The White-Shirt Effect: Learning from Failed Expectations

Lutz Prechelt

(a chapter from: Menzies, Williams, Zimmermann: *Perspectives on Data Science for Software Engineering*, Morgan Kaufman, pp.353-357, 2016)

A Story

Revelation

Shortly after waking up unusually peacefully on this mild, friendly, sunny, simply wonderful August morning, you have a revelation: Programmers will make *much* fewer mistakes if they are wearing a white shirt!

You are a software manager and oversee the work of more than 700 software engineers, so you straighten up, excited. This is it! This is what will turn your good software development organization into a top-class software organization! Higher productivity! Improved quality! Happier engineers *and* customers! (Remark: I should point out that this is a deliberately silly example to make it easier to focus on the principles in the story rather than getting distracted by certain details. But back to our story:)

Yet, you also know that sound management practices will require you to prove your insight at least correlationally. You get to work immediately and will end up having invested half of your capacity for the next five weeks to prepare your coup.

Work, work, work

You convince half of the key people in your organization of the White Shirt Effect (WSE) and convince most of the other half that the WSE is at least worth checking out, so your organization backs you up in the subsequent steps.

Having followed the literature on the latest techniques for defect-correction-to-defect-insertion mapping (DCDIM) you know how to find out with good reliability (once a defect has been corrected) who has produced a defect. The data needed to do this is available in your bug trackers and version archives. Easy.

You decide you will find the daily white-shirt bit for each software engineer via photos. Much tougher. You enlist one of your best toolmakers. A few days later, a prototype software works that can determine who is wearing a white shirt and who is not, based on reusable face recognition technology and a custom white-shirt-bit plugin.

Over several days, you carefully calm down the privacy concerns by demonstrating that the single-purpose, in-house software solution will indeed only capture the white-shirt bit and nothing else. Your organization now allows you to install the 25 cameras required to capture nearly everybody often enough.

You have an assistant fill the face recognition database with clean portraits of each of your software engineers.

You let the solution run for a week meanwhile two helpers capture manually as much of the white-shirt bits as they can to validate the reliability of the camera-based solution. The software proves to work very well.

Knowing that the more interesting defects will take some time to surface, you announce the beginning of a six-month trial period: You explain the WSE to all of your software engineers and tell them to wear a white shirt at least 2 days and at most 4 days every week and to, please, randomize the weekdays.

You find that, in your initial enthusiasm, you had overlooked what comes now: A majority of your software engineers opposes the shirt rule (some nearly revolt against it), because they are not used to be told what to wear and many of them do not like shirts ("With a collar? Are you serious??"), let alone white ones.

It takes a stressful and dramatic plenary meeting, in which you remind them of the data-driven management principles of your organization and praise the wonderful consequences of the WSE, to get those people to agree, grumblingly. You estimate the white-shirt compliance rate will now be around 85%, which ought to be enough.

After this nice reminder that data science is by far not as straightforward (nor cheap) as its proponents claim, you turn back to your other duties, which have piled up a bit, and let your infrastructure accumulate data.

Disappointment

After six months, you evaluate your data. Comparing the white-shirt days to the non-white-shirt days, you find that the defect insertion rates per line-of-code added or modified is lower for only 6% of your engineers, is higher(!) for 9% of them (Ouch! Maybe those programmers who claimed "I *cannot* work in a shirt" were at least partially right? You swear because you recognize you have not recorded who they were.), and is not significantly different for the rest.

The WSE does not appear to exist.

The Right Reaction

There are two possible reactions at this point:

1. Sigh, admit defeat, give away the cameras to the engineers, and hope everybody quickly forgets your silly WSE project. Or:
2. Think "Hey, what's going on here?" and follow up to understand *why* the WSE is not working as you expected.

It should be obvious that only option 2 respects the spirit of data science. Besides, with option 1 you would have wasted a lot of effort and money.

How might the follow-up work? Well, hopefully you did not pursue a WSE revelation without having any idea of how and why such an effect would arise. (If you did, option 1 may be preferable, really.) Rather, you will have some mental model of the causation mechanisms underlying the effect and can now investigate intermediate stages to understand why the end result is different from your expectation.

In the WSE case, the model might say

1. white shirts radiate an air of purity
2. therefore, engineers will be less likely to interrupt a colleague wearing a white shirt,

3. therefore, that colleague can work more undisturbedly,
4. which is known to reduce the frequency of mistakes.
5. To a lesser degree, the wearer of the shirt will also feel purer,
6. therefore, s/he will be less inclined to write unclean, disorderly code,
7. and unclean code is known to at least tend to increase the frequency of mistakes.

You could now go and check each of these items for instance 1, 2, 3, 4, 5, 6 by means of surveys and 4, 7 by re-reading the literature. Thinking about the problem more closely, you might come up with interesting follow-up studies to be performed in your organization. For instance, you might be able to use the "local history" functionality of your engineers' IDEs to detect interruptions and then check whether they correlate with wearing white shirts or not (using the white-shirt bit or a smaller-scale manual data collection) and correlate with the insertion of defects or not (using DCDIM).

Following up in this manner has several positive effects:

- your understanding of your organization will increase, so that future improvement ideas are more likely to be helpful;
- your self confidence and hence curiosity (an important resource!) will be restored;
- you will learn to find cheaper ways for validating new ideas incrementally;
- you may stumble over other problems worth addressing.

In the WSE case, learning to think more incrementally would probably be the most valuable outcome of following up: For instance, invalidating the connection from wearing white shirts to fewer interruptions would have invalidated the WSE expectation much more cheaply.

Practical Advice

Trying to generalize the above discussion, we come up with the following data science rules for the testing of expectations for which not all data is yet available.

Always think of a causation model

Learning from failed expectations is possible only when you can dissect your observations and derive intermediate insights. A causation model provides intermediate points of reference at which you can get those insights. Without a causation model, there is no chance to follow up in this manner and failing expectations will waste a lot of resources.

Think of a causation model, *before* you check the expectation

The causation model usually provides opportunities for *partial* checks of your expectation that are so much cheaper that you should usually do them first.

Be wary of failing expectations

Being aware that hypotheses are often wrong provides the best motivation to come up with such partial checks. Try to do this, no matter how enthusiastic you may be about your hypothesis initially.

Be ready to accept inferior types of evidence

Even the partial checks may be shockingly expensive if you insist on data that is highly reliable and objective. Cheaper data sources, such as the surveys in the WSE example, are often preferable for initial validation, even if they are noisy, biased, or even (shudder!) subjective.

For researchers: Know the FNR

If you are an academic researcher who wants to publish, you know that studies about expectations that failed tend to be harder to publish, because they rarely provide immediate engineering progress. Be aware that the Forum for Negative Results (FNR), a permanent special section of the Journal of Universal Computer Science (J.UCS) specifically calls for such submissions: Stories of failed expectations that come with explanation.