



# Assignments in the Joint Course on Software Engineering

Kay Schützler, Zoran Budimac

*3rd Workshop Software Engineering Education and Reverse Engineering,  
Ohrid, Macedonia, 2003*

## The perfect/ideal assignment

- ▶ **Motivates students**
  - Real life tasks
  - Managable with a reasonable amount of effort
- ▶ **Repeats and practices contents**
  - Allows a practical view on theoretically presented lecture contents
- ▶ **Allows autonomy/independence**
  - Allows students to develop some creativity in their solutions

## Aims of assignments in our SE course

- ▶ Put lecture contents into case study context
  - Show applications of lecture contents
- ▶ Allow autonomy/independence
  - Include creative elements and discussions
  - Allow more than one “correct” solution
- ▶ Prepare for real life tasks
  - Not too small and too well-formed examples
- ▶ Encourage team work
  - Through separable tasks and/or discussion elements

## Assignments and course topics

### Part I: Introduction to software engineering

- 1. What is software engineering
- 2. Quality criteria for software products
- 3. Software process models
- 4. Basic concepts for software development documents

### Part II: Requirements engineering

- 5 Results of the “analysis and definition” phase
- 6 Cost estimation
- 7. Function-oriented view
- 8. Data-oriented view
- 9. Rule-oriented view
- 10 Structured analysis
- 11. State-oriented view
- 12. Scenario-oriented view
- 13 Object-oriented analysis
- 14 Formal software specification and program verification

Kay Schützler,  
Zoran Budimac:  
Assignments in the  
SE course

### Part III: Software Design

- 15. Overview of design activities
- 16. Structured design
- 17. Object-oriented design

### Part IV: Implementation and testing

- 18. Implementation
- 19 Systematic testing
- 20 Functional testing

### Part V: Advanced problems

- 21 Software metrics
- 22. Maintenance
- 23. Reverse engineering
- 24. Quality of software development process and its standardisation
- 25. Introduction to software ergonomics
- 26. User manuals
- 27. Project management
- 28. Configuration and version management

## Assignments in detail: Topic 5: Results of "analysis and definition" phase

- ▶ Review requirements specification of case study  
"Seminar Organisation"
  - Become familiar with lecture's main case study
  - Get to know assignments style
  - Discover advantages of team work („Four (to six) eyes see more than two...“)

### Slide from Topic 5

## Review

- ▶ **Background:**  
arbitrary SW documents (e. g. requirements specification) have to be equally precise as programs
- ▶ **Method (Review):**  
Inspection of the documents by a group of evaluators by static reading the document  
(remark: 2 – 5 evaluators + author of the document)
- ▶ **Process:**
  1. Preparation of the participants
  2. Review meeting of the group:  
sequentially or by points of emphasis
  3. Produce a protocol

IEEE Std 1028-1988,  
Standard for Software  
Reviews and Audits

## Review Protocol (contents scheme)

Document:  
Participants:  
Leader:  
Protocol:  
Date, time of the meeting:

---

1. Summary
2. Problems of the document
  - 2.1 inaccuracies
  - 2.2 errors
  - 2.3 missing information
3. Remarks concerning the structure of the document
4. Remarks concerning the review meeting  
(preparation of the participants, length of the meeting, points of emphasis)

## Discussion of solutions (1)

Taken from:  
<http://www.informatik.hu-berlin.de/swt/intkoop/se/assignments/solutions/>

Solution: 1. Review the requirements document 'Seminar organization'

Authors: Zoran Budimac, Kay Schützler  
Date: Dec. 17, 2002

Inaccuracies	Discussion	Observed by
D60: "If a lecturer conducts a seminar, this information should be saved." (Where and to what extent?)	OK	Students - Ger
F40: Pre-condition: "ask for client's, company's, and lecturer's opinion". (Is this pre-condition or triggering event?)	Pre-condition could be modified to "client's, company's, and lecturer's suggestion" Or: "Isn't that a triggering event rather than a pre-condition?" (Kay)	Students - Ger
F30: Description: ... evaluations collection (Can it be done by the software?)	Probably not :-)	Students - Ger, Industry - Yu
Version description. In version 2.2 'F115 added'. (What happened to this function later? There is no trace of it)	It should be stated more clearly that v. 3.0 reorganized and renumbered functions.	Industry - Yu
F80: For every presentation participant with following data: ... conductor. (Who/what is conductor?)	Lecturer or presentation custodian? Probably custodian (which as well might be the lecturer him/herself.)	Industry - Yu
Errors	Discussion	Observed by
F23: Post condition success: company got the bill and the registration form. (Registration form?)	Registration proof is probably the right term.	Students - Ger

