e²Logos: A Novel Software for Evaluating Online Student Project Reports

Results from a comparative usability study with undergraduate teaching assistants

Panagiotis Apostolellis
Department of Computer Science
University of Virginia
Charlottesville, VA, USA
panaga@virginia.edu

Philip Hart
Department of Computer Science
University of Virginia
Charlottesville, VA, USA
ph9aa@virginia.edu

Ketian Tu
Department of Computer Science
University of Virginia
Charlottesville, VA, USA
kt9sh@virginia.edu

Abstract—**Most assignments in engineering design courses include open-ended, real-word projects, where groups of students work together and produce a collection of deliverables, which need to be reported and evaluated by the instructor so that the students can act upon the feedback. The complexity of the work often demands that project artifacts be reported online, but there is no software designed to assess and grade web-based technical reports. This paper introduces e²Logos, a novel custom grading/feedback tool designed for evaluating online reports and presents results from a comparative usability study with Gradescope, a popular grading tool for PDF submissions. Findings from grading two project phases in two semesters for a computer science Human-Computer Interaction (HCI) course, revealed that e²Logos was perceived as more efficient, motivating, dependable, and attractive than Gradescope, by using the User Experience Questionnaire (UEQ) and our own usability goals. However, it was not shown to improve grading consistency among graders. Implications for designing similar software are presented as design requirements, along with our plans to evaluate e²Logos for its effectiveness in improving learning outcomes in Project-Based Learning (PBL) courses.**

Keywords—usability test; grading software; web annotation tools; project-based learning; student feedback.

I. INTRODUCTION

Within the field of engineering, managing and contributing to complex projects are essential parts of the learning process [1]. A common method to achieve this in many upper-level engineering courses is to use open-ended, client-driven, team-based problems, most commonly known as Model-Eliciting Activities (MEAs) [2]. MEAs are a form of PBL that have been applied increasingly to engineering courses over the past decade, offering a form of assignment intended to emulate the design process of a derived solution for real-world problems within a limited time [3]. Throughout these activities, students are encouraged to integrate and apply knowledge from past and current courses toward producing a cohesive solution for an open-ended problem.

Tackling a complex, open-ended design problem can be challenging for engineering students, particularly with a lack of experience in multi-disciplinary skills like self-reliance, collaboration, and time management [4]. Furthermore, there is often an inconsistency between the instructor's and students' learning objectives within the context of semester-long, real-world projects [5]. Thus, engaging students with MEAs is difficult due to the requirements for consistent and accessible feedback throughout project development. To apply MEAs to a large class, the grading and feedback process must be expedited yet remain simplistic for graders [4].

Many methodologies are used to grade MEAs and other project-based deliverables, including self-assessment, peer assessment, co-assessment, and performance assessment, all of which involve a way of evaluating work whether it be from the students themselves, their peers, or staff [6]. Another method is specifications grading, which introduces a pass/fail system toward assignments, focusing more on certain learning objectives being met to earn a certain grade [7]. Adaptive rubrics have also been used and even integrated within modern grading software. This approach has been shown to help graders focus on deeply understanding student work before deciding on point deductions through tailoring the rubric based on student submissions [8].

Many tools are available for the grading of online assignments, such as various Learning Management Systems (LMS) or dedicated grading tools like Gradescope. Gradescope is geared toward the grading of handwritten work and quiz style questions [9], while LMSs may provide administration of course content, scheduling, announcements, assignments, quizzes, and other functionalities in addition to grading student work [10]. These systems are well equipped to grade work that can be uploaded, such as images or PDFs [9], but not for student submissions in an online (website) format. Moreover, student work often entails different problem-solving patterns when tackling open-ended design problems [3], meaning outcomes may vary. Thus, grading needs to be specific and tailored to the project. To our knowledge, there is no grading tool that provides customized grade deductions, collaborative grading, and support for within-context commenting of web-based technical reports, a widely used format in PBL courses.

The novel tool e²Logos, evaluating electronic logos (from the Greek work *λόγος*, for speech), aims to address challenges faced in courses where online technical and reflective reports are a significant part of the evaluation process. e²Logos combines assessment, annotation, feedback, and dialogue features and is built on the open-source Hypothes.is platform [11]. The primary focus of this grading tool is to provide timely and consistent feedback to students and assess PBL outcomes. The tool has a client-side interface for instructors and graders to grade students' work, and for students to access their grades and individual feedback. There is also a backend management

 ISBN: 978-1-68558-078-0

portal for instructors to review and release grades for projects. e²Logos has been deployed and tested in an undergraduate engineering course at a large public U.S. university during fall 2022, collecting 1444 comments from 6 graders in the class.

This paper aims to provide context for the design of e²Logos and examine its usability to respond to the question: *How does e²Logos compare to an established grading software (Gradescope), in terms of efficiency and ease of use for grading students' online project work?*

The paper begins with a review of relevant grading and annotation tools (Section II), then presents the design of e²Logos (Section III), continues with the description of the usability studies (Section IV) and the results of our analysis (Section V), finishing with a discussion of main findings (Section VI), and the conclusions and future work (Section VII).

II. BACKGROUND

A. Importance of feedback in PBL student work

While engaging with PBL work, students must synthesize past knowledge with new skills learned from their current course. However, design courses in higher education require multidisciplinary skills and not all students will perform similarly, especially when first introduced to PBL, but they can improve in the proper learning environment [12]. Feedback should aim to reduce the gap between current understanding and the desired goal [13], hopefully, setting a personalized measure of current collective achievement and indicating how to improve upon that achievement.

Good feedback should be timely and efficient [13], and using grading tools streamlines the grading process. However, the challenge for instructors is in being accurate and consistent in grading open-ended projects that have more than one correct solution [14]. There are currently many software tools that support grading but may not be best suited for grading while providing feedback essential to guiding PBL work.

B. Computer-based assessment tools for PBL work

There are many educational technologies used by higher education institutions to evaluate student learning; some are more suited for only grading, while others offer a way to display a grader's feedback. In this review, we will mainly focus on computer-based tools for grading and evaluating digital student submissions.

Grading/Feedback Tools. Gradescope is a platform that allows instructors to assess handwritten assignments and exams online, with such features as automated grading, peer review, and customizable grading rubrics [9]. It has been shown to save instructors time and improve the consistency and fairness of grading. In a study of two undergraduate mechanical engineering courses, Gradescope reduced the instructor's grading time by approximately 2.5 hours, while both the rubric structure and the ease of switching between submissions helped ensure that grades were consistent for all students [15]. It has also been used to gather real-time feedback from students, as demonstrated in an introductory data science class where instructors used Gradescope's tagging system to track student learning objectives and adjust their curriculum based on the feedback received [16].

