Responsible Citizenship in a Technological Democracy

Note 7: Safety, Reliability, and Testing

Safety and Reliability

Safety and reliability are two special concerns of public policy. Indeed one of the main reasons we have governments *at all* is to ensure our collective safety and security, at least in so far as it's possible. It's why we have things as diverse as speed limits, food labeling, police, and clinical trials of new disease therapies. So, while safety and reliability might be thought of as special cases of risks, as discussed in the previous note, they deserve a little special attention. They also let me introduce some material on the notion of testing.

Perhaps this is too pedantic, but by "safe" we mean that no harm will be done. By "reliable" we mean that something works in a completely predictable way – we can *rely* on it to do so. That seems simple enough. However, maybe by the time you finish reading this note you won't be so sure.

Let's start with the principle way that we determine safety and reliability – testing. The first point to make is that there is no general method of testing that applies to all situations. An appropriate method has to be created for each situation – and that's the first problem. It's not easy to design good tests. What I plan to do here is just give you a sample of two kinds of testing – one for physical, engineered products, and one for drug approval. Between these two I think you will get exposed to most of the issues in testing.

Testing physical, engineered products.

To be concrete, let's assume we're testing something really simple. This is going to be a bit contrived to keep it simple, but suppose you are testing a shovel. Yea, a shovel. Here's the simplified version of the sort of procedure engineers would go through:

> **Step 1: Determine the "failure modes" of interest**. In defining "safe" above we said "no harm will be done". That's fine as a general statement, but what does it mean for a shovel? What specifically could cause "harm"? Suppose that we determine that a failure mode for a shovel is that if you stomp on it too hard in very hard soil it could crumple up and actually trap your foot, possibly even gouge it. OK, crumpling is a failure mode – there may be others but let's stick with this one for the example.
>
> **Step 2: Design a "test procedure" to determine where the failure modes occur.** There is no question that if you apply enough force to the shovel, it will crumple. The question is how much force is required to make that happen. Oh, and perhaps we'd better also consider how long is that force applied. Maybe a given force applied for a second won't crumple the shovel, but the same force applied for an hour would. The details don't matter, but I'm sure you can imagine a device that automatically "steps" on the shovel with various amounts of force and various lengths of time to determine where it fails.

> **Step 3: Decide whether the failure points can happen in the "expected use" of what we're testing.** Even *really* heavy people weigh only a few hundred pounds, so there is a limit to how much force they can apply to the shovel by standing on it. It also seems unlikely that anyone would stand on a shovel for an hour, so there must be some limit on the time a real shovel in real use would be subjected to a given force. If the failure points for our shovel require a ten thousand pound person standing on it for twenty four hours, it is probably safe.

These is a whole field of "safety engineering" with elaborate standards and procedures. These three steps are a simplification of what safety engineers would do, but they capture the flavor of the testing of many engineered systems. There are problems with each of the steps, unfortunately.

It's not so easy to a priori determine all the failure modes, for example. What if, when muddy, it's easy for the user's foot to slip off the shovel and cause an injury? What if, when trying to pry up an unusually heavy load, the handle shatters and causes an injury? What if the design of the shovel places a unique strain on the user's body that, in turn, causes a repetitive motion injury? The notion of "harm" is a lot more elusive than one might think, and what is seen as "harm" with 20/20 hindsight may well not have been envisioned initially, even by the best intentioned, rigorous search.

Let me come back to the problems with step 2 later. In step 3, the problems are similar to those in step 1 – envisioning the "expected use" completely is hard. For example, what about the jerk who, frustrated with his/her inability to dig, jumps off a step-ladder onto the shovel? That translates into a lot more force than just standing on it. What about the sociopath that sharpens the blade of the shovel and uses it as a weapon? The point is that it's a lot harder to prospectively define either "harm" or "expected use" than you might have thought!

And, if it's hard for something as simple as a shovel, just think about how much harder it is for more complex systems – an airliner, the space shuttle, or Windows. Although some people had thought about it, most had not considered the use of an airliner as a missile to attack tall building prior to 9/11, for example.

