Teaching HCI to Hundreds of Undergraduate Software Engineering and Computer Science Students

Keith Andrews

Graz University of Technology, Graz, Austria
kandrews@tugraz.at
https://isds.tugraz.at/keith

Abstract. As HCI has become more mainstream, introductory HCI courses have transitioned in many universities from more specialised elective courses taught in postgraduate degree programmes to compulsory courses taught in the first or second year of undergraduate degree programmes. At many universities, this transition means that class sizes can jump from one or two dozen students to many hundreds of students. This paper collects some of my experiences and advice for teaching HCI to such large class sizes, including redesigning the course to an online environment during the COVID pandemic.

Keywords: human-computer interaction · education · teaching · large class size · high enrolment · thinking aloud · heuristic evaluation · Sapphire.

1 Introduction

In summer semester 1990, Human-Computer Interaction (HCI) was first offered at Graz University of Technology as an elective course for Master's students in Technical Mathematics / Computer Science (TM), which was a five-year straight Master's degree (Diplomstudium). Ludwig Reinsperger and Peter Sammer taught the first iterations of the course, using Ben Shneiderman's classic textbook [1].

I began teaching part of the course in 1992 and have been solely responsible for it since 1997 [2], as can be seen in Table 1. As time progressed, I wrote my own set of lecture notes, now comprising some 242 pages (PDF) [3], which have been used by dozens of other HCI educators around the world. For specific parts of the course, I recommend a variety of books and resources.

During the 1990s, I came to the realisation that it would be beneficial for students to come into contact with the principles of HCI much earlier in their academic careers, rather than just prior to finishing their Master's degree. As planning got underway at the university for the introduction of separate Bachelor's and Master's degrees (a result of Austria's participation in the Bologna

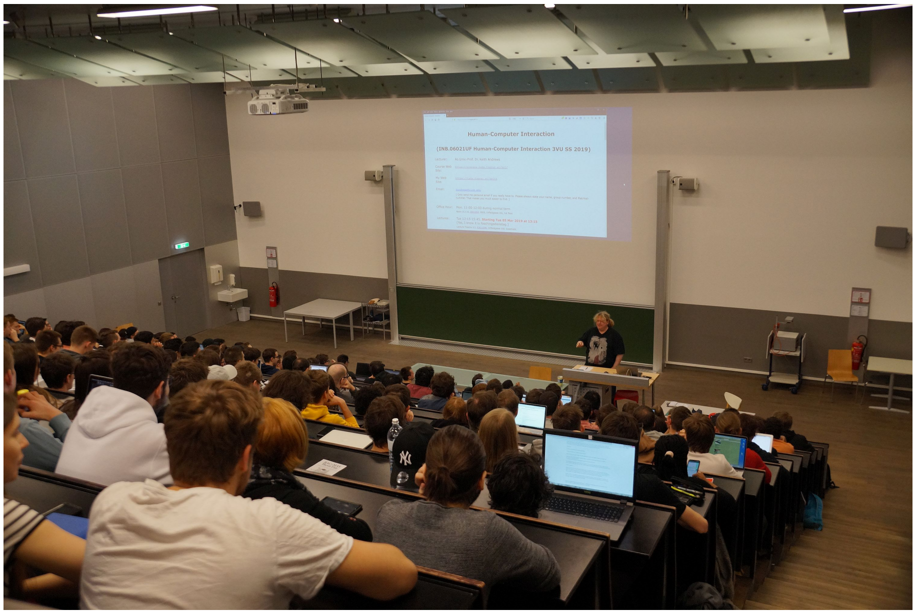

Human-Computer Interaction
(INB.06021UF Human-Computer Interaction 3VU SS 2019)
Lecturer:
Ao.Univ.-Prof. Dr. Keith Andrews
Course Web Site:
My Web Site:
Email:
Office Hour:
Mon. 11:00-12:00 during normal term
Lectures:
Tue 13:15-15:45. Starting Tue 05 Mar 2019 at 13:15
[Yes, I know it is Faschingsdienstag.]

Fig. 1: The first lecture of the HCI course in summer semester 2019.

Process), I successfully argued that HCI should be taught much earlier: in the first year of the Bachelor's degree.

Starting in the summer semester of 2003, I taught the HCI course as a compulsory course for first-year Bachelor students of Software Engineering and Management (SEM), scheduled as 3 contact hours per week in the 2nd semester and worth 4.5 ECTS credits. Throughout the 1990s, the numbers of students had remained relatively manageable, typically a few dozen (say 30–50) participants. With its new placement as a compulsory course in the second semester of the Bachelor's degree, at a stroke, in 2003, the number of participants jumped tenfold: from a few dozen to several hundred, as can be seen in Table 1.

