

# WORK-AS-IMAGINED SOLUTIONEERING: TEN TRAPS ALONG THE YELLOW BRICK ROAD

On major projects, some surprises unfold slowly via ‘work-as-imagined solutioneering’. Based on observations in several industries, **Steven Shorrock** presents ten traps that we can all fall into.

In the book *The Wonderful Wizard of Oz*, Dorothy is lost in a faraway land, and must travel the “road of yellow brick” to the Emerald City, where she will find Oz, the Great Wizard, who could help her get back to Kansas. Along the road, Dorothy is joined by three characters also in need of help from the Wizard: the Scarecrow who is in need of a brain, the Tin Woodman who is in need of a heart, and the Lion who is in need of courage. The three join Dorothy and her dog Toto on the yellow brick road, only to find their journey tormented by hazards and traps. Some of these are simply troublesome, like uneven, broken and missing bricks, and branches blocking the path. Others are deadly, including a very deep and wide ditch with “many big, jagged rocks at the bottom”, a “pack of great wolves”, a “great flock of wild crows”, a “swarm of black bees”, and “monstrous beasts with bodies like bears and heads like tigers”.

The road symbolises a path to a solution, but the road was not as imagined. And as it turned out, neither was the solution. At work, the chances are that you have come across a designed ‘solution’ that that did not solve the problem, perhaps even making your work more difficult. It could be a new computerised system, a new policy, or new performance target. Perhaps you’ve even found yourself on the yellow brick road yourself, blindsided by traps along the way.

In this article, I outline ten such traps on the yellow brick road to problematic

solutions. The traps are presented in the typical sequence in which they arise in a process that I will call *work-as-imagined solutioneering*.

## Trap 1. Complex problem situation

The process of work-as-imagined solutioneering starts with a complex problem situation. Complex problem situations occur in systems with:

- a variety of stakeholders with conflicting goals,
- complex interactions between stakeholders and other elements of the socio-technical system (visible and invisible, known and unknown, designed and evolved, static and dynamic),
- multiple constraints (social, cultural, procedural, technical, temporal, economic, regulatory, legal, etc), and
- multiple perspectives on the nature of the problem.

This is the first trap. In complex problem situations, problems tend to be interconnected to form what Russell Ackoff – one of the grandparents of modern systems thinking – called a ‘mess’: a system of problems. Solving one isn’t enough.

“Complex problem situations are hard to understand and have no obvious solutions. This is unappealing to most people.”

## Trap 2. Complexity is reduced to something simple

Complex problem situations are hard to understand and have no obvious solutions. This is unappealing to most people. Understanding complex problem situations requires that we seek to understand:

- the various expressions of, and influences on, the problem,
- the stakeholders or people that influence the situation, and those affected,
- the work affected,
- the various contexts of work (e.g., physical, ambient, social, cultural, technological, economic, organisational, regulatory, legal), and
- the history of the problem situation and system as a whole.

At least one of these forms of understanding is typically lacking (usually more than one, and sometimes all five). This is partly because getting this understanding requires trust and expertise, which are often in short supply. And it is partly because, once a problem is identified, there is a perceived urgency to do something in order to reduce anxiety.

So the critical activities needed to understand complexity are often neglected, and complexity is reduced to something simple, such as ‘poor performance’, ‘non-compliance’ or ‘human error’. The second trap has been set.



**Trap 3. Someone has a ready-made solution**

While there may be little understanding of the complex problem situation, solutions are at hand. Past experience, ideas from other contexts, committee-based idea-generation, or diktats from authority figures make a number of appealing ‘solutions’ available. These form the third trap. Examples include:

- rules
- procedures
- checklists
- mandatory training
- commercial off-the-shelf products
- ‘automation’
- quantified performance targets and limits
- measures
- reporting lines
- performance reviews
- incentives
- punishments, and
- reorganisation.

Most of these are not inherently bad. What is bad is introducing them – any of them – without a proper understanding of the context and the problem situation within that context. But the focus soon turns to the ‘solution’.

