Das Virtuelle Unix Labor

– Statusbericht SS2005 –

Hubert Feyrer <hubert@feyrer.de>

Inhalt

- Motivation
- Informationswissenschaftliche Aspekte
- Ergebnisverifikation mit Domänenspezifischen Sprachen
- Evaluation
- Status & Ausblick

Motivation

- Vorlesung “Systemadministration” am Fachbereich Informatik/Mathematik der FH Regensburg
- Wahl-Vorlesung besteht seit 1992, seit 2002 Pflichtvorlesung im Studiengang “Allgemeine Informatik”
- Mangel an Übungsmöglichkeiten für fortgeschrittene Themen
- Idee eines dedizierten Übungssystems ca. 2000
- Von 2001 bis 2004 im Rahmen des Hochschul-Wissenschaftsprogramms I (HWP) des BMBF entwickelt

Motivation

- Ergebnis: Webbasierte Lernumgebung zum Buchen von Kursen, vorbereiten dedizierter Übungsrechner, zum Absolvieren von Übungen und Erstellen von Auswertungen
- Erweiterung der Grundfunktionalität durch Themenbereiche der Informationswissenschaft

Wissenschaftliche Aspekte

- Grundlagenarbeit: Virtuelles Labor für Systemadministration
- Didaktik & Einbettung in Präsenz- und virtuelle Veranstaltung
- Verifikation der Übungsergebnisse mittels Domänenspezifischer Sprachen
- Unterstützung der Lerner durch tutorielle Komponente
- Anpassung des Systems an unterschiedliche Lernertypen bzgl. Hilfestellung und Auswertung

Details zur Ergebnisverifikation mit Domänenspezifischen Sprachen

Ergebnisverifikation

- Analyse der Übungssysteme am Ende der Übung
- Detailiertes Feedback an den Benutzer bzgl. Übungserfolg
- Zwei Komponenten: Aufgabentext, Tests
- Aufgabentext: für den Studenten, Klartext (HTML)
- Tests: für das System, greifen auf bestehende Test-Primitive zurück
- Test-Primitive: Anfangs sehr spezifisch für jeweiligen Test, später mit Hilfe von Parametern generalisiert

Beispiele: check-directory-exists check-fi le-contents check-fi le-exists check-program-output netbsd-check-installed-pkg netbsd-check-rcvar-set netbsd-check-user-shell solaris-check-installed-pkg unix-check-fi le-owner ...

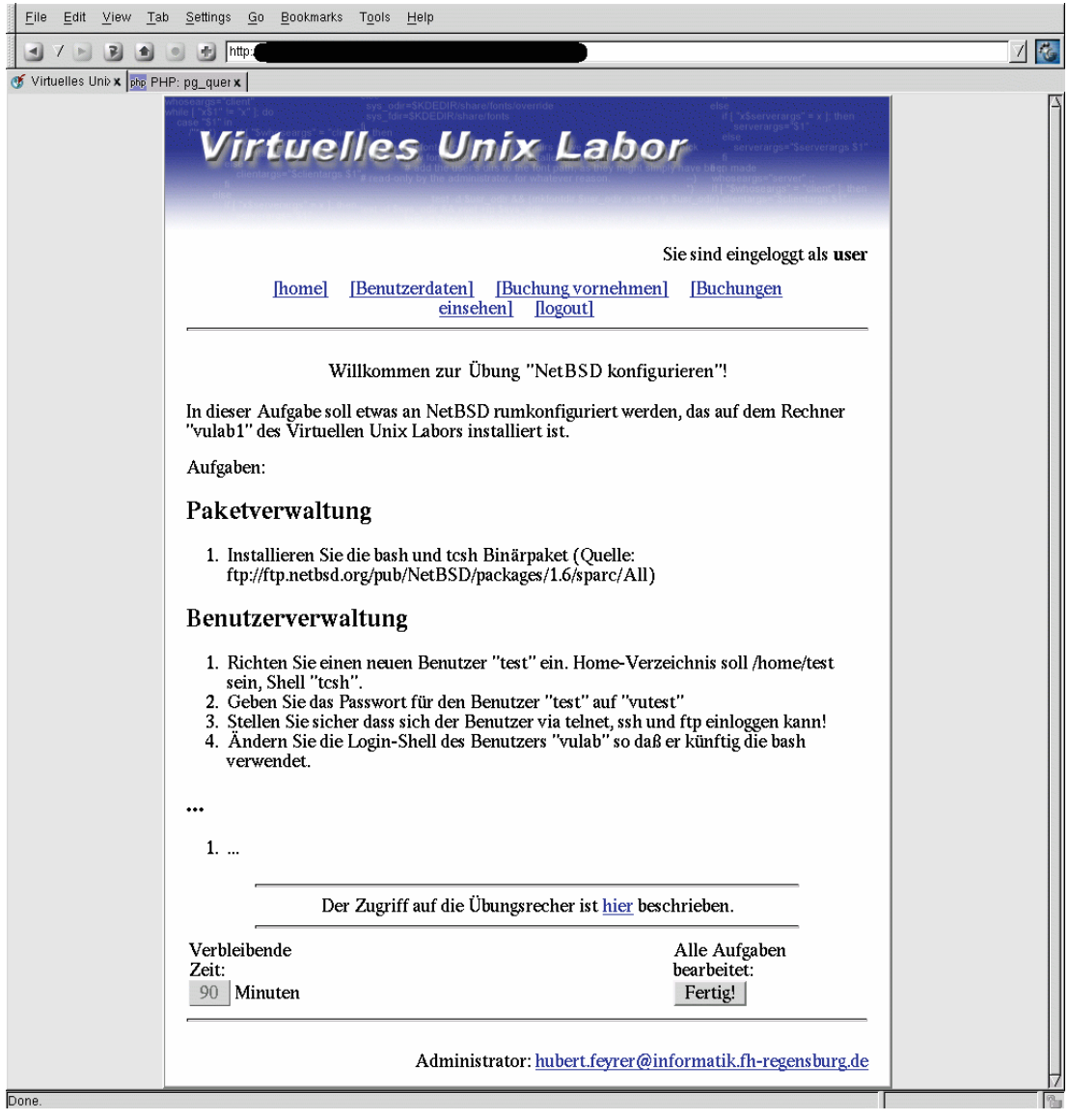
Ergebnisverifikation Schritt I

- Definieren der Tests via Web-Frontend durch den Administrator
- Am Übungsende: Abarbeiten der Tests, Ergebnis für jeweilige Übung in Datenbank
- Auswertung: Liste von Tests & Testergebnissen
- Problem: keine klare Zuordnung zwischen Aufgabentext und Tests (und -ergebnissen)

