Myths and Misconceptions in Cybersecurity

October 2024

Gene Spafford @spaf@mstdn.social

spaf@cerias.purdue.edu

How is a Field Defined?

- History
- Some accepted body of knowledge
- Generally-accepted references & standards
- Common terminology

- Myths and misconceptions are when the above are not widespread

Cybersecurity isn’t new

- The Ware Report — 1967
- Project MULTICS — 1969
- The Anderson Report — 1970
- Trusted Computer System Evaluation Criteria (Orange Book) — 1983

Standard & Common

- Several versions of CBoK, such as that for CISSP, A+, UK CyBoK
- Several good standards for both practice and personnel
  - NIST especially, with CSF and related
  - ISACA, (ISC)^2
  - BSA
  - ISO
- Standard references and textbooks
  - E.g., Bishop, Pfleeger, Bellovin, Anderson, Stallings, …
- Common terms of art
  - E.g., virus, ransomware, ROP, side channel

Some Significant Myths

- Myth #1: We have a clear definition of cybersecurity
- Myth #2: More technology is better
- Myth #3: Technology is the solution
- Myth #4: Patching is somehow security
- Misconception: speed and cost are most important

Myth #1: We have a clear definition of cybersecurity

Cyber Security

Let’s start with an intuitive definition: a system is secure if it is protected against all forms of threat.

Can we achieve that? Let’s give it some thought.

Cyber Security

Random hackers?

Check!

Cyber Security

Malware?

Probably.

Cyber Security

Nation State Hackers?

Probably not.

Cyber Security

UFO Invasion?

What? No!

Cyber Security

Extinction Event Meteor Impact

Definitely not.

Cyber Security

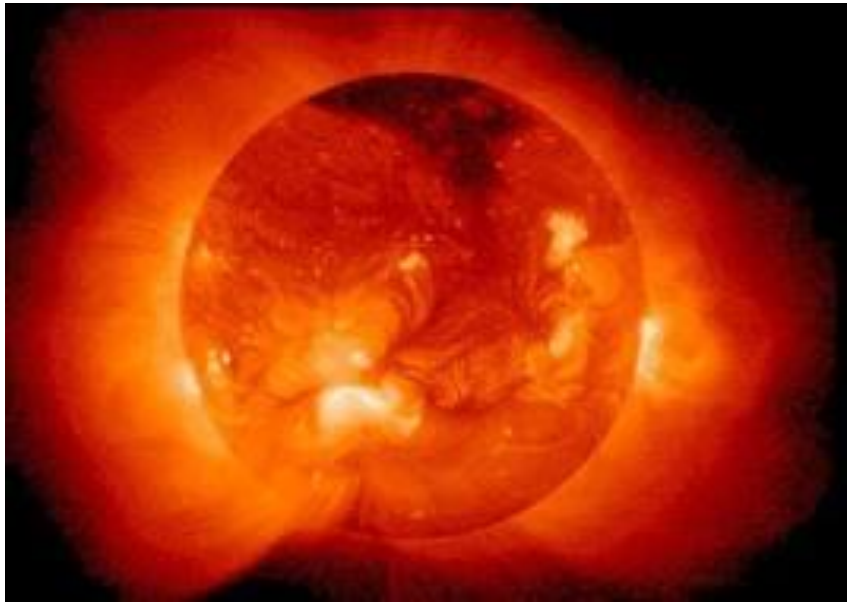
Maybe if we set up colonies on Mars and gave them backup copies?