## Discussion of solutions (2)

F30: Actor: presentation custodian. (Lecturer is missing)	OK	Industry - Yu
D60: If a lecturer conducts a seminar ... (Shouldn't this be D51, because it is logically close to D50? Moreover D60 somehow should have its equivalence as LD60?)	2 <sup>nd</sup> : Not necessarily needed, because the second document is a more detailed version of the first. So D60 may be seen as an addition to D50. (But, inconsistency again! Why is information of D60 not part of D50?)	Industry - Yu
F23: It is an extension of F20 (LF20) and is not traceable to LF20.	It is not necessary for everything in functional specification to be traceable to preliminary requirements specifications.	Industry - Yu
<b>Missing information</b>	<b>Discussion</b>	<b>Observed by</b>
Several parts of document too short (e.g. 2, 7, 9, 10)	OK	Students - Ger
Missing part with test-cases	OK	Students - Ger
Missing Glossary (except the one in preliminary specifications)	OK	Students - Ger, Industry - Yu
Version info are not complete. (What happened before v. 2.2. What is changed in v 2.3?)	Lost somewhere. Check with Natasa.	Industry - Yu
Preliminary spec. 1. Goals: ... to manage seminars, clients and lecturers (where are seminar presentations?)	They are self-contained in seminars, based on the assumptions that no one would manage 'dead' seminars without presenting them.	Industry - Yu
Preliminary spec. Some important functions are missing (billing, cancelling, checking out, statistical reports,...)	It is too early to mention these specific functions in preliminary specifications. They are contained somewhere in mentioned functions.	Industry - Yu
F90 (queries): missing information about data that should be produced	Too early for such details in this document.	Industry - Yu
F90 (queries): (Are these two the only possible queries?)	No. But others of this kind are not defined precisely. Maybe it is an indirect requirement to make the internal database queryable.	Zjb

## Assignments in detail: Topic 6: Cost estimation

- ▶ Apply Function Points method to case study "Seminar Organisation"
  - Discover differences between theory and practice
  - Learn how to deal with imprecise (verbal) requirements specifications

## Assignments in detail: Topic 10: Structured analysis

- ▶ Review product model of case study "Seminar Organization"
  - Examine example Data Flow Diagrams and Data Dictionaries
  - Discover advantages and disadvantages of Structured Analysis
  - Again deal with imprecise (verbal) and additionally more precise (Semi-formal) documents

## Assignments in detail: Topic 13: Object-oriented analysis

- ▶ Derive a use case diagram and a class diagram from a problem description
  - Understand (and read carefully) verbal documents of other authors
  - Practice recognition of object oriented entities in texts
  - Manage different point of views (text interpretations) between team members

## Assignments in detail: Topic 14: Formal software specification

- ▶ Formal specification techniques: Algebraic specification and specification with Z
  - Retrieve a deeper understanding of lecture examples by modifying them
  - Discover the preciseness of formal specifications
  - Practice to think of all possible cases for the specified entities

## Assignments in detail: Topic 19: Systematic testing

- ▶ Apply regression testing tool ATOS to a small example programme
  - Get to know the concept of regression tests
  - Derive sensible test cases for undocumented software
  - Understand the concept behind the tool

## Assignments in detail: Topic 20: Functional testing

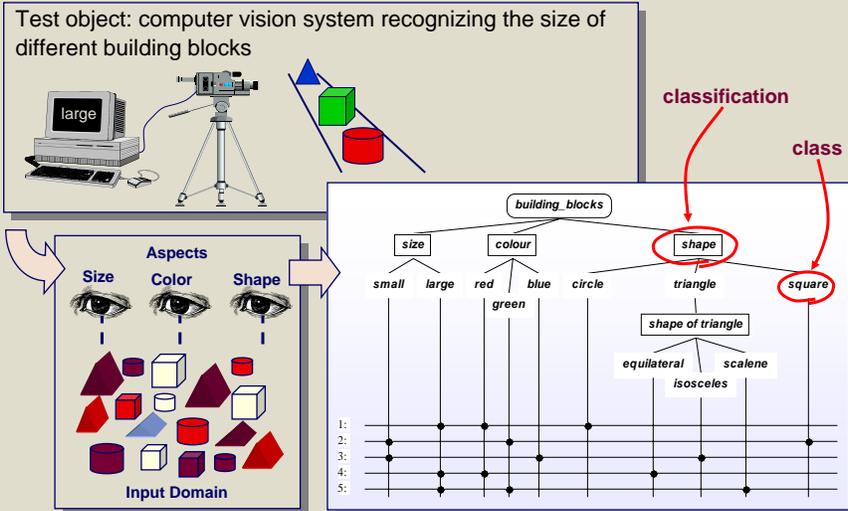
- ▶ Build a classification tree for one business process of case study “Seminar Organisation”
  - Practice classification tree method
  - Use the method’s standard tool (CTE: Classification Tree Editor)
  - Again derive sensible test cases (but now from the requirements specification, not from a program)

