

From Whence Come Failures

Steve Tengler, User Experience Director

I was recently on a project where the dogmatic Program Manager – let’s call him Seth – required a Failure Modes and Effects Analysis (FMEA) for each change in the design, regardless of the type of change being requested and the potential impact therein. There was a part of me that thought this inflexible requirement was overkill, as did most of the Human Factors staff (e.g. “We’re correcting the wording of one prompt! What doesn’t Seth get?!”). But within the dank recesses of my mind (where forgotten requests to take out the trash bounce around the echo chamber without impediment), I disagreed with my Human Factors brethren and believed that FMEA’s actually do have a place in the User Experience world.

Before we dig into this, though, let’s make sure everyone understands what an FMEA is. According to Wikipedia (the website that has antiquated the moldy encyclopedias harvesting spores at my Dad’s house), FMEAs are “... a procedure in production development and operations management for analysis of potential failure modes within a system for classification by the severity and likelihood of the failures.” ([See more here](#) [1]) Yeah, yeah ... reads like a tombstone. Perhaps a layman’s way of saying that would be “Imagine Homer Simpson using a doohickie, and something causes him to say or scream, ‘D’oh!’ What caused that exclamation, how many times did Homer cry out, and how intense was the displeasure?”

Consider our household fridge, which has a little annoying failure: if the refrigerator portion is shut with sufficient force, the freezer pops open slightly, thereby letting everything thaw while the motor unsuccessfully attempts to overcome the injection of warm air. A definite “D’oh!” What’s the frequency of this? I have two 7-year old boys, so on a scale of 1-10 with 1 being “near never” and 10 being “constantly”, I’d give it an 8. Severity? Well, nobody dies or gets injured and we usually (unfortunately) notice before the frozen vegetables need to be pitched. That said, I’d give it a “6” since the refrigerator’s demise must be looming. The last variable in this scenario is detection (i.e. “How unlikely are you to notice the problem before it ships to market?”), which I’d give a “9” since it’s pretty easy to walk away or pass without noticing the inch-wide gap. Now, as the Chief Fridge Operator (CFO) of my household, I see this problem as something that needs to be addressed since the three variables multiplied together ($8 \times 6 \times 9$) yield a much higher score (432) than other issues -- like projectile food assaulting me as I whip open the door ($7 \times 2 \times 10 = 140$). So herein lies the power of an FMEA: it helps me prioritize where to address issues with design resources, additional piece cost, and testing procedures. As you can see by the picture (below), the two adult engineers of the household shopped around for a solution (finite resources!), bought a sticky-backed strap (cost) and confirmed that it worked (testing procedure). Now on to the next problem on the list: how do we quell the flying food?!

For the sake of our discussion about the FMEA tool, though, onward and forth we go to the most complex part: how to apply it towards the Human-Machine Interface (HMI). I would argue that there are three main categories of HMI failures: the hardware, the software and the fleshware.

Hardware Failures

This is the meat of most FMEAs: the mechanical failures. “This part can fail by doing <x> to it in <y> fashion under <z> conditions.” And so the engineer will bolster the strength or bracing of a given bracket, housing, sealant, etc. within the thoughts of normal usage.

The part that makes this interesting for HMI is the various failure modes that become relevant depending upon the selection of interfaces. For instance, Ian Crosby wrote a great article in Appliance Design regarding failure modes that must be considered when adding touchscreens to the user interfaces within the kitchen (e.g. fridge, stove, [see the full article here](#) [2]). When the Human Factors engineer is considering a shift from knobs to touchscreen, [s]he’s probably examining the task completion, design flexibility, language flexibility, etc. and hasn’t stopped to consider the new failure modes that must be considered for THAT environment and THAT user interface.

