



Techn. Fakultät - Erwin-Rommel-Str. 60 - 91058 Erlangen

Dipl.-Inf. Benjamin Oechslein  
(PERSÖNLICH)

## WS 12/13: Auswertung für Übungen zu Betriebssysteme

Sehr geehrter Herr Dipl.-Inf. Oechslein,

Sie erhalten hier die Ergebnisse der automatisierten Auswertung der Lehrveranstaltungsevaluation im WS 12/13 zu Ihrer Umfrage vom Typ "Übung":

- Übungen zu Betriebssysteme -

Es wurde hierbei der Fragebogen - u\_w12 - verwendet, es wurden 23 Fragebögen von Studierenden ausgefüllt.

Die Note 1 kennzeichnet hierbei eine maximale Güte, die Note 5 eine minimale Güte für die einzelnen Fragen bzw. Mittelwerte.

Auf der nächsten Seite zeigt der zuerst angegebene "Globalindikator" Ihre persönliche Durchschnittsnote über alle Kapitel-Indikatoren, deren Noten danach folgen.

Der Kapitel-Indikator für "Globalfragen für alle LV-Typen" ist trotz der Prozentangaben bei den Einzelfragen momentan noch ungewichtet, eine E-Mail mit dem daraus berechneten Lehrqualitätsindex (LQI) wird noch nachgeliefert.

Für die Ergebnisse aller Einzelfragen werden je nach Fragen-Typ die Anzahl und Verteilung der Antworten, Mittelwert und Standardabweichung aufgelistet.  
Die Text-Antworten für alle offenen Fragen sind jeweils zusammengefasst.

Auf der letzten Seite befindet sich eine Profillinie im Vergleich zu den Mittelwerten aller Rückläufer für diesen Fragebogen-Typ. Die Profillinie eignet sich auch zur Präsentation in der LV.

Eine Einordnung Ihrer Bewertung ist nach Abschluss der Ergebnisauswertung unter <http://www.tf.fau.de/studium/evaluation> --> Ergebnisse --> WS 12/13 möglich, hierzu die Bestenlisten, Percentile, etc. einsehen.

Bitte melden Sie an [eva@techfak.uni-erlangen.de](mailto:eva@techfak.uni-erlangen.de) die Anzahl der ausgegebenen TANn, wenn Sie das bis jetzt versäumt haben.

Mit freundlichen Grüßen

Michael Wensing (Studiendekan, [michael.wensing@ltt.uni-erlangen.de](mailto:michael.wensing@ltt.uni-erlangen.de))  
Jürgen Frickel (Evaluationskoordinator, [eva@techfak.uni-erlangen.de](mailto:eva@techfak.uni-erlangen.de))



## Dipl.-Inf. Benjamin Oechslein

WS 12/13 • Übungen zu Betriebssysteme  
 ID = 12w-Ü BS  
 Erfasste Rückläufer = 23 • Formular u\_w12 • LV-Typ "Übung"

### Globalwerte

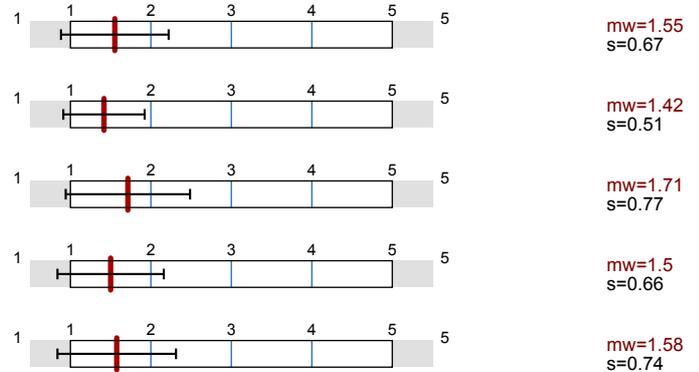
## Globalindikator

Globalfragen für alle Lehrveranstaltungs-Typen (mit Gewichtung)

Übung im Allgemeinen

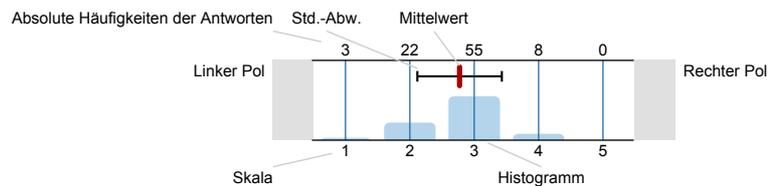
Didaktische Aufbereitung

Präsentation des Übungsleiters



## Legende

Frage



n=Anzahl  
 mw=Mittelwert  
 s=Std.-Abw.  
 E.=Enthaltung

Klick on british flag to get the english survey  
 Achtung: Beim Anklicken der Sprachsymbole verlieren Sie alle bisherigen Eintragungen !

