

Improving the Requirements Inspection Abilities of Computer Science Students through Analysis of their Reading and Learning Styles

Mr. Anurag Goswami, North Dakota State University

Anurag Goswami is a Ph. D. Candidate in the department of Computer Science at North Dakota State University. His main research interests include empirical software engineering, human factors in software engineering, and software quality. He is a member of the IEEE Computer Society.

Dr. Gursimran Singh Walia, North Dakota State University

Gursimran S. Walia is an associate professor of Computer Science at North Dakota State University. His main research interests include empirical software engineering, software engineering education, human factors in software engineering, and software quality. He is a member of the IEEE Computer Society. Contact him at gursimran.walia@ndsu.edu

Mr. Ganesh Padmanabhan, North Dakota State University

I am a research-oriented applications developer with more than ten years of professional programming experience primarily in the area of cognitive and vision research. I have been working at the Center for Visual and Cognitive Neuroscience at North Dakota State University since 2005. I have expertise in mathematics, software development, and hardware and software, trouble-shooting. I have had experience with computer programming from the time when FORTRAN was popular as a language and personal computers were just being introduced. As an undergraduate, programming was a tool to augment my interests in biology and psychology. In graduate school, I had the opportunity to hone my skills, and formalize my computer science and neuroscience education. Since then, I have had experience implementing experiments involving virtual immersion, EEG recording, gaze tracking, image processing, statistical analyses, and various types of computer modelling.

Dr. Mark E McCourt, Department of Psychology, North Dakota State University

DR. MARK E. McCOURT, JAMES A. MEIER and DALE HOGOBOOM DISTINGUISHED PROFESSOR, NORTH DAKOTA STATE UNIVERSITY DEPARTMENT OF PSYCHOLOGY, DIRECTOR, NDSU NIH IDeA PROGRAM CENTER FOR VISUAL AND COGNITIVE NEUROSCIENCE

Mark E. McCourt earned his Ph.D. in Psychology from the University of California, Santa Barbara in 1982. At NDSU he was awarded the James A. Meier and Dale Hogoboom Professorships in 2004 and 2009, respectively. In 2004 he was awarded an NIH/IDeA Centers of Biomedical Research Excellence (COBRE) grant which established the NDSU Center for Visual and Cognitive Neuroscience, which is funded through 2021 and which he currently directs. Dr. McCourt is internationally known for his scientific contributions in the area of human perception and cognition, particularly in the areas of brightness/lightness perception, spatial attention, and multisensory integration. He has published over 100 peer-reviewed papers and given numerous contributed and invited talks. He is a member of the Editorial Board for the international journals *NeuroReport* and *Vision*, and is an Associate Editor for the journal *Frontiers in Human Neuroscience*. Dr. McCourt is a regular reviewer for over 50 scientific journals, and has reviewed for major funding agencies such as NIH, NSF, AFOSR, the Netherlands Organization for Scientific Research, the US-Israel Bi-National Science Foundation, the Canada Research Chairs Program, the Canada National Sciences and Engineering Council, and the Wellcome Trust. Dr. McCourt has received over \$31M in competitive grant funding over his career from NIH, NSF, AFOSR, and other sources.

Improving the Requirements Inspection Abilities of Computer Science Students through Analysis of their Reading and Learning Styles

1. Introduction

Due to the complex nature of software development process, there is an increasing demand for skilled software engineers that is expected to grow more in future [1]. Students are expected to graduate with the necessary skillset for pursuing their careers in software industry. However, research [2] reports that students in academic settings mostly work on small scale projects and lack an exposure to industrial strength artifacts as well as team-based activities that are expected of them when beginning their jobs in software industry [3]. Therefore, there is a need for educators to train students on skills essential for their jobs in software industry. One such skill deals with their abilities to review industrial strength software artifacts (e.g., requirements and design documents) to find and fix faults early in the development process.

On that end, *Software Requirements Specification* (SRS) is considered as one of the most critical phases wherein requirements of the software system under development are gathered from technical and non-technical stakeholders are translated into *Natural Language* (NL) document (i.e. SRS document) [4]. Faults manifest during SRS development due to the inherent nature of NL (i.e. ambiguity, imprecision, and vagueness). These faults, if left undetected can propagate to the later stages where they are harder to find and fix [5].

Therefore, to train students to be able to deliver quality software product on time and to help them understand the nature of faults committed during the software development, we simulated the way industries trains their newly hired employees on a widely used software verification technique known as *inspections* [6]. During the inspection, skilled inspectors review a software artifact to detect and report faults which can then be fixed by the document author. As an example, Microsoft routinely train their new developers on inspections to educate them about the benefits of software inspections as well as to teach them on the nature of faults committed during the development of software artifacts at Microsoft. While educators [7], [8] have tried to emulate the inspection trainings; students report large number of false positives and the efforts to understand the factors (e.g., educational background, experience, comprehension skills) that impact their inspection output have met with limited success.

