Software and Systems Modeling (2022) 21:1531–1550
https://doi.org/10.1007/s10270-021-00966-y

REGULAR PAPER

Are models better read on paper or on screen? A comparative study

Mohamed El-Attar[1]

Received: 16 April 2021 / Revised: 1 August 2021 / Accepted: 15 October 2021 / Published online: 9 January 2022

Abstract

Is it really better to print everything, including software models, or is it better to view them on screen? With the ever increasing complexity of software systems, software modeling is integral to software development. Software models facilitate and automate many activities during development, such as code and test case generation. However, a core goal of software modeling is to communicate and collaborate. Software models are presented to team members on many mediums and two of the most common mediums are paper and computer screens. Reading from paper or screen is ostensibly considered to have the same effect on model comprehension. However, the literature on text reading has indicated that the reading experiences can be very different which in turn effects various metrics related to reader performance. This paper reports on an experiment that was conducted to investigate the effect of reading software models on paper in comparison with reading them on a computer screen with respect to cognitive effectiveness. Cognitive effectiveness here refers to the ease by which a model reader can read a model. The experiment used a total of 74 software engineering students as subjects. The experiment results provide strong evidence that displaying diagrams on a screen allows subjects to read them quicker. There is also evidence that indicates that on screen viewing induces fewer reading errors.

Keywords Paper-based reading · Screen-based reading use case diagrams · Feature diagrams · Student-based experiments · Controlled experiment · Model comprehension · Model representation

1 Introduction

The activity of software modeling is often considered from the perspective of the modeler and what can the modeler achieve by creating a software model. At the heart of the software modeling process are two actors: the modeler and the model reader. Software modeling is not just about creating a model. Software modeling is an activity by which a modeler documents mental concepts (usually abstractions about a software system) for the purpose of sharing these concepts with other stakeholders who in turn will need to reverse engineer the mental concepts by reading the models. Software modeling can be divided into two activities: model construction and model comprehension. If a model reader can reconstruct the same mental concepts specified by the modeler, then the modeling activity is deemed a success, otherwise it would be deemed a failure [1, 2]. Consequences of a failed modeling process can be severe as the development team are acting upon misinformation [3].

More recently, the perspective of the model reader (model comprehension) has started to garner attention from the modeling community. In particular, there is now focus that software models, at the diagram and notation levels, would be cognitively effective. A cognitively effective diagram or notation is one that allows its readers to read it quickly (facilitating navigation) whilst committing the fewest possible reading errors. In 2009 for example, Moody [4] presented the Physics of Notations framework which consisted of nine principles that guides notation designers to develop notations that are cognitively effective. Another framework that focuses on notation evaluation and design with the purpose of increasing their cognitive effectiveness is the Cognitive Dimensions (CDs) framework presented by Green [5], Green and Petre [6] and later by Blackwell and Green [7].

A recent survey conducted by Badreddin et al. [8] presents the state of software modeling in practice over the past decade. The survey reveals there is an increase in the practice of modeling and design [8]. The survey also reveals

Communicated by Timothy Lethbridge.

✉ Mohamed El-Attar
Mohamed.El-Attar@zu.ac.ae

[1] College of Technological Innovation, Zayed University, P.O. Box 144534, Abu Dhabi, UAE

that there is a general decline in the perception of modeling tools support for activities that involve communication and collaboration with others. In fact, participants of the survey reported that they often use pen and paper and whiteboards as a modeling platform for the purpose of communication and collaboration with others [8]. The survey also states that there is a significant increase in the use of software models in brainstorming sessions. When analyzing these reports collectively, it can be deduced that modeling is indeed a core activity that is perceived as very useful by practitioners for communication and collaboration; however, the presentation of models is more often preferred to be on paper or whiteboard. The survey results explain that the reluctance of using software modeling tools to communicate and collaborate while reasoning about models may be attributed to the perception of modeling tools being overly complex, requiring a significant learning curve and difficult to use [8]. The survey data also shows that there is a declining trend of transcribing models that are developed on paper to a modeling tool [8]. As such, it can be deduced that the models that are created informally on paper for the purposes of communication and collaboration are perhaps discarded, or at least not considered again using a software modeling tool. However, is such practice actually detrimental to the value gained from software modeling? In particular, this paper provides a comparative study on the effects of using paper versus screen on model comprehension. Is model comprehension improved when models are read on paper than on screen, or vice versa?

Beyond the realm of software engineering practice, this research study is highly pertinent to software engineering education. Prior to the coronavirus pandemic, students were already increasingly learning through online environments. In a typical software engineering degree, a student would be exposed to many software diagrams and notations throughout their studies. While students use software modeling tools, many of their education activities occur on paper. For example, examinations in many universities occur on paper. Many activities require students to read, comprehend and reason about a diagram (a software model) in order to perform various analysis and answer questions. The motivation for this study has significantly risen during the COVID-19 pandemic as online learning has effectively become the only means. Examinations and other educational activities were all forced to be online. Is the fact that students are now only considering software modeling on screen affecting their performance?

Naturally, a hand-sketched drawing of a diagram on paper is different from a software tool drawing of a diagram on paper. The cognitive effectiveness of these two types of presentations may be different. This paper only considers the latter situation, that is when diagrams are produced using software tools. The reason for this scope limitation is that a hand-sketched drawing on paper has two independent variables: (a) the fact that it is hand-sketched and (b) the fact that it is on paper. The effect of these two *treatments* cannot be separated from one another. In order to separate the first variable, the diagrams presented on paper were developed using software modeling tools. Certainly, hand-sketched diagrams can be created using software tools and can be shared electronically. This means that another comparison of the cognitive effectiveness of hand-sketched diagrams on paper and on screen can also be made. However, this conjecture is beyond the scope of this paper.

The remainder of this paper is structured as follows: Sect. 2 discusses research work related to paper reading in comparison with screen reading. The experimental design is presented in Sect. 3. Section 4 presents the experiment results and provides a discussion and an interpretation of these results. Finally, Sect. 5 concludes and suggests future work.

2 Reading on paper versus screen: a brief literature review

The study of the effects of reading on paper vs. screens on comprehension is very important and is not new as evident by the literature. There exist numerous publications dedicated to this field of research and as such this section will present a brief literature review. The object of comprehension in most of the studies is text. The overarching trend of results from these studies favor reading on paper, at least in the earlier studies. For example, the results of one of the earliest studies suggested that both speed and accuracy of proof-reading a text were impaired when the text was presented on screens [9]. Here the term *accuracy* refers to the ability of a reader to read correctly and not commit any reading mistakes. Another study in a Norwegian school context indicates that students who read texts in print scored significantly better on the reading comprehension test than students who read the texts digitally [10]. A study presented in Rasmusson [11] shows that paper-based reading is better when the text being read is short with much factual information. When the element of time pressure is introduced, text learning was less effective on screen than on paper [12]. Reading from paper led to better information retention and knowledge [13].

As technology evolves, other technology-based mediums are evaluated, such as tablets and e-readers. In these studies, the difference between the effects of these newer mediums and paper on reading speed and comprehension is not significant. A student-based study that compared tablets and printed books found no significant difference between groups with regards to reading speed or level of comprehension [14]. Similar findings were reported in an experiment that compared the efficacy of e-readers with paper and computer screens

[15]. The term *efficacy* here refers to the cognitive effectiveness of using a particular medium.

Another set of published studies compare readers' preferences with respect to reading from a computer screen or printed text. For example, a student-based study involving 254 students indicates that learners preferred print copies of text materials for reasons of portability, dependability, flexibility, and ergonomics [16]. Preference to paper-based reading was also prevalent in a set of four reading experiments [17]. Preference can be attributed to the ergonomic differences between paper and screen reading. It is well known that reading from a screen for a prolonged period causes eye strain. The medical term for such condition is Computer Vision Syndrome (CVS) [18]. Symptoms of CVS include eyestrain, tired eyes, irritation, burning sensation, redness, blurred vision, and double vision [18].

But the outlook for on-screen reading is not grim, to the contrary in fact. Technology has evolved drastically from the time many of the reading studies were conducted and published. In fact, a time line of studies published in this area indicate that the magnitude of the difference in reading comprehension between paper and screen followed a diminishing trajectory [19]. Moreover, reading diagrams is not the same as reading text and they are processed differently according to the dual channel theory [20]. According to this theory, pictorial and verbal materials are processed in the human mind using separate systems [20]. The most important distinction is that visual representations are processed in parallel while textual representations are processed serially by the auditory system [21]. Therefore, it is not safe to generalize the results of studies that used text as the object of comprehension to the context of software model comprehension.

In summary, the results of previous studies cannot be generalized to the context of reading software models. The study presented in this paper compares reading software models on two popular mediums and their effects on model comprehension. This study is very important given the current state of software modeling in industry, in particular, with the constantly increasing utilization of software modeling in development and the documented evidence of the use of paper and computer screens as presentation mediums [8].

