

Do students like tool usage in a Software Engineering course?

Michael Ritzschke, Olga Schiemangk, Klaus Bothe

9th Workshop "Software Engineering Education and Reverse Engineering"
Neum, Bosnia and Herzegovina, 31 August – 5 September 2009

Contents

- Tools in SE course at HU (Overview)
- In detail: assignment – tool support
- Students feedback
- Summary

Which tools did we use in 2009?

Assignments

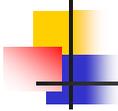
- | | | |
|----|--|-----------------|
| 1. | Review requirements specification "SemOrg" | |
| 2. | Function points | |
| 3. | Develop an OOA model | objectiF |
| 4. | Formal specifications | Z/EVES |
| 5. | Metrics | CCCC |
| 6. | Select test cases functionally by the | CTE |
| 7. | Select regression test cases by | ATOS |
| 8. | Test coverage with | SOTA |

3

Motivation: Why using tools?

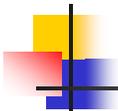
- Students learn to work with different tools, like in practical software development
- Software companies even expect that students are familiar with tools
- Tools: implement theoretical ideas of software engineering
- Solving assignments: we hope tools will introduce fun and liveliness

4



Motivation: Why using tools? (cont.)

- Students get an impression about benefits and shortcomings of tools, e.g.
 - Bugs in tools
 - Stability
 - Online help
 - Usability
 - Platform dependencies



Tools and developers?

- Commercial Software
 - **objectiF**: Tool for Model-Driven Development with UML in Java, .NET and C++ (Company: microTOOL)
 - **CTE XL**: Classification Tree Editor eXtended Logics (DaimlerChrysler); supports functional testing
- Open-source Software
 - **CCCC**: Tool for metrics related to source code (by Tim Littlefair)
 - **Z/EVES**: Supports the Z notation; originally available from ORA Canada (by Mark Saaltink), its status is unknown now.
- Developed at HU
 - **ATOS**: GUI-oriented regression testing, capture-and-replay technique
 - **SOTA**: Test coverage tool

objectiF: www.microtool.de/objectif

CTE XL: www.berner-mattner.com

CCCC: ccc.sourceforge.net



CCCC - C and C++ Code Counter

A free software tool for measurement of source code related metrics by Tim Littlefair

The CCCC tool was developed as a testing ground for a number of ideas related to software metrics in an MSc project. The research project is described at <http://www.chs.ecu.edu.au/~tittlef>

My research project is now (hopefully) coming to an end. The descriptive page will remain on the net, and some material (e.g. the finished thesis) may be added to it, but when my registration as a student expires, soon afterward I would expect to lose the ability to change material on this site, hence the effort to get this one up and running as a forum to cover the onward development of the CCCC tool (for as long as there is any). Many thanks to SourceForge for providing this forum to me at no cost for this purpose. Check out <http://sourceforge.net> if you are interested in their policy of providing free web hosting for open source projects.

In addition to hosting the page you are reading at present, SourceForge support a range of services for their projects. <http://sourceforge.net/projects/ccc> is their standard summary page related to the CCCC project, which provides access to all of these services. In the future I hope to use SourceForge to host anonymous CVS access. For the present, I have set up mailing lists and bug tracking and http download access to the most recent beta release ccc-3pre48.tar.gz.

I have set up a number of mailing lists related to CCCC on the SourceForge site: there is [one for announcements](#), [one for discussions about the use of present versions of CCCC](#), and [another for discussions relating to the future development process](#) (including proposals for new features or changes to old ones).

The interface to search or add to the bug tracking database for CCCC is available via https://sourceforge.net/bugs/?group_id=7763.

The last version of CCCC which I released was 2.1.4, which I put out some time in September 1997. Between then and December 1998, I tried to concentrate on writing up the project thesis. Since December 1998 I have been working on a new release of CCCC, which I plan to designate 3.0. The new version fixes a number of bugs reported in 2.1.4, but also has a lot of new code, and will presumably introduce many new bugs.

