Incorporating Human Error Education into Software Engineering Courses via Error-based Inspections

Vaibhav Anu

Department of Computer Science
North Dakota State University
Fargo, USA

vaibhav.anu@ndsu.edu

Gursimran Walia

Department of Computer Science
North Dakota State University
Fargo, USA

gursimran.walia@ndsu.edu

Gary Bradshaw

Department of Psychology
Mississippi State University
Mississippi State, USA

glb2@psychology.msstate.edu

ABSTRACT

In spite of the human-centric aspect of software engineering (SE) discipline, human error knowledge has been ignored by SE educators as it is often thought of as something that belongs in the realm of Psychology. SE curriculum is also severely devoid of educational content on human errors, while other human-centric disciplines (aviation, medicine, process control) have developed human error training and other interventions. To evaluate the feasibility of using such interventions to teach students about human errors in SE, this paper describes an exploratory study to evaluate whether requirements inspections driven by human errors can be used to deliver both requirements validation knowledge (a key industry skill) and human error knowledge to students. The results suggest that human error based inspections can enhance the fault detection abilities of students, a primary learning outcome of inspection exercises conducted in software engineering courses. Additionally, results showed that students found human error information useful for understanding the underlying causes of requirement faults.

Keywords

Human error; requirements inspection; taxonomy; psychology

1. INTRODUCTION

Requirements engineering (RE) is essentially a social and highly human-centric activity, which involves several stakeholders (e.g., developers, end users, analysts) working with each other to determine customer needs and translating them into clear and precise set of requirements. There exists a strong need to get system requirements right, as failure to do so will lead to systematic failure in software products built as a result of incorrect requirements.

Because of the above-mentioned importance of obtaining correct requirements, RE is an integral part of software engineering (SE) curriculum [7]. Students are trained on various RE activities like eliciting, analyzing, specifying, and validating requirements. However, these technical aspects of RE by themselves are not sufficient to equip the students, who are future software engineers, with all skills required to produce quality requirements. This is because RE in the real world, as stated earlier, is very human-centric. Consequently, RE requires students to be trained on a

variety of hard and soft skills, which are associated with a wide range of disciplines like cognitive psychology, sociology, and reliability engineering.

In this paper, we explore the feasibility of using requirements validation exercises to train SE students on an important RE skill that is associated to the science of Cognitive Psychology: *human errors*. Human error researchers study the cognitive and psychological processes that produce errors in human behavior. Human error research can help understand the nature and mechanism of errors that people can make during the RE process.

Safety-critical domains like aviation and medicine have successfully used human error research for process improvement [2, 14]. However, human error knowledge is still viewed by Software Engineering (SE) students, faculties, and practitioners to be a component of Psychology and most students are not aware of human errors and their importance in creating quality requirements. Because, RE is human-centric, it is not surprising that most RE faults can be traced back to human errors [13]. Lanubile and Basili were the first ones to note that requirements faults essentially are manifestations of underlying human errors [4]. They proposed that requirements validations should be driven by underlying human errors rather than the conventional fault-based (e.g. fault checklist) methods. They coined the term, *error-based inspections*, which require inspectors to analyze faults for underlying human errors (through a process called error abstraction) and then use the error information to find additional related faults. In comparison to error-based inspections, fault-based inspections (which are conventionally taught in SE courses) simply guide inspectors to focus on specific faults (manifestation of errors) like incorrectness, incompleteness, and ambiguity using a pre-defined set of questions.

The severe lack in human error training for SE students is also reflected in the fact that SE textbooks and literature has competing, and sometimes contradictory definitions of the terms errors and faults. These two terms are often used interchangeably [4, 12]. Lanubile and Basili separated the terms error and fault as follows:

Error: An error is a failure in human cognition. An error occurs when a developer's thought process is flawed. Example of human cognitive failures include inattention, carelessness, and misunderstanding.

Fault: A fault is a concrete manifestation of an error. For example, an incorrect fact getting recorded in a requirements artifact.