2.1 Paper versus screen: physics and psychology

Paper and screen have different physical characteristics that can affect reading. In a study conducted by Mangen et al. [10], students scored significantly better when reading on paper because paper gives spatio-temporal markers while reading. It was determined that touching the paper and turning pages enhances memory which makes it easier to remember where (the location) of something was read [10]. Scrolling a computer screen hinders the ability to remember the location of information. It was also demonstrated that paper readers generally make a better calibration than screen readers, which improves results as readers do not stop studying too soon [12].

Viewing a screen for a couple of hours before bedtime can be disruptive to sleep as the blue light of the screen may suppress the body's production of melatonin [22]. In the software development spectrum, sleep deprivation can be especially problematic as an experiment conducted by Fucci et al. [23] shows that sleep-deprived developers make more fixes to syntactic mistakes in the source code. The authors foresaw that sleep-deprivation can be disruptive to other types of development activities, modeling included [23].

Some physical attributes of screen viewing are more advantageous than paper viewing. In the experiment conducted by Kretzschmar et al. [24], the authors studied eye movement, brain activity and reading speed and found no evidence to support that reading on screen is more effortful than reading on paper. In fact, it was found that older participants of the experiment were able to read faster and with less effort on screen due to the back lighting providing a better contrast [24].

The superiority of paper reading in many experiments can be attributed to more psychological than physical reasons. The mental state and comfort of a reader is not an aspect to be taken lightly or dismissed and can have a profound effect on a reader's performance. In a later experiment conducted by the same authors of Ackerman and Lauternman [12], it was found that it was possible to overcome screen inferiority but only for those subjects who preferred reading from screens priori [25]. This is a significant finding as it can be argued that younger generations of readers are likely to have different attitudes and preferences than the older generations.

An important physiological and physical aspect to consider is the behavior of a reader when reading from the two mediums. An example favoring paper is that paper is lightweight and portable and hence it allows the reader a large degree of mobility freedom whereas a large computer screen is largely stationary and forces the modeling activity to be fully sedentary, which may negatively impact the reader. On the other hand, people with poor organization skills may prefer to simply read from a screen simply because if they constantly lose printed documents. Reading from a screen without resorting to printing is also a green-activity that benefits the environment, an aspect that weighs heavily for many people (as it should for all people).

2.2 Motivation

The results of an experiment do not necessarily need to prescribe a change in the way we perform modeling in order to be useful. It is the role of researchers to provide evidence of the efficacy of current practices in comparison with alternative practices. In case current trends are more beneficial (or

not), this fact should be corroborated with empirical evidence rather than anecdotal evidence.

Practically, the results of the experiment, regardless of which medium it favors, will have a significant impact. As of 2016, it is estimated that there are 21 million professional software developers [26]. It is also estimated that the number of individuals currently working in the IT (Information Technology) sector is higher than 50 million professionals [27]. Software modeling as a practice is also increasing in industry [8]. Assuming that only a small subset of these professionals perform software modeling as part of their job responsibilities, this means that software modeling is still being performed by a significant population of software professionals and their practices are affecting an equally significant number of software modeling activities and projects. Despite opinion formed based on anecdotal evidence of the supremacy of a particular medium, it is safe to argue that not all software modelers are practicing their profession in the same manner and using the same medium. Therefore, for modelers who are currently using a suboptimal medium, the results of the experiment should prompt them to change how they practice modeling. As for modelers who were using the better medium, the results offer welcomed empirical evidence of their correctness.

A natural argument here is that software engineers would naturally want to perform all their development duties (modeling or otherwise) on computers. However, given the natural supremacy of reading text on paper than on screen, it can be argued that some software engineers may choose to print software-related material (models, code, test reports) rather than view them on screen, assuming that the effect will be the same if the material is actually models (diagrams). Another reason why software modeler readers may choose to print models is the well-known fact that looking at a computer screens for a lengthy period of time will cause eye strain [18]. There lacks data that decisively indicate that *all* software modeler readers look at their screen for short periods of time and thus it can be assumed that there are modelers who require a lengthy period of time to read their models and therefore would be inclined to print the models instead.

Printing large diagrams is perhaps rare due to physical and logistical challenges, but printing smaller diagrams that can appear on A4 paper (or similar size) do not suffer the same challenges if a modeler decides to print them. Diagrams shown on A4 paper can represent a subsystem of a much larger system and hence printing them and viewing them (in focus) can be of value to modelers. It is a very reasonable activity for a designer to reason about a particular module of the system rather than reason about the design of the entire system altogether simultaneously.

The experiment presented in this paper is concerned with models that are drawn with software tools not sketches. Anecdotally, modeling activities during a project's infancy stages are performed using sketches before being transformed into a softcopy using software tools. The softcopy version of the models can then be used to generate code skeletons. But is that where the lifecycle and usefulness of these models end? Viewing A4-size software-built diagrams can still be useful beyond code generation. These software-built models can be retrieved (on screen or printed on paper) to be examined and changed for a multitude of reasons: for example, to perform model refactoring in order to make designs more flexible and reusable, or to perform maintenance activities where the models would change to add, remove or edit features.

Another reason the experiment setting which uses software-built diagrams on paper (and screen) is important as it is highly pertinent to the education sector. Some examinations are conducted on paper while others are conducted online (in part due to COVID-19). Regardless of the medium, the exams show software engineering students diagrams that are software-built not sketched. The results of the experiment would indicate which medium would be more beneficial to the student. Considering the large number of IT-related degrees and students around the world who are enrolled in them, the results of this investigation have a far reaching impact in the education spectrum.

It should be noted that the purpose of the experiment is not to compare sketches with software-built models. The purpose of the experiment is to examine the physical effect of the mediums on cognitive effectiveness. While this is similar to traditional paper vs. screen experiments, this experiment is the first (to the best of the author's knowledge) that makes this comparison when the material to be read is diagrams not text, which as mentioned previously are different because they are processed differently according to the dual channel theory [20]. This distinction is not insignificant and may (and will) yield different results than historical trends in text-based paper vs. screen experiments. This is because according to the dual-channel theory hypothesized by Allan Paivio in 1971, visual and textual information is processed differently along distinct channels in the human mind [28]. In text-based paper vs. screen experiments, the subjects are only exposed to text thus using and quickly overloading the verbal processing channel in the human information-processing system while completely underutilizing the pictorial processing channel. While the literature reports many experiments that exposes subjects to pictures, this would be the first experiment of its kind that exposes the subjects to pictures (software models in that case) while comparing their performances between reading from paper vs. screen. Hence, it is unchartered territory to study the effect of such factors (variables) on cognitive effectiveness as setup in the experiment presented in this paper.

3 Experimental design

This section describes a controlled experiment that took place at the campus of Alfaisal University in Riyadh, Saudi Arabia.

3.1 Experiment definition

The main purpose of this experiment is to investigate the effect of reading software models on paper in comparison with on screen with respect to model comprehension. In particular, the efficacy of the reading mediums is assessed based on their cognitive effectiveness. Recall that cognitive effectiveness is a measure of the ease by which readers can read a model. At a concrete level, easiness is measured in terms of speed and accuracy by which the models can be read [4]. A better performance in this regard would be reflected in mediums that allow reader to read quicker and to commit fewer reading errors. For both mediums, the diagrams presented will be developed using a software modeling tool. Hence, two treatments exist: (a) paper presentations and (b) screen presentations. The experiment consisted of two parts, a UML use cases diagrams [29] part and a feature diagrams [30] part. In each part, the subjects were required to consider a diagram using the two mediums of presentation. Therefore, the experiment revolves around the independent variables of the mediums and the diagram types. To assess the effect of the two presentation mediums on cognitive effectiveness, two dependent variables were recorded: the response time for the subjects to answers questions by retrieving information from the presented models (T), and the reading errors committed when answering the questions (E). A 2×2 factorial experiment design is used.

3.2 Experiment context

The experiment involved students at Alfaisal University in Riyadh, Saudi Arabia. The experiment consisted of two parts, using two different diagrams in order for the comparison evaluation not to be based on one particular notation type. The two experiment parts are completely independent. The use cases diagrams experiment part took place during the Spring 2019 semester while the feature diagrams experiment part took place during the Spring 2020 semester. Note that the second experiment part was completed before the coronavirus pandemic had reached Saudi Arabia. The experiment specifically used these two diagrams types as it revealed that "it is only in the requirements phase that more than half of the participants reported using modeling frequently. Across other activities, modeling remains rather low [8]." As such, requirements related diagram types are the most pertinent to actual practice trends and would be most useful to deploy in this experiment.