Upto 2003, the HCI course had been formally structured into separate but linked lectures and practicals, with students technically able to take one without the other, although they were strongly encouraged to take both together. In 2004, the course was formally unified into a single entity: lectures with integrated practicals, with 3 contact hours (3VU). In 2006, the HCI course was included as a compulsory course into the newly introduced Bachelor's degree in Computer Science (CS), also in the 2nd semester.

2 Handling the Numbers

Early considerations included how to handle the supervision of practical work and how to manage the grading of hundreds of students. Since it would be

Table 1: Evolution of the HCI course at Graz University of Technology. SS indicates summer semester, WS indicates winter semester. The three lecturers involved are Ludwig Reinsperger (LR), Peter Sammer (PS), and Keith Andrews (KA). The three degree programmes are the Master's degree in Technical Mathematics / Computer Science (TM), and the Bachelor's degrees in Software Engineering and Management (SEM) and in Computer Science (CS). The number of students is derived from the number of issued certificates (both passes and fails).

Year	**Lecturer(s)**	**Degree**	**Semester**	**Students**	**Remarks**
1990 SS	LR	TM	8		
1991 SS	LR, PS	TM	8	37	
1992 SS	LR, PS, KA	TM	8	18	
1993 SS	PS, KA	TM	8	59	
1994 SS	PS, KA	TM	8	52	
1995 SS	PS, KA	TM	8	46	
1996 SS	PS, KA	TM	8	44	
1997 SS	KA	TM	8	49	
1997 WS	KA	TM	8	27	
1998 SS	KA	TM	8	28	
1998 WS	KA	TM	8	31	
1999 SS	KA	TM	8	46	
1999 WS	KA	TM	8	29	
2000 SS	KA	TM	8	40	
2000 WS	KA	TM	8	36	
2001 SS	KA	TM	8	52	
2001 WS	KA	TM	8	47	
2002 SS	KA	TM	8	64	
2002 WS	KA	TM	8	66	
2003 SS	KA	SEM	2	293	→ BSc
2004 SS	KA	SEM	2	250	→ 3VU
2005 SS	KA	SEM	2	203	
2006 SS	KA	SEM + CS	2	245	
2007 SS	KA	SEM + CS	2	237	
2008 SS	KA	SEM + CS	2	217	
2009 SS	KA	SEM + CS	2	205	
2010 SS	KA	SEM + CS	2	212	
2011 SS	KA	SEM + CS	2	218	
2012 SS	KA	SEM + CS	2	222	
2013 SS	KA	SEM + CS	2	229	
2014 SS	KA	SEM + CS	2	250	→ Sapphire
2015 SS	KA	SEM + CS	2	282	
2016 SS	KA	SEM + CS	2	263	
2017 SS	KA	SEM + CS	2	344	
2018 SS	KA	SEM + CS	2	307	
2019 SS	KA	SEM + CS	2	313	
2020 SS	KA	SEM + CS	4	30	74 → COVID
2021 SS	KA	SEM + CS	4	222	COVID
2022 SS	KA	SEM + CS	4	252	COVID
2023 SS	KA	SEM + CS	4	278	

impossible for a single lecturer to handle such large numbers, it was decided that a number of tutors (teaching assistants) would be assigned to the course. In addition, grouping students into groups of four for the practical exercises, rather than dealing with individual submissions, also helps keep the logistics more manageable. In summer semester of 2019, the last regular iteration of the course before the COVID restrictions, at the start of the course there were 330 students, in 91 groups of 4 (or 3), with 9 tutors (see Fig. 2); 313 completed the course, in terms of being issued a certificate at the end of the course, including both passes and fails.

Part of the logistics of dealing with large numbers of participants is responding to changes. Some students drop out a few days or sometimes even several weeks into the course. Sometimes, students are unresponsive, leaving their group colleagues in the lurch. Other times, members of a group simply do not get on with each other and request to move to another group. Managing these changes is often more time-consuming than preparing the lectures. Having a small first exercise towards the beginning of the course helps ameliorate this by forcing students to meet up in their groups, collaborate, and hand in some work. Those inclined to potentially drop out, then at least do so early on, causing less disruption.

I usually meet my tutors once a week for two hours, with extra meetings scheduled as required, particularly around grading after submission deadlines. Each tutor typically supervises around 10 groups of 4 students, responding to questions by email and holding three face-to-face meetings with each group as the term progresses. The final exam (MC Test) is held in tutorial groups, in-person, in a large lecture theatre at the end of the course.

