2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement

Issues and Opportunities for Human Error-based Requirements Inspections: An Exploratory Study

Vaibhav Anu, Gursimran Walia
Department of Computer Science
North Dakota State University
Fargo, USA
vaibhav.anu@ndus.edu,
gursimran.walia@ndus.edu

Wenhua Hu, Jeffrey C. Carver
Department of Computer Science
University of Alabama
Tuscaloosa, USA
whu10@crimson.ua.edu,
carver@cs.ua.edu

Gary Bradshaw
Department of Psychology
Mississippi State University
Starkville, USA
glb2@psychology.msstate.edu

***Abstract*—[Background] Software inspections are extensively used for requirements verification. Our research uses the perspective of human cognitive failures (i.e., human errors) to improve the fault detection effectiveness of traditional fault-checklist based inspections. Our previous evaluations of a formal human error based inspection technique called Error Abstraction and Inspection (EAI) have shown encouraging results, but have also highlighted a real need for improvement. [Aims and Method] The goal of conducting the controlled study presented in this paper was to identify the specific tasks of EAI that inspectors find most difficult to perform and the strategies that successful inspectors use when performing the tasks. [Results] The results highlighted specific pain points of EAI that can be addressed by improving the training and instrumentation.**

Keywords—human error, software requirements, inspection

I. Introduction

Software inspection is a proven quality verification approach that works by helping developers find faults at the source of their (faults') injection. Contemporary inspection techniques, however, cannot help identify all the faults because they only focus inspectors' attention on different type of faults (e.g., incorrect or missing functionality) recorded in software artifacts and not on the source of faults (i.e., *human errors*) [1, 2]. Therefore, we believe that, an inspection approach that can help inspectors focus on the underlying human errors (as opposed to manifestations of errors - faults) will be a significant improvement over standard fault-checklist based inspections. Moreover, the software development process, especially during requirements development, is human-centric and it is not surprising that most software faults and failures can be traced back to *errors* in human cognition [2, 3].

To help inspectors in understanding human errors, Lanubile et al [1] proposed the *Error Abstraction and Inspection (EAI)* approach, which adds an extra step to standard Fault Checklist (FC) inspection. The extra step requires inspectors to retrospectively abstract underlying errors from the faults that were found during the FC inspection. This process is called *Error Abstraction* (EA). After the EA step, inspectors use the abstracted human error information to re-inspect the requirements document for additional faults (this step of EAI is called *error-informed re-inspection*).

Building on Lanubile's work, we created a *Human Error Taxonomy (HET)* to make inspectors more effective during the error abstraction (EA) step by providing inspectors with structured error information. HET provides inspectors with a structured list of the most commonly occurring requirements phase human errors. HET was developed with the help of a Cognitive Psychologist (Dr. Bradshaw, co-author) after performing a systematic literature review that combined research from Software Engineering with research from Cognitive Psychology. Furthermore, as inspectors (who generally are software engineers) are not expected to have formal human error training, we have reduced the HET into an intuitive decision tree framework called *Human Error Abstraction Assist (HEAA)*. Details about HET and HEAA appear in Section II. The following paragraphs provide an overview of the prior research on EAI and the motivation for current study.

Prior Research: Walia and Carver [2], and Anu et al [4] have shown that the error-based inspection approach, EAI, is a significant improvement over traditional fault-based inspection. Our previous experiments [5] provide evidence that, EAI helps locate up to 150% more faults (that are otherwise left undetected when using only the FC inspection approach). We have also provided evidence that the performance of inspectors during the Error Abstraction (EA) leg of EAI has a strong positive impact on their fault detection effectiveness [6]. That is, a better understanding of the human errors committed during the creation of a requirements document leads to an improved fault detection effectiveness when inspecting the document using the EAI approach.