Table 1 The dependent variables and their corresponding hypotheses for the two experiment parts

Dependent variable	Null hypothesis (Ho)	Alternative hypothesis (Ha)
Experiment—use cases part		
Response time	(Ho1): T (Paper) = T (Screen)	(Ha1): T (Paper) ≠ T (Screen)
Errors committed	(Ho2): E (Paper) = E (Screen)	(Ha2): E (Paper) ≠ E (Screen)
Experiment—feature diagrams part		
Response time	(Ho3): T (Paper) = T (Screen)	(Ha3): T (Paper) ≠ T (Screen)
Errors committed	(Ho4): E (Paper) = E (Screen)	(Ha4): E (Paper) ≠ E (Screen)

3.3 Hypotheses formulation

The hypotheses formulation is based on the two recorded dependent variables (T and E). For the two variables and in both experiment parts, the alternative hypotheses indicate that the subjects will have significantly different performance levels when using the two presentation mediums. The null hypotheses indicate that the performance levels will be the same irrespective of the presentation medium. The hypotheses testing will be initially set as two-tailed to investigate the possibility of either medium outperforming the other. If a statistically significant result is observed, an additional one-tail test will be performed to determine which medium outperformed the other. The hypotheses formulation is shown in Table 1.

3.4 Subject selection

All students were registered in a Software Requirements Engineering course at the time of their involvement in the experiment. The subjects participated in the experiment on voluntary basis. The students were currently enrolled in the undergraduate software engineering degree at Alfaisal University in Riyadh, Saudi Arabia. The students were familiar with the notations of use case modeling and feature diagrams as part of their standard course study. Therefore, the experiment did not contain a tutorial component of the notations but a seminar was given to the students to explain the format of the experiment and what is the nature of the prescribed experimental tasks. The experiment parts were conducted at a time during the semester when the subjects have already been introduced and assessed on their knowledge and skillset of the two types of diagrams. Some students were motivated to participate in the experiment as they wanted to be exposed to the latest research trends. Other students with aspirations to pursue graduate studies wanted to partake in the experiment to experience empirical software engineering firsthand as

Table 2 The experimental design of the use cases experiment

Use case diagrams experiment part				
Session 1	Group A	Movie theatre (paper)	Group B	Movie theatre (screen)
Session 2	Group A	UTube (screen)	Group B	UTube (paper)
Feature diagrams experiment part				
Session 1	Group C	PDF (paper)	Group C	PDF (screen)
Session 2	Group D	ERP (screen)	Group D	ERP (paper)

subjects before becoming researchers themselves. For each experiment part, the subjects were divided into two groups. Since the purpose of this investigation is not to assess the students themselves, the groups were designed to have similar ability levels. The subjects were divided into two groups based on their assessment performances with the diagrams types and their overall academic standing. Two students with similar ability levels were randomly assigned to a group belonging to an experiment part. For the use case diagrams part of the experiment, the two groups (A and B) contained 18 students each, for a total of 36 students. For the feature diagrams part of the experiment, the two groups (C and D) contained 19 students each, for a total of 38 students. The subjects who were involved in the first experiment part (use case diagrams part) did not partake in the second experiment part. The subjects were not informed about the hypotheses under investigation to reduce the risk of bias.

3.5 Experimental tasks and artifacts

The presentation medium is the only treatment under investigation. Therefore, for the two experiment parts, the subjects considered one distinct diagram in the first session. In the first session, one group would have that diagram presented to them on paper while the other group would have the diagram presented to them on screen. For the use case diagrams part, the first diagram pertained to a movie theatre management system while the second diagram pertained to a video streaming service. For the feature diagrams experiment part, the first diagram pertained to features that would exist in the popular document processing software Adobe [31]. The second feature diagram related to features from an ERP (Enterprise Resource Planning) business management system. The experimental design is presented in Table 2. The experimental artefacts are available in a resources package available online [32].

Once again, the focus of the experiment is not to assess the artefacts themselves. Therefore, the diagrams used in this experiment needed to be similar in size and complexity. Tables 3 and 4 show the size attributes of the use case and feature diagrams used, respectively. Additional criteria were also identified and satisfied for the purpose of not having the relative complexity of the diagrams influencing the results of the experiment:

Table 3 Use case diagram sizes in terms of the number of graphical constructs

	Movie theatre	UTube
Nodes		
Use cases	11	11
Regular	8	8
As a rectangle	1	1
As a classifier	1	1
Abstract with extension points	1	1
Actors	3	3
Total Nodes	14	14
Relationships		
Communication	5	4
Extend	5	8
Include	4	2
Total relationships	14	14
Total elements	28	28

1. Each diagram is constructed using the same layout when presented on paper or on screen. This criterion is important as prior research in the field of model comprehension have shown that varying layouts can affect model comprehension [33].
2. The diagrams needed to be of a significant size to allow meaningful statistical analysis to be performed. To satisfy this criterion, diagrams used in prior experimental research in model comprehension were used as a benchmark, such as Purchase et al. [33, 34], Gopalakrishnan et al. [35]; and Reijers and Mendling [36], and the diagrams used in this experiment were designed to be larger (in terms of nodes and edges).
3. The diagram dimensions (in terms of height and width) are made to be the same as viewed on paper and on screen.

Upon completing the experimental tasks, the subjects are provided with a post-experiment questionnaire ("Appendix 1"). The questionnaire is designed to solicit qualitative data that can shed light and corroborate findings from the quantitative data.

Table 4 Feature diagram sizes in terms of the number of graphical constructs

	PDF diagram	ERP diagram
Features and cardinality		
Mandatory features	40	33
Optional features	31	38
Dead features	2	2
Attributes	2	2
Refer	1	1
Feature cardinality	7	7
Group cardinality	4	4
Total	87	87
Relationships		
Require	4	4
Exclude	4	4
Generalization	4	4
Implementation	4	4
Alternative	14	13
Or	1	2
And	12	15
Total relationships	42	46
Total elements	130	133
Configurations		
	45	49

3.6 Instrumentation

Conducting the experiment online may have the benefits of involving a larger number of participants. However, the experiment was designed to be conducted at the software engineering laboratory at Alfaisal University due to the following reasons:

1. To ensure that the subjects do not collaborate with each other.
2. To ensure that the diagram dimensions are maintained between the two mediums.
3. To ensure the size of screens used are the same and that the screens are large enough to present the entire diagrams without scrolling.
4. To ensure the diagrams are printed on the same size paper.
5. To provide assistance to subjects with regards to facilitating the operation or the logistics of the experiment, hence minimizing interruptions.
6. To prompt the subjects to maintain their focus on the experimental tasks.

The diagrams were all printed on A4-size paper. The screen sizes at the software engineering laboratory are large enough to present an entire diagram using the same dimensions as that printed on A4-size paper without the need to scroll.

For both sessions, the subjects were presented with questions using the open-source learning platform Moodle [37]. The subjects also submitted their answers through Moodle. There are a number of benefits to using Moodle:

1. Time keeping is performed via a build-in function thus eliminating human timing errors.
2. The questions were close-ended and non-subjective and thus Moodle can perform the scoring, thus eliminating human scoring errors.
3. The order of the questions can be randomized which further mitigates the threat of subjects collaborating with each other.
4. Students can submit their responses to open-ended questions via Moodle thus eliminating legibility issues which may arise if the subjects provide hand-written responses.

3.7 Analysis procedure

For the time and errors variables, all quantitative data will be considered as discrete data. When considering errors data, each error will be assigned equal weight as there no evidence that will safely allow the assignment of unequal weights to the different types of errors. The analyses performed are as follows:

1. Normality tests are performed on all data sets.
2. Correlation tests between the time and errors variables are performed.
3. Hypotheses testing is then performed and corroborated with size effects calculations.
4. Qualitative analysis is performed using inductive thematic coding [38].

The statistical analyses performed considers raw data from each experiment part separately. It would not be appropriate to combine raw data from the two experiment parts within the same statistical tests.

3.8 Scoring and measurement

Scoring and measurement of both the time and errors variables will be performed automatically by Moodle. The correct answers were pre-programmed into Moodle; however the results are never revealed to the subjects during the experiment. The subjects submit their responses to the open-ended questions in the post-experiment questionnaire through Moodle as well, which in turn generates a spreadsheet with their responses.

Table 5 Normality test results

Experiment part	Medium	*n*	Variable	*p* value	Skewness	Kurtosis	Shapiro–Wilk	Normal?
Use case diagrams	Paper	36	Time	0.006	− 0.51	− 1.03	0.91	✖
	Screen	36		0.002	0.47	− 1.28	0.89	✖
	Paper	36	Errors	0.057	0.12	− 1.10	0.94	✔
	Screen	36		0.0001	1.47	0.86	0.74	✖
Feature diagram	Paper	38	Time	0.0001	1.80	4.26	0.83	✖
	Screen	38		0.0001	3.87	18.20	0.59	✖
	Paper	38	Errors	0.055	0.46	− 0.74	0.94	✔
	Screen	38		0.001	0.48	0.02	0.89	✖

Table 6 Correlation results

Experiment part	Medium	r statistic	*n*	*p*	Correlation?
Errors versus time correlation					
Use case diagrams	Paper	0.219	36	0.2001	✖
	Screen	0.051	36	0.7675	✖
Feature diagrams	Paper	0.149	38	0.3705	✖
	Screen	0.077	38	0.6441	✖

3.9 Experiment data and replication

Facilitating experiment replication is important. All experimental artefacts, raw data and statistical analyses files can be downloaded [32].