3 Course Coverage

The HCI course at Graz University of Technology is an introductory course, aimed at providing an overview of the field to computer science and software engineering students. The field of HCI is extremely broad, as illustrated in Fig. 2, so choices regarding scope and focus had to be made. I chose to focus the theoretical part of the course (the lectures) on the bottom right corner ("Development Process" and "Computer") of Fig. 2, while only touching the other two areas ("Human" and "Use and Context"). This seemed to be a good match to the computer science and software engineering students at the university.

In terms of the practical part of the course (the practical exercises), two possible avenues were explored:

1. Have the students design and build an interface.
2. Have the students evaluate an existing interface.

I decided to focus on evaluation rather than design for the following reasons. Firstly, second semester students could not be assumed to have a solid enough background in programming to build user interfaces, which would require provision of significant extra support. Secondly, grading evaluation reports seemed

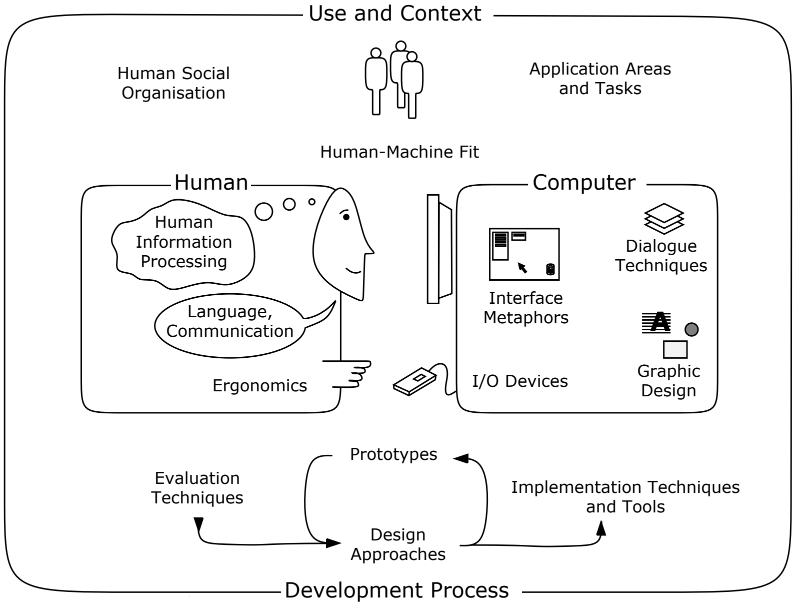


Fig. 2: The broad field of HCI. Redrawn and adapted from Fig. 1 of the classic ACM SIGCHI Curricula for Human-Computer Interaction [4, page 16].

to be easier and less subjective than grading user interface designs, particularly with large numbers of students. Thirdly, having students run user tests at this early stage of their studies would give them the potentially belief-changing experience of users thinking and acting differently to their own expectations, and the practical experience of running a user test would hopefully remain with them for the rest of their studies and careers.

4 Practical Exercises

A real-life scenario is used to make the practical work more realistic. Each student group plays the role of usability consultants who have been contracted to evaluate a web site. The tutor plays the role of their client, the manager of the assigned web site. At the start of the semester, each tutor suggests two publicly available web sites, which are suitable for evaluation in the course. They must be a) neither too good nor too bad in terms of their usability, i.e. offering enough issues to find, whilst not being disastrously poor, and b) not too small, offering enough scope to explore.

The five practical exercises used in the HCI course are shown in Table 2. In essence, the exercises can be boiled down to three tasks:

- Heuristic Evaluation (HE): Plan and Report.
- Thinking Aloud Test (TA): Plan and Report.

Table 2: The five practical exercises which contribute to the final grade, as used in summer semester 2023.

Ex	Title	Type	Points	%
1	Heuristic Evaluation Plan (HE Plan)	group	43	6.5
2a	Heuristic Evaluation (Individual Evaluations)	individual	70	10.6
2b	Heuristic Evaluation Report (HE Report)	group	116	17.5
3	Thinking Aloud Test Plan (TA Plan)	group	58	8.7
4a	Thinking Aloud Test (TA Report)	group	192	29.0
4b	Thinking Aloud Test (TA Full Videos)	group	84	12.7
5	Multiple Choice Test (MC)	individual	100	15.1
			663	**100.0**

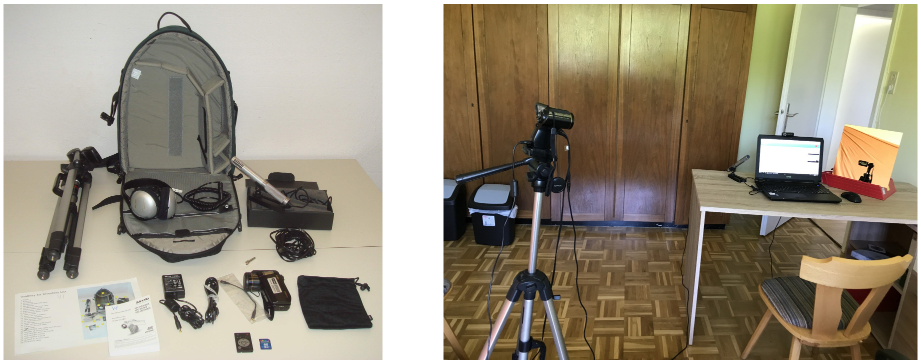
– Multiple Choice Test (MC).