### Allgemeines zur Person

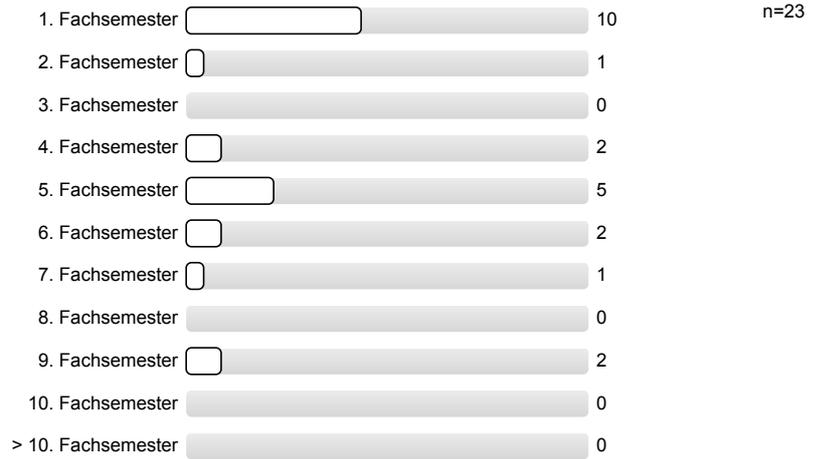
<sup>2\_A)</sup> • Ich studiere folgenden Studiengang:

INF • Informatik  20 n=23  
 IuK • Informations- und Kommunikationstechnik  3

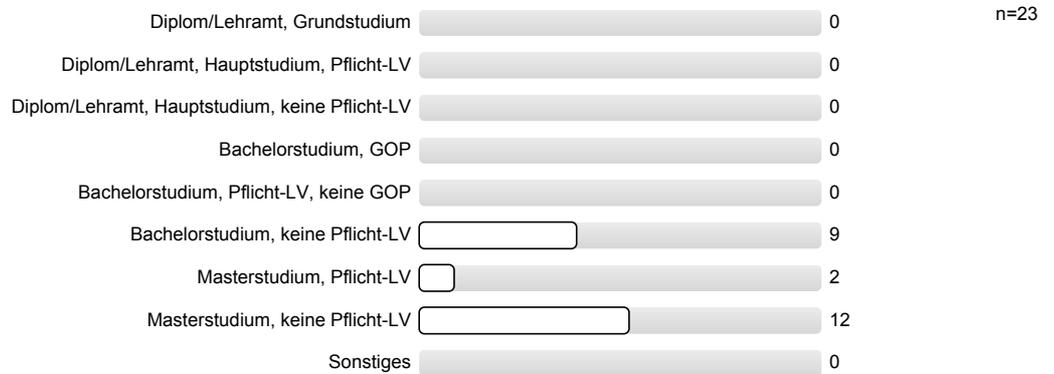
<sup>2\_B)</sup> • Ich mache folgenden Abschluss:

Dipl. • Diplom  0 n=23  
 B.Sc. • Bachelor of Science  10  
 M.Sc. • Master of Science  12  
 Staatsexamen  1  
 Dr.-Ing. • Promotion  0  
 PhD • Doctor of Philosophy  0  
 Dipl.-Ing. mit Zusatzzertifikat  0  
 M.Sc.(hons) • Master of Science with Honours  0  
 Zwei-Fach-Bachelor of Arts  0  
 Sonstiges  0

2\_C) • Ich bin im folgenden Fachsemester:

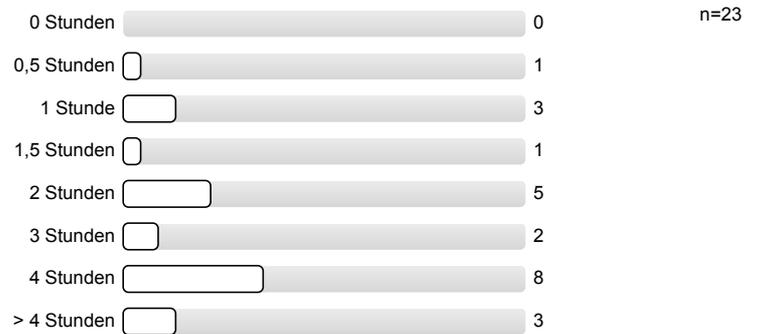


2\_D) • Diese Lehrveranstaltung gehört für mich zum . . . .