4 Experiment results

The experiment results are presented in Sects. 4.1 and 4.2. Section 4.1 presents the quantitative data, analysis and interpretation. Section 4.2 presents the qualitative analysis and discussion.

4.1 Quantitative results and analysis

The quantitative analysis presented in this section is in line with the analysis procedure previously outlined in Sect. 3.7.

4.1.1 Normality tests

The Shapiro–Wilk test [39] was used to perform the normality tests. The Shapiro–Wilk test was selected for this experiment since there is no causal explanation as to the distribution of the data. Normality tests results are shown in Table 5 which indicate that the majority of data sets *do not* conform to a normal distribution. Accordingly, the data sets will be conservatively treated as nonparametric.

4.1.2 Correlation analysis

Correlation analysis is performed between the time and errors variables to determine if they are independent or if they influence each other. The correlation analysis was performed using the Spearman correlation test [40] with the correlation coefficient ($r \neq 0$) set at the standard 0.05 level. The results are shown in Table 6, which indicate that there is no correlation detected between any of the data sets. The scatter plots are available in the resources package [32].

4.1.3 Hypothesis testing and effect sizes

The hypothesis testing was performed using the Mann–Whitney U statistic to test for differences between the medians of related samples. There are four hypothesis tests conducted, two for each experiment part. The two tests in each experiment part correspond to the subject performances with respect to the reading time (T) and reading errors committed (E) variables.

Effect size measures were calculated using Cliff's delta [41–43, 43]. When comparing two data sets, if the calculated confidence interval of Cliff's delta contains negative numbers only then this result would be considered to favor screen-based reading. Meanwhile, if the calculated confidence interval is entirely within the positive range, then the result would be considered to favor paper-based reading.

Table 7 presents the Mann–Whitney test results. According to the results, there is a statistical significance observed in both experiment parts with respect to the time variable. How-

Table 7 Mann–Whitney test results

Experiment part	Variable	Difference between medians	95% CI	Mann–Whitney U statistic	2-tailed p	1-tailed p
Use case diagrams	Time	242	124– 403	291	**0.0001**	**0.0001 (Screen)**
	Errors	1	− 1 to 3	549.5	0.2651	
Feature diagrams	Time	1638	1499–1820	24.5	**0.0001**	**0.0001 (Screen)**
	Errors	5	4–7	115.5	**0.0001**	**0.0001 (Screen)**

Bold values indicate statistically significant result observed

Table 8 Cliff's delta calculations

Part	Variable	Cliff's delta	Variance	Statistical significance?	Favoring	Confidence interval around delta	
						Maximum	Minimum
Use case diagrams	Time	0.5509	0.0127	✔	Screen	**0.7316**	**0.2977**
	Errors	0.1520	0.0204	✖		0.4103	− 0.1288
Feature diagrams	Time	0.9661	0.0011	✔	Screen	**0.9939**	**0.8202**
	Errors	0.8534	0.0051	✔	Screen	**0.9429**	**0.6491**

Bold values indicate statistically significant result observed

ever, with respect to the errors committed variable, statistical significance was only observed in the feature diagrams part of the experiment. When there is more than one test per variable, the likelihood of a Type-1 error increases. The Family-Wise Error Rate (FWER) increases as there is two tests per variable. The Bonferroni corrected α can be calculated as $0.05/2 = 0.025$. Therefore, the FWER can be adjusted to: $1-(1-0.025)^2 = 0.049$. As such, the significance valuation is not changed. With the three statistically significant results having a p value of 0.0001, the likelihood of there being a Type-1 error is extremely low. This also means that re-running the experiment will mostly likely also yield the same statistically significant results of each experiment part.

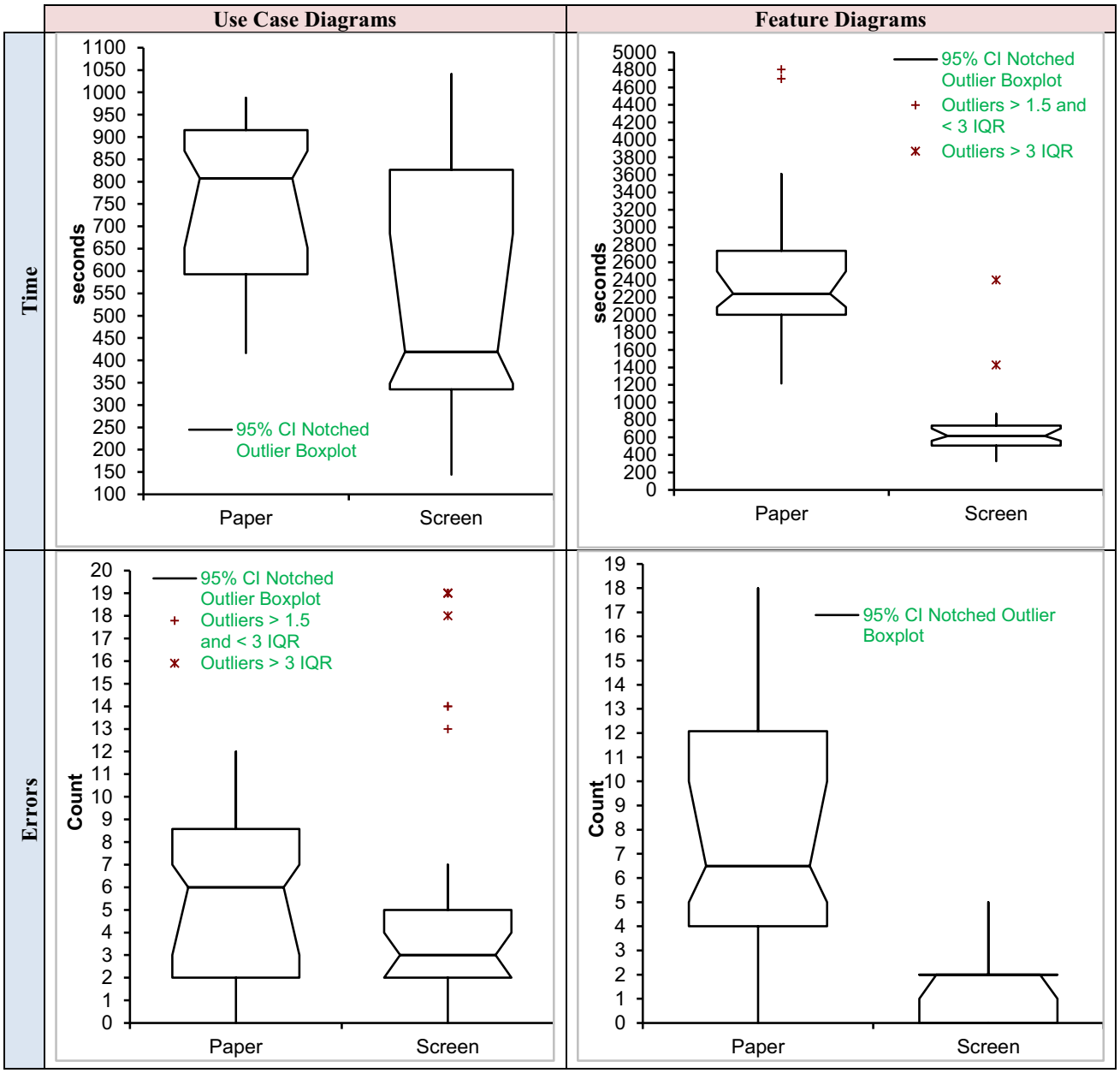
A further set of 1-tailed tests were performed to determine which presentation medium does the statistical significance favor. For all three one-tail tests performed, the results favor screen-based reading. Table 8 present the effect size calculations which corroborate the observations from the hypothesis tests presented in Table 7.

It is interesting to investigate the ordering effects between the two mediums in the two experiment parts, particularly carryover effects. In both experiment parts, the group that considered the diagrams on paper first experienced a drop in response times in the second session; − 54% in the use cases part and − 76% in the feature diagrams part. Similarly, the group that considered the diagrams on screen first experienced an increase in response times in the second session; + 7% for the use cases part and + 198% in the feature diagrams part. However, the variance in the response changes is significantly different. Therefore, the carryover effect cannot be accurately attributed to the medium type but likely dependent on the diagram type. With respect to errors committed, the group that considered diagrams on paper first experienced a drop in the errors committed in the second session; − 25% in the use cases part and − 83% in the feature diagrams part. The group that considered diagrams on screen first experienced an increase in errors committed of + 307% in the feature diagrams part and a decrease in the use cases part of − 19%. This further indicates no pattern in the results due to ordering effects. The changes observed are therefore likely attributed to the diagram types considered in each experiment part. A 2-way ANOVA test was conducted to determine the interaction between the mediums and diagram types. For the response times and errors committed variables, the tests produced a p value of 0 which indicates that the relationship between medium and the two variables is indeed heavily dependent on the diagram type. This is expected as the two diagrams types are significantly different in semantics, visual organization and number of notational constructs presented that the subjects had to consider. As the subjects are exclusively either involved in the use cases or feature diagrams experiment part, the interaction characteristics does not threaten the validity of the results. Moreover, as mentioned in Sect. 3.7, the raw data from the two experiment parts were not combined while performing any statistical analysis.