Our research is novel as it tries to improve students' inspection performance by understanding the way an individual perceive and process the information. This preference of individuals to process, retain, and recall information is known as *Learning Styles* (LS) [9]. Research [10] in psychology showed that each individual vary in their LS and understand information better if it is presented in their LS (e.g., understand better by diagrams/flowcharts rather than reading/verbal communication). Since SRS documents are written in a standard NL fashion and may not be conducive all type of LS preferences. We also utilize eye tracking technique to understand eye movements of students as they perform inspections of externally developed requirements document. Our research on eye tracking builds on previous work done in areas of comprehension [11], UML class diagrams [12], computer interface evaluation [13].

Motivated from these factors, this research reports results from an academic inspection training study with an objective to understand the relationship of eye movement and LS data vs. inspection output that in turn can be utilized by educators to facilitate inspection training. The improved inspection training will help students to acquire inspection skills which in turn lead to

reduction in skill gap between academia and industry. Thirty-nine graduate and undergraduate students went through the inspection process and individually inspected two different documents using fault checklist technique. We analyzed the relationship between inspection performance and LSs along with eye movement data by taking individual as well as average data for all students. The results showed that eye movements are significantly correlated with the number of faults detected in general as well as for certain LSs. We also provide insights on how educators can utilize this information (eye movements and cognitive preference of students) to design their inspection training that would enable the detection of more number of faults.

2. Background

This section describes the background on software inspections, LS, and eye-tracking. Section 2.1 describes the inspections process. Section 2.2 describes LS dimensions and instrument used to measure LS. Section 2.3 details eye-tracking concept and terms used in the study.

2.1 Inspections

The concept of inspection was introduced by Fagan [14] where the inspection leader chooses a team of skilled individuals from pool of inspectors who will perform the inspection. Then, the team-members individually review a software work-product to identify faults which is returned to the document/code author who then fix these faults.

Due to the importance of inspections in software industry, academicians have used variety of techniques to train their students in inspections ranging from giving hands on experience [15] to using web based systems [16] for saving in class training time. Research in academia also reported results with an objective to improve their training process like: comparing tool based and paper based inspections [17], teaching inspection via active learning [7]. While students learn how to write software, they don't get enough training on how to read someone else's work product and how to find problems in an externally produced artifact. This research is trying to address this limitation.

2.2 Learning Styles and Index of Learning Styles

LSs was first introduced by Kolb [18]. Since then, multiple versions of LS models [19] were developed by the psychologists and validated its use in academic environment [10]. This study utilized Felder Silverman Learning Style model [20] and an instrument known as *Index of Learning Styles* (ILS) that is used to measure LS of individuals [21]. The ILS is an online questionnaire (empirically validated for its reliability and construct validity [22]) that consists of 44 questions where each dimension has 11 questions. A brief description of four LS dimensions is described in Figure 1(a). The LS score of an individual across four dimensions is denoted by 'X' on the top of a category as shown in Figure 1(b). A score between 5-7 and 9-11 states that a person has a moderate and strong preference towards a category in a dimension.

Concept of LS had been used widely in academia to find out the teaching strategies required to improve students' score [9], using tools to enhance their collaborative learning [23], using LS of faculty members to understand changes in utilizing technology [24]. In this study, LS instrument is being used to understand how LSs can impact the inspection output and what can be done to improve the inspection output of subjects with varying LSs.

		Sensing				Intuitive			
		<ul style="list-style-type: none"> •Facts •Concrete content and data •Careful with details •Follow existing ways 				<ul style="list-style-type: none"> • Abstract • Conceptual and innovative • Oriented towards theories and meaning • Discovering possibilities 			
PERCEIVING	Visual				Verbal				
	<ul style="list-style-type: none"> • Visual representations. • Pictures, diagrams, flow charts, time line • Video, demonstration 				<ul style="list-style-type: none"> • Written and spoken explanations 				
PROCESSING	Active				Reflective				
	<ul style="list-style-type: none"> • Trying things out • Working in groups • Discussing, explaining • Brainstorming 				<ul style="list-style-type: none"> • Thinking things first • Working alone • Writing summaries 				
	Sequential				Global				
	<ul style="list-style-type: none"> • Linear, orderly • Learn in small logical steps 				<ul style="list-style-type: none"> • Holistic • Context and relevance of the subject • Learn in large jumps 				

Figure 1(a). Felder Silverman Learning Style Model

ACT									X			R
	11	9	7	5	3	1	1	3	5	7	9	11
						<--	-->					
SEN									X			I
	11	9	7	5	3	1	1	3	5	7	9	11
						<--	-->					
VIS					X							V
	11	9	7	5	3	1	1	3	5	7	9	11
						<--	-->					
SEQ		X										G
	11	9	7	5	3	1	1	3	5	7	9	11
						<--	-->					

Figure 1(b). Sample ILS score of an individual

2.3 Eye Tracking

Following are the eye tracking terms utilized in this study:

- *Fixation*: is a point where eyes are relatively stationary and an individual is taking in the information.
- *Saccade*: Quick eye movement between fixations.
- *Scanpaths*: are complete saccade-fixation-saccade sequence and interconnecting saccades.
- *Gaze*: is the sum of fixations durations in an area. They are also known as “dwell”, “fixation cluster”, or “fixation cycle”.
- *Region of Interest (ROI)*: is an analysis method where eye movements that fall under certain area is evaluated (ROI in this study is the area where fault exist in the document).