Ergebnisverifikation Schritt II

- Kopplung zwischen Aufgabentext, Tests und Feedback
- Definition von Tests weiterhin via Web-Frontend durch Administrator
- Hinterlegen von Hinweisen für Auswertung im Aufgabentext: An welcher Stelle soll das Ergebnis welcher Tests gezeigt werden?
- Ausgabe der Auswertung nur bei Abfrage von Feedback, nicht bei Einsicht in Aufgabe vor bzw. während Übung (via PHP-Funktionen)

Ergebnisverifikation: Übung



Ergebnisverifikation: Auswertung

Aufgaben:

Paketverwaltung

1. Installieren Sie die bash und tcsh Binärpaket (Quelle: ftp://ftp.netbsd.org/pub/NetBSD/packages/1.6/sparc/All)

Pakete installiert? (pkg_info -e)	Nein

Benutzerverwaltung

1. Richten Sie einen neuen Benutzer "test" ein. Home-Verzeichnis soll /home/test sein, Shell "tcsh".

"test" finger(1)bar?	OK
Korrektes Home-Verzeichnis? (finger, test -d)	OK
Shell richtig gesetzt? (finger)	OK
Eintrag in /etc/master.passwd?	OK

2. Geben Sie das Passwort für den Benutzer "test" auf "vutest"

Passwort richtig gesetzt? (getpwnam(3), crypt(3))	OK

3. Stellen Sie sicher dass sich der Benutzer via telnet, ssh und ftp einloggen kann!
4. Ändern Sie die Login-Shell des Benutzers "vulab" so daß er künftig die bash verwendet.

Login-Shell vulab? (chfn/chsh, finger)	Nein

Ergebnisverifikation: Auswertung 2

Paketverwaltung

1. Installieren Sie die bash und tcsh Binärpaket (Quelle: ftp://ftp.netbsd.org/pub/NetBSD/packages/1.6/sparc/All)

tcsh installiert? (pkg_info -e tcsh) OK

Bestanden:	3	(37%) |ooo
Nicht bestanden:	5	(62%) |ooooo
Summe:	8	(100%)

bash installiert? (pkg_info -e bash) Nein

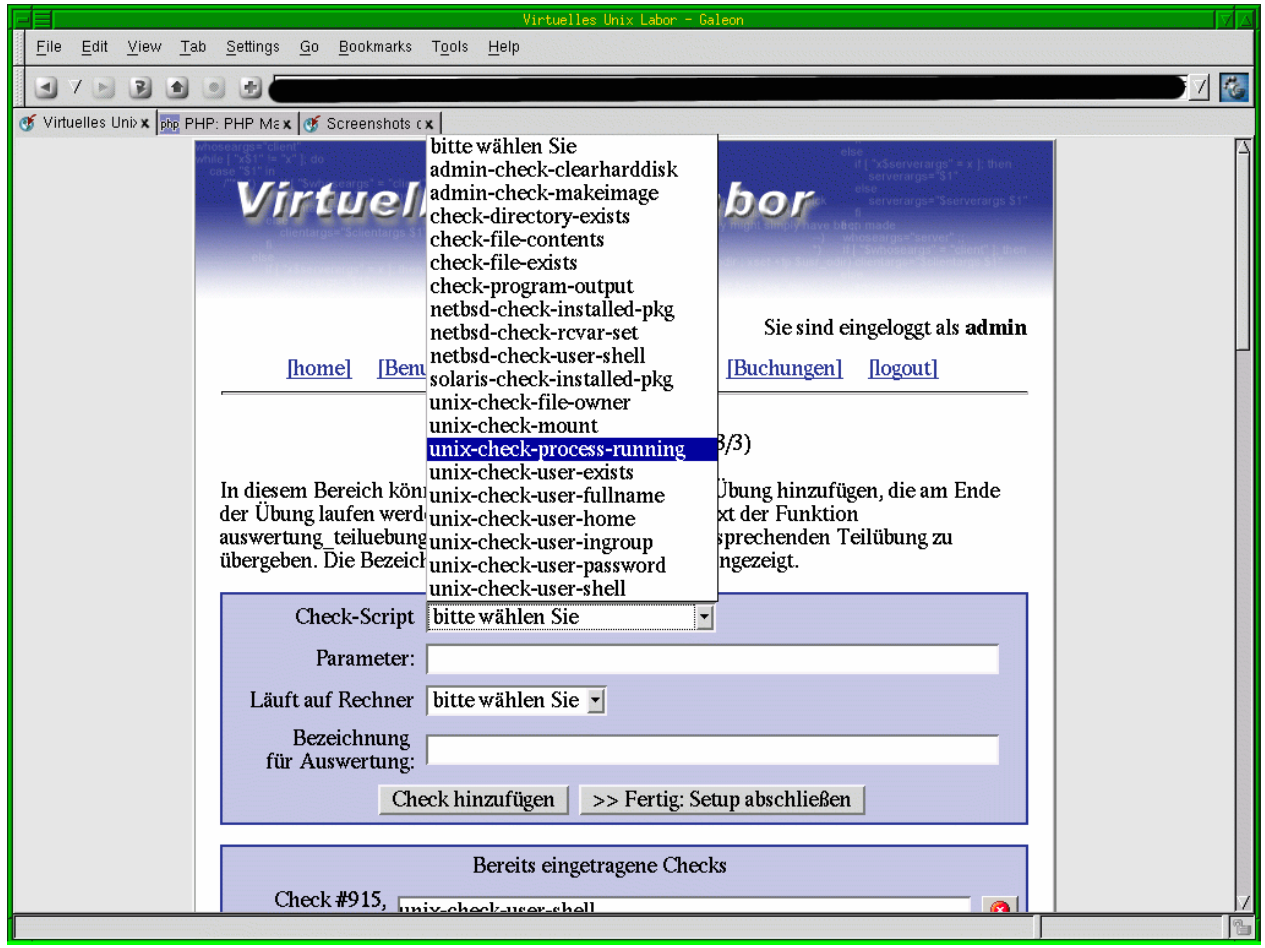
Bestanden:	1	(12%) |o
Nicht bestanden:	7	(87%) |ooooooo
Summe:	8	(100%)

