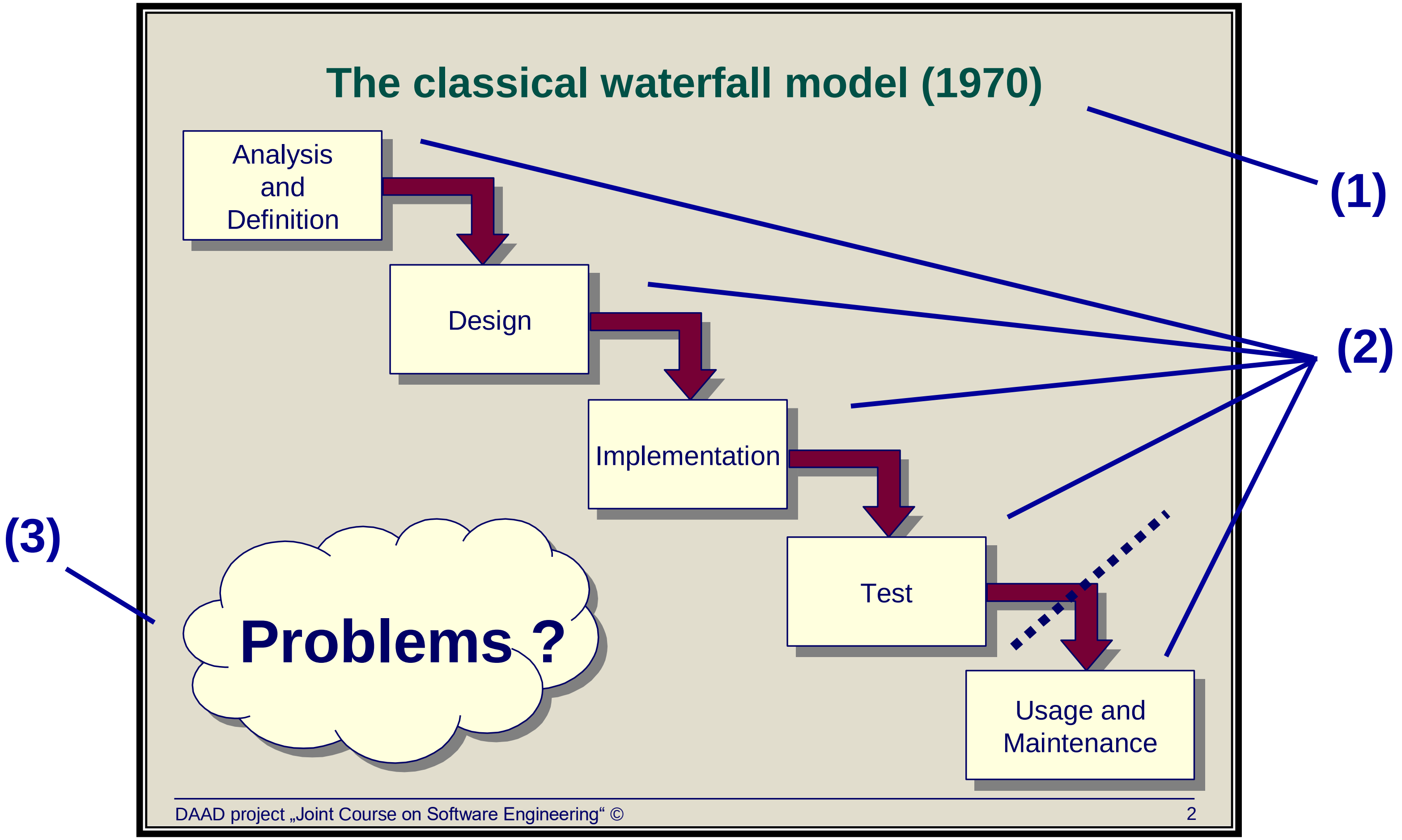
Slides

Students

Additional information:

- Explanation of contents
- Interactivity with students: questions to students, discussion
- Students can ask questions
- Use of lecture notes
- Overview information: goals of the topic, what is important, hints for exams, what did you learn

Slides alone can hardly be reused without additional information



The classical waterfall model (1970)

Analysis and Definition

Design

Implementation

Test

Usage and Maintenance

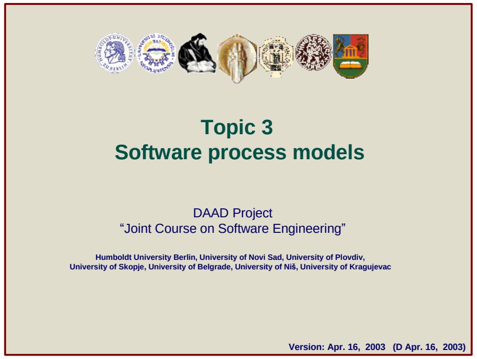
Problems ?

DAAD project „Joint Course on Software Engineering“ ©

2

The one above is just a slide with a headline (1), some boxes with notions (2), and a question in a cloud (3)

Slides – explained by lecture notes

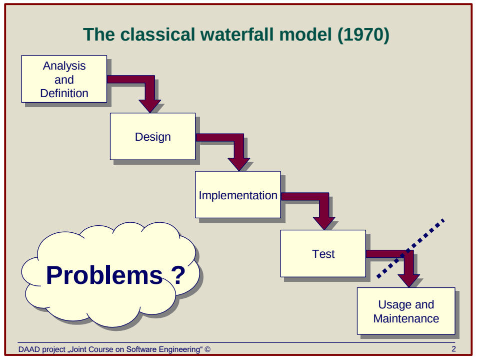


Topic 3
Software process models

DAAD Project
“Joint Course on Software Engineering”

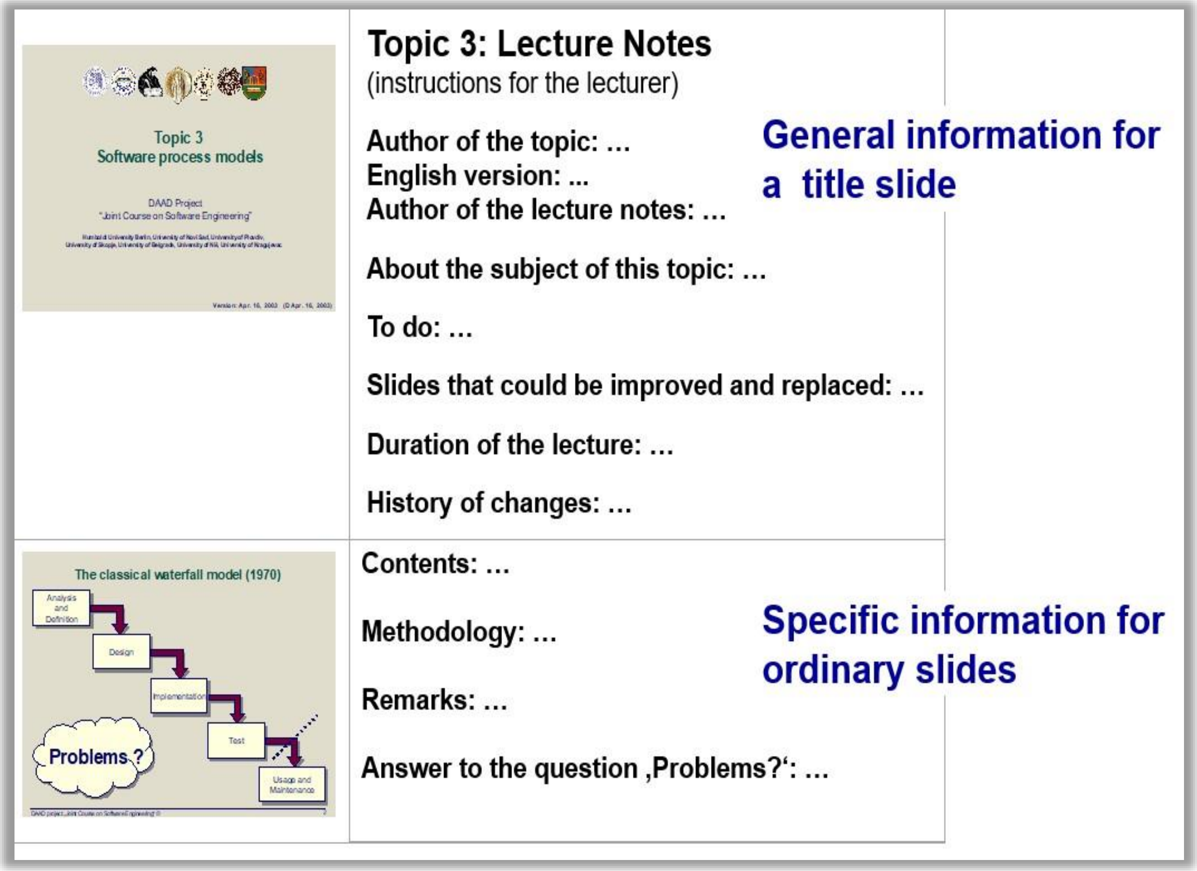
Humboldt University Berlin, University of Novi Sad, University of Plovdiv,
University of Skopje, University of Belgrade, University of Niš, University of Kragujevac

Version: Apr. 16, 2003 (D Apr. 16, 2003)



The classical waterfall model (1970)

DAAD project „Joint Course on Software Engineering“ © 2



Topic 3
Software process models

DAAD Project
“Joint Course on Software Engineering”

Version: Apr. 16, 2003 (D Apr. 16, 2003)

Topic 3: Lecture Notes
(instructions for the lecturer)

Author of the topic: ...
English version: ...
Author of the lecture notes: ...