Motivation: During our previous studies [4, 6], we also found that, even though EAI has shown promise as an inspection approach, there are areas of EAI that require significant refinements. We found that improving the *error-informed re-inspection* step was one of the major areas we need to focus upon (as it currently is an ad hoc process, which each inspector devises on their own). Therefore, the current paper describes an exploratory experiment that was designed and conducted to understand: *what are the major avenues for improving the error-informed re-inspection step of EAI*? To achieve insights into the improvement-avenues, it was first necessary to identify and understand the strategies that successful inspectors use when performing the error-informed

This study was supported by National Science Foundation Awards 1423279 and 1421006

978-1-5090-4039-1/17 $31.00 © 2017 IEEE
DOI 10.1109/ESEM.2017.62

re-inspection step. The identified re-inspection strategies can be used to train subjects in future EAI evaluations.

Another area that we have continuously tried to improve during our various evaluations [4–7] is the error abstraction (EA) step of the EAI approach. Based on the results and subject-feedback from the previous studies, we have added minor refinements to the EA tool (i.e., HEAA). However, our main goal with the current study was *not* to evaluate the HEAA itself. Instead, we wanted to focus on the issues that inspectors face when using HEAA to abstract human errors from faults.

Primary Goal of Current Study: Primary goal of conducting the controlled study described in this paper was to gather and analyze data about: (1) issues that inspectors face when abstracting human errors from faults and, (2) strategies that inspectors use when using the abstracted human error information to locate additional related faults.

II. Background

HET: The HET classifies 15 commonly occurring requirements phase human errors (identified from the software engineering literature) into three high-level human error types. The *three high-level error types of HET* are *slips*, *lapses*, and *mistakes*, and were identified from a widely respected human error theory proposed by James Reason [8]. Reason proposes that, when faced with a situation that demands problem-solving, humans perform two cognitive activities: *planning* and *execution*. Reason refers to the cognitive failures (or human errors) that occur during planning as *mistakes*, and the cognitive failures that occur during execution as *slips* and *lapses*. *Slips* are a result of inattention while executing routine tasks. *Lapses* also occur when executing well-planned routine tasks, but are failures of memory (e.g., forgetting a step of the plan). *Mistakes* are planning failures that occur when trying to solve an unfamiliar problem. Details about HET's 3 error types and 15 error classes can be found as a technical report in [9].

HEAA: HEAA [9] is an error abstraction (EA) guidance framework, which was developed to help focus inspectors' attention on the human errors (contained in the HET) that occur across the various requirements engineering (RE) activities (e.g., elicitation, analysis, specification). Subjects in a previous EAI study [7] provided feedback that instead of starting with categorizing the human errors into Slips, Lapses or Mistakes, it is more intuitive to focus on RE activities. Hence, creation of HEAA required the authors to distribute the 15 error classes of HET across various RE activities. HEAA is a control flow style process, wherein control statements appear (inside decision nodes) in a top to bottom order. When using HEAA, inspectors have to make decisions at three (3) levels. At Level 1, the inspector has to decide the RE activity in which the fault originated (i.e., the human error occurred). At Level 2, the inspector has to decide the high level human error type that was committed (slips/lapse/mistake). Based on decisions taken for Levels 1 and 2, at Level 3, subjects have to select an adequate human error class. For inspectors' convenience, we have unpacked HEAA's decision tree into a detailed, self-explanatory, and stepwise system that can be found in [9].

Section III describes the subjects, artifacts, and experiment steps of the controlled study conducted to achieve the goal.

III. Experiment Design

Study Subjects: Fifteen (15) Graduate students enrolled in the *Software Development Processes* course at North Dakota State University (NDSU) participated in the study. The course is a breadth course on software engineering topics covering the entire software development lifecycle.

Study Artifacts: An artifact that described the requirements for a Restaurant Interactive Menu (RIM) system and contained naturally occurring faults was used during the study. The RIM system allows restaurant owners to control the inventory, and restaurant customers to order and pay bills. We used the RIM SRS during this study to enable comparison of the results with a previous empirical study [4] that also used RIM and evaluated the usefulness of the EAI inspection approach.

