

**Debugging Away From Failure**  
**Or How To Avoid Unrecoverable Loss**

John Bartucz

Department of Curriculum and Instruction

University of Minnesota

EPSY 5124 - Debugging Failure

Dr. David DeLiema

December 22, 2021

## **Abstract**

*Failure* means many things to many people. From a low grade on a math test in elementary school to a space shuttle explosion, the word has different meanings and consequences for all of the individuals and groups involved. How can a single event be placed at so many different places on the continuum from success to failure by so many different people? This paper discusses the idea that what is seen as a single failure event is actually a timeline of *faults* and *errors*, which lead up to a point at which there is a feeling of an unrecoverable loss of opportunity. Each participant and observer views this timeline through their own multiple lenses of expectations. It is possible, however, to avoid the feeling of unrecoverable loss by allowing space for “debugging” between error and failure.

## Introduction - What is failure?

The word *failure* means many different things in many different contexts and is notoriously hard to define (Clifford, 1984; Feltham, 2014; D. DeLiema, personal communication, Sep. 22, 2021). Even with a single well-known and well-documented event that is universally described as a failure, such as the burning of the Notre Dame Cathedral, it is impossible to pinpoint exactly one ultimate failure. Instead it exists as a timeline of errors (Peltier et al, 2019), and even with a hypothetically perfectly-accurate record of all related events, every person would come up with their own version of what the point of failure actually is. One person might say that the failure occurred when fire expanded out of control; one might say when the fire suppression system did not work; one might say when the security officers did not call the fire department in time; one might simply say “when the cathedral burned”. How is it possible that such a unique and well-documented event has such a wide discrepancy in its main descriptor as *failure*?

The simple reason is that failure is not an event, but a feeling. Despite millions of people viewing the burning of Notre Dame as a failure, their own unique knowledge, experience, and culture will define what that means for them. A child who has only seen the cathedral in cartoons will have a much different view than an elderly Parisian who has attended service every day for decades. Despite this huge variance, there is still a common thread in all of these very personal interpretations. The first part of the common thread is paradoxically that failure is a uniquely personal experience. The second part of the common thread is that there is a feeling of loss. And finally, what fully defines the feeling of failure is that the loss is unrecoverable. Whether it is a failed exam,

a missed penalty shot, or a burned cathedral, the feeling of failure necessarily includes the feeling that the result can not be changed.

This paper puts forth the definition that the moment of failure is when an individual has “the first-person feeling of an unrecoverable loss of the potential for success.” It is all the more powerful if the individual feels personally responsible.

### Errors vs Failures

Clearly the critical word in this definition is “unrecoverable”. If we have many chances to succeed, any single one that is missed does not constitute failure. Unrecoverable is what separates “failure” from “error”. Reason (1990) produced the following figure to describe the different kinds of errors:

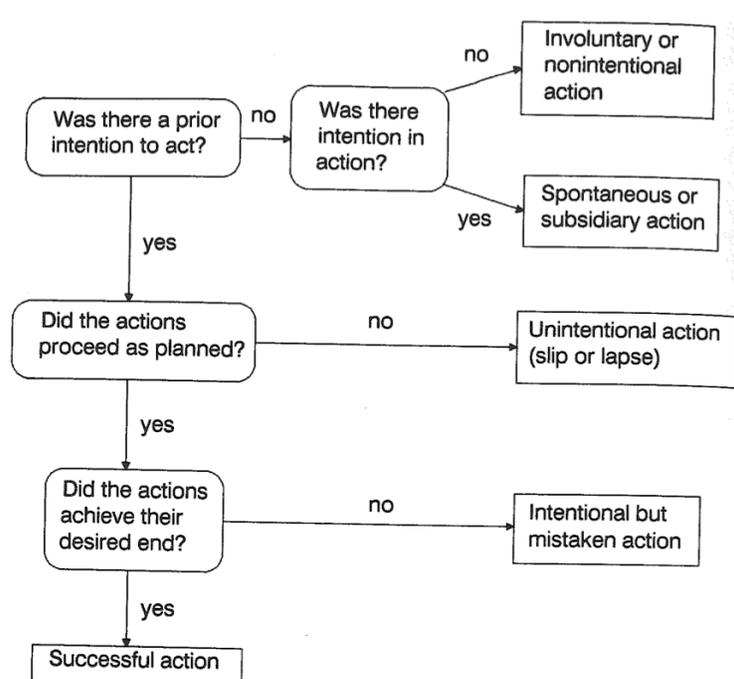


Fig. 1 - Algorithm for distinguishing the different kinds of intentional behavior (Reason, 1990, p. 6)

