



FAU • Dekanat der TF
Martensstraße 5a
91058 Erlangen

FAU • Dekanat der TF • Martensstr. 5a • 91058 Erlangen

Herr
M. Sc. Bernhard Heinloth
(PERSÖNLICH)

SS22: Auswertung zu "22s-Ü BST"

Sehr geehrter Herr M. Sc. Heinloth,

im Rahmen der Lehrveranstaltungsevaluation im SS22 erhalten Sie hiermit die Auswertung zu Ihrer Umfrage vom Typ "Übung":

- Übungen zu Betriebssystemtechnik -

Es wurden hierfür 21 Fragebögen vom Typ "t_s22_v+ü42" von den Studierenden ausgefüllt.

Die 4 Indikatoren zeigen den mit der Anzahl der Antworten gewichteten Mittelwert der Skalafragen in den genannten Fragenkapiteln.

Der Mittelwert der 4 Indikatoren bildet den Globalindikator bzw. den Lehrqualitätsindex (LQI).

Für die Einzelfragen und Indikatoren kennzeichnet der Wert 1 hierbei eine maximale Güte, der Wert 5 eine minimale Güte.

Bei den Einzelfragen werden je nach Fragen-Typ die Anzahl und Verteilung der Antworten, Mittelwert und Standardabweichung aufgelistet.

Die Text-Antworten für jede offene Frage sind zusammengefasst aufgelistet.

Die Profillinien zeigen den Vergleich zu den Mittelwerten aller Rückläufer der Technischen Fakultät.

Der LQI und die Indikatoren werden bei genügend (ab 5) Rückläufern zur Qualitätssicherung durch die Studienkommissionen und die Erstellung der Bestenlisten verwendet.

Mit freundlichen Grüßen

Rolf Wanka (Studiendekan, tf-studiendekan-lehre@fau.de)
Jürgen Frickel (Evaluationskoordinator, tf-evaluation@fau.de)



Globalwerte

Globalindikator

Indikator • Organisation, Inhalte und Kompetenzen der LV (Kap. 3)



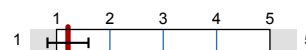
mw=1,23
s=0,42

Indikator • Struktur der LV (Kap. 4)



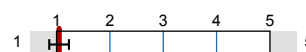
mw=1,3
s=0,55

Indikator • Durchführung der LV (Kap. 5)

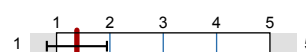


mw=1,21
s=0,38

Indikator • Zufriedenheit und Kompetenzerwerb (Kap. 6)



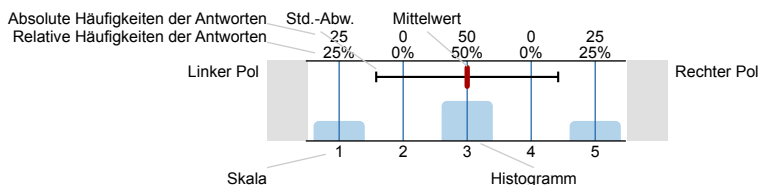
mw=1,05
s=0,18



mw=1,38
s=0,56

Legende

Fragetext



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

2. Studierender und Lehrveranstaltung

2.1) Ich studiere folgenden Studiengang und Abschluss:

INF • Informatik (B.Sc.) n=21

INF • Informatik (M.Sc.)

2.2) Ich bin im folgenden Fachsemester (im Bachelor):

1. Fachsemester n=2

2. Fachsemester

3. Fachsemester

4. Fachsemester

5. Fachsemester

6. Fachsemester

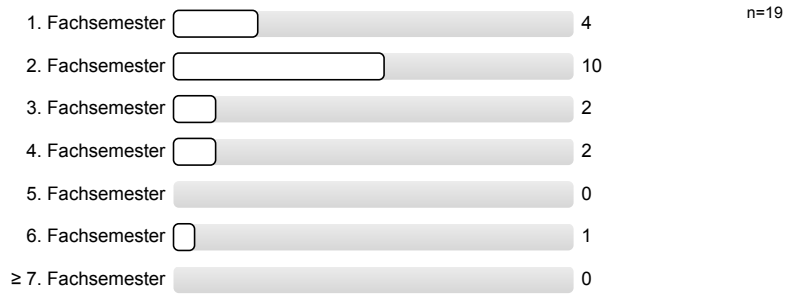
7. Fachsemester

8. Fachsemester

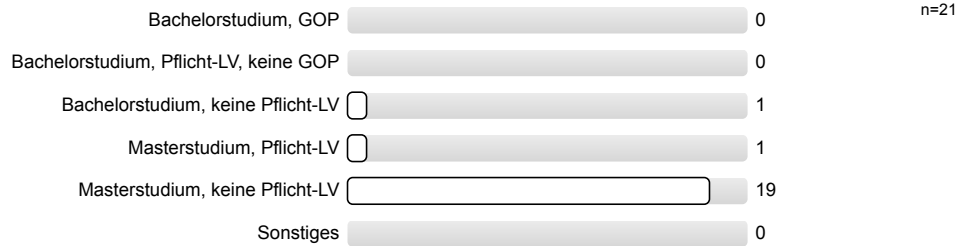
9. Fachsemester

≥ 10. Fachsemester

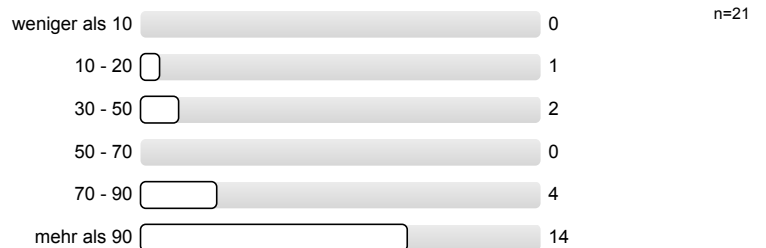
2.3) Ich bin im folgenden Fachsemester (im Master):



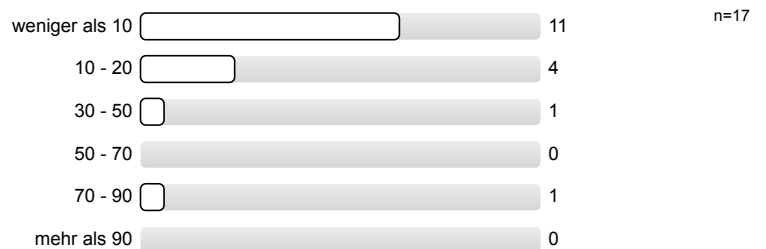
2.5) Diese Lehrveranstaltung (LV) gehört für mich zum
 ("keine Pflicht-LV": Wahl einer anderen, alternativen LV wäre möglich)