Experiment Procedure and Exploratory Research Questions: The experimental procedure was carefully designed to gather insights that can be used to improve the *Error Abstraction (EA)* and the *error-informed re-inspection* steps of EAI. To do so, we first provided all the 15 subjects with a set of 16 known faults in the RIM SRS and asked subjects to use the HEAA tool to abstract human errors from the 16 faults. Please note that all subjects were provided with the same 16 RIM faults. The outcome of this task was *15 individual Error Report Forms* containing human errors committed during the creation of RIM SRS. The output of this step helped us compare the EA results of all subjects on same set of faults. The first research question (RQ1) investigated this aspect:

RQ1: At what level of the EA process (i.e., when using HEAA) are subjects making most of the misjudgments? Because HEAA is a control flow style process, if a subject makes an incorrect decision in an initial decision level, then the rest of the flow is automatically rendered incorrect. This question itself has two sub-exploratory areas: (1) RE activity being the first decision level - Which RE activities are most difficult to choose correctly?, and (2) For subjects who picked the correct RE activity for a given fault, how many were able to pick the correct error type (slips, lapse, mistake)?

After the subjects abstracted human errors from the given 16 RIM SRS faults, they were given the expected error abstraction results for each of the 16 faults. The expected abstraction results were decided upon by the researchers after discussing with the Cognitive Psychologist, Dr. Bradshaw. The subjects were asked to use the provided human error information to inspect RIM SRS and locate additional faults related to the provided human errors. The idea behind giving subjects the expected error abstraction results for each of the 16 given faults was to understand how individual subjects use the *same human error information* to find additional related faults. The outcome of this step was *15 individual Fault Report Forms* containing new faults in RIM SRS. The motivation behind asking subjects to perform this task (i.e., error-informed inspection) was to evaluate if human error information helps inspectors in identifying additional faults (that are overlooked when only fault-checklist inspection is used). The following research question was devised to analyze the error-informed inspection data collected during this task:

RQ2: Does human error information lead to additional faults being identified during error-informed inspection?

RQ2 had two sub-exploratory areas: (1) When provided with the human error (that caused a fault), where in the SRS do subjects look in order to find other related faults that were caused by the human error?, and (2) Why, if at all, are some inspectors more effective at finding faults when using human error information?

IV. Data Analysis and Results

The results are organized around the two research questions (RQ's) described in Section III.

RQ1: At what level of the EA process (i.e., when using HEAA) are subjects making most of the misjudgments?

Table I provides an overview of EA results reported by the 15 subjects. Each row in Table I provides EA accuracy at different HEAA levels across all the subjects for each fault. As an example, for Fault #2, 13 out of 15 subjects were able to select the correct RE activity (where the fault originated). Therefore, we only evaluated the rest of the abstraction data of those 13 subjects who selected the correct RE activity. Our analysis showed that, 11 of those 13 subjects selected the right error type (slips/lapse/mistake). Furthermore, only 9 of the remaining 11 subjects selected the correct human error class for Fault #2. Overall, 9 out of 15 subjects provided the expected error abstraction result for Fault #2 (i.e., only 60% of subjects were able to accurately abstract the human error that caused Fault #2). Similar analysis was done for all 16 faults.

Fig. 1 compares the EA accuracies achieved by the subjects at the 3 levels of HEAA. Observations from Table I and Fig. 1:

At RE Activity Level of HEAA: Based on the EA results, subjects frequently misjudged the RE activity in which the faults originated. Figure 1 shows that the subjects had the most difficulty when picking the adequate RE activity wherein the human error occurred (and resulted in the insertion of the fault being analyzed). Overall, for all 16 faults, subjects achieved an EA accuracy of 67% at the RE activity level. HEAA is a decision flow process and selecting an appropriate RE activity is the first decision that the subjects have to make. Selecting the incorrect RE activity essentially renders the rest of the abstraction effort futile.