As part of our NSF grant, through interaction with SE and psychology researchers, we have developed a human error taxonomy (HET) of requirement errors based on comprehensive psychological frameworks of human errors proposed in the Cognitive Psychology literature. The current paper tries to explore whether HET can be used to educate students on human errors and tracing back faults to errors, which in turn could help them in detecting other related faults (manifestations of errors). To evaluate

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SIGCSE '17, Mar 08–11, 2017, Seattle, WA, USA

© 2017 ACM. ISBN 978-1-4503-4698-6/17/03...\$15.00

DOI: <http://dx.doi.org/10.1145/3017680.3017782>

this, two error-based inspection studies were conducted in separate graduate-level SE courses to evaluate if error-based inspections are useful in training students on human errors.

The rest of this paper is structured as follows: In Section 2, we provide a brief background about human error research and the human error instruments used to train students during the inspection studies. In Section 3, we provide the primary goals of conducting the inspection studies and the procedure followed during the studies. Section 4 describes the analyses performed on the inspection data collected during inspection studies. Section 5 provides a summary of results and Section 6 draws conclusions regarding learning imparted through inspection exercises.

2. BACKGROUND

Researchers have used several human error classification systems, like Reason's Swiss Cheese model [9], Human Factors Analysis & Classification system (HFACS) [11], Norman's classification [5] and Rasmussen's taxonomy [8] to show how people's actions and decisions can be erroneous in different situations. Most of these human error classification systems find their fundamental principals in the theory proposed by James Reason in his well-respected body of work on human errors [9]. Section 2.1 describes Reason's error classification and its applicability through both, everyday examples of human errors and RE specific examples.

2.1 Reason's Classification of Human Errors

Reason's theory provides a framework to understand the specific breakdowns (or human errors) in human information processing. Reason proposed that human errors are induced during two primary cognitive activities that humans perform when faced with any problem-solving situation: *planning* and *execution* [9, 10]. Reason further associated the human errors that happen during planning and execution to commonly observed erroneous human behaviors. The human errors of execution are associated with inattentiveness, carelessness, and forgetfulness. Errors associated with inattentiveness and carelessness are referred to as *slips*, and those associated with forgetfulness are called *lapses*. The human errors of planning are induced due to lack of adequate knowledge when creating the plan and are referred to as *mistakes*.

Slips, which result from inattention while executing routine tasks, can be exemplified in common day-to-day activities like typing incorrectly or "fat-fingering" due to inattention or carelessness. *Lapses* also occur when executing routine tasks, but are failures of memory. For instance, having planned to repair/replace a broken machine-part, but forgetting it due to some interruption (e.g., taking a break) is a common lapse. From RE perspective, slips and lapses typically occur during mundane activities like typing, taking notes, reading, and filing.

Mistakes, which are planning failures, occur when planning a solution for an unfamiliar problem. For example, a doctor misdiagnosing a patient either due to not studying this patient's symptoms properly or not having any experience whatsoever with the symptoms exhibited by this specific patient. Mistakes are generally the result of lack of adequate knowledge, which in turn arises when working in novel situations. In novel situations, individuals attempt to use rules or procedures that have successfully worked in similar situations in the past. Mistakes are particularly applicable to RE as RE is a creative activity where software engineers are trying to build a solution for a new problem (i.e., a new user need).

Owing to its applicability in a wide-range of domains, Reason's theory is frequently used for training students and industry practitioners on human errors. Educational material on Reason's

human error theory is used for training the operators of safety-critical equipment in domains like medicine, aviation, and nuclear power plants [6]. In medicine, patient safety is a major concern and many academic healthcare institutions rely on Reason's theory to impart patient safety education to healthcare providers like nurses, hospital administrators, and physicians. An example of such a safety training module published by Duke University School of Medicine can be found here: http://patientsafetyed.duhs.duke.edu/module_e/definitions.html

Furthermore, in both aviation and healthcare, researchers frequently use Reason's theory to develop educational content to train individuals like medical students, healthcare personnel, flight safety crew, and pilots on human errors [6, 14]. Provided the wide-spread usage of Reason's theory in academic institutions as well as in the industry for imparting human error training, it can be inferred that Reason's theory is an appropriate learning resource for SE students to learn about human errors.

2.2 Error-Based Inspections and HET

Originally proposed by Lanubile et al [4], error-based inspections add an extra step to the fault-based inspection technique. The extra step requires inspectors to analyze the faults found during fault-based inspection for underlying human errors and then use the human error information to find additional faults. Faults are analyzed for underlying human errors using a process called *Error Abstraction* (EA), wherein the inspector retrospectively analyses each fault to determine the human error that led to the injection of the fault. The premise of error-based inspections is that if the inspectors can be made aware of the underlying human errors, they can use error information to find additional faults that are either overlooked or missed when just focusing on faults.