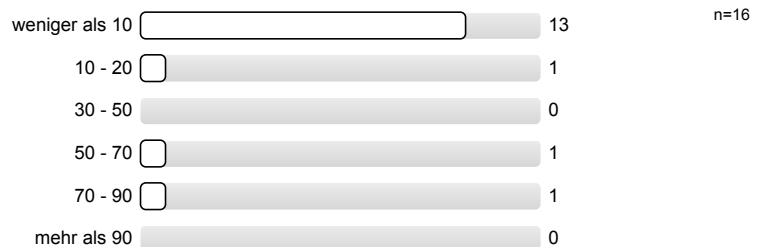
2.7) An Prozent dieser LV habe ich synchron in Präsenz teilgenommen.



2.8) An Prozent dieser LV habe ich synchron online (Zoom, Teams, o. ä.) teilgenommen.

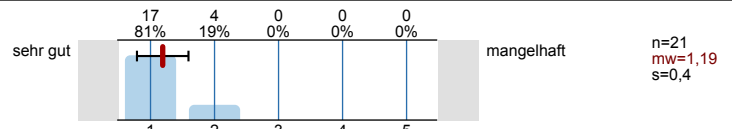


2.9) An Prozent dieser LV habe ich asynchron (Aufzeichnung, o. ä.) teilgenommen.

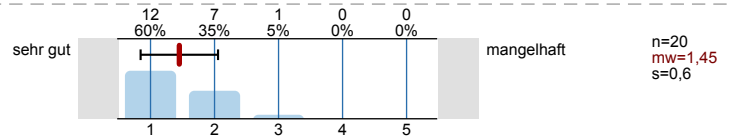


3. Organisation, Inhalte und Kompetenzen der Lehrveranstaltung

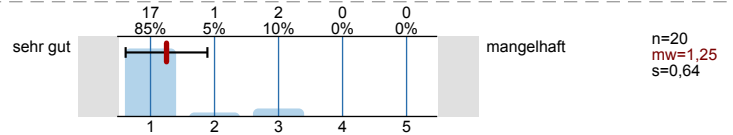
3.1) Wie gut war die Durchführung der LV organisiert?



3.2) Wie gut war die LV inhaltlich organisiert und mit evtl. zugehörigen LVen abgestimmt (Vorl. • Übg. • Prakt. • ...)?

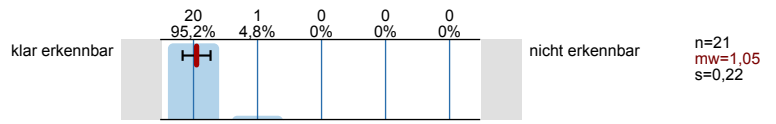


3.3) Die LV entspricht den im Modulhandbuch eingetragenen Inhalten und Kompetenzen.

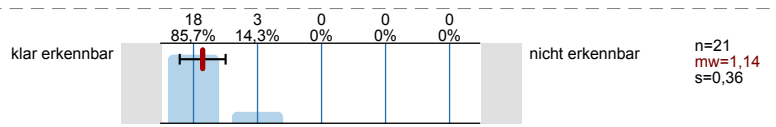


4. Struktur der Lehrveranstaltung

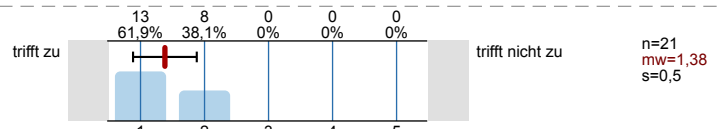
4.1) Zielsetzungen und Schwerpunkte des Inhalts waren:



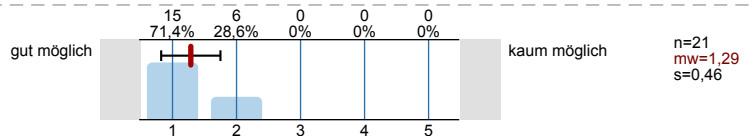
4.2) Der rote Faden der LV (synchron bzw. asynchron) war:



4.3) Der dargebotene Stoff war nachvollziehbar, es war genügend Zeit zum Mitdenken vorhanden.

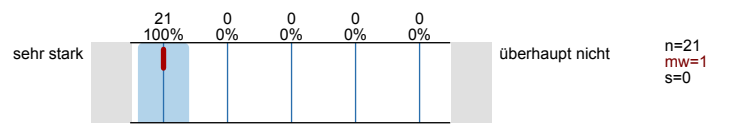


4.4) Mit den Medien, Begleitmaterialien, Literaturhinweisen und Hinweisen in der LV selbst waren Vor- und Nachbereitung:

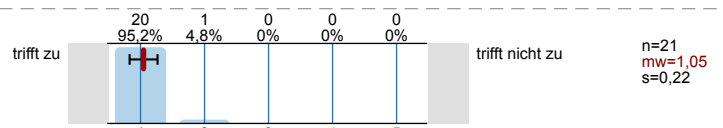


5. Durchführung der Lehrveranstaltung

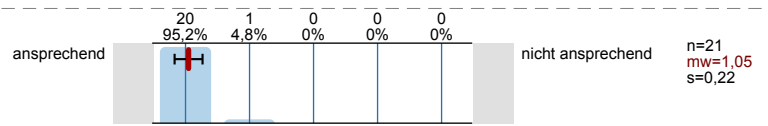
5.1) Die Dozentin/Der Dozent wirkte engagiert und motiviert bei der Durchführung.



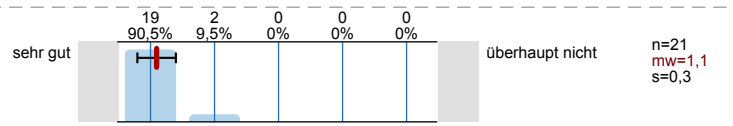
5.2) Die Dozentin/Der Dozent förderte das Interesse am Themenbereich.



