Using Eye Tracking to Investigate Reading Patterns and Learning Styles of Software Requirement Inspectors to Enhance Inspection Team Outcome

Anurag Goswami, Gursimran Walia
Department of Computer Science
North Dakota State University
Fargo, USA
anurag.goswami@ndsu.edu,
gursimran.walia@ndsu.edu

Mark McCourt, Ganesh Padmanabhan
Department of Psychology
North Dakota State University
Fargo, USA
mark.mccourt@ndsu.edu,
ganesh.padmanabhan@ndsu.edu

ABSTRACT

Background – Inspecting requirements and design artifacts to find faults saves rework effort significantly. While inspections are effective, their overall team performance rely on inspectors' ability to detect and report faults. Our previous research showed that individual inspectors have varying LSs (i.e., they vary in their ability to process information recorded in requirements document). To extend the results of our previous LS research, this paper utilizes the concept of eye tracking (to record eye movements of inspectors) along with their LSs to detect reading patterns of inspectors during requirements inspections. **Aim** – The objective of this research is to analyze the reading trends of effective and efficient inspectors using eye movement and LS data of individual inspectors and virtual inspection teams. **Method** - The current research uses data (LS, eye tracking, and inspection) from thirteen inspectors to find its impact on inspection effectiveness and efficiency. **Results** - Results from this study show that, inspectors who detect more faults during inspection, focus significantly more at the fault region to find and report faults as opposed to comprehending requirements information. Results also showed Inspection teams with diverse inspectors outperform similar teams and spend more time in comprehending information at the fault region. Additionally, results showed that inspectors with SEQ LS significantly tends to focus more at fault locations and are preferred for inspection. **Conclusion** – These results can aid the selection of inspectors during the inspection process thus improving software quality

CCS Concepts

• **Software and its engineering→Software creation and management** • **Software verification and validation→ Empirical software validation.**

Keywords

Requirements; software inspection; learning style, eye tracking.

1. INTRODUCTION

Software industries use inspection techniques to find and fix faults committed during the early stages of software lifecycle (where they are easiest and cheapest to detect). This results in significant cost

saving and significant improvements in software quality [38]. To have most impact on software quality, inspections are performed during the requirement stage (wherein functional and non-functional requirements are gathered from both technical and non-technical stakeholders). The requirements development process is especially prone to errors as it uses *Natural Language* (NL) to document requirements, and is prone to ambiguity, impreciseness, and vagueness due to the inherent nature of NL [7].

Among different techniques (e.g., NL to State transitions [1], checklist based inspections [36], scenario based reading [43], ad hoc inspections [40]) used to detect NL faults in software requirements, inspections (introduced by Fagan [16]) is recognized as the most effective verification technique. Inspection is an empirically validated and widely recognized technique to find and fix faults in various software artifacts (requirement, design, code, interfaces) [14]. The technique involves review of a software artifact by a team of skilled inspectors who detect and report faults. Major inspections steps are described below:

- *Planning*: inspection manager selects inspection team from the pool of inspectors.
- *Preparation*: inspectors are given a brief overview of the artifact under inspection and they review the artifact to detect and record faults found during the inspection.
- *Meeting*: individual inspectors meet in team meeting phase where they discuss and consolidate a master fault list;
- *Rework*: the faults are handed back to the author of the artifact to fix.

Since inspection was introduced, various versions of inspections concept [31, 37] emphasize different steps of inspection process. However, evidence shows that, *preparation* (i.e., individual ability of inspectors) significantly impacts the overall inspection team performance [39]. To that end, researchers have tried to understand the impact of individual factors on inspection performance by experimenting with *educational background* and *level of technical degree* [9, 11]. However, empirical results at major software organizations showed that non-technical inspectors detected significantly larger number of faults recorded in NL requirements documents as compared to the people with higher technical degrees [2, 9]. These results led us to hypothesize that inspector's ability to detect faults during inspection may be affected by the ways they *perceive* and *process* information known as *Learning Styles* (LS).

Cognitive psychologists have studied and experimented with LSs to evaluate learning strengths of individuals [3, 34] and concluded that, individuals vary in the ways they perceive and process recorded information. For example, some individuals tend to work in groups and are involved in active discussions to resolve a problem while some prefer to think about the information first and

work alone. Research results from cognitive psychology that uses LSs has successfully crossed over to academia where LSs of students are utilized to improve their course performance [18, 19]. Research showed that an individual will learn better if information is presented in the preferred LS [3]. Academia have also experimented to create diverse project teams [41] by using personality inventory [27]. The idea behind was to create diverse personality teams where group contribution is higher than the sum of individual contributions. Software engineering researchers have also borrowed similar concept to create heterogeneous inspection teams to improve performance [33] using Myers-Briggs Type Indicator (MBTI) instrument. The results from this research was not promising as MBTI is a personality inventory [35] as compared to LSs which measures information comprehension strengths of individuals. The only research that used LSs in SE domain suggested that LSs could be used to select tools for gathering requirements (in a geographical distributed team) from different stakeholders (with varying LS's) during the elicitation process [4].

Motivated from the results above, we hypothesized that LSs could be used to create inspection teams with inspectors of diverse LSs to improve their inspection performance. To gain initial evidence, we performed multiple empirical studies with students [22, 24] and software professionals [23] to investigate impact of their LSs on inspection performances. The results from prior investigation verified our hypothesis that, using LS disparity to create inspection teams resulted in improved inspection team performance. Motivated by these results, we believe that high performing inspectors and inspection teams (who detect more number of faults in less time) have certain LSs and that, they tend to read inspection document in a certain fashion, to comprehend information depending on their LS preference.

Researchers in software engineering domain have characterized eye movements of software engineers during program comprehension [6], source code reviews [46], UML class diagrams [47], computer interface evaluation [21], user behavior in www search [24] to understand the reading patterns in the past. Some researchers have also used LSs to understand the reading patterns of individuals with different LSs preferences [8, 32]. Inspired by the previous work, the current research investigates the relationship between eye movement of inspectors during inspection and their inspection performance. An eye-tracking device was used to record eye movements of participants as they inspect a NL requirements document. During the experiment, the participants reported their LSs and performed an individual inspection of a requirements document (on a computer monitor) in an eye tracking laboratory settings. We analyzed the effect of LSs by measuring eye movement data of inspectors belonging to different LS groups with respect to their inspection effectiveness and efficiency. The results show that eye movement are significantly correlated with the inspection outcome in general and more positively correlated for inspectors with certain LS preferences.