A LMS is a type of software that helps educators administer, document, track, report, automate, and deliver educational courses [10]. LMSs have become increasingly popular, especially due to the transition to online learning during the COVID-19 pandemic [17]. In a recent study, the use of Moodle [18], a popular LMS, was evaluated as an e-learning platform in a project-based undergraduate course [19]. Students worked in groups of 3 to 5 on an open-ended project throughout the semester. A survey administered at the end of the semester revealed that 10% of students cited the feedback mechanism as their favorite aspect of using Moodle, while 15% reported that the tool made it difficult to locate work and had too many confusing links on the page. Overall, the use of Moodle as a LMS in a PBL course was seen as a useful tool for instructors to provide feedback to students, but there were challenges in terms of navigation and organization. The evaluation of another popular LMS, Blackboard [20], in terms of its usefulness in an undergraduate computer literacy course, revealed that immediate feedback on online quizzes was the most helpful aspect of Blackboard, while collaborative work and communication with peers and instructors were rated as the least effective aspects [21].

iRubric is a web-based rubric development, assessment, and sharing software, commonly integrated with LMS platforms to facilitate matrix-style grading [22]. During evaluation of student work, a grader must select a pre-defined rubric criterion and then write specific feedback in a table format. A study evaluating iRubric found that it streamlined the grading process by promoting a consistent grading element throughout the university-wide adoption of the tool, as a replacement of the previously used paper rubrics [23]. Most contemporary LMSs provide embedded grading functionality using rubrics, where instructors can define grading criteria and associate them with specific learning outcomes. Canvas, as an example, has been shown to be very effective for assessing student learning using its rubric tool by gauging the students' level of achievement in some disciplinary area [24]. Such tools are limited for assessing online work as grading is disassociated from the work and graders must switch multiple times between web submission and the rubric hosted on the LMS, plus the assigned criteria are fixed to evaluating broader outcomes/expectations. Others have worked on developing rubrics for STEM courses to facilitate goal setting and self-evaluation [25], but such tools have not made it into an interactive software, nor have they been tested for their usability.

Annotation Tools. Hypothes.is is an open-source software platform that allows users to annotate web pages and PDF files with highlights and comments [11]. In a study conducted in an undergraduate engineering course, students who used Hypothes.is to annotate and discuss articles in a group performed better than those who did not in the final exam. While there was no quantitative data available on the instructor's perspective, the researcher noted that Hypothes.is promotes communication and peer review, which are essential factors for effective PBL [26]. In another study, the use of Hypothes.is in combination with a Google Doc was found to be effective for annotating articles and summarizing points made by groups of students [27]. Hypothes.is can also be integrated with a LMS to provide added grading functionality. However, this

 ISBN: 978-1-68558-078-0

integration only allows students to annotate an instructor-selected online resource and provide a single score based on the quality of student annotations.

Diigo is a social bookmarking tool that allows users to add digital sticky notes to web pages [28]. It is frequently used in educational settings due to its ability to annotate and organize data. A case study conducted in a technology course introduced Diigo to pre-service teachers through multiple lectures and gathered feedback from both students and instructors. The majority of students had a favorable impression of the tool, and instructors reported that students engaged more deeply with course concepts through searching and annotating course content. However, some participants expressed concerns that Diigo offered too many features for a bookmarking tool [29]. Another tool for sharing digital content, Digication [30], allows students to submit a snapshot of their website reports through LMS integration, but commenting can only happen on the live website and lacks grading functionality.

An empirical study of EDUCOSM, a set of tools for asynchronous collaborative knowledge construction, in a statistics course determined that digital systems equipped with annotation technology improved a student's affinity for learning on a collaborative document through student markings [31]. A more recent study examined the efficacy of digital annotations for feedback in comparison to other modes of delivering feedback to students, and found that a single mode of feedback, electronic annotations or digital recordings, were better for offering detailed and personalized feedback [32]. Another study evaluating a custom web-based tool for providing corrective feedback to English essays via annotations, showed that the gap between high-level and low-level student performance was eliminated through the application of corrective feedback [33]. Overall, examining annotation technologies has shown to benefit a grader when generating feedback for students [34]. However, such technologies are largely focused on providing students with feedback while analyzing digital submissions and are deprived of any grading functionality.

A focus of this review has been to examine the current functionalities and usability of grading and feedback tools in order to determine what types of tools are most efficient for evaluating online student work. However, none of the tools reviewed have been directly evaluated for their usability or compared with each other. In 2021, Gradescope introduced (as a beta version) a new format for grading essay-type assignments that provided combined grading and annotation functionality for digital submissions. However, submissions were restricted to a PDF format and the lack of a collaborative grading made it difficult to resolve grading inconsistencies between graders. Overall, the tools available for providing effective, personalized, and collaborative feedback, such as annotation software, lack course management and grading functionalities. Conversely, tools that enable course management and grading lack a way of providing personalized and specifically marked feedback in a collaborative manner. A tool that automates collaborative grading, while allowing a grader's in-context feedback to be as specific as possible is hypothesized to expedite and offer consistency to the grading process for online, open-ended, group design projects.

III. E^{2}LOGOS DESIGN

A. Extracting requirements for good feedback of PBL work

In order to reach the point of developing e^{2}Logos, the lead author tested combinations of different tools for assessing the design work of student groups in two HCI courses. Over multiple semesters, different tools included the use of Google Spreadsheets for grading and feedback as notes (exported and released to students as PDF files); a combination of Diigo or Hypothes.is (used for within-context feedback) with Google Spreadsheets (for grading); and Gradescope's *Essay* assignment format (released as beta for the 2021-22 academic year only). None of these approaches proved effective in accommodating the unique demands of leaving good graded feedback within the context of rich online technical reports that students generated while reporting their project work.

The outcome of these iterations was a compiled list of design requirements that could satisfy the identified demands. This list was derived from personal experience, conversations with colleagues teaching similar courses, and feedback from teaching assistants who helped grade project design work.

1) Within-context feedback and grading: A crucial learning factor for PBL work is for students to review and understand the provided feedback within the context of their own work. Prior approaches combining tools, such as [27] and our own experimentation, decoupled feedback and grading from the students' own work, making it hard for them to understand how and where their work could be improved.