About the subject of this topic: ...

To do: ...

Slides that could be improved and replaced: ...

Duration of the lecture: ...

History of changes: ...

General information for a title slide

The classical waterfall model (1970)
Analysis and Definition
Design
Implementation
Test
Usage and Maintenance
Problems ?

Contents: ...

Methodology: ...

Remarks: ...

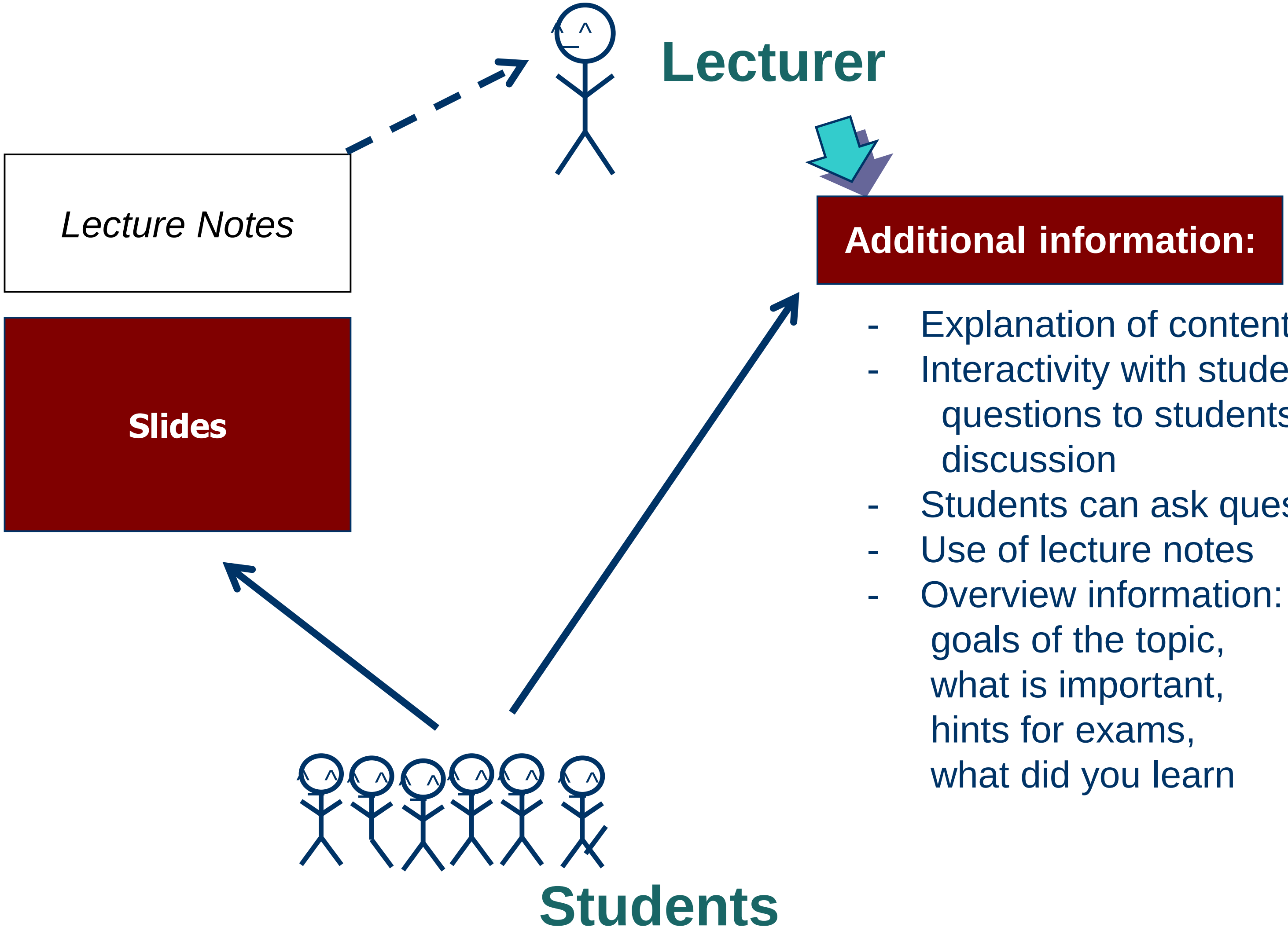
Answer to the question ‚Problems?‘: ...

Specific information for ordinary slides

Contents

- Introduction
- Lectures at Tirana
- JCSE presentations
- ➡ Self-Study Slides
- Summary

Face-to-face lectures



Lecturer

Lecture Notes

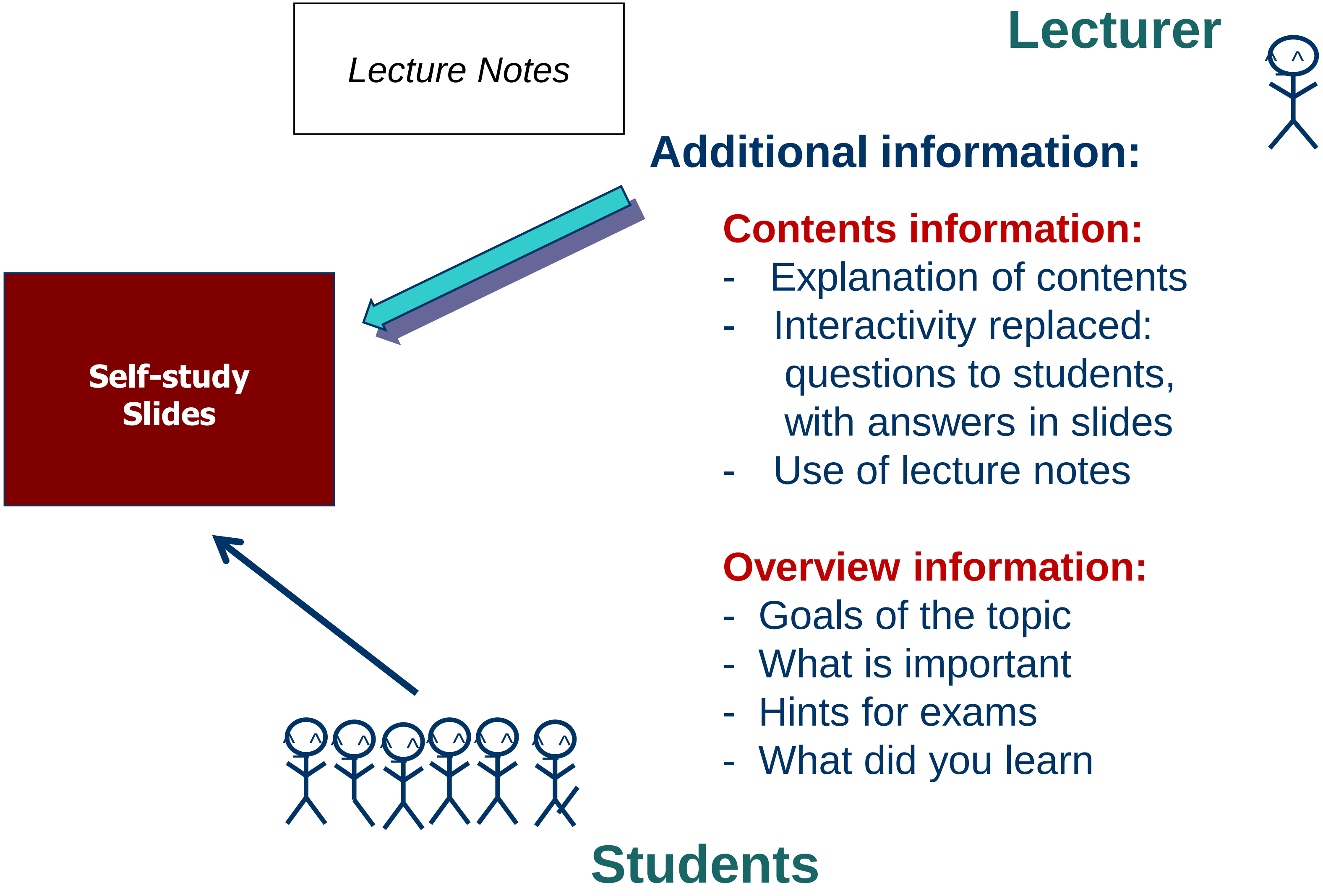
Slides

Additional information:

- Explanation of contents
- Interactivity with students: questions to students, discussion
- Students can ask questions
- Use of lecture notes
- Overview information:
  goals of the topic,
  what is important,
  hints for exams,
  what did you learn

Students

Self-study slides: enrichment of lecture slides



Lecture Notes

Lecturer

Additional information:

Contents information:

- Explanation of contents
- Interactivity replaced: questions to students, with answers in slides
- Use of lecture notes

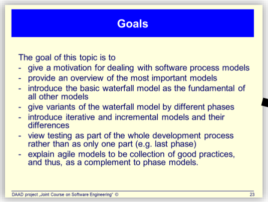
Overview information:

- Goals of the topic
- What is important
- Hints for exams
- What did you learn

Self-study Slides

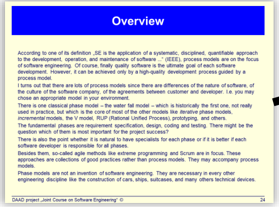
Students

Enrichment of traditional slides for self-study



Goals

The goal of this topic is to
- give a motivation for dealing with software process models
- provide an overview of the most important models
- introduce the basic waterfall model as the fundamental of all other models
- give variants of the waterfall model by different phases
- introduce iterative and incremental models and their differences
- view testing as part of the whole development process rather than as only one part (e.g. last phase)
- explain agile models to be collection of good practices, and thus, as a complement to phase models.



Overview

According to one of its definition „SE is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software ...“ (IEEE), process models are on the focus of software engineering. Of course, finally quality software is the ultimate goal of each software development. However, it can be achieved only by a high-quality development process guided by a process model.

I turns out that there are lots of process models since there are differences of the nature of software, of the culture of the software company, of the agreements between customer and developer. I.e. you may chose an appropriate model in your environment.

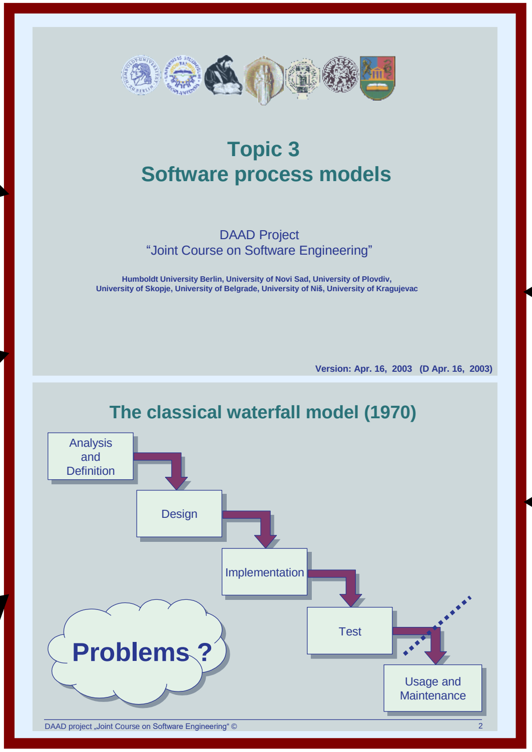
There is one classical phase model – the water fall model – which is historically the first one, not really used in practice, but which is the core of most of the other models like *iterative* phase models, *incremental* models, the V model, RUP (Rational Unified Process), prototyping, and others.

The fundamental phases are requirement specification, design, coding and testing. There might be the question which of them is most important for the project success?

There is also the point whether it is natural to have specialists for each phase or if it is better if each software developer is responsible for all phases.

Besides them, so-called agile methods like extreme programming and Scrum are in focus. These approaches are collections of good practices rather than process models. They may accompany process models.

Phase models are not an invention of software engineering. They are necessary in every other engineering discipline like the construction of cars, ships, suitcases, and many others technical devices.



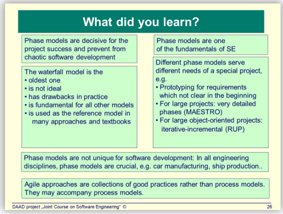
Topic 3
Software process models

DAAD Project
“Joint Course on Software Engineering”

Humboldt University Berlin, University of Novi Sad, University of Plovdiv,
University of Skopje, University of Belgrade, University of Niš, University of Kragujevac

Version: Apr. 16, 2003 (D Apr. 16, 2003)

The classical waterfall model (1970)



What did you learn?

Phase models are decisive for the project success and prevent from chaotic software development

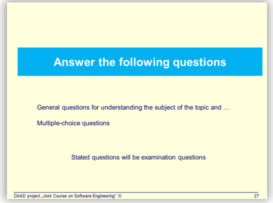
Phase models are one of the fundamentals of SE

The waterfall model is the
- oldest one
- is not ideal
- has drawbacks in practice
- is fundamental for all other models
- is used as the reference model in many approaches and textbooks

Different phase models serve different needs of a special project, e.g.
- Prototyping for requirements which not clear in the beginning
- For large projects: very detailed phases (MAESTRO)
- For large object-oriented projects: iterative-incremental (RUP)

Phase models are not unique for software development: In all engineering disciplines, phase models are crucial, e.g. car manufacturing, ship production..

Agile approaches are collections of good practices rather than process models. They may accompany process models.



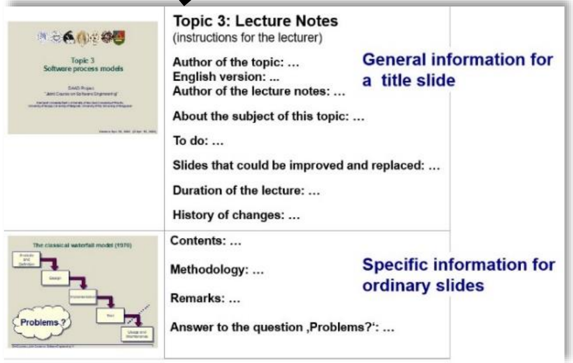
Answer the following questions

General questions for understanding the subject of the topic and …

Multiple-choice questions

Stated questions will be examination questions

Overview information



Topic 3: Lecture Notes
(instructions for the lecturer)

Author of the topic: …
English version: …
Author of the lecture notes: …

About the subject of this topic: …

To do: …

Slides that could be improved and replaced: …

Duration of the lecture: …

History of changes: …

Contents: …

Methodology: …

Remarks: …

Answer to the question ‚Problems?‘: …

General information for a title slide

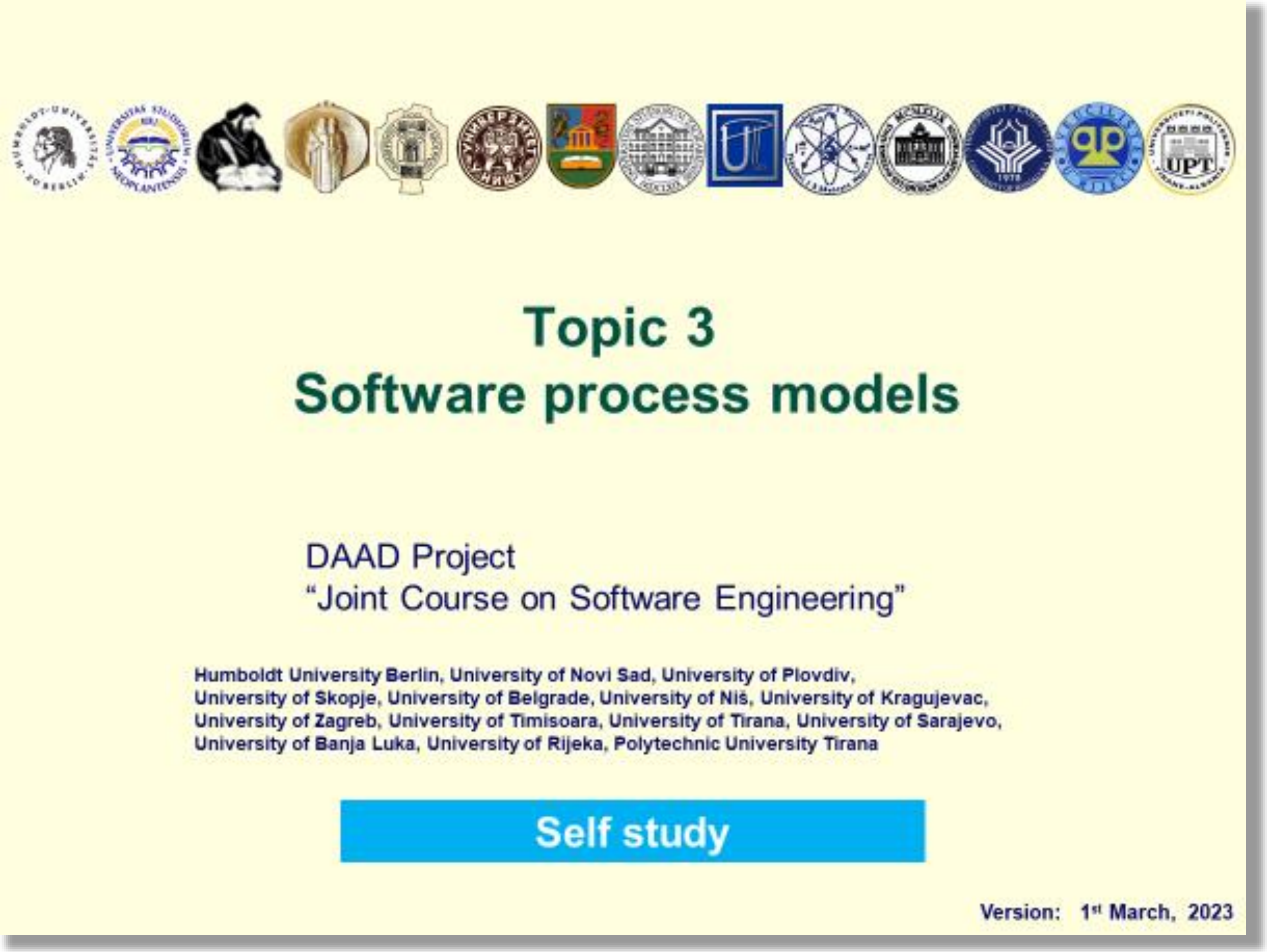
Specific information for ordinary slides

