Applying Reason (1990): Identifying the causes of user errors

Trevor J. McIntyre, Michael J. Crites, Noah J. Wheeler, & Samuel J. Levulis

TEXAS TECH UNIVERSITY

Introduction

Usability and Design, Divorced

In practice, usability evaluations fail to influence design for several reasons (Hornbaek & Stage, 2006). Feedback from evaluations meant for designers and engineers (reports) are perceived to have low utility by designers and engineers. Even though the field of usability has matured, one of the strongest, most useful forms of feedback is still having designers and other stakeholders observe the test. Reports are failing to effectively communicate why issues occurred or what needs to be done about them (Molich, Ede, Kaasgaard, and Karyukin, 2004). Designers perceive reports to have low utility because they often do not indicate the generalizable cause of the error or any design activity that could have influenced the design to prevent the error.

The Culture of a "Test-and-Fix" Approach

Usability evaluations are the most frequently conducted design activity (Vredenburg, Mao, Smith, & Carey, 2002), suggesting some practitioners are either attempting to test their way to good design (a practice criticized in Hartson & Pyla, 2012) or that errors uncovered during testing are not being seen as a failure to conduct a particular front-end design activity (e.g., user requirements analysis).

Specific Purpose

We assert that classifying usability errors according to an adaptation of Reason's (1990) GEMS allows practitioners to:

- Directly address root causes of usability issues
- Provide more guidance about the specific cause of the error to designers and engineers
- Suggest design activities that could address/prevent such errors in the future

Resulting in:

- Higher evaluation report utility by designers and engineers
- A greater appreciation for the necessity of front-end design activities

Errors in Usability

Error as a Usability Component

- Nielsen (1993) included error as one of five components of usability, leading to a field-wide focus.
- Error classification schemes used for evaluations are often made-up specifically for a single test or use standard scales which correspond to dimensions of frequency or severity (Dumas & Redish, 1999; Rubin, 1994; Nielsen, 1993).
- Metrics of frequency or severity, while potentially useful, do not by themselves illuminate the cause of the error nor do they give any indication of design activities that could have prevented the error.

References

Boren, T., & Ramey, J. (2000). Thinking aloud: Reconciling theory and practice. *IEEE Transactions on Professional Communication*, 43(3), 261-278.

Dumas, J. S. & Redish, J. C. (1999). *A practical guide to usability testing*. Wilmington, NC: Intellect Ltd.

Ericsson, K. A., & Simon, H. A. (1984). *Protocol analysis: Verbal reports as data*. MIT press.

Fu, L., Salvendy, G., & Turley, L. (2002). Effectiveness of user testing and heuristic evaluation as a function of performance classification. *Behaviour & Information Technology*, 21(2), 137-143.

Hartson, R., & Pyla, P. S. (2012). *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier.

Hornbaek, K., & Stage, J. (2006). The interplay between usability evaluation and user interaction design. *International Journal of Human-Computer Interaction*, 21(2), 117-123.

Mack, R., & Montaniz, F. (1994, June). Observing, predicting, and analyzing usability problems. In *Usability inspection methods* (pp. 295-339). New York, NY: Wiley.

Mayhew, D. J., & Follansbee, T. J. (2012). User experience requirements analysis within the usability engineering lifecycle. In Jacko, J. A. (Ed.), *Human-computer interaction handbook: fundamentals, evolving technologies, and emerging applications* (pp. 945-953). CRC Press.

Mendoza, V., & Novick, D. G. (2005, September). Usability over time. In *Proceedings of the 23rd Annual International Conference on Design of Communication- Documenting & Designing for Pervasive Information* (pp. 151-158). ACM.

Molich, R., Ede, M. R., Kaasgaard, K., & Karyukin, B. (2004). Comparative usability evaluation. *Behaviour & Information Technology*, 23(1), 65-74.

Nielsen, J. (1993). *Usability Engineering*. San Diego, CA: Morgan and Kaufman.

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. New York, NY: Basic Books.

Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics*, (3), 257-266.

Reason, J. (1990). *Human error*. Cambridge university press.

Rubin, J. (1994). *Handbook of usability testing: How to plan, design, and conduct effective tests*. New York, NY: Wiley.

Senders J, Moray N. (1991). *Human Error: Cause, Prediction, and Reduction*. Mahwah, NJ:Lawrence Erlbaum.

Thorndike, E. L. (1898). Animal intelligence: An experimental study of the associative processes in animals. *Psychological Monographs: General and Applied*, 2(4), i-109.

Vredenburg, K., Mao, J. Y., Smith, P. W., & Carey, T. (2002, April). A survey of user-centered design practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 471-478). ACM.