For each of these, detailed instructions [5] and an extensive set of materials [6] are provided. Each group presents their work at three face-to-face meetings: M1 (HE Plan), M2 (HE Report and TA Plan) and M3 (T3 Report).

At the start of the semester, students perform a heuristic evaluation [7, 8] (HE) of a (publicly available) web site. The first task is to write a HE Plan (Ex1) [9]. This also functions as a starting point to ensure that group members make contact with one another and are serious about participating in the course. Then, each of the students in the group serves individually as an evaluator and assembles a list of positive and negative findings accompanied by short video clips (Ex2a). Finally, the students come together to aggregate their individual findings and write the HE Report (Ex2b) [10].

In the middle of the semester, students run a face-to-face thinking aloud test (TA) [11, 12] of the same web site with five test users, who they recruit from among their friends, family, and colleagues. Again, the first task is to write a TA Plan (Ex3) [13]. Then, the group performs a pilot test with one user followed by the real test with four users. Five usability kits with recording equipment are available for students to borrow, one is shown in Fig. 3a. A typical test setup is shown Fig. 3b. Screen recording is done on the laptop with a webcam. Additionally, the external video camera records the screen, keyboard, mouse, and user's facial reactions in the mirror placed next to the laptop. Finally, the students analyse their results, formulate findings, produce video clips to illustrate each finding, and write their TA Report (Ex4a) [14]. Full session recordings are handed in offline and graded separately (Ex4b).

The multiple choice test (MC Test) at the end of the semester (Ex5) assesses knowledge of the theoretical material contained in the lecture notes and covered in class. Ten questions are asked, each with four parts. Rather than one of the four parts being true and the others false (where, on average, simply guessing would obtain 25% of the points), any of the four parts can be either true or false, and the exact combination has to be achieved in order to gain the points for that question. Sample test are available on the course web site [15]. Students must



(a) One of the usability kits available for students to borrow.

(b) The test setup used by one of the groups in summer semester 2017 [14].

Fig. 3: Usability kit and test setup. The image in (b) is used under the terms of a Creative Commons Attribution 4.0 International (CC BY 4.0) licence.

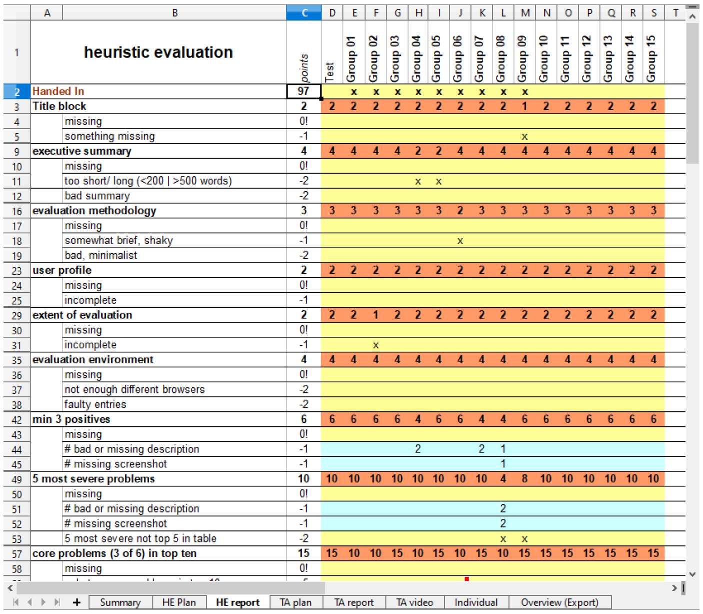
answer at least two of the ten questions correctly, in order to pass the course. After the multiple choice test has been marked, a grading review is scheduled, where students can optionally receive detailed face-to-face feedback and ask any questions they may have. Students then have one chance to retake the MC Test.

5 Grading Management with Sapphire

For the first 10 years of running the HCI course for undergraduates (2003–2013), grading was managed using a spreadsheet, one per tutor, as shown in Fig. 4. The grading system is based on individual ratings grouped into rating blocks. Each rating block is assigned an initial number of points, and deductions are made depending on individual rating criteria. Most of the ratings involve a binary decision as to whether or not the corresponding criterion applies. If so, an x is entered into the corresponding cell. This helps maintain consistency in grading across the tutors. Both fixed and percentage deductions are possible, and in some cases, a per-item deduction can be applied. It is also possible to define ratings which deduct a variable number of points or assign a variable number of bonus points, within a certain range, based on the assessment of the tutor.