9

Z/EVES: oracanada.com



ORA Canada

ORA Canada

[Home](#)

[Contact us](#)

[What's new?](#)

Products and services

Z/EVES

[Documentation](#)

[Screen shots](#)

[EVES](#)

[Ada'95](#)

Reports and Collections

[ORA Canada](#)

[Bibliography](#)

[Automated Deduction](#)

[Bibliography](#)

Z/EVES

As of June 2005, ORA Canada can no longer distribute Z/EVES.

Z/EVES 2.4.1 was the last version released. This version includes a graphical user interface that allows Z specifications to be entered, edited, and analysed in their typeset form; supports the incremental analysis of specifications; and manages the synchronization of the analysis with modifications to the specification. Some [screen shots](#) are available.

Z/EVES uses state-of-the-art formal methods techniques from Europe and North America, integrating a leading specification notation with a leading automated deduction capability. The resulting system supports the analysis of Z specifications in several ways:

- syntax and type checking,
- schema expansion,
- precondition calculation,
- domain checking, and
- general theorem proving.

What's New?

ORA Canada has been inactive since 2005.

Unfortunately, we are unable to distribute EVES or Z/EVES any longer, because we do not own the intellectual property in EVES and no longer have rights to it. However, Mark Saaltink has been working on connecting the Z part of Z/EVES to a new prover. This should be released very soon (in Spring 2009). While the new system is not as powerful, it is a start, and the eventual hope is to make it all open source, so that interested users can improve it.

10

Contents

- Tools in SE course at HU (Overview)
- In detail: assignment – tool support
- Students feedback
- Summary

Neum, Bosnia and Herzegovina , 2009

Assignments (website)

Software Engineering

Prof. K. Bothe

Summer semester 2009

Assignments

Precondition for admission to examination: 75% of reachable points.

Points: you can get maximum 10 points for each assignment.

Mode of delivery: printed on paper.

The annotated solutions to the assignments are distributed for discussion during the class and they are collected after the class as a basis for examination.

Teamwork: Assignment tasks are normally solved in groups of three people. Please talk about deviations to that rule with Dr. Ritzschke before.

Assignments overview

	Theme	Beginning	Delivery	Evaluation	Tool	Demo. in Lec.
Assignment 1	Review Requirements specifications	21.04.09	11.05.09	20.05.09	-	-
Assignment 2	Function point method	04.05.09	18.05.09	27.05.09	-	-
Assignment 3	OOA model	11.05.09	03.06.09	10.06.09	objectF	11.05.09
Assignment 4	Formal software specification	18.05.09	10.06.09	17.06.09	Z/EVES	-
Assignment 5	Classification tree method	25.05.09	15.06.09	01.07.09	CTE	25.05.09
Assignment 6	Test coverage	03.06.09	22.06.09	01.07.09	SOTA	03.06.09
Assignment 7	GUI oriented regression test	10.06.09	29.06.09	15.07.09	ATOSj	10.06.09
Assignment 8	Metrics	15.06.09	06.07.09	15.07.09	cccc	-

Tools (website)

Software Engineering

Prof. K. Bothe
Summer semester 2009

Tools

In the course of the lecture you will work with some software engineering tools. Several of them are demonstrated during the lectures.

OO CASE TOOL: [objectF](#)

TEST TOOLS:

- *CTE XL* - Classification Tree Editor
 - [Users guide](#)
- *SOTA*
 - [Installation hints](#)
 - [User manual](#) (pdf, 1,9MB)
 - [Setup of test environment for SOTA assignment](#) (HUSemOrg)
 - [Setup of test environment for sample program1](#)
 - [Setup of test environment for sample program2](#)
- *ATOM*
 - [Installation hints](#)
 - [User manual](#) (pdf: [one slide per page](#), [two slides per page](#))
 - [HTS language specification\(pdf-file\)](#)
 - [Project setup for HUSemOrg](#)

Z-SPECIFICATION: [Z/EVES](#)

Information about:

- User manuals
- Download information
- Installation guides

General activities of students to use tools



- First contact: Tool demonstration in the lectures
 - examples, hints, remarks, overview of documents
- Preparation: Download, installation, test with sample application
- Become familiar with tool: Get practical know-how by using the tool, work with tutorials and applications, use the help-instructions and comments to find answers to questions
- Solving the assignments: hopefully with more liveliness and fun ...