We just have to live with the fact that, as a practical matter, it's impossible to predict all the bad things that can happen. That doesn't mean we shouldn't try. We should certainly expect those responsible to try *very* hard to predict failure modes and ensure they fall outside any reasonable expected use! But we shouldn't be too surprised either if really creative failures occur.

> **Concept:** In practice it is impossible to predict all possible failure modes and all possible (mis)uses of a technology.

Some people, especially in Europe, are suggesting that we should adopt the "precautionary principle" with respect to new technologies. The precautionary principle simply says that no new technology should be adopted unless it can be proven to have *no* negative effects. Alas, it's simply not possible because, at least as a practical matter, it's impossible what all the negative effects might be.

Now let's go back and focus on step 2 – determining a test procedure. Think back to the procedure described for testing the shovel and recall that there are two quantities – the force applied and the amount of time the force is applied. That's a problem!

If there had been just one thing to consider, say the force, we could have simply designed the test procedure to slowly increase the force until the shovel crumpled. But because there are two we need to hold one constant and vary the other then change the first and do it again. The problem is that the number of tests you need to perform becomes infinite if you want to do this for *all* values of force and time.

Fortunately, you don't have to test the shovel for *all* values of force and time. To explain why I need to make a brief detour to teach you a bit of mathematics. I know, I know, I promised no equations and no mathematics prerequisite – but I *didn't* promise not to *teach* you a bit of mathematics! ☺

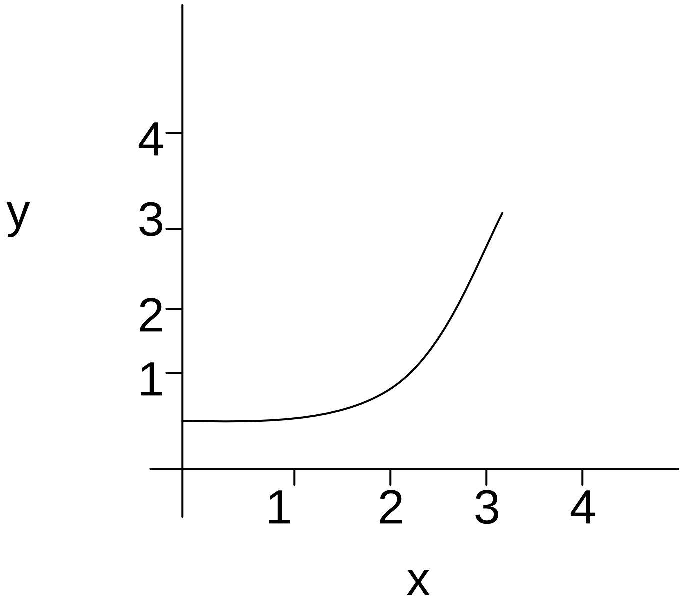
We geeks are fond of equations. Maybe the most ubiquitous of them is

$$y = f(x)$$

It merely says that, if you give me a specific number for x, then I can use the formula (or recipe if you prefer), f, to compute another number, y. For example,

$$y = x^2$$

says that if x is 2, then y is 2^2, or 2×2, or 4 – or if the value of x is 12 then y is 12^2 or 144. Big deal. We're also fond of graphing such functions, as is shown below. It's just a visual way of showing the value of y for various values of x.



Now, here’s the mathematical concept you ought to know in order to be a responsible citizen:

> **Concept**: Functions come in two flavors: *continuous* and *discontinuous* (or *discrete*).
>
> Continuous functions have the property that a small change in *x* produces a small change in *y*. You can draw the graph of such functions without lifting your pencil from the paper.
>
> Discontinuous functions do not have this property. Small changes in *x* may produce large changes in *y*. You must lift your pencil from the paper to draw the graph of a discontinuous function.

As an example, consider a function whose value is zero for all values of x except x=5, and for x=5, the value of the function is one. This function never has the value ½ or ¾ or anything else except zero or one. Moreover, its value is one only when *x* is *precisely* 5 – just a tiny bit bigger or smaller and the function’s value is zero. Now do you see why you need to lift your pencil from the paper to graph it?