The right-hand side of the diagram clearly shows the different kinds of errors and their causes. However, none of them individually or even collectively can be used as a definition of failure. The reason that they are not failures is that they are missing the results. If any of the right-hand-column events occur, but there is no discernible change in outcome, there is no failure. However, without one of the right-hand-column errors, there can be no failure. So what is the link between the two?

To borrow an idiom from systems engineering, the general flow of failure has three stages (International Organization for Standardization, 2011):

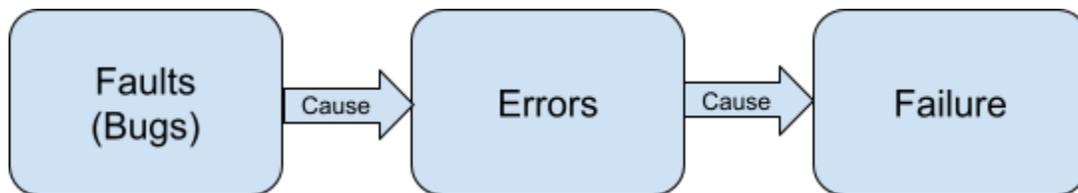


Fig. 2 - Flow chart depicting the causal chain from Faults to Errors to Failure

Faults can be categorized very similarly to the left-hand boxes as shown in Fig. 1 above. Were faults introduced through a lack of understanding (of the problem or the tools)? An unintentional slip or lapse (for example, a typo)? Something else? In software engineering in particular, faults are often described as “bugs” and can have many causes. Ko and Myers (2005) created a ‘Swiss cheese’-model of the causes of faults in software systems.

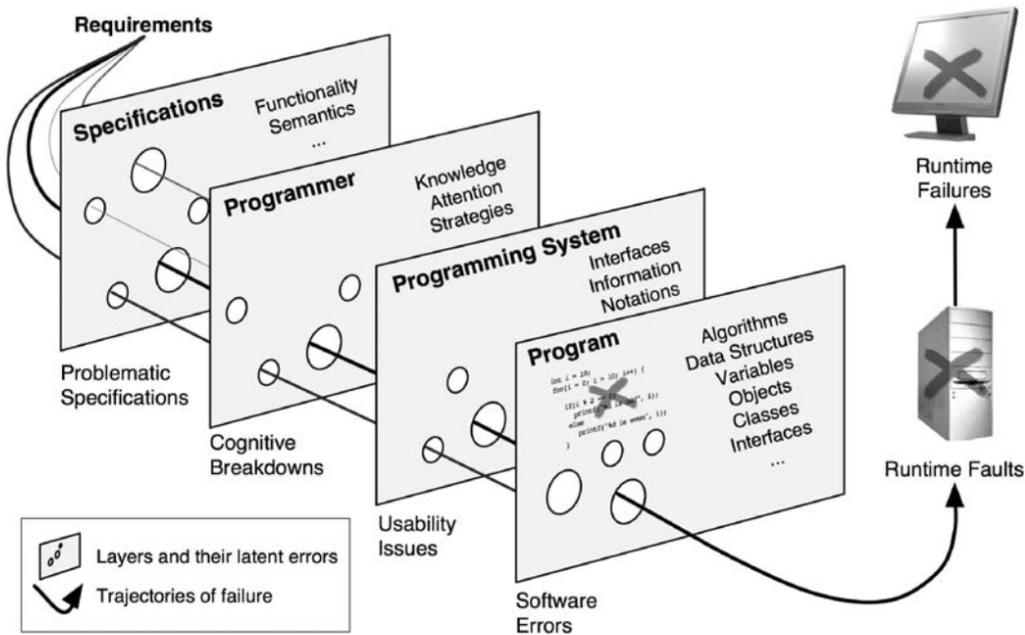


Fig. 3 - Dynamics of software error production, based on Reason’s systemic view of failure (Ko and Myers, 2005)