2. BACKGROUND

This section details the background on learning style model and eye-tracking used in this research. Section 2.1 describes the Felder Silverman Learning Style model and its dimensions. Section 2.2 elaborates the instrument used to measure the LS a. Section 2.3 describes the eye-tracking terms and studies that used LS of individuals to track eye movements.
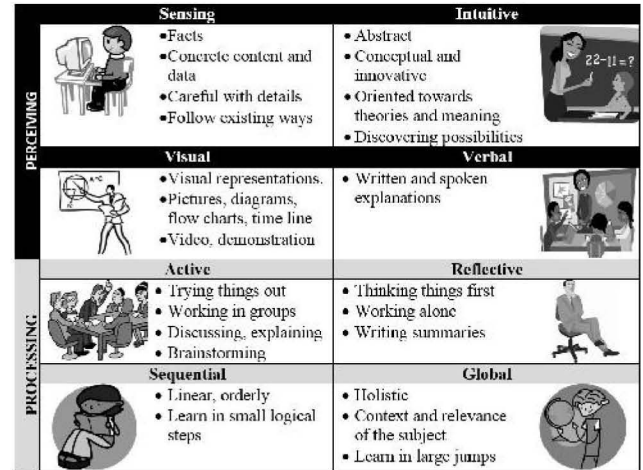

Figure 1. Felder Silverman Learning Style Model

2.1 Felder Silverman Learning Style Model (FSLSM)

Kolb introduced the concept of LSs [28] and developed the first LS instrument to measure individual ability and preference to perceive and process information. Since LS model was introduced, psychologists have developed various versions of LS models [13] and validated its use in education [19]. This research uses the most advanced model (FSLSM) to measure characteristic strengths and preferences of an individual [17]. FSLSM has four dimensions out of which Sensing/Intuitive and Visual/Verbal relates to perceiving part and remaining two dimensions (i.e.) Active/Reflective and Sequential/Global relates to the processing part. Brief description of LS dimensions is shown in Figure 1.

2.2 Index of Learning Styles (ILS)

The ILS (Figure 2) is an instrument used to measure the LS of an individual has been empirically validated for its reliability and construct validity [20]. ILS is an online questionnaire that contains 44 questions. Each LS dimension has 11 questions resulting in 44 questions across four LS dimensions. For example, in Visual/Verbal dimension, out of 11 questions, if a person selects 9 answers that are in favor of visual and 2 answers in favor of verbal then the LS score in that dimension will be 9-2=7 towards visual category. The final score in each dimension is denoted by a symbol 'X' on the top of the score. An example output of ILS is shown in Figure 2. ILS score ranging from 1-3 represents a balanced person towards both the categories in a dimension. A score between 5-7 and 9-11 states that a person has a moderate and strong preference towards a category in a dimension.
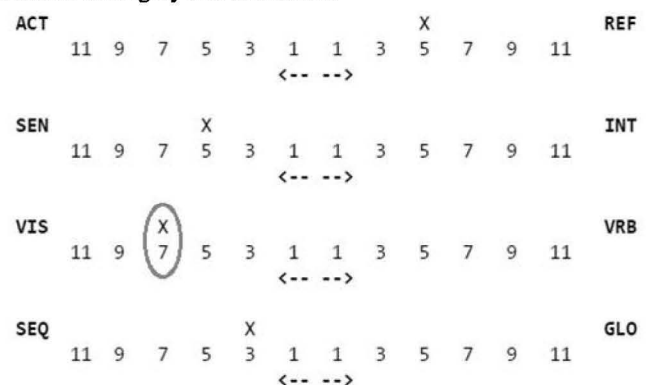

Figure 2. Example result of the questionnaire on the ILS

2.3 Eye Tracking

Eye movements and pattern of eye movements refers to the amount of cognitive processing by an individual. This system emerged from the results of Javal's gaze motion research conducted in 1879 that used mirrors to observe the eye movements of subjects while reading [45]. Javal noticed that people do not read in a linear fashion. Instead, they incorporate fixations and saccades. Eye tracking works by reflecting infra-red light onto an eye, and recording the reflection pattern with a sensor system. Following are the eye tracking terms used in this research:

- *Fixation:* is a point where eyes are relatively stationary and an individual in 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.

These eye movements represent amount of cognitive processing involved by an individual [26]. Eye tracking technology is widely used in domains ranging from market research [12] to evaluation of user interfaces [25] to common interactive tasks [48]. Research in cognitive psychology [8, 33] also used eye tracking to understand visual/verbal and sequential/global learning preference of individuals by generating fixations on an information displayed on computer monitor. The results from study showed that visual learners have more fixations at area where images are displayed on the screen and verbal learners have more fixations where text was written on the screen. Also, sequential learners showed less vertical eye movement as compared to global learners. We used eye-tracking in this research to investigate how inspectors comprehend (i.e. their focus of attention on different parts and reading approach of requirement document) information recorded in NL requirements document and how does that correlates with their LS and inspection performance?

3. EXPERIMENT DESIGN

To evaluate the relationship of reading patterns and LSs of requirements inspectors on their inspection performance, a study was conducted in Computer Science course where LS of participants were gathered via online questionnaire. Participants were then trained on fault-checklist based inspection process. Participants then individually performed requirements inspection in an eye-tracking laboratory settings. Their raw LSs, eye-tracking data along x & y axis, and timestamp and fault data (i.e. number of faults found and time taken) was collected to analyze the reading patterns of individual inspectors and virtual inspection teams of different LSs using EyeMMV toolbox [30]. EyeMMV is a complete utility for post-experiment eye movement analysis for generating *fixation, scanpaths,* and *heatmaps*. This toolbox was used to visualize *fixations* (i.e. certain area where inspectors focused for a larger amount of time during inspection), *scanpaths* (i.e. what is the path of reading requirements document during inspection), and *heatmaps* (i.e. areas, represented by different colors, in the document which gained more attention of an inspector during the inspection process).

Research Questions (RQ): Three RQ's were used in this study:

RQ1: Is overall inspection performance affected by the way inspectors read requirements document?

RQ2: Does inspection teams, ranging from dissimilar to similar, based on LSs have a particular reading pattern that impacts their inspection performance?
RQ3: Does inspectors belonging to a particular LS category have a reading pattern which supports inspection outcome positively?

Participating Subjects: Thirteen graduate students enrolled in Requirements Engineering course at North Dakota State University (NDSU) participated in this study. Requirements Engineering course covers the understanding of requirements development technique and various phases (e.g., elicitation, analysis, specification, validation) which includes requirements inspection technique. The course required students to learn software inspections and their impact on software quality improvement.

Artifact: Two requirements documents developed externally, Loan Arranger System (LAS) and Parking Garage System (PGCS), were used by the participants during the training and actual inspection. LAS is responsible for grouping loans into bundles based on user specified characteristics and then sell to other financial institutions. PGCS is responsible automated parking management with the entry and exit of vehicles with the use of access card/daily tickets. Both the documents were developed by Microsoft developers and were documented in plain English. The LAS document was 11 pages long seeded with 30 realistic faults and PGCS document was 14 pages long and was seeded with 34 faults. The faults were seeded by the Microsoft researchers to represent realistic faults. Both the documents have been used previously in inspection studies [10, 42] and consists of almost similar fault density (LAS: 2.72 faults per page and PGCS: 2.42 faults per page).

Eye Tracking Apparatus: EyeLink 1000 desktop mount from SR research was used to track eye movement of participants. It is a non-invasive system that sits at the bottom of computer monitor/projection area (i.e. below the tracked area the participant is viewing) which allows free movement of participant. The eye tracker, shown in Figure 3, consists of: (a) high-speed camera, (b) infrared illuminator, and a host PC (connected via Ethernet) dedicated to process camera data. It allows eye movement recordings with a sample frequency of 250-2000 Hz. It has a tracking range of $32°$ x $25°$ with an accuracy greater than 0.5 degree and a resolution less than 0.01 degree.