Maintaining the spreadsheets and making adjustments to ratings or rating groups involved a significant amount of work and great care had to be taken not to damage the cell formulae. It was also impossible to provide detailed incremental feedback on a student-by-student basis as the term and grading progressed.

In 2014, a new web-based submission and grading management system called Sapphire [16] entered service. Sapphire was custom-built in Ruby/Rails and HTML/CSS/JavaScript [17, 18] and is available as an open-source project [19]. Sapphire reproduces the concept of ratings and rating groups in a shared online environment with user accounts, roles, and permissions. Students register for the



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	heuristic evaluation		points	Test	Group 01	Group 02	Group 03	Group 04	Group 05	Group 06	Group 07	Group 08	Group 09	Group 10	Group 11	Group 12	Group 13	Group 14	Group 15
2	Handed In		97		x	x	x	x	x	x	x	x	x						
3	Title block		2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2
4		missing	0!																
5		something missing	-1										x						
9	executive summary		4	4	4	4	4	2	2	4	4	4	4	4	4	4	4	4	4
10		missing	0!																
11		too short/ long (<200 | >500 words)	-2					x	x										
12		bad summary	-2																
16	evaluation methodology		3	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3	3
17		missing	0!																
18		somewhat brief, shaky	-1							x									
19		bad, minimalist	-2																
23	user profile		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
24		missing	0!																
25		incomplete	-1																
29	extent of evaluation		2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2
30		missing	0!																
31		incomplete	-1			x													
35	evaluation environment		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
36		missing	0!																
37		not enough different browsers	-2																
38		faulty entries	-2																
42	min 3 positives		6	6	6	6	6	4	6	6	4	4	6	6	6	6	6	6	6
43		missing	0!																
44		# bad or missing description	-1					2			2	1							
45		# missing screenshot	-1									1							
49	5 most severe problems		10	10	10	10	10	10	10	10	10	4	8	10	10	10	10	10	10
50		missing	0!																
51		# bad or missing description	-1									2							
52		# missing screenshot	-1									2							
53		5 most severe not top 5 in table	-2									x	x						
57	core problems (3 of 6) in top ten		15	15	10	10	15	10	15	10	15	10	15	15	15	15	15	15	15
58		missing	0!																

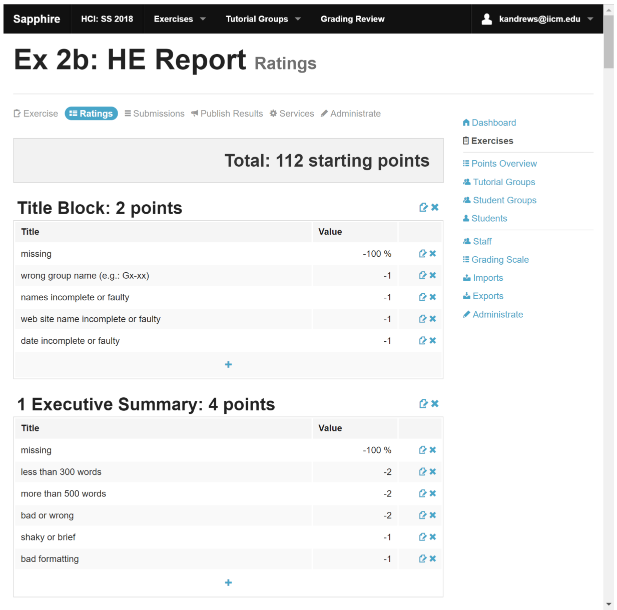
Fig. 4: Part of the spreadsheet used for grading for the first ten years of the Bachelor's course.

course on the university's campus management system, TUGRAZonline, and are then imported into Sapphire using a CSV file. In addition, a lecturer can create, edit, or delete students and student groups through Sapphire's user interface. Both individual and group exercises are supported. A submission component allows students and student groups to upload their exercise submissions online to Sapphire. Submission deadlines and late deadlines can be set and managed.

Sapphire's ratings editor, shown in Fig. 5, is used by the lecturer to configure points, ratings, and rating groups. The grading interface, shown in Fig. 6, is used by tutors to grade submissions. The interface is responsive and can comfortably be used on a tablet while simultaneously viewing a submission on a laptop. If a rating is changed after some grading has already taken place, tutors are notified of such changes in the interface.