2) Personalized adjustment of score and feedback: A unique aspect of assessing project work is that fixed rubrics fail to capture adequately the element of quality. Thus, it is imperative that a grader has the flexibility to adjust the score and adapt the feedback provided to specific submissions for the same identified rubric item. In comparison, attempting to do so in Gradescope will change the score and associated comment to all submissions the item has been applied to.

3) Collaborative grading: Due to the open-ended nature of design projects it is rather challenging to achieve inter-grader consistency. Communication between graders is, therefore, necessary, allowing the instructor and more experienced staff to provide guidance to all graders and decrease grading inconsistencies among student submissions.

4) General feedback and regrade requests: Considering the difficulty of evaluating design work objectively, it is necessary to provide overall guidance to students after all grading is done. Additionally, student groups should be able to dispute the way their work has been assessed by requesting a regrade and providing their rationale.

The list of design requirements included above guided the design and implementation of e^{2}Logos, which is described briefly in the next section. Some of these requirements were also tested during the usability studies and findings in support of them are presented under the Discussion section.

 ISBN: 978-1-68558-078-0

B. *Technical Design of e²Logos*

Before discussing the technical aspects of e²Logos, it is crucial to note that the tool is built using the open-source Hypothes.is software [11], which allows users to annotate and converse on websites across the internet. We repurposed Hypothes.is to include a grading component but maintained all functionality for providing feedback on an online project report, including highlighting text within the page, adding and replying to comments, navigating to highlighted text when selecting a comment, all while storing this type of information in a PostgreSQL database. e²Logos, similar to Hypothes.is, is mainly a web-based client administered as a Chrome extension that communicates with a backend server through API calls. The server application is developed using various Pylons Project packages, such as the pyramid web framework, colander, and deform, as well as Elasticsearch for annotation lookup. The client (Chrome extension) is based on React for user interface and logic, and Redux for session management.

The backend website handles all administrative work and allows instructors to create courses, groups, and assignments. Assignments have an associated rubric, which is currently uploaded as a json file, but in future versions would be created using a dedicated rubric creation tool similar to Gradescope. Through the website, an instructor can assign students to groups, assign teaching assistants as graders (including a lead grader role with elevated privileges), and release grades to students. Graded comments are listed in buckets according to what assignment and group those annotations belong to, including the total score for each group. Searching by group name, grader username, or assignment name filters these buckets to only show the relevant comments. Since the backend website has not been included in the usability test, we will not elaborate further on its functionality.

The main operation of e²Logos is accomplished through the extension, where graders can evaluate a project report on the online submission itself (website or PDF). After navigating to the report's URL and selecting the desired assignment to be graded from the drop-down menu of the Rubric tab (Figure 1a), graders follow this workflow: a) they highlight the relevant text on the page and select "Grade" from a pop-up menu, b) they select the appropriate rubric item from the list and the corresponding pre-defined comment appears in the Comments tab (Figure 1b), which c) they can then edit in terms of text and/or score to fit the submission's quality, and d) they click "Post" to submit the item to the backend. This simple workflow satisfies the two first design requirements listed above. As a safety measure, graders cannot apply/deduct more points for an item beyond the thresholds defined by the instructor. Graders also have the option to only comment or highlight text without applying a rubric item, to simply provide feedback to students.

Graders can navigate between group websites by selecting the group's name from the drop-down menu at the top. If the website has been graded already, comments are fetched from the backend and displayed through highlights on relevant text within the page and text-based feedback in the Comments tab. Commenting includes a rich-text editor that can be used to emphasize specific parts or even embed an image.

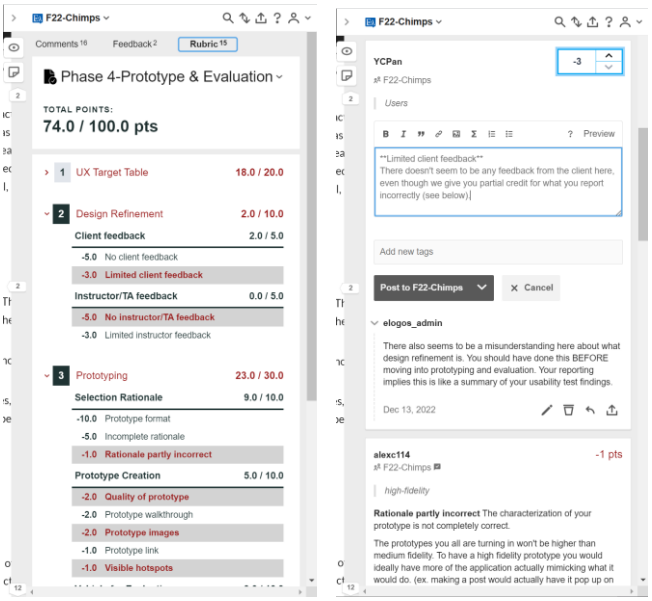

Figure 1. The Rubric tab (a-left) for grading online reports, along with the augmented Comments tab (b-right) for editing/adjusting graded comments.

To augment the grading and feedback process even further, lead graders and instructors can either edit existing comments and scores or add further explanations to justify an applied rubric item (as is the case with the added comment by *elogos_admin* in Figure 1b). To facilitate design requirement #3, we included a "Graders only" option for replying to existing comments, which allows inter-grader communication about project assessment hidden from student view (see Figure 2). This feature, in addition to having multiple graders work on the same project report—the tool recognizes changes and allows a grader to update the highlights, comments, and rubric deductions with any changes made by someone else—makes e²Logos a much more collaborative grading tool than other currently available software.

Finally, the tool allows graders to leave general feedback through the Feedback tab, which may include comments on the overall quality of the submission or advice for future work. The same tab can be used for student regrade requests, where students can question some parts of the assessment or provide extra rationale for their design choices that the grader(s) might have missed, contributing to a more equitable grading process. This is similar to functionality offered by Gradescope and satisfies our last design requirement from the list above. Future versions of e²Logos will allow an instructor to choose if students could reply to existing comments for regrade purposes, offering their rationale on a comment-by-comment basis.

It is also worth mentioning that in an effort to increase grading consistency, a common challenge for evaluating PBL and especially design work, e²Logos allows for multiple grader-groups to evaluate the same sample submission. In this scenario, every grader has their own group which they use to assess a sample project report. The instructor is able, then, to switch between grader-groups reviewing the applied rubric items (gray items in Figure 1a), commenting and providing extra guidance to graders during a grading practice session.