Experiment Procedure: Steps followed during the course of the study are listed below:

Step 1 - Learning Styles questionnaire survey: at the beginning of the experiments, all participants were handed out Felder Silverman's LS questionnaire (could be accessed at: https://www.engr.ncsu.edu/learningstyles/ilsweb.html).
Participants answered all 44 multiple choice questions and, the LS results are generated for each participant on ILS scale. For each dimension on ILS (Active/Reflective, Sensing/Intuitive, Visual/Verbal, and Sequential/Global), the participant has score towards one category. Hence, only four LS categories (from each dimension) form LS of an individual with a score of either 1 or 3 or 5 or 7 or 9 or 11.

Step 2 - Training on Inspecting SRS for Faults: Participants were instructed in-class by the instructor on how to use fault-checklist method to detect faults in requirements document. Examples were provided to the students which guides them on how to detect and report faults using sample examples. The training duration was of 70 minutes. To ensure that students understood the fault inspection, as part of their course objectives, students applied fault-checklist
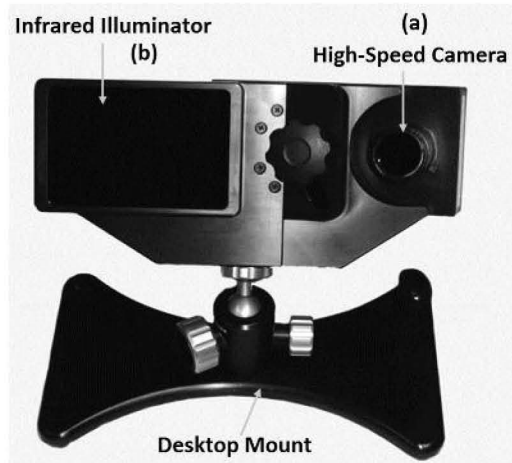
Figure 3. EyeLink 1000 desktop mount

Figure 4. Inspection task in eye-tracking laboratory

technique on LAS document (Step 2a) followed by a reflection of their inspection results (Step 2b).

Step 2a – Inspection of LAS document: during this step, subjects used their training to perform an individual inspection of LAS document and reported faults.

Step 2b – Reflection of LAS inspection results: One of the researcher evaluated the faults reported by each participant and provided them feedback about true and false positives. Next, post inspection reflection was performed wherein, participants were provided a list of original 30 faults in LAS document (that they had inspected) and were asked to reflect upon the faults they saw (but did not reported) or missed during the inspection by comparing it against their reviewed fault list. Table 1 shows a sample of reflection document and each column is described as follows:

- *Defect#:* represents the defect ID in seeded fault list.
- *Req.#:* indicates the requirement ID(s) where fault is present.
- *Type:* denotes categorization of faults into different fault categories. For example, ambiguity (A) in the requirements.
- *Description:* brief description of the problem for an author to be able to understand and fix it.
- *Is it a defect:* Whether students to agree or disagree that the fault represents an actual requirement problem?
- *Did you see this:* Whether they were saw this fault ('yes' or 'no') during the inspection?
- *Did you report this:* Whether ('yes' or 'no') they reported this fault during inspection of LAS document?
- *Explain:* this field allowed a brief explanation if their response in the three earlier fields were inconsistent.

Post Reflection, students discussed their doubts regarding inspections and reflection of their faults with researchers. A week after this exercise (to avoid *fatigue* effect), subjects were provided with the quick recap of fault checklist based inspection technique.

Step 3 - Inspecting PGCS requirements via Eye tracker - Next, each participant performed an individual inspection in eye-tracking laboratory as shown in Figure 4. Each participant individually read through the PGCS document on a computer monitor (rotated in portrait mode as requirements were documented in portrait mode). Throughout the process, EyeLink 1000 eye tracker sitting at the bottom tracked their eye movements during the inspection. One of the researchers was present in the eye-tracking laboratory that assisted the participants during the inspection process. At the beginning of the experiment, researcher gave the overview of the PGCS SRS document to be inspected, fixed eye distance with eye-tracking instrument, calibrated (i.e. focusing at known locations on the computer screen) and validated (determining whether apparatus estimation of eye position is indeed close to the known position of the targets) eye movements, and drift correction (to correct small drifts in the calculation of gaze position). During the experiment, if participant needed break, researcher paused the inspection process and resumed it after performing drift correction.

During the initial setup, participants were aligned at the center of a computer monitor (both horizontally and vertically). The distance between eye and camera was fixed between 55cm – 60cm. This distance was chosen through pilot testing with sample participants to evaluate if they were comfortable in reading document from the monitor. Also, this distance lies in the optimum distance to capture eye movements by the eye-tracker as suggested by SR research in EyeLink 1000 manual (i.e. between 40 cm – 70 cm) [15].

Participant's eye movements were calibrated with the eye-tracker, using sampling rate of 250Hz, and then validated via EyeLink 1000 built in software. The validation checks for the eye movement readings and compares it with the calibration reading to avoid any deviation in the data. Based on the comparison, results are generated (on the eye-tracking system screen) as poor, fair, good, and excellent. If results are poor, the validation process was repeated again. Before starting the inspection process, a final drift correction of participant's eye was performed (a corrective adjustment based on raw eye position of a participant).

Each page of requirements document is displayed on the monitor at a resolution of 1080 × 1920 in portrait mode. Using right and left

Table 1. Sample of reflection form for LAS document

Defect #	Req. #	Type	Description	Is it a defect?	Did you see this?	Did you report this?	Explain.
1	1,2	A	Are the reports in these requirements the same or separate?				
2	1,2	O	When do the updates occur? Are they effective immediately?				

click on mouse, participants were able to switch pages of PGCS document forward and backward. Eye-tracker at the bottom of the monitor captures the eye movements along 'x' and 'y' axis of the monitor continuously during the entire inspection task. Inspection task also required the participants to talk-it-out-loud any faults that they discover during the inspection process so that they don't have to look away from the screen to disrupt eye-tracking. A voice recorder was used to assist the fault reporting. The participants speak out loud the place (line number) where they found fault and describe the fault to establish why it represents a problem.

During the entire process, participants were allowed to take break(s). In such case, researcher with the control of eye-tracking system paused the eye-tracking and the voice recorder. During breaks, participants were encouraged to relax their eyes, look away from monitor but were not allowed to move their chair to avoid disruption of eye calibration. Whenever participant wanted to resume, drift correction was performed again and inspection was resumed (i.e. recording of eye movement was started again) from the same page where inspection was paused. After completing inspection, recorded faults (in the form of audio) are transcribed as a fault list along with the timing data (i.e. start and finish times, time when each fault was found, breaks) for each participant.

Step 4 -Post-inspection: Researchers again provided a complete list of defects in the PGCS document to the students (similar to LAS reflection) and asked them to reflect on their inspection experience. We also discussed the issues they may have faced in eye tracking environment to gain insights into their inspection results.

4. DATA COLLECTION

The eye tracking data (i.e. raw coordinates and timestamp) of each participant was used as an input into EyeMMV toolbox [29] which is an offline analysis tool. The tool runs under MATLAB environment and uses different functionalities to identify *fixations*, *saccades*, generate *heatmaps*, and analyze *ROI* (i.e. eye movements on a particular area where faults are present).