OK, now back to testing. Fortunately, all the equations that describe nature are continuous. That means that if we test our shovel for two values of force that are “pretty close” to each other, or two values of time that are likewise “pretty close”, we can expect that the behavior of the shovel at all intermediate values of force and time will also be “pretty close” to what we observe at the points we tested. And that in turn means that we only have to perform a finite number of tests. Problem solved.

How close is “pretty close”? Well, it depends. For our purposes it’s enough that you understand the general approach. You don’t need to actually perform tests to ponder safety and reliability policy.

Unfortunately there are some very important discontinuous functions in our modern world – computers, and "information technology" more generally. I'm sure you've heard that everything in a computer is represented as 0's and 1's. That's right. There are no intermediate values – no ½'s or ¾'s. The values of *x* and *y*, and every intermediate value in creating *y* from *x* are just patterns of zeros and ones. And that means that, at least in principle, a seemingly small change in *x* can produce an enormous change in *y*. And that, in turn, means that you can't just test values that are "pretty close" and expect that the behavior in between those points will be "pretty close" to that at the points you tested – in fact it might be wildly different And that, in turn, means that you have to test for *all* values of the input. And that, finally, means that the number of tests required is so enormous that, in practice, it can't be done.

Maybe a feel for the magnitude of the numbers would help clarify the statement that "… in practice, it can't be done." First a reminder about mathematical notation. 10^2 means 10×10. 10^3 means 10×10×10, and so on. This is just a convenient way to write very large numbers in a compact way. For example, the number of atoms in the entire universe is about a google, or 10^{100} (yes, that is where the company name came from).

As we said, computers store information in digital form – essentially a pattern of 0's and 1's. To completely test it we'd have to perform the test for every such pattern. On my laptop there are $10^{10,000,000,000}$ such patterns. That's not a typo – the superscript is a one followed by 10 zeros! If every atom in the universe could perform 10^{100} tests per second there would not be enough time since the "big bang" to complete all the tests on my lonely little laptop.

So, a complete test can't be done. Software companies, however, *do* test their products – in fact they spend a lot of money doing it. The way they do it is a slight modification of the way we test continuous systems. They sample possible input values that seem "pretty close" and *hope* that the product's behavior for intermediate input values behaves like a continuous function. It's been estimated that the Office suite (Word, Excel, etc.) contains about a half a million "bugs" at any given time – so it's clear that the hope is not well founded.

Our inability to completely test computers and other digital systems is becoming more and more important as computers are "embedded" in other kinds of products. In fact, they are just about embedded in every kind of product – I recently learned, for example, that my electric razor has a micro-computer in it to manage the battery use. Any such product then also becomes a hybrid described by some continuous and some discontinuous functions, and the later part of it can't be completely tested. Since the computer in these systems is often used to control the physical components, the behavior of the physical components can appear discontinuous as well.

Whether my electric razor computer is completely tested or not probably doesn't matter much. But what about the one running the autopilot on the plane you're riding in, or controls the firing of nuclear-tipped missiles?

> **Concept:** Because the mathematics that describes computers and other digital systems is discrete (discontinuous), the only way to test them is, in practice, impossible because the number of tests is so enormous.

Testing of new drugs.

The Food and Drug Administration is charged with approving new drugs – approval is given only if the drug is both safe and effective. Ultimately that's going to have to involve giving the drug to sick people, but that's potentially risky, so the FDA process involves a series of progressive steps that try to proceed ethically to reduce that risk. At each step FDA and a local "Institutional Review Board", or IRB, review the information available to decide whether to proceed to the next step.

When the FDA gets an application for a new drug approval, its first concern is safety. So before it's tested on people it's first tested on animals to see if there are any obvious negative side effects. If there are none, it may proceed to "clinical trials".

Typically clinical trials are performed in three stages. Failure at any stage stops the process.