5.3) Der Präsentationsstil der Dozentin/des Dozenten war:

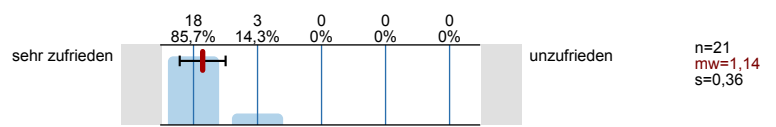


5.4) Die Dozentin/Der Dozent ging auf Fragen und Belange der Studierenden ein (synchron und asynchron).

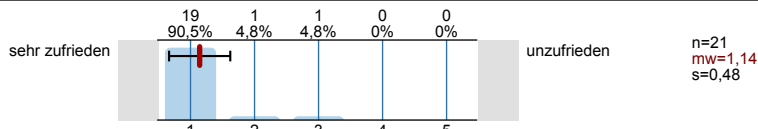


6. Zufriedenheit und Kompetenzerwerb

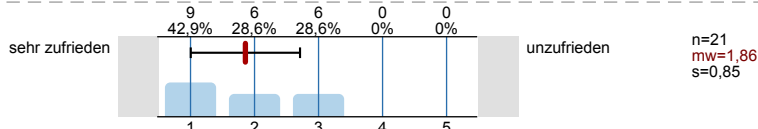
6.1) Wie zufrieden sind Sie insgesamt mit der LV?



6.2) Wie zufrieden sind Sie mit der LV bezüglich Ihres eigenen Kompetenzerwerbs?

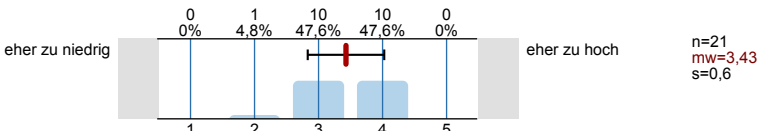


6.3) Wie zufrieden sind Sie mit dem Verhältnis zwischen Lernerfolg/Kompetenzerwerb und eigenem Zeitaufwand?

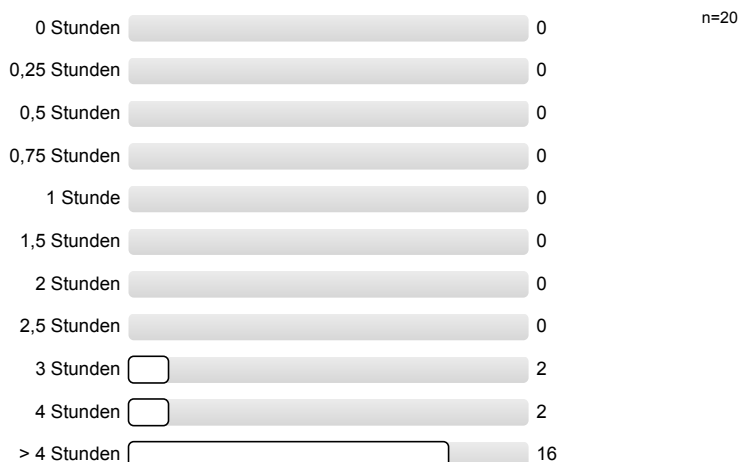


7. Zusätzliche Informationen für die Dozentin/den Dozenten

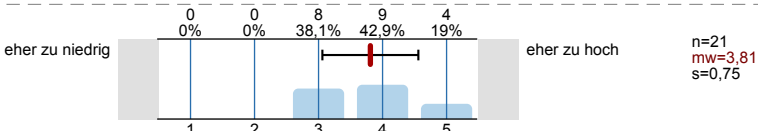
7.1) Der Schwierigkeitsgrad des Stoffes war für mich:



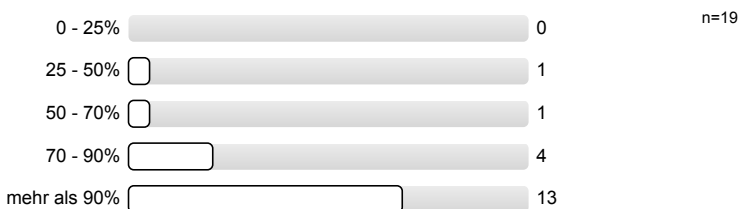
7.2) Mein Durchschnittsaufwand für Vor- und Nachbereitung dieser LV (ohne den LV-Besuch) betrug pro Woche:



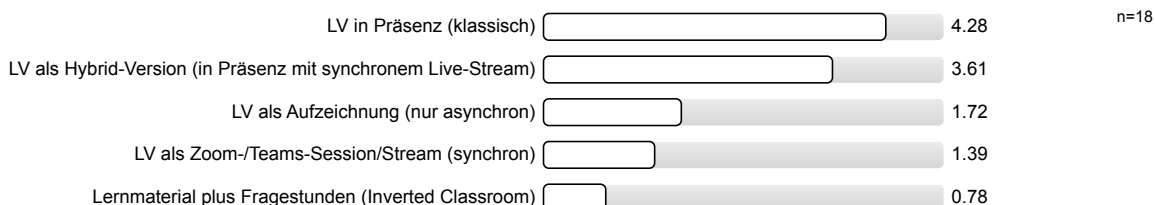
7.3) Meinen zeitlichen Durchschnittsaufwand für diese LV fand ich:



7.4) Ich habe bei etwa Prozent dieser LV zeitnah (mit höchstens 1 Woche Verzug) mitgearbeitet.



7.5) Welche Darbietungs-Form hätten Sie sich für diese LV gewünscht (bitte mit Drag-and Drop nach Priorität sortieren)?



7.6) An der Durchführung der Lehrveranstaltung gefielen mir folgende Aspekte besonders:

- Tafelübung hat Spass gemacht
- Erzählstunden in der RÜ waren super
- Übungsleiter waren immer erreichbar