 ISBN: 978-1-68558-078-0

IV. METHODS

A. Usability Study Context

In order to test the efficacy of the tool from the graders' perspective, we conducted a usability test of e²Logos in fall 2022 (F22). Since we have been using Gradescope's Essay submission feature for the 2021-2022 academic year, we also tested the usability of Gradescope for grading technical reports in spring 2022 (S22). Gradescope discontinued the use of this type of submission since summer 2022, so we were unable to collect data beyond that point. Both grading tools were tested on an upper-level engineering course on HCI, usually taken by third- and fourth-year undergraduate students at a large public U.S. university. The course employed a PBL approach, where student work is broken down into four project phases throughout the whole semester and is submitted as a technical report on a website. The reports typically include a variety of static text and dynamic content, like image carousels, embedded Google slide presentations, or links to external applications (e.g., YouTube videos and Figma prototypes).

Since Gradescope's Essay assignment type required uploading of student work as a PDF file, student groups in S22 were instructed to export their website to a PDF file. The grading rubric was created by the instructor of the course in Gradescope's dedicated tool or a json file for each one of the semesters, respectively. The rubrics were broken down in categories (e.g., *Prototyping*) and sub-categories (e.g., *Rationale*), even though Gradescope did not support the extra level like e²Logos did (see Figure 1). The teaching assistants (TAs) of the course were then tasked to use each grading software to grade the last two phases of the project in each semester. Grading of the first two phases was used as practice, so TAs could familiarize themselves with the tools' features and their application. The grading process involves three passes (TAs, project lead TA, instructor), where each user grades and leaves feedback on student work, as well as suggestions for improvement. When grading is completed—usually within a week—submissions are returned and scores/feedback are reviewed by students, either on Gradescope (S22) or the website itself with the associated e²Logos Chrome extension installed (F22). Since the tool was under development during the same time, graders were asked to install (unpack) the extension on their browser instead of downloading it from the Chrome store. As part of this usability test, we did not evaluate the effectiveness of the feedback provided with the tools, in terms of student learning.

The usability study was approved by the Institutional Review Board of the University of Virginia with protocol IRB-SBS#5515/2022.

B. Measuring Instruments

Assessing the efficacy of the two grading tools involved a two-prong strategy. The TAs and instructor first created a UX target table, inspired by a typical usability engineering process [35], to measure specific UX goals related to the project report assessment (Table I). Our decision about goals was led by the two key outcomes included in our research question: efficiency and ease of use.

TABLE I. UX TARGET TABLE FOR EVALUATING THE GRADING TOOLS

Goal	Measure	Instrument	Metric
Efficiency	User performance	BT1: Finish project grading	Avg. time on task
Efficiency	Critical incidents (limitations)	BT1: Finish project grading	Avg. # of instances impeding grading
Efficiency	User performance	BT2[a]: Apply a predefined deduction	Avg. time on task
Accuracy	Experienced usage error	BT2[a]: Apply a predefined deduction	Avg. # of errors
Ease of use	Experienced usage error	BT3: Leave a comment to a deduction	Avg. # of errors
Ease of use	Experienced usage error	BT4: Remove rubric deduction	Avg. # of errors
Efficiency	User performance	BT5: Write general feedback	Avg. time on task
Ease of use	User performance (communication)	BT6: Communicate grading issues/questions	Avg. # of times a comment was left for lead TA or instructor
Efficiency	User performance (consistency)	BT7[b]: Reviewing an existing graded project	Avg. time on task
Effectiveness	User performance (consistency)	BT7[b]: Reviewing an existing graded project	Avg. # of changes to existing grading
Effectiveness	User performance (consistency)	BT7[b]: Reviewing an existing graded project	Avg. # of critical incidents

a. This measurement involved selecting a rubric item from the lower sections, like a Bonus.

b. Data for BT7 were measured both during the lead TA's and the instructor's grading review.

Benchmark tasks (BTs) 1-6 referred to TAs as graders, while BT7 referred to the lead TA and instructor as grader-reviewers. We then used a questionnaire for measuring the perceived user experience (UX) by the TAs. We chose the User Experience Questionnaire (UEQ) [36] over the System Usability Scale (SUS) [37] or similar instruments, because the former covers a broader range of subjective measures related to using interactive software. More specifically, the UEQ gathers insights about an application's perceived usability in terms of six factors: *attractiveness*, *perspicuity* (commonly known as learnability), *efficiency*, *dependability* (also known as user control and freedom), *stimulation*, and *novelty*.

C. Participants

Nine undergraduate college students were recruited for the usability studies over the two semesters. In each semester, the five students that served as TAs for the HCI in Software Development course, part of the Computer Science department curriculum, were the grader-participants that helped evaluate the two grading applications. Only one of them was a returning TA in F22, who also served as the lead TA in that semester. No demographic data was recorded about the participants, as they were deemed irrelevant to the outcomes of the study. Since participants were mainly conducting their regular TA duties, no compensation was given for their participation. The instructor of the course remained the same in both terms.

 ISBN: 978-1-68558-078-0

D. Procedure

The procedure followed during each one of the semesters was exactly the same, with the grading tool being the only difference. The TAs would start grading the project submissions either on Gradescope (S22) or the website itself with the e²Logos extension (F22), using the same pre-defined rubric. Since grading of the earlier phases was not recorded for testing purposes, TAs had the opportunity to learn using the software. While conducting the student project report grading for phases 3-4, TA participants were asked to log the different data requested in the UX target table (see *Metric* in TABLE I.). This was done on a separate spreadsheet created specifically for this case in each semester. When all grading was done, the lead TA would take over to review submitted grades and TA feedback/comments. During this process, the lead TA would often need to coordinate with the graders to resolve any concerns with grading. While e²Logos provided the option to collaborate through replies to graded comments hidden from students (Figure 2), grading review on Gradescope was done offline through platforms and tools like email, GroupMe, or Discord. After the second pass, the instructor reviewed the graded submissions and made any final adjustments to grading/feedback. Similarly, communication with TA-graders was done either outside the tool (S22) or within the tool (F22), for resolving grading inconsistencies. Both the lead TA and the instructor logged their data (i.e., BT7 in TABLE I.), before recording their updated project submission score.

Right after the last project phase was graded at the end of the semester, the TAs would complete the UEQ—created and administered on Qualtrics—to capture their overall experience using each software. No discussion would precede this evaluation to avoid influencing the participants' opinion. Two open-ended fields were added to the UEQ to record the positive aspects and points of improvement for each software.