The PGCS document was marked with various ROI's where faults were present. The XY_{start} and XY_{end} coordinates of region where faults were present in the document were calculated using a software tool known as IrfanView [44]. We simply dragged the mouse pointer to select the ROI and released after selecting. IrfanView automatically records the coordinates of the region selected in a text file. ROI measure was used to analyze whether all

participants in general and participants belonging to a certain LS were able to focus at the areas where faults were present. The tool extracted the gaze data of each participant and filtered the results (e.g., eye blinks where no data was recorded). The resulting data was then used for analysis in this study.

Apart from eye movement data along x & y axis and time for each participant, other variables were calculated and saved into separate files (.mat extension) using raw eye tracking data into EyeMMV toolbox. The variables are listed below:

- T_{total}: Total time (in seconds) spent on each page by the participant while inspecting the PCGS document.
- $T_{fixation}$: Total fixation time (in seconds) by the participant on each page during the inspection.
- S_{linear}: ratio of linear reading (i.e. linear saccades) to all saccades on each page.
- F_{ROI}: total number of fixations by the participant at ROI in the PGCS document.
- T_{ROI}: total time taken (in seconds) by the participant to read through the ROI's during inspection.

As mentioned earlier, EyeMMV tool enabled us to visualize *fixations*, *scanpaths*, and *heatmaps* (See Figure 5). Fixation events were detected and visualized by using an algorithm based on spatial and temporal constraints [21]. Apart of eye movement data (i.e. x & y axis, and time), the fixation identification depended of three basic parameters: two spatial parameters (tolerance 1: *t1* and tolerance 2: *t2*) measured in pixels and one minimum duration (minDur). The algorithm suggests having *minDur* value between 100ms to 900ms. The parameter *t1* depicts how tight a fixation cluster will be and *t2* depicts the discrimination between two clusters. As inspection task requires reading of a document in which fixations are for a very small duration, we used *t1=50, t2=25, and minDur=150ms* to visualize fixations.

For visualizing heatmaps (Figure 5), EyeMMV uses a parameter of gridSize (*defined in pixels*) which is used to generated heatmap from the point data. Grid size is inversely proportional to the number of different regions generated on heatmaps and Bicubic interpolation was also used to smooth out the heatmap generated. For generating heatmaps in our experiment, we used *gridSize=135*. Heatmaps followed a color scale that helps investigator to find out the area which received attention at different levels of focus (based on different colors). The color scale in our experiment uses blue,
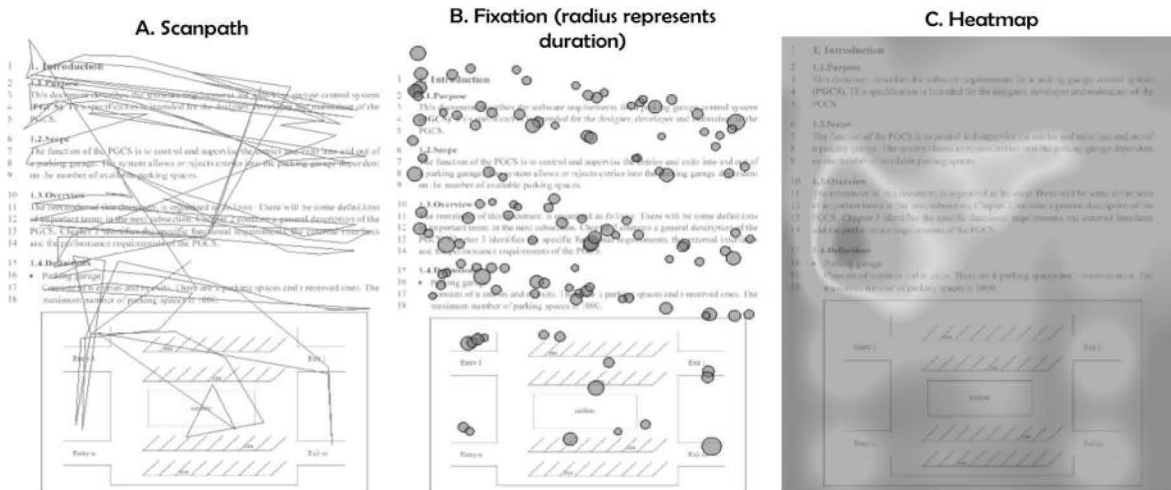
Figure 5. A sample *scanpath*, *fixation* and *heatmap*

Table 2. Sample inspection and eye tracking data of four participants

ID	Unique Faults	Efficiency	$T_{fixation}$	T_{total}	S_{linear}	F_{ROI}	T_{ROI}
1	8	13.71	111.71	148.38	80.55%	1740	494.30
2	5	6.25	93.57	207.80	76.61%	671	283.17
3	3	4.61	132.34	168.69	83.95%	612	229.51
4	5	12.5	80.33	113.45	82.44%	568	201.25

green, yellow, orange, red to represent minimum (blue) to maximum (red) region of attention.

For linear and random reading patterns (i.e. saccades), a criteria angle of 30° was used for measuring linear saccades. Hence, any saccade (i.e. rapid movement between two fixations) less than or equal to 30° was considered as forward/linear reading.

5. EVALUATION CRITERIA

This section explains the procedure used to evaluate the research questions. For each participant, raw coordinates along with timestamp (in milliseconds) for each page of PGCS SRS document was collected. Raw eye movement data (sample shown in Table 2) was written into *Export Data Format* (EDF) file by EyeLink system. To read these files into MATLAB system, they were first converted into *American Standard Code for Information Interchange* (ASCII) format. This raw data also consists of unwanted data (e.g., no eye tracking data during blinks). So, when converted into MATLAB file, these set of unwanted data was cleansed before further analysis. Next, for evaluating reading patterns, eye movement data (for each participant) from each page was collected in the form of variables described in detail below:

- During inspection, each participant spent some time reading and reporting faults on each page. The time spent on each page was calculated in milliseconds (ms). Hence, total of fourteen durations (one per page) was calculated by EyeMMV toolbox computed as Ttotal (converted in seconds for analysis).

- Also, while inspecting, participant tend to focus on certain parts in the document (where eyes are relatively stationary and coordinates are roughly the same as time increases) to detect faults which involves cognitive processing. These focus points resulted in fixations (Figure 5) and the radius of each fixation depicts the time spend on fixating. The overlapping fixations (i.e. circles) depicts the area where participant focused again while reading requirements during inspection. Due to this, some areas are more populated as compared to others. For each participant, the fixation time was calculated on each page during the inspection and referred to as Tfixation.

- We also investigated the motion (linear/random) in which participants with different LSs tend to read during inspection, referred as *scanpaths* (Figure 5). For recording scanpaths, only first occurrence of reading a page was taken into account because while reading back, inspectors tend to search with an objective to extract some information which might not involve their usual way of reading and can impact results. While converting raw eye movement data, ratio of linear movement (angle <= 30°) to the entire types of movements (all saccades) were also calculated for each page, documented as S_{linear}. Mean of data gathered from all pages for each inspector to find out the overall reading pattern of an inspector.