- - Übungen bauen schön aufeinander auf
- Bernhard ist top motiviert, allerdings sagt er zu selten "historisch gewachsen" :(
- Man lernt echt viel, vor allem nochmal zu dem ganzen Kernel/Userspace + Isolation
- Bernhard gibt den coolen Onkel, den sich ein jedes pubertierende Betriebssystem wünschen würde, während es selbstbestimmte mitunter wilde Veränderungen an seiner Code Basis durchlebt.
Er hat immer ein offenes Ohr für Probleme und gibt weise Ratschläge, z.B. wenn sich das heranwachsende Betriebssystem dafür schämt das sich sein 4-stufiges Paging nicht schnell genug entwickelt. Oder gibt ein gutes Vorbild ab wie es sich von seinem kindlichen Userspace trennen kann.
Auf der anderen Seite unterstützt er es auch wenn es sich von seinen alten Binärenfreunden trennt und nun lieber mit den coolen rauchenden Elfen abhängen will oder sich um einen eigenen Zoo mit Zackenbärschen, Feldsperlingen und anderen wilden ASCII Tieren kümmern will.
Nebenbei führt er noch einen gefragt Aufklärungsblog auf dem er noch zusätzliche Ratschläge, z.B. wie man sicher neue Kinder erzeugen kann, veröffentlicht.
Bis es schlussendlich selbst am Ende des Semesters mit seiner Kuhherde in die Freiheit des Erwachsenwerdens entlassen wird.
Abschließend kann man ihm nur noch eine hoffentlich erfolgreiche Dissertation und im Anschluss noch viele entspannte nachuniversitäre Abende in der nicaraguanischen Hängematte wünschen. ;)

Die anderen beiden Betreuer Philipp und Dustin sollen zum Schluss auch nochmal lobend erwähnt werden und werden hoffentlich die Veranstaltung in den kommenden Semestern weiterverfeinern und so weiterhin so gut zu führen.

- Bernhard ist immer sehr enthusiastisch und das ist ansteckend. Außerdem finde ich toll wie Bernhard die herangehenden sehr gut und ausführlich erklärt und direkt aufzeigt wo man gerne was vergisst. Man macht diese Fehler erstens wahrscheinlich trotzdem und zweitens würde es wahrscheinlich niemandem helfen die Lösungsansätze vorzuenthalten. BS programmieren ist so schon aufwändig genug.
- Bernhard ist sehr sehr engagiert. Obwohl man an der ein oder anderen Stelle gemerkt hat, dass die Umstellung auf 64 Bit Probleme mit sich brachte, reagierte er sehr professionell und solide. Kurzfristige Blogbeiträge und asynchroner Support via Issue aus Git haben neben den vielen Rechnerübungen sehr geholfen.
- Die Möglichkeit, zur Not auch Online an der Lehrveranstaltung teilnehmen zu können, fand ich sehr gut.
- Die hilfreichen Rechnerübungen und der schnelle Support über Gitlab-Issues
- Die Übung hat (fast) immer sehr viel Spaß gemacht, nicht zuletzt durch die humorvolle Art der Betreuer. Ich finde es super, dass bereits während des Semesters auf Schmerz und Leid reagiert wurde und fehlendes Wissen mittels der Emails und der Blogbeiträge ergänzt wurde. Das Angebot der virtuellen Teilnahme im Krankheitsfall war super, auch wenn ich es nicht wahrnehmen musste, aber gerade in Coronazeiten doch sehr sinnvoll.

Wie gewohnt die beste Veranstaltung des Semesters (trotz des Schmerzes) :)

- Dozenten sehr motiviert!
- Gute 24/7 Betreuung! Vielen Dank nochmal an alle Tutoren die in der RÜ, im IRC und bei den Gitlab Issues geholfen haben :)
- Meines Erachtens war der Kompetenzerwerb in BST deutlich höher als in BS, da man in letzterem eigentlich nur sehr grundlegende Teile eines Betriebssystems implementiert hat. BST war meines Erachtens deutlich anspruchsvoller und ist erheblich mehr in die Tiefe gegangen. Somit hat man auch wirklich verstanden, wie Teile eines Betriebssystems funktionieren und insbesondere, wie Isolation mit all den ausgefallenen Techniken (Paging, Systemcalls, Ringwechsel) im Detail umgesetzt wird. Deshalb war auch mit einem der Fächer, dem ich in diesem Semester einen besonderen Kompetenzerwerb zuschreiben würde. Das ist somit fast auch als Kritik an BS zu verstehen, weil ich bei letzterem, gerade auch weil es deutlich einfacher war, nicht so viel ggü. SP gelernt habe und somit BST fast schon als "Pflichtfach" nach BS ansehe, wenn man sich für das Thema interessiert.

Allerdings gebe ich zu, dass mich das Fach auch etwas abgeschreckt hat, mich dauerhaft mit Betriebssystementwicklung, welche über die konzeptionelle Ebene hinausgeht, zu beschäftigen ...

- Sehr interessant

7.7) An der Durchführung der Lehrveranstaltung gefiel mir Folgendes weniger, und ich schlage zur Verbesserung vor:

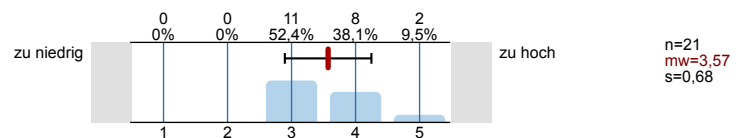
- - Man fühlt sich manchmal ziemlich ins kalte Wasser geschmissen (ich schau eigentlich nur Aufgabe 3 und bisschen die 4 an)
- Da hat Bernhard dann schon Schadenskontrolle gemacht, vom zeitlichen Aufwand wars aber recht extrem.
Immerhin erweitert man nochmal seine i4-antrainierte Frustrationstoleranz
- Dazu dann als Folge der zeitliche Aufwand: Mit gerade mal 17,5 ECTS plus Tutor dieses Semester waren die Wochen (plus Wochenenden) oft randvoll gefüllt
- Der Arbeitsaufwand der Übungsaufgaben war sehr hoch.
- Wäre gut, wenn man uns nicht immer sagt, dass die nächste Aufgabe (nach der 3. Aufgabe) leichter wird... mittlerweile vertraue ich dieser Aussage nicht. Die Aufgaben nach der 3. hatten zwar nicht so einen riesigen Umfang, aber ich fand sie trotzdem schwer :(
- Aufzeichnungen / Streams der Tafelübung für alle!