The Human Error Taxonomy (HET) [1] was developed to support the error abstraction (EA) leg of error-based inspections. HET provides a structured list of the most commonly occurring requirements phase human errors. Without the HET, the inspectors have to rely on their creativity and their experience with the requirements development process when abstracting human errors from faults. This is because error abstraction is a retrospective investigation to determine what human error/s could have led to the injection of the fault being investigated. HET is an instrument that aids the inspector during error abstraction by providing a tangible list of commonly occurring requirements phase human errors.

The HET (Figure 1) classifies the requirements phase human errors into Reason's slips, lapses, and mistakes. HET was developed by first collecting data, from SE literature, about the common failure

Figure 1. Human Error Taxonomy (HET)

processes (human errors) associated with RE. Next, the identified errors were interpreted in light of human information processing limitations (slips, lapses, and mistakes). For example, SE literature reports that requirement practitioners sometimes analyze user needs without having knowledge about the overall functionality of the system, which leads to incorrect specifications. These errors (Application Errors in Figure 1) are a classic example of Reason's mistakes, as a problem is being solved without having adequate knowledge about the problem-space. Fifteen such commonly occurring human errors were identified and classified as either a slip, a lapse, or a mistake to create the HET. An Error Abstraction (EA) instrument called Human Error Abstraction Assist (HEAA) [3] was also developed to act as a *human error intervention tool* during requirements inspections. HEAA is also used as a training tool to teach students the *faults to human errors* trace-back process.

3. STUDY DETAILS

Inspection studies in two separate SE graduate-level courses were conducted using the error-abstraction technique (supported by HET). In this section, we present the study goals, and other details.

3.1 Study Objectives

The major objective of having students perform error-based inspections was to help them understand the underlying human errors that can lead to insertion of requirement faults. Additionally, we hypothesize that focusing on underlying human errors will enable a higher coverage of faults that are otherwise overlooked or missed when only focusing on the fault information:

Objective 1: Evaluate whether error-inspections (supported by HET and EA instrument) can help enhance students' fault detection abilities as compared to fault-based inspections.

Objective 2: Evaluate whether students find error-based inspections a useful learning resource for understanding human errors that can occur during the requirements development process.

3.2 Study Design

This section describes the procedure followed during inspection exercises along with the data collected during the study run. The courses were chosen because they required students to learn about requirements inspections as part of their learning objectives.

3.2.1 Exercise I

Participants: 16 graduate students enrolled in the *Software Requirements Definition and Analysis* course at North Dakota State University were trained and subsequently inspected requirements for a software system for errors and corresponding faults.

Procedure: This inspection exercise was conducted over a two-week period. During the first week, students were trained on common requirement fault types and how to use fault-checklist technique to locate faults in a SRS document. Then, students used the training to inspect an externally developed document that specified requirements for a restaurant order/inventory management system called Restaurant Interactive Menu (RIM) and reported faults. **The result of this step was 16 individual fault lists.** Next, students were trained on requirements phase human errors (HET) using Reason's account of cognitive failing (slips, lapses, mistakes) and how to analyze faults to abstract human errors followed by a re-inspection of requirements document guided by the identified error information. Next, students individually analyzed their faults (found during the first step) and abstracted human error/s for each fault, and classified each abstracted error into one of the 15 human error classes of HET. **The result of this step was 16 individual error lists containing human errors that occurred during the creation of RIM requirements.** Next, the students individually re-inspected the RIM document using the identified human errors to locate new faults (that were not found during the first inspection). **This step resulted in 16 new fault lists.**

Data collected: The faults found by each student were validated for true-positive by comparing it against a list of true-positive faults known to be present in the RIM document. We removed the false-positives prior to data analysis. Similarly, the error abstraction correctness of the human errors reported by each subject was validated by comparing student's abstraction result with the abstraction results obtained by consulting a Cognitive Psychologist, Dr. Bradshaw (co-author). The data collected from each student included: (1) # of real faults found by each student during the first inspection of RIM; (2) # of human errors the student correctly abstracted and classified from the faults they found during the first step; (3) # of new faults found by the student during re-inspection of RIM using the identified errors.