> **Stage 1:** At this stage the FDA is still principally concerned with safety. Animal tests don't prove that the drug is safe since no animal reacts just like a human, so the drug is given to a relatively small number of healthy people to see if they have unacceptable negative reactions. The number is small so that if there is a reaction, only a few people will suffer. They are healthy because, again, if there is a reaction, they presumably can tolerate it better than sick people.
>
> **Stage 2:** The emphasis at this stage is principally on effectiveness, but there is still a concern about safety. A somewhat larger group of sick people are randomly divided into two groups – but they are not told which group they belong to. One group is given the new drug and the other group is given either (1) a placebo (e.g., a sugar pill), or (2) a current drug for the same illness. Again the number of people involved is modest to limit the damage if something goes wrong. The primary output, however, is to see whether the group receiving the new drug does better than the group getting the placebo – or at least as well as the group getting the currently available drug.
>
> **Stage 3:** We get to this point only if there is no evidence of unacceptable negative effects, and at least some evidence that the new drug is effective. But, because the number of people involved in stages 1 and 2 is small, one can't be confident that the drug will be safe and effective for everyone. So, stage 3 is structured like stage 2, but involves many more subjects – typically a thousand or more.

By the way, the process described above for stages 2 and 3, is called "randomized, double blind" (RDB) testing. It's "randomized" because the two groups are chosen at random. It's "double blind" because neither the subjects nor the people that will evaluate the results know who was assigned to which group.

An Aside on Safety, Reliability and Failure Analysis

Just as with risks, there are new methods for analyzing physical engineered systems for safety and reliability. However, the majority of techniques we use to make products safe date from a time before this kind of analysis existed, so the techniques we use resulted from analysis of things that proved themselves unsafe or unreliable. Steam boilers used to blow up all the time and kill people, for example. Slowly, over a long period of time, each failure was analyzed and a fix found. As a result, today steam boilers don't blow up.

It's still the case, in fact, that failure analysis is an important tool. After every airplane crash, for example, the Transportation Safety Board painstakingly reassembles the parts of the plane and scrutinizes the flight recorder and cockpit voice recorder to be sure they understand *exactly* what happened and how it can be prevented in the future. Despite the fact that the new techniques of safety analysis are great, the reality is that failure analysis is still a critical tool.

An aside on the aside: medical "ex-plants". A medical ex-plant is something that was implanted into a patient and then taken back out because it failed. Artificial hips and knees, for example, eventually wear out and need to be replaced. Similarly, cardiac pacemakers eventually fail and need to be replaced.

A policy issue that seems just plain nuts is that such explants are never subjected to failure analysis. Principally from concern over malpractice suits, hospitals never return explants to their manufacturer for such analysis – so we the public are deprived of one of the best ways in which implants could be improved!

Policy Relevance

As noted earlier, ensuring safety is one of the principal functions of government. A common tool for achieving this is regulation. Every elevator you ride in will have either a current certificate of inspection or a notice of where it can be seen – that's because of a regulation that elevators must be inspected yearly. As we just saw, new drugs must be approved by the FDA – that's because the FDA has "regulatory authority" over drugs. The bottle of milk you buy will have a notice saying "pasteurized" because there is a regulation requiring it.

Regulations tend to accumulate. It's easy to add one when a new problem is uncovered, but virtually impossible to remove one that has been overtaken by events. The yearly certification of elevators, for example, arose over a hundred years ago when they were far less safe and reliable than they are today. Have you ever seen an elevator that *wasn't* certified to be safe? We could probably relax the certification regulation to a less frequent inspection, but that just doesn't happen. It always seems "safer" to keep the old regulations. That has some problems that raise interesting policy issues – see, for example, the note on "personalized medicine" below..

New technologies also occasionally require new kinds of regulation. One of the questions in Note 3, for example, asked whether Congress should consider requiring "complete testing" of computer based, electronic voting systems. Well, now you know that is not possible. Given that a switch to electronic voting seems both desirable and inevitable, what are the regulatory options? Having confidence in the election process is essential to a democracy – how do we gain that confidence if complete testing of the voting machines isn't possible?

There is a similar concern with randomized, double blind testing of drugs. The trend, post human DNA decoding, is toward “personalized medicine” – medicine or medical procedures that will work *great* for me, but not for you! How do we test the safety and reliability of such medicines? RDB doesn’t work because, by definition, personalized medicine works for only one person. There’s no way to create a thousand randomly chosen people to test such a medicine. Again, what are the regulatory options? Either we have to replace the RDB regulations, which we noted above seems hard to do, or we forgo personalized medicine, which seems dumb.