objectiF

- Students get the description of a

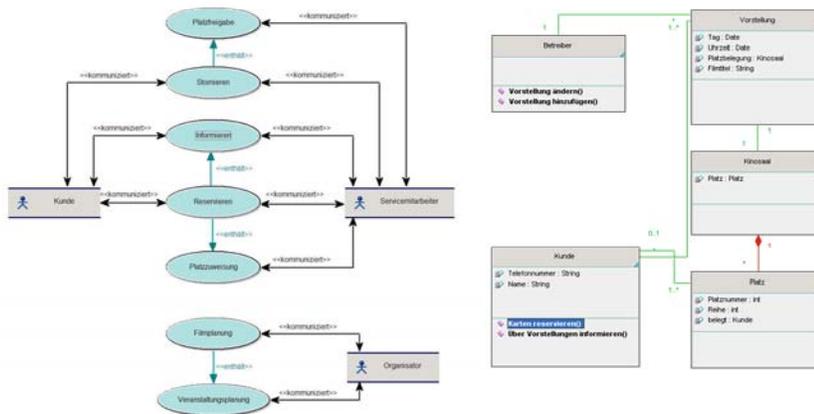
Cinema booking system

- seat reservation, some cinema halls
- booking: row and place number, possibility to ask for neighbouring places
- ...

- Tasks: Use the tool objectiF to develop a
 - use case diagram
 - class diagramm

Tool support: draw graphical elements (use cases, classes)

objectiF: sample solution



Z/EVES

- Students get the description of a queue

create	generates an empty queue
extend	appends a new element at the end
get_first	affords the first element
delete	cuts the first element
not_empty	tests, if the queue is not empty (empty queue affords false)

- Tasks: Describe the queue
 - use the Z-notation
 - algebraic specification

Tool support: Z Mini Editor, Syntax check

Z/EVES

The screenshot displays the Z/EVES tool interface. On the left, the 'Z/EVES Mini Editor' window shows a menu with options: Clear, Schema Box, Action Box, Generics Box, Copy, Cut, and Paste. Below the editor is a toolbar with various symbols for logical, basic, box drawing, subscripts, and tools. At the bottom, there are 'Special Words' including: let, if, then, else, local, global, dom, ran, id, dir, mod, seq, iseq, prefix, suffix, in, disjoint, partition, bag, subbag, pre.

On the right, the 'Syntax Proof' window shows the following Z-notation specification for a queue:

```

Queue
queue_elements : seq Z
  
```

Below the specification, several algebraic specifications are listed:

```

new
ΔQueue
queue_elements' = {}

is_empty
∃Queue
msg! : {true, false}
msg! = true ⇔ queue_elements = {}

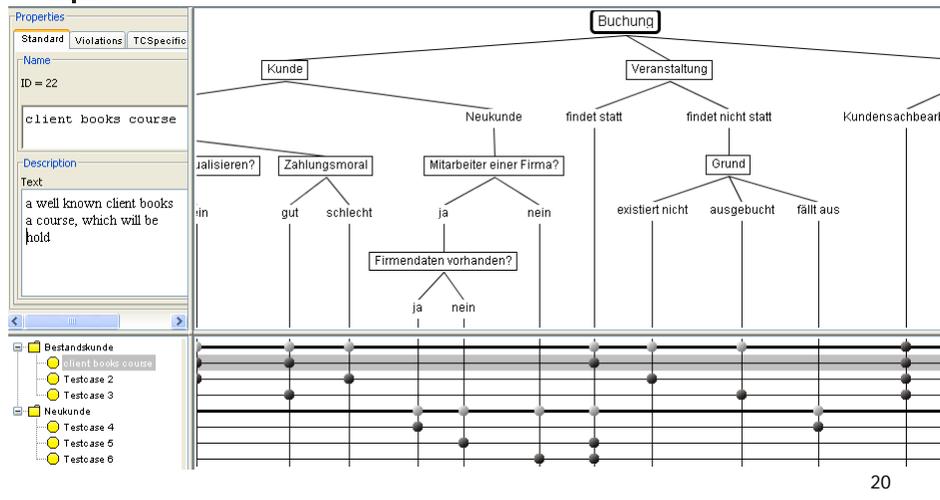
enqueue
ΔQueue
inp? : Z
queue_elements' = queue_elements ^ {inp?}
  