Eye movement system is the result from Javal’s gaze motion research in 1879. The system used set of mirrors to observe the eye movement of participants while reading [25]. The results showed that people tend to incorporate fixations and saccades instead of reading in a linear fashion. Modern eye tracking system works by reflecting infra-red light on an eye, and recording the reflection pattern. Early research [26] in eye tracking showed that, people tend to incorporate regressive *fixations* and *saccades* (instead of reading in a linear fashion) when faced with comprehension difficulty to review their understanding and retention.

These eye movement factors represents the amount of cognitive processing involved by an individual [27]. Cognitive psychologists used eye tracking technology [28], [29] to understand Visual/Verbal and Sequential/Global LS preference of individuals by displaying information on a computer monitor. The results showed that visual learners tend to focus at the pictures whereas, sequential learners read sentences, took more time to read the information, and had less vertical eye movements. This study utilized eye movements to understand the reading patterns of students as they review requirements document. We hypothesize that eye tracking in conjunction

with LSs can provide insights into factors that could be manipulated to design effective inspection training modules thereby enhancing students' skills by increase in inspection output.

3. Experiment Design

To understand the relation between eye tracking with LSs to gain insights into factors to design effective inspection training, we analyzed the eye movement of students during the requirements inspection and measured their LSs along with their inspection output (# of faults found and time spent) to understand whether certain eye movements and LSs are better suited for software reviews. We also wanted to determine the ways to facilitate the inspection training and review process. As a side benefit, performing inspection would also help students understand fault prone areas of requirement documents that can be carried over to their future career in industry.

3.1 Research Questions (RQs)

The following research questions were investigated in this study:

Research Question 1: Do SE students vary in their eye movements during the requirements inspection?

Research Question 2: What eye movement factor(s) most positively impact the fault detection ability of software inspectors?

Research Question 3: How does eye movements and LSs correlate with inspection output?

3.2 Participating Subjects

Thirteen (13) graduate students and twenty-six (26) undergraduate students at North Dakota State University (NDSU) participated in the study. Undergraduate students were enrolled in System Analysis and Design course and graduate students were enrolled in Requirements Engineering course. Both the courses focus specifically on learning how to perform inspections (of requirements and design documents) and understand its impact on overall software quality.

3.3 Artifact

Participants used two different requirements documents for inspection as described in Table 1. Both the documents were developed in plain English externally and utilized in several studies as well as by Microsoft to train their newly hired employees on the inspection process [30], [31].

Table 1: Software requirements documents used

SRS	Developer	Purpose	Length	Faults
Loan Arranger System (LAS)	Microsoft professionals	In-class inspection training	11 pages	30
Parking Garage Control System (PGCS)	University of Maryland	Inspection in eye tracking settings	14 pages	34

3.4 Eye Tracking Instrument

To record eye movements of students during the inspection, a non-invasive EyeLink 1000 desktop mount from SR research that sits at the bottom of the viewing area was used (Figure 2) in this research. It records eye movements with a sample frequency of 250-2000Hz along with tracking range of 32° x 25°. It has accuracy greater than 0.5 degree and a resolution less than 0.01 degree. The EyeLink 1000 desktop mount consists of three main components: high speed camera, infrared illuminator, and a host PC to record eye movement data captured from camera.



Figure 2. Inspection task in eye-tracking laboratory

3.5 Experiment Procedure

The experiment was conducted in a series of steps described below:

Step 1 – Collecting Learning Styles: Students from both the courses went through a LS questionnaire (can be accessed at: <https://www.engr.ncsu.edu/learningstyles/ilsweb.html> [21]) that has 44 multiple choice questions. The result of this survey is LS score across one category in each dimension as shown in Figure 1(b).

Step 2a – Inspection Training: Students from both the courses were trained on how to detect and report faults using fault-checklist technique. Examples were provided by the instructor to guide the inspection process. The training was held during a classroom session and lasted for 70 minutes.

Step 2b – Inspection and Reflection of LAS SRS: During this step, each participant read LAS document to perform individual inspection and reported faults. The faults were reported in the fault list and false positives were marked by one of the researcher. Fault forms were returned back to the students along with the seeded fault list to perform reflection on their inspection performance. Students were asked to read through each fault description to comment on whether they saw (but did not reported) or missed faults during inspection.