Mein eigener Aufwand

3\_A) Mein Durchschnittsaufwand für Vor- und Nachbereitung dieser Übung beträgt pro Übungsstunde (45 Min.):

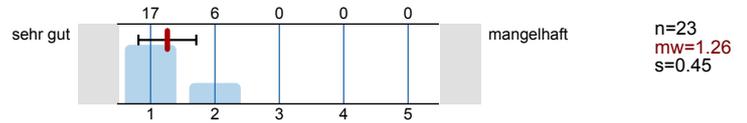


3\_B) Ich besuche etwa . . . Prozent dieser Übung.

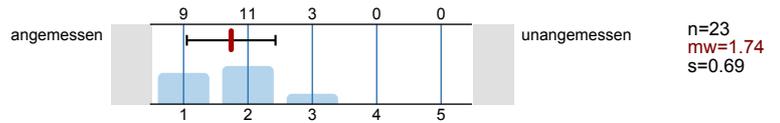


Globalfragen für alle Lehrveranstaltungs-Typen (mit Gewichtung)

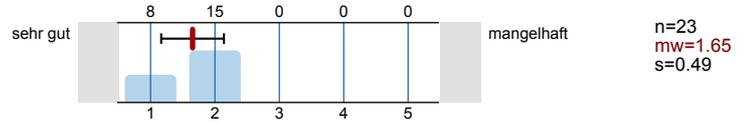
4\_A) • Bitte benoten Sie die Übung insgesamt (50%):



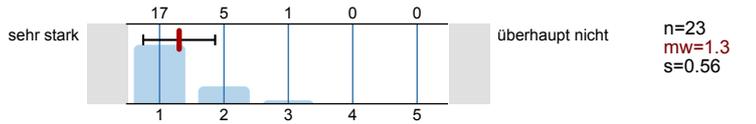
4\_B) • Der notwendige Arbeitsaufwand für diese Übung ist (12,5%):



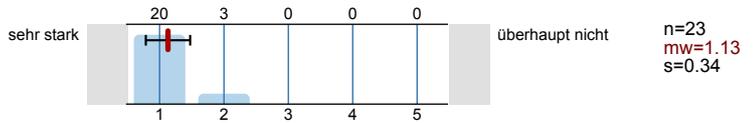
4\_C) • Wie ist die Übung strukturiert (12,5%)?



4\_D) • Der Übungsleiter wirkt engagiert und motiviert bei der Durchführung der Übung (12,5%).

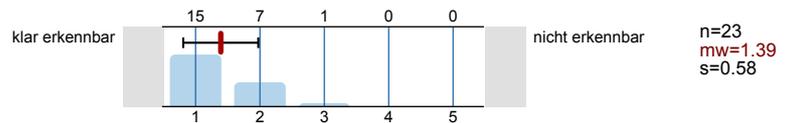


4\_E) • Der Übungsleiter geht auf Fragen und Belange der Studierenden ein (12,5%).

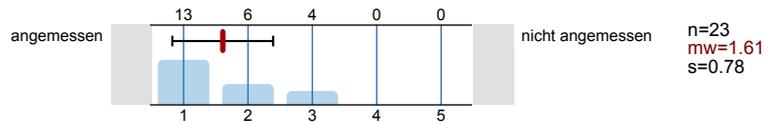


### Übung im Allgemeinen

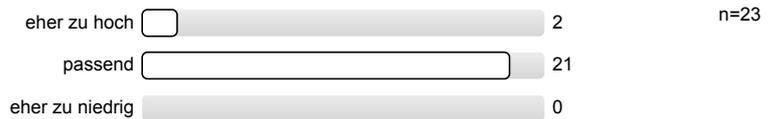
5\_A) Zielsetzungen und Schwerpunkte des Übungsinhalts sind:



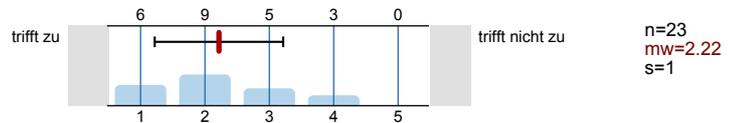
5\_B) Der Schwierigkeitsgrad der Übung ist:



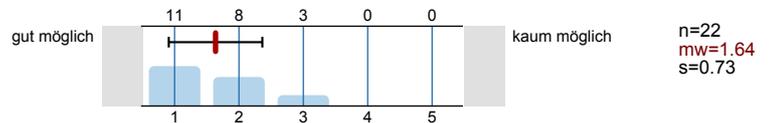
5\_C) Der Schwierigkeitsgrad der Übung ist:



5\_D) Die zur Verfügung gestellten Unterlagen sind in Menge und Qualität den Zielen der Übung angemessen.