Cyber Security

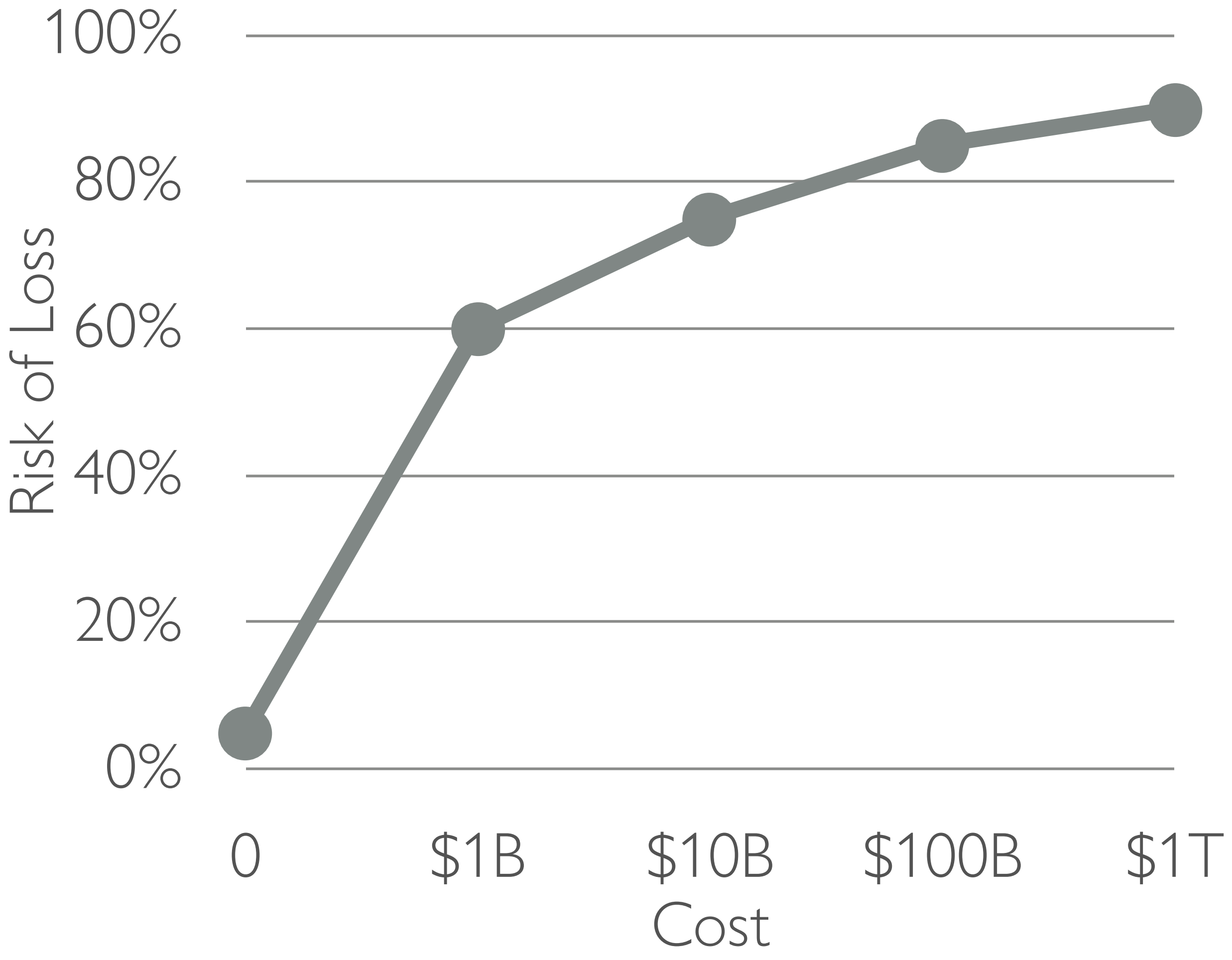
Maybe if we set up colonies on Mars and gave them backup copies?

No, eventual death of the Sun will mean end of the inner planets.

And I know what you're thinking…eventually, we have "heat death" of the Universe.



Cyber Security



As a definition, maybe that isn’t helpful — we can’t ever achieve it.

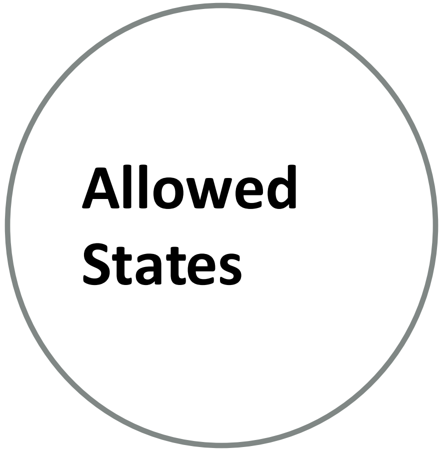
Actually, this exposes an issue: security is an economics issue more than an engineering task — how much to spend to minimize risk

Another Attempt

Let’s approach this as a problem of system design. Can we do a better job?