4.2 Interpretation and discussion of the results

This section provides an interpretation of the quantitative analysis results. To aid the interpretation of the results, a visual comparison of the group performances with the two presentation mediums is presented in Table 9. Moreover, Cliff's delta can be used to provide insight not just on the

Table 9 Visual comparison of the group performances using the two presentation mediums



existence of statistical significance, but also the magnitude of difference between two populations. The magnitude of Cliff's delta is assessed using the threshold levels of $d < 0.147$ "negligible", $d < 0.33$ "small", $d < 0.474$ "medium" and otherwise "large" as provided in Romano et al. [44].

A major finding from the experiment is the vast difference in time performances which was observed in both experiment parts. In the use cases part of the experiment the average completion times was roughly twice as much when reading from paper in comparison with reading from computer screens. In the feature diagrams part of the experiment, the difference was much wider, roughly four times longer when reading from paper. In fact, Cliff's delta for the time variable in both experiment parts fall in the "large" category. The degree of difference between the two experiment parts with respect to the response time variable is also interesting. In the feature diagrams part of the experiment, the difference between paper and screen is roughly twice as that observed in the use case diagrams part. This perhaps can be attributed to the difference in size between the diagrams used in the two experiment parts. Recall that the feature diagrams used contain roughly five times the number of graphical constructs as the use cases experiment part. As shown in Table 9, the reading times of subjects in the feature diagrams experiment

Table 10 Subject responses to Q1 and Q2 in the post-experiment questionnaire

Question	Experiment part	On paper	On screen	No difference
Do you prefer reading diagrams on screen or on paper?	Use case diagrams	5	9	22
	Feature diagrams	5	13	20
Do you find it easier reading diagrams on paper or on screen?	Use case diagrams	3	30	3
	Feature diagrams	3	30	5

part is roughly four times longer that than in the use cases part. As such, it can be argued that as the diagrams get larger, the difference of the efficacy of the presentation mediums on reading times also becomes larger.

For the reading errors variable, the size of the diagrams did not seem to effect the subjects. The performance of the subjects overall in the two experiment parts were ostensibly the same. However, the Cliff's delta for the feature diagrams part falls in the "large" range while in the use cases part it falls in the "small" range, with no statistical significance in the latter. Once again, it can be deduced that there is a correlation between the size of the diagram and the degree of difference between the two presentation mediums. As the diagrams become larger, the difference between paper-based reading and screen-based also becomes larger.

The post-experiment questionnaire contained two closed ended questions. The questions **Q1** and **Q2** asked the subjects about their medium preference and which medium they found easier to read from, respectively. The results for **Q1** indicate that the majority of subjects feel indifferent with respect the mediums, followed by reading on screens and then reading on paper. For **Q2**, the results indicate that the majority of the subjects perceived reading on screen enables them to read diagrams easier (Table 10).

4.3 Qualitative analysis

Qualitative analysis is used to provide a more accurate interpretation to the quantitative findings observed by analyzing the subject responses to open-ended questions in the post-experiment questionnaire. The open-ended questions were formulated to better understand the reasons for the subject preferences and the reasons they perceived a particular medium to enable them to read the models easier. The complete set of responses are presented in "Appendix 2". Most of the responses to the open-ended questions were either empty (a subject would just enter a random character in the text field to bypass the question), irrelevant, or neutral. The responses shown in "Appendix 2" are only those that are informative and add value with respect to interpreting the results.

The qualitative analysis process is iterative and it is concerned with subjecting the data (textual quotes) to thematic coding [38]. No presumptions about the data were made and hence inductive coding was used. The process is performed as follows:

1. An initial data sample is retrieved.
2. A set of codes are created to *cover* the data sample.
3. Another data sample is retrieved.
4. The codes created for previous samples are used to try and cover the new sample. If the existing codes do not provide the appropriate coverage, then new codes are defined or refined. The new code set is applied to all previously retrieved data samples.
5. Steps 3 and 4 are repeated until all data samples are enumerated.
6. The codes are categorized and documented in a codebook.

The final set of codes defined are stored in a codebook which is shown in "Appendix 2". The set of codes are then divided into three categories: "Enhances Reading", "Physical Attributes and Effects", and "General Preferences and Norms". Within each category the codes are set as either good, bad or neutral codes (if applicable). The code categorization and the qualitative results for the three categories are shown in Tables 11, 12 and 13, respectively. In general, the amount of qualitative data retrieved is lower than expected however they do offer some insight to the findings from Sects. 4.1 and 4.2.

In Table 11, it can be seen that the subjects have overwhelmingly indicated that the screen medium improves their ability to read the diagrams. Code-9g (easier to locate graphical symbols) received the highest number of citations. Interestingly, there has been two citations of Code-6g (more appropriate for big diagrams) in which two subjects deem on screen reading more appropriate in case the diagrams were larger. One of those two subjects made a Code-6b (harder to navigate) citation expressing that the diagrams would be harder to navigate if it was larger and presented on paper and that the diagram at such point will need to be drawn on a whiteboard or posted up on a wall. The citations for Code-2g (graphical constructs popping), Code-10g (better focus) and Code-12g (improved contrast display) indicate that screens help point-out the various graphical constructs to the subjects which in turn helps them focus better.

In the 'Physical Attributes and Effects' category (Table 12), the subjects' responses favored the two mediums evenly with respect to good and bad codes. Perhaps not surprisingly is that subjects have cited Code-7g (comfort) four times for paper, indicating they find reading on paper to

Table 11 Qualitative results for the Enhances Reading category

	Enhances reading						
	Good codes			Bad codes		Neutral codes	
	Paper	Screen		Paper	Screen		
Code-2g Graphical constructs popping	0	1	Code-6b More appropriate for big diagrams	1	0	Code-1n Exam reading	2
Code-6g More appropriate for big diagrams	0	2					
Code-8g Quicker to locate graphical constructs	1	1					
Code-9g Easier to locate graphical constructs	0	5					
Code-10g Better focus	0	1					
Code-12g Improved contrast display	0	1					
Code-13g More appropriate to view on medium	0	1					
Total	1	12	Total	1	0	Total	2

Table 12 Qualitative results for the physical attributes and effects category

	Physical attributes and effects				
	Good codes			Bad codes	
	Paper	Screen		Paper	Screen
Code-1g brighter	0	2	Code-1b dim	2	0
Code-3g looking up	0	2	Code-2b looking down	1	0
Code-4g less stress on eyes	1	0	Code-3b glare is tiring	0	1
Code-7g comfort	4	0	Code-4b higher workload and tiredness	0	2
			Code-5b accessibility issue of no printer available	1	0
Total	5	4	Total	4	3

Table 13 Qualitative results for the general preferences and norms category

	General preferences and norms			
	Good codes		Neutral codes	
	Paper	Screen		
Code-5g general preference	1	2	Code-2n no difference	4
Code-11g convenient	0	1	Code-3n normally prints everything on paper	1
			Code-4n expected norm as software engineers	1
			Code-5n generally used to medium	1
Total	1	3	Total	7

be more comfortable (for their eyes). This finding is further corroborated with the Code-3b (glare is tiring) and Code-4b (higher workload and tiredness) citations in which the subjects indicate that the screen glare and viewing the screen in general strains their eyes. In fact, it was expected that there would be more citations against screen viewing. The lower than expected number of citations can be attributed to the fact that most subjects were able to complete the experimental task which required them to look at screens in a period of time that is short enough not to induce eye strain. The lower than expected number of citations may also be attributed to newer screen technologies that better address eye strain issues in their design. One unexpected finding is the citations for Code-2b (looking down) and Code-3g (looking up) in which the subjects indicated that viewing diagrams at their eye level on the screen helped them navigate the diagrams quicker than looking down at the paper which was at the desk level.

In the 'General Preferences and Norms' Category (Table 13), Code-2n (no difference) received that highest number of citations indicating that that subjects were mostly indifferent between the two mediums with respect to their general preference. This result is in line with the quantitative results obtained in Sect. 4.2.