**Trap 4. Compromises to reach consensus**

As the solution is revealed, people at the blunt end are now at the sharp end of a difficult process of design and implementation. There is a lack of expertise in how to do this, and disagreements emerge as people start to see a number of complications. But consensus and the stability of the implementing group is critical, and this is the foundation of the fourth trap. The idea is put out for comment, usually to a limited audience. There are further insights about the problem situation and context system, but these arrive in a haphazard way, instead of through a process of understanding involving design and systems thinking. Eventually, compromises are made to achieve consensus and the ‘solution’ is specified further. Then plans are made for its realisation. The potential to resolve the problem situation is hard to judge because neither the problem situation nor the context is properly understood.

**Trap 5. The project becomes a thing unto itself**

The focus now turns to realisation. The problem situation and context, which were always out of focus, are now out of view. The assets and real needs of all stakeholders were never in view, but the needs of the stakeholders who are invested in the roll-out of the solution have been met: they can now feel reassured that something is being done. The focus now switches from *what* to *how*: how can we implement this idea? Often this involves a heavy and inflexible plans, processes, structures, tools, management systems, and documentation requirements.

“At work, the chances are that you have come across a designed ‘solution’ that that did not solve the problem, perhaps even making your work more difficult.”

**Trap 6. Authorities require and regulate it**

As the ‘solution’ gets more attention, authorities come to believe that it is a Good Thing. Sometimes, solutions will be mandated and monitored by those with regulatory power, but detached from the context of work. Now there is no going back (except to Trap 4 and 5).

**Trap 7. The solution does not resolve the problem situation**

The solution is deployed, but it is not even the same as the original idea. More compromises have been made along the way, in terms of the concept, design, or implementation (or all three). An unwanted surprise emerges at this point: the problem remains (albeit in a different form)! The feedback loops from the sharp end to the blunt end, however, contain delays and distortion.

**Trap 8. Unintended consequences**

Not only does the solution not resolve the original problem, but it also brings new problems that were never imagined! In general terms, this might mean more demand, more pressure, more friction, more complexity, or more use of resources. Such surprises often appear in the interfaces between different stakeholders, departments, organisations, etc. The parts of the system just don’t fit. This may relate to the provision of monitoring, analysis, tools, materials, and technical support. Or it might just be that the deployed ‘solution’ cannot even function as intended, designed or implemented.

**Trap 9. People adapt and game the system**

At this point, operational work has to continue, somehow, *despite* the ‘solution’. And so it is necessary to adapt and compensate. Many work-as-imagined solutions can be worked-around (e.g., ‘gaming the system’). This is typical of measures (especially when combined with targets or limits) and processes, but we also work around clumsy technology, or indeed any of the ‘solutions’ listed under ‘Trap 3’. Have a think about how you have worked around each of them.

**Trap 10. It looks like it works**

The adaptation and gaming, combined with feedback lags and poor measures, give the illusion that the deployed solution is working, at least to those not well connected to the context of work-as-done. By not illuminating work-as-done, which is successfully compensating for and hiding the flaws in work-as-imagined, the illusion of successful implementation is maintained. This trap is almost invisible.

Of course, there may well be a vague sense that there are ‘teething issues’, but this is easily rationalised away. Too often, we are left with gaps between the four ‘varieties of human work’: work-as-imagined, work-as-prescribed, work-as-done, and work-as-disclosed (Shorrock, 2016). There is a lack of alignment between how people think others work, how people are supposed to work, how people say they work, and how people actually work.

By this stage, the project team that worked on the originally intended solution has probably moved on. The deployed system remains and now we must imagine a solution for both the original problem and the new problems.

## Back to the Yellow Brick Road

In the book, which is rather different to the film, the traps are of course quite different to those above. But some are analogous. Interestingly, it is the Great Wizard who adapts and games the system (Trap 9): Dorothy's three companions are fooled into receiving convincing counterfeits.

*"Oz, left to himself, smiled to think of his success in giving the Scarecrow and the Tin Woodman and the Lion exactly what they thought they wanted. How can I help being a humbug," he said, "when all these people make me do things that everybody knows can't be done? It was easy to make the Scarecrow and the Lion and the Woodman happy, because they imagined I could do anything. But it will take more than imagination to carry Dorothy back to Kansas, and I'm sure I don't know how it can be done."*

**"Not only does the solution not resolve the original problem, but it also brings new problems that were never imagined!"**

Indeed, the Wizard did not take Dorothy back to Kansas. How she got back was not how she imagined.