YES NO MAYBE

Research in the 1970s and 1980s looked at system state.

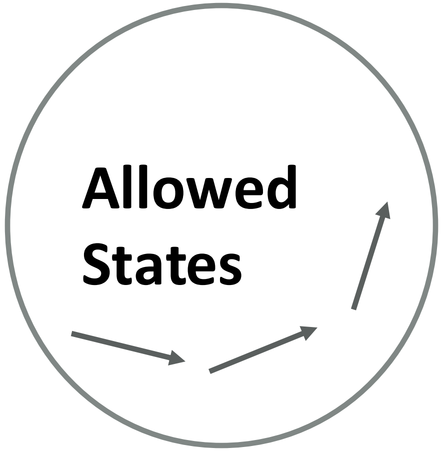


Allowed States

There are a set of states that are defined to be “okay” or “safe.”

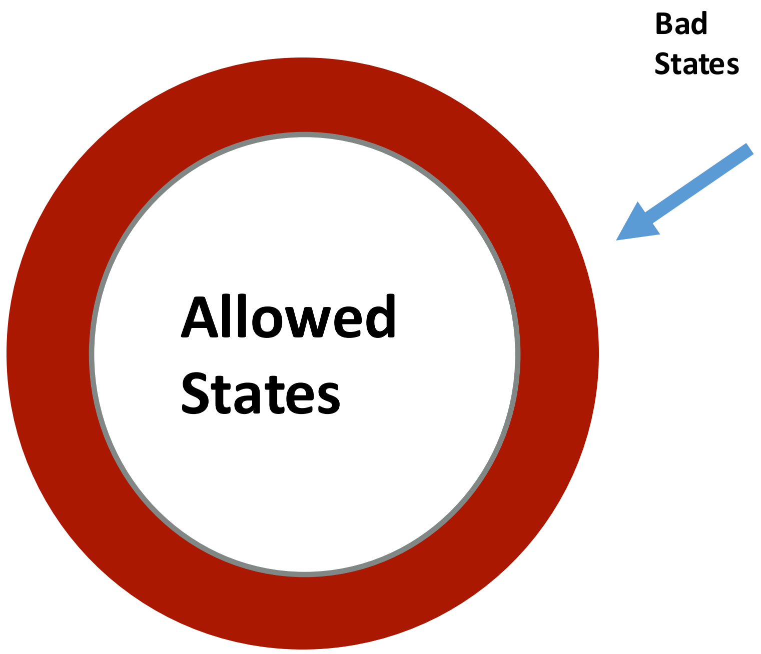
(Depends on policy, such as Bell-LaPadula or Biba or….)

As a system executes, it changes state.



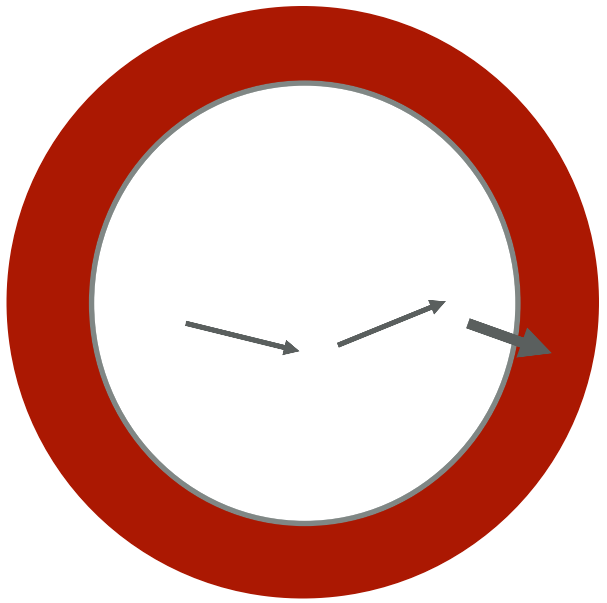
Each *valid* operation results in a state of the system that is also defined to be “okay.”