In Sect. 2.1, a comparison of the physical and psychological impacts of the two mediums on readers is presented. In light of the results of the experiment, it can also be argued that the subjects' preferences for medium is evenly divided likely due to their age. The subjects are college students who are all born roughly around the turn of the millennium. Had the subjects been older then perhaps paper would have been cited more favorably and vice versa had the subjects been born 20 years later. However, as the preferences were evenly divided, it is not expected that preference had psychologically affected the subjects to perform better with screens. The back lighting of screens providing better contrast perhaps have also attributed to the quicker reading speeds as was the case in the experiment conducted by Kretzschmar et al. [24]. The qualitative data refers to screens as "brighter" which can be interpreted as having a back lighting that offers a better contrast. The conditions of the experiment and the duration of the prescribed tasks do not investigate the issue of sleep disruption and deprivation due to prolonged viewing of a computer screen. Finally, the spatio-temporal markers offered by reading from paper, which improves memory, is perhaps the reason why the subject performances with respect to the errors committed variable were not as significantly different as they were with the response times variable.

4.4 Threats to validity

The threats to the validity is presented in accordance with the standard classification presented in Wohlin et al. [45].

4.4.1 External validity

External validity is concerned with how safe the results observed in this experiment can be generalized to other contexts. To this end, it should be noted that the results of this experiment should not be safely generalized to the following contexts:

- The models are drawn on whiteboard or if they were posted on a wall.
- The models are hand-sketched.
- The models are presented using a projector.
- Other diagram types are used.
- The diagrams used contained color.
- Multiple related models are being considered at the same time whereby a hyperlinking feature can actually affect on-screen reading only.
- In industrial settings where the diagrams are being considered by professionals who can be classified to have intermediate or senior level experience with the given type of diagram.
- If the diagrams were larger in terms of dimensions and graphical constructs.

The experiment results do provide a forecast on the results that maybe observed in those contexts. For example, it can be seen from the difference in the results obtained in the two experiment parts that as the diagrams get bigger, the results would further favor on screen viewing. But what if the diagrams were excessively large? A computer screen, unless specifically designed to be very large, can only show a part of the diagram and the user would need to constantly scroll or zoom out to navigate the diagram to be able to obtain a contextual perspective of the model presented.

The subjects used are students and it can be argued that the results obtained by seasoned professionals may have been different. In a paper previously published in this journal in 2019, it has been shown that the performance of the subjects in model comprehension experiments is not dependent on their status as students or professionals [46]. The performance of the subjects is mainly dependent on their familiarity, training and expertise with a given diagram type [46]. The training and expertise level of the students used can be equated to junior professionals. It can be argued that the percentage of software engineering professionals who are classified as junior is naturally significant, and hence the results of this experiment is pertinent to a very large population.

If different types of diagrams are used, such as UML class, activity or statechart diagrams [29], then the results perhaps may have been different. The experiment was specifically designed to use feature and use case diagrams as both are requirements-related diagram types and requirements mod-

eling is the most common type of modeling in practice according to the survey results presented in Badreddin et al. [8]. As such, it is expected that the results of this experiment is pertinent to a significant percentage of the software modeling activities performed in industry.

4.4.2 Internal validity

The results of subject-based experiments can be influenced by fatigue and maturity. However, in this experiment, fatigue and maturity is not expected to have affected the experiment as the task durations were not long. The subjects were able to complete their experimental tasks in an average less than 27 min, respectively. Subjects are accustomed to such task durations are thus are not expected to have experienced fatigue and maturity. This fact is corroborated by the subjects' responses to two corresponding questions in the post-experiment questionnaire (see "Appendix 1").

The different ordering of the questions mitigates the chance for collaboration. This means that while each subject is tasked with answering the same set of questions, randomizing the ordering means that each subject is likely to have had a unique experience. Different question orders may have helped certain subjects or it may have been an impediment to other subjects. There is no means to accurately measure this effect. However, the qualitative results did not reveal that question ordering was an issue during the experiment.

4.4.3 Construct validity

Construct validity is concerned with having the measure covering the specific target theoretical construct. The experiment is highly constrained to limit the number of independent variables. Nevertheless, the experiment did not use all the available types of screen and paper sizes. The experiment did not used all types of diagrams. The experiment did allow screen users to utilize tool-based features such as zooming in and out.

4.4.4 Conclusion validity

Homogeneity amongst the subjects, apart from the treatments that are being investigated, is paramount is such type of experiments. Meaning that the levels of all factors between corresponding subject groups, except for the presentation mediums, need to be similar in order not to influence the results of the experiment. The first factor is the subjects' knowledge of the two diagram types. Use case modeling and feature diagrams are part of the standard course outline for a Software Requirements course in which all subjects who participated in the experiment were enrolled. Use case and feature diagrams are allotted the same lecture time, including in-class exercises and assessment units. Therefore, the subjects are expected to have received an equal level of training with the two diagram types. A subject who may have acquired additional training, for example through online courses, may perform better than other subjects but no subjects have indicated that they have received such extra-curricular training. As mentioned previously, the subjects were divided amongst the corresponding groups based on their overall academic standings and their performances in assessments related to the two diagram types, in an effort to equate the ability levels of the groups. The diagrams used within a given experiment part were also of similar size in order to prevent the relative complexity of the diagrams from affecting the results of the experiment. As for the actual presentation mediums, the subjects are very familiar with the software engineering laboratories and using the computers in the laboratories as part of their course studies. The subjects are also equally acquainted with considering software diagrams on paper during various types of in-class assessments such as quizzes, exams and exercises.

A standard 2×2 factorial design was specifically used to mitigate against order and learning effects. However, order and learning effects cannot be practically fully eliminated. Bias was not considered to have been an issue in this experiment. The author of this paper is a software modeling researcher and maybe biased towards advocating software modeling in general. However, the author gains no benefit from the results favoring either presentation medium and no stake in either presentation medium.

Domain familiarity is an aspect that may have influenced the results of the experiment in case the students are significantly more familiar with one domain over the other. The domains selected are ubiquitous and available to the public and hence it is not believed that the subjects would be significantly more familiar with one domain over the other. The qualitative results did not reveal that domain familiarity was an issue during the experiment.

Ideally one would rather only record the time required to read the diagram but practically there is no effective means to stop the timer while the subjects record their answers. For a particular question as shown in "Appendix 3", a subject can go back-and-forth repeatedly between the model presented and the question to complete answering it. However, to reduce the time it takes a subject to actually record their answer, the subjects were provided the list of possible answers as checkboxes, which is much quicker for them to simply click rather than having them write the answers by hand. The added time of a click is considered to be minimal.

5 Conclusion and future work

There is a constantly increasing appreciation of the role of software modeling in industry, evident by a constantly

increasing application of software modeling. Software modeling can be used in different phases in the software development lifecycle [8]. Software modeling can also be used for different purposes within a project, in particular, to analyze, communicate and collaborate. Communication and collaboration are at the heart of software modeling and its success is dependent on the success of its two main actors: the modeler and the model reader. Models are being presented on various presentation mediums for different purposes. This paper reports on an experiment that investigates the effect that presenting software models on two popular mediums (computer screen and paper) has on cognitive effectiveness, i.e. from the perspective of model readers. Upon completing the experiment, there is strong evidence that indicate that when model readers view diagrams on screen they are able to read the diagrams quicker. The evidence also suggests that viewing diagrams on screen induces fewer reading errors. However, further empirical studies would be required to support the latter hypothesis. The evidence also suggests that as diagrams get larger in terms of dimensions and the quantity of graphical constructs, the gap between the efficacy of viewing on screen and paper becomes larger, in favor of screen viewing. The qualitative analysis support the main findings related to the response times and reading errors committed variables as subjects cited that the physical attributes of on screen viewing were more favorable to them. Despite the superiority of on screen viewing with respect to cognitive effectiveness, the subjects were evenly divided with regards to their general preference. The data indicates that there is advantages and disadvantages with regards to the physical attributes and characteristics of viewing on the two mediums. While, on screen viewing allowed the subjects to *perform* better in their experimental tasks, the subjects found that paper viewing is more comfortable. An interesting conjecture to investigate in the future is the effect of *comfort* on subjects performing modeling activities.

Section 4.4.1 outlines a list of contexts that would be considered unsafe to generalize the results of this experiment within. This list is also a guide for future work. For example, whiteboards are very popular in workplaces and classrooms, and they are commonly used for communication and collaboration purposes. Would on screen viewing still be better than viewing on a whiteboard? According to Badreddin et al. [8], diagrams that are created for the purpose of communication and collaboration are often hand-sketched. It would also be interesting to investigate the effect of reading hand-sketched diagrams in comparison with diagrams that are created using software tools. This experiment used use case and feature diagrams and as such it is also interesting to compare the effect of reading on screen in comparison with on paper in case other types of diagrams were used, such as UML class, activity or statechart diagrams.

One very interesting conjecture to investigate is the difference between reading on paper in comparison with on screen in case the diagrams contained colors, other than black and white. Color is the most cognitively visual variable as the human visual system is highly sensitive to variations in color and can accurately and quickly distinguish between them [47, 48]. The human mind can detect differences in color three times faster than shape which makes color more easily remembered [49, 50]. Color has a strong role in object recognition [51], and becomes more effective in object recognition when shape is not diagnostic [52], which is most often the case in software engineering notations. The RGB color model (the color model humans see when viewing actual physical objects) is converted to an XYZ color model on screen and a CMYK color model on paper. Would the existence of colors in the diagrams and the use of different color models affect reading from the two mediums differently?