Slide from Topic 20

## Classification tree method (orig. developed by Daimler-Chrysler)

- ▶ Steps in Classification Tree method:
  1. Choose object i.e. procedure or sub-system;
  2. Determine the input domain of the object;
  3. Determine the aspects for testing;
  4. Form classifications by partitioning the input domain into classes according to these aspects;
  5. Repeat 3 and 4 as appropriate and form classification tree;
  6. Form combination table from the tree;
  7. Mark test cases and form a text version for generation of test sets;
  8. Define test sets;
  9. Derive expected output from the specification;
  10. Perform test and evaluate output;
  11. Check program coverage.

## Classification tree method: example



## Use Case (FP20): Booking: from registration to booking (1)

**Use case:** booking: from registration to booking

- ▶ **Goal:** registration notification and sending invoice to the client
- ▶ **Category:** primary
- ▶ **Preconditions:** -
- ▶ **Post condition success:** client is notified
- ▶ **Post condition failure:** notification to clients that presentation is fully booked, or does not exist, or a booking for the client has been already made
- ▶ **Actor:** client manager, client, company
- ▶ **Triggering event:** client registration is available

## Use Case (FP20): Booking: from registration to booking (2)

► **Description:**

- (1) client data retrieval (1 complex function)
- (2) presentation verification (1 middle function)
- (3) booking undertaking (1 simple function)
- (4) registration notification and sending invoice (1 simple function)
- (5) sending invoice copy to the accounts department (1 simple function)

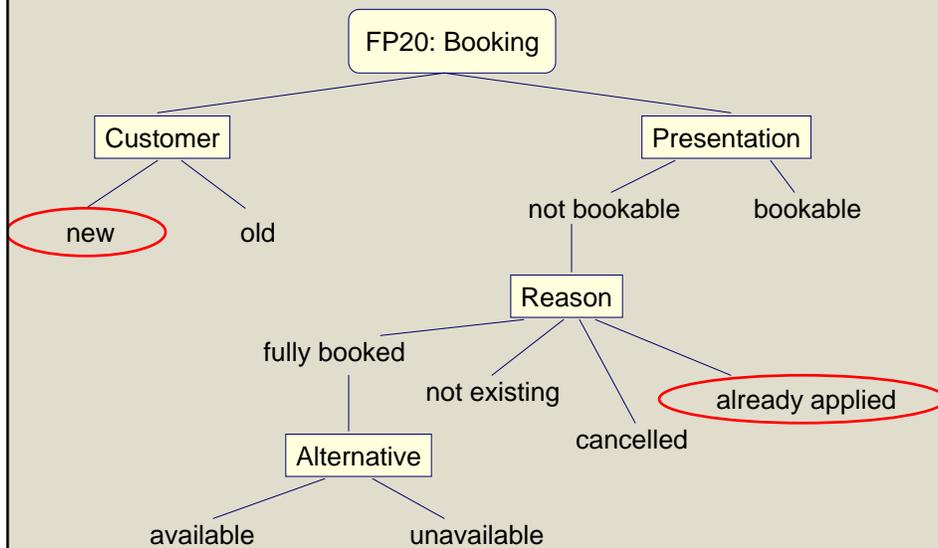
► **Extension:**

- (1A) client data actualization
- (1B) when client is associate of the company, associated company data are updated and accessed
- (1C) invoice verification