Step 3 – Inspecting PGCS requirements with Eye tracker: After a week (to avoid fatigue effect), each participant inspected PGCS document in eye-tracking laboratory as shown in Figure 2. One of the researchers was present in the eye-tracking laboratory to assist participants during the inspection by: a) adjusting or re-calibrating the eye-tracker; b) start/stop or pause/resume the inspection. Each page of PGCS requirements document was displayed on a computer monitor at a resolution of 1080×1920 . During the inspection, participants used left and right click of the mouse to move between the pages. The eye-tracker at the bottom of the monitor recorded eye movements throughout the inspection task along 'x' and 'y' axis. To report faults without disrupting the eye tracking, participants were asked to talk-it-out-loud. A voice recorder was used to assist fault reporting where participant mentioned the line number where they found the fault and explain why it represents a problem. To avoid fatigue effect during the inspection task, participants were allowed to take breaks without changing the distance of their chair (i.e. moving) from the eye tracker. At the end of inspection, fault recordings were transcribed into a fault list along with the timing data (i.e. start and stop time, breaks, time when each fault was found). The entire process resulted in 39 fault lists (one per student).

4. Data Collection

This section details the raw and calculated eye tracking data that was collected during the inspection of PGCS document. The eye movement data was used as an input in EyeMMV tool (<https://github.com/krasvas/EyeMMV>) that was utilized to identify *fixations*, *saccades*, and analyze areas where faults were present (i.e. *ROI*). Twenty-seven places were identified in the PGCS document where fault exists and were marked as ROI's with the help of a software tool known as IrfanView (<http://www.irfanview.com>): that calculates coordinates in the form of X_{start} , Y_{start} and X_{end} , Y_{end} axis of marked region automatically. EyeMMV tool extracted the gaze data of each participant and removed any noise (i.e. eye blinks where no data was recorded) present. The filtered data was then analyzed as the relationship between inspection effectiveness and eye movements of participants during inspection. For each participant, the eye movement data from each page was collected and are described below in detail:

- T_{total} : each participant spent some time in milliseconds (ms) on each page during the inspection. Hence, 14 durations for each page (converted into seconds) were calculated.
- $T_{fixation}$: During the inspection, participant focused on certain areas of document (where eyes are relatively stationary) that involves cognitive processing to detect faults. These led to generation of *fixations* (Figure 3B) and time spent for fixating each page was calculated.
- S_{linear} : To find reading style (i.e. linear/random) of inspectors, we calculated *scanpaths* (Figure 3A). Only first occurrence of reading was considered for calculating scanpaths because participant tend to search for information while reading back which may not involve their usual way of reading.
- F_{ROI} : total number of fixations by the participant at the region where faults exist in the PGCS document.
- T_{ROI} : total time taken (in seconds) by the participant to read through the ROI's during inspection.

Mean value of each variable was calculated to analyze the overall eye movement of a participant.

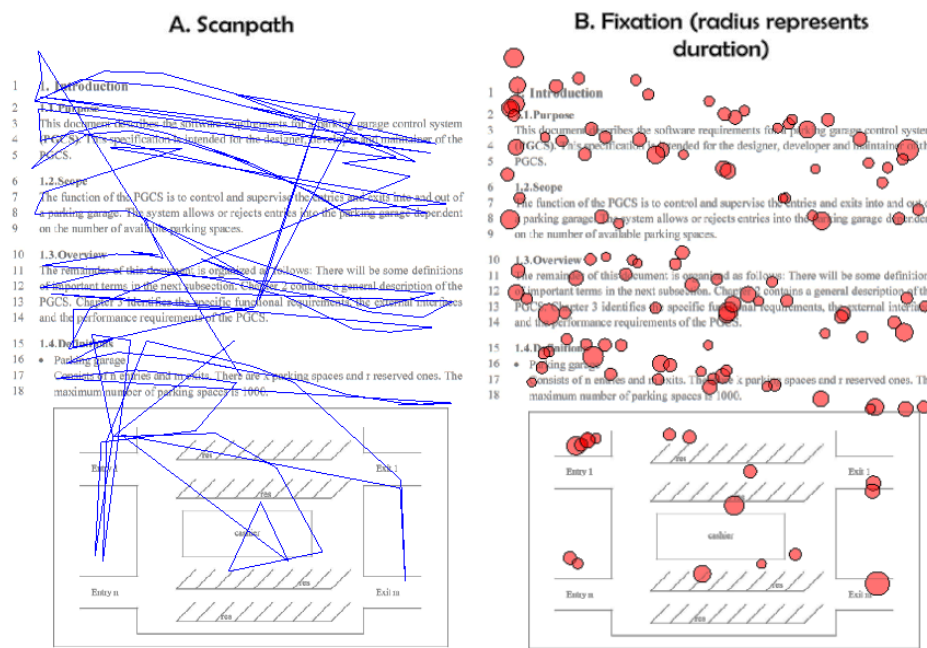


Figure 3. Sample Scanpaths and Fixations

5. Results

This section analyzes the impact of eye movement and LSs of inspectors on the inspection effectiveness (# of faults) and efficiency (faults/hour).

5.1 Analysis of Eye Tracking Data

This section presents the general trends and variation in the eye movements (via eye tracking data) across undergraduate and graduate students and their correlation to inspection effectiveness (# of faults found) and efficiency (fault rate).