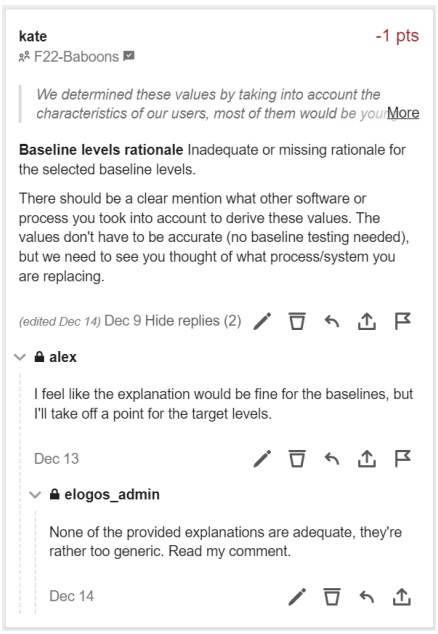

Figure 2. Within-tool communication between TA grader (*kate*), lead TA (*alex*), and instructor (*elogos_admin*), which is hidden from students (the lock icon indicates "Graders only" replies).

V. Results

A. Analysis of UX Goals

The analyzed e²Logos data derived from a total of 506 rubric items and comments applied/left by the five TAs for the two graded project phases included in the study (86 items/comments deleted), and 111 items/comments left by the instructor (6 of them deleted). For Gradescope, there were 419 final comments left by the five TAs and the instructor (there is no grader designation stored during grading and no record of deleted items/comments in that software). Per our initial UX evaluation plan (Table I), we consolidated all data from both semesters in a spreadsheet, identifying any missing data points and outliers. The only outliers removed—more than two standard deviations from the mean—were two extreme times recorded by one TA for BT5 in S22-Phase 3. Table II summarizes the final values for each benchmark task per semester and project phase, as well as the total average values. Even though we broke down BT7 data (logged during grading review) to comments and rubric items edited/added/removed, we decided to aggregate them in one value (#changes) per our original BT7 metric. Gradescope was used to assess/grade a total of 38 student reports in PDF format (19 project groups in S22), while e²Logos was used to assess/grade 20 online student reports (10 project groups in F22). Statistical analysis included the comparison of the Total calculated values (bold).

TABLE II. Logged UX Goal Data for Graded Project Phases

		e²Logos [N=20 [a]]			Gradescope [N=38 [a]]		
	Metric	***Ph-3***	***Ph-4***	***Total***	***Ph-3***	***Ph-4***	***Total***
BT1	mins	62.00	61.20	**61.60**	60.53	71.32	**65.92**
BT1	#incidents	0.30	0.10	**0.20**	2.32	**1.95	**2.13**
BT1 [b]	*#items*	*10.70*	*10.10*	***10.40***	*8.68*	**6.32*	***7.50***
BT1 [b]	*#comment*	*5.40*	*4.67*	***5.03***	*6.16*	*5.26*	***5.71***
BT2	secs	3.50	1.70	**2.60**	8.58	**6.16	**7.37**
BT2	#errors	0.00	0.00	**0.00**	0.37	*0.37	**0.37**
BT3	#errors	0.00	0.00	**0.00**	0.68	**0.58	**0.63**
BT4	#errors	0.90	0.00	**0.45**	1.11	0.37	**0.74**
BT5	secs	135.00	175.40	**157.44**	194.00	155.47	**170.88**
BT6	#contacts	0.70	0.70	**0.70**	0.37	0.53	**0.45**
BT7 [c]	mins	35.88	23.38	**29.63**	28.14	[e]	**28.14**
BT7 [c]	#changes	5.25	6.25	**5.75**	4.45	[e]	**4.45**
BT7 [c]	#incidents	0.00	0.50	**0.25**	0.44	[e]	**0.44**
BT7$_i$ [c]	mins	28.30	20.60	**24.45**	27.11	14.71	**20.91**
BT7$_i$ [c]	#changes	6.70	3.50	**5.10**	6.21	3.26	**4.74**
BT7$_i$ [c]	#incidents	0.10	0.00	**0.05**	1.26	**0.47	**0.87**
BT7 [b]	*$\Delta 1_{score}$* [d]	*3.00*	*4.40*	***3.70***	*4.05*	*3.11*	***3.58***
BT7 [b]	*$\Delta 2_{score}$* [d]	*4.50*	*3.13*	***3.81***	*4.42*	[e]	***4.42***

a. Sample size denotes the number of total observations; some metrics had missing data.

b. Data in italics were not included in the original UX goals but were analyzed for context.

c. The first BT7 metrics are from the lead TA's review and the last from the instructor's (i) review.

d. Difference between instructor and TAs (Δ1), and instructor and lead TA (Δ2) project scores.

e. The lead TA did not complete their review for project phase 4, therefore no values are included.

* Gray BTs were significant at the <0.05 level (*) or highly significant at the <0.001 level (**).

 ISBN: 978-1-68558-078-0

We used an independent-samples (unequal variance assumed), two-tailed Student's t-test to examine any statistical difference between the measured UX targets for the two grading tools. The null hypothesis was that there will be no difference between the two grading tools in terms of measured outcomes and sample data drawn from the observations were partly normally distributed. For non-normally distributed data, a non-parametric test's results are reported, using Mann-Whitney U test [38]. Missing values were handled by excluding cases on an analysis-by-analysis basis and only significant findings are reported below (indicated with gray in Table II).

Our analysis found that using e²Logos presented TA graders with a statistically significantly lower number of critical incidents ($U = 112, n = 58, p < 0.001$) as compared to Gradescope, while they applied a higher number of rubric items for the two project phases ($t = -6.36, n = 58, p = 0.041$). While applying an item lower in the rubric, TA graders were significantly slower ($U = 45.50, n=58, p < 001$) and did more errors ($U = 280, n = 58, p = 0.013$) than when completing the same task with e²Logos. TAs also did a significantly higher number of errors when completing the most common task of leaving a comment using the rubric in Gradescope than using e²Logos ($U = 240, n = 58, p = 0.002$). Finally, during the instructor's review of the graded submissions, the instructor reported significantly more critical incidents when using Gradescope than when using e²Logos ($U = 204.50, n = 58, p = 0.001$). No other UX target was found to reject the null hypothesis.

B. UEQ Comparison Analysis

We used an independent-samples (equal variance), two-tailed Student's t-test to examine if the two grading tools performed equally well in terms of the six UX factors recorded by the UEQ. The results of the t-tests with significance values are summarized in Table III and depicted in Figure 3. Results indicate that e²Logos outperformed Gradescope—rejecting the null hypothesis—in *attractiveness (d = 2.44), efficiency (d = 2.20), dependability (d = 2.32)*, and *stimulation (d = 1.68)*, while there was no statistical difference in *perspicuity (d = 1.29)* and *novelty (d = 1.84)*. A Cronbach's alpha test indicated that all six scales were reliable above a threshold of $\alpha = 0.711$ for both samples (tools), with average $\alpha = 0.789$. For assessing e²Logos the average was $\alpha = 0.7287$, while Gradescope's assessment using the UEQ yielded an $\alpha = 0.662$. All results indicate an acceptable to good internal consistency considering the small sample of the study [39].