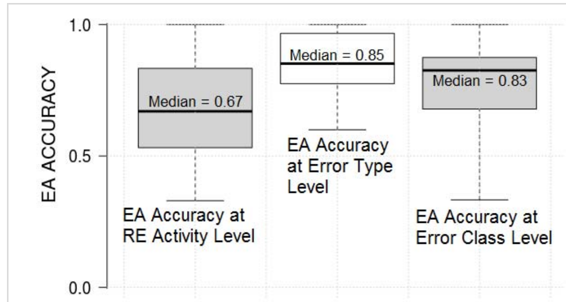

Fig. 1. Error Abstraction (EA) Accuracy at three HEAA-levels

Table II provides the frequencies with which RE activities were confused with each other. For example, altogether, for all 16 faults we expected 104 occurrences of analysis activity in the error abstraction data reported by the subjects. Subjects correctly reported analysis activity 69 times (out of the 104 expected occurrences). Hence, there were 35 remaining occurrences where the expected RE activity was analysis, but subjects incorrectly reported some other activity. Table II shows that ***Elicitation*** was the activity that subjects found hardest to pick correctly. That is, subjects picked other RE activities for the faults that originated in the elicitation activity. Overall, subjects selected elicitation activity correctly in only 41.67% of the cases. Subjects confused elicitation activity with specification activity in most cases.

Elicitation was also the activity that was most frequently incorrectly selected instead of the other activities. That is, elicitation activity was selected when another RE activity was expected. In 33 such instances, subjects chose elicitation when the expected RE activity was either analysis (in 16 incorrect instances) or specification (17 incorrect instances). Table II highlighted the fact that focused training-materials on elicitation activity need to be added to error abstraction (EA)

TABLE I. Progressive EA Correctness at the Three Decision Levels of HEAA

Fault #	Number of subjects who chose the correct RE activity (Level 1 of HEAA)	Number of subjects who chose the correct Error Type (Level 2 of HEAA)	Number of subjects who chose the correct Error Class (Level 3 of HEAA)	Overall Correctness: Number of subjects who reported correct EA result for the fault (correct at all 3 levels)
Fault 1	100% (15/15)	93.33% (14/15)	100% (14/14)	93.33% (14/15)
Fault 2	86.67% (13/15)	84.62% (11/13)	81.82% (9/11)	60% (9/15)
Fault 3	66.67% (10/15)	100% (10/10)	100% (10/10)	66.67% (10/15)
Fault 4	53.33% (8/15)	75% (6/8)	83.33% (5/6)	33.33% (5/15)
Fault 5	80% (12/15)	83.33% (10/12)	90% (9/10)	60% (9/15)
Fault 6	66.67% (10/15)	80% (8/10)	37.5% (3/8)	20% (3/15)
Fault 7	33.33% (5/15)	60% (3/5)	33.33% (1/3)	6.67% (1/15)
Fault 8	66.67% (10/15)	80% (8/10)	87.5% (7/8)	46.67% (7/15)
Fault 9	73.33% (11/15)	63.64% (7/11)	85.71% (6/7)	40% (6/15)
Fault 10	53.33% (8/15)	87.5% (7/8)	57.14% (4/7)	26.67% (4/15)
Fault 11	46.67% (7/15)	100% (7/7)	71.43% (5/7)	33.33% (5/15)
Fault 12	86.67% (13/15)	100% (13/13)	76.92% (10/13)	66.67% (10/15)
Fault 13	66.67% (10/15)	70% (7/10)	71.43% (5/7)	33.33% (5/15)
Fault 14	46.67% (7/15)	85.71% (6/7)	83.33% (5/6)	33.33% (5/15)
Fault 15	53.33% (8/15)	100% (8/8)	87.5% (7/8)	46.67% (7/15)
Fault 16	100% (15/15)	93.33% (14/15)	64.29% (9/14)	60% (9/15)

TABLE II. THE FREQUENCIES WITH WHICH RE ACTIVITIES WERE CONFUSED WITH EACH OTHER

Expected \ Selected	Elicitation	Analysis	Specification	Management	Total expected occurrences of the RE activity in the abstraction data (for all 15 subjects)
Elicitation	10 (41.67%)	3 (12.50%)	10 (41.67%)	1 (4.17%)	24
Analysis	16 (15.38%)	69 (66.35%)	14 (13.46%)	5 (4.81%)	104
Specification	17 (15.18%)	12 (10.71%)	83 (74.11%)	0 (0.00%)	112
Management	(0.00%)	(0.00%)	(0.00%)	(0.00%)	0

training. Involvement of multiple sources of requirements (as several domain-experts act as sources) during elicitation activity made it hard for inspectors (who were not involved during elicitation of the requirements being inspected) to come up with elicitation-related error scenarios. Compared to elicitation, inspectors had to think about only one actor when creating error-scenarios for activities like analysis or specification (analyst and author, respectively).

