Human Error and Mitigation Strategies

DANIELE MAZZEI - IUM 22-23 (all rights reserved)

Human Error

Most industrial accidents are caused by human error: estimates range between 75 and 95 percent.

How is it that so many people are so incompetent?

Answer: **They aren’t. It’s a design problem.**

 (all rights reserved)

Why?

We design equipment that requires people to be fully alert and attentive for hours, or to remember archaic, confusing procedures even if they are only used infrequently, sometimes only once in a lifetime.

We put people in boring environments with nothing to do for hours on end, until suddenly they must respond quickly and accurately.

Or we subject them to complex, high-workload environments, where they are continually interrupted while having to do multiple tasks simultaneously.

DANIELE MAZZEI - IUM 22-23 (all rights reserved)

Why?

Interruptions are a common reason for error, not helped by designs and procedures that assume full, dedicated attention yet that do not make it easy to resume operations after an interruption.

 (all rights reserved)

Human attitude towards errors

“We caught the culprit.”

But it doesn’t cure the problem: the same error will occur over and over again. Instead, when an error happens, we should determine why, then redesign the product or the procedures being followed so that it will never occur again or, if it does, so that it will have minimal impact.

 (all rights reserved)

ROOT CAUSE ANALYSIS

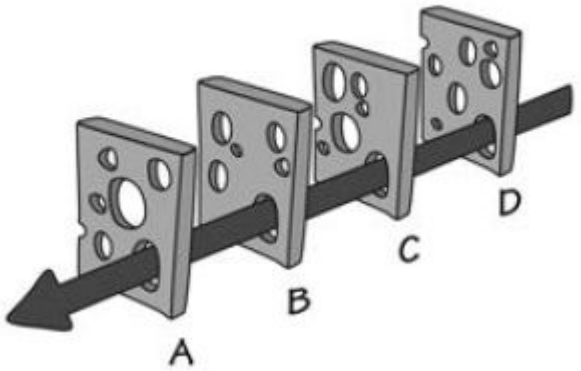
investigate the accident until the single, underlying cause is found.

What this ought to mean is that when people have indeed made erroneous decisions or actions, we should determine what caused them to err.

This is what root cause analysis ought to be about. Alas, all too often it stops once a person is found to have acted inappropriately.

most accidents do not have a single cause:
there are usually multiple things that went wrong

This is what James Reason has called the
“Swiss cheese model of accidents”



 (all rights reserved)