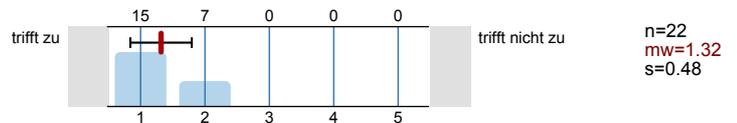


5\_E) Anhand des erarbeiteten Übungsmaterials ist die Vertiefung des Vorlesungs-/Modulinhalts:

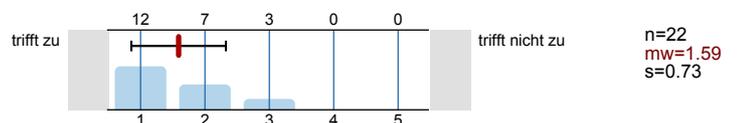


### Didaktische Aufbereitung

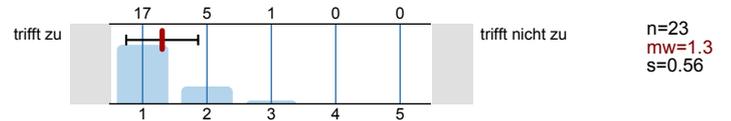
6\_A) Die Präsentation von Aufgaben und Lösungen ist nachvollziehbar, es ist genügend Zeit zum Mitdenken vorhanden.



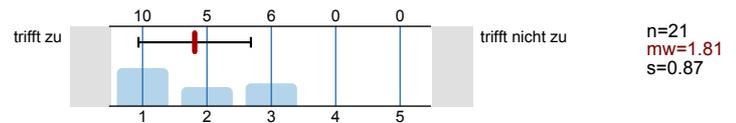
6\_B) Die Anwendbarkeit des Übungsstoffes wird z.B. durch Beispiele gut verdeutlicht.



6\_C) Die Übungsform (Aufgabenbehandlung, Programmieren, etc.) ist gut zur Vermittlung des Stoffes geeignet.

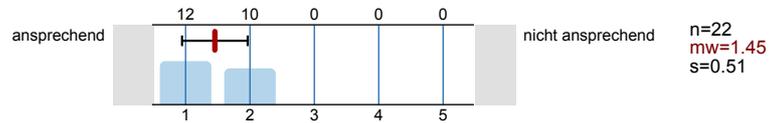


6\_D) Der Bezug zur Vorlesung und den Prüfungsanforderungen wird hergestellt.

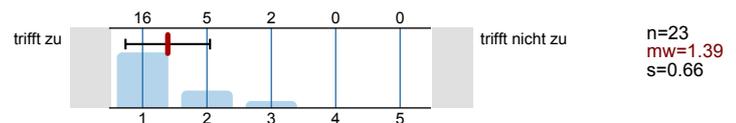


### Präsentation des Übungsleiters

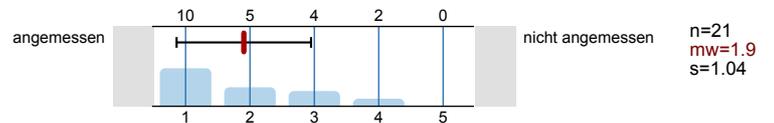
7\_A) Der Präsentationsstil des Übungsleiters ist:



7\_B) Ich werde gut zum selbstständigen Lösen von Aufgaben angeleitet.



7\_C) Der Einsatz und das Zusammenspiel von Medien (Tafel, Overhead-Projektor, Beamer, etc.) ist:



### Weitere Kommentare

9\_A) An der Lehrveranstaltung gefällt mir besonders:

- + Teamarbeit bei den Programmieraufgaben + sehr motiviertes Team von Übungsleitern + Übung wird sehr kollegial gehalten, nicht abgehoben und unantastbar wie in manch anderen Veranstaltungen + in den RU wird man bestens begleitet und unterstützt, findet man leider auch zu selten + ausreichend Zeit (3 Wochen) um die Aufgaben zu bearbeiten, verzögerte Abgaben werden auch noch besprochen
- - Man erweitert in aufeinander aufbauenden Aufgaben im Laufe des Semesters kontinuierlich ein kleines Betriebssystem, dadurch sieht man die einzelnen Komponenten jeweils gut im Zusammenhang  
- Kompetente Übungsleiter und größtenteils gut durchdachte Aufgabenstellungen
- Benjamin, Daniel und Gabor vertiefen während der Abgabe nochmals den Stoff, stellen gezielt Fragen und nehmen sich dafür genug Zeit. Das Lösen der Aufgaben macht einfach richtig Spaß :)
- Betriebssystem selber implementieren ist einfach cool ;)
- Daniel und Gabor sind hoch motiviert und sehr kompetent.
- Dass man ein eigenes Betriebssystem schreiben konnte, das Engagement der Übungsleiter, dass die Übungsleiter in den Rechnerübungen immer auf Fragen eingegangen sind und sich meist intensiv mit den Problemen auseinandergesetzt haben
- Die Implementierung des Betriebssystems ist wirklich spannend und macht viel Spaß.  
Das Abgeben der Aufgaben ist sehr gut gelöst.  
Die Übungsleiter sind motiviert und machen einen sehr kompetenten Eindruck.  
Man entwickelt sein eigenes Betriebssystem und bekommt nicht für jede Aufgabe eine neue Vorgabe (ist manchmal jedoch auch schlecht, wenn man Fehler gebaut hat, die aufwändig zu beheben sind).
- Eigenes BS Schreiben macht Spass  
Tafelanschift statt Beamer  
Rechnerübung sehr hilfreich  
Tutoren sehr kompetent
- Learning by doing.
- Motivierte Übungsleiter, interessante Aufgaben - hat mir sehr gut gefallen, weiter so.
- Sehr anspruchsvolle aber trotzdem gut lösbare Aufgaben. Man weiß, was man zu tun hat und kann auch selbstständig viel machen. Die Übung hilft perfekt, um den Stoff wirklich zu verstehen. Tutoren sind super engagiert, sehr nett und sehr hilfsbereit und auch gut verfügbar (in anderen Rechnerübungen muss man teilweise eine halbe Stunde warten, bis ein Tutor mal Zeit hat).
- Sehr gutes Übungskonzept, ein eigenes Betriebssystem ohne allzu große Vorgaben bauen ist sehr interessant und lernreich. Zudem sehr kompetente Übungsleiter!
- Viel Praxis ("mein eigenes Betriebssystem"), Theorie aus der Vorlesung kann oft auch praktisch umgesetzt werden, man erfährt damit

auch die Probleme (v.a.: Nebenläufigkeitsprobleme) selbst

- Wahnsinnig interessante Übung, die aufeinander aufbauenden Aufgaben führen zwar manchmal zu Problemen, aber es ist sehr motivierend, das eigene Betriebssystem auf echter Hardware laufen (oder eben nicht ;) ) zu sehen!
- cooles Projekt. Gut betreute Rechnerübungen (zumindest wenn sonst fast keiner kommt). Abgaben werden auch angenommen, wenn noch nicht alles perfekt ist und dürfen auch zusammen mit dem Betreuer während der Abgabe noch verbessert werden.
- nette Übungsaufgaben mit relativ umfangreicher Beschreibung
- praktische Anwendung
- sehr gutes Übungskonzept mit sehr interessanten Aufgaben die die Konzepte gut anwenden  
sehr motivierte Tutoren die auf alle Probleme eingehen und diese kompetent lösen  
beste Übung bis jetzt

<sup>9\_B)</sup> An der Lehrveranstaltung gefällt mir Folgendes weniger, und ich schlage zur Verbesserung vor:

- - Termine der RÜ nicht auf direkt aufeinanderfolgende Wochentage legen. So hat man zwischen den RÜ einer Woche selbst mehr Zeit die Aufgaben zu bearbeiten und evtl. Probleme zu identifizieren. Wir waren doch hier und da doch mal für eine Woche blockiert, weil wir auf die nächste RÜ warten mussten.
- Das vorgegebene Codegerüst hat zu viele Klassen (und ist trotzdem nicht wirklich portabel), gängige Entwurfsmuster (hint: Singleton) werden völlig ignoriert.
- Die Anzahl der Klassen/Komponenten ist zu groß. Muss ich für den Fadenwechsel nun in Guarded\_Scheduler, Scheduler, Dispatcher, Kickoff oder toc\_switch schauen? Muss ich den Faden gerade als Coroutine, Entrant, Customer, Thread oder Application betrachten? Die gewählten Verzeichnisnamen und die Gruppierung der Klassen mögen wohlüberlegt sein und Sinn ergeben, intuitiv klar sind sie aber auch beim Bearbeiten der letzten Aufgabe noch nicht.
- Die Dokumentation / Aufgabenbeschreibungen könnten auf Konsistenz überprüft werden.
- Die Klassendokumentation ist an wenigen Stellen ein wenig ungenau oder mehrdeutig
- Die Übung ist insgesamt wirklich super, drei kleine Verbesserungsvorschläge:
  - 1. Die erste Übung ist im Vergleich zu den anderen sehr aufwendig, vor allem weil man sich erst mal komplett in alles einarbeiten muss und auch noch so grundlegende Dinge von C++ verstanden werden müssen. Eine Hilfe wäre es, wenn z.B. die setRepeatRate-Methode vorgegeben werden würde und man bei der setLED-Methode sich dann daran orientieren könnte und wenn von ein paar Klassen schon die Ruempfe vorgegeben werden würden. Dann würde der Einstieg deutlich leichter fallen. So war ich kurz davor, nach der ersten Übung aufzugeben, weil ich nicht geglaubt habe, dass ein ganzes Semester lang leisten zu können und im Nachhinein muss ich sagen, wäre das ein grosser Fehler gewesen.
  - 2. Eine Zwischenabgabe in der Rechnerübung eine Woche vorher ermöglichen. Das würde die normale Abgabe am Mittwoch entzerren und man könnte eine Übung mit der man schon fertig ist früher abhaken. So kann man sich oft kaum noch erinnern, warum man manche Dinge so gemacht hat. Wenn man eine Woche vorher erlauben würde, würde die Rechnerübung ja trotzdem nicht von Abgaben ueberschwemmt, weil ja viele dann einfach noch nicht fertig wären. In EZS funktioniert das auch ganz gut.
  - 3. Die Klassendokumentation ist wirklich super, aber ohne die Tafelübung hat man trotzdem keine Chance, die Übung alleine zu schaffen. Evtl. könnte man öfter, wie bei Aufgabe 2, ein paar Folien ins Netz stellen, wo die wichtigsten Infos auf der Tafelübung zusammengefasst sind.
- Die prinzipiell wünschenswerten aufeinander aufbauenden Aufgaben haben leider auch Nachteile:
  - Manche bugs machen sich nicht gleich, sondern erst beim Bearbeiten späterer Aufgaben bemerkbar, was die Fehlersuche deutlich erschweren kann; jede Gruppe arbeitet dadurch außerdem recht bald auf einer unterschiedlichen Basis
  - Um neue Bestandteile zu integrieren, müssen oft bestehende Mechanismen "zerstört" oder in ihrem Verhalten modifiziert werden; es fällt dadurch manchmal schwer, den Überblick zu behalten, was genau diese oder jene Klasse momentan tun muss, und die einzelnen Aufgaben/Konzepte auseinanderzuhalten.
- Es ist oftmals nicht ganz nachvollziehbar in welchem Ordner sich welche Datei befindet. Manche Klassen finden sich nicht in der Dokumentation, nur in der Aufgabenstellung, besonders bei Aufgabe 6 ist die Aufgabenstellung etwas schwer nachzuvollziehen. Die Architektur von OOSTuBS lässt an einigen Stellen zu wünschen übrig.
- FCFS-Prinzip bei der Abgabe nervt. Um bei den ersten mit dranzukommen, muss man vermutlich schon am Tag zuvor seinen Namen an die Tafel schreiben. Würde mich nicht wundern, wenn jemand das bald mit wasserfestem Edding macht. Wäre doch mal nett die Liste nach LCFS-Prinzip abzuarbeiten.  
Zusätzlich zu den Rechnerübungen wäre noch die ein oder andere Übung schön, die etwas auf die Anforderungen in der Prüfung hinweist.  
Komisches Klima im CiP-Pool. Entweder ich zerfließe oder ich erfriere. Aber da könnt ihr ja auch nix für.
- Ich hätte mir Folien zu den Übungen gewünscht.
- Klassenvorgaben/-hierarchie manchmal etwas seltsam. Statt der uebermaessigen Vererbung koennte man auch einfach nur eine Methode hinzufuegen/veraendern (z.B. Organizer)
- Man weiß manchmal bis zur Abgabe nicht wirklich, ob das, was man programmiert hat, funktioniert oder doch kaputt gehen kann (insbesondere bei Themen wie Synchronisation) - sonst wirklich top.
- Manche Dokumentationen (zu) gut auf Website versteckt

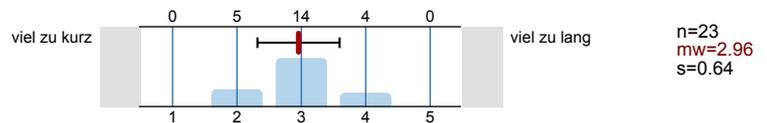
- Objektorientierungs-Overkill. Viele Klassen sind einfach unnötig viel Code für elementare Dinge (Plugbox) oder werden in Instanzmethoden gepackt, ohne, dass das Objekt einen Zustand hätte (CPU::disable\_int()). Häufig dauert es auch eine Weile, bis man sich in der Vielzahl von Klassen zurecht gefunden hat (Locker vs Guard vs Secure, Bell vs Bellringer vs Buzzer, ...)
- Ziemlich viel in Richtung Parallellität, Threads etc. Vielleicht ein bisschen mehr Richtung Treiber etc? Z.B. ordentliche Darstellung von Grafik o.Ae.?
- die Reihenfolge der Übungsgruppen bei der Abgabe sollte zufällig sein, FCFS ist nicht wirklich fair wenn sich Leute schon mittags eintragen, die Übung aber erst um 4 beginnt ..
- uebungsfolien waeren hilfreich, fuer den fall das man eine uebung nicht besuchen konnte.