Viewers can be configured for specific file types, such as PDF or HTML, and are opened within Sapphire as required. It is also possible to configure automated checkers for specific files, say for HTML validation or to run (part of) a file through an external plagiarism detection service. Finally, Sapphire provides export facilities for exporting both submissions (for archival purposes) and detailed grading reports (as spreadsheets).

One of the main benefits of Sapphire is that it is now possible to publish fine-grained intermediate results for each individual student online, as grading progresses during the term. In addition, Sapphire's commenting system allows a textual explanation to be attached to any applied rating, and written feedback



Ex 2b: HE Report Ratings

Total: 112 starting points

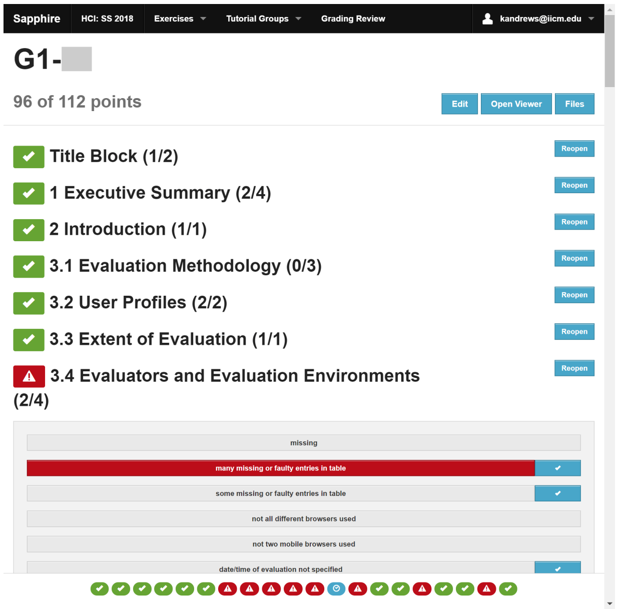
Title Block: 2 points

Title	Value
missing	-100 %
wrong group name (e.g.: Gx-xx)	-1
names incomplete or faulty	-1
web site name incomplete or faulty	-1
date incomplete or faulty	-1

1 Executive Summary: 4 points

Title	Value
missing	-100 %
less than 300 words	-2
more than 500 words	-2
bad or wrong	-2
shaky or brief	-1
bad formatting	-1

Fig. 5: Part of Sapphire's ratings editor, used by the lecturer to configure points, ratings, and rating groups.



G1-[redacted]

96 of 112 points

✓ Title Block (1/2)

✓ 1 Executive Summary (2/4)

✓ 2 Introduction (1/1)

✓ 3.1 Evaluation Methodology (0/3)

✓ 3.2 User Profiles (2/2)

✓ 3.3 Extent of Evaluation (1/1)

⚠ 3.4 Evaluators and Evaluation Environments (2/4)

missing

many missing or faulty entries in table ✓

some missing or faulty entries in table ✓

not all different browsers used

not two mobile browsers used

date/time of evaluation not specified ✓

Fig. 6: Part of Sapphire's grading interface, used by a tutor to grade a submission.

to be attached per exercise. Once grading of a particular exercise is finished, the lecturer can publish the provisional grading of all (or some) tutorial groups at the push of a button. Each student is notified by email when their own grading results become available, and only has access to these.

For the final grades, Sapphire's grading scale editor provides a graphical tool for the lecturer to set and review grade boundaries, as shown in Fig. 7. At least 50% of the points must be achieved for a pass, in this case 332 of the 663 points available. The grade boundaries can be adjusted up or down with the mouse.