Ergebnisverifikation: Quellcode

```
...
<h2> Benutzerverwaltung </h2>
<ol>
<li> Richten Sie einen neuen Benutzer "test" ein. Home-Verz
     soll /home/test sein, Shell "tcsh".

     <?php auswertung_teiluebungen( 910, 911, 912, 913 ); ?

<li> Setzen Sie das Passwort für den Benutzer "test" auf "v

     <?php auswertung_teiluebungen( 914 ); ?>

...
```

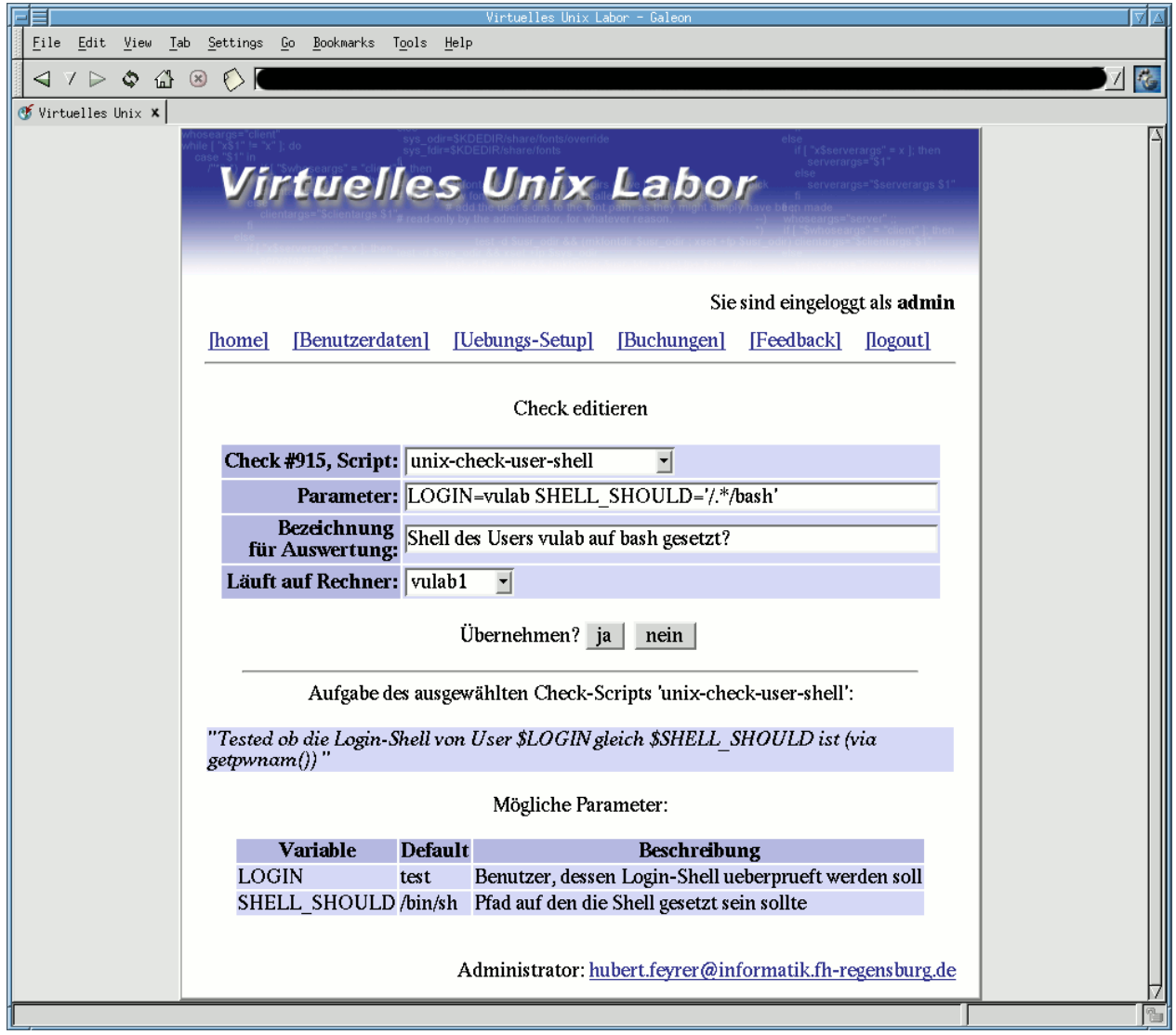
Ergebnisverifikation: Fortsetzung

- Bestehender Satz von Test-Primitiven wird via Web-Frontend parametrisiert, Test-Nummern zugewiesen und Text für Auswertung hinterlegt:



Ergebnisverifikation: Fortsetzung

Editieren der Parameter:



Ergebnisverifikation: DSL 1

- Pro Übung ca. 40 Tests
- Eingabe via Web-Frontend ist umständlich, Fehleranfällig, und Aufgabe und Tests können nicht an einer Stelle definiert werden
- Lösung: hinterlegen der Check-Daten im Aufgabentext (Quellcode), und auslesen via Präprozessor
- Übungstext erhält Aufruf von Primitiven (Aktivatoren) in Sequenz $\Rightarrow$ Domänenspezifische Sprache (Domain Specific Language, DSL) zur Ergebnisüberprüfung
- Sprachdefinition bei DSLs nicht über formale Grammatik, lexikalische und syntaktische Analyse und Codegenerierung, sondern angelehnt an existierende Sprache. Hier: PHP

Ergebnisverifikation: DSL 2

Test-Daten in Aufgabentext:

feyrer@wl445:/home/feyrer/work/vulab/code/public_html/texte

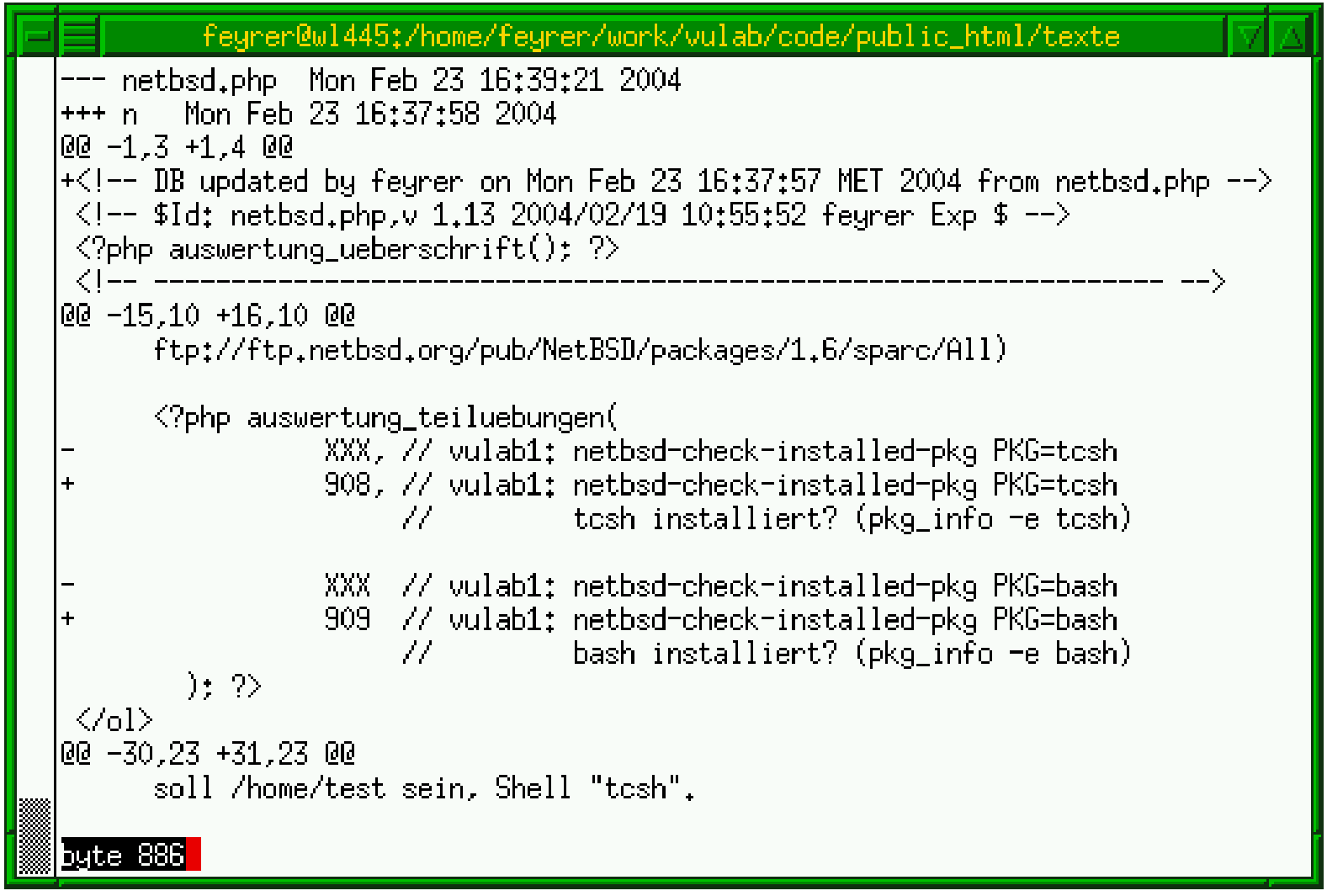
```
<p>

Aufgaben:
<p>

<h2> Paketverwaltung </h2>
<ol>
<li> Installieren Sie die bash und tcsh Bin\xe4rpaket (Quelle:
     ftp://ftp.netbsd.org/pub/NetBSD/packages/1.6/sparc/All)

     <?php auswertung_teiluebungen(
                XXX, // vulab1: netbsd-check-installed-pkg PKG=tcsh
                     //         tcsh installiert? (pkg_info -e tcsh)

                XXX  // vulab1: netbsd-check-installed-pkg PKG=bash
                     //         bash installiert? (pkg_info -e bash)
        ); ?>
</ol>


<h2> Benutzerverwaltung </h2>
<ol>
<li> Richten Sie einen neuen Benutzer "test" ein, Home-Verzeichnis
```

Ergebnisverifikation: DSL 3

Test-Daten in Datenbank übernehmen:

Ergebnisverifikation: DSL 4

Aktualisierten Übungstext überprüfen:



Ergebnisverifikation: Zusammenfassung

- Erstellen einer domänenspezifischen Sprache
- Iteratives Vorgehen: Definition von Primitiven, Verallgemeinerung durch Parameter, Erstellen eines Prozessors
- Test-Relevante Daten können beim Aufgabentext gehalten werden (“Data Structure Representation” Pattern)
- Dadurch leichtere Wartung der Übungen (“all in one place”)
- Da Abfragesprache auf PHP basiert können komplexe Abfragen definiert werden (“Language Specialisation” Pattern)

Ergebnisverifikation: Zusammenf. 2

- Selektion und ggf. Iteration für komplexe Aufgabenstellungen machbar
- Weiterhin: “System Frontend” zur Aktualisierung der Übungsrechner-Konfiguration durch spezielle Test-Primitive
- Effektive Überprüfung basierend auf Zustand der Übungssysteme entspricht Administrator-Verhalten
- Literatur:
  - John Cocke, J. T. Schwartz: Programming Languages and their Compilers, 1970.
  - Diomidis Spinellis: Notable design patterns for domain-specific languages, 2001.

Details zur Evaluation

Evaluierte Bereiche

- Prüfungen (Klausuren)
- Fragebogen
- Methodik
- Ergebnisse
- Sonstiges

Evaluation: Prüfungen

Semester and year	Date of test	Number of tests	Lecture tested	Lecture given	VUlab used?
SS 2002	23.07.2002,	37*	SA	SA	no
SS 2002	16.07.2002,	1*	SY	-	no
WS 2002/03	06.02.2003,	7*	SA	-	no
WS 2002/03	07.02.2003,	30*	SY	SY	no
SS 2003	24.07.2003,	1*	SY	-	no
WS 2003/04	03.02.2004,	1*	SY	-	no
WS 2003/04	06.02.2004,	3*	SA	-	no
SS 2004	14.07.2004,	33*	SA	SA	yes
WS 2004/05	27.01.2005,	4*	SA	-	no

Evaluation: Prüfungen

Ausgewertete Gebiete:

- Gesamtnote
- Gesamtpunktezahl
- Ergebnisse ohne VUlab
- Ergebnisse mit VUlab

Evaluation: Fragebogen

Bereiche:

- Zielpublikum
- Übungsverlauf
- Verwendung von Lernmitteln
- Benutzerakzeptanz

Evaluation: Methodik 1

Verfahren der beschreibenden Statistik:

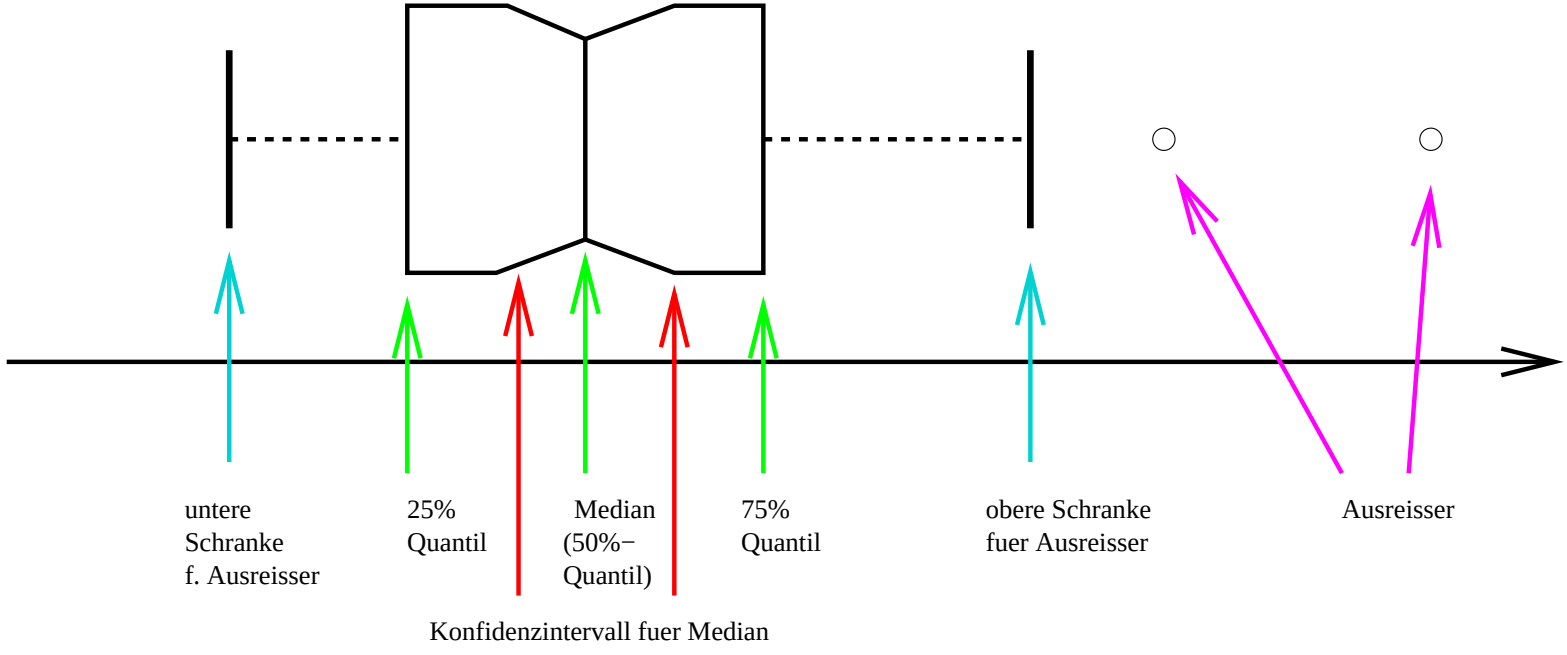
1. Histogramme
2. Mittelwert: nur bei intervallskalierten Daten
3. Median: bei intervall- und ordinalskalierten Daten (50%-Quantil)
4. Modus: bei intervall-, ordinal- und nominalskalierten Daten

Annahme: Noten und Punkte sind intervallskaliert!

Evaluation: Methodik 2

Visuelles Verfahren:

1. Darstellung von Median, 25%- und 75%-Quantilen, Grenzen zur Kennzeichnung von “Ausreißern” sowie Konfidenzintervalle für den Median zur visuellen Evaluation mittels Box/Whisker-Plots:



Evaluation: Methodik 3

Verfahren der Prüfstatistik:

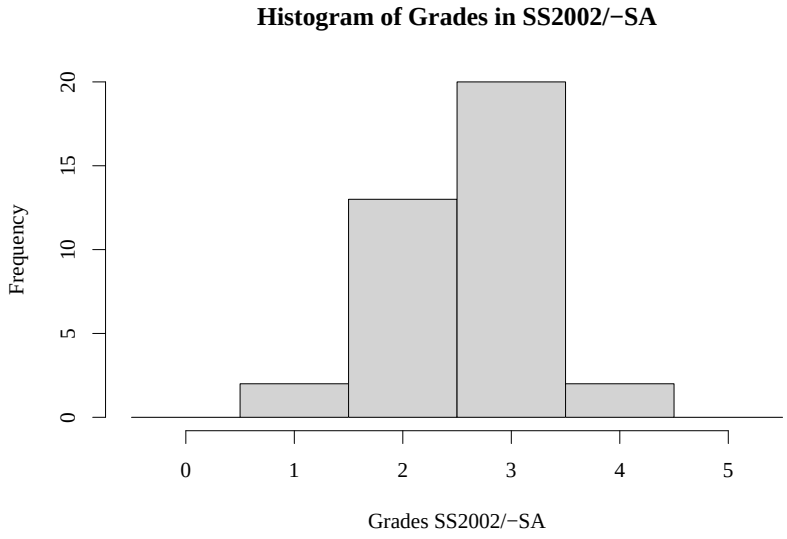
1. Vergleich von Verteilungen mittels Man-Whitney/Wilcoxon-Test (p_w)

   Berechnung in R:

```
> R_sa_ss2002_noten
 [1] 4 2 2 3 3 3 2 2 3 1 2 3 3 3 2 3 2 2 2 3 3 3 2 1 3 4 2 2 3 3 3 3 3 3
> R_sa_ss2004_noten
 [1] 2 3 2 2 3 3 4 2 4 4 3 2 3 2 2 2 2 3 2 4 3 4 4 3 2 2 3 3 3 3 2 4 3
> wilcox.test(R_sa_ss2002_noten, R_sa_ss2004_noten)

        Wilcoxon rank sum test with continuity correction

data:  R_sa_ss2002_noten and R_sa_ss2004_noten
W = 533.5, p-value = 0.3267
alternative hypothesis: true mu is not equal to 0
```

Evaluation: Methodik 4

Zugehörige Histogramme:



Evaluation: Methodik 5

2. Vergleich der Varianz von Verteilungen mittels F-Test (p_F)
3. Vergleich des Mittelwerts von Verteilungen mittels Student's t-Test (p_t)
4. Test auf Gleichheit des Mittelwerts von Verteilungen, aus [Störmer, 1971]: Berechnung von Hilfswert H, und Vergleich gegen Wert u, abhängig von genauem Test und gewünschter Signifikanz:

Alternative hypothesis	u	$S = 95\%$	$S = 99\%$
is not equal:	u_{ne}	1.959964	2.575829
is greater:	u_{gt}	1.644854	2.326348

Wichtig: Vergleiche mit ungleicher Anzahl von Samples!