The requirements document was also divided into twenty-seven ROI's where faults were seeded. We wanted to evaluate the impact of various LSs on the eye movements (i.e. time taken, fixations, etc)

at regions where faults existed. Hence, different variables were computed at the ROI's and are stated below:

- Fixation count for each participant was derived at the region where fault exist (F_{ROI}) at each ROI,
- Time taken to read through each fault area (T_{ROI}) was also calculated.

Mean of these variables from each ROI was calculated which results in overall tendency to focus and time taken by each participant at regions where faults existed.

For evaluating reading patterns of various LSs, participants were divided into various possible LS clusters (e.g., ACT-SEN-VIS-GLO). Eye movement data for each participant within a cluster was combined and a mean value was calculated to represent overall reading pattern of participants as inspectors in a particular LS. To have better visualization of reading patterns of inspectors with varying LS preferences, heatmaps (Figure 5) were generated to compare regions which received different levels of attention by the inspectors belonging to different LS dimensions.

6. ANALYSIS AND RESULTS

This section analyzes the effect of LSs and eye movements of the inspectors on their performance during inspection task.

6.1 Eye Movements vs. Inspection Data

As mentioned in Section 3, participants inspected PGCS requirements and verbalized faults found while reading the document on a computer monitor. During this process, we also recorded their eye movements (to understand their reading patterns). Using the *fault and timing data* (to calculate fault count and fault rate) and *eye tracking data* (5 variables namely – Tfixation, Ttotal, Slinear, FROI, and TROI described in Section 4), we performed multiple regression with model selection (forward selection) to gain insights into the reading patterns of more *effective* (found more faults) and *efficient* (found faults faster) inspectors. Table 3 reports the results in the form of correlation and p-values between 5 eye movement variables and 2 inspection performance variables.

Major observations from Table 3 are listed below:

- While all 5 variables were positively correlated with inspection *effectiveness*, F_{ROI} and T_{ROI} were significantly correlated. That is, participants who reported more faults, were able to quickly spot ROI's and then spent more time fixating on ROI's to detect and log faults present.

- In terms of inspection *efficiency*, Tfixation and Slinear were negatively correlated. However, only Tfixation had a significantly negative correlation. That is, fixation on ROI's is more important than the total fixation (which in fact may be due to lack of information processing). Interestingly, more time spent on each page (Ttotal) led inspectors to find faults significantly faster.

- Also, subjects reading in a linear fashion, were negatively correlated with efficiency. Based on the participant behavior (observed by one of the researcher during experiment), *sequential* readers tend to read and comprehend the entire

Table 3. Eye Tracking vs. Inspection Performance

		$T_{fixation}$	T_{total}	S_{linear}	F_{ROI}	T_{ROI}
Effectiveness	Correlation	0.155	0.192	0.322	0.545	0.461
	P-value	0.307	0.265	0.142	0.027	0.056
Efficiency	Correlation	-0.542	0.530	-0.058	0.014	0.150
	p-value	0.028	0.031	0.426	0.482	0.313

$T_{fixation}$ - *Fixation Time Per Page;*
T_{total} - *Time Per Page;*
S_{linear} - *Linear Saccade Per Page (in %);*
F_{ROI} - *Total Fixations at ROI;*
T_{ROI} - *Total Duration at ROI*

information to understand the system first and then searched for faults which results in more inspection time.

Overall, effective inspectors tend to fixate more at the regions where faults exist to comprehend information instead of spending more time to fixate non ROI's on each page. This might be due to the fact that, during the inspection, effective inspectors find odds in reading NL requirements mostly at ROI's. Therefore, they fixate to find and report faults documented in the requirements at ROI's which led to significantly positive correlation between number of fixations at ROI (as opposed to the total fixation) and inspection effectiveness. Also, inspectors who spent more time fixating on each page were significantly less efficient because of the difficulty in processing requirements information. Following up on this result, in later sections, we present analysis of LSs of inspectors that exhibited such behavior (i.e., tendency to fixate on ROI's) to be make more informed decisions for selecting inspection team.
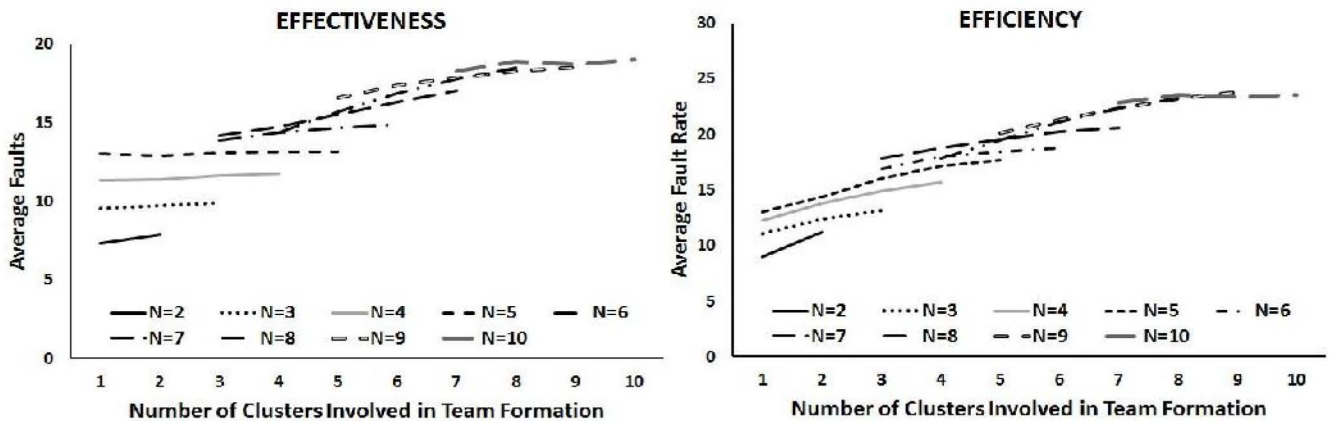
6.2 LSs vs. Inspection Team Performance

To evaluate the impact of LSs on team based performance, we created virtual inspection teams of varying sizes from 2 to 10 inspectors using an automated tool (more details can be found in the report [5]) that works in the following steps:

- *Input*: Individual inspectors' LS data, fault data (fault count, time taken), and team size (that user can specify).

- *Processing*: Generates all possible combination of virtual inspection teams (i.e. teams that never met together) sorted from "most dissimilar" to "most similar" teams (teams that involved highest number of clusters to teams that involve lowest number of clusters) in terms of the LSs of inspectors belonging to a virtual team. This *sorting* is done using a variety of multivariate statistical techniques (i.e., *Principal Component Analysis* - PCA, *Cluster Analysis* - CA and *Discriminant Analysis* - DA). More details on these techniques also appear in [5].

- *Output*: inspection effectiveness and efficiency for each team generated.

We manipulated LS disparity to create virtual inspections for team size of 2-10 inspectors, because past research [31] shows that inspections are most cost effective up to 10 inspectors. Therefore, for each inspection team size (e.g., N = 3); using LS and fault data from 13 subjects, we generated virtual inspection teams ($^{13}C_3$ - ranging from most dissimilar to similar in terms of LS disparity) and outputs inspection performance for each virtual team created. This process was repeated until team size 10.