9\_C) Zur Lehrveranstaltung möchte ich im Übrigen anmerken:

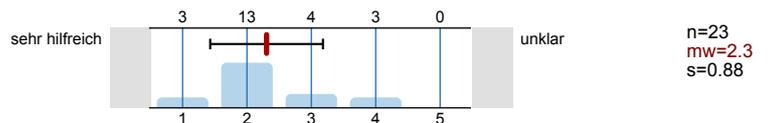
- Da vom Lehrstuhl die Möglichkeit EZS und BS zusammen zu machen waren weniger Überschneidungen wünschenswert Mo14-16 Do 14-16 da auch noch andere Fächer genug Überschneidungen liefern
- Daniel, Benjamin, Gabor, ihr seid ein tolles Team!
- Der Bearbeitungszeitraum von 3 Wochen für eine Aufgabe ist schon recht lang
- Die an sich guten Tafelanschriften nochmal aufbereitet in Folienform zur Verfügung zu haben, wäre nützlich!
- Die Übungsleiter sind hilfsbereit und immer gut drauf.
- Es ist selten SO kompetente und freundliche Betreuer gestellt zu bekommen.
- Es wäre schön, wenn man die in der Übung besprochenen Inhalte auch zusammengefasst als pdf von der Website bekommen könnte (mit den Dingen, die in der Übung als wichtig zu beachten genannt wurden)
- Freu mich schon auf BST!
- Hat sehr viel Spass gemacht!
- Im großen und ganzen eine sehr empfehlenswerte Veranstaltung.
- debuggen manchmal etwas aufwendig, aber man gewöhnt sich (meistens) dran ;-)

Optionale Zusatzfragen des Übungsleiters

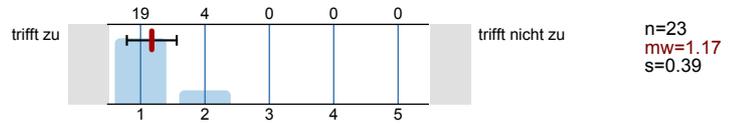
10\_A) Die Bearbeitungszeit für die einzelnen Aufgaben fand ich (Mitte: genau richtig)



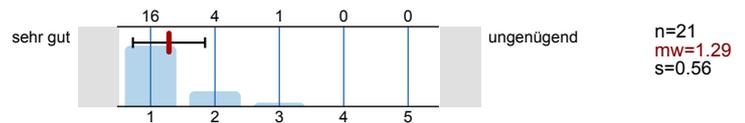
10\_B) Die Aufgabenstellungen und die Dokumentation auf der Webseite fand ich



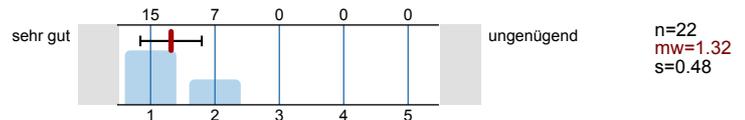
10\_C) Der Praxisanteil (Entwurf und Implementierung des OOSTuBS/MPStuBS-Übungsbetriebssystems) hat mir gut gefallen.