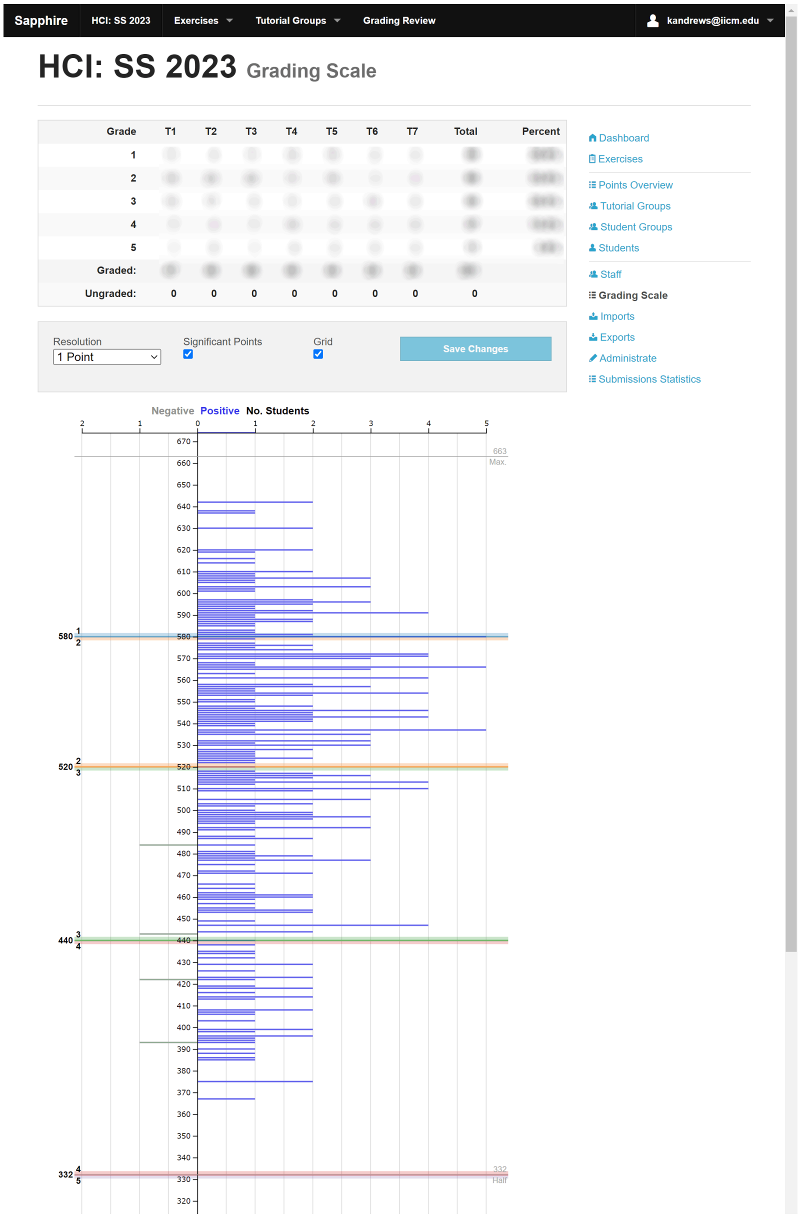
6 Adapting to COVID

With the rapid spread of COVID, the HCI course in summer semester 2020 had to be suspended after the first lecture. Initially, 74 students had signed up for the course (rather than the usual 300+), an artefact of the course having been moved from the 2nd to the 4th semester, resulting in a temporary one-year dip, as can be seen in Table 1.

Moving the course entirely online was challenging for a number of reasons:

- The course is built around students running face-to-face thinking aloud tests with external test users. COVID-19 restrictions made this impossible.
- We looked at switching to remote user testing, but that would require significantly more resources for supervision and grading, since each student would essentially be running their own remote test(s) and we would have to view every one of them, rather than selecting one of the test videos from each group to view.
- Replacing user testing with something else would mean completely redesigning a large part of the course.
- Furthermore, I show a large number of videos and video clips in class, which are integral to explaining and illustrating the course material and practical exercises. Some of the videos are from published or broadcast video material: these can be shown in class under an exemption in Austrian copyright law, but cannot be streamed or republished. Other video clips are from previous user tests, where I have permission from the test users to show (parts of) them in class, but do not have permission to stream or publish them.

In the end, since there was no end in sight to the pandemic, I had to design a special COVID version of the course for an (almost) entirely online environment. Lectures were moved online with Webex. I could only show those videos which were already publicly available on the internet. There were no thinking aloud tests. Instead, the heuristic evaluation was extended, with each student evaluating with two devices (desktop and mobile) rather than just one. The three client meetings with the tutors were reduced to two and were also moved online with Webex. Solely the multiple choice test was held face-to-face, under the university's special COVID arrangements (face coverings, social distancing, check of COVID green status, cleaning and airing between exam sittings, etc.).



HCI: SS 2023 Grading Scale

Grade	T1	T2	T3	T4	T5	T6	T7	Total	Percent
1	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
2	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
3	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
4	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
5	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
Graded:	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	
Ungraded:	0	0	0	0	0	0	0	0	

Resolution: 1 Point
Significant Points ☑
Grid ☑

Fig. 7: Part of Sapphire's grading scale editor, used by the lecturer to set and review grade boundaries. The grade boundaries can be adjusted up or down with the mouse. In the Austrian system, grades 1 to 4 are passes and grade 5 is a fail. The summary statistics have been deliberately blurred.

The COVID version of the course was taught in Sep 2020 as a block course, with 30 students eventually participating. The persistence of COVID made it necessary to teach the COVID version of the course in summer semester 2021 and summer semester 2022 too. In summer semester 2023, the course reverted back to its previous format with in-person lectures and meetings, and face-to-face thinking aloud testing. In 2023, there were 278 students in 71 groups with 7 tutors.