Techniques to Identify Causes of Error

Think-aloud Protocol (Boren & Ramey, 2000; Ericsson & Simon, 1984)

Widely used method to understand users' motivations for action during testing. Using this method, practitioners have users speak their thoughts during testing.

Issues with Method:

Nørgaard & Hornbæk (2006) found that, in practice, this method is biased towards problems the evaluator already knows about. Evaluators tend to explore problems they have foreseen or have deemed more severe.

Affinity Diagramming (originator: Jiro Kawakita; in user experience: Beyer & Holtzblatt, 1998)

The practitioner conducting this method clusters errors and issues together that are reasoned to be similar in nature, creating a hierarchical structure of overall issues.

Issues with Method:

While the method attempts to achieve a higher-level understanding of the cause of error production, its subjectivity is likely to lead to inconsistencies across tests and teams, as Molich et al. (2004) has shown.

Root Cause Analysis (as detailed by Norman, 2013)

Norman (2013, p. 42) simplifies "root cause analysis" as "asking 'why' until the ultimate, fundamental cause of the activity is reached."

Issues with Method:

This activity, by its very definition, is likely to reveal what caused a user to commit an error, but it still may not be structured or prescriptive enough to reveal design activities that would address the error.

It follows from the previous investigation that an ideal classification system for the usability practitioner would not only determine the cause of the error, but also indicate what design activity to conduct in order to reduce errors of that type in the future.

Reason's (1990) GEMS Framework

Table 1. The GEMS framework adapted from Reason (1990)

Error Type	Description	Example
Skill-based Slip	Incorrect execution of a planned action	Missed key during typing
Skill-based Lapse	Incorrect omission of a stored, planned action	Going to the kitchen to make tea; instead making coffee
Rule-based Mistake	Applying an incorrect "if-then" rule	A toddler, with previous iPad experience, swipes to advanced the page forward in a physical book
Knowledge-based Mistake	Incorrect reasoning due to a lack of stored rules for how to deal with the novel situation	Patient presents with collection of symptoms the doctor has never seen; diagnosis based on more familiar subset of symptoms

The Solution

Background of the Solution

- Our approach is based on Reason's (1990) generic error-modeling system (GEMS) (Table 1).
  - Classifies errors based on the level of behavior the user is operating under when committing the error: skill-based, rule-based, or knowledge-based
  - Based on Rasmussen's (1986) SRK framework
- Researchers have used Rasmussen's (1986) approach as a framework for an error classification scheme (Fu, Salvendy, & Turley, 2002; Mack & Montaniz, 1994; Senders & Moray 1991).
- In contrast to our usability evaluation-based approach, their use was incorporated into an inspection-based method (i.e., expert evaluation).

Table 2. Reason's (1990) GEMS framework adapted for usability evaluation

Error Type	Known error characteristics	Recommendation	Design Activity
Skill-based Slip	More likely to be encountered by expert; Occurrence likely decreases with time	Improve success of actions by making them easier to physically execute. Always provide feedback for actions.	•Literature Review •Further usability evaluation (consider experts)
Skill-based Lapse	More likely to be encountered by expert; Occurrence likely decreases with time	Provide some reminder of the user's initial goal, if possible. User's often correct a lapse when their initial goal isn't achieved.	•Artifact analysis (to determine how users remind themselves currently)
Rule-based Mistake	More likely to be encountered by novice; Occurrence likely remains stable over time	Ensure the interface doesn't look similar to something else that works differently. Ensure that the task models users' expectations.	•Task analysis •Contextual Inquiry and Interviews
Knowledge-based Mistake	More likely to be encountered by novice at first exposure; Occurrence in expert users likely increases over time as they explore what can and cannot be done with the product (Mendoza & Novick, 2005)	Ensure users understand that they can achieve their goal with the product and also understands how. Capture frustrations and feature requests with the current design and build solutions into the next version.	•User Profile (due to possible assumptions made about the user) •Participatory design •Competitor analysis (to get a sense of common patterns in competing products) •Think-aloud or recall session

Specific Contribution

- Table 2 is a general implementation of this framework. Due to the domain-specific, task-specific nature of usability evaluation, the table can be adapted to better fit a particular evaluation.

Classifying errors in this framework allows practitioners to:

- Apply known characteristics about that type of error
- Apply recommendations specific to that type of error
- Propose design activities to help remedy the current error and prevent errors of that type from occurring again in the future.

Future Directions

The framework needs to be evaluated to determine what value is added over traditional forms of error classification schemes, if any, when considering the entire user-centered design lifecycle. In addition, it remains to be seen whether designers and engineers would perceive reports based on this framework as having higher utility and whether organizational demeanor, such as the acceptance of front-end design activities, would change after implementing such reports.