Figure 6 compares the average number of unique faults detected (*effectiveness* – left side) and fault detection rate (*efficiency* – right side) for virtual inspection teams of size 2 to 10. Each line in the graph represents inspection effectiveness and efficiency for a team size. The number of clusters involved in team formation is always equal to team size [5]. The more number of clusters involved, the more dissimilar a team for a particular team size would be (as it increases the LS disparity among inspectors). For example, in team size 4, inspection teams created from a total of four clusters would be the most dissimilar team whereas, teams created from one cluster will be the team with inspectors of most similar LSs. An evident observation from the Figure 6, is that, effectiveness and efficiency of inspection teams increases with an increase in the LS disparity.

To evaluate this effect, we performed regression analysis to see *whether dissimilarity in LSs yields higher fault coverage and higher fault rate for inspection teams?* The results in Table 4 shows that inspection performance of larger teams (i.e., N = 6-9 for *effectiveness* and N = 4-9 for *efficiency*) significantly favors LS dissimilarity amongst individual inspectors. For smaller team sizes (e.g., team size 3), inspection performance is not significant because smaller teams are created from less number of clusters (e.g., team size 3 has maximum 3 clusters) which does not lead to LS disparity among team members in dissimilar teams. Also, for larger team sizes (e.g., team size 10), the more number of cluster creation leads to the repetition of team members (due to 8 possible LS categories) with same LS preference which does not results in unique fault reporting.

Figure 6. LSs disparity vs. Inspection effectiveness and efficiency for team size 2 to 10

Table 4. LS vs. Inspection Performance of Virtual Team

Team Size	Effectiveness	Efficiency
2	P=0.294	P=0.006
3	P=0.072	P=0.100
4	P=0.017	P=0.011
5	P=0.086	P=0.002
6	P=0.025	P=0.042
7	P<0.001	P=0.002
8	P=0.001	P=0.001
9	P=0.003	P=0.001
10	P=0.215	P=0.259

6.3 Eye Tracking vs, Virtual Inspection Teams

The tool (used to evaluate LS dissimilarity for virtual teams) was modified to use eye movement data of individual inspector as an input and output the overall eye movement data for all virtual inspection teams in each team size. We performed linear regression analysis to evaluate eye tracking factors that affects the high inspection performance of dissimilar teams of varying inspection team sizes. The results (Table 5) shows the linear regression output of eye tracking vs. LS dissimilarity data of team size 2 to 10. The shaded portion in table 5 shows significant positive correlation wherein, bold and italics denote negative correlation. Major observations from Table 5 are listed below:

- High-performing teams spent less time fixating per page ($T_{fixation}$) and overall spent less time on each page (T_{total}). This was especially true for larger teams;

Table 5. Eye movement vs. virtual inspection teams of size 2-10

Team Size	Fixation Time Per Page $T_{fixation}$	Time Per Page T_{total}	Total Fixations at ROI F_{ROI}	Total Duration at ROI T_{ROI}
2	P=0.003	P=0.006	P=0.109	P=0.010
3	P=0.103	P=0.100	P=0.556	P=0.026
4	P=0.011	P=0.029	P=0.058	P=0.004
5	P=0.003	P=0.003	P=0.003	P=0.003
6	P=0.038	P=0.038	P–0.038	P–0.038
7	P–0.129	P–0.074	P<0.001	P<0.001
8	P=0.001	P=0.001	P<0.001	P<0.001
9	P<0.001	P<0.001	P<0.001	P=0.001
10	P–0.198	P–0.198	P–0.198	P–0.198

- Conversely, fixating on ROI's (F_{ROI}) and total time spent at fault locations (T_{ROI}) resulted in increased inspection performance for larger team sizes (N=6 to 9).

Overall, creating inspection teams guided by LSs led to improved inspection performance (effectiveness and efficiency) and if individual inspectors naturally tend to spent more time processing information and extracting faults by fixating at the area where faults are present, it resulted in improved performance as well.

6.4 LS vs. Eye Tracking

We also wanted to investigate the impact of eye movement of individual LSs category (e.g., ACT vs REF) on inspection outcome to gain more insights into how eye movement factors relates to different LS categories. Therefore, individual LS categories were separated from LS of all participants and results in individual LS categories consists of following number of participants: ACT (4) vs REF (9), SEN (8) vs INT (5), VIS (10) vs VER (3), and SEQ (7) vs. GLO (6). All individual LS category data (inspection and eye tracking) was collected. Total fixations at ROI were taken as one tenth of the actual value ($F_{ROI}/10$) to accommodate all results in the graph. While comparing the eye movement performance of each LS category, it was found that, participants with **ACT** - Active LS (Figure 7) had the highest number of fixation time per page and time per page (which were shown to be negatively correlated to inspection performance in Section 6.1 and 6.2). Also shown in Figure 7, participants with **SEQ**- Sequential LS preference had maximum number of total fixation at ROI and total duration at ROI (that was positively correlated with inspection performance).

Figure 7. Comparison of eye movements of LS categories

Table 6. Multiple regression results comparing inspection performance of LS categories based on eye movement factors

$T_{fixation}$ - *Fixation Time Per Page*, T_{total} - *Time Per Page*, S_{linear} - *Linear Saccade Per Page (in %)*, F_{ROI} - *Total Fixations at ROI*, T_{ROI} - *Total Duration at ROI*
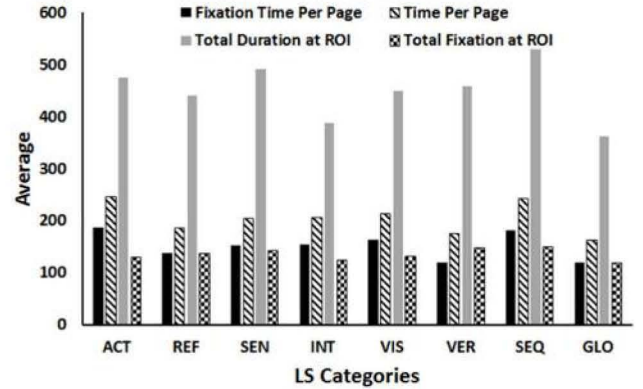
	ACT	$T_{fixation}$	T_{total}	S_{linear}	F_{ROI}	T_{ROI}
Effectiveness	Correlation	-0.624	-0.359	-0.822	-0.118	-0.102
	P-value	0.188	0.321	0.089	0.441	0.449
Efficiency	Correlation	-0.997	-0.923	-0.134	-0.199	-0.452
	p-value	0.002	0.039	0.433	0.401	0.274
	REF	$T_{fixation}$	T_{total}	S_{linear}	F_{ROI}	T_{ROI}
Effectiveness	Correlation	0.305	0.371	-0.267	0.584	0.524
	P-value	0.218	0.163	0.244	0.049	0.074
Efficiency	Correlation	-0.368	-0.341	-0.012	-0.006	-0.074
	p-value	0.165	0.185	0.488	0.494	0.425

	SEN	$T_{fixation}$	T_{total}	S_{linear}	F_{ROI}	T_{ROI}
Effectiveness	Correlation	0.660	0.533	0.053	0.845	0.831
	P-value	0.037	0.087	0.450	0.004	0.005
Efficiency	Correlation	-0.227	-0.380	0.780	0.261	0.171
	p-value	0.294	0.176	0.011	0.266	0.343
	INT	$T_{fixation}$	T_{total}	S_{linear}	F_{ROI}	T_{ROI}
Effectiveness	Correlation	-0.674	-0.475	-0.843	-0.799	-0.875
	P-value	0.106	0.209	0.036	0.052	0.026
Efficiency	Correlation	-0.881	-0.748	-0.815	-0.896	-0.967
	p-value	0.024	0.073	0.046	0.023	0.004