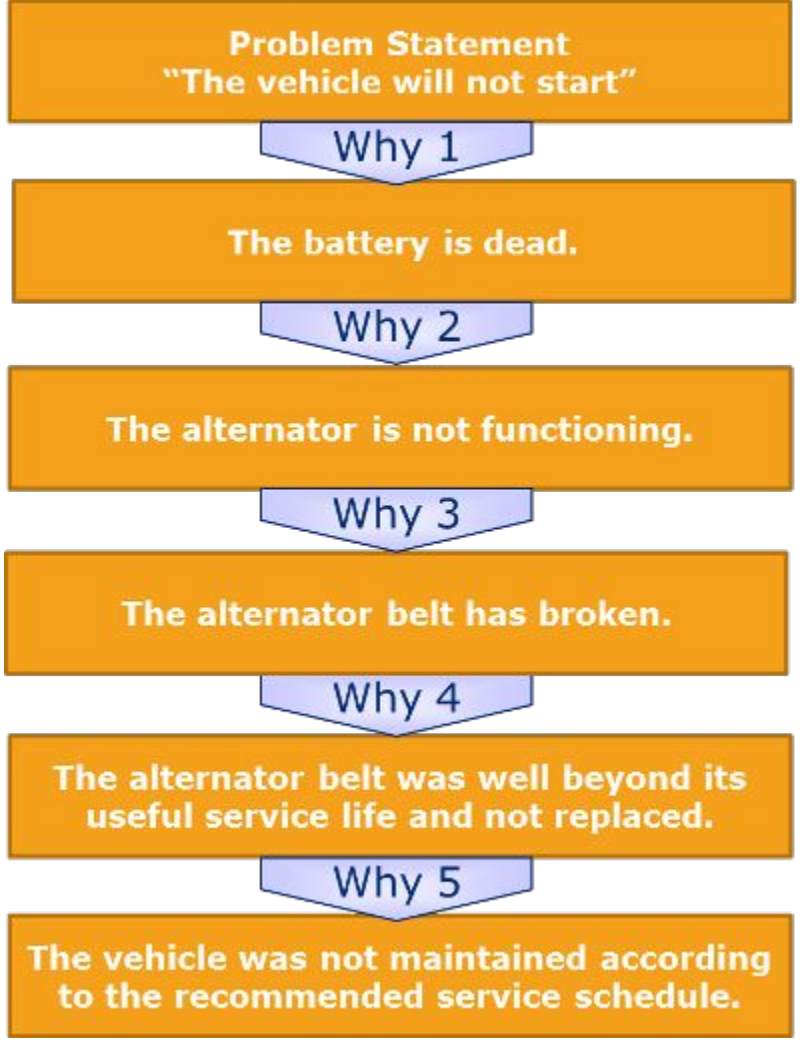
THE FIVE WHYS

originally developed by Sakichi Toyoda and used by the Toyota Motor Company as part of the Toyota Production System for improving quality.

Today it is widely deployed. Basically, it means that when searching for the reason, even after you have found one, do not stop: ask why that was the case.

And then ask why again. Keep asking until you have uncovered the true underlying causes.

Does it take exactly five? No, but calling the procedure “Five Whys” emphasizes the need to keep going even after a reason has been found.



 (all rights reserved)

5 REASONS TO USE 5 WHYS

IDENTIFY THE CAUSE, NOT JUST THE SYMPTOMS

Dig deep and find the underlying issues that led to the problem rather than using a quick-fix solution or playing the blame game.

PERFORM AN EVIDENCE-BASED ANALYSIS

Don't assume or jump to conclusions about the source of the problem - make sure you have proof that it's the cause, every step of the way.

ELIMINATE ISSUES IN YOUR SYSTEM FOR GOOD

Be proactive rather than reactive. When issues arise, prevent their reoccurrence to save time and increase the quality of your system.

SEEK IMPROVEMENTS AND WELCOME CHANGE

Encourage your stakeholders to constantly seek ways to improve and adapt your process to ensure its long-term success.

BUILD A CULTURE THAT EMBRACES PROGRESS

Encourage your team to raise issues and concerns without fear or judgement, and to seek long-term solutions rather than the easy way out.

POWERNOODLE®
BETTER QUALITY DECISIONS

POWERNOODLE.COM/MODELS-ARTICLE/5-WHYS

DANIELE MAZZEI - IUM 22-23 (all rights reserved)

People attitude toward errors

We can’t fix problems unless people admit they exist.

When we blame people, it is then difficult to convince organizations to restructure the design to eliminate these problems.

 (all rights reserved)

Why do people err?

Because the designs focus upon the requirements of the system and the machines, and not upon the requirements of people.

Most machines require precise commands and guidance, forcing people to enter numerical information perfectly.

But people aren’t very good at great precision.

People are creative, constructive, exploratory beings. We are particularly good at novelty, at creating new ways of doing things, and at seeing new opportunities. Dull, repetitive, precise requirements fight against these traits.

 (all rights reserved)

DEFINITIONS: ERRORS

Human error is defined as any deviance from “appropriate” behavior.

The word appropriate is in quotes because in many circumstances, the appropriate behavior is not known or is only determined after the fact. B t still, error is defined as deviance from the generally accepted correct or appropriate behavior.

Error is the general term for all wrong actions.