Here’s another example: one automotive company has two interesting tests required for any HMI that will be

From Whence Come Failures

Published on Industrial Maintenance & Plant Operation (<http://www.impomag.com>)

packaged in the trunk (*yes, there are even interfaces in the trunk!). The first test is affectionately known as the Golf Bag Test, which literally requires the engineer to take a fully-loaded golf bag – which in my case is even heavier considering I carry a couple of extra, illegal clubs in it – and toss it directly at the component. No, this isn't a perfectly repeatable test with measured forces, but it illustrates consideration of the customer's usage pattern. Another such test is commonly used within multiple industries: the Drop Test. Usually this involves taking the newly manufactured component and dropping it from six feet high to a solid surface. Why do this? Because dropping it WILL happen at the assembly plant by accident (remember: humans err!) and that item might still hit the streets since damage isn't always visible. Therein, it behooves manufacturers to make sure the part can survive assembly as to reduce service costs and customer dissatisfaction.

Another common hardware failure usually considered within Design FMEAs is an ergonomic snafu. "What if the shorter female customer cannot reach this or have the strength to lift it from this angle?" "What if the button is too small for a large male with a gloved hand?" Anthropometric tables abound for nearly all demographics ([example](#) [3]) and the simple hardware failure of can't fit, can't reach, can't turn, can't lift, or can't pull might make the difference between a customer being completely pleased with your product or returning it for a full refund.

Software Failures

People reading this title are probably thinking about the horrific, sweat-down-your-back software failures: blue screen of death, display lock-ups, or other such catastrophic endings ("Son of a ... I just lost three hours of work!").

Yeah, those stink ... but that's not my focus.

I'm talking about where the software design has failed to provide all of the information required to satisfy the use case. There are a multitude of possible scenarios, but here are a few examples:

- **DATABASE OR CLIENT OUTDATED:** Imagine you bought a new luxury SUV with all the bells and whistles, which included an 8-inch touchscreen, embedded navigation and Advisor-assisted telematics. Serious cabbage and coin! After speaking with an attractive-sounding Advisor (they always sound attractive, don't they?!), you think directions have been downloaded to friend's newly built house. Here's the problem: his entire subdivision doesn't exist in your nav system since the house construction happened AFTER your car was built and the streets didn't even exist then. In fact, your car literally thinks the requested point is in the middle of a massive field without an entrance for miles. How does your HMI respond?
- **UNFORESEEN INPUT/OUTPUT:** Who amongst us hasn't seen a display or entry line that was too short for the text (and either gets truncated, or poorly wrapped to the next line?) Or how about seeing a ## or XX when the input exceeds what the programmer perceived as the maximum? I've actually gotten on a scale that registered *** after measuring me and, I must say, that's tremendously humbling.
- **UNFORSEEN OR POORLY EXPLAINED ERROR:** "Your TechPro200x.3 has failed via data processing error 1052XB1A. Press "OK" to continue." Yes, understanding how the software might fail, comprehending possibilities for the customer to correct the issue or seek assistance, and making all of that clear to even the densest customer is a Herculean task at best. Nevertheless, this can be the most frustrating user experience of any, and dominates discussions about how to create a winning user interface ([example](#) [4]). A classic example of underestimating the effects of not offering clear explanations to system's errors was the Therac-25 case, where several people died from radiation poisoning because, amongst other reasons, the fault messages simply said "MALFUNCTION" followed by a number from 1 to 64 without explanation within the display or manual ([View here](#) [5]).

The key – just like any of the failures, the software engineer -- in conjunction with the human factors engineer --

From Whence Come Failures

Published on Industrial Maintenance & Plant Operation (<http://www.impomag.com>)

must stop and say, "How might my software get sideways?"

Fleshware Failures

The fleshware failures are, as far as I've seen, the most frequently forgotten or missed in any given FMEA. They are the situations that arise when the human or user makes a mistake. What, you ask? Humans error?? Yes, I'm sure that must be a surprise to some of you (typically the megalomaniacs of the crowd), but to err is human.