3.2.2 Exercise II

Participants: 34 graduate (Master's and PhD) students enrolled in the *Software Development Processes* course participated.

Procedure: Like the first exercise, students in this study were trained on the requirements phase human errors (HET), how to abstract human errors from faults (EA instrument), and using the error information to locate faults recorded in the requirements document. The primary focus of this exercise was to evaluate whether performing an error-based inspection of an externally developed requirements document can help students understand requirement phase human errors. To evaluate that, students were

Table I. Post-study survey (Inspection Exercise II)

<i>Questions that evaluated effectiveness of error-based inspections for imparting knowledge of human errors</i>			
		N	Mean(SD) Median
Q1	I feel confident I can distinguish between a slip and a lapse.	33	3.9 (0.6) 4
Q2	I feel confident I can distinguish between a slip and a mistake	33	4.1 (0.8) 4
Q3	HET documentation had sufficient detail to allow me to understand human errors that occur during the requirements development process	33	3.8 (1) 4
<i>Questions related to educational value of human errors and error-based inspections</i>			
Q4	The effort spent in learning human errors is valuable and worthwhile in finding faults in requirements document.	33	3.97 (0.9) 4
Q5	I am confident that human errors represent real problems in the requirements development process.	33	4.2 (0.8) 4
Q6	Human error information helped me detect faults I might have otherwise overlooked	33	3.9 (0.8) 4
<i>Questions related to effectiveness of trainings provided about human errors and error-based inspections</i>			
Q7	Rate the usefulness of training while abstracting human errors from faults	33	3.7 (0.7) 4
Q8	Rate the usefulness of training when using abstracted human error information to find new faults.	33	3.8 (0.6) 4

provided with a document that specified requirements for a Parking Garage Control System (PGCS), which was seeded with 30 realistic faults. The students were then given 10 (randomly selected out of the 30 seeded) faults in PGCS and asked to analyze these 10 faults to abstract and classify human errors into one of the error class of HET. **The result of this task was 34 individual error lists containing human errors (and their classifications) that may have occurred during creation of PGCS requirements.** Next, the students were asked to re-inspect the PGCS document using the abstracted human errors with the goal of locating the remaining faults. **The result of this task was 34 individual fault lists.** The inspection exercise was followed by a survey that gathered students' feedback on the error-based inspection exercise and their understanding of the human errors and cognitive failure mechanisms that affect the requirements development process.

Data Collected: The quantitative data collected from each student included: (1) # of human errors the student correctly abstracted and classified from the 10 given faults in PGCS, and (2) # of real faults found by the student in PGCS using the error information. The quantitative data gathered from the survey is shown in Table I. Subjects rated each question on a 5-point Likert scale (ranging from "1- Strongly Disagree" to "5 – Strongly Agree" for Q1 through Q6 and ranging from "1- Not at all Useful" to "5 – Extremely useful" for Q7 and Q8). Survey questions Q1, Q2, and Q3 evaluated students' abilities to distinguish between human errors types (slips, lapses, and mistakes). Survey questions Q4, Q5, and Q6 evaluated the usefulness of HET and error-based inspections in helping students understand the requirement phase errors and the faults caused by them. Survey questions Q7 and Q8 evaluated the training usefulness on how to abstract errors and re-inspect requirements for faults. The mean (along with the standard deviation) and median scores for each question appears in Table I and discussed later.

4. RESULTS AND ANALYSIS

This section analyzes the quantitative and qualitative data collected during the two inspection exercises and is organized around the two primary objectives stated in Section 3.1.

4.1 Objective 1: Using Error Information to Detect Faults

Study 1's primary goal was to determine whether the identification of root cause of faults (i.e., human errors) supported by an error based inspection can help students discover faults that were otherwise overlooked or missed during the fault-based inspection. To help understand the effectiveness of HET and EA instruments during the error abstraction and re-inspection, we compared the number of faults found in RIM SRS document during the first inspection (using fault checklist) vs. the new faults found during

Figure 2. Comparison of number of faults detected

Figure 3. Effect of human error knowledge on fault detection effectiveness

the re-inspection (using the identified human error information). The results showed that, students found an average of 6 faults during the fault-based inspection but that number increased to an average of 14 new faults (that were not found during the first inspection) during the error-based inspection. While we certainly expected the number of new faults found during the re-inspection to be greater than zero (because of the fact that students were inspecting the same document the second time); however, an average fault effectiveness increase of 233% is noteworthy. The results from one-sample t-test (when using a comparison value of 6 faults) found this increase to be significant at $p < 0.001$.