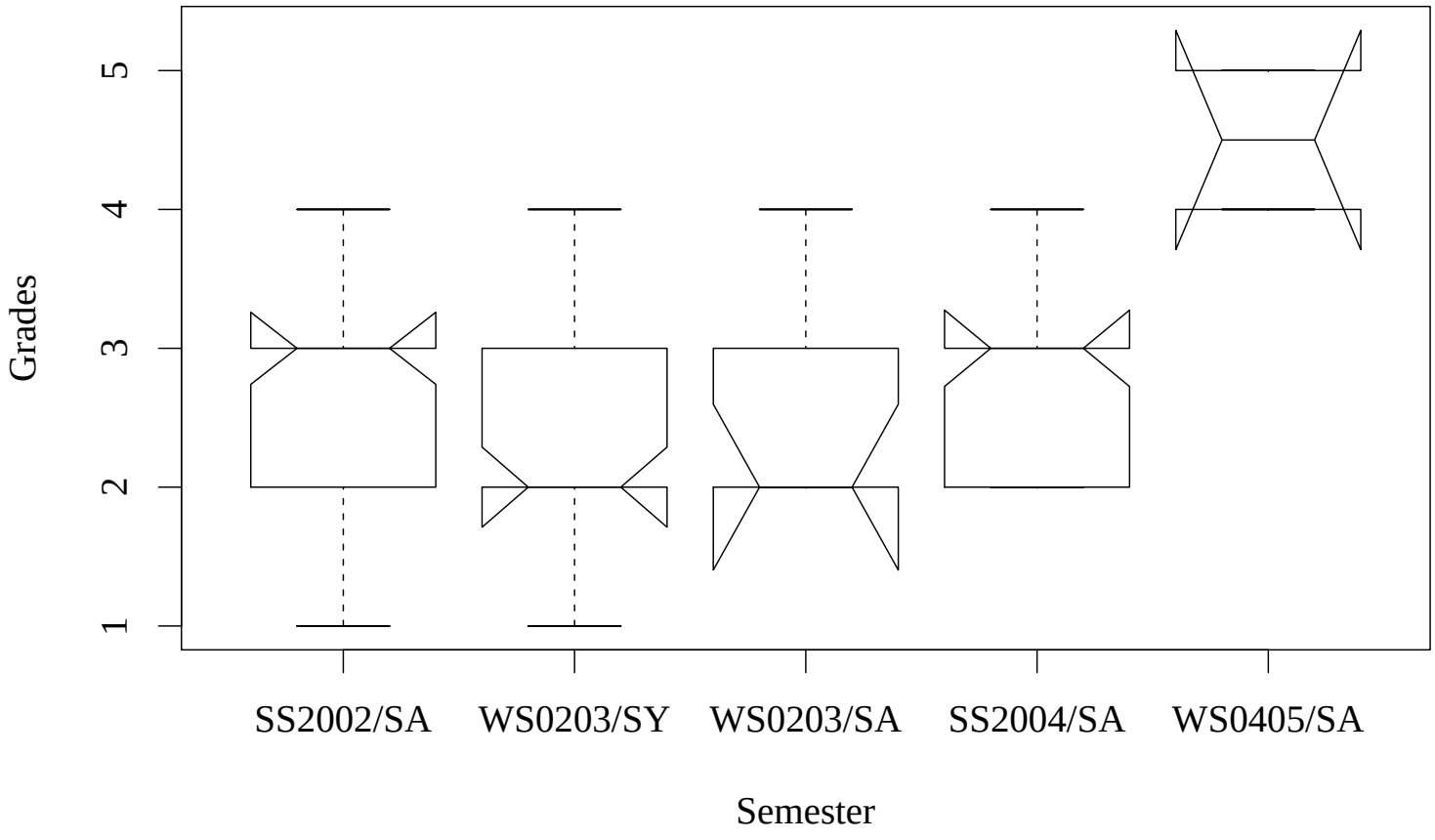
Evaluation: Methodik 6

Vergleich verschiedener Werte gegeneinander mittels Matrizen:

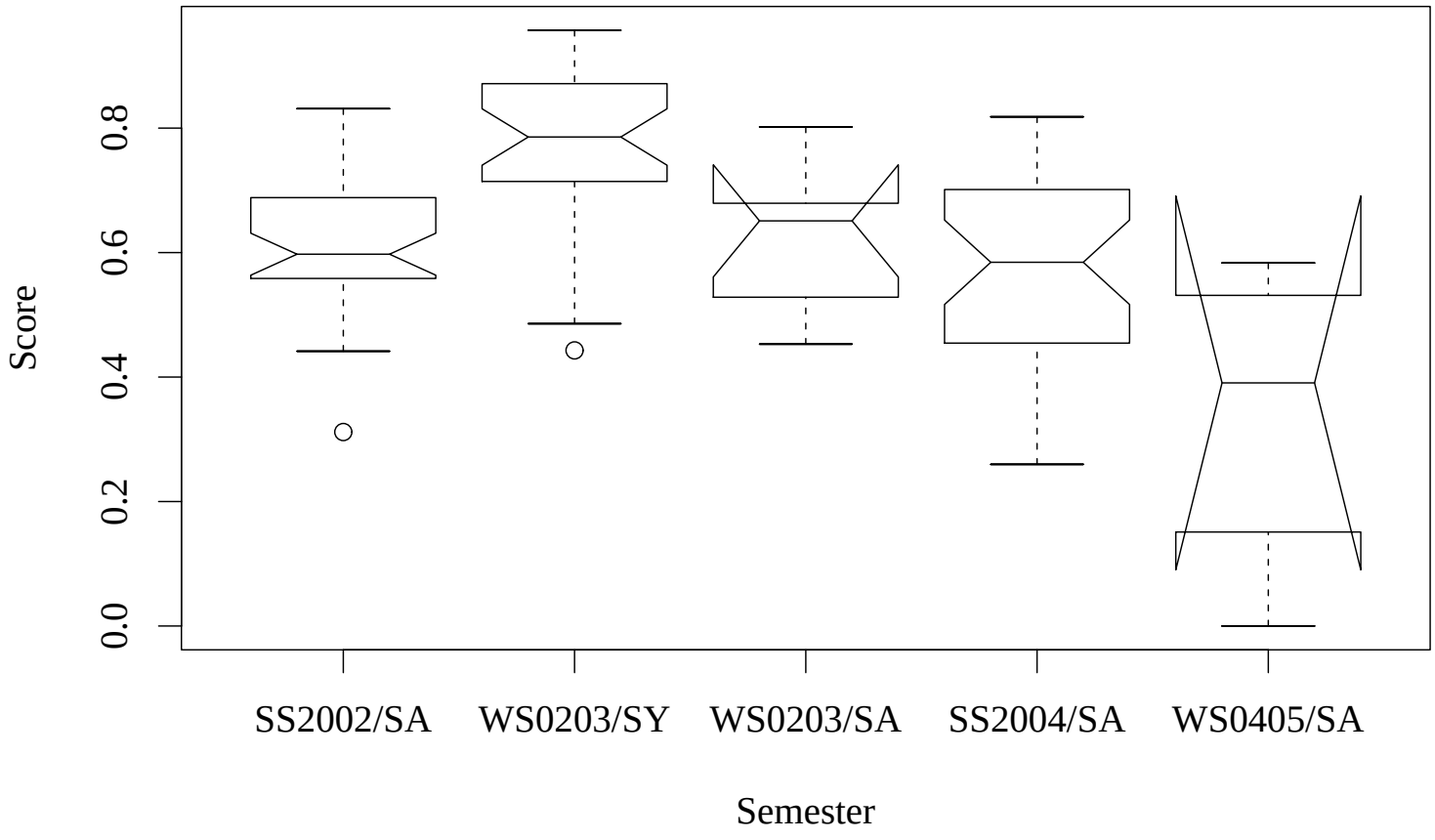
$\downarrow p_F/p_w \rightarrow$	SS2002/SA	WS0203/SY	WS0203/SA	WS0405/SA
SS2002/SA	X	0.8382	0.6215	0.4982
WS0203/SY	0.1000	X	0.4343	0.4911
WS0203/SA	0.5157	0.8165	X	0.7706
WS0405/SA	0.6227	0.2073	0.3977	X

$\downarrow$ H/$p_t \rightarrow$	SS2002/SA	WS0203/SY	WS0203/SA	WS0405/SA
SS2002/SA	X	0.7916	0.6500	0.6042
WS0203/SY	-0.2653	X	0.5235	0.5437
WS0203/SA	0.4672	0.6642	X	0.7930
WS0405/SA	0.5678	0.6741	0.2778	X

Ergebnisse: Prüfungsnoten



Ergebnisse: Gesamtpunktzahlen



Ergebnisse: Einzelfragen

- Ergebnis von Einzelfragen ohne VUlab:
  - Wenig direkt vergleichbares Material, unterschiedliche Prüfungsaufgaben
  - Ergebnisse im WS0405 schlechter als andere Semester
  - Schlechte Ergebnisse nicht auf VUlab zurückführbar
- Ergebnis von Einzelfragen mit VUlab:
  - Ergebnisse SS2004 (VUlab) signifikant besser als WS0405 (kein VUlab)
  - Kein direkter Vergleich zwischen SS2004 und früheren Semestern aufgrund unterschiedlicher Prüfungsfragen

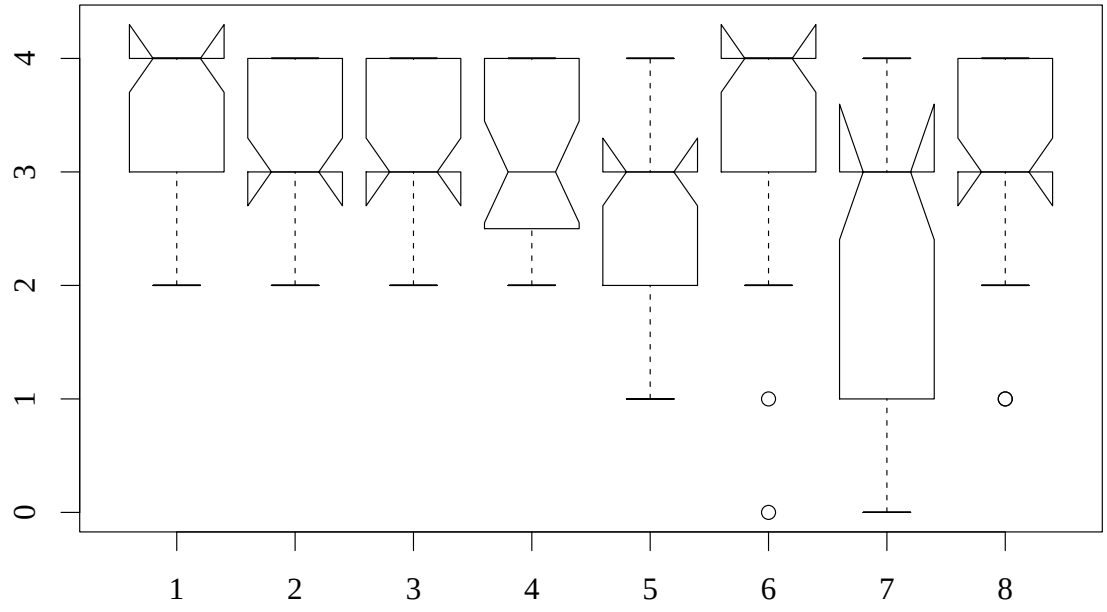
Ergebnisse: Prüfungen

⇒ Verdacht auf keinen Einfluss des VUlabs auf Prüfungsleistung liegt nahe, ist jedoch anhand der vorhandenen Prüfungsdaten nicht endgültig zu verifizieren.

Nähere Untersuchungen mit besserer Vergleichbarkeit ist angebracht, jedoch nicht im Rahmen des bestehenden Vorlesungsbetriebes umsetzbar

Ergebnisse: Lernmittel

Fragebogen: Wodurch haben Sie über das Thema Systemadministration gelernt?