To provide an overview of the results; Table 2 compares the average eye tracking data (first 7 rows) and inspection output (last 2 rows) for undergraduate and graduate students. The data is average for each page (and for each ROI). As mentioned in Section 4, S_{linear} is the percentage of linear eye movement of participant while inspecting requirements document and calculated as the ratio of linear saccades to total saccades in each page. To normalize the comparison, S_{linear} data for each participant was averaged for each page. Similarly, percentage fixation is the ratio of fixations at ROI to the total fixations on a page and percentage duration is the ratio of duration at ROI to total duration in each page.

Table 2. Eye tracking and Inspection Data

	Undergraduate	Graduate
T_{total} (Sec)	179.29	204.78
T_{fixation} (Sec)	126.48	152.46
S_{linear}	82.52%	81.96%
F_{ROI}	1301.85	1348.46
T_{ROI}	413.03	452.18
% Fixation per ROI	11.90%	11.50%
% Duration per ROI	11.97%	11.61%
TotalFaults	4.81	4.62
Efficiency (faults/hour)	7.42	6.38

Major insights from Table 2 are discussed below:

- Generally, both undergraduate and graduate students demonstrated linear reading pattern (S_{linear} values of 82.52% and 81.96% respectively). This trend of linear scanpath was consistent across individual subjects. This means, during inspection of NL requirements, students tend to read requirements in a linear fashion irrespective of their LS preference to comprehend the information presented.
- Contrary to our expectations, undergraduate students (as opposed to graduate students) had higher average linear saccade, percentage fixations per ROI, percentage duration per ROI. Consequently, undergraduate students demonstrated higher fault detection effectiveness and efficiency (not by a big margin though). These results hold true with the previous study at Microsoft [32] where higher level of technical education (i.e. Bachelors vs. Masters vs. Doctorate) was inversely correlated with inspection performance. Hence, it is necessary to train students with higher technical knowledge on reading requirement document from a customer's perspective as not bog down with more design details and try to fixate on fault prone areas of software artifacts so that their effort is well spent.

5.2 Eye Movement vs. Inspection Analysis

To quantify the impact of eye tracking ($T_{fixation}$, T_{total} , S_{linear} , F_{ROI} , T_{ROI}) on inspection output (fault count and fault rate), multiple regression analysis was performed to evaluate which factors

significantly impact inspection output. The results for both undergraduate and graduate students was combined for this analysis and reported in Table 3. Major observations follow:

Table 3. Students vs. eye movement

		T_{fixation}	T_{total}	S_{linear}	F_{ROI}	T_{ROI}
Effectiveness	Correlation	0.083	0.18	-0.162	0.294	0.193
	p-value	0.308	0.137	0.163	0.035	0.12
Efficiency	Correlation	-0.361	-0.314	-0.104	-0.104	-0.209
	p-value	0.012	0.026	0.264	0.263	0.101

- Out of five, four eye movement variables (T_{fixation} , T_{total} , F_{ROI} , T_{ROI}) were positively correlated to inspection effectiveness but only F_{ROI} was significantly correlated. That is, participants who were able to find inconsistencies (spend more time) at ROI's were in turn able to detect and report larger number of faults. Hence, educators should try to train students so that they are able to comprehend requirements faster, are able to identify fault prone areas which would in turn lead to an increase in number of faults reported during inspection.
- While all five eye movement factors were negatively correlated to inspection efficiency, T_{fixation} and T_{total} had a significant negative correlation. That means, fault detection rate will reduce if more time spent fixating at the region where no fault exists. Hence, it is necessary for students to spend time to fixate at ROI rather than spending time to fixate across entire document to increase inspection efficiency. Therefore, students' are required to be trained on how to identify the fault prone requirements and focus extensively on those requirements to enhance their inspection output.
- It was observed by one of the researchers that participant first read the document to comprehend the information presented and re-visit pages to search and verify the information or to understand it again for detecting faults. This results in increase of inspection time and reduced inspection efficiency. To reduce this search overhead, we can use this information to rearrange the requirements content so it is easy to recall supporting information (e.g., assumptions and dependencies) when reviewing functional or system specific information.

5.3 Eye Tracking and LSs vs. Inspection Effectiveness

For eye movement (i.e. linear vs. random), raw LS data revealed that, *Sequential* and *Global* LSs were fairly distributed among all participants. The linear trend might be due to the fact that the PGCS document was written as paragraphs in NL which led participants to review it in a linear fashion despite of their LS preference. We also wanted to investigate whether eye movement of individuals of certain LSs favors inspection effectiveness positively as compared to other LSs. This was done to understand if requirements documents need to be written in a way that is conducive to LSs of students that may enable higher comprehension of information which in turn will improve their performance.