```

CTE XL

- Students know the functional specification document "Seminar organization" and a selected use case /F20/ *From the registration of a client to the reservation of a course (booking process)*
- Tasks
 - a) use the classification tree method to prepare a **functional test**
 - b) find out which persons/objects/situations are relevant for the booking process and build classes
 - c) generate test cases (combine classes)

Tool support: Drawing classification tree, generating and marking test cases, creation of documents ...

CTE XL



SOTA (Detail Information: Presentation by M.Hildebrand)

- Realize a **structure-oriented test** for a part of the semorg.gui-package (test object: *ClientWindow.java*)
- Tasks
 - a) use SOTA, Eclipse, HUSemorg and MySQL to realize different test runs for the Window "client" (use Buttons, text inputs, arrays ...)
 - b) use SOTA to check the result (e.g. Instruction coverage) and start further test runs till the coverage reaches a given border (e.g. 90%)

Tool support: Instrumentation of java-files

SOTA

The screenshot shows the Eclipse IDE with the following components:

- Left Panel (Package Explorer):** Shows the project structure with folders like 'Algorithmen', 'blat7', 'GGT', 'grafik_1', 'Heap', 'husemorg', and 'src'. The 'src' folder contains the 'semorg' package.
- Center Panel (Code Editor):** Displays the source code for `ClientWindow.java`. The code includes imports for `semorg.sql.table` and `semorg.sql.util`, and defines a `public class ClientWindow` with an inner class `InputChangeListener`. Comments indicate the window is for editing lecturer data.
- Right Panel (GUI Preview):** Shows a window titled "Neu - Kunde*" with a form for entering customer data. Fields include "Anrede" (set to "Herr"), "Titel" (set to "Prof"), "Vorname", "Nachname" (set to "Bothe"), "Strasse", "PLZ/Ort", and "Land" (set to "Zusatz").

Tool support: Evaluation the test runs (different metrics, graphical view of instruction coverage, ...)



SOTA

Name	FEEC	CO	C1	C2	MMDC	MCDC	C3	MBI
ClientBookingWindow.java	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
ClientListWindow.java	20,00%	16,05%	11,11%	8,33%	6,25%	0,00%	8,33%	13,89%
ClientTableProvider.java	87,50%	91,94%	91,18%	80,00%	80,00%	60,00%	80,00%	88,24%
ClientWindow.java	37,66%	78,08%	36,89%	36,79%	41,91%	10,34%	37,50%	0,00%
ClientWindow	62,96%	84,50%	51,25%	55,88%	57,69%	17,65%	55,22%	0,00%
ClientWindow (Shell)	100,00%	100,00%	---	---	---	---	---	100,00%
ClientWindow (Shell, Cljer)	100,00%	100,00%	---	---	---	---	---	100,00%

```

Shell instance used the as parent shell of the window.
*/
private ClientWindow(Shell parent) {
createE
shell = new Shell(parent, SWT.CLOSE);
checkIr
shell.addListener(SWT.Close, new Listener() {
public void handleEvent(Event event) {
confirm
if (inputChanged) {
onSave
if (confirmClose()) {
InputC
if (inputEditingEnabled && input != null)
Client.lockedIds.remove(new Integer(input.getId()));
} else
event.doit = false;
} else if (input != null && inputEditingEnabled)
Client.lockedIds.remove(new Integer(input.getId()));
}});
}
    
```

ATOSj

- Realize a part of **regression test** for the use case /F20/ (*booking process*)
- Tasks
 - a) use the capture function from ATOSj to register your inputs in different windows (e.g. seminary typ, docent, client, public seminar ...)
 - b) start the test sequences (replay) and prepare them manually so, that they runs without any mistakes



Tool support: Capture the inputs



ATOSj - HUSemOrg

Projekt Bearbeiten Hilfe

HUSemOrg

Testpakete1 *SeminarTyp Dozent Kunde Firma ÖffentlicheVerans...