TABLE III. T-TEST COMPARISON STATISTICS FOR UEQ SCALES

	e²Logos		Gradescope		Statistics	
Scale	***Mean***	***STD***	***Mean***	***STD***	***t stat***	***p value***
Attractiveness	1.30	0.88	-0.83	0.87	3.852	0.005*
Perspicuity	1.35	0.84	0.20	0.89	2.100	0.069
Efficiency	1.55	0.74	-1.05	1.18	4.183	0.003*
Dependability	0.65	0.86	-0.80	1.10	2.329	0.048*
Stimulation	1.00	0.64	-0.35	0.80	2.946	0.019*
Novelty	0.85	1.04	-0.35	0.65	2.186	0.060

* Indicates statistical significance at the 0.05 level (95% confidence intervals).

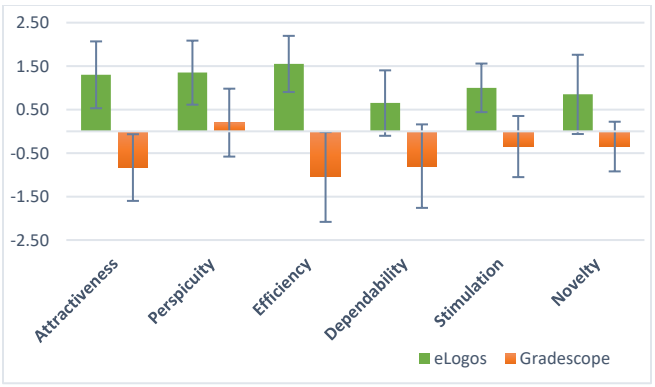

Figure 3. UEQ comparison between e²Logos and Gradescope.

C. Qualitative Findings

Analysis of participant input through the UEQ open-ended questions involved reviewing and grouping responses in relevant themes. We considered using thematic analysis, but the pool of responses was too limited for this technique to yield any significant benefit. Overall, e²Logos was perceived very positively by the TAs, mostly commenting about the constant availability of a rubric for easy reference, being able to highlight and indicate within the web page exactly what an item (deduction) refers to, and the flexibility offered by being able to adjust points and comments, customizing the score and feedback to each project. On the downside, there were a couple of complaints about the screen space that the sidebar occupied, obstructing part of the text while reading/grading. Another complaint was about a URL identifier (authuser) that was added to the websites by Chrome based on the active Google account and as a result comments/highlights were not displayed. This demanded manually deleting the identifier text from the browser's address bar. Finally, the issue of having to download and install the extension manually due to it not being an official extension in the Chrome store, was commented by one TA.

For Gradescope, positive feedback included the ease of reviewing the rubric using the sidebar (very similar to e²Logos), while also being able to search for a rubric item using the search box embedded in a drop-down menu that appeared after highlighting text. The workflow was found to be fairly intuitive, while TAs who have been involved with the course in prior semesters commended the significant improvement over using a grading spreadsheet. Downsides mainly had to do with the restrictions imposed by the non-searchable PDF format, which often included non-selectable text (depending on the website export process used by the students submitting the report). The PDF format made loading each submission rather slow (reports were often more than 30 pages) and prevented inclusion of dynamic content, demanding graders to visit the actual website to check and evaluate that material. The long drop-down menu with all deductions was found cumbersome to navigate, while not being able to adjust grading based on each submission's quality was noted multiple times as restrictive. Finally, the lack of collaboration while grading—comments left by another grader would not update automatically—was commented by two of the participants.

 ISBN: 978-1-68558-078-0

VI. Discussion

Overall, the findings from our analysis comparing the newly developed tool with an established grading software for evaluating digital online reports was very positive. However, the lack of similar usability studies on grading and annotation software does not allow us to compare our findings with prior work. Therefore, we will focus on discussing our comparison results as an effort to extract design implications for developing similar interactive grading software, also acknowledging the limitations of the current work.

A. UX Outcomes and Design Implications

Correlating the results from the UEQ and our own UX targets (Table I), it is obvious that e²Logos satisfied the most significant goal of efficiency as compared to Gradescope's Essay assignment type. The most critical task for grading software of applying a predefined deduction from the rubric (BT2) took significantly less time on average, $M_{elogos} = 2.60s$ *vs* $M_{Gradescope} = 7.37s$, with no errors reported across the two project phases for our own tool, including both grading and commenting (BT3). Additionally, there were statistically fewer critical incidents reported for e²Logos both during TA grading (BT1), $M_{elogos} = 0.20$ *vs* $M_{Gradescope} = 2.13$, and instructor review (BT7), $M_{elogos} = 0.05$ *vs* $M_{Gradescope} = 0.87$, an important indicator of improved efficiency, as well. Even though grading time on average was not decreased significantly using our tool, it is important to note that TAs left significantly more comments on average using e²Logos, $M_{elogos} = 10.40$ *vs* $M_{Gradescope} = 7.50$.

The new tool was also found to be more dependable, even if marginally, than Gradescope's Essay assignment type. We believe this stems from the flexibility that e²Logos offers to graders, as well as the fact that it is very responsive and robust compared to Gradescope, which frequently crashed or took a long time to load large PDF files. Regarding inter-grader communication, despite e²Logos offering a within-tool mechanism for collaboration and resolution of grading concerns, logged comments by TAs revealed that they perceived using outside tools like Discord or Groupme as "easy" and "unproblematic". Also, even though our tool was found more motivating to use, there was no significant difference in terms of clarity and ease of use (perspicuity for the UEQ). We attribute that to the multiple issues and unfinished features that TAs had to tolerate due to using the software being under development. Some features, such as the lead TA's access to edit/delete comments, were added at a later iteration of development, undoubtedly affecting the grading experience.