7.8) Sonstiges:

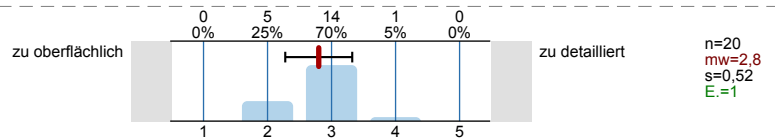
- BST a.k.a Bug Such Tortur
- Man sollte den Übungsleitern nicht mehr hoffnungsvoll glauben wenn diese sagen "die nächste Aufgabe wird einfacher" :D

8. Zusatzfragen von der Dozentin/vom Dozenten

8.1) Der Umfang der Aufgaben ist:



8.2) Die Aufgabenstellung ist:



8.3) Weitere Anmerkungen zu den Aufgaben bezüglich Vorgaben, Umfang, Inhalt, Bearbeitungsaufwand:

- - BST ist so aufwändig, dass es alleine etwas anstrengend ist...
 - ich fand es nicht so tragisch, dass es hin und wieder noch geholpert hat. Cool, dass ihr euch die Mühe macht
 - die beiden einwöchigen Aufgaben waren relativ knapp bemessen. V.a. fork() + map() war im Vergleich zu dem quasi schon vorgegebenen send/recv/reply sehr aufwändig
- A3 war die die wirklich schlimme Aufgabe, A4 auch noch bissl. Rest war zwar sehr aufwaendig, aber im i4-Rahmen.
- Die Aufteilung ist gut gelungen, der Zeitpunkt der "Bergwoche" im Plan der Aufgaben ist nicht optimal aber lässt sich vermutlich nicht unbedingt verhindern. Eine Aufgabe schon vorgestellt zu haben, an der man auch während der "Bergwoche" arbeiten kann wäre gut.
- Die Erweiterung der Aufgabenstellung auf 7 Aufgaben finde ich gut. Die Alternative aus SS19 mit den 3-4 größeren Aufgabenblöcken stelle ich mir deutlich schwerer vor.
 - Da es kaum Vorgaben gab, wie genau die zu implementierenden Funktionen auszusehen haben, hatte man zwar immense Freiheiten, was aber leider idR dazu führte, dass bereits implementierte Teile in nachfolgenden Aufgaben mehrmals neu bearbeitet werden mussten.
 - Die Übungsaufgaben halfen sehr, die Funktionsweise der behandelten Betriebssystemkonzepte zu verstehen, und motivierten, sich näher mit dem Thema zu beschäftigen.
 - Der gesamte notwendige Bearbeitungsaufwand war dennoch deutlich zu hoch.
- Die Folien sind erste Klasse und immer hilfreich bei der Bearbeitung der Aufgaben.
- Es war durch weniger Vorgaben im Code zwar schwieriger als BS, aber ich habe das Gefühl, dadurch auch mehr gelernt zu haben. Nicht gerade selten waren wir in späteren Aufgaben gezwungen, älteren Code nochmal umzuschreiben oder zu erweitern, wodurch die Abgaben mehr Projektfeeling bekommen haben.
 - Die Schwierigkeit war zwar relativ hoch, aber durch die ausgezeichnete Hilfestellung hatte ich nie das Gefühl, etwas nicht schaffen zu können. Würde diese Schwierigkeitsstufe beibehalten, sonst verfehlt man glaube ich die Realität, in der BS-Entwicklung einfach hart ist und man schon ganz genau wissen muss, was man tut.
 - Der Umfang war allerdings schon sehr hoch. BST hat den Großteil meines Semesters verschlungen, ich hatte eigentlich 4 Module belegt, wobei ich für 1 weiteres mit Pflichtabgaben noch ein wenig Zeit aufwenden musste und die beiden anderen dann mangels Zeit einfach hinten runtergefallen sind und ein Fall für die Semesterferien werden. Das weiß man aber vorher denke ich und mir fällt auch wenig sinnvolles ein, wie man das verbessern kann (das Aufteilen der Aufgaben hat es wahrscheinlich schon erheblich verbessert).
- Ich finde, dass der Aufwand im Verhältnis zu BS wesentlich höher ausgefallen ist und dass deswegen auch eine höhere ECTS-Bewertung (mind. 10ECTS) angestrebt werden sollte. (Auch damit man in dem Semester, in welchem man BST belegt, mehr Zeit für BST hat und nicht mit den ganzen anderen Modulen in Verzug gerät)
 - Im Allg. sind ja die Module des I4 immer sehr Aufwendig, aber insbesondere BST ist meiner Meinung nach viel zu "unterbewertet" (was die ECTS anbelangt)
- Ist viel Arbeit, aber das kommt nicht unerwartet
- Nachdem dies die erste Durchfuehrung der Uebung auf 64bit war, gab es ein paar Stolpersteine, die nicht ersichtlich aus der Aufgabenstellung waren. Allerdings denke ich, dass mit den Informationen aus dem Blog durchaus einige, wenn nicht sogar alle, damit beseitigt werden.
- War sehr gut eine "Prufferwoche" nach Aufgabe 3 (paging) zu haben, da diese von sehr vielen gebraucht wurde um die Aufgabe noch fertig zu machen bzw. das BS wieder zum laufen zu bringen.
- sehr viel Bearbeitungsaufwand.. aber finde ich nicht schlimm. kann auch n paar zeitintensivere module geben :)

8.4) Haben Sie (gerne auch konkrete) Vorschläge, wie wir die Bearbeitung einzelner Aufgaben(teile) noch mehr unterstützen können?

- Allgemein wäre es trotz freiem Software-Engineering hilfreich, wenn gewisse Andeutungen zu möglichen Datenstrukturen in der Übung besprochen würden. Das wurde teilweise später per Mail nachgetragen, sodass man, wenn man früh angefangen hat, oft erst danach mitbekommen hat, wie man es eigentlich richtig tun würde (wobei ich da wegen des ersten BST-64bit-Durchlaufs drüber hinwegsehen kann). Mir geht es auch mehr um Hinweise in der Tafelübung, wo man z. B. bespricht wie: "Verwendet für die Freispeicherverwaltung zwei LinkedLists und speichert für jeden Knoten die Startadresse und Anzahl an freien Seiten ab" (konzeptionelle Dinge sind oft genauso hilfreich wie vorgegebene Methodenrümpfe)

Auch eine kleine Einführung zu "Wie benchmarke ich richtig?" (evtl. als Seminar) wäre nett. Da könnte man auch etwas über die reine Anwendung in StuBS rausgehen, und einen kleinen Einblick in andere Gebiete (z. B. WCET) geben.

- Bei der Paging Aufgabe, fand ich es super hilfreich als in der RÜ erwähnt wurde, dass man ja "nur" 3 Methoden (map, resolve, unmap) schreiben muss. Finde, dass die Methoden Signatur auf jeden Fall auch in die Aufgabenbeschreibung getan werden soll.