Contents information

Example:

Topic 3
Software process models

Topic 3: First slide



Topic 3
Software process models

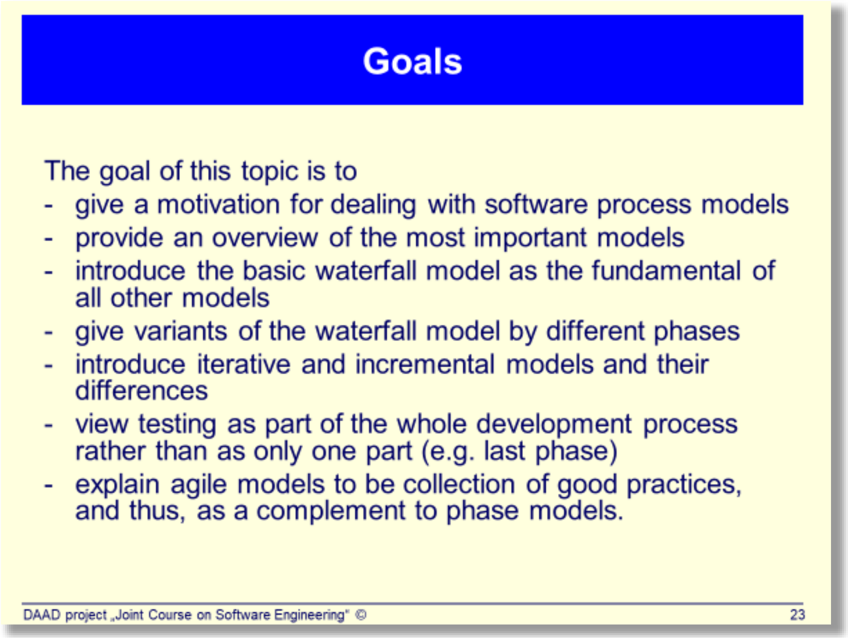
DAAD Project
"Joint Course on Software Engineering"

Humboldt University Berlin, University of Novi Sad, University of Plovdiv,
University of Skopje, University of Belgrade, University of Niš, University of Kragujevac,
University of Zagreb, University of Timisoara, University of Tirana, University of Sarajevo,
University of Banja Luka, University of Rijeka, Polytechnic University Tirana

Self study

Version: 1st March, 2023

In the beginning of the self-study slides

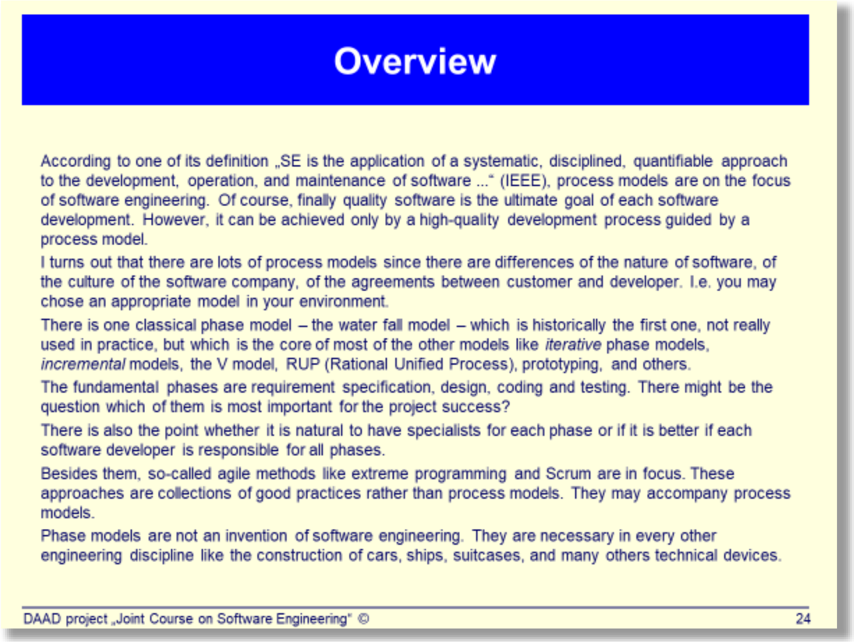


Goals

The goal of this topic is to

- give a motivation for dealing with software process models
- provide an overview of the most important models
- introduce the basic waterfall model as the fundamental of all other models
- give variants of the waterfall model by different phases
- introduce iterative and incremental models and their differences
- view testing as part of the whole development process rather than as only one part (e.g. last phase)
- explain agile models to be collection of good practices, and thus, as a complement to phase models.

DAAD project „Joint Course on Software Engineering" © 23



Overview

According to one of its definition „SE is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software ..." (IEEE), process models are on the focus of software engineering. Of course, finally quality software is the ultimate goal of each software development. However, it can be achieved only by a high-quality development process guided by a process model.

I turns out that there are lots of process models since there are differences of the nature of software, of the culture of the software company, of the agreements between customer and developer. I.e. you may chose an appropriate model in your environment.

There is one classical phase model – the water fall model – which is historically the first one, not really used in practice, but which is the core of most of the other models like *iterative* phase models, *incremental* models, the V model, RUP (Rational Unified Process), prototyping, and others.

The fundamental phases are requirement specification, design, coding and testing. There might be the question which of them is most important for the project success?