At Error Type Level of HEAA: As shown in Fig. 1, once subjects picked the right RE activity, they were able to pick the correct Error Type (slips/lapse/mistake) in most cases. Overall, inspectors were able to pick the correct error type 85% of the times (once they picked the right RE activity). In a similar previous study that evaluated error-based inspections [4], it was found that inspectors were able to abstract the correct error types only 62% of the times (as compared to 85% correctness shown by subjects of current study). Table III provides a comparison between the EA accuracies achieved by subjects of current study vs a similar previous study (that also evaluated the EAI inspection approach). Hence, the recent improvements that we have added to the HEAA tool and EA training have helped the inspectors in selecting the correct error types.

At Error Class Level of HEAA: Overall, inspectors were able to pick the correct human error class in most of the cases (average accuracy of 75.7% was observed at the error class level). However, for Faults number 6, 10, and 16 (in Table I), the accuracies at error class level were observed to be very low (37%, 57%, and 64% respectively). It was found that the expected error classes for these three faults belonged to the Mistake error type. This showed that subjects found it difficult to pick error classes under the *Mistake* error type. This may be because Mistake error type has 11 error classes under it, whereas Slip and Lapse only have 2 error classes each [9].

Overall Accuracy of Error Abstraction: The last column of Table I shows the overall correctness achieved by subjects for each of the 16 faults. An individual subject's error-abstraction result for a fault was counted to be correct only if the subject selected the expected RE activity, the expected Error Type, and finally the expected Error class for the fault. The mean overall-accuracy for all subjects was found to be around 45%, which shows that the training around the error abstraction leg of EAI needs to be improved significantly. A closer look at faults that had very low overall accuracy (e.g., Faults 7, 11 and 14 in Table I) showed that for such faults, most of the subjects abstracted the incorrect RE activity and therefore their rest of error abstraction effort was left incorrect by default.

TABLE III. EA ACCURACY COMPARSION

	Current Study: 15 students *inspected* RIM SRS	Previous Study [4]: 16 students inspected RIM SRS
Avg. accuracy when selecting *Error Type*	84.80%	62%
Avg. accuracy when selecting *Error Class*	75.70%	38%

RQ2: Does human error information lead to additional faults being identified during error-informed inspection?

Effectiveness of EAI as a Requirements Inspection Technique: We analyzed the number of additional faults located by individual subjects during the error-informed reinspection of the RIM SRS. As can be seen in Fig. 2, all 15 subjects were able to find new faults using the human error information provided to them. At an average, subjects located ***11.4*** new faults using the error information supplied to them. In a similar previous study [4], we had found that inspectors were able to locate an average of 6.75 faults in the RIM SRS when using the Fault-checklist (FC) technique (unlike the current study design, a typical EAI process starts with FC inspection of the SRS, followed by error abstraction from faults found during FC inspection, followed by reinspection using abstracted errors). Based on this information, we ran a one-sample t-test to determine whether the subjects in the current study were able to locate a significantly larger number of faults (than 6.75) when performing the error-informed inspection of RIM SRS. The p-value was found to be 0.000512, which showed that, during the current study, subjects were able to find a significantly larger number of faults (than 6.75) during the error-informed inspection of RIM SRS.