(Depends on policy, such as Bell-LaPadula or Biba or….)



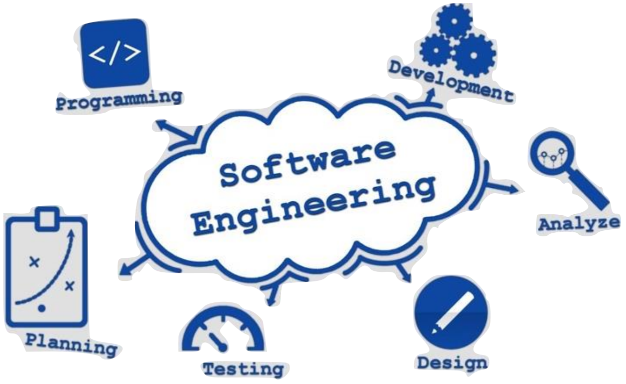
We also have "bad" states. We don't want these to occur.

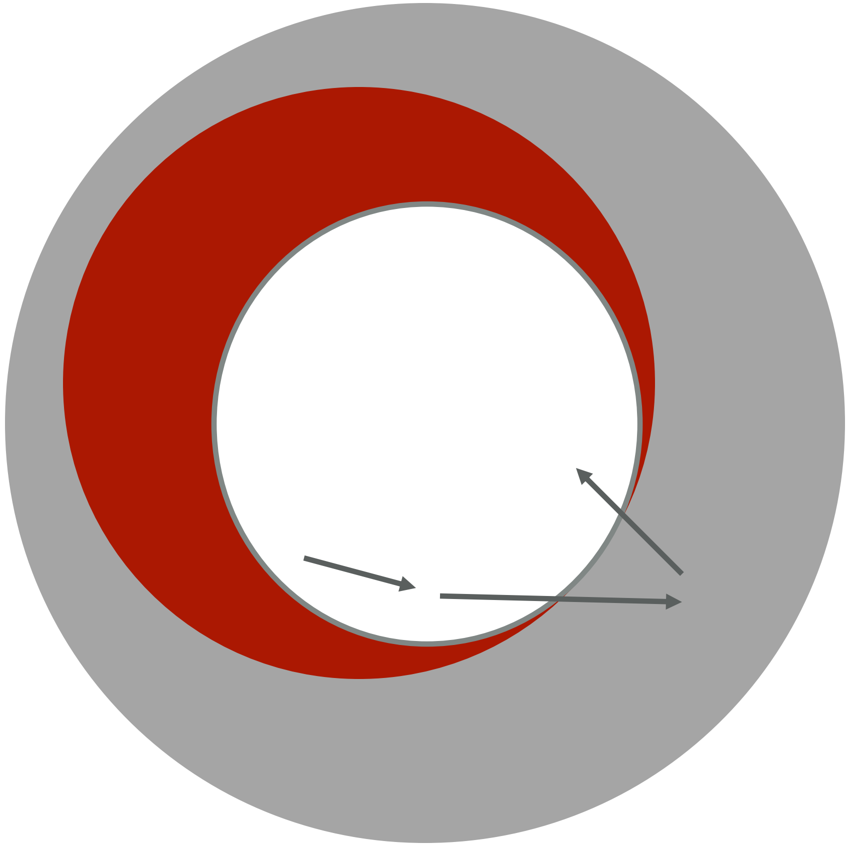
We don’t want to enter “bad” states.
We especially don’t want to remain in them!



- This notion of “allowed states” is a match to the concept of “system specification” in software engineering.
- Execution of a state not in the specification is a “fault” that can result in a “failure.” A failure in a protected system is a security failure.