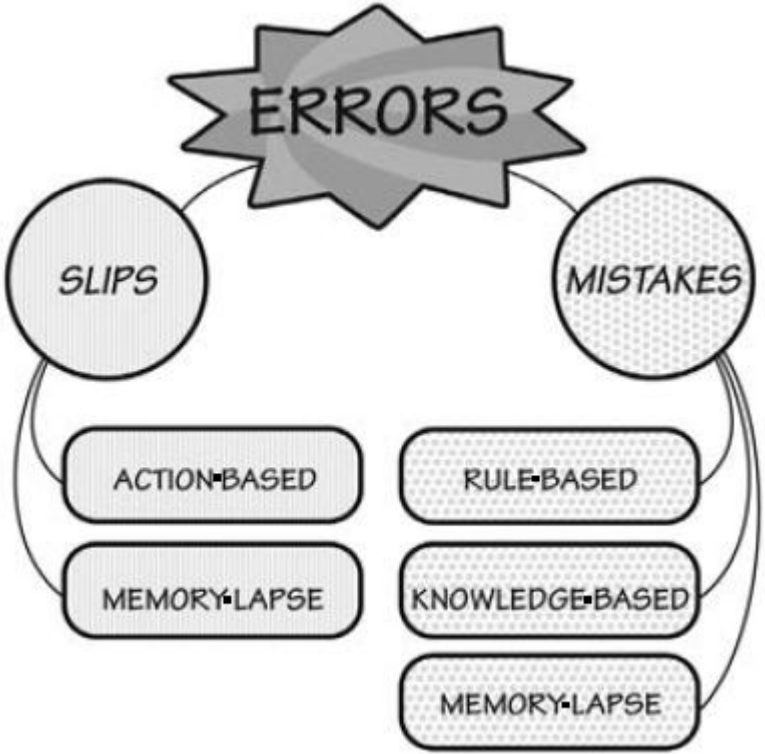
 (all rights reserved)

Slips and Mistakes

There are two major classes of error:

slips (lapsus)

mistakes (errori cognitivi)



 (all rights reserved)

SLIPS (lapsus)

A slip occurs when a person intends to do one action and ends up doing something else.

With a slip, the action performed is not the same as the action that was intended.

There are two major classes of slips: action-based and memory-lapse.

In action-based slips, the wrong action is performed.

In memory lapses, memory fails, so the intended action is not done or its results not evaluated.

 (all rights reserved)

Slips (lapsus)

Example of an action-based slip. I poured some milk into my coffee and then put the coffee cup into the refrigerator. This is the correct action applied to the wrong object.

Example of a memory-lapse slip. I forget to turn off the gas burner on my stove after cooking dinner.

 (all rights reserved)

Mistakes (cognitive errors)

A mistake occurs when the wrong goal is established or the wrong plan is formed. From that point on, even if the actions are executed properly they are part of the error, because the actions themselves are inappropriate; they are part of the wrong plan.

With a mistake, the action that is performed
matches the plan: **it is the plan that is wrong**

 (all rights reserved)

Mistakes (cognitive errors)

In a **rule-based mistake**, the person has appropriately diagnosed the situation, but then decided upon an erroneous course of action: the wrong rule is being followed.

In a **knowledge-based** mistake, the problem is misdiagnosed because of erroneous or incomplete knowledge.

Memory-lapse mistakes take place when there is forgetting at the stages of goals, plans, or evaluation. **(dimenticanza)**

DANIELE MAZZEI - IUM 22-23 (all rights reserved)

Mistakes (cognitive errors)

Example of **rule-based mistakes**. A mechanic diagnosed a defect on a car battery but decided to do not replace the battery because still working at 50% of performances

Example of **knowledge-based mistake**. Weight of fuel was computed in pounds instead of kilograms.

Example of **memory-lapse mistake**. A mechanic failed to complete troubleshooting because of forget a step.