When provided with the correct human error (that caused a fault), where in the SRS do subjects look in order to find other related faults that were caused by the human error. We performed an interpretive analysis (a qualitative data analysis technique [10]) on the fault data provided by subjects during the error-informed inspection of RIM. The goal of this analysis was to obtain insights into how subjects make use of error-information to find additional faults that are related to the errors. Currently, the error-informed inspection is an ad-hoc process, wherein individual inspectors devise their own strategies to locate new faults. We observed the location of all the faults reported by the subjects (not just the true-positives)

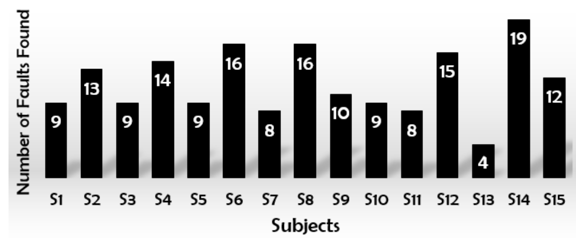

Fig. 2. Number of New Faults Found During Error-informed Inspection

TABLE IV. INTERPRETIVE ANALYSIS: STRATEGIES USED BY SUBJECTS DURING ERROR-INFORMED INSPECTION OF RIM SRS

Strategy Title	Strategy Description	Example
Additional faults in the same requirement	Subjects reviewed the requirement that contained the original fault when looking for additional faults. The *rationale* was that the same requirement may contain more faults (not necessarily similar to original fault) because the requirements engineer/s who worked on creating the requirement were already under the influence of the human error (that was abstracted from the given original fault)	As an example, for Fault #1, the error happened while specifying the requirement titled, 'RIM_CUSTLOGIN_S01'. This requirement can be found between lines Line 77 to 124 in the RIM SRS [9]. Multiple subjects used this strategy to report other faults in the same requirement. Furthermore, three subjects successfully found a new fault (true-positive) on Line #105.
Additional similar faults in other similar requirements	Subjects reviewed the RIM SRS [9] to first find any requirements that were similar to the requirement in which the original given fault was located. Next, if subjects were able to identify any similar requirement, they reviewed the identified requirement to find faults that were similar to the original fault. The rationale was that if a human error occurred while creating a specific type of requirement, then it is possible that the same human error might have occurred while creating other similar requirements.	As an example, Fault #2 is located in the requirement stated between lines 77-124, more specifically in the constraints section of the requirement. Fault #2 occurred due to carelessness while performing numerical calculations. Subjects looked at constraints sections of other requirements, specifically where numerical calculations may have been performed by requirements engineer/s. Eleven (11) subjects successfully used this strategy to find a similar calculation-related fault in a different requirement.
Additional faults in related requirements	The creators of RIM SRS have attached a related requirements section with every requirement. Subjects reviewed requirements related to the original requirement (in which the original given fault was located). The rationale was that if a human error occurred during the creation of a requirement, then the human error might have affected the creation process of related requirements as well.	As an example, Fault #3 is in a requirement titled, 'RIM_REQUEST_HELP_S03'. Four subjects reviewed a related requirement titled, 'RIM_ORDER_ENTREE_D08' and reported faults (found to be false-positives) in the latter requirement.
Additional similar faults in other requirements	For simplistic faults like 'missing information or missing words', subjects just read through the whole SRS to find if there are other instances where requirement-sentences were missing or words in the sentence were missing (rendering the requirement incomplete or ambiguous).	As an example, Fault #8 states that "Hacker is listed in the list of actors. However, the hacker has no role in this requirement". Subjects simply looked at the 'Actor' sub-section of all requirements to find if other requirements had faulty 'Actor' list. Four subjects successfully found a fault on Line #346. The fault was that the 'Actor' sub-section of the requirement was left blank (whereas all use cases must have at least one actor).

for each of the 16 fault-error combinations provided to them. Please recall that for each of the 16 RIM faults that we provided to the subjects, we also provided the human error that caused the fault. This analysis necessitated interpretation because, for each original given fault, we needed to compare and contrast the location of the reported faults (which subjects deemed to be related to the original fault) with the location of the original fault. Furthermore, we had no quantitative data about: why a subject thought that a certain reported fault was related to the original fault and human error (that was provided for the original fault). We also had no quantitative data about what prompted the subject to look for a related fault at a particular location. Therefore, we needed to derive meaningful interpretations (from the locations of the reported faults) about how a subject (or multiple subjects) found the particular reported faults. In other words, we were looking for prompts that subjects created in their mind when looking for new faults.