We also analyzed whether this increase was distributed across all the subjects and the resulting comparison is shown in Figure 2. For example, Subject S5 found 8 faults during the first inspection (grey column for Subject S5); then used HET to abstract human errors and found 15 additional faults when re-inspecting using the identified error information (black column for subject S5). This increase in number of new faults found using the underlying error information is consistent across all 16 subjects. Based on this result, each subject was able to use the human error information to detect a significantly large number of new faults that were left undetected during the first inspection.

Additionally, we evaluated whether the significant increase in detection of new faults was rooted in a clear understanding of human errors (as opposed to the second review of the same artifact). To do that, we evaluated the effect the knowledge of human errors had on a student's ability to detect faults during error-based inspection. A student's knowledge of human errors was measured by calculating the percentage of human errors they correctly abstracted and classified from the faults they found during the first inspection. The percentage value was calculated by dividing the number of human errors a subject correctly abstracted and classified by the total number of human errors reported by the subject (during the error abstraction leg of RIM inspection).

We correlated the percentage of human errors correctly abstracted by the students against the number of new faults found when using human error information. The correlation was found to be strongly positive as shown in Figures 3 ($r = 0.79$ and $p = 0.0003$). This also signifies that a student's understanding of human errors can be used to predict their fault detection effectiveness during error-based inspections.

The improved fault detection effectiveness (Figure 2) and the correlation analysis (Figures 3), together, show that teaching students regarding human errors, and having them perform error-based inspections (supported by HET and EA instruments) is beneficial in improving their ability to detect errors and corresponding faults in requirements documents. As a side benefit, this exercise when repeated across different SRS documents can

also be used as a checklist of errors to avoid when developing their own requirements documents.

4.2 Objective 2: Usefulness of HET and Error Training

Another study goal was to determine whether students find error-inspections (supported by HET) a useful learning resource to understand the psychological accounts of human errors, which in turn can help train students to identify requirements errors committed during the development process. The post-study survey data collected from *Exercise II* (Table I) was used to perform this evaluation. As mentioned before, students provided feedback on survey questions using a 5-point Likert scale.

The first three questions (Table I) evaluated the students' perception about the effectiveness of error-based inspections (supported by HET) for imparting knowledge of human errors to SE students. Questions 1 and 2 evaluated whether students were able to distinguish between the human error types (slips vs. lapses vs. mistakes). As can be seen in Table I, in most cases (median = 4), the students were confident that they could distinguish between different types of cognitive failures (i.e., planning errors or execution errors). This was also consistent with their correctness of error classifications reported in individual error lists. Question 3 evaluated whether error descriptions and examples contained in the HET training document was detailed enough to help students gain an understanding of requirement human errors. A median feedback rating of four (4) indicated that, students agreed upon the benefit of using HET in understanding the human errors that can occur during the requirements development phase.

Next set of questions (Q4, Q5, and Q6 in Table I) evaluated the worthiness of effort spent in learning about human errors by performing an error inspection of the same document that they had previously inspected for faults. The quantitative results have already shown (albeit with a different group of subjects) that effort spent in learning about human errors result in a big increase in software quality (by enabling larger fault coverage). The qualitative scores (*median* and *mean* values of 4 and greater) from subjects reflect the usefulness of human error knowledge in improving the inspection performance of participating subjects. Students also reported that, the errors contained in HET represented real problems, which motivates us to use HET for developing effective *error prevention training materials* that can be used to avoid the occurrence of errors during the requirements development process.

Final set of questions (Questions Q7 and Q8) gathered feedback for improving the training and usefulness of HET and EA instruments. The training instruments provided to students could be found in [3]. The overall perception toward the trainings and instruments was found to be positive, with mean scores of above 3.7 for both the questions but more improvements can be made (e.g., adding more examples of errors for each error type).