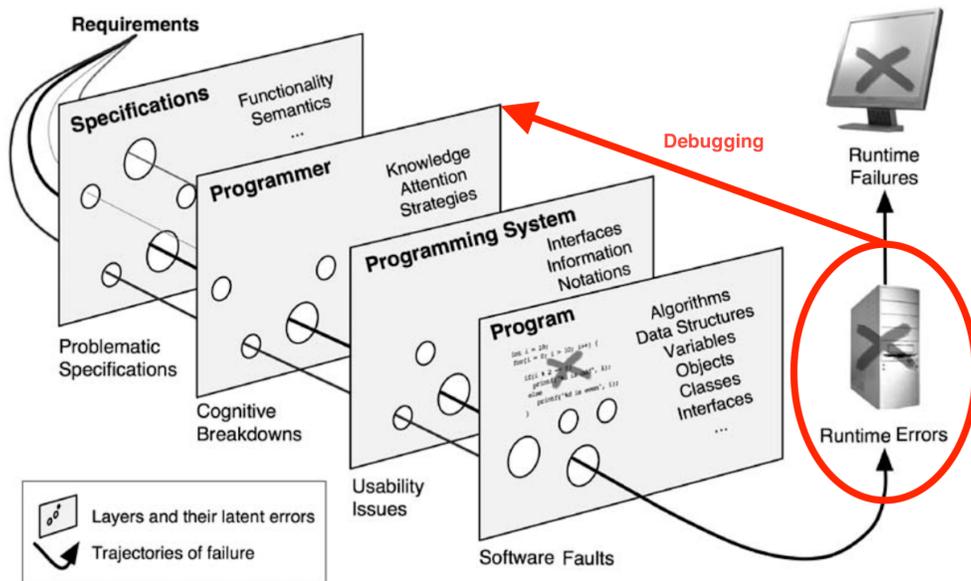
In this diagram, the authors use the terms “Software Errors” and “Runtime faults” in the opposite way to the ISO standard listed in Fig. 2, but the underlying meaning holds. Instead of the layers of swiss cheese attempting to filter out “software errors”, they are trying to eliminate faults (bugs), because those faults will lead to runtime errors and potentially an ultimate system failure (not doing what it is supposed to do, or doing something it is not supposed to do).

Why, then, can we not devise a perfect block of cheese which filters out all possible faults and eliminates runtime errors? Unfortunately, it is not possible to develop a perfectly fault-tolerant system given finite resources (Schlichting and Schneider, 1983). It is not possible to predict all the possible variations of human error in advance, and

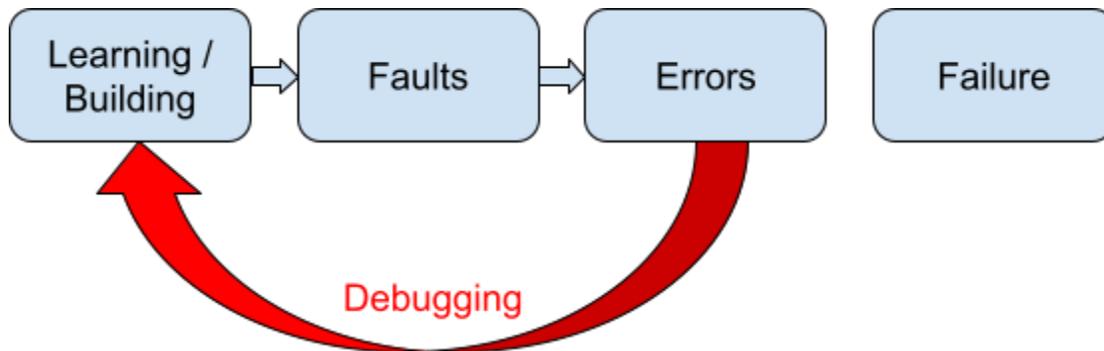
therefore it is not possible to develop a Ko-like-model in which none of the holes match up at any time. In addition, “as software systems become increasingly large and complex, the difficulty of detecting, diagnosing, and repairing software problems has also increased.” (Ko and Myers, 2005) It seems we are doomed to failure.