Systems have multiple perspectives and a multitude of diagrams types are created to model these perspectives. This means that the development team would require to consider multiple diagrams of similar or different types in order to reason about a system appropriately. Navigation using software modeling tools can be conveniently done via hyperlinking and using other automated features, whereas navigation on paper can only be done physically. As such, future work can be directed towards addressing the effectiveness of reading a collection of diagrams on paper in comparison with on screen.

Appendix 1: Post-experiment questionnaire

1. Which reading presentation medium did you prefer?
☐ Reading on paper ☐ Reading on computer screen

2. Which reading presentation medium did you find easier (quicker with less reading errors) to read?
☐ Reading on paper ☐ Reading on computer screen

3. For the presentation medium you preferred to use, explain what did you like and dislike about it and about using it:

I liked:

……

……

I disliked:

……

……

4. For the presentation medium you did not prefer to use, explain what did you like and dislike about it and about using it:

I liked:

……

……

I disliked:

……

……

5. Did you feel any timing pressures, fatigue or boredom?
☐Yes ☐No

6. Any additional comments about your experience in the experiment:

……

……

Appendix 2: Qualitative data and codebook

This appendix presents the set of codes used (Table 14) and the participants' responses to open ended questions in the post-experiment questionnaire.

Table 14 Codebook

Good codes
Code-1g Brighter
Code-2g Graphical constructs popping
Code-3g Looking up
Code-4g Less stress on eyes
Code-5g General preference
Code-6g More appropriate for big diagrams
Code-7g Comfort
Code-8g Quicker to locate graphical constructs
Code-9g Easier to locate graphical symbols
Code-10g Better focus
Code-11g Convenient
Code-12g Improved contrast display
Code-13g More appropriate to view on medium
Bad codes
Code-1b Dim
Code-2b Looking down
Code-3b Glare is tiring
Code-4b Higher workload and tiredness
Code-5b Accessibility issue of no printer available
Code-6b Harder to navigate
Neutral codes
Code-1n Exam reading
Code-2n No difference
Code-3n Normally prints everything on paper
Code-4n Expected norm as software engineers
Code-5n Generally used to medium

Responses to open-ended questions

1. The screen is just brighter. (Code-1g[Screen])
2. The screens pops the diagrams whereas the paper has a dimming effect. (Code-2g[Screen], Code-1b[Paper])
3. I felt that reading on paper was similar to being in an exam. (Code-1n[Neutral])
4. Not sure I see a big difference. (Code-2n[Neutral])
5. Easier to find things when looking at the diagram in front of me on the screen rather than looking down on paper. (Code-9g[Screen], Code-3g[Screen], Code-2b[Paper])
6. Reading on paper was less stressful on my eyes. (Code-4g[Paper])
7. I liked it more on screen. (Code-5g[Screen])
8. I normally print things so I preferred the diagram on paper. (Code-3n[Neutral])
9. I think software diagrams are more appropriate to be viewed on a computer screen. (Code-13g[Screen])
10. I was more comfortable viewing the diagrams on paper. (Code-7g[Paper])
11. I was able to find things quicker on paper. (Code-8g[Paper])
12. The screen was right up there so I could find things easier. (Code-3g[Screen], Code-9g[Screen])
13. I can find things easier on the screen. (Code-9g[Screen])
14. Can't say which one I preferred more but as software engineers I think we should get used to seeing models on a screen. (Code-2n[Neutral], Code-4n[Neutral])
15. Reading on paper is more comfortable. (Code-7g[Paper])
16. Mostly they are both the same. (Code-1n[Neutral])
17. Paper is just not a lit up as a screen. (Code-1b[Paper])
18. It helps me focus more and that is why I think I see things quicker on screen but I think if I have to look at larger diagrams for longer, then my eyes would be more comfortable reading from paper. (Code-10g[Screen], Code-8g[Screen], Code-6g[Screen], Code-7g[Paper])
19. The same for both. (Code-2n[Neutral])
20. The glare from the screen was tiring after focusing for a long time. (Code-3b[Screen])
21. It was just easier to find things on the screen. (Code-9g[Screen])
22. Paper is more comfortable on the eyes. (Code-7g[Paper])
23. The brighter light of the screens made my spot things easier. (Code-1g[Screen], Code-9g[Screen])
24. I preferred reading on paper. (Code-5g[Paper])
25. I think it was more convenient when I was reading on screen. (Code-11g[Screen])
26. My eyes got tired from looking at the screen. (Code-4b[Screen])
27. The screen exercise put a higher workload on my eyes. (Code-4b[Screen])
28. Can't really say I felt much difference but I think I would rather view on screen if the diagrams were bigger. (Code-2n[Neutral], Code-6g[Screen])
29. Better reading on paper. I like printing things than reading them on screen. (Code-11g[Paper], Code-3n[Neutral])
30. The screen made the symbols and lines pop and become easier to find. (Code-2g[Screen])
31. I liked the screen as it was more convenient. (Code-11g[Screen])
32. Seeing the relationships on screen was easier. (Code-9g[Screen])
33. I don't have a printer so I am used to reading things on screen. (Code-5n[Neutral], Code-5b[Paper])

34. Same experience. (Code-2n[Neutral])
35. I think for black and white diagrams the screen helps with the contrast. (Code-12g[Screen])
36. Definitely on screen is better. (Code-5g[Screen])
37. If the diagrams were bigger then I am not sure paper would be easy to navigate unless it was posted on a wall or something. (Code-6g[Screen], Code-6b[Paper])

Appendix 3: Experiment questionnaires

Below are the questionnaires used in the two experiment parts. Note that the appendix only shows the questions. All possible answers were provided to the subjects as checkboxes to click. The use of checkboxes reduces the time the subject needs to record the answer and therefore the time recorded represents only what was required by the subject in the to perform model comprehension.

Questionnaire for the use case diagrams experiment part

- Q1. Identify all include relationships.
- Q2. Identify all extend relationships.
- Q3. Identify all abstract use cases without extension points
- Q4. Identify all abstract use cases with extension points
- Q5. Identify all use cases with extension points
- Q6. Identify all use cases presented as classifiers (shown as anything else other than an oval)
- Q7. Identify all use cases with extension points

Questionnaire for the feature diagrams experiment part

- Q1. Identify all the Features that are Mandatory
- Q2. Identify all the Features that are Optional
- Q3. Identify all the Attribute Features
- Q4. Identify all the Dead Features
- Q5. Identify all the Refer Features
- Q6. Identify all the Require relationships
- Q7. Identify all the Exclude relationships
- Q8. Identify all the Implementation relationships
- Q9. Identify all the Generalization relationships
- Q10. Identify all the Features that have a specified Feature Cardinality
- Q11. Identify all the Features that have a specified Group Cardinality

Appendix 4: Descriptive statistics

	Medium	*n*	Min	1st quartile	Median	95% CI	3rd quartile	Max	IQR
Use case diagrams Time	Paper	36	416	592.9	807.5	652–869	915.4	988	322.5
	Screen	36	144	335.3	419.0	348–684	826.8	1041	491.4
Use case diagrams errors	Paper	36	0	2.0	6.0	3–7	8.6	12	6.6
	Screen	36	0	2.0	3.0	2–4	5.0	19	3.0
Feature diagrams time	Paper	36	1215	2000.6	2242.5	2089–2497	2731.8	4807	731.3
	Screen	36	328	505.5	617.5	561–700	735.2	2400	229.7
Feature diagrams errors	Paper	36	0	4.0	6.5	5–10	12.1	18	8.1
	Screen	36	0	0.0	2.0	1–2	2.0	5	2.0

References

1. Hitchman, S.: The details of conceptual modelling notations are important-a comparison of relationship normative language. Commun. Assoc. Inf. Syst. **9**(1), 10 (2002)
2. Irani, P., Ware, C.: Diagramming information structures using 3D perceptual primitives. ACM Trans. Comput. Hum. Interact. (TOCHI) **10**(1), 1–19 (2003)
3. Braude, E.J.: Software Design: From Programming to Architecture. Wiley, Hoboken (2004)
4. Moody, D.L.: The 'physics' of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Trans. Softw. Eng. **35**(6), 756–779 (2009)
5. Green, T.R.: Cognitive dimensions of notations. People and computers V, pp 443–460 (1989)