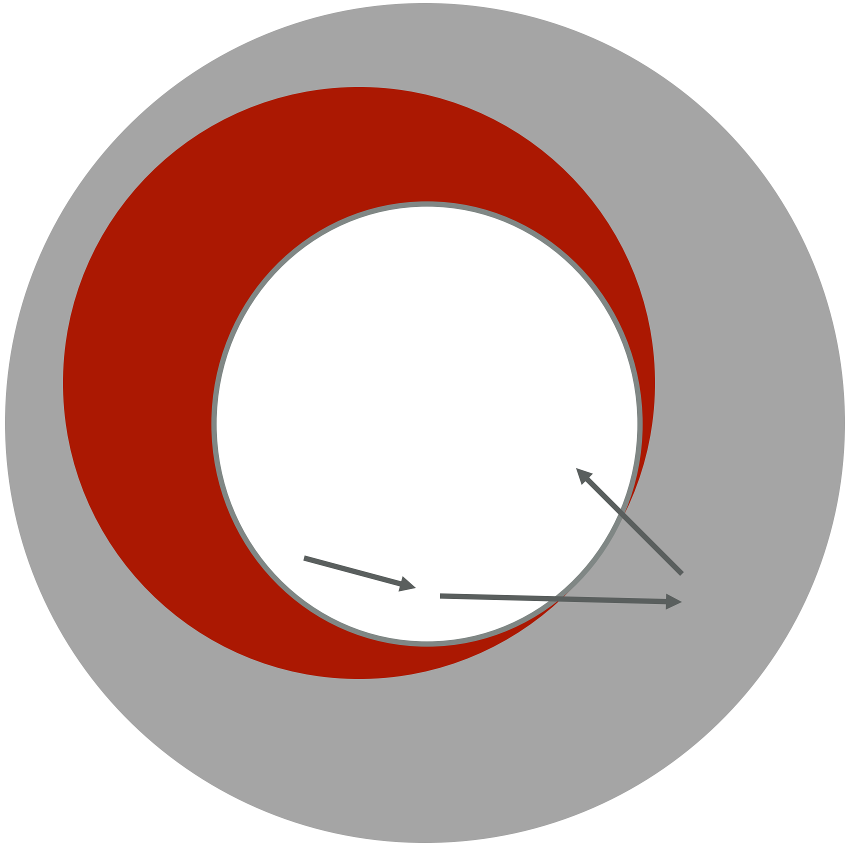
(Yes, security was a driver in the development of software engineering.)





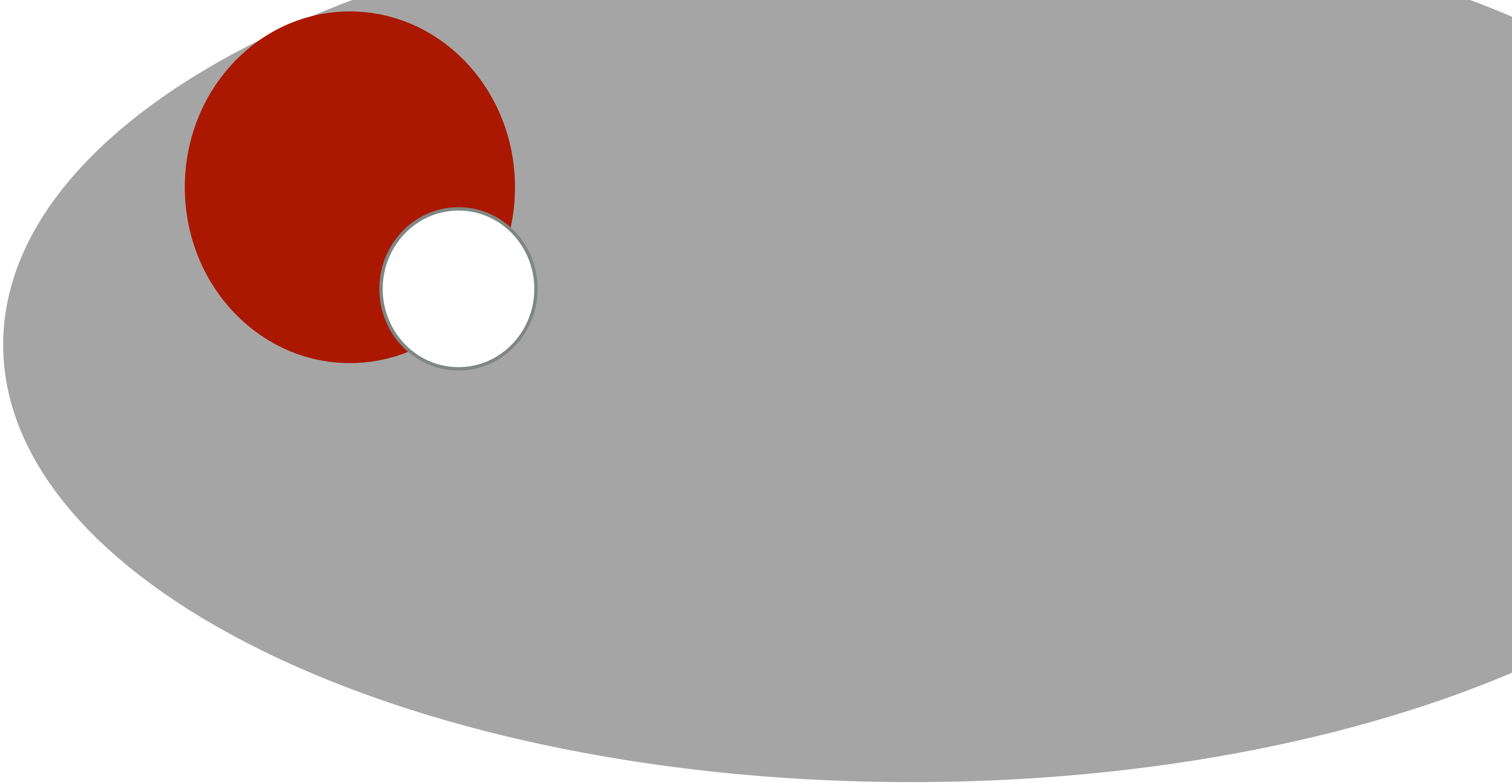
We also have “undefined” states. These aren’t specified.