Nummer	Kommando
1	START
2	ACTION, MAIN, MENU, "MainMenu", SELECT, SUBITEM, "Ersterfassung", "SeminarTyp"
3	ACTION, "SeminarTypWindow", EDITBOX, "ShortTitelText", EDIT, "typ123"
4	ACTION, "SeminarTypWindow", EDITBOX, "TitelText", EDIT, "TESTEN MIT ATOSj"
5	ACTION, "SeminarTypWindow", EDITBOX, "TopicText", EDIT, "Testen mit ATOSj"
6	ACTION, "SeminarTypWindow", EDITBOX, "DurationText", EDIT, "6"
7	ACTION, "SeminarTypWindow", EDITBOX, "AudienceText", EDIT, "Informatikstudenten"

Seminarorganisation

Anwendung Stammdatenlisten Ersterfassung Geschäftsprozesse Administration Hilfe

Neu - Seminartyp*

Nummer:

Kurztitel:

Titel:

Dauer:

Unterlagen:

Zielsetzung:

Tool support: Replay test sequences to test later releases (check, if the functionality has changed)



ATOSj - 18.06.2016

Projekt Bearbeiten Hilfe

HUSemOrg

Testpakete1 SeminarTyp Dozent Kunde Firma ÖffentlicheVerans... Cleanup

Testpakete1 "Testpakete1" ausführen

Testeinstellungen

Testsequenz bei einem Fehler beenden Testpaket bei einer Fehler

Protokolldatei anlegen Interaktive Testsequenzen

100 Verzögerung in ms Testsequenzen ohne Clean

Nummer	Testsequenz	Interaktiv	Cleanup	Status	Durchläufe
2	Programmstart			Erfolgreich	1
3	SeminarTyp			Erfolgreich	1
4	Dozent			Erfolgreich	1
5	Öffentlicheveranstaltung			Erfolgreich	1
6	Kunde			Erfolgreich	1

Seminarorganisation

Anwendung Stammdatenlisten Ersterfassung Geschäftsprozesse Administration Hilfe

Neu - Dozent*

Name:

Adresse:

Telefon:

Handy:

Fax:

E-Mail:

Nummer	Kommando	Status	Durchläufe	Dauer
1	ACTION, MAIN, MENU, "MainMenu", SELECT, SUB...	Erfolgreich	1	506
2	ACTION, "SeminarTypWindow", EDITBOX, "ShortT...	Erfolgreich	1	10016
3	ACTION, "SeminarTypWindow", EDITBOX, "TitelT...	Erfolgreich	1	10016



CCCC-Tool

- CCCC.exe calculates different software metrics (for Java and C)
- Tasks
 - a) use the tool to investigate the MVG-metric (McCabe's Cyclomatic Complexity) for two given java-Files
 - b) for high MVG-values: look to the program-structure and discuss. Have the programs also a high complexity? Is the maintenance difficult?

Tool support: Investigates different metrics for whole modules and all functions



CCCC-Tool

Detailed report on module Error

Metric	Tag	Overall	Per Function
Lines of Code	LOC	43	*****
McCabe's Cyclomatic Number	MVG	57	*****
Lines of Comment	COM	6	*****
LOC/COM	L_C	7.167	
MVG/COM	M_C	9.500	

Functions

Function prototype	LOC	MVG	COM	L_C	M_C
error(int, int, int) definition Error.java:8	34	57	3	11.333	19.000

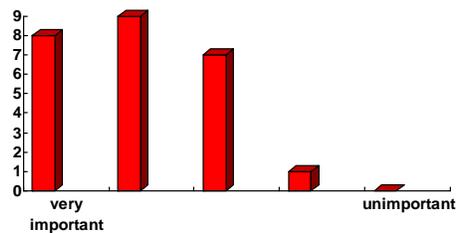
Contents

- Tools in SE course at HU (Overview)
- In detail: assignment – tool support
- Students feedback
- Summary