This analysis revealed four major strategies (shown in Table IV) that subjects used in order to locate new faults related to the given fault-error combinations.

Why, if at all, are some inspectors more effective at finding faults when using human error information? : We analyzed the fault data collected during the error-informed inspection of RIM SRS to determine if some inspectors are better than others when identifying new faults using human error information. A very crude analysis of the fault data reported by the subjects showed that the subjects who reported the most faults were clearly the subjects who were more successful during the error-informed inspection of RIM SRS. The three subjects who reported the most number of faults during the error-informed inspection of RIM SRS were S6, S8, and S14 (they reported 30, 27, and 40 faults respectively). The average number of reported faults by all subjects was 19 faults per subject. As can be seen in Fig. 2, subjects S6, S8, and S14 also had the highest fault-detection effectiveness among all the subjects (these subjects found 16, 16, and 19 true-positive faults respectively as shown in Fig. 2). A deeper evaluation of the Fault Report Forms of subjects S6, S8, and S14 highlighted the fact that these three subjects performed a more thorough review of the RIM SRS (during the error-informed inspection step) than the other inspectors. That is, for each of the 16 given fault-error combinations, these three inspectors covered the entire RIM SRS when trying to locate additional related faults. The other inspectors generally restricted their additional fault-search to only areas (in RIM SRS) that were in close vicinity of the original fault's location.

Based on these observations, Section VI summarizes the major issues for EAI approach and the improvement avenues.

V. THREATS TO VALIDITY

Because this study was an exploratory study with the primary goal of determining the problems faced by inspectors during various EAI steps, it has a number of threats to validity. One major threat to the interpretive analysis (provided in Table IV) is that the number of participating subjects were only 15 and hence, drawing strategy-related conclusions from just 15 subjects' data may not be accurate or complete (that is, there may be other strategies that inspectors use during error-informed inspection that this study did not reveal). The low number of participating subjects (15) is also a threat to the generalizability of other major results of this study.

A threat to this study's external validity comes from the fact that participating subjects were students in a classroom environment, rather than professional software engineers in a real project environment. We were able to control this threat to a certain extent, as the subjects were a mix of Computer Science and Software Engineering graduate (Masters and PhD) students, who are expected to have some industrial experience. Another threat is that the RIM SRS was created by student developers (by interacting with real clients) and the faults and errors in RIM SRS may not be representative of those found in an industrial strength SRS. We intend to address these threats in our future EAI-evaluations.

VI. Summary of Issues and Opportunities based on Results of Data Analysis

This section describes the main issues identified for EAI approach and the opportunities for improvement that were observed during the data analysis (shown in Section IV).

RQ1: At what level of the EA process (i.e., when using HEAA) are subjects making most of the misjudgments?

Issues: Two major issues identified for the error abstraction process (using HEAA) of EAI are: (i) When analyzing faults for abstracting human errors using HEAA, inspectors find it difficult to retrospectively think about the RE activity wherein a human error might have occurred and led to the injection of the fault (that is being analyzed). Subjects only achieved an abstraction accuracy of 67% at the RE activity level of HEAA, as shown in Fig. 1. (ii) Inspectors find it difficult to select appropriate error classes under the Mistake error type.