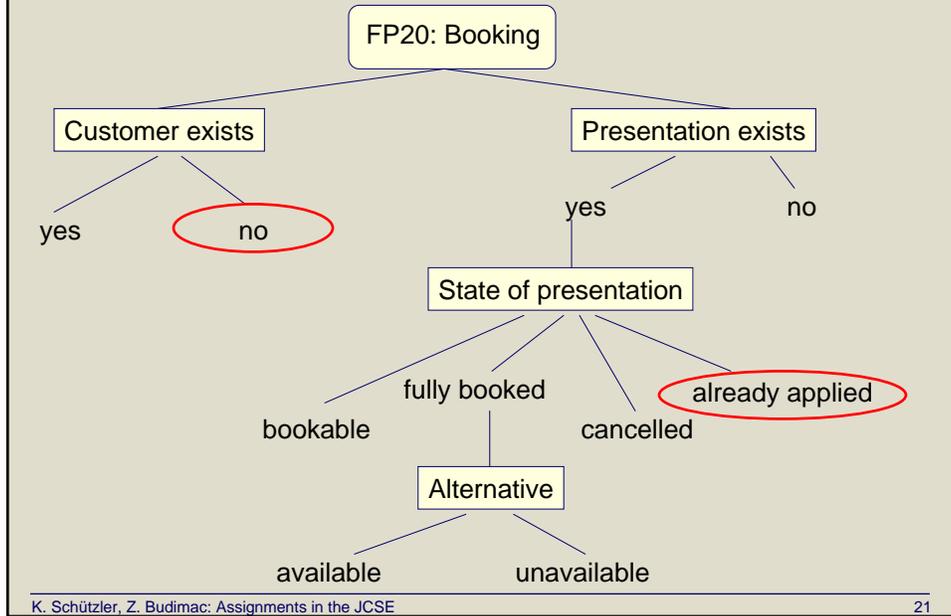
► **Alternatives:**

- (1A) inclusion of a new client
- (2A) when the presentation is fully booked, offer alternative one
- (2B) notification of "false presentation", if the presentation does not exist

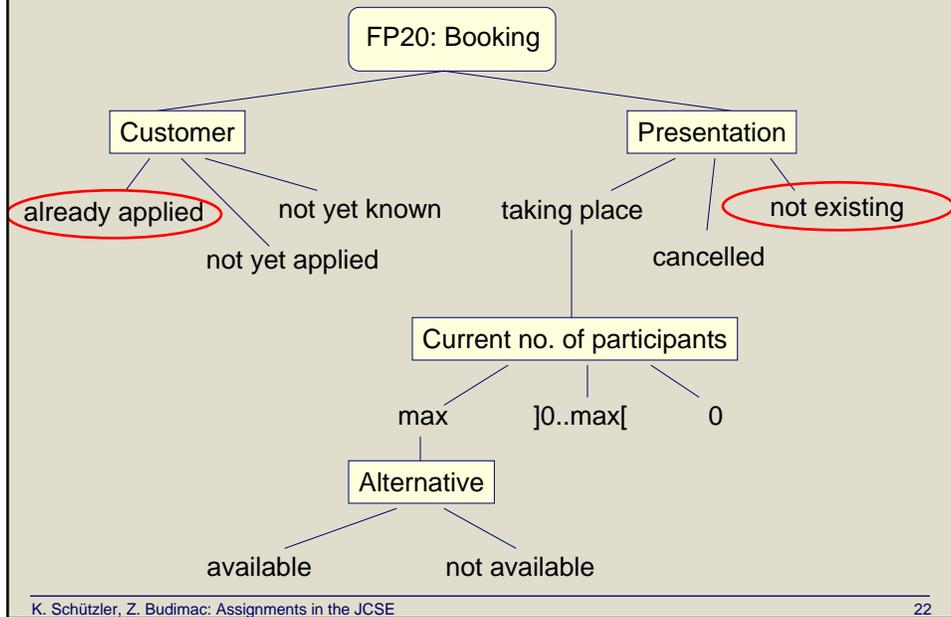
## CTE Assignment: Solution 1



## CTE Assignment: Solution 2



## CTE Assignment: Solution 3



## Assignments in detail: Topic 21: Software Metrics

- ▶ Play around with a software metrics tool
  - Discuss metrics values concerning the measured objects
  - Discover the idea of metrics as indicators
  - Get some easy points (as a reward for bearing assignments so far...)

## Conclusions

- ▶ Students don't find assignments too motivating
  - Especially reviews tend to be stressing
  - Playing around with tools (more at the end of the course) gives more fun
- ▶ They admit that practice isn't always that motivating either
  - In fact practice stresses even more...
- ▶ By now a small but good pool of assignments
  - Assignments fit to the course
  - Assignments give a taste of reality

## Outlook

- ▶ **Bigger pool of assignments wanted**
  - To varyate more from year to year
  - To include other topics like configuration management, project management, ...
- ▶ **All assignments mentioned in this presentation available in English versions**
  - To be found at the JCSE web-site
- ▶ **Discussions of sample solutions and other assignments will follow**
  - Also to be presented at the JCSE web-site