LS preference of an individual is formed with the combination of categories across four dimensions (e.g., ACT-SEN-VIS-SEQ). In our raw data, out of 39 members, only three had preferences towards *Verbal* LS and therefore, *Visual vs. Verbal* dimension was removed and remaining categories were analyzed (*Active*-ACT, *Reflective*-REF, *Sensing*-SEN, *Intuitive*-INT, *Sequential*-SEQ, and *Global*-GLO). Participants were grouped into their respective LS clusters using six LS categories across three dimensions. Due to less number of participants both undergraduate and graduate student data was combined to form clusters. The final LS clusters

were: *ACT-INT-GLO* (four), *ACT-SEN-GLO* (nine), *ACT-SEN-SEQ* (five), *REF-SEN-GLO* (nine), *REF-SEN-SEQ* (six), *ACT-INT-SEQ* (two), *REF-INT-GLO* (two), and *REF-INT-SEQ* (two). We performed multiple regression test to evaluate the correlation of eye movement factors of each LS vs. inspection output. Out of eight LS clusters, *ACT-INT-SEQ*, *REF-INT-GLO*, and *REF-INT-SEQ* were removed from analysis (since each had only two participants) and might deviate the results. The results appear in Table 4 and discussed below:

Table 4. Eye movement vs performance of different LSs

LS			T_{fixation}	T_{total}	S_{linear}	F_{ROI}	T_{ROI}
ACT-INT-GLO	Effectiveness	Correlation	0.802	0.91	0.252	0.883	0.876
		p-value	0.099	0.045	0.374	0.059	0.062
	Efficiency	Correlation	0.495	0.667	0.505	0.651	0.639
		p-value	0.253	0.166	0.247	0.175	0.181
ACT-SEN-GLO	Effectiveness	Correlation	-0.571	-0.34	-0.333	-0.689	-0.787
		p-value	0.054	0.185	0.191	0.02	0.006
	Efficiency	Correlation	-0.720	-0.623	-0.244	-0.930	-0.940
		p-value	0.014	0.036	0.264	0.001	0.001
ACT-SEN-SEQ	Effectiveness	Correlation	-0.284	-0.25	-0.1	0.543	0.37
		p-value	0.322	0.342	0.437	0.172	0.27
	Efficiency	Correlation	-0.600	-0.580	-0.418	0.218	0.033
		p-value	0.142	0.153	0.242	0.362	0.479
REF-SEN-GLO	Effectiveness	Correlation	-0.288	-0.265	-0.346	-0.003	-0.273
		p-value	0.226	0.246	0.181	0.497	0.239
	Efficiency	Correlation	-0.261	-0.390	-0.256	0.226	-0.056
		p-value	0.249	0.150	0.253	0.279	0.443
REF-SEN-SEQ	Effectiveness	Correlation	0.753	0.76	-0.226	0.695	0.669
		p-value	0.042	0.04	0.333	0.063	0.073
	Efficiency	Correlation	0.261	0.297	-0.483	0.134	0.069
		p-value	0.309	0.284	0.166	0.400	0.448

- Participants with **REF-SEN-SEQ** LSs has the maximum number of significantly positive eye movement factors (T_{fixation} , T_{total}) with inspection effectiveness. The other two factors (F_{ROI} , T_{ROI}) are positively correlated and are close to being significant. Therefore, inspectors who tend to think about the system, follow details carefully, and perform inspection step by step in a linear fashion may find inconsistencies and hence, fixate at the fault region and find larger number of faults.
- Interestingly, participants with **ACT-INT-GLO** has all eye movement factors that are positively correlated with effectiveness and efficiency. One factor is significant (T_{total}) and two eye movement factors (F_{ROI} , T_{ROI}) that are quite close to being significant to inspection effectiveness. Hence, if inspectors who prefer to take small jumps during inspection, try to find meanings and possibilities of fault in requirements may focus on region where fault exists and can lead to higher fault detection with higher fault detection rate.
- Also, **REF-SEN-GLO** (for effectiveness) and **ACT-SEN-GLO** (for both effectiveness and efficiency) participants had all eye movement factors that were negatively correlated. But **ACT-SEN-GLO** LS has two eye movement factors (F_{ROI} , T_{ROI}) with strong negative correlation with inspection effectiveness and it has maximum (four) variables that has significant negative correlation with efficiency. This result give an insight that it will be a good practice to train students to inspect requirements documents in a step by step logical fashion else it will affect the rate by which inspectors report faults.

Although the number of participants in some LS clusters were small, these results indicate that during inspection, inspectors who think about the requirements of the system presented first and then proceed step by step by following fault checklist process can focus more at the fault region (i.e. ROI) and hence, detect and report significantly large number of faults. Results also reveal that; ACT-SEN-GLO LS has the most significant negative impact on inspection performance.

6. Discussion

The objective of our study was to investigate the impact of individual eye movement and LS factors to improve student's understanding of requirements inspection. The results of the analysis can be utilized by academicians to train students to reduce the necessary skill gap between academia and industry by helping students to acquire the required inspection skills. Based on the results, following is the brief discussion for each research question:

Research Question 1: Do SE students vary in their eye movements during the requirements inspection?

To find whether reading trend of students vary based on their level of education, we compared the eye movements and inspection performance of undergraduate vs. graduate students (Table 2). During the comparison, it was found that both undergraduate and graduate students tend to read requirements document in a linear fashion. Among both, undergraduate students spend more time fixating at the fault region and had higher inspection effectiveness and efficiency as compared to graduate students. Generally, fixations were scattered across entire document. Therefore, educators should train students on reading requirements and develop inspection training in a manner that enables students to comprehend information easily to reduce inspection time and increase their tendency to focus at the area where faults may exist. Also, students with higher technical knowledge should inspect requirements from a customer's perspective (what system is supposed to do?) and not as a developer (how system is supposed to achieve functional requirements?). Developing inspection training that enables students to detect large number of faults with reduced inspection time would result in high cost savings to software industry in terms of saving re-work effort and time.