 (all rights reserved)

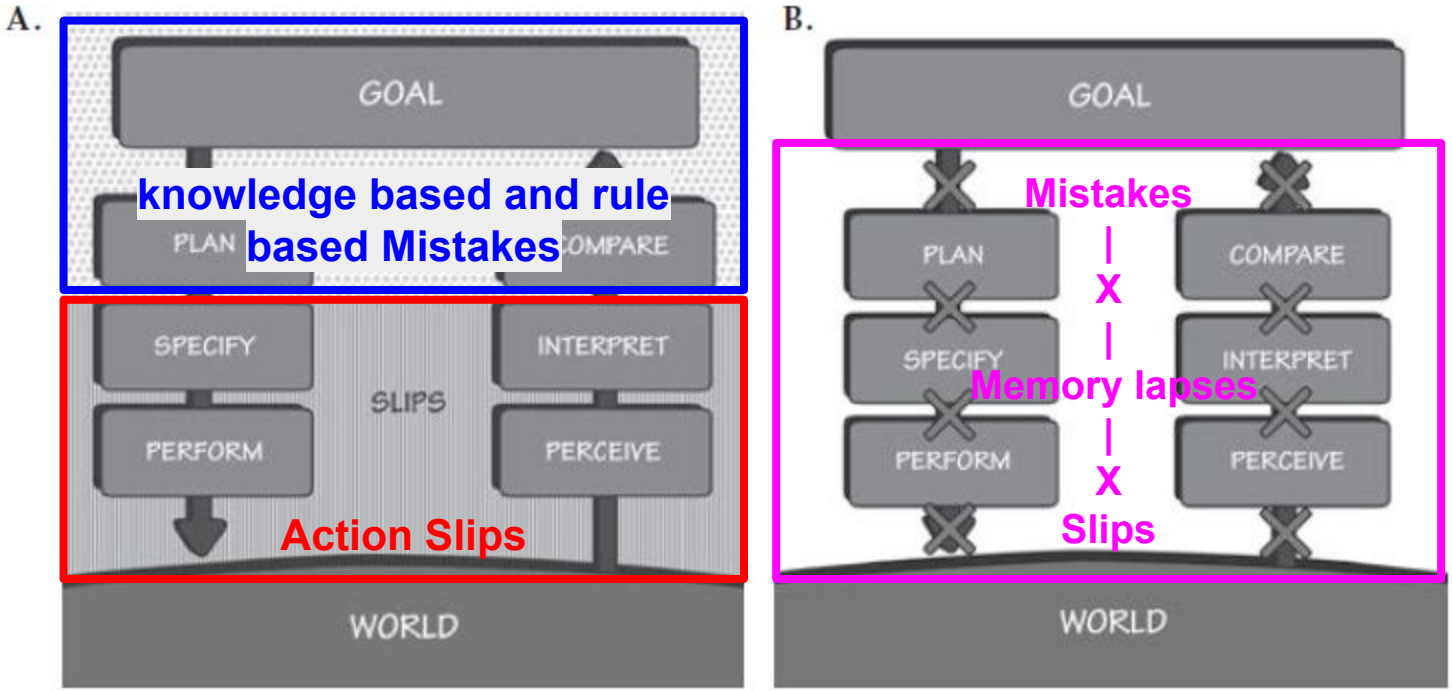


FIGURE 5.2. **Where Slips and Mistakes Originate in the Action Cycle.** Figure A shows that action slips come from the bottom four stages of the action cycle and mistakes from the top three stages. Memory lapses impact the transitions between stages (shown by the X's in Figure B). Memory lapses at the higher levels lead to mistakes, and lapses at the lower levels lead to slips.

 (all rights reserved)

Slips and Mistakes

- novices are more likely to make mistakes than slips
- experts are more likely to make slips and mistakes-memory lapses (dimenticanza).

Mistakes often arise from ambiguous or unclear information about the current state of a system, the lack of a good conceptual model, and inappropriate procedures.

most mistakes result from erroneous choice of goal or plan or erroneous evaluation and interpretation.

All of these come about through poor information provided by the system about the choice of goals and the means to accomplish them (plans), and poor-quality feedback about what has actually happened.

DANIELE MAZZEI - IUM 22-23 (all rights reserved)

Interruptions

A major source of error, especially memory-lapse errors, is interruption.

When an activity is interrupted by some other event, the cost of the interruption is far greater than the loss of the time required to deal with the interruption: it is also the cost of resuming the interrupted activity.

To resume, it is necessary to remember precisely the previous state of the activity: what the goal was, where one was in the action cycle, and the relevant state of the system.

Most systems make it difficult to resume after an interruption.