	VIS	$T_{fixation}$	T_{total}	S_{linear}	F_{ROI}	T_{ROI}
Effectiveness	Correlation	0.328	0.255	-0.028	0.720	0.689
	P-value	0.178	0.238	0.469	0.009	0.014
Efficiency	Correlation	-0.485	-0.573	0.351	0.098	0.003
	p-value	0.078	0.042	0.160	0.393	0.497
	VER	$T_{fixation}$	T_{total}	S_{linear}	F_{ROI}	T_{ROI}
Effectiveness	Correlation	-0.170	0.396	-1.00	-0.179	-0.254
	P-value	0.446	0.370	0.005	0.443	0.418
Efficiency	Correlation	-0.665	-0.418	-0.857	-0.671	-0.727
	p-value	0.269	0.453	0.172	0.266	0.241

	SEQ	$T_{fixation}$	T_{total}	S_{linear}	F_{ROI}	T_{ROI}
Effectiveness	Correlation	0.515	0.473	-0.105	0.848	0.847
	P-value	0.118	0.142	0.412	0.008	0.008
Efficiency	Correlation	-0.003	-0.054	-0.052	0.620	0.593
	p-value	0.497	0.454	0.456	0.069	0.080
	GLO	$T_{fixation}$	T_{total}	S_{linear}	F_{ROI}	T_{ROI}
Effectiveness	Correlation	-0.542	-0.239	-0.754	-0.747	-0.856
	P-value	0.133	0.324	0.042	0.044	0.015
Efficiency	Correlation	-0.859	-0.676	-0.373	-0.926	-0.890
	p-value	0.014	0.070	0.233	0.004	0.009

To enable such comparison for each LS category, we performed multiple regression to find the eye movement factors in each LS category that impacts inspection performance (i.e. effectiveness and efficiency). The results are shown in Table 6. The results (Table 6) shows the following observations:

- *SEN* LS category has the maximum number of significantly positive supporting factors of eye movement ($T_{fixation}$, F_{ROI}, and T_{ROI}) for high inspection effectiveness.
- *SEQ* category had the second highest number of eye movement factors (F_{ROI} and T_{ROI}) that significantly supports inspection effectiveness positively and are important in terms of high performance (as reported in Section 6.1).
- For inspection efficiency, there was no significant positive factor in eye movement of different LS categories but participants who tend towards *INT* and *GLO* LS category had eye movement factors that significantly affects inspection efficiency in a negative manner.
- Interestingly, for *VER*- verbal learners (conducive for reading a NL document like SRS); reading in a linear fashion is significantly negatively correlated to the overall inspection effectiveness (p=0.005).

Based on the results, it could be said that participants that tend towards *Sequential* learning (and *Sensing* and *Reflective* to some extent) are preferred for requirements inspections by fixating on the ROI's. Therefore, a certain combination of LSs should be selected that would enable inspectors to focus on areas where faults may exist and enable enough diversity to be able to reduce overlap of faults and increase inspection performance.

7. THREATS TO VALIDITY
Tracking eye movements with EyeLink 1000 desktop mount for our experiment had number of validity threats. Participating subjects were students in academic settings and are likely not represent professionals in industry setting. While performing inspection, Participants tend to lean forward or backward on chair during inspection which changes their distance and might go out of range (40 cm > distance > 70 cm) from the eye tracking apparatus. Hence, some eye movement data might have gotten lost and were not recorded. We were also able to handle some of the validity threats. Participants were trained by a single instructor for inspection training which handled training bias. Same PGCS document was given to each participant for inspection which addresses the heterogeneity effect. Participants were given enough time to perform inspection task and were allowed to take breaks which handled the fatigue effect. The accuracy of eye movements while inspecting PGCS document was assured by calibrating and validating the eye movement before actual inspection. After each break, drift correction of participant's eye from eye-tracking apparatus was performed and then inspection task was resumed. This addresses accuracy of eye movement data.

8. DISCUSSION
The objective of this study was to investigate the relationship between eye movement of inspectors and inspection performance in general as well as inspection performance when LS for each inspector is taken into account. The study also evaluated the impact of each LS and different eye movement factors that can affect inspection performance positively (or negatively). The results from Section 6 showed that, in general, *fixations at ROI had statistically significant positive effect on inspection effectiveness of individual inspectors as well as virtual teams*. This means, effective inspectors tend to spend more time focusing at the area where faults exist to understand and report faults documented in those areas of

the requirements document. Conversely, ineffective inspectors find it difficult comprehending the requirement document and therefore, they fixate throughout the document. Results also showed that, overall, *teams with inspectors of diverse LSs performed significantly better as compared to teams with similar inspectors*. The result is later also validated via eye tracking data which showed that, *diverse inspectors spent significantly spent more time at ROI while detecting faults during inspection*. Additionally, results also suggested that, inspectors with *SEQ LS preference had eye movement factors (F_{ROI}, T_{ROI}) that significantly impacts inspections in positive manner*.

In spite of small data size, the results show that inspectors who detect more faults during inspections, understand requirements information faster, and focus more at the fault areas to find and log faults and that *SEQ-SEN-REF-VIS cluster of inspectors exhibited this tendency to fixate on ROI's*. This resulted in greater inspection effectiveness and efficiency for this cluster. These results provide us with initial evidence and motivate us for further investigation. We plan to replicate this study with higher number of participants in future along with software professionals in industry. We also plan to investigate the reading patterns of inspectors on identifying specific fault types to be able to train inspectors on identifying those fault types or fault prone areas in requirements document.

9. REFERENCES
[1] Aceituna, D., Do, H., Walia, G. S. and Lee, S.-W. Evaluating the use of model-based requirements verification method: A feasibility study. In *Proceedings of the Empirical Requirements Engineering (EmpiRE), 2011 First International Workshop on* (Trento, Italy, 30 Aug, 2011).

[2] Albayrak, Ö. and Carver, J. C. Investigation of individual factors impacting the effectiveness of requirements inspections: a replicated experiment. *Empirical Software Engineering*, 19, 1 2014), 241-266.

[3] Allert, J. Learning style and factors contributing to success in an introductory computer science course. In *Proceedings of the Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on* (Finland, Sept, 2004).

[4] Aranda, G. N., Vizcaíno, A., Cechich, A. and Piattini, M. A cognitive-based approach to improve distributed requirements elicitation processes. In *Proceedings of the Cognitive Informatics, 2005.(ICCI 2005). Fourth IEEE Conference on* (Irvine, USA, 8-10 Aug., 2005).

[5] Bavanari, H. *Software Inspection Team Formation Based on the learning Style of Individual Inspectors*. Master's Paper, North Dakota State University, 2012.

[6] Bednarik, R. and Tukiainen, M. An eye-tracking methodology for characterizing program comprehension processes. In *Proceedings of the Proceedings of the 2006 symposium on Eye tracking research & applications* (2006).