The story, and our experience, reminds us that top-down work-as-imagined solutioneering – like everything else – has limits. In the end, it tends not to solve the original problem and comes with unintended consequences, which are compensated for in ways that are hard to see.

So, next time you notice a 'problematic solution', either developing or deployed, perhaps it is worth trying to understand how it came to be. How did the 'solution' itself make sense during the process of its development? If work is now more difficult and less effective, the chances are that you will find a few of the traps above, which – by the way – we can all fall into. But more importantly, perhaps you can intervene to help realign work-as-imagined with work-as-done. **S**

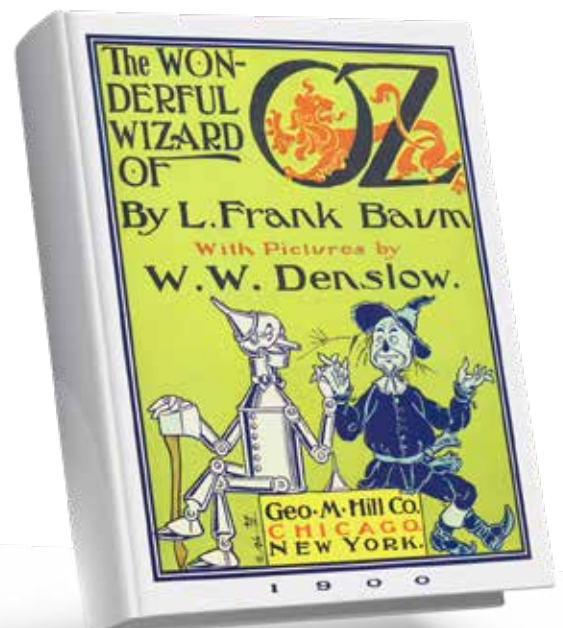
## Reference

Shorrock, S. (2016, 5 December). The varieties of human work. *Humanistic Systems*. <https://humanisticsystems.com/2016/12/05/the-varieties-of-human-work/>

This article is adapted from Shorrock, S. (2018, 3 June). Work-as-imagined solutioneering: A 10-step guide. *Humanistic Systems*. <https://humanisticsystems.com/2018/06/03/work-as-imagined-solutioneering-a-10-step-guide/>



Dr Steven Shorrock is Editor-in-Chief of *HindSight*. He works in the EUROCONTROL Network Manager Safety Unit. He is a Chartered Psychologist and Chartered Ergonomist & Human Factors Specialist with experience in various safety-critical industries working with the front line up to CEO level. He co-edited the book *Human Factors & Ergonomics in Practice* and blogs at [www.humanisticsystems.com](http://www.humanisticsystems.com)