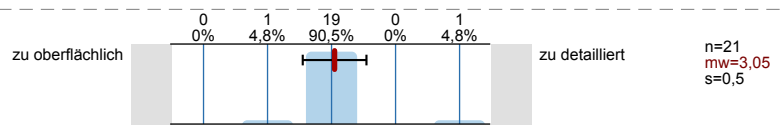
- Die Aufgabenstrukturierung war an sich gut die aber die Hauptschwierigkeiten waren definitiv Aufgaben 3-5. Bei der Paging Aufgabe ist der Hauptzeitfresser definitiv die Free Memory Liste gewesen, an sich ist dies auch keine algorithmisch schwere Aufgabe jedoch ist das beachten aller möglichen Randfälle und deren sinnvolles testen doch schwerer als gedacht gewesen. Man könnte die Aufgabe sicher dadurch entschärfen zumindest ein Skelett für die zu verwendende Linked List bereitzustellen. Bei der Userspace Separierung war die Schwierigkeit am Anfang vor allem das schreiben der Makefiles und das verstehen des Linkers, was mit reinem SP wissen nicht so trivial ist. Dies wurde zwar zum Glück sehr schnell nachgeliefert, aber vllt könnte man in Zukunft hierfür ein eigenes Make und Linker Seminar anbieten um das ganze vorher nochmal zu lernen. Die Forkaufgabe hatte eigentlich nur das Problem das die eine Woche eigentlich zu kurz für ihre Bearbeitung war und man durch das Fehlen eines Abgabegesprächs vllt den einen oder anderen konzeptionellen Fehler mehr hat der sich dann vor allem in der COW Aufgabe rächt.
- Die Blogposts und Hinweise in den Aufgabestellungen sind hilfreich, da wo man sie nicht gebraucht hat, kann man sie auch einfach nur überfliegen.
- Ich denke, dass die Blogbeiträge tatsächlich ausreichen, um die Aufgaben sinnvoll bearbeiten zu können. Mehr RUE wären natürlich immer schön, aber unrealistisch.
- Obwohl zu Beginn der Lehrveranstaltung angemerkt wurde, dass sich mehr Freiheiten bei der Implementierung gewünscht wurden, hätte eine vorgegebene Struktur den Arbeitsaufwand bestimmt reduziert, da man dann eventuell seltener bereits implementierte Teile vorheriger Aufgaben refactoren müsste. Oft war der aufwendigste Teil einer Aufgabe das Finden, Behandeln und Debuggen unzähliger Randbedingungen. Hier würde ich mir mehr Testprogramme (ähnlich wie der Testcode beim PageFrameAllocator, eventuell auch UnitTests) wünschen, damit man diese Randfälle frühzeitig erkennen und bearbeiten kann, und diese einem nicht in den folgenden Teilen einer Aufgaben das Debuggen erschweren.
- Ofters die typischen Fallstricke (die nach dem ersten 64bit-Durchlauf ja jetzt bekannt sein sollten) noch als Kästchen (auch nochmals, halt hervorgehoben) in die Aufgabenstellungen rein. Beispiel:
 - Bei A3 nicht die vollen 4 GB pro App mappen

Etwas mehr Rahmenkonstruktion bieten, damit die Implementierungen nicht völlig in die Wallachei abhauen, z.B. wie die dann vorgeschlagenen Paging-Schnittstellen von Bernhard

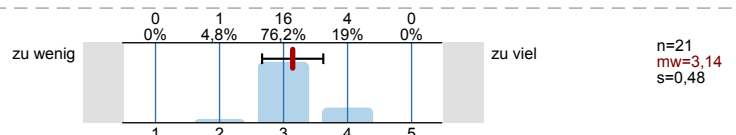
- Vielleicht könnte man bei der Paging Aufgabe mehr Support bei der Implementierung der Liste liefern. Wie man eine Liste grundsätzlich baut, wird wohl allen klar sein, aber an der eigentlichen Implementierung gibt es dann doch sehr viel Fehlerpotential, was einem aber erst hinterher in Form von komischen Pagefaults um die Ohren fliegt. Die Liste war für uns auch um einiges komplizierter als die Logik des Paging selbst, teilweise haben wir noch in Aufgabe 5 Bugs in der Liste gefunden.. Das könnte vielleicht in Form einer definierteren Codevorgabe für die Listenaufgabe passieren mit Methodensignaturen und Beschreibungen. Weglassen würde ich das Selbst-Implementieren der Liste aber nicht.

Bei Aufgabe 4 war es irgendwie schade, dass uns das Makefile schreiben dann doch komplett abgenommen wurde. Ohne die Vorgabe hätten wir zwar wahrscheinlich ewig gebraucht (falls es am Ende überhaupt funktioniert hätte), aber so fühlt es sich auch ein wenig falsch an. Vielleicht könnte man ein zusätzliches Seminar über Makefiles (über den Stoff von SP hinaus) und konkret die BS(T) Makefiles einfügen oder mehr Hilfestellung in der Aufgabenstellung einfügen, so dass zumindest Teile davon doch selbst geschrieben werden können.

8.5) Die Tafelübung ist



8.6) Die (über die Webseite erreichbaren) weiterführende Informationen sind:



8.7) Weitere Anmerkungen bezüglich der angebotenen Hilfestellungen (Webseite/Blog, Tafel- und Rechnerübungen sowie Chat/Mails/GitLab):

- Blog war super. Mails gehen doch gerne mal unter
- Die Blogbeiträge sind super.
- Die Foliensätze der Tafelübung sind zur Nachbereitung sehr umfangreich. Eine Version ohne Animationen wäre wünschenswert aber ist verständlicherweise beim ersten Durchlauf nicht verfügbar
- Die RÜs und auch die schnelle Bearbeitung der Issues sind wirklich sehr hilfreich, vielen Dank an euch Dustin, Philipp und Bernhard :)
- Die Tafelübungen und Informationen auf der Website waren sehr detailliert und halfen sehr bei der Bearbeitung der Aufgaben. Den Weblog fand ich super! Die ergänzenden Hinweise/Erklärungen/Informationen waren sehr hilfreich und informativ! In den Rechnerübungen waren die Tutoren eine sehr gute Unterstützung zum Finden und Lösen sehr hartnäckiger Fehler. Bei auftretenden Problemen/Fragen außerhalb der Rechnerübung erhielt man immer zeitnah Hilfe im Chat.
- Fand die Anmerkungen über GitLab hilfreich
- Generell wäre es für die späteren Aufgaben evtl. sinnvoll in einer FAQ Page oder ähnlichem die generischen Fehlerbilder pro Aufgabe kurz zusammenzufassen. Beispielsweise das ein vergessener TLB flush zu womöglich sehr komischen Race Conditions führen kann etc. Bei den meisten Aufgaben wurde dies zwar mittels der Blogbeiträge nachgeliefert, aber vllt wäre eine zentrale Stelle da passender und würde die Betreuer ein bisschen entlasten.