While most of the result in this subsection reported qualitative data, we also analyzed the error abstraction quantitative data collected during *Exercise II* to evaluate: *to what extent have the students understood human error types* (slips, lapses, and mistakes). For this

analysis, the human error abstraction data provided by the students for the 10 given faults in PGCS (during Inspection Exercise II) was compared against the expected abstraction results. The expected abstraction results for the 10 given faults were obtained by consulting the Cognitive Psychology expert, Dr. Bradshaw. For observed contributions, we counted the slips, lapses, and mistakes reported by each student separately and then calculated the mean number of slips, lapses, mistakes for all students. As shown in Table II, the expected and observed contribution of slips, lapses and mistakes are quite close. The purpose of this analysis was to evaluate if error abstraction using HET is effective in helping students understand human error types (slips, lapses, and mistakes). Results show that most of the students were able to trace back faults to the correct human error type, which in turn shows that students exhibited an improved understanding of the human error types.

5. DISCUSSION

In this section, we first summarize the results organized around the two primary objectives (Section 3.1). Next, we discuss the educational value of error-based inspections with respect to the desired learning outcomes of conducting requirements inspections in SE courses.

5.1 Summary of Results

Objective 1. Results indicate that using error-based inspections supported by HET (the human error instrument used in this study) can significantly increase students' fault-detection effectiveness (Figure 2). One confounding variable here is that during error-based inspections, students inspect the same document twice, first using fault-checklist and then using abstracted errors. The first inspection might raise the fault-detection ability of the students so that they perform better during the second inspection. We were able to partially control such intellectual maturation by having a substantial time gap (8-10 days) between the first inspection of RIM and re-inspection of RIM (using human errors). Therefore, the increase in fault-detection effectiveness (shown in Figure 2) is a true demonstration of the usefulness of error-based inspections in helping students find more faults.

Objective 2. Results indicate that students found error-based inspections (supported by HET) a useful resource for learning human errors that affect the requirements development process. Specifically, students indicated that learning about human errors is valuable and useful for increasing their (students') fault-detection effectiveness. Students also indicated that after performing the inspection exercise, they were able to both understand and distinguish between human errors (slips, lapses, mistakes) that affect the people involved in any creative activity (e.g., requirements development). Regarding the usefulness of training, the students indicated that the training and learning resources provided helped them in successfully carrying out the tasks associated with error-based inspections.

5.2 Evaluation of Educational Value of Error-Based Inspections

The first desired learning outcome of conducting inspection exercises in SE courses is to help the students gain experience with validating (i.e., detecting/removing faults from) a requirements document. Our analysis has shown that using error-based inspections (supported by HET) helps students improve their ability to find larger number of faults thereby enhancing the quality of software artifact being produced.

The second desired learning outcome of conducting inspection exercises in SE courses is to help the students gain an enhanced

Table II. Students' understanding of human errors

	Slips	Lapses	Mistakes
Expected Contribution to 10 faults	3	1	6
Mean Observed Contribution (SD)	2.92 (0.76)	2.04 (0.83)	5.04 (1.04)

understanding of the format and content of a real requirements document. This learning outcome is addressed by both fault-based and error-based inspections, with error-based inspections providing students multiple opportunities to scrutinize the documents. Additionally, we provided students with different types of document (student-developed RIM SRS and industrial-strength PGCS SRS) that included the type of errors and faults that are likely to be committed by both students and industry practitioners.

The third desired learning outcome of conducting inspection exercises in SE courses is to help the students in improving their understanding of faults that are commonly found in requirements documents. Error-based inspection provides students with the opportunity to learn not only about the faults that commonly occur, but also the opportunity to learn about the cognitive failures that occur during the requirements development and lead to faults getting injected into formal requirements documents. This can also help students in avoiding committing these errors themselves when developing their own requirements document or when they begin their career in industry.

6. CONCLUDING REMARKS

The application of human error knowledge to improve human reliability has been recognized and successfully utilized in various domains like aviation and medicine. In this paper, we have described the use of error-based inspections (supported by HET) for providing human error training to SE students, with a particular focus on the human errors that affect requirements development process. Two error-based requirements inspection exercises were conducted in graduate-level SE courses. The students provided feedback about the effectiveness of error-based inspections for imparting human error knowledge, and usefulness of human errors in increasing students' fault detection abilities.