Research Question 2: What eye movement factor(s) most positively impact the fault detection ability of software inspectors?

It is necessary for educators to design high quality inspection training for students that can reduce inspection effort and increases inspection outcome. Hence, we compared different eye movement factors (Table 3) that can help educators in inspection training design. While comparing eye movement factors, it was found that inspectors who focus at the fault region significantly (i.e. find inconsistencies at ROI) found higher number of faults and should avoid spending more time at other regions of requirements document that can affect inspection fault rate negatively. This result could help educators to train their students to avoid spending unnecessary time looking for faults once they understood the requirements they are read.

Research Question 3: How does eye movements and LSs correlate with inspection output?

We also wanted to investigate whether students' LS preference with their eye movement data have an impact of inspection outcome. Students were grouped into their respective LS cluster (e.g., **Reflective-Sensing-Sequential** or **REF-SEN-SEQ**) and inspection effectiveness as well as efficiency were compared for each LS cluster against their eye movement factors (Table 4). Results show that, students with different LS preference had different eye movement factors and

that affects inspection outcome. Results also revealed that students with certain LSs (i.e. **REF-SEN-SEQ**) spent significant amount of time fixating on the document and spent more time fixating at the fault region which favors inspection effectiveness positively.

From the results, it is suggested that educators should train students to understand as well as to adopt qualities of LSs (during inspection) that have a positive impact on inspection performance. Hence, students should be trained towards inspecting requirements document in a logical fashion (i.e. step by step and avoiding random jumps), thinking about the details of system in the document (to avoid any assumption or omission of details), and focusing more at the fault region by avoiding unnecessary time spent on detecting faults. Also, **Active-Intuitive-Global** or **ACT-INT-GLO** and **REF-SEN-SEQ** had eye movement factors (except S_{linear}) positively correlated with inspection efficiency but the correlation was not significant. This introduces a thought that; apart from students' understanding, inspection may also rely more on the way requirements documents are written that leads to fixations and quick fault detection. Hence, educators can use the relationship between eye movement and LSs vs inspection performance to improve their training by teaching students to follow inspection process and focus more at areas where faults usually manifest (i.e. ROI's) in requirements document. This reduces the effort spent and lead to detection of high number of faults with higher pace. We plan to evaluate this aspect in future studies in hope of training students better in software inspections.

7. Conclusion and Future Work

Based on the results in this study, LS and eye tracking factors of students do have an impact on inspection outcome. The results show that, LS and eye movement factors can be utilized by academicians to train students to acquire the required inspection skills. This leads to reduction in re-work effort and time for software industry and reduces skill gap between industry and academia. The results indicated need for a training technique that could be utilized by educators to train students to focus more on fault prone areas. Results also gave insights for the need of a writing technique of requirements where faults could be detected easily. While this study reports the results in context of software requirements, it can be used to train students in other necessary software skills (development of requirements/design document, writing quality code) as well as to find a way to improve their performance. These results motivate us for further investigation where future work would include higher number of participants to find effect of LS of students (that were not included in this study) on inspection performance to enhance their training. Another future work includes investigating effect of certain LS categories (e.g., *Sensing vs. Intuitive*) may have on students' performance during requirements inspections.

References

- [1] T. Samson, "Demand for software engineers keeps climbing -- and so do the salaries," *InfoWorld*, 2012.
- [2] C. B. Simmons and L. L. Simmons, "Gaps in the computer science curriculum: an exploratory study of industry professionals," *J. Comput. Sci. Coll.*, vol. 25, no. 5, pp. 60–65, 2010.
- [3] A. Radermacher, G. Walia, and D. Knudson, "Missed Expectations: Where CS Students Fall Short in the Software Industry," *CrossTalk Mag.-J. Def. Softw. Eng.*, no. Jan/Feb, pp. 4–8, 2015.
- [4] A. Goswami and G. Walia, "An empirical study of the effect of learning styles on the faults found during the software requirements inspection," in *In Proceedings of the 24th IEEE International Symposium on Software Reliability Engineering (ISSRE)*, 2013, pp. 330–339.
- [5] B. Boehm and V. R. Basili, "Software Defect Reduction Top 10 List," *Computer (Long. Beach. Calif.)*, vol. 34, pp. 135–137, 2001.
- [6] E. P. Doolan, "Experience with Fagan's inspection method," *Softw. Pract. Exp.*, vol. 22, no. 2, pp. 173–182, 1992.