Die Idee mit den Merge Requests ist an sich sehr gut, jedoch sollte für die positive Work/Lehre Balance des Betreuer Teams die Möglichkeit für Issues die außerhalb normaler Arbeitszeiten mitunter bearbeitet wurden, stärker einzuschränken für den absoluten Notfall oder bei Krankheit vor allem sollte die Pandemie nächsten Sommer noch weiter eingedämmt sein.

- Hab letztenendes kaum darauf zugreifen wollen/müssen, aber ich denke die Chats nochmal deutlicher hervorzuheben/den einfachsten Weg zu zeigen wie man sich einklinken kann (wahrscheinlich Rocketchat?) wäre hilfreich.
- Ich fände es gut, wenn GitLab Issues weiterhin angeboten werden, da man hier konkret auf Probleme des eigenen Codes eingehen kann. Ansonsten würden die Tutoren sehr viel Zeit mit einer Gruppe in der RÜ verbringen, um alles zu debuggen.
- Super, finde ich toll. Sollten so weiter aufrecht erhalten werden. Das Feedback auf GitLab zeigt auch öfter noch ein paar Denkfehler oder Implementationsfehler auf, die man selbst gar nicht gesehen hat.
- Zusätzliche Anmerkungen zu den Aufgaben waren teilweise über viele Kommunikationskanäle geteilt, z. B. Mail, Blog auf der Webseite, i4pad. Es wäre wünschenswert, wenn das auf einem Ort platziert werden würde, wo man das dann auch übersichtlich später nachschlagen kann.

Die Länge mancher Hinweise waren von der Länge her auch etwas zu viel des Guten (z. B. die Hinweise des Linkers).

- alles super und zeigt wieder das hohe engagement. eventuell aber ein paar zu viele kanaele..
- super!

8.8) Wie fanden Sie den Abgabemodus (Gespräch in der Rechnerübung, Codereview via GitLab)?

- - Gitlab-Nutzung war gut. Anmerkungen sind so direkt im Code
- Am liebsten waere mir, komplett online abzugeben. Aber an sich gut, da man direkt Feedback erhaelt und noch Fehler direkt begleichen kann.
- An sich ist die Codereview via GitLab sicher eine schöne Ergänzung. Allerdings empfand ich in BS (vor Corona) damals gerade die Abgabegespräche als mit das lehr- und hilfreichste in der Übung und sie haben auch einfach Spaß gemacht. Deshalb fand ich es in BST etwas schade, dass diese oft eher kurz oder nicht behandelt wurden. Ist natürlich bei der hohen Teilnehmeranzahl verständlich, vielleicht kann man da in zukünftigen Jahren mit weniger Studierenden wieder mehr den Fokus drauf legen.
- Das Gespräch ist sehr gut und gibt einem nochmal ein besseres Verständnis für den Inhalt und der Merge Request hilft nochmal kleine Unsauberheiten zu verbessern.
- Den Abgabemodus (Gespräch/Vorstellung in der Rechnerübung + Merge Request + ggf Nachbearbeitung einzelner Codeabschnitte) fand ich gut.
Manchmal kamen die Codereview Issues allerdings sehr kurzfristig vor Beginn der neuen Übungsaufgabe.
- Die Mischung aus Codereview und Abgabe im Gespräch ist auch wenn es viel Aufwand auf Seite der Korrektoren bedeutet sehr hilfreich.
- Es war unklar, ob es in Ordnung ist, sich die Arbeit aufgabenweise mit dem Gruppenpartner zu teilen. Das hat bei Irritationen bei einer Abgabe geführt. Ich denke es wäre gut, solche Dinge im Vorhinein klar zu besprechen.
- Gut.
- Hervorragend. Bernhard an den Eigenheiten von C++ verzweifeln zu sehen ist eine der schönsten Momente von BST. Was zum Teufel ist eigentlich ein 0x800000000_chongus ?
- Ich fand die Abgabe am besten, als man in der RÜ grob über alles geredet hat (evtl. da schon was ausgebessert hat) und dann im MR noch Details/Randfälle korrigiert wurden, die man bei der RÜ nicht gesehen hat.
- Sehr gut, zumindest besser als nur jeweils eines von beidem.
- Viel angenehmer in der Rechnerübung, weil ein Gespräch über die Abgabe mehr Lernpotenzial hat als die Abgabe über GitLab.
- gut
- habe primaer via gitlabd abegeben. hat meistens sehr gut funktioniert.

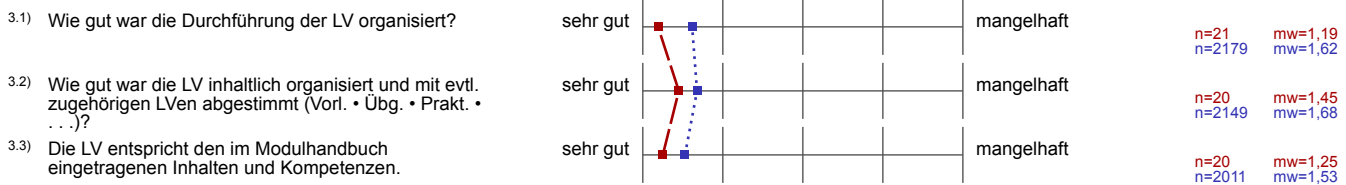
Profillinie

Teilbereich: TF • Virtual-Class-Umfragen
 Name der/des Lehrenden: M. Sc. Bernhard Heinloth
 Titel der Lehrveranstaltung: Übungen zu Betriebssystemtechnik
 (Name der Umfrage)