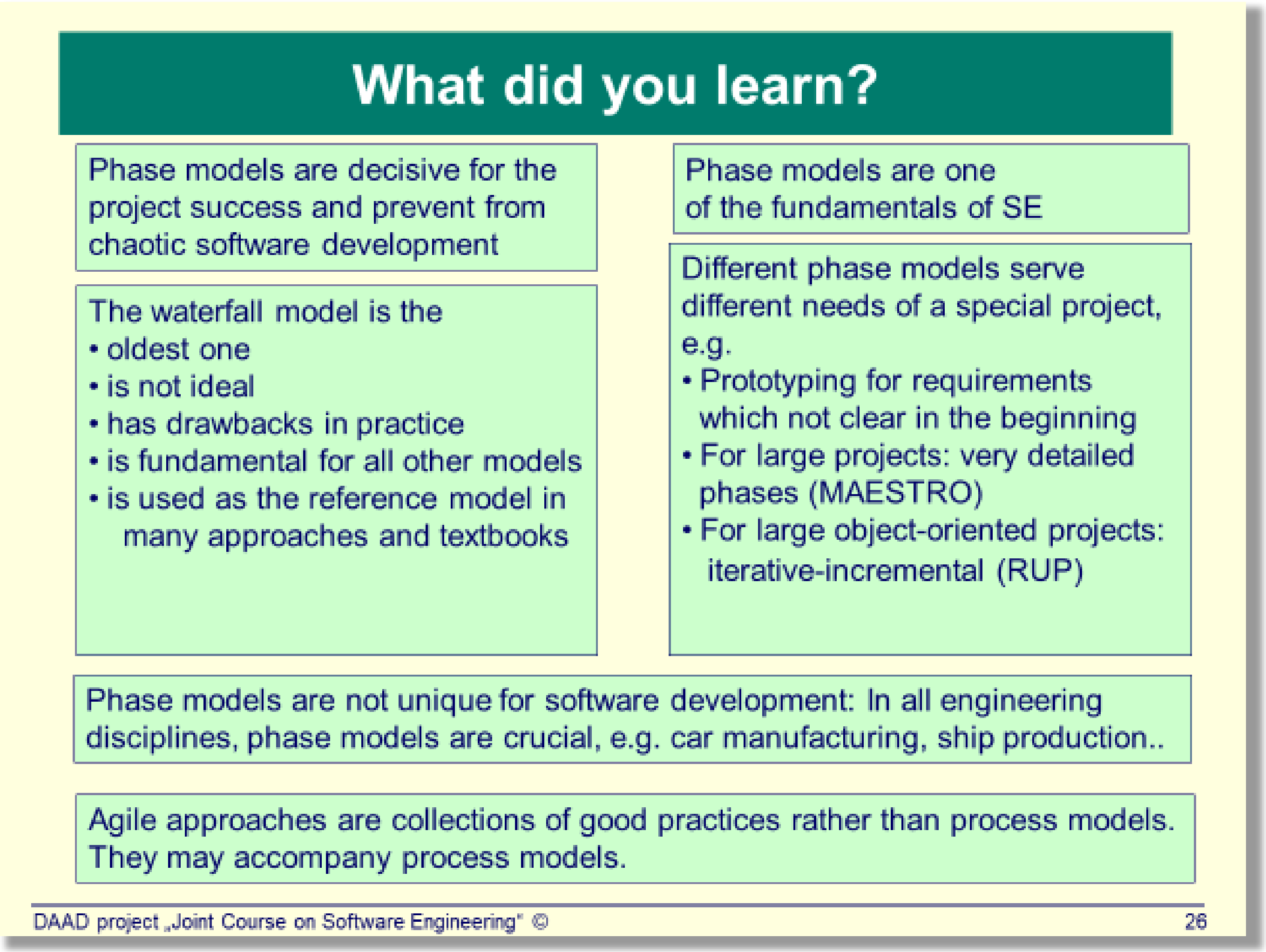
There is also the point whether it is natural to have specialists for each phase or if it is better if each software developer is responsible for all phases.

Besides them, so-called agile methods like extreme programming and Scrum are in focus. These approaches are collections of good practices rather than process models. They may accompany process models.

Phase models are not an invention of software engineering. They are necessary in every other engineering discipline like the construction of cars, ships, suitcases, and many others technical devices.

DAAD project „Joint Course on Software Engineering" © 24

Summary at the end



What did you learn?

Phase models are decisive for the project success and prevent from chaotic software development

The waterfall model is the

- oldest one
- is not ideal
- has drawbacks in practice
- is fundamental for all other models
- is used as the reference model in many approaches and textbooks

Phase models are one of the fundamentals of SE

Different phase models serve different needs of a special project, e.g.

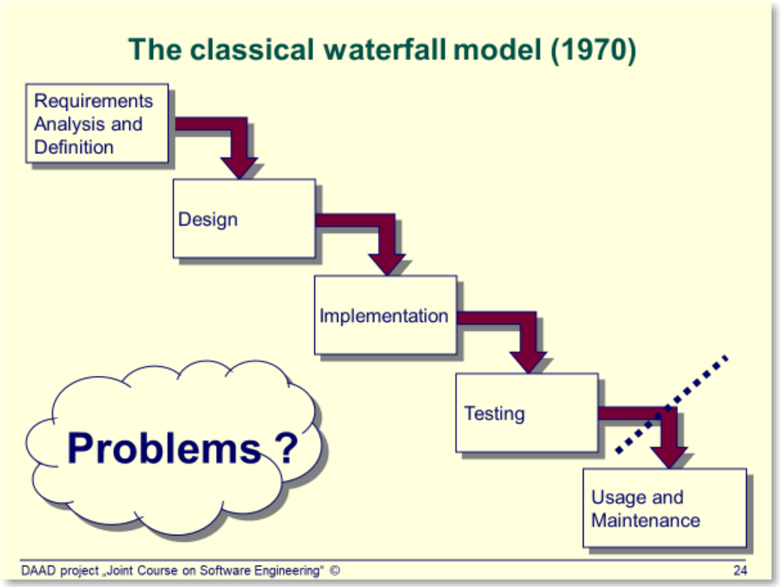
- Prototyping for requirements which not clear in the beginning
- For large projects: very detailed phases (MAESTRO)
- For large object-oriented projects: iterative-incremental (RUP)

Phase models are not unique for software development: In all engineering disciplines, phase models are crucial, e.g. car manufacturing, ship production..

Agile approaches are collections of good practices rather than process models. They may accompany process models.

Contents information

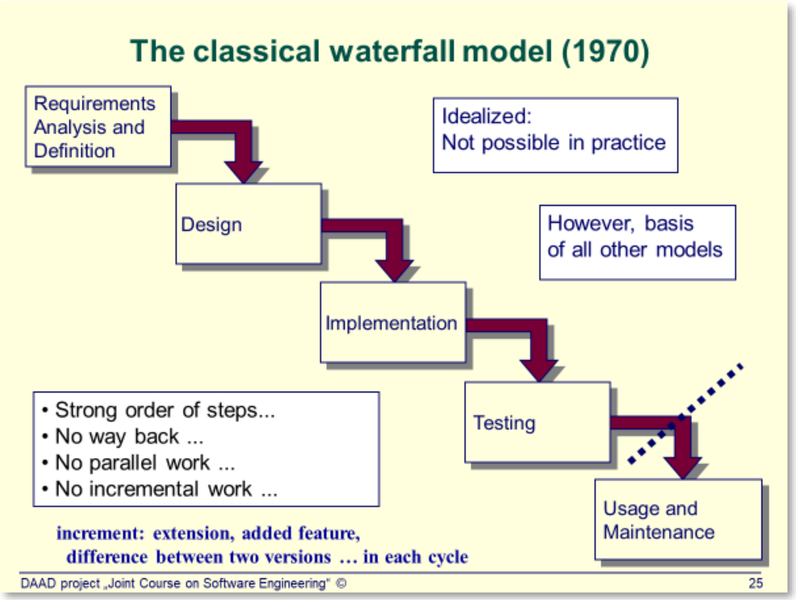
Information from lecture notes to slides



The classical waterfall model (1970)

DAAD project „Joint Course on Software Engineering" © 24

New slide for self-study: answer the question



The classical waterfall model (1970)