### Debugging to avoid failure

All of that being said, there is one extremely important arrow missing from all of these diagrams, and that is the idea of “debugging”. If the primary difference between “errors” and “failures” is the point at which the outcome is “unrecoverable”, this implies that at some point, the process was still “recoverable”. Looking again at the Ko and Myers diagram (with the faults/errors locations switched to match convention), we can add a debugging arrow which can occasionally reverse the flow:



This allows the programmer to avoid failure at the error stage and means that each error becomes an impetus for learning instead of failure.



Given an infinite amount of time and testing, the Failure stage will never be reached. But as Schlichting and Schneider wrote, we do not have an infinite amount of time or resources (1983).

What we can learn from this is that debugging, contrary to popular belief, is not evidence of failure, but evidence of learning how to avoid failure. And it is precisely this - the constructive avoidance of failure, that should be appreciated. Building robust systems which are designed to minimize errors is critical, but it is equally critical to understand that these systems will always be made of swiss cheese and the ability to debug is not only invaluable, it is exactly what we mean when we are talking about the popular terms “grit” and “growth mindset”.

### **Setting expectations to avoid the feeling of failure**

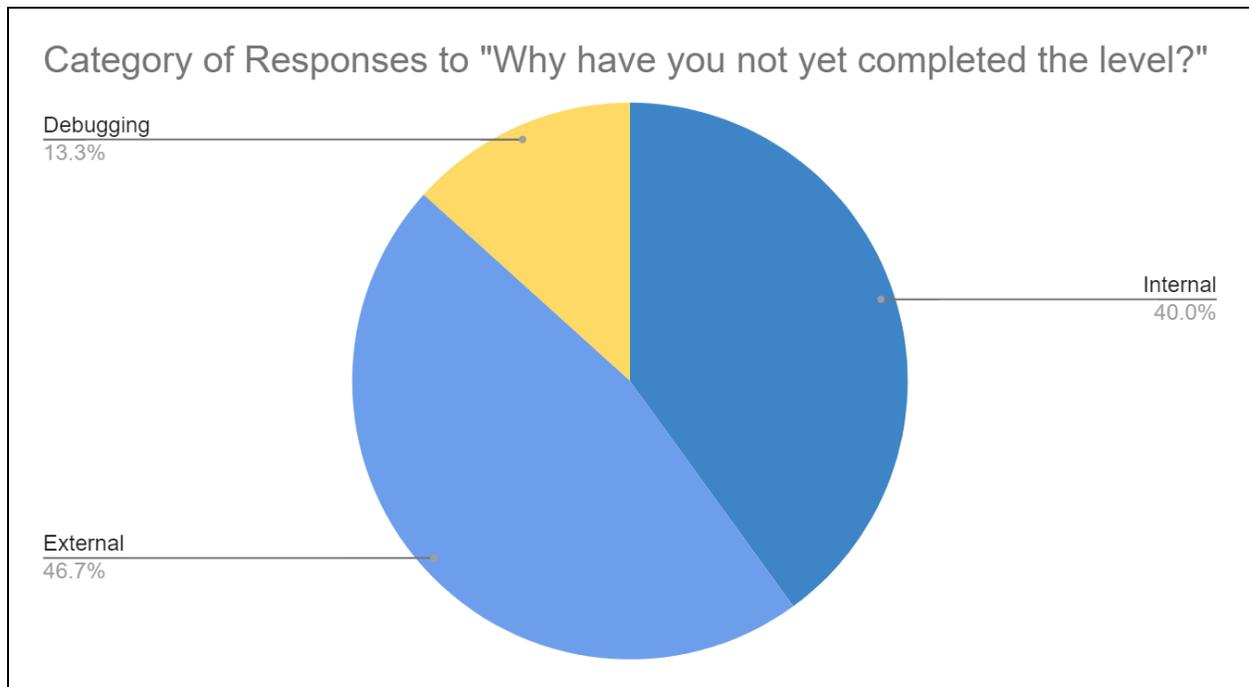
If the feeling of failure is so negative, why is there so much literature on constructive failure? (Bransford, 1999; Clifford, 1984; Feltham, 2014; Kapur 2008, 20016; Juul