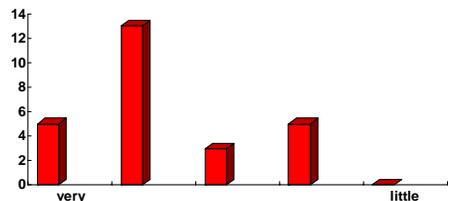
Neum, Bosnia and Herzegovina , 2009

Generally questions

1. Relevance:
Is the use of software tools in the course important for your professional career?

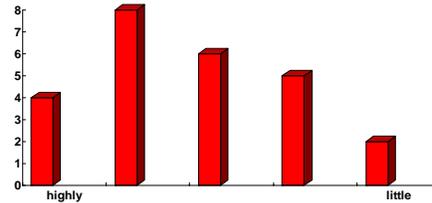


2. Competence: Did you get more professional competence about software tools by our assignments?

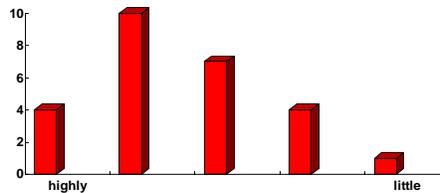


Generally questions

3. Technical profit:
Did the use of the tools
have relevance for
solving the assignments
(effect on efficiency)?



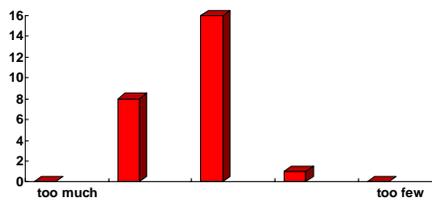
4. Functional profit:
Had the use of tools
relevance for better
understanding the
theoretical concepts
from the lecture?



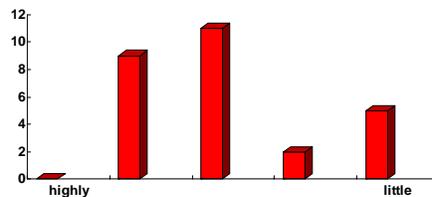
31

Generally questions

**5. Was the use of
software tools time-
consuming?**



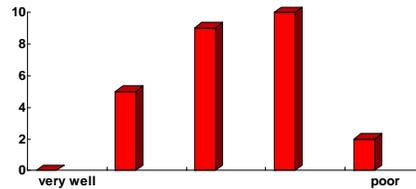
**6. Fun: Did you use the
tools with fun?**



32

Generally questions

7. Operability: How did you experience the tool operability in general?



33

Individual questions

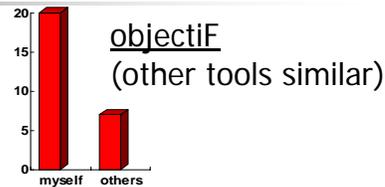
Similar questions for each individual tool:

- Technical profit: working with CCCC, CTE, SOTA, ATOS, Z/EVIS, objectiF
- Fun: working with CCCC, CTE, SOTA, ATOS, Z/EVIS, objectiF
-

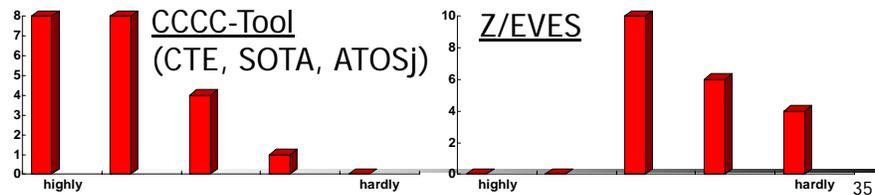
34

Individual Questions

0. Who did work with the tool?



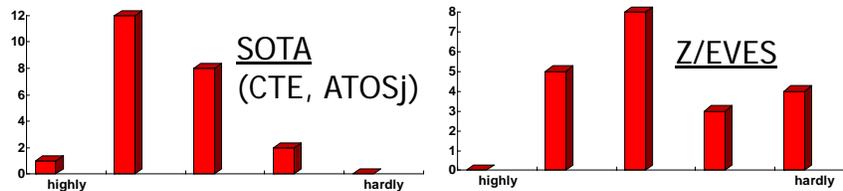
1. Technical profit:
Did the use of the tool have relevance for solving the assignments (effect on efficiency)?