Idealized:
Not possible in practice

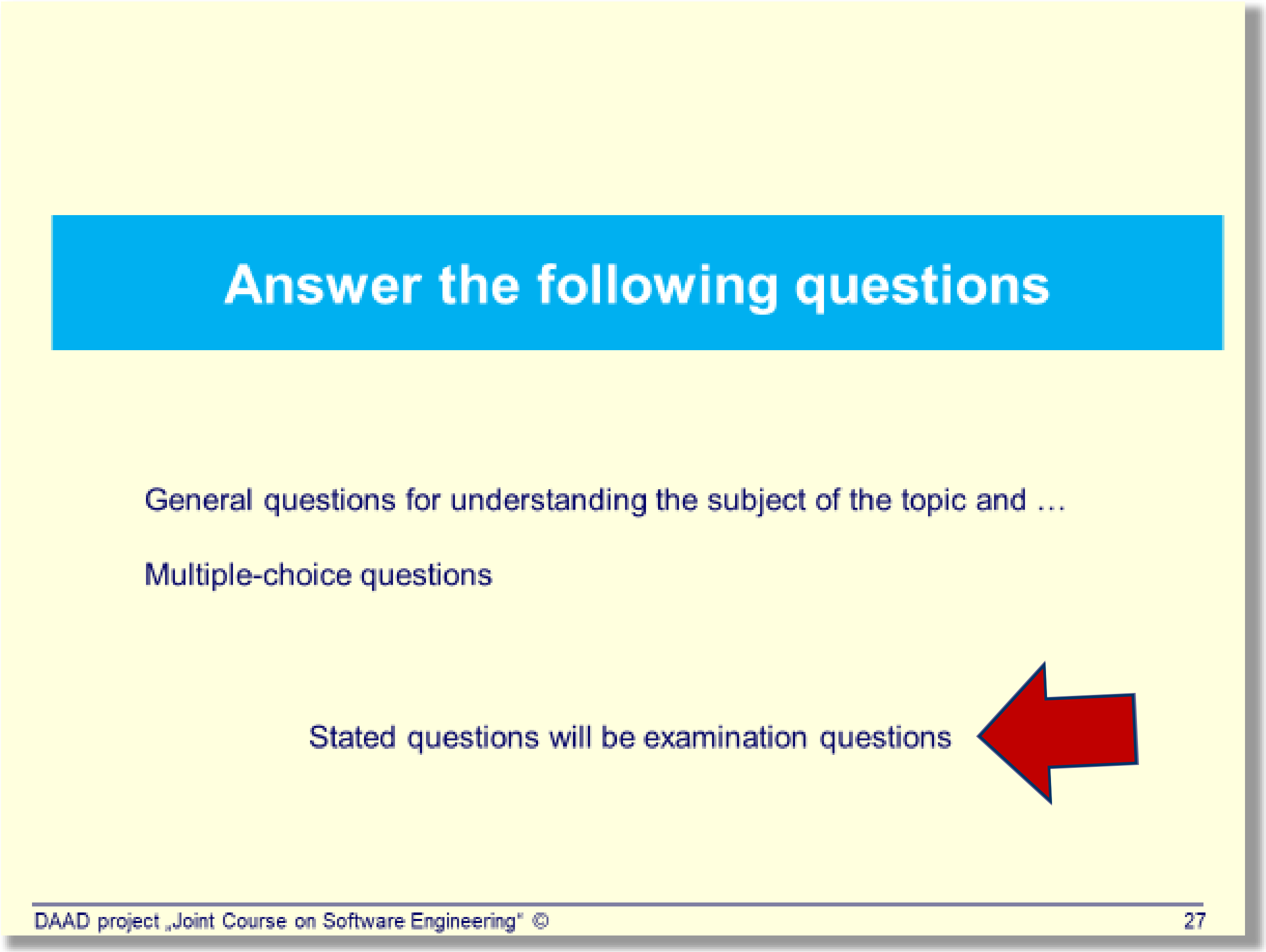
However, basis of all other models

- Strong order of steps...
- No way back ...
- No parallel work ...
- No incremental work ...

increment: extension, added feature, difference between two versions … in each cycle

DAAD project „Joint Course on Software Engineering" © 25

Preparation for examination



Answer the following questions

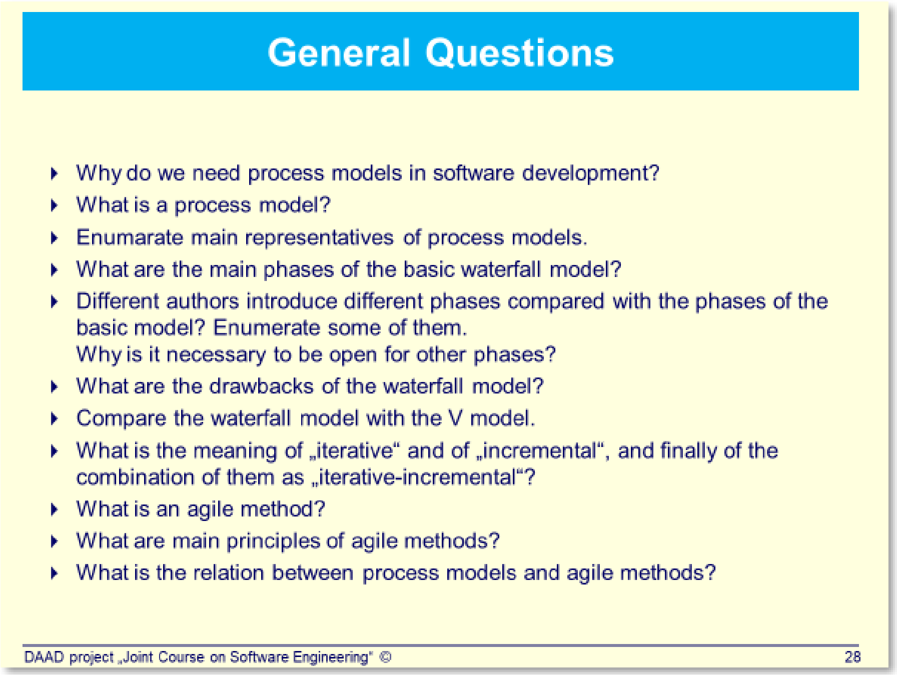
General questions for understanding the subject of the topic and …

Multiple-choice questions

Stated questions will be examination questions

DAAD project „Joint Course on Software Engineering" © 27

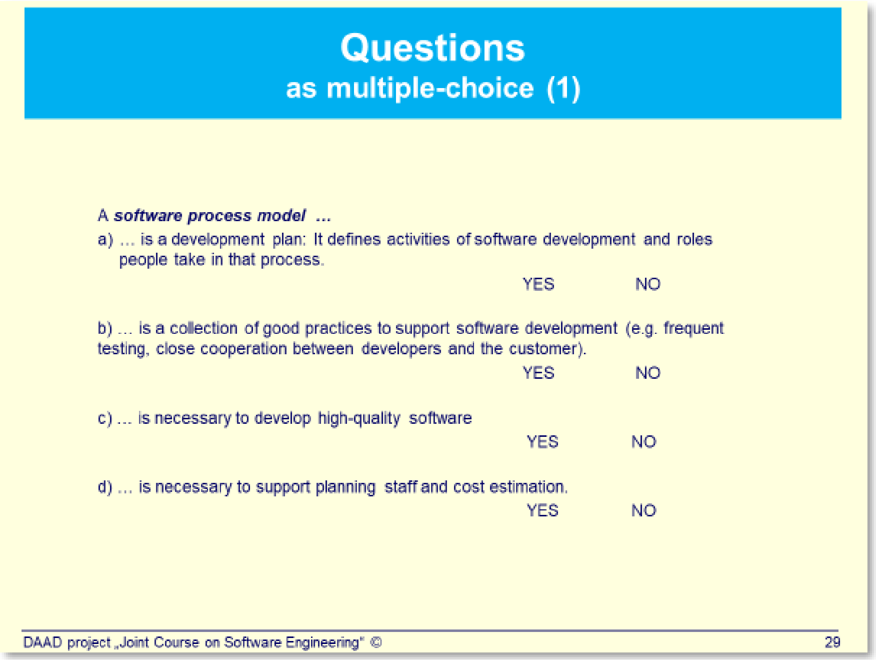
Examination questions: textual – multiple choice



General Questions

- Why do we need process models in software development?
- What is a process model?
- Enumarate main representatives of process models.
- What are the main phases of the basic waterfall model?
- Different authors introduce different phases compared with the phases of the basic model? Enumerate some of them.
  Why is it necessary to be open for other phases?
- What are the drawbacks of the waterfall model?
- Compare the waterfall model with the V model.
- What is the meaning of „iterative“ and of „incremental“, and finally of the combination of them as „iterative-incremental“?
- What is an agile method?
- What are main principles of agile methods?
- What is the relation between process models and agile methods?

DAAD project „Joint Course on Software Engineering“ © 28



Questions
as multiple-choice (1)

A ***software process model*** ...

a) ... is a development plan: It defines activities of software development and roles people take in that process.

YES NO

b) ... is a collection of good practices to support software development (e.g. frequent testing, close cooperation between developers and the customer).

YES NO

c) ... is necessary to develop high-quality software

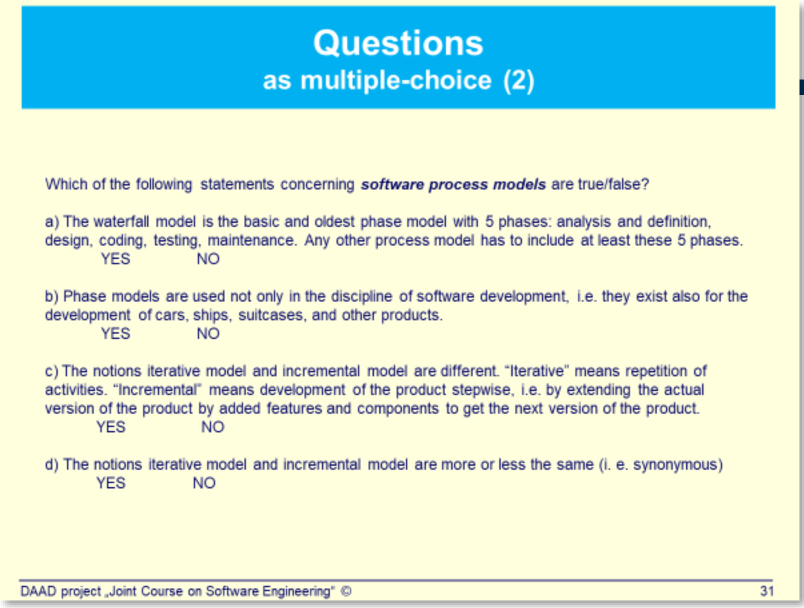
YES NO

d) ... is necessary to support planning staff and cost estimation.

YES NO

DAAD project „Joint Course on Software Engineering“ © 29

Multiple choice: five areas



Questions as multiple-choice (2)

Which of the following statements concerning ***software process models*** are true/false?