Entering undefined states is an error. This may lead to a fault.



Undefined states might not be “bad” states.

They might even lead back to “okay” states.
Because they are undefined, we do not know.



What it probably really looks like

- Most software today operates in the “undefined” state space because we have never defined its proper behavior.

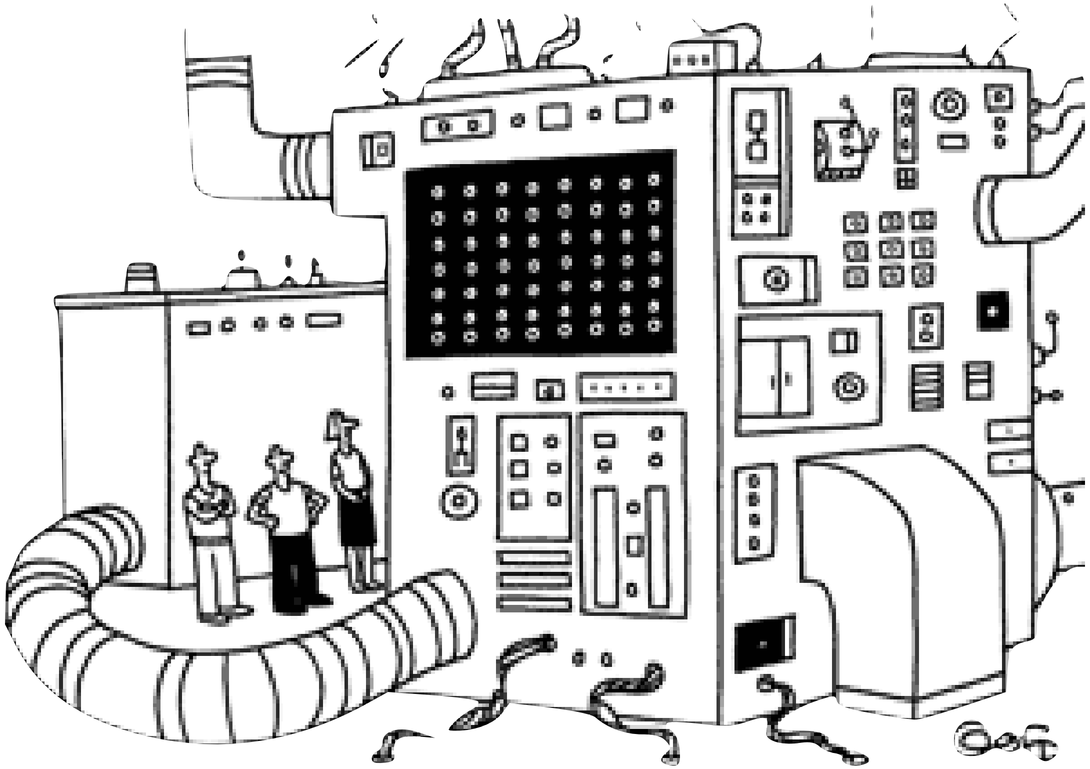
- Formal specifications are time-consuming and expensive. They also require expertise to define, and to build software to match.