 (all rights reserved)

wrong feedbacks

Unnecessary, annoying alarms occur in numerous situations. How do people cope? By disconnecting warning signals, silencing bells etc

The problem comes after such alarms are disabled, either when people forget to restore the warning systems (there are those memory-lapse slips again), or if a different incident happens while the alarms are disconnected.

At that point, nobody notices.

Warnings and safety methods must be used with care and intelligence, taking into account the tradeoffs for the people who are affected.

The design of warning signals is surprisingly complex.

DANIELE MAZZEI - IUM 22-23 (all rights reserved)

Speech as feedback

More and more of our machines present information through speech. But like all approaches, this has both strengths and weaknesses.

It allows for precise information to be conveyed, especially when the person's visual attention is directed elsewhere.

But if several speech warnings operate at the same time, or if the environment is noisy, speech warnings may not be understood. Or if conversations among the users or operators are necessary, speech warnings will interfere.

Speech warning signals can be effective, but only if used intelligently.

 (all rights reserved)

Error prevention

It should not be possible for one simple error to cause widespread damage.

Here is what should be done:

- **Understand** the causes of error and design to minimize those causes.
- **Do sensibility checks.** Does the action pass the “common sense” test?
- **Make it possible to reverse actions**—to “undo” them—or make it harder to do what cannot be reversed.
- Make it **easier for people to discover the errors** that do occur, and make them easier to correct.
- **Don’t treat the action as an error**; rather, try to help the person complete the action properly. Think of the action as an approximation to what is desired.

DANIELE MAZZEI - IUM 22-23 (all rights reserved)

ADDING CONSTRAINTS TO BLOCK ERRORS

Prevention often involves adding specific constraints to actions. In the physical world, this can be done through clever use of shape and size (bocchettone benzina/diesel).

Electronic systems have a wide range of methods that could be used to reduce error. One is to **segregate controls**, so that easily confused controls are located far from one another.

Another is to use **separate modules**, so that any control not directly relevant to the current operation is not visible on the screen, but requires extra effort to get to.

DANIELE MAZZEI - IUM 22-23 (all rights reserved)

UNDO

Perhaps the most powerful tool to minimize the impact of errors is the Undo command in modern electronic systems, reversing the operations performed by the previous command, wherever possible.

The best systems have multiple levels of undoing, so it is possible to undo an entire sequence of actions.

 (all rights reserved)

CONFIRMATION AND ERROR MESSAGES

Many systems try to prevent errors by requiring confirmation before a command will be executed, especially when the action will destroy something of importance.

But these requests are usually ill-timed because after requesting an operation, people are usually certain they want it done.

A better check would be a prominent display of both the action to be taken and the object, perhaps with the choice of “cancel” or “do it.”

The important point is making salient what the implications of the action are.

Person: Delete “my most important file.”
System: Do you want to delete “my most important file”?
Person: Yes.
System: Are you certain?
Person: Yes!
System “My most favorite file” has been deleted.
Person: Oh. Damn.

Warning messages are surprisingly ineffective against mistakes

 (all rights reserved)

CONFIRMATION AND ERROR MESSAGES

What can a designer do?

• Make the item being acted upon more prominent. That is, change the appearance of the actual object being acted upon to be more visible: enlarge it, or perhaps change its color.

• Make the operation reversible. If the person saves the content, no harm is done except the annoyance of having to reopen the file. If the person elects Don’t Save, the system could secretly save the contents, and the next time the person opened the file, it could ask whether it should restore it to the latest condition.

 (all rights reserved)

SENSIBILITY CHECKS

Electronic systems have another advantage over mechanical ones: they can check to make sure that the requested operation is sensible.

Suppose I wanted to transfer $1,000 into a Korean bank account in won ($1,000 is roughly ₩1,000,000). But suppose I enter the Korean number into the dollar field.

Oops—I'm trying to transfer a million dollars. Intelligent systems would take note of the normal size of my transactions, querying if the amount was considerably larger than normal.