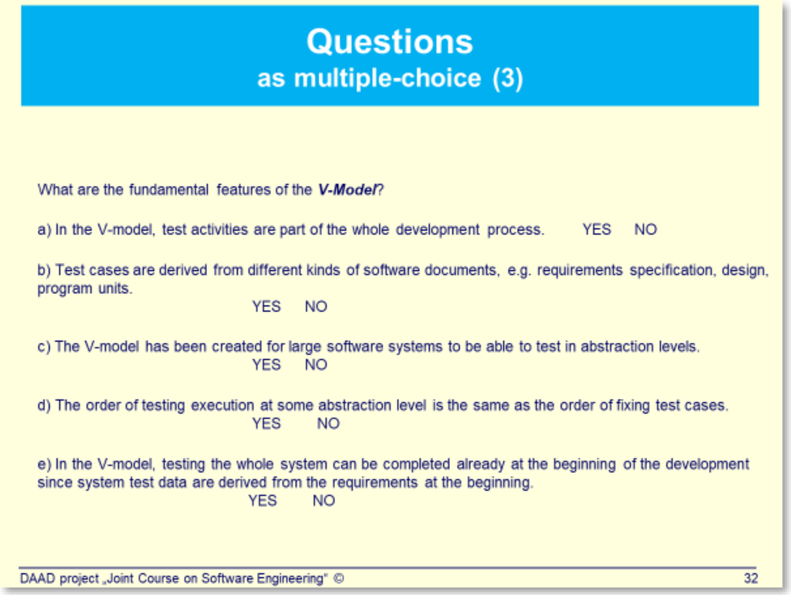
a) The waterfall model is the basic and oldest phase model with 5 phases: analysis and definition, design, coding, testing, maintenance. Any other process model has to include at least these 5 phases.
YES NO

b) Phase models are used not only in the discipline of software development, i.e. they exist also for the development of cars, ships, suitcases, and other products.
YES NO

c) The notions iterative model and incremental model are different. "Iterative" means repetition of activities. "Incremental" means development of the product stepwise, i.e. by extending the actual version of the product by added features and components to get the next version of the product.
YES NO

d) The notions iterative model and incremental model are more or less the same (i. e. synonymous)
YES NO

DAAD project „Joint Course on Software Engineering" © 31



Questions as multiple-choice (3)

What are the fundamental features of the ***V-Model***?

a) In the V-model, test activities are part of the whole development process. YES NO

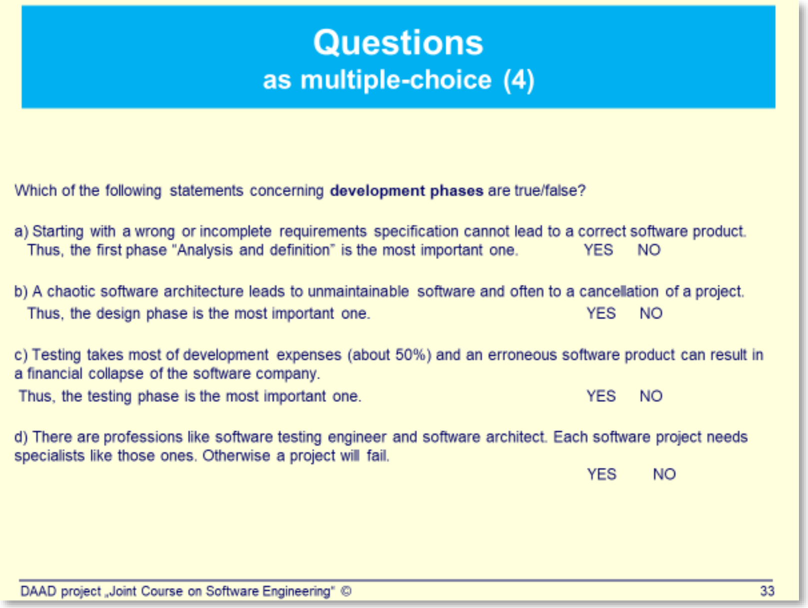
b) Test cases are derived from different kinds of software documents, e.g. requirements specification, design, program units.
YES NO

c) The V-model has been created for large software systems to be able to test in abstraction levels.
YES NO

d) The order of testing execution at some abstraction level is the same as the order of fixing test cases.
YES NO

e) In the V-model, testing the whole system can be completed already at the beginning of the development since system test data are derived from the requirements at the beginning.
YES NO

DAAD project „Joint Course on Software Engineering" © 32



Questions as multiple-choice (4)

Which of the following statements concerning **development phases** are true/false?

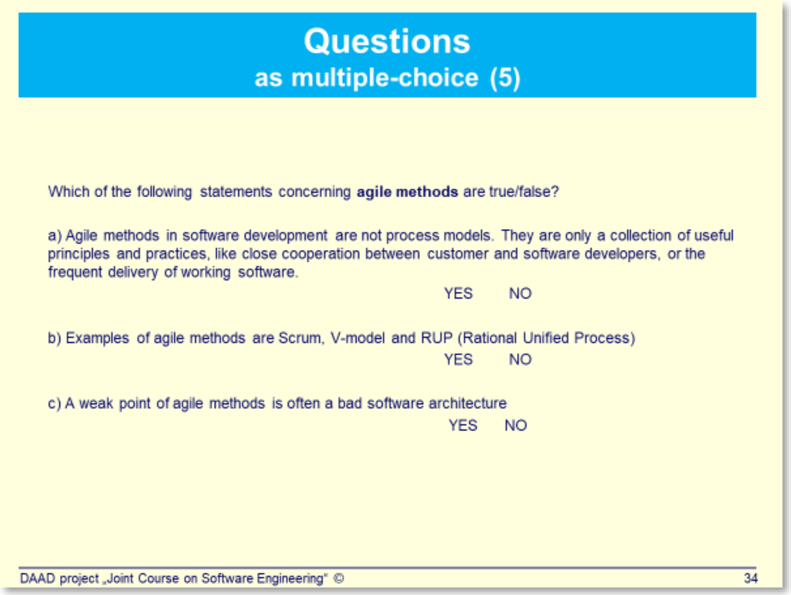
a) Starting with a wrong or incomplete requirements specification cannot lead to a correct software product. Thus, the first phase "Analysis and definition" is the most important one. YES NO

b) A chaotic software architecture leads to unmaintainable software and often to a cancellation of a project. Thus, the design phase is the most important one. YES NO

c) Testing takes most of development expenses (about 50%) and an erroneous software product can result in a financial collapse of the software company. Thus, the testing phase is the most important one. YES NO

d) There are professions like software testing engineer and software architect. Each software project needs specialists like those ones. Otherwise a project will fail.
YES NO

DAAD project „Joint Course on Software Engineering" © 33



Questions as multiple-choice (5)

Which of the following statements concerning **agile methods** are true/false?

a) Agile methods in software development are not process models. They are only a collection of useful principles and practices, like close cooperation between customer and software developers, or the frequent delivery of working software.
YES NO

b) Examples of agile methods are Scrum, V-model and RUP (Rational Unified Process)
YES NO