Qualitative feedback from the participants is fully supportive and explanatory of these findings. e²Logos proved to be more dependable than Gradescope in providing graders with the flexibility of adjusting the applied score and associated comment to fit the quality of assessed project work (satisfying our 2nd design requirement). Collaborative grading was only attempted in the early phases because it would interfere with accurate logging of individual grading in the final two phases that were used in our study; therefore, we have no solid findings about our 3rd requirement besides the ease of communicating through replies on graded comments. Such communication, however, did not yield more consistent grading results with e²Logos, as is shown by the calculated average project score difference between instructor and TAs, $\Delta 1_{elogos} = 3.70$ *vs* $\Delta 1_{Gradescope} = 3.58$, as well as instructor and lead TA, $\Delta 2_{elogos} = 3.81$ *vs* $\Delta 2_{Gradescope} = 4.42$. We attribute this to the limitations discussed below and the level of subjectivity that is involved in grading design work, a necessary evil of HCI projects.

B. Limitations

We need to acknowledge that the usability test had a rather limited sample of just nine participants, which does not allow us to draw statistically robust conclusions. The unbalanced number of projects between semesters might have affected the quality of grading done by the TAs (i.e., graders in S22 being more rushed to finish grading), but more importantly, the quality of the project submissions themselves was probably a confounding factor for the logged grading data between the two semesters (i.e., some submissions being harder to grade). We need also recognize that the process of logging data in the spreadsheet had a negative impact both on grading accuracy (distraction) but also on the measured completion time (overhead caused from switching between application and logfile). Finally, testing of e²Logos happened while the tool was being developed with different features added and refined between graded project phases. This had the unintended side-effect of influencing the measured user experience (e.g., negative comment about needing to manually update the extension).

VII. Conclusions and Future Work

This paper reports the results of a first-of-its-kind usability study comparing the efficiency and learnability of a newly developed grading tool, e²Logos, against Gradescope's Essay assignment type, for grading open-ended design projects. The findings were encouraging, revealing that the new tool was perceived as superior to its competitor in terms of efficiency, dependability, stimulation, and attractiveness based on the UEQ. Logged data while grading two phases of a design project in an HCI class, as well as open-ended comments by the participants (TAs in the course), justified the perceived higher efficiency and dependability (user control and freedom) of e²Logos. Finally, our contribution includes a list of design requirements we argue any similar software should satisfy.

Our immediate plans involve finishing development of the e²Logos Chrome extension, making it available and testing it in more courses at the university. This will allow collecting data from a much larger sample and assessing more accurately the usability of the tool, this time comparing it with the benchmark data set of typical interactive products offered by the UEQ researchers [40]. Even more importantly, we plan to evaluate the learning efficacy of the type and quality of feedback that can be provided by the tool. This will entail collecting data from students in courses that employ e²Logos, but it demands that reviewing and acting on the provided feedback is part of the learning objectives of the course. Such an approach might involve techniques like feedforward [41], with students' academic performance being compared between the ones who access the feedback on e²Logos and the students who do not review (or respond to) their graded comments on the platform. Overall, we hope our findings and design requirements tested can provide guidance for future design and development of assessment tools of online student PBL work.

 ISBN: 978-1-68558-078-0

ACKNOWLEDGMENTS

We would like to thank all teaching assistants who dedicated extra time to learn the new tool and log their grading data during the two semesters this study was conducted.

REFERENCES

[1] B. Pérez and Á. L. Rubio, "A Project-Based Learning Approach for Enhancing Learning Skills and Motivation in Software Engineering," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, Feb. 2020, pp. 309–315. doi: 10.1145/3328778.3366891.

[2] J. Zawojewski, H. Diefes-Dux, and K. Bowman, *Models and modeling in engineering education: Designing experiences for all students*. Rotterdam, the Netherlands: Sense Publishers, 2008.

[3] T. Pinar Yildirim, L. Shuma, and M. Besterfield Sacre, "Model-eliciting activities: assessing engineering student problem solving and skill integration processes," *Int J Eng Educ*, vol. 26, no. 4, pp. 831–845, 2010.

[4] Y. Kharitonova, Y. Luo, and J. Park, "Redesigning a software development course as a preparation for a capstone: An experience report," in *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, Feb. 2019, pp. 153–159. doi: 10.1145/3287324.3287498.

[5] J. J. Olarte, C. Dominguez, A. Jaime, and F. J. Garcia-Izquierdo, "Student and Staff Perceptions of Key Aspects of Computer Science Engineering Capstone Projects," *IEEE Transactions on Education*, vol. 59, no. 1, pp. 45–51, Feb. 2016, doi: 10.1109/TE.2015.2427118.

[6] V. Van den Bergh, D. Mortelmans, P. Spooren, P. Van Petegem, D. Gijbels, and G. Vanthournout, "New assessment modes within project-based education - the stakeholders," *Studies in Educational Evaluation*, vol. 32, no. 4, pp. 345–368, Jan. 2006, doi: 10.1016/J.STUEDUC.2006.10.005.

[7] L. B. Nilson and C. J. Stanny, *Specifications Grading : Restoring Rigor, Motivating Students, and Saving Faculty Time*. Stylus Publishing, 2014.

[8] M. Carmosino and M. Minnes, "Adaptive Rubrics," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, Feb. 2020, vol. 7, no. 20, pp. 549–555. doi: 10.1145/3328778.3366946.

[9] A. Singh, S. Karayev, K. Gutowski, and P. Abbeel, "Gradescope: A fast, flexible, and fair system for scalable assessment of handwritten work," in *L@S 2017 - Proceedings of the 4th (2017) ACM Conference on Learning at Scale*, Apr. 2017, pp. 81–88. doi: 10.1145/3051457.3051466.

[10] R. K. Ellis, "Field Guide to Learning Management Systems." ASTD Learning Circuits, 2009. Accessed: Mar. 11, 2023. [Online]. Available: http://www.astd.org/NR/rdonlyres/12ECDB99-3B91-403E-9B15-7E597444645D/23395/LMS_fieldguide_20091.pdf

[11] Hypothes.is. https://web.hypothes.is/ (accessed Feb. 25, 2023).

[12] M. C. Kitsantas, "Supporting Student Self-Regulated Learning in Problem-and Project-Based Learning," *Interdisciplinary Journal of Problem-Based Learning*, vol. 7, no. 2, pp. 9–14, 2013, doi: 10.7771/1541-5015.1339.

[13] J. Hattie and H. Timperley, "The Power of Feedback," *Rev Educ Res*, vol. 77, no. 1, pp. 81–112, Nov. 2007, doi: 10.3102/003465430298487.

[14] M. E. Cardella, H. A. Diefes-Dux, M. Verleger, A. Fry, and M. T. Carnes, "Work in progress - Using multiple methods to investigate the role of feedback in open-ended activities," *Proceedings - Frontiers in Education Conference, FIE*, 2011, doi: 10.1109/FIE.2011.6143106.