I flub up all the time...seriously. I'm a walking error waiting to happen and, what makes me feel like a bigger buffoon is my wife is not only nearly-error-proof but also coincidentally present whenever I do err. Here's a great example: this morning I started the coffee maker without putting the coffee grounds in the filter. Pretty stupid, eh? And I've done it before! My legitimate excuse is I'm at the trough of caffeine-loading; all six cylinders are misfiring. But since I'm not the only addicted zombie of the working world, I'm betting I'm not alone in having done this (please comment below if you have, if for no other reason than to make me feel better...PLEASE!!) So the FMEA line for this flub-up might read "2" for frequency, "7" for severity (since I subsequently ran out of time to fulfill my morning requirement and suffered in efficiency all day), and a "1" for detection (i.e. any moron should notice clear water pouring forth). That's a pretty low overall score ($2 \times 7 \times 1 = 14$), which probably explains why Mr. Coffee never added the cost of a coffee-grounds-detection system. Nevertheless, I'd be willing to bet a Starbucks Venti that their FMEA doesn't include that specific error.

Why ignore the human? Wikipedia didn't. In fact, they even included us in the FMEA definition: "*Failure modes* are any errors or defects in a process, design, or item, **especially those that affect the customer**, and can be potential or actual." We tend to forget that the user is integral to the system. To combat this tendency, some engineering groups refer to themselves as the Human-Systems Integration (HSI) group rather than Human-Machine Interface (HMI) for exactly that reason - to specifically remind themselves that the human is nearly-always a crucial element of the system.

Still though, why forget the human? I think there are two reasons. First and foremost, the component is tangible and present; the user is amorphous and undefined. It's easy to forget there's a vast world out there when you're deep in the weeds, examining the circuit board layout or sheer forces exerted on the bracing fasteners. The failures still exist, but the situation or person is long gone. The part returns from the field with little or no explanation of why the fastener failed, so that engineer sees a torqued part rather than the full picture.

But another reason we tend not to include these errors is what I call the Safari Factor. Maybe you have been on a safari, but I have never traveled to the far corners of the Earth since it costs a LOT OF MONEY and it's not an easy trip. We go to the zoo. It's thirty minutes away, it's \$50 for the whole family (including snack shop food, easy-access parking, gas for the trip, and entry fees), and the map shows us where each neatly-bundled quasi-adventure is located. BUT if I truly wanted to know how animals interact with flora and fauna, I must go to their natural habitat and watch them in their 'hood. In User Experience design, it is no different. Ride-alongs cost a lot of money, they aren't conveniently located and they require a lot more time -- but they are chock full of insightful adventures and interactions that you'd never find in the sterile test lab.

A great example is driver inattention, which -- in my humble opinion -- is a better term than driver distraction since, let's face it, the blame should reside with us individually. Anyway, the government realized a few years back that the only way to assess the root cause of accidents was to instrument a bunch of vehicles and watch millions of miles of naturalistic driving to see the causes of crashes, near crashes and minor incidents. And what did they find? "Eyes off the road" was a major contributor, so they have started to address some of the high FMEA items ([see the fact sheet here](#) [6]). How much did this cost? Tens of millions of dollars. How long did it take for study design through full analysis? Several years. Was it better than theorizing from the annals of a cube farm? Immeasurably so!

Back to the Therac-25 example, no one within the designer's white tower could have imagined the odd combination of keystrokes that would lead to the ultimate failure. Nor would they have imagined that operators would ignore repeated failures because they did not believe the complaints of their patients. Yes, there were plenty of software and hardware failures of that system, but the operators absolutely contributed.

So if your FMEA is light on fleshware content, consider doing a few ride-alongs with your customers. Hang out in their kitchens and see if they slam the fridge or run an empty coffee maker. Or just follow me around for a few

From Whence Come Failures

Published on Industrial Maintenance & Plant Operation (<http://www.impomag.com>)

days and take copious notes.

Source URL (retrieved on 04/26/2015 - 1:52am):

<http://www.impomag.com/articles/2012/02/whence-come-failures>

Links:

[1] http://en.wikipedia.org/wiki/Failure_mode_and_effects_analysis

[2] http://www.appliancedesign.com/Articles/Feature_Article/BNP_GUID_9-5-2006_A_1000000000001001975

[3] <http://www.ergoworkinggroup.org/ewgweb/.../AnthropometricsTables.doc>

[4] http://www.useit.com/papers/heuristic/heuristic_list.html

[5] <http://en.wikipedia.org/wiki/Therac-25>

[6] http://www.vtti.vt.edu/PDF/100-Car_Fact-Sheet.pdf