6. Green, T.R.G., Petre, M.: Usability analysis of visual programming environments: a 'cognitive dimensions' framework. J. Vis. Lang. Comput. **7**(2), 131–174 (1996)
7. Blackwell, A., Green, T.: Notational systems–the cognitive dimensions of notations framework. In: HCI Models, Theories, and Frameworks: Toward an Interdisciplinary Science. Morgan Kaufmann (2003)
8. Badreddin, O., Khandoker, R., Forward, A., Masmali, O., Lethbridge, T.C.: A decade of software design and modeling: a survey to uncover trends of the practice. In: Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, pp. 245–255 (2018)
9. Wright, P., Lickorish, A.: Proof-reading texts on screen and paper. Behav. Inf. Technol. **2**(3), 227–235 (1983)
10. Mangen, A., Walgermo, B.R., Brønnick, K.: Reading linear texts on paper versus computer screen: effects on reading comprehension. Int. J. Educ. Res. **58**, 61–68 (2013)
11. Rasmusson, M.: Reading paper-reading screen-a comparison of reading literacy in two different modes. Nord. Stud. Educ. **35**(01), 3–19 (2015)
12. Ackerman, R., Lauterman, T.: Taking reading comprehension exams on screen or on paper? A metacognitive analysis of learning texts under time pressure. Comput. Hum. Behav. **28**(5), 1816–1828 (2012)
13. Noyes, J., Garland, K., Robbins, L.: Paper-based versus computer-based assessment: is workload another test mode effect? Br. J. Educ. Technol. **35**(1), 111–113 (2004)
14. Dündar, H., Akçayır, M.: Tablet vs. paper: the effect on learners' reading performance. Int. Electron. J. Elem. Educ. **4**(3), 441–450 (2012)
15. Margolin, S.J., Driscoll, C., Toland, M.J., Kegler, J.L.: E-readers, computer screens, or paper: does reading comprehension change across media platforms? Appl. Cogn. Psychol. **27**(4), 512–519 (2013)
16. Spencer, C.: Research on learners' preferences for reading from a printed text or from a computer screen. J. Distance Educ. **21**(1), 33–50 (2006)
17. Köpper, M., Mayr, S., Buchner, A.: Reading from computer screen versus reading from paper: does it still make a difference? Ergonomics **59**(5), 615–632 (2016)
18. Blehm, C., Vishnu, S., Khattak, A., Mitra, S., Yee, R.W.: Computer vision syndrome: a review. Surv. Ophthalmol. **50**(3), 253–262 (2005)
19. Kong, Y., Seo, Y.S., Zhai, L.: Comparison of reading performance on screen and on paper: a meta-analysis. Comput. Educ. **123**, 138–149 (2018)
20. Mayer, R.E., Moreno, R.: Nine ways to reduce cognitive load in multimedia learning. Educ. Psychol. **38**(1), 43–52 (2003)
21. Bertin, J.: Semiology of Graphics: Diagrams, Networks, Maps. University of Wisconsin Press, Madison (1983)
22. Kresser, C.: How artificial light is wrecking your sleep, and what to do about it. February, 22, 2013 (2013)
23. Fucci, D., Scanniello, G., Romano, S., Juristo, N.: Need for sleep: the impact of a night of sleep deprivation on novice developers' performance. IEEE Trans. Softw. Eng. **46**(1), 1–19 (2018)
24. Kretzschmar, F., Pleimling, D., Hosemann, J., Füssel, S., Bornkessel-Schlesewsky, I., Schlesewsky, M.: Subjective impressions do not mirror online reading effort: concurrent EEG-eyetracking evidence from the reading of books and digital media. PLoS ONE **8**(2), 6178 (2013)
25. Lauterman, T., Ackerman, R.: Overcoming screen inferiority in learning and calibration. Comput. Hum. Behav. **35**, 455–463 (2014)
26. Evans Data Corporation: Global developer population and demographic study 2016 V2. https://evansdata.com/reports/viewRelease.php?reportID=9. Accessed 21 Mar 2021
27. CompTIA: IT Industry Outlook 2021. https://www.comptia.org/content/research/it-industry-trends-analysis. Accessed 21 Mar 2021
28. Sternberg, R.J.: Introduction to optimizing learning in college: tips from cognitive psychology. Perspect Psychol Sci **11**(651), 10–1177 (2016)
29. OMG: Unified modeling language, version 2.5.1. Object Management Group, Inc (2017). https://www.omg.org/spec/UML/2.5.1. Accessed 4 Apr 2021
30. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-oriented domain analysis (FODA) feasibility study. DTIC Document (1990)
31. Adobe. Adobe acrobat. https://acrobat.adobe.com/us/en/acrobat.html. Accessed Apr 2021
32. El-Attar, M.: Statistics package—software model comprehension: paper vs. screen. https://faculty.alfaisal.edu/sites/default/files/papervsscreendata.zip. Accessed 25 July 2021
33. Purchase, H.C., et al.: Empirical evaluation of aesthetics-based graph layout. J. Empir. Softw. Eng. **7**(3), 233–255 (2002)
34. Purchase, H.C., et al.: Comprehension of diagram syntax: an empirical study of entity relationship notations. Int. J. Hum Comput. Stud. **61**(2), 187–203 (2004)
35. Gopalakrishnan, S., et al.: Adapting UML activity diagrams for mobile work process modelling: experimental comparison of two notation alternatives. In: Third IFIP WG 8.1 Working Conference, PoEM 2010, Delft, The Netherlands, pp. 145–161 (2010)
36. Reijers, H.A., Mendling, J.: A study into the factors that influence the understandability of business process models. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **41**(3), 449–462 (2011)
37. Dougiamas, M.: Moodle. https://moodle.org/. Accessed 21 Mar 2021
38. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qual. Res. Psychol. **3**(2), 77–101 (2006)
39. Shapiro, S.S., Wilk, M.B.: An analysis of variance test for the exponential distribution. Techno Metrics **14**, 355–370 (1972)
40. Spearman, C.: The proof and measurement of association between two things (1961)
41. Cliff, N.: Dominance statistics: ordinal analyses to answer ordinal questions. Psychol. Bull. **114**(3), 494–509 (1993)
42. Cliff, N.: Answering ordinal questions with ordinal data using ordinal statistics. Multivar. Behav. Res. **31**(3), 331–350 (1996)
43. Cliff, N.: Ordinal Methods for Behavioral Data Analysis. Lawrence Erlbaum Associates, Mahwah (1996)
44. Romano, J., Kromrey, J.D., Coraggio, J., Skowronek, J.: Appropriate statistics for ordinal level data: should we really be using t-test and Cohen's d for evaluating group differences on the NSSE and other surveys. In annual meeting of the Florida Association of Institutional Research, vol. 177 (2006)
45. Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: Experimentation in Software Engineering: An Introduction. Kluwer (2000)
46. El-Attar, M.: A comparative study of students and professionals in syntactical model comprehension experiments. Softw. Syst. Model. **18**(6), 3283–3329 (2019)
47. Mackinlay, J.: Automating the design of graphical presentations of relational information. ACM Trans. Graph. **5**(2), 110–141 (1986)
48. Winn, W.D.: An account of how readers search for information in diagrams. Contemp. Educ. Psychol. **18**, 162–185 (1993)
49. Lohse, G.L.: A cognitive model for understanding graphical perception. Hum.-Comput. Interact. **8**(4), 353–388 (1993)
50. Treisman, A.: Perceptual grouping and attention in visual search for features and for objects. J. Exp. Psychol. Hum. Percept. Perform. **8**, 194–214 (1982)
51. Bramão, I., Reis, A., Petersson, K.M., Faísca, L.: The role of color information on object recognition: a review and meta-analysis. Acta Physiol. (Oxf) **138**(1), 244–253 (2011)

52. Wurm, L.H., Legge, G.E., Isenberg, L.M., Luebker, A.: Color improves object recognition in normal and low vision. J. Exp. Psychol. Hum. Percept. Perform. **19**, 899–911 (1993)

Mohamed El-Attar completed his B.Eng. Computer Systems degree from Carleton University, Ottawa, Ontario, Canada in 2003. He then completed his Ph.D. in Software Engineering from the University of Alberta, Edmonton, Alberta, Canada in 2009. Dr. El-Attar worked in industry in Canada and USA as a software tester and front-end developer, respectively. After a 1.5 years of industrial experience, Dr. Mohamed El-Attar started his academic career at King Fahd University of Petroleum and Minerals (KFUPM) in Dhahran, KSA, as an Assistant Professor. He was promoted at KFUPM to Associate Professor in only 4 years. After 6 years at KFUPM, Dr. El-Attar worked 1 year at Mississippi State University, USA, followed by one more year at the University of Ontario Institute of Technology, Canada. Dr. El-Attar, then returned to KSA to work at Alfaisal University where he is now the Chair of the Software Engineering department. Dr. Mohamed El-Attar many undergraduate and graduate courses at various institutions across multiple continents. He supervised a number of graduate students. His research focuses on the two main areas of software modeling and requirements engineering. In particular, his research is concerned with, UML, especially use case models, model consistency assurance, model transformation, human aspects in software modeling and secure software engineering. His research has been published in the most prestigious software engineering journals and conferences such as: IEEE Transactions on Software Engineering, Empirical Software Engineering, Information and Software Technology, Systems and Software, Software and Systems Modeling, Secure Software Engineering and Requirements Engineering.