AND NOW FOR SOMETHING COMPLETELY DIFFERENT

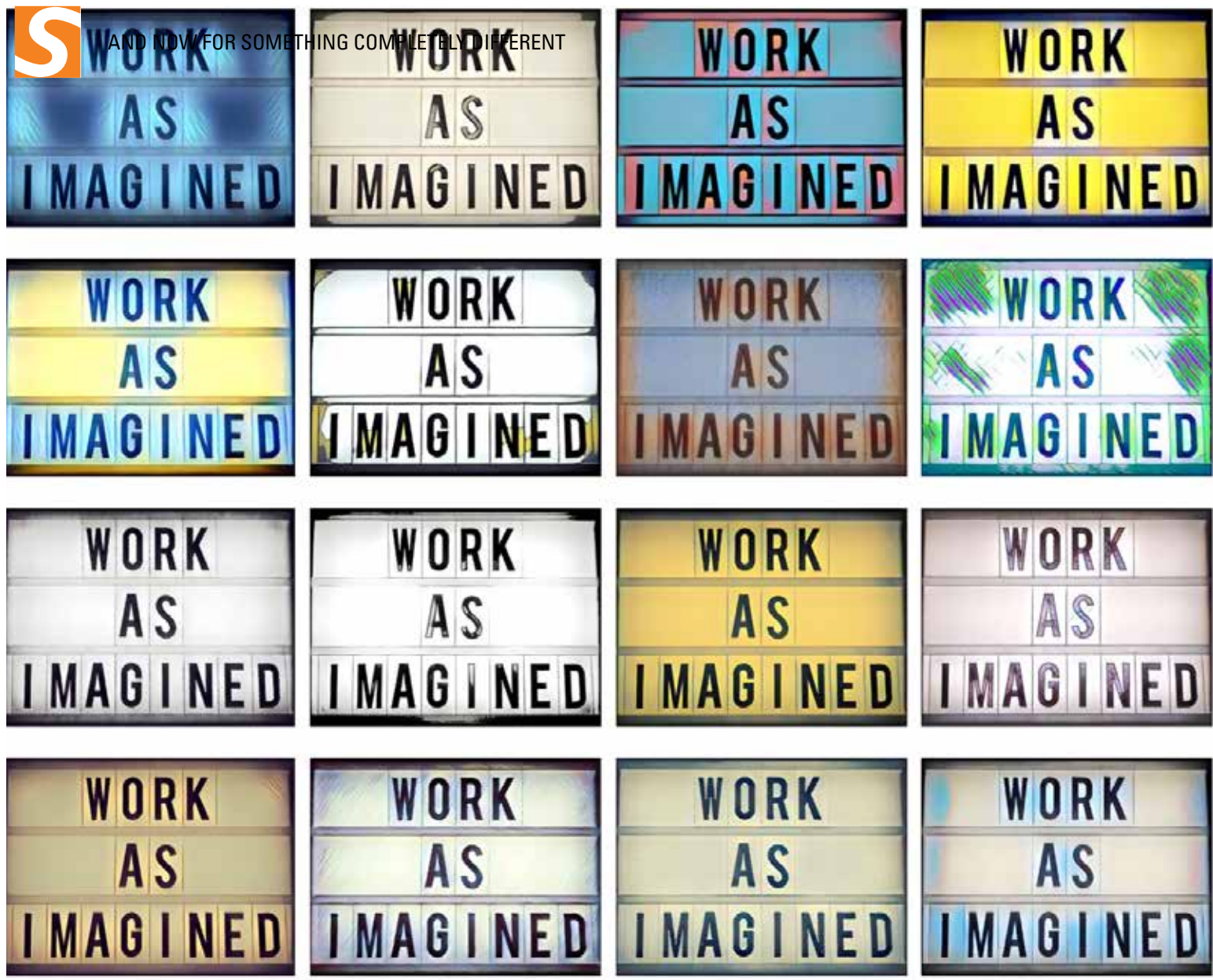


Photo: Steven Shorrock

“Over the last few years there has been a call to enshrine ‘saying sorry’ in law. This became the ‘duty of candour’. When this was conceived it was imagined that people would find the guidance helpful and that it would make it easier for frontline staff to say sorry to patients when things have gone wrong. Patient advocates thought it would mean that patients would be more informed and more involved and that it would change the relationship from an adversarial to a partnership one. In practice this policy has created a highly bureaucratic process which has reinforced the blame culture that exists in the health service. Clinical staff are more fearful of what to say when something goes wrong and will often leave it to the official process or for someone from management to come and deliver the bad news in a clinical, dispassionate way. The simple art of talking to a patient, explaining what has happened and saying sorry has become a formalised, often written, complied duty. The relationships remain adversarial and patients do not feel any more informed or involved as before the duty came into play.”

**Suzette Woodward, Patient Safety Lecturer and Former Paediatric Intensive Care Nurse**

“With the installation of a fully computerised system for ordering all sorts of tests (radiology requests, lab requests, etc.) work-as-imagined (and work-as prescribed) was that this would make work more efficient and safer, with less chance of results going missing or being delayed. Prior to the installation, there was much chat with widespread talk of how effective and efficient this would be. After installation, it became apparent that the system did not fulfil the design brief and while it could order tests it could not collate and distribute the results. So work-as-done then reverted to the system that was in place before where secretaries still had to print results on bits of paper and hand them to consultants to action.”

**Craig McIlhenny, Consultant Urological Surgeon**