[7] Berry, D. M. and Kamsties, E. *Ambiguity in requirements specification*. Springer, 2004.

[8] Cao, J. and Nishihara, A. Understand learning style by eye tracking in slide video learning. *Journal of Educational Multimedia and Hypermedia*, 21, 4 2012), 335-358.

[9] Carver, J. The impact of background and experience on software inspections. *Empirical Software Engineering*, 9, 3 2004), 259-262.

[10] Carver, J., Shull, F. and Basili, V. Observational studies to accelerate process experience in classroom studies: an evaluation. In *Proceedings of the Empirical Software Engineering, 2003. ISESE 2003*.

[11] Carver, J. C., Nagappan, N. and Page, A. The impact of educational background on the effectiveness of requirements

inspections: An empirical study. *Software Engineering, IEEE Transactions on*, 34, 6 2008), 800-812.

[12] Chandon, P., Hutchinson, W. J. and Young, S. H. *Measuring the value of point-of-purchase marketing with commercial eye-tracking data*. INSEAD, 2001.

[13] Charkins, R., O'Toole, D. M. and Wetzel, J. N. Linking teacher and student learning styles with student achievement and attitudes. *Journal of Economic Education*1985),111-120.

[14] Doolan, E. Experience with Fagan's inspection method. *Software: Practice and Experience*, 22, 2 1992), 173-182.

[15] EyeLink, I. *User Manual version (1/10/2005)*. SR Research Ltd, Mississauga, Ontario, Canada, 2005-2008.

[16] Fagan, M.E.*Advances in software inspections*.Springer, 2001.

[17] Felder, R. M. Are learning styles invalid?(Hint: No!). *On-Course Newsletter*2010), 1-7.

[18] Felder, R. M. and Henriques, E. R. Learning and teaching styles in foreign and second language education. *Foreign Language Annals*, 28, 1 1995), 21-31.

[19] Felder, R. M. and Silverman, L. K. Learning and teaching styles in engineering education. *Engineering education*, 78, 7 1988), 674-681.

[20] Felder, R. M. and Spurlin, J. Applications, reliability and validity of the index of learning styles. *International Journal of Engineering Education*, 21, 1 2005), 103-112.

[21] Goldberg, J. H. and Kotval, X. P. Computer interface evaluation using eye movements: methods and constructs. *International Journal of Industrial Ergonomics*, 24, 6 1999), 631-645.

[22] Goswami, A. and Walia, G. An empirical study of the effect of learning styles on the faults found during the software requirements inspection. In *Proceedings of the Software Reliability Engineering (ISSRE), 2013 IEEE 24th International Symposium on* (Pasadena, CA, 4-7 Nov., 2013).

[23] Goswami, A., Walia, G. and Singh, A. Using Learning Styles of Software Professionals to Improve their Inspection Team Performance.In *Proceedings of the SEKE*(Pittusburgh, 2015).

[24] Granka, L. A., Joachims, T. and Gay, G. Eye-tracking analysis of user behavior in WWW search. In *Proceedings of the Proceedings of the 27th annual international ACM SIGIR conference on Researchand development in information retrieval* (2004).

[25] Ivory, M. Y. and Hearst, M. A. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, 33, 4 2001), 470-516.

[26] Just, M. A. and Carpenter, P. A. A theory of reading: from eye fixations to comprehension. *Psychological review*, 87, 4 1980), 329.

[27] Keirsey, D. and Bates, M. M. *Please understand me*. Prometheas Nemesis, 1984.

[28] Kolb, D. A. *Experiential learning: Experience as the source of learning and development*. Prentice-Hall Englewood Cliffs, NJ, 1984.

[29] Krassanakis, V., Filippakopoulou, V. and Nakos, B. EyeMMV toolbox: An eye movement post-analysis tool based on a two-step spatial dispersion threshold for fixation identification. *Journal of Eye Movement Research*, 7, 1 2014), 1-10.

[30] Mandala, N. R., Walia, G. S., Carver, J. C. and Nagappan, N. Application of kusumoto cost-metric to evaluate the cost effectiveness of software inspections. In *Proceedings of the Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement* (Lund, Sweden, 17-22 Sep., 2012).

[31] Martin, J. and Tsai, W. T. N-fold inspection: A requirements analysis technique. *Communications of the ACM*, 33, 2 1990), 225-232.

[32] Mehigan, T. J. and Pitt, I. Detecting Learning Style through Biometric Technology for Mobile GBL. *International Journal of Game-Based Learning (IJGBL)*, 2, 2 2012),55-74.

[33] Miller, J. and Yin, Z. A cognitive-based mechanism for constructing software inspection teams. *Software Engineering, IEEE Transactions on*, 30, 11 2004), 811-825.

[34] Montgomery, S. M. Addressing diverse learning styles through the use of multimedia. In *Proceedings of the Frontiers in Education Conference* (1995).

[35] Myers, I. B., McCaulley, M. H. and Most, R. *Manual: A guide to the development and use of the Myers-Briggs Type Indicator*. Consulting Psychologists Press Palo Alto, CA, 1985.

[36] Parnas, D. L. and Lawford, M. The role of inspection in software quality assurance. *Software Engineering, IEEE Transactions on*, 29, 8 2003), 674-676.

[37] Parnas, D. L. and Weiss, D. M. Active design reviews: principles and practices. *Journal of Systems and Software*, 7, 4 1987), 259-265.

[38] Perry, W. E. *Effective Methods for Software Testing: Includes Complete Guidelines, Checklists, and Templates*. John Wiley & Sons, 2006.

[39] Porter, A., Siy, H., Mockus, A. and Votta, L. Understanding the sources of variation in software inspections. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 7, 1 1998), 41-79.

[40] Porter, A. A., Votta Jr, L. G. and Basili, V. R. Comparing detection methods for software requirements inspections: A replicated experiment. *Software Engineering, IEEE Transactions on*, 21, 6 1995), 563-575.

[41] Rutherfoord, R. H. Using personality inventories to help form teams for software engineering class projects. *SIGCSE Bull.*, 33, 3 2001), 73-76.

[42] Shull, F., Carver, J. and Travassos, G. H. An empirical methodology for introducing software processes. *ACM SIGSOFT Software Engineering Notes*, 26, 5 2001), 288-296.

[43] Shull, F., Rus, I. and Basili, V. How perspective-based reading can improve requirements inspections. *Computer*, 33, 7 2000), 73-79.

[44] Skiljan, I. IrfanView. *URL http:// irfanview. tuwien. ac. at/. Retrieved June*2012).

[45] Tinker, M. A. A photographic study of eye movements in reading formulae. *Genetic Psychology Monographs*1928).

[46] Uwano, H., Nakamura, M., Monden, A. and Matsumoto, K.-i. Analyzing individual performance of source code review using reviewers' eye movement. In *Proceedings of the Proceedings of the 2006 symposium on Eye tracking research & applications* (2006).

[47] Yusuf, S., Kagdi, H. and Maletic, J. I. Assessing the comprehension of UML class diagrams via eye tracking. In *Proceedings of the Program Comprehension, 2007. ICPC'07. 15th IEEE International Conference on* (2007).

[48] Zhai, S. What's in the eyes for attentive input. *Communications of the ACM*, 46, 3 2003), 34-39.