c) A weak point of agile methods is often a bad software architecture
YES NO

DAAD project „Joint Course on Software Engineering" © 34

Contents

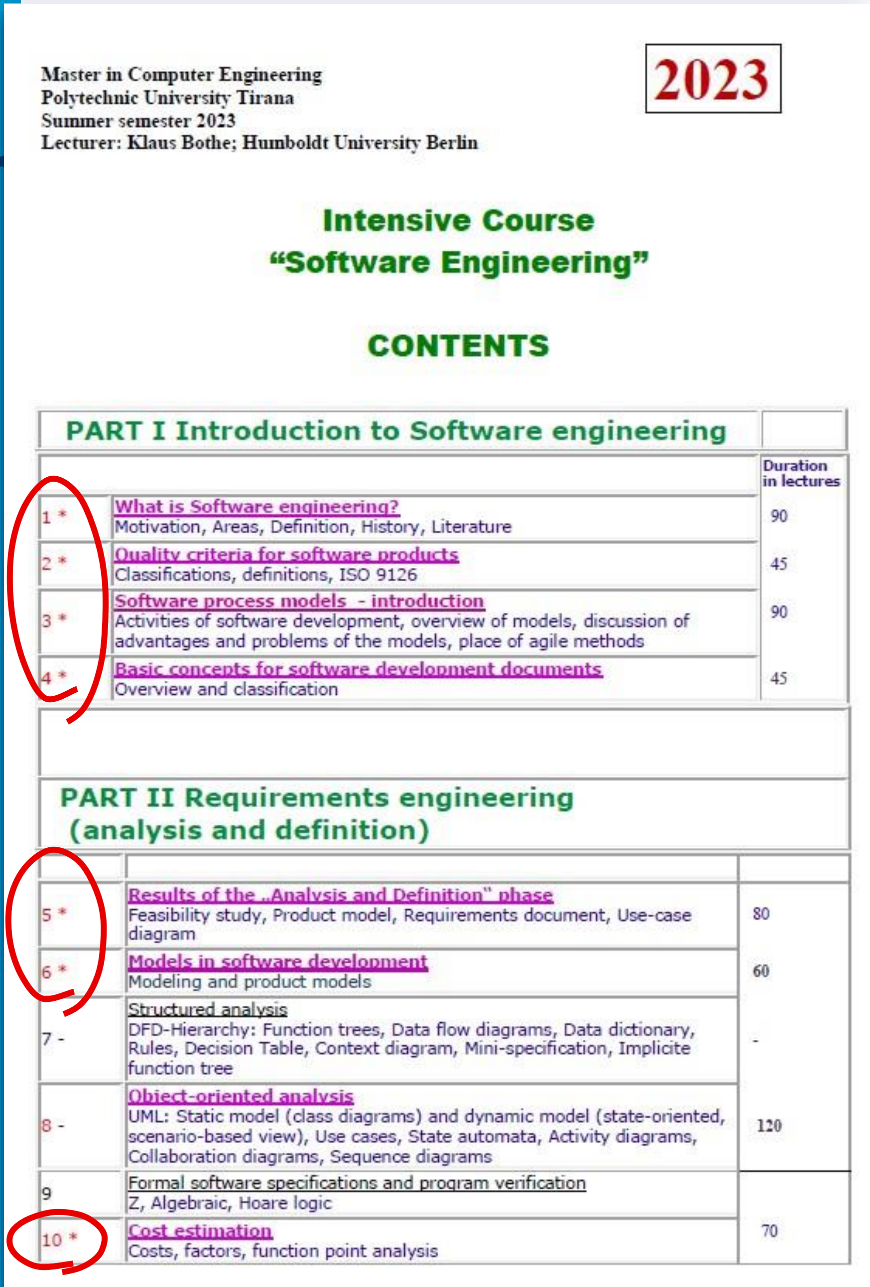
- Introduction
- Lectures at Tirana
- JCSE presentations
- Self-Study Slides
- ➡ Summary

JCSE at Polytechnic University Tirana

In 2023, from 20 JCSE topics handled as

- self-study slides: 10
- face-to-face presentations: 4

Slides for self-study and lectures (Tirana, 2023)



Master in Computer Engineering
Polytechnic University Tirana
Summer semester 2023
Lecturer: Klaus Bothe; Humboldt University Berlin

2023

Intensive Course
"Software Engineering"

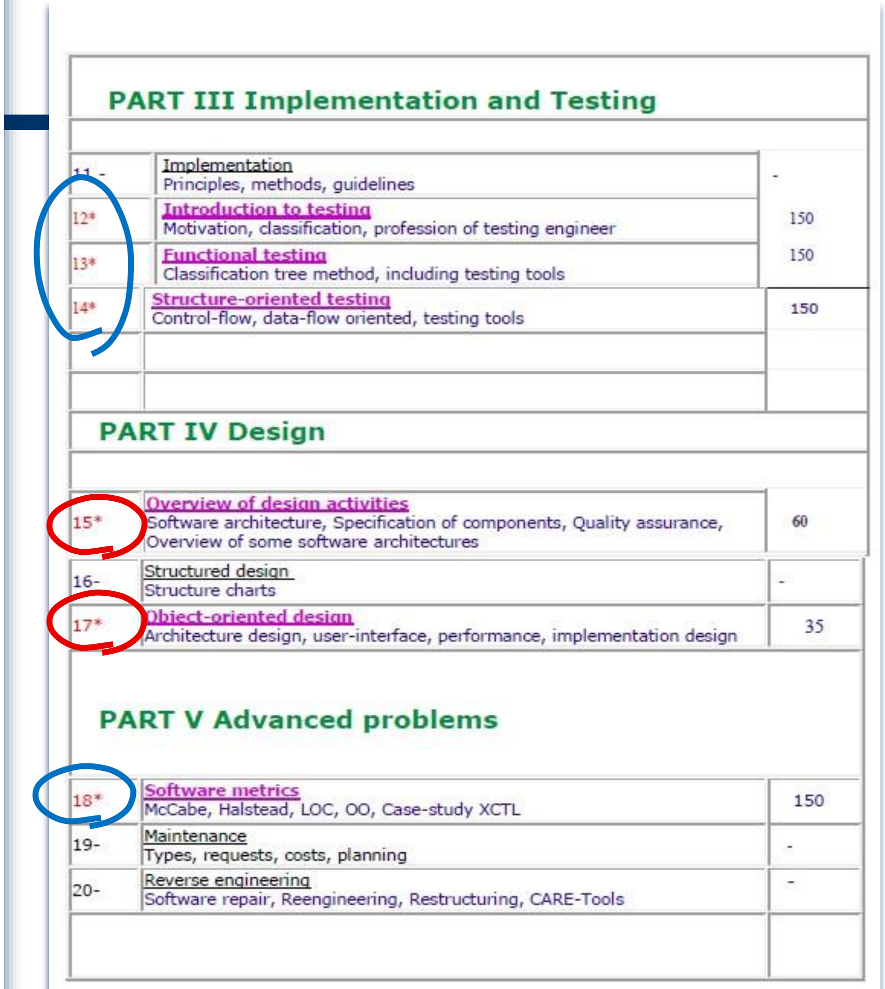
CONTENTS

PART I Introduction to Software engineering

		Duration in lectures
1 *	What is Software engineering? Motivation, Areas, Definition, History, Literature	90
2 *	Quality criteria for software products Classifications, definitions, ISO 9126	45
3 *	Software process models - introduction Activities of software development, overview of models, discussion of advantages and problems of the models, place of agile methods	90
4 *	Basic concepts for software development documents Overview and classification	45

PART II Requirements engineering (analysis and definition)

5 *	Results of the „Analysis and Definition" phase Feasibility study, Product model, Requirements document, Use-case diagram	80
6 *	Models in software development Modeling and product models	60
7 -	Structured analysis DFD-Hierarchy: Function trees, Data flow diagrams, Data dictionary, Rules, Decision Table, Context diagram, Mini-specification, Implicite function tree	-
8 -	Object-oriented analysis UML: Static model (class diagrams) and dynamic model (state-oriented, scenario-based view), Use cases, State automata, Activity diagrams, Collaboration diagrams, Sequence diagrams	120
9	Formal software specifications and program verification Z, Algebraic, Hoare logic	
10 *	Cost estimation Costs, factors, function point analysis	70



PART III Implementation and Testing

11 -	Implementation Principles, methods, guidelines	-
12*	Introduction to testing Motivation, classification, profession of testing engineer	150
13*	Functional testing Classification tree method, including testing tools	150
14*	Structure-oriented testing Control-flow, data-flow oriented, testing tools	150

PART IV Design

15*	Overview of design activities Software architecture, Specification of components, Quality assurance, Overview of some software architectures	60
16-	Structured design Structure charts	-
17*	Object-oriented design Architecture design, user-interface, performance, implementation design	35

PART V Advanced problems

18*	Software metrics McCabe, Halstead, LOC, OO, Case-study XCTL	150
19-	Maintenance Types, requests, costs, planning	-
20-	Reverse engineering Software repair, Reengineering, Restructuring, CARE-Tools	-

○ Self-study

○ Lectures

Feedback of students in 2023

Question to students: How useful are the new slides for self-study?
(Which parts are understandable, which parts have to be improved …)

> “We would like to add our feedback for the self-study slides.
> We all found them understandable, and easy to learn,
> while also being informative and educative.
> None of us found any difficulty in understanding the material,
> both with the self-study lessons,
> as well as with the lessons in class.”

😉

Unfortunately, only one feedback of all teams (team 7 – the best one)

Polytechnic University Tirana: intensive course JCSE since 2007

532 students in 15 years (2007 – 2023, without 2020, 2021)

2007

19 – 24 March 2007

17 students

4th semester

Master

2008

21 – 26 April 2008

32 students

1st semester (15)

3rd semester (17)

Master

2009

22 – 27 June 2009

17 students

2nd semester

Master Tirana (11)

6th semester

Bachelor Struga (6)

2010

31 May – 5 June 2010

14 students

2nd semester

Master Tirana

2011

24 – 29 2011

25 st

2nd semester

Master Tirana

2012

semester

Master Tirana

2013

Master Tirana

2014

2 – 6 June

nts

er

Master Tirana

Huge problem in 2023: increasing number of students (exams, assignments)

Effort to check assignments and exams and problems with copying (exams + assignments)

2015: **18 Master students**

2016: 32 Master students

2017: **35 Master** students

2018: 35 Master students

2019: 30 Master students

2022: 45 Master students

2023: **67/55** Master students

Thank you