The Error Modes Checklist

Fourteen ways people predictably fail at an interface, and the design question that catches each one.

Every entry describes how a failure happens rather than who is at fault, and each one has a different fix. The checklist is read against a task broken into its individual steps. For each step, the question is which of the fourteen that step invites, and what in the design prevents it. Where the answer is not known, that is something to test rather than a box to tick. This is a prompt list for a design review, and it does not replace a full error analysis carried out against a task breakdown.

DOING — the plan was right and the action went wrong

01 The familiar action takes over CAPTURE ERROR

A well-worn sequence runs instead of the one intended: pressing Save when Save As was meant. Nothing feels wrong at the time.

> Do a routine path and a serious one start the same way, sit in the same place, or feel the same to perform?

02 Right action, wrong thing DESCRIPTION ERROR

The action is correct and lands on the neighbour: the row above, the second account in the list, last month's file.

> At the moment of choosing, is there anything beyond the label to tell the two apart?

03 Right action, wrong state MODE ERROR

The action would have been correct had the system been in the state the person believed it was in. Caps lock is the harmless version; editing the live record rather than the draft is not.

> Does the screen show the state the system is in, rather than the last instruction given to it?

04 The value is the wrong size TRANSPOSITION

The right kind of number entered wrongly: a decimal point one place out, two digits swapped, pounds where kilograms were meant. It looks ordinary, so it is accepted.

> Is a value that cannot be right refused as it is typed, rather than checked afterwards?

05 The step lost to an interruption OMISSION

Something interrupts, and the step that was coming next is the one that disappears. The longer the diversion, the more likely the loss.

> Does the screen show where the work was left, or does getting back rely on memory?

06 The step after the point is reached POST-COMPLETION ERROR

The final step is dropped because the purpose has already been met: the cash is collected and the card stays in the machine.

> Does a required step fall after the moment the person has got what they came for?

CHECKING — the information was there and went unused

07 The check that was never made PREMATURE CLOSURE

The check is cleared without being performed, because the answer is assumed in advance. This is why a confirmation screen catches less than teams expect: it is read as a step to get past.

> Does the check require something that cannot be done without reading what is being checked?

08 Checking the wrong thing carefully WRONG REFERENT

The check is made properly, against the previous version, a similar record or the wrong row. Diligence offers no protection here.

> At the point of checking, does the item on screen show what makes it different from its neighbours?

09 Acting on an out-of-date screen STALE STATE

The display is behind the system, and the person acts on what it still shows. Nothing indicates that the two have parted company.

> Is the age of what is on screen evident, and is action held back while it is uncertain?

THINKING — the plan itself was wrong

10 Carrying something in the head RETRIEVAL ERROR

A code, a figure or a place in a list has to be held from one screen to the next, and it is misremembered or lost altogether.

> Is what is needed to decide shown where the decision is made, rather than one screen earlier?

11 The rule that nearly always works STRONG-BUT-WRONG RULE

A rule that has never failed is applied to the one case where it does not hold. The more reliable it has been, the faster it is reached for and the less it is questioned.

> Is the exception visible in the interface itself, or only in the guidance and the training?

12 A clear idea of the system, and wrong MENTAL MODEL

Everything that follows is sensible given that idea, and unclear feedback is read as confirming it, so the mistake survives contact with the screen.

> Does the system show how it works and what it is about to do, rather than only what it has done?

TRUSTING — the work passes to something or someone else

13 Taking the system's word for it AUTOMATION BIAS

The output is accepted because checking costs more than agreeing. What the system fails to raise is missed as often as what it gets wrong.

> Is the evidence needed to overrule the system available at the moment of the decision?

14 The shortcut that becomes the method ROUTINE VIOLATION

The official route is slower than one that works, so the shortcut becomes normal practice. Nobody decides to lower the standard, and it moves anyway.

> Is the correct way also the easiest way when people are busy?

WHAT TO DO WITH A HIT

A likely error mode is a design input rather than a bug report. The order of remedy runs: make the error impossible; failing that, let the system absorb it; failing that, make it reversible. A warning message sits below all three, and training below that.

An error that cannot be designed out becomes something to test, with the pass mark agreed before any session is run. One that is neither designed out nor tested is a known risk, and belongs in the record as one.

Fourteen entries will not cover every product. The formal methods behind this list — SHERPA applied to a task breakdown, among others — work through a task step by step and produce the full set.

Outside this list. Three families need separate treatment: mistakes in handing work between people, the conditions the work is done in, and shortcuts adopted by a whole team rather than one person.

Sources. Reason, J. (1990). *Human Error*. Cambridge University Press. · Norman, D. A. (1981). Categorization of action slips. *Psychological Review*, 88(1), 1–15. · Rasmussen, J. (1983). Skills, rules and knowledge. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(3), 257–266. · Embrey, D. E. (1986). SHERPA: a systematic human error reduction and prediction approach. · Byrne, M. D., & Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive Science*, 21(1), 31–61. · Altmann, E. M., & Trafton, J. G. (2002). Memory for goals. *Cognitive Science*, 26(1), 39–83. · Parasuraman, R., & Manzey, D. H. (2010). Complacency and bias in human use of automation. *Human Factors*, 52(3), 381–410. · ISO 9241-110:2020, clause 5.6. All sources are paraphrased; nothing above is quoted. The grouping into four families is an editorial arrangement rather than a taxonomy reported in any single source.