7 Concluding Remarks

Teaching HCI as a compulsory undergraduate course is a challenging undertaking when many hundreds of participants are involved. However, with careful planning, logistics, and technical support it can be managed successfully, even if a global pandemic throws a spanner in the works.

Acknowledgements. Many tutors, former students, and colleagues have contributed to the success of the HCI course at Graz University of Technology over the years. I would especially like to acknowledge the contributions of the following people. Ludwig Reinsperger and Peter Sammer taught the first HCI course at the university, before passing the flame on to me. Karl Voit built the first version of the grading spreadsheet. Patrick Hackl maintained and extended the spreadsheet over five years, followed by Vesna Krnjic, Sabrina Huber, Laura Bom, and Christoph Bichler. Thomas Kriechbaumer and Matthias Link built Sapphire over several years and a Bachelor's and Master's thesis. Stefan Pranger maintained the Sapphire server for many years, followed by Amel Smajic. It is currently maintained by Duaa Nakshbandi and Bakir Haljevac.

References

1. Shneiderman, B.: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley (1986)
2. Andrews, K.: *Human-Computer Interaction*. 2023. https://courses.isds.tugraz.at/hci/.
3. Andrews, K.: *Human-Computer Interaction: Course Notes*. May 23, 2023. https://courses.isds.tugraz.at/hci/hci.pdf.
4. Hewett, T.T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, G., Strong, G., Verplank, W.: ACM SIGCHI Curricula for Human-Computer Interaction. Technical Report, ACM (1992). https://doi.org/10.1145/2594128
5. Andrews, K.: *HCI Practical Exercises*. 2023. https://courses.isds.tugraz.at/hci/practicals/.
6. Andrews, K.: *Materials for HCI Practicals*. 2023. https://courses.isds.tugraz.at/hci/practicals/materials/.
7. Nielsen, J.: Heuristic Evaluation. In: Usability Inspection Methods. Ed. by J. Nielsen and R.L. Mack, pp. 25–62. Wiley (1994)
8. Nielsen, J., Molich, R.: Heuristic Evaluation of User Interfaces. In: Proc. Conference on Human Factors in Computing Systems (CHI'90), pp. 249–256, Seattle, Washington, USA (1990). https://doi.org/10.1145/97243.97281

9. Leski, F., Mandl, F., Weber, M., Ganster, P.: *Heuristic Evaluation Plan*. Mar. 23, 2017. https://cdn.isds.tugraz.at/hci/reports/ss2017/g3-02-tugraz-en/heplan/heplan.html.
10. Leski, F., Mandl, F., Weber, M., Ganster, P.: *Heuristic Evaluation Report*. Apr. 20, 2017. https://cdn.isds.tugraz.at/hci/reports/ss2017/g3-02-tugraz-en/he/he.html.
11. Krug, S.: Rocket Surgery Made Easy. New Riders (2009)
12. Nielsen, J., Clemmensen, T., Yssing, C.: Getting Access to What Goes On In People's Heads?: Reflections on the Think-Aloud Technique. In: Proc. 2nd Nordic Conference on Human-Computer Interaction, pp. 101–110. ACM Press, Aarhus, Denmark (2002). https://doi.org/10.1145/572020.572033
13. Leski, F., Mandl, F., Weber, M., Ganster, P.: *Thinking Aloud Test Plan*. Apr. 24, 2017. https://cdn.isds.tugraz.at/hci/reports/ss2017/g3-02-tugraz-en/taplan/taplan.html.
14. Leski, F., Mandl, F., Weber, M., Ganster, P.: *Thinking Aloud Test Report*. May 25, 2017. https://cdn.isds.tugraz.at/hci/reports/ss2017/g3-02-tugraz-en/ta/ta.html.
15. Andrews, K.: *HCI Sample Tests*. 2023. https://courses.isds.tugraz.at/hci/practicals/test/.
16. Sapphire, *Sapphire Course Grading Management System*. 2023. https://sapphire.isds.tugraz.at/.
17. Kriechbaumer, T.: *Sapphire Back-End: A Web-Based Course Grading Management System*. Bachelor's Thesis, Graz University of Technology, Austria. Feb. 28, 2014. https://ftp.isds.tugraz.at/pub/theses/kriechbaumer-2014-bsc.pdf.
18. Link, M.: *Sapphire Frontend: A Web-Based Course Grading Management System*. Master's Thesis, Graz University of Technology, Austria. Oct. 27, 2021. 194 pp. https://ftp.isds.tugraz.at/pub/theses/mlink-2021-msc.pdf.
19. Link, M., Kriechbaumer, T.: Sapphire – Course Management System, Graz University of Technology. (2023). https://github.com/Sapphire-CM/Sapphire/