10\_D) Das Betreuungsverhältnis in den Rechnerübungen fand ich



10\_E) Insgesamt bewerte ich die Rechnerübung mit der Note

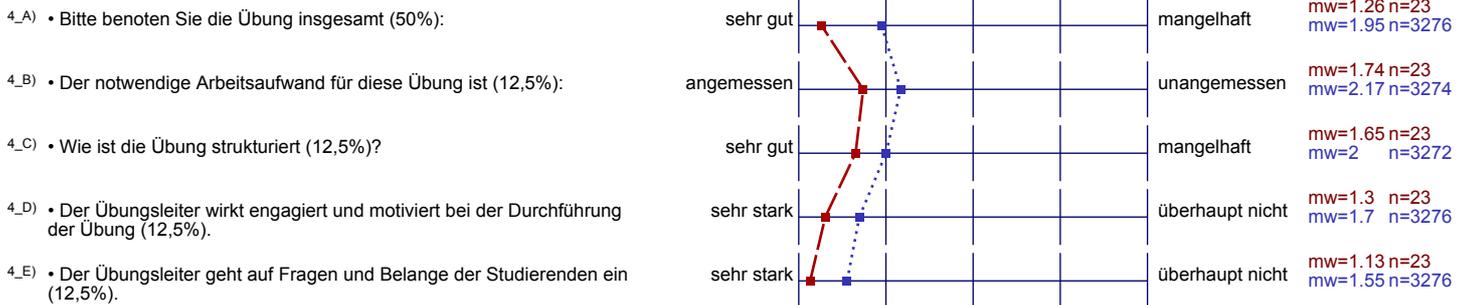


# Profillinie

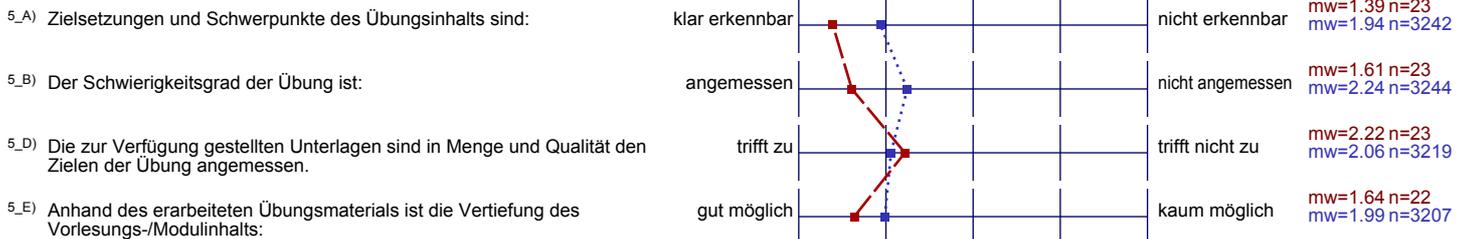
Teilbereich: Technische Fakultät (TF)  
 Name der/des Lehrenden: Dipl.-Inf. Benjamin Oechslein  
 Titel der Lehrveranstaltung: Übungen zu Betriebssysteme (12w-Ü BS)  
 (Name der Umfrage)

Vergleichslinie: Mittelwert aller Übungs-Fragebögen im WS 12/13

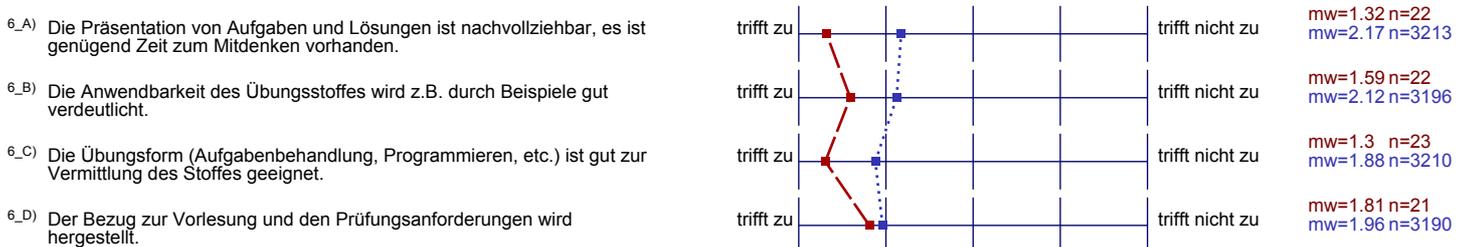
## Globalfragen für alle Lehrveranstaltungs-Typen (mit Gewichtung)



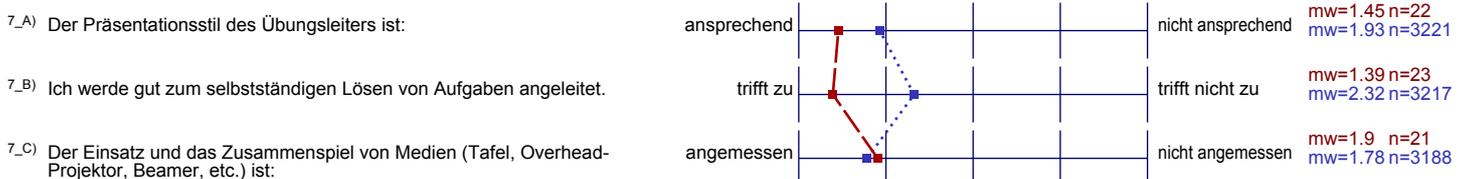
## Übung im Allgemeinen



## Didaktische Aufbereitung



## Präsentation des Übungsleiters



Optionale Zusatzfragen des Übungsleiters