- [7] W. W. Schilling, "Teaching software inspection effectiveness: An active learning exercise," in *2012 Frontiers in Education Conference Proceedings*, 2012, pp. 1–6.
- [8] A. Goswami and G. Walia, "Teaching Software Inspections to Software Engineering Students through Practical Training and Reflection," in *123rd Annual Society of Engineering Education Conference on Computer Education*, 2016, vol. 2016–June.
- [9] C. Tulbure, "Do different learning styles require differentiated teaching strategies?," in *Procedia - Social and Behavioral Sciences*, 2011, vol. 11, pp. 155–159.
- [10] R. Felder and L. Silverman, "Learning and teaching styles in engineering education," *Eng. Educ.*, vol. 78, no. June, pp. 674–681, 1988.
- [11] R. Bednarik and M. Tukiainen, "An eye-tracking methodology for characterizing program comprehension processes," *Proc. 2006 Symp. Eye Track. Res. Appl. - ETRA '06*, p. 125, 2006.
- [12] S. Yusuf, H. Kagdi, and J. I. Maletic, "Assessing the comprehension of UML class diagrams via eye tracking," in *IEEE International Conference on Program Comprehension*, 2007, pp. 113–122.
- [13] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: Methods and constructs," *Int. J. Ind. Ergon.*, vol. 24, no. 6, pp. 631–645, 1999.
- [14] M. E. Fagan, "Design and code inspections to reduce errors in program development," *IBM Syst. J.*, vol. 15, no. 3, pp. 182–211, 1976.
- [15] J. C. Carver, L. Jaccheri, S. Morasca, and F. Shull, "A checklist for integrating student empirical studies with research and teaching goals," *Empir. Softw. Eng.*, vol. 15, no. 1, pp. 35–59, 2010.
- [16] A. Hazeyama and A. Nakano, "Web-based inspection process support environment for software engineering education," in *Proceedings of the Fourth ICSE Workshop on Software Engineering over the Internet*, 2001.
- [17] F. Macdonald and J. Miller, "A comparison of tool-based and paper-based software inspection," *Empir. Softw. Eng.*, vol. 3, no. 3, pp. 233–253, 1998.
- [18] D. Kolb, "Experiential learning: Experience as the source of learning and development," *J. Organ. Behav.*, vol. 8, no. 4, pp. 359–360, 1984.
- [19] R. J. Charkins, D. M. O'Toole, and J. N. Wetzel, "Linking teacher and student learning styles with student achievement and attitudes," *J. Econ. Educ.*, vol. 16, no. 2, pp. 111–120, 1985.
- [20] R. M. Felder, "ARE LEARNING STYLES INVALID? (HINT: NO!)," *On-Course Newsletter*, Sept. 27, 2010, 2010.
- [21] B. A. Solomon and R. M. Felder, "Index of learning styles," *Raleigh, NC North Carolina State Univ. Available online*, 1999.
- [22] R. M. Felder and J. Spurlin, "Applications, Reliability and Validity of the Index of Learning Styles," *Int. J. Eng. Educ.*, vol. 21, no. 1, pp. 103–112, 2005.
- [23] C.-J. Huang, J.-J. Liao, H.-Y. Shen, N. N. Aye, Y.-W. Wang, H.-X. Chen, D.-X. Yang, Y.-C. Luo, and Y.-T. Chuang, "Using learning style-based diagnosis tool to enhance collaborative learning in an undergraduate engineering curriculum," *Comput. Appl. Eng. Educ.*, vol. 19, no. 4, pp. 739–746, 2011.
- [24] S. Matrosova Khalil, "From resistance to acceptance and use of technology in academia," *Open Prax.*, vol. 5, no. 2, pp. 151–163, 2013.
- [25] M. A. Tinker, "A photographic study of eye movements in reading formulae.," *Genet. Psychol. Monogr.*, 1928.
- [26] A. L. Yarbus, "Eye movements and vision," *Neuropsychologia*, vol. 6, no. 4, p. 222, 1967.
- [27] M. Just and P. Carpenter, "A theory of reading: from eye fixations to comprehension.," *Psychol. Rev.*, vol. 87, no. 4, pp. 329–354, 1980.
- [28] T. J. Mehigan and I. Pitt, "Detecting Learning Style through Biometric Technology for Mobile GBL," *Int. J. Game-Based Learn.*, vol. 2, no. 2, pp. 55–74, 2012.
- [29] J. Cao and A. Nishihara, "Understand learning style by eye tracking in slide video learning," *J. Educ. Multimed. Hypermedia*, vol. 21, no. 4, pp. 335–358, 2012.
- [30] J. Carver, F. Shull, and V. Basili, "Observational studies to accelerate process experience in classroom studies: An evaluation," in *Proceedings - 2003 International Symposium on Empirical Software Engineering, ISESE 2003*, 2003, pp. 72–79.
- [31] F. Shull, J. Carver, and G. H. Travassos, "An empirical methodology for introducing software processes," *ACM SIGSOFT Softw. Eng. Notes*, vol. 26, no. 5, p. 288, 2001.
- [32] J. Carver, N. Nagappan, and A. Page, "The impact of educational background on the effectiveness of requirements inspections: An empirical study," *IEEE Trans. Softw. Eng.*, vol. 34, no. 6, pp. 800–812, 2008.