Overall, students indicated that they are highly confident that error-based inspections are a useful learning resource for learning about human errors that affect the requirements development process. The students also indicated that they found the knowledge of human errors both meaningful and useful in detecting those faults in requirements documents that they usually overlook when just focusing on the fault information.

Results also showed that using human error information significantly improved the fault detection effectiveness of students. In many cases, the fault-detection effectiveness of students increased by a factor of two or three when they used human error information to re-inspect the requirements (Figure 2). This shows that error-based inspection and human error knowledge are beneficial in increasing students' ability to validate requirements documents, a primary learning outcome of inspection exercises.

From an industry skill perspective, the students also get to learn the fault-checklist inspections as a part of error-based inspections. This is because fault-based inspections are a pre-requisite to error-based inspections. That is, fault-based inspections need to be carried out first, in order to initiate the error discovery. Fault-based inspections are the most widely used inspection technique in the industry. Hence, it is important that SE students learn fault-based inspections. From a course instructor's perspective, there is no need for them to make changes to their existing coursework (i.e., assignments and projects). Error-based inspections deliver human error knowledge by simply replacing the conventional fault-based inspections (which already are an integral part of many SE courses).

These inspection studies were performed in graduate-level SE courses. Our future work includes performing studies with undergraduate SE and Computer Science students. We also plan to improve the training material on human errors based on the subjective feedback obtained from students during the two studies.

7. ACKNOWLEDGMENTS

This study was supported by National Science Foundation Awards 1423279 and 1421006. Our thanks to the students of the two graduate courses who participated in the inspection exercises and the instructors of the two courses.

8. REFERENCES

- [1] Anu, V., Walia, G.S., Hu, W., Carver, J.C. and Bradshaw, G. 2016. Effectiveness of Human Error Taxonomy during Requirements Inspection: An Empirical Investigation. *2016 International Conference on Software Engineering and Knowledge Engineering, SEKE* (San Francisco, CA, USA)
- [2] Diller, T., Helmrich, G., Dunning, S., Cox, S., Buchanan, A. and Shappell, S. 2014. The Human Factors Analysis Classification System (HFACS) applied to health care. *American jour. of medical quality : the official journal of the American College of Medical Quality*. 29, 3 (2014), 181–90.
- [3] Error-Based Inspection of PGCS Requirements Using Human Error Taxonomy (HET): 2015. http://vaibhavanu.com/SDP_Inspection.html
- [4] Lanubile, F., Shull, F. and Basili, V.R. 1998. Experimenting with Error Abstraction in Requirements Documents. *In Proceedings of the 5th International symposium on software metrics* (Bethesda, MD, USA, 1998).
- [5] Norman, D.A. 2002. *The Design of Everyday Things*.
- [6] Patey, R., Flin, R., Cuthbertson, B.H., MacDonald, L., Mearns, K., Cleland, J. and Williams, D. 2007. Patient safety: helping medical students understand error in healthcare. *Quality & safety in health care*. 16, 4 (2007), 256–259.
- [7] Pyster, A. 2009. *Graduate Software Engineering 2009 Curriculum Guidelines for Graduate Degree Programs in Software Engineering*. ACM.
- [8] Rasmussen, J. and Vicente, K.J. 1989. Coping with human errors through system design: implications for ecological interface design. *International Journal of Man-Machine Studies*. 31, 5 (1989), 517–534.
- [9] Reason, J. 1990. *Human error*. Cambridge University Press.
- [10] Reason, J. 1997. *Managing the risks of organizational accidents*. Ashgate.
- [11] Shappell, S.A. and Wiegmann, D.A. 2000. The Human Factors Analysis and Classification System – HFACS. *Security*. (2000), 19.
- [12] Walia, G.S. and Carver, J.C. 2009. A systematic literature review to identify and classify software requirement errors. *Information and Soft. Technology*. 51, 7 (2009), 1087–1109.
- [13] Walia, G.S. and Carver, J.C. 2013. Using error abstraction and classification to improve requirement quality: Conclusions from a family of four empirical studies. *Empirical Software Engineering*. 18, 4 (2013), 625–658.
- [14] Wiegmann, D. and Detwiler, C. 2005. Human Error and General Aviation Accidents : A Comprehensive , Fine-Grained Analysis Using HFACS. *Security*. (2005), 1–5.