2013; Reason, 1990). The reason is that most of these papers confound the stages of error and failure. Consider the case of Idit in Heyd-Metzuyanim (2105). Idit was initially very successful in mathematics, but as she entered 9th grade, she receded. At what point does Idit's errors in math transition into "failure"? It is only when she herself decides that her lack of ability is unrecoverable. Even though she did not have perfect scores in mathematics in elementary and middle school, she earned scores in the high 80-90% range (Heyd-Metzuyanim, 2015). This was acceptable to her because those scores met her (and her parents') expectations. As she got into high school, her scores began to drop below those expectations and she began to rely on the narrative that "she isn't good at fractions." Despite understanding the material during class and at home, she was unable to perform on tests. According to Heyd-Metzuyanim, "stories similar to Idit's appear to be common, especially with girls who study in traditional mathematics classrooms." The expectations of a certain level of performance were causing the students to change their perspective from "errors that are acceptable, that can be fixed/debugged, and are learning experiences" to "failure that is unrecoverable." This, in turn, led them to rely even more on the narrative that they are "not good at math" in order to lessen the expectations, which leads to a cycle of learned helplessness (Clifford, 1984).

### **Classroom Research**

In my research using the game "Baba Is You" (n = 15), students were intentionally pushed to levels they could not finish and then asked why they felt that they were unable to complete those levels. In coding the responses, I created three distinct categories of causal attribution. Despite the low number of participants and the undeniable

possibility of attributional errors (Bennett, 2017), one interesting result appeared which relates directly to this discussion.



Category 1, "Internal": for 6 out of the 15 respondents, answers fell into this category. Typical responses to the question "Why do you think you haven't completed this level yet?" in category 1 would be "I'm not smart enough", "I have no idea what I'm doing", or "I just don't get it." Category 2, "External": for 7 out of the 15 respondents typical responses would be "the level is too hard", "there are no instructions", or "the game is really confusing." Category 3, with only two responses, was the most interesting. The two responses were "I just haven't figured it out yet." and "I just need a little more time." These responses not only demonstrate the ideal response in a growth-mindset / productive failure classroom, they are exactly demonstrating the thought processes of students who are debugging (Alderman et al, 2015).

## **Conclusion and discussion**

Two common barriers prevent students from progressing in their learning. One is the perception that they have reached a point from which their goal is unrecoverable. We have used this definition as the point of failure. When students feel that they have failed at a task, especially when they fail at related tasks repeatedly, they are in danger of protecting themselves by lowering their expectations and the expectations of their peers, parents and teachers (Mikulincer, 1994). The second barrier is the lack of instruction in the field of debugging. Although debugging has been widely studied as it pertains to computer science (McCauley et al., 2008), it is a tactic which benefits students' general problem-solving skills and self-efficacy (Ahn et al., 2021). In fact, as this paper shows, the strategy of debugging problems can reroute students' perception of not achieving an immediate goal from an unrecoverable failure to an error which can be addressed. If the expectations of the student are initially set such that encountering errors is inevitable and debugging strategies are explicitly taught, it is possible that this will lead specifically to the benefits that a growth mindset offers (Yeager et al., 2019).

This is not to say that teaching students the strategies of debugging is a magic bullet. Teaching and learning debugging is hard (McCauley et al., 2008), successfully teaching growth mindsets is hard (Miller, 2019), and converting curricula to allow for errors and debugging instead of failure is hard (Wormeli, 2018). That being said, it is possible to allow students to find space, to manage their expectations, and to teach them to find and fix their errors before they feel that their goal has been irrecoverably lost. Debugging gives students a chance to avoid failure.