Opportunity: An avenue for improvement is that, when using EAI inspections, the training about RE activities needs to be comprehensive and special attention needs to be given on training subjects about what goes into *requirements elicitation*. In all our empirical evaluations of the EAI approach as a requirements inspection technique, we have focused more on training subjects about error types (slips/lapses/mistakes) and the different type of human errors that can occur during the requirements development process. Consequently, subjects performed well when abstracting Error Types and Error Classes. As shown in Fig. 1, subjects achieved median abstraction accuracies of 85% and 83% respectively at these two levels. However, the current analysis has highlighted the need for increasing focus on training subjects about RE activities. We intend to create comprehensive training material around RE activities and provide a dedicated training session on RE in our future studies. Another observation here is that those inspectors who were involved in the SRS artifact development may be better at identifying the RE activities where the fault originated. We intend to test this hypothesis in our future EAI studies. In future, we also intend to compare, at the three levels of HEAA, the cost of misjudgments committed by inspectors during the error abstraction process. The cost comparison will be performed in terms of the rework effort (time) associated with correcting the misjudgments.

RQ2: Does human error information lead to additional faults being identified during error-informed inspection?

First, regarding the effectiveness of the EAI approach, the analysis showed that human error information helped all 15 subjects in locating new faults in RIM SRS (see Fig. 2). This result confirms the results from our previous studies [4, 5] that EAI is a useful addition to the traditional fault-checklist (FC) inspections for improving requirements quality. Our analysis showed that subjects were able to locate an average of eleven (11) new faults in RIM SRS using human error information.

Opportunity: Our analysis was able to uncover four strategies (shown in Table IV) that inspectors used successfully when performing the error-informed inspection step of EAI. We intend to use these strategies to train subjects about error-informed inspection in our future EAI-evaluations.

VII. Conclusion and Future Work

This study has reported the major problem areas for a human error based inspection approach called EAI and the opportunities for improving the fault detection effectiveness of the EAI approach. It was found that error abstraction (EA) remains a primary area of concern and during EA training we need to focus on describing requirements engineering activities to the inspectors. Our analysis also revealed the various *error-informed inspection strategies* that we could train inspectors on. We expect that this will improve the overall fault detection effectiveness of the EAI approach. We intend to incorporate these findings into our future EAI evaluations.

Acknowledgment

This study was supported by NSF Awards 1423279 and 1421006. We would like to thank the students enrolled at NDSU and the course instructor for their participation.

References

[1] F. Lanubile, F. Shull, and V. R. Basili, "Experimenting with Error Abstraction in Requirements Documents," In Proceedings of the 5th International symposium on software metrics, 1998.

[2] G. S. Walia and J. C. Carver, "Using error abstraction and classification to improve requirement quality: Conclusions from a family of four empirical studies," Empir. Softw. Eng., vol. 18 (4), pp. 625–658, 2013.

[3] T. DeMarco and T. Lister, Peopleware: Productive Projects and Teams, Third Edition. Addison-Wesley Professional, 2013.

[4] V. Anu, G. Walia, G. Bradshaw, W. Hu, and J. C. Carver, "Usefulness of a Human Error Identification Tool for Requirements Inspection: An Experience Report," in 23rd International Working Conference on Requirements Engineering: Foundation for Software Quality, 2017.

[5] V. Anu, G. S. Walia, W. Hu, J. C. Carver, and G. Bradshaw, "Using A Cognitive Psychology Perspective on Errors to Improve Requirements Quality: An Empirical Investigation," in IEEE 27th International Symposium on Software Reliability Engineering (ISSRE'16), 2016.

[6] V. Anu, G. S. Walia, W. Hu, J. C. Carver, and G. Bradshaw, "Error Abstraction Accuracy and Fixation during Error-based Requirements Inspection," in 27th IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW '16)., 2016.

[7] V. Anu, G. S. Walia, W. Hu, J. C. Carver, and G. Bradshaw, "Effectiveness of Human Error Taxonomy during Requirements Inspection: An Empirical Investigation," in International Conference on Software Engineering and Knowledge Engineering, SEKE, 2016.

[8] J. Reason, Human error. New York: Cambridge University Press, 1990.

[9] V. Anu and G. S. Walia, "The Error Abstraction and Inspection (EAI) Approach: Experimental Package," 2016. [Online]. Available: http://vaibhavanu.com/NDSU-CS-TP-2016-002.html

[10] R. Elliott and L. Timulak, "Descriptive and interpretive approaches to qualitative research," Handb. Res. Methods Clin. Heal. Psychol., 2005.