 (all rights reserved)

MINIMIZING SLIPS

Slips most frequently occur when the **conscious mind is distracted**, either by some other event or simply because the action being performed is so well learned that it can be done automatically, without conscious attention. As a result, the person **does not pay sufficient attention to the action** or its consequences.

It might therefore seem that one way to minimize slips is to ensure that people always pay close, conscious attention to the acts being done.

Bad idea!

Skilled behavior is subconscious, which means it is fast, effortless, and usually accurate.

DANIELE MAZZEI - IUM 22-23 (all rights reserved)

MINIMIZING SLIPS

Many slips can be minimized by **ensuring that the actions and their controls are as dissimilar as possible**, or at least, as physically far apart as possible.

The best way of mitigating slips is to provide perceptible feedback about the nature of the action being performed, then very perceptible feedback describing the new resulting state, coupled with a mechanism that allows the error to be undone.

For example, the use of machine-readable codes has led to a dramatic reduction in the delivery of wrong medications to patients.

(These scans do increase the workload, but only slightly. Other kinds of errors are still possible, but these simple steps have already been proven worthwhile.)

 (all rights reserved)

MINIMIZING SLIPS

Common engineering and design practices seem as if they are deliberately intended to cause slips.

Rows of identical controls or meters is a sure recipe for description-similarity errors.

Situations with numerous interruptions, yet where the design assumes undivided attention, are a clear enabler of memory lapses—and almost no equipment today is designed to support the numerous interruptions that so many situations entail.

Procedures should be designed so that the initial steps are as dissimilar as possible.

 (all rights reserved)

Error mitigation

The Swiss cheese metaphor suggests several ways to reduce accidents:

• Add more slices of cheese.

• Reduce the number of holes (or make the existing holes smaller).

• Alert the human operators when several holes have lined up.

Each of these has operational implications:

More slices of cheese means mores lines of defense **(more checks and control steps)**

Reducing the number of critical safety points where error can occur is like reducing the number or size of the holes **(Reduce distraction and design for error mitigation)**

DANIELE MAZZEI - IUM 22-23 (all rights reserved)

Design Principles for Dealing with Error

People are flexible, versatile, and creative. Machines are rigid, precise, and relatively fixed in their operations. There is a mismatch between the two,

Difficulties arise when we do not think of people and machines as collaborative systems, but assign whatever tasks can be automated to the machines and leave the rest to people. This ends up requiring people to behave in machine like fashion, in ways that differ from human capabilities.

We expect people to monitor machines, which means keeping alert for long periods, something we are bad at.

DANIELE MAZZEI - IUM 22-23 (all rights reserved)

Design Principles for Dealing with Error

What we call “human error” is often simply a human action that is inappropriate for the needs of technology. As a result, it flags a deficit in our technology. It should not be thought of as error.

We should eliminate the concept of error: instead, we should realize that people can use assistance in translating their goals and plans into the appropriate form for technology.

Therefore, the best designs take that fact as given and seek to minimize the opportunities for errors while also mitigating the consequences. Assume that every possible mishap will happen, so protect against them.

 (all rights reserved)

key design principles

• **Put the knowledge required to operate the technology in the world.** Don’t require that all the knowledge must be in the head. Allow for efficient operation when people have learned all the requirements, when they are experts who can perform without the knowledge in the world, but make it possible for non-experts to use the knowledge in the world. This will also help experts who need to perform a rare, infrequently performed operation or return to the technology after a prolonged absence.

• **Use the power of natural and artificial constraints:** physical, logical, semantic, and cultural. Exploit the power of forcing functions and natural mappings.

 (all rights reserved)

key design principles

- **Bridge the two gulfs,** the Gulf of Execution and the Gulf of Evaluation. Make things visible, both for execution and evaluation. On the execution side, provide feedforward information: make the options readily available. On the evaluation side, provide feedback: make the results of each action apparent. Make it possible to determine the system's status readily, easily, accurately, and in a form consistent with the person's goals, plans, and expectations.

DANIELE MAZZEI - IUM 22-23 (all rights reserved)

https://uxdesign.cc/how-to-prevent-your-users-from-making-mistakes-d641c6260b29

 (all rights reserved)