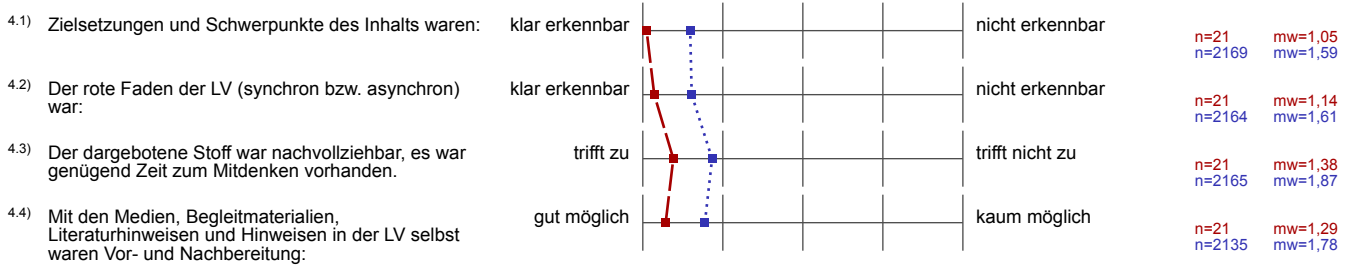
Vergleichslinie: Mittelwert aller Übungs-Fragebögen im SS'22

Verwendete Werte in der Profillinie: Mittelwert

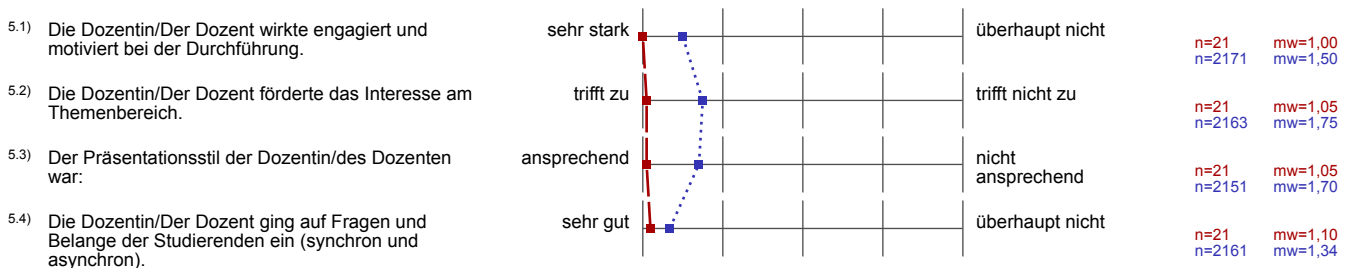
3. Organisation, Inhalte und Kompetenzen der Lehrveranstaltung



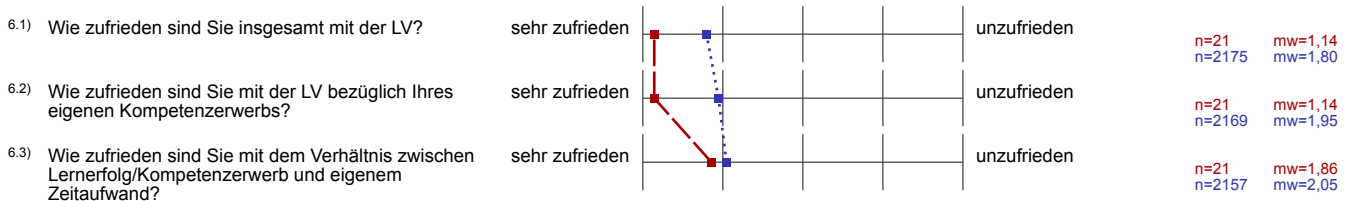
4. Struktur der Lehrveranstaltung



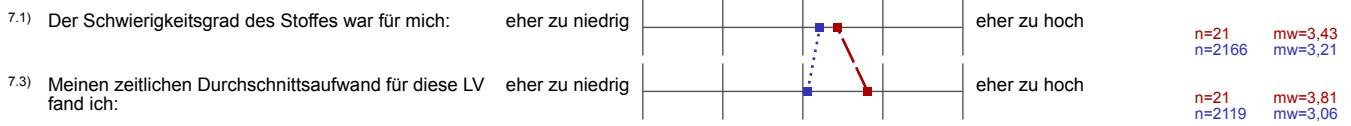
5. Durchführung der Lehrveranstaltung



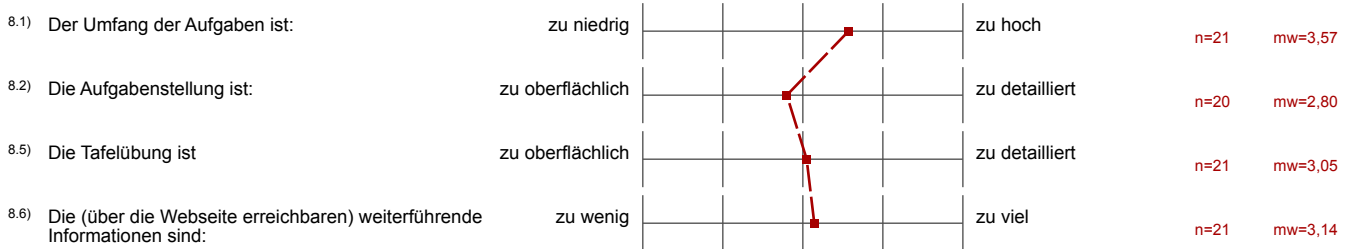
6. Zufriedenheit und Kompetenzerwerb



7. Zusätzliche Informationen für die Dozentin/den Dozenten



8. Zusatzfragen von der Dozentin/vom Dozenten

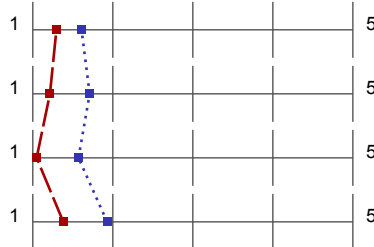


Profillinie für Indikatoren

Teilbereich: TF • Virtual-Class-Umfragen
 Name der/des Lehrenden: M. Sc. Bernhard Heinloth
 Titel der Lehrveranstaltung: Übungen zu Betriebssystemtechnik
 (Name der Umfrage)

Vergleichslinie: Mittelwert aller Übungs-Fragebögen im SS'22

Indikator • Organisation, Inhalte und Kompetenzen der LV (Kap. 3)



mw=1,30 s=0,55
mw=1,61 s=0,80

Indikator • Struktur der LV (Kap. 4)

mw=1,21 s=0,38
mw=1,71 s=0,89

Indikator • Durchführung der LV (Kap. 5)

mw=1,05 s=0,18
mw=1,57 s=0,84

Indikator • Zufriedenheit und Kompetenzerwerb (Kap. 6)

mw=1,38 s=0,56
mw=1,93 s=0,98