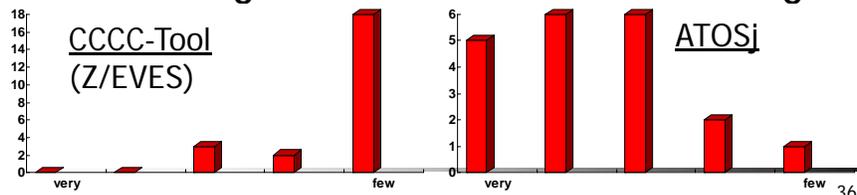
35

Individual Questions

2. Relevance: Is the using that tool important for your professional career?



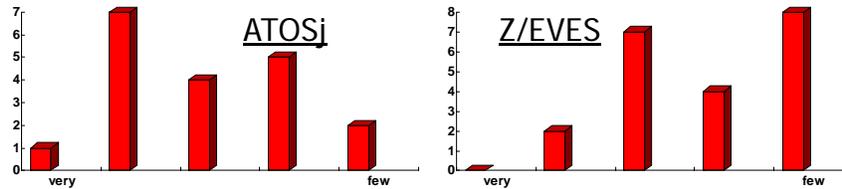
3. Was the using of software tools time-consuming?



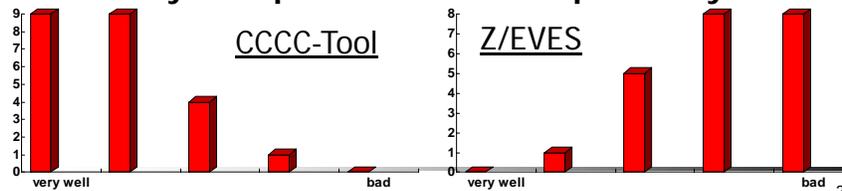
36

Individual Questions

4. Fun: Did you use the tool with fun?



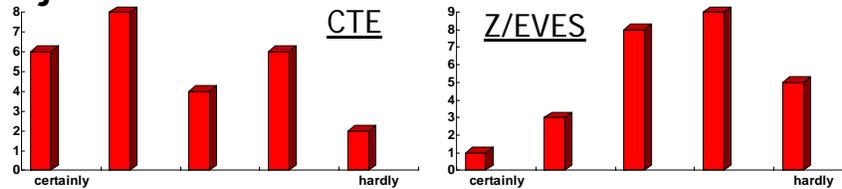
5. How did you experience the tool operability?



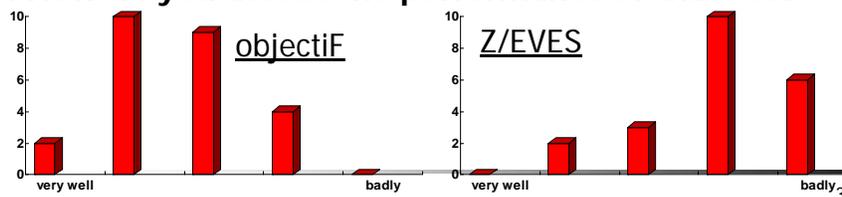
37

Individual Questions

6. Sustainability: Would you use that tool to realize projects in the future?



7. How do you assess the performance of that tool?



38



Students comments

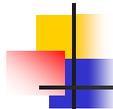
- Other tools also useful: Junit, SVN (subversion)
- Tools mentioned in job offers should be selected
- Widely used tools should be taken instead of in-house products (HU: ATOS, SOTA)
- Some tools were applicable only under Windows (students partly works with Linux)
- More time necessary for the installation than for use in the assignment
- Assignments with tool usage were well-prepared

39



Contents

- Tools in SE course at HU (Overview)
- In detail: assignment – tool support
- Students feedback
- Summary



Summary

- Tools are necessary for lively lessons in SE
 - Working in teams is valuable for preparation of professional work
 - Software engineering requires the using of models and tools
 - Efforts for the staff:
 - installation of tools in computer lab
 - information on tools at website
about tool installation, usage and downloads
 - assignments: solved by staff before
-

41



Summary

Thank you for your attention!

42