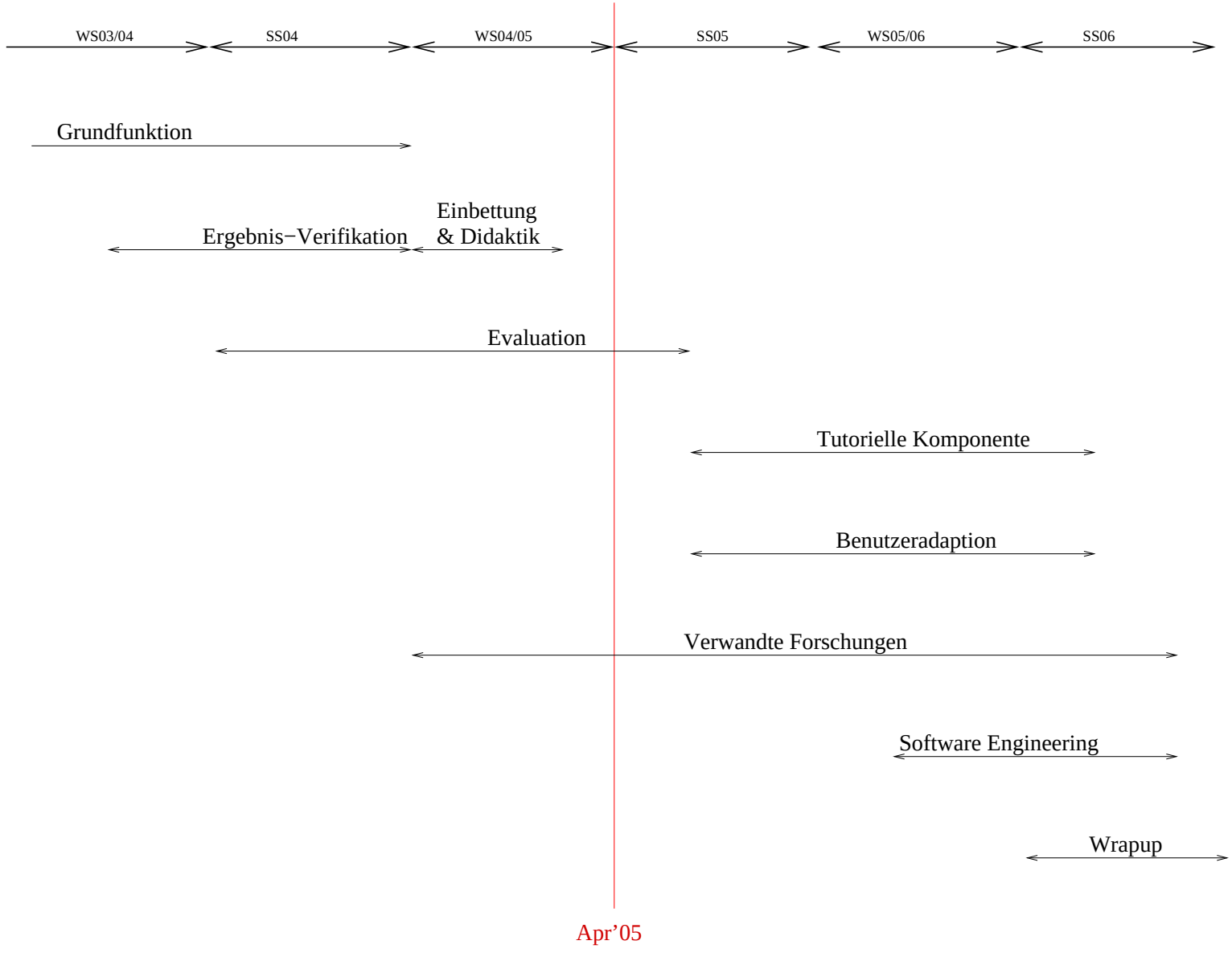
1=Lecture, 2=Script for lecture, 3=Practical exercises, 4=Virtual Unix Lab, 5=Analysing FH-machines, 6=Analysing own machines, 7=Books, 8=Online information

Evaluation: Sonstiges

- Auswertung der Ergebnisse der Übungen im VUlab
- Usability-Betrachtung für User-Interface und Übungsablauf
- Bewertung anhand von Personas bzw. durch Fokusgruppen
- Experten-Review von Übungen und Auswertungen
- etc.

Status & Ausblick

Zeitplan


WS03/04
SS04
WS04/05
SS05
WS05/06
SS06
Grundfunktion
Einbettung
& Didaktik
Ergebnis−Verifikation
Evaluation
Tutorielle Komponente
Benutzeradaption
Verwandte Forschungen
Software Engineering
Wrapup
Apr'05

Gedanken: Tutorielle Komponente

- Ersetzt in rein virtuellen Systemen den Blick des Lehrers über die Schulter
- Unterstützt Lerner bei Bearbeitung der gestellten Aufgaben
- Kooperativ bzw. automatisch
- Basierend auf Überprüfung des Zwischenergebnisses
- Für unbearbeitete Punkte werden Vorgehensweisen vorgeschlagen

Gedanken: Benutzeradaption 1

- Erstellung von Nutzerprofilen: Anfänger, Fortgeschrittener, ...
- Basierend auf
  - Übungsergebnissen anderer Lerner (bei derselben Übung)
  - vorangegangenen (unterschiedlichen) Übungen
- Unterschiedliche Hilfestellung für unterschiedliche Benutzer
- Für
  - Feedback während und nach Übungen
  - Ggf. Ausgangskonfiguration der Übungssysteme

Gedanken: Benutzeradaption 2

- Ggf. basierend auf explizit eingestellter Benutzer-Präferenzen
- Analyse basierend auf Ergebnissen einzelner Test-Gruppierungen, z.B. Editieren von Dateien, Handhaben von Bereichen wie Netzwerk, System Startup, etc.

Weitere Betätigungsfelder

- Sprachanpassung des User-Interface, zumindest Deutsch/Englisch (i18n, l10n, m17n)
- Bessere Auswertungsfunktionen
- Umstellung von realen Übungsmaschinen auf Virtuelle Maschinen (VMware, qemu, ...)
- Erweiterung um weitere Übungs-Betriebssysteme (neben Solaris, NetBSD): Linux, Windows
- Erweiterung um weitere Dienste (neben NIS, NFS): Mail, Web, Datenbanken, Firewall, Kerberos, ...
- Neben Setup-Szenarien auch Debugging- und Troubleshooting-Szenarien
- Fundraising für Entwicklungsarbeiten

Danke!

http://www.feyrer.de/vulab/

hubert@feyrer.de