## References

- Ahn, J., Sung, W., & Black, J. B. (2021). Unplugged Debugging Activities for Developing Young Learners' Debugging Skills. *Journal of Research in Childhood Education*, 1–17. <https://doi.org/10.1080/02568543.2021.1981503>
- Alderman, J., Brain, T., Choi, T., & Pereira, L. (2015). *A Field Guide to Debugging*. A Field Guide to Debugging | p5.js. Retrieved December 23, 2021, from <https://p5js.org/learn/debugging.html>
- Bennett, K, Nelson, T. D. (Ed.). (2017). Chapter 20: CAUSAL ATTRIBUTIONS & SOCIAL JUDGMENTS. In *Getting grounded in social psychology: The essential literature for beginning researchers*. Routledge, Taylor & Francis group.
- Bransford, & Schwartz, D. L. (1999). Rethinking transfer: A simple proposal with multiple implications. *Review of Research in Education*, 24, 61–100. <https://doi.org/10.3102/0091732x024001061>
- Clifford, M. M. (1984). Thoughts on a theory of constructive failure. *Educational Psychologist*, 19(2), 108–120. <https://doi.org/10.1080/00461528409529286>

Feltham, C. (2014). *Failure*. Routledge, Taylor & Francis.

<http://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=1779>

[156](#)

Heyd-Metzuyanim, E. (2015). Vicious Cycles of Identifying and Mathematizing: A Case Study of the Development of Mathematical Failure. *Journal of the Learning Sciences*, 24(4), 504–549.

<https://doi.org/10.1080/10508406.2014.999270>

International Organization for Standardization. (2011). Road vehicles — Functional safety (ISO Standard No. 26262-1:2011).

<https://www.iso.org/obp/ui/#iso:std:iso:26262>

Juul, J. (2013). *The art of failure: An essay on the pain of playing video games*. MIT Press.

Kapur, M. (2008). Productive Failure. *Cognition and Instruction*, 26(3), 379–424.

<https://doi.org/10.1080/07370000802212669>

Kapur, M. (2016). Examining Productive Failure, Productive Success, Unproductive Failure, and Unproductive Success in Learning. *Educational Psychologist*,

51(2), 289–299. <https://doi.org/10.1080/00461520.2016.1155457>

Koschmann, T., Kuutti, K., & Hickman, L. (1998). The Concept of Breakdown in Heidegger, Leont'ev, and Dewey and Its Implications for Education. *Mind, Culture, and Activity*, 5(1), 25–41.

[https://doi.org/10.1207/s15327884mca0501\\_3](https://doi.org/10.1207/s15327884mca0501_3)

McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: A review of the literature from an educational perspective. *Computer Science Education*, 18(2), 67–92.

<https://doi.org/10.1080/08993400802114581>

Mikulincer, M. (1994). *Human Learned Helplessness*. Springer US.

<https://doi.org/10.1007/978-1-4899-0936-7>

Miller, D. I. (2019). When Do Growth Mindset Interventions Work? *Trends in Cognitive Sciences*, 23(11), 910–912. <https://doi.org/10.1016/j.tics.2019.08.005>

Peltier, E., Glanz, J., Gröndahl, M., Cai, W., Nossiter, A., Alderman, L.

(2019, July 18). Notre-Dame came far closer to collapsing than people knew. This is how it was saved. The New York Times.

<https://www.nytimes.com/interactive/2019/07/16/world/europe/notre-dame.html>

Reason, J. T. (1990). *Human error*. Cambridge University Press.

Schlichting, R. D., & Schneider, F. B. (1983). Fail-stop processors: An approach to designing fault-tolerant computing systems. *ACM Transactions on Computer Systems*, 1(3), 222–238. <https://doi.org/10.1145/357369.357371>

Wormeli, R. (2018). *Fair isn't always equal, second edition: Assessing and grading in the differentiated classroom* (Second edition). Stenhouse Publishers.

Yeager, D. S., Hanselman, P., Walton, G. M., Murray, J. S., Crosnoe, R., Muller, C., Tipton, E., Schneider, B., Hulleman, C. S., Hinojosa, C. P., Paunesku, D., Romero, C., Flint, K., Roberts, A., Trott, J., Iachan, R., Buontempo, J., Yang, S. M., Carvalho, C. M., ... Dweck, C. S. (2019). A national experiment reveals where a growth mindset improves achievement. *Nature*, 573(7774), 364–369. <https://doi.org/10.1038/s41586-019-1466-y>