[15] S. Atwood and A. Singh, "Improved Pedagogy Enabled by Assessment Using Gradescope," in *2018 ASEE Annual Conference & Exposition Proceedings*, Jun. 2018. doi: 10.18260/1-2--30627.

[16] A. W. Stevens, "Assessing Student Learning Using a Digital Grading Platform," *Applied Economics Teaching Resources (AETR)*, vol. 1, no. 1, pp. 18–24, 2019, doi: 10.22004/AG.ECON.294011.

[17] S. A. Raza, W. Qazi, K. A. Khan, and J. Salam, "Social Isolation and Acceptance of the Learning Management System (LMS) in the time of COVID-19 Pandemic: An Expansion of the UTAUT Model," *Journal of Educational Computing Research*, vol. 59, no. 2, pp. 183–208, Apr. 2021, doi: 10.1177/0735633120960421.

[18] Moodle | Open-source learning platform. https://moodle.org/ (accessed Feb. 25, 2023).

[19] Y. A. Hussain and M. Jaeger, "LMS-supported PBL assessment in an undergraduate engineering program-Case study," *Computer Applications in Engineering Education*, vol. 26, no. 5, pp. 1915–1929, Sep. 2018, doi: 10.1002/cae.22037.

[20] Blackboard | Educational Technology Services. https://www.blackboard.com/ (accessed Mar. 12, 2023).

[21] F. Martin, "Blackboard as the Learning Management System of a Computer Literacy Course," *MERLOT Journal of Online Learning and Teaching*, vol. 4, no. 2, pp. 138–145, 2008.

[22] RCampus, "iRubric: Home of free rubric tools." https://www.rcampus.com/indexrubric.cfm (accessed Mar. 02, 2023).

[23] D. Myers, A. Peterson, A. Matthews, and M. Sanchez, "One Team's Journey with iRubrics," *Curr Issues Emerg Elearn*, vol. 4, no. 1, pp. 248–261, Jul. 2018.

[24] F. Burrack and D. J. M. Thompson, "Canvas (LMS) as a means for effective student learning assessment across an institution of higher education," *Journal of Assessment in Higher Education*, vol. 2, no. 1, pp. 1–19, Jan. 2021, doi: 10.32473/JAHE.V2I1.125129.

[25] B. Huang and M. S. Y. Jong, "Developing a Generic Rubric for Evaluating Students' Work in STEM Education," in *Proceedings - 2020 International Symposium on Educational Technology, ISET 2020*, Aug. 2020, pp. 210–213. doi: 10.1109/ISET49818.2020.00053.

[26] D. Grossu, "Using the Hypothesis Tool in a Synchronous Learning Environment," 2021. Accessed: Feb. 25, 2023. [Online]. Available: https://www.merlot.org/merlot/viewMaterial.htm?id=773409069

[27] C. C. Goller, M. Vandegrift, W. Cross, and D. S. Smyth, "Sharing Notes Is Encouraged: Annotating and Cocreating with Hypothes.is and Google Docs †," *J Microbiol Biol Educ*, vol. 22, no. 1, pp. 1–4, Apr. 2021, doi: 10.1128/jmbe.v22i1.2135.

[28] Diigo. https://www.diigo.com/ (accessed Feb. 26, 2023).

[29] V. P. Dennen, M. L. Cates, and L. M. Bagdy, "Using Diigo to Engage Learners in Course Readings: Activity Design and Formative Evaluation," *International Journal for Educational Media and Technology*, vol. 11, no. 2, pp. 3–15, 2017.

[30] DIGI[cation] | Make Learning Visible. https://www.digication.com/ (accessed Mar. 09, 2023).

[31] P. Nokelainen, J. Kurhila, M. Miettinen, P. Floreen, and H. Tirri, "Evaluating the role of a shared document-based annotation tool in learner-centered collaborative learning," in *Proceedings - 3rd IEEE International Conference on Advanced Learning Technologies, ICALT 2003*, 2003, pp. 200–203. doi: 10.1109/ICALT.2003.1215056.

[32] T. Ryan, M. Henderson, and M. Phillips, "Feedback modes matter: Comparing student perceptions of digital and non-digital feedback modes in higher education," *British Journal of*

Educational Technology, vol. 50, no. 3, pp. 1507–1523, May 2019, doi: 10.1111/bjet.12749.

[33] S.-W. Yeh and J.-J. Lo, "Using online annotations to support error correction and corrective feedback," *Comput Educ*, vol. 52, no. 4, pp. 882–892, May 2009, doi: 10.1016/j.compedu.2008.12.014.

[34] J. Wolfe, "Annotation technologies: A software and research review," *Comput Compos*, vol. 19, no. 4, pp. 471–497, Dec. 2002, doi: 10.1016/S8755-4615(02)00144-5.

[35] R. Hartson and Pardha. S. Pyla, *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. Elsevier, 2012.

[36] B. Laugwitz, T. Held, and M. Schrepp, "Construction and Evaluation of a User Experience Questionnaire," in *HCI and Usability for Education and Work (SAUB 2008) Lecture Notes in Computer Science*, vol. 5298, A. Hlzinger, Ed. Berlin, Heidelberg: Springer Verlag, 2008, pp. 63–76. doi: 10.1007/978-3-540-89350-9_6.

[37] J. Kirakowski and M. Corbett, "Measuring User Satisfaction," in *Proceedings of the Fourth Conference of the British Computer Society on People and computers IV*, 1988, pp. 329–338.

[38] K. L. Sainani, "Dealing With Non-normal Data," *PM&R*, vol. 4, no. 12, pp. 1001–1005, Dec. 2012, doi: 10.1016/J.PMRJ.2012.10.013.

[39] K. S. Taber, "The Use of Cronbach's Alpha When Developing and Reporting Research Instruments in Science Education," *Res Sci Educ*, vol. 48, no. 6, pp. 1273–1296, Dec. 2018, doi: 10.1007/S11165-016-9602-2/TABLES/1.

[40] M. Schrepp, A. Hinderks, and J. Thomaschewski, "Construction of a Benchmark for the User Experience Questionnaire (UEQ)," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, no. 4, p. 40, 2017, doi: 10.9781/IJIMAI.2017.445.

[41] N. Duncan, "'Feed-forward': improving students' use of tutors' comments," *Assess Eval High Educ*, vol. 32, no. 3, pp. 271–283, 2007, doi: 10.1080/02602930600896498.

 ISBN: 978-1-68558-078-0