- We have only general requirements, and no detailed specifications. We usually don’t fully define “correct” with respect to either!

A Consequence of “Design”

A program that has not been specified cannot be incorrect; it can only be surprising.

<u>Proving a Computer System Secure</u>, W. D. Young, W.E. Boebert and R.Y. Kain, The Scientific Honeyweller (July, 1985), vol. 6, no. 2, pp. 18-27.



“It was just going to be a laser printer before we started adding features.”

```
irror_mod = m
et mirror object to mirror
irror_mod.mirror_object
operation == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
 operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
 operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

selection at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.active
("Selected" + str(modifier
irror_ob.select = 0
 bpy.context.selected_obj
ata.objects[one.name].sel

int("please select exactly

- OPERATOR CLASSES ----

types.Operator):
 X mirror to the selected
ject.mirror_mirror_x"
rror X"

context):
ext.active obje
```

We Don’t Have a Clear Definition

- What is security?
  - Security -> Trusted -> Trustworthy -> Resilient -> Risk
- Without a clear definition, we don’t have metrics.
- What about safety? Privacy?
- We get stuck with folk wisdom and old concepts

Why Is a Definition Helpful?

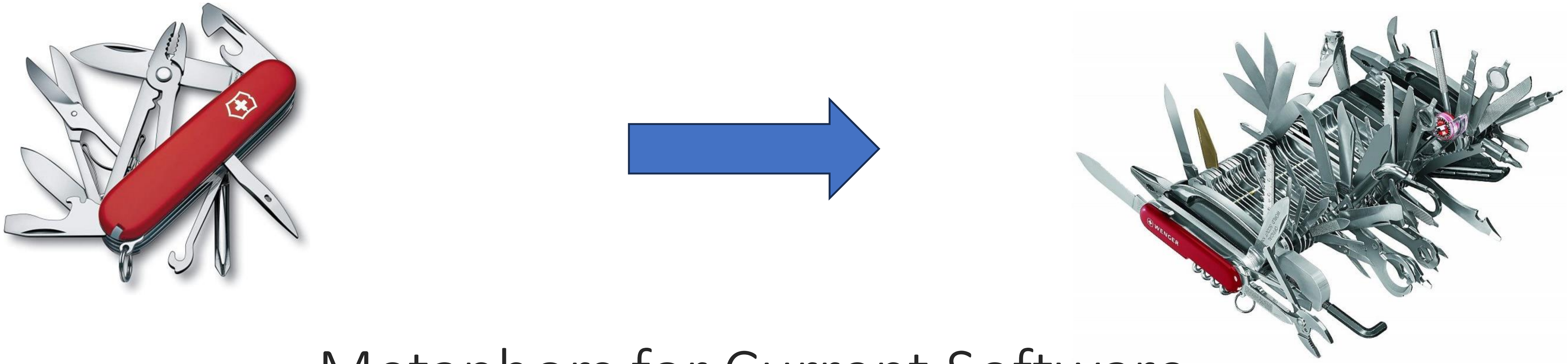
- Drives principles of design and operation
- Enhances communication of goals
- Supports development of tools
- Enables developments of metrics

Consider: How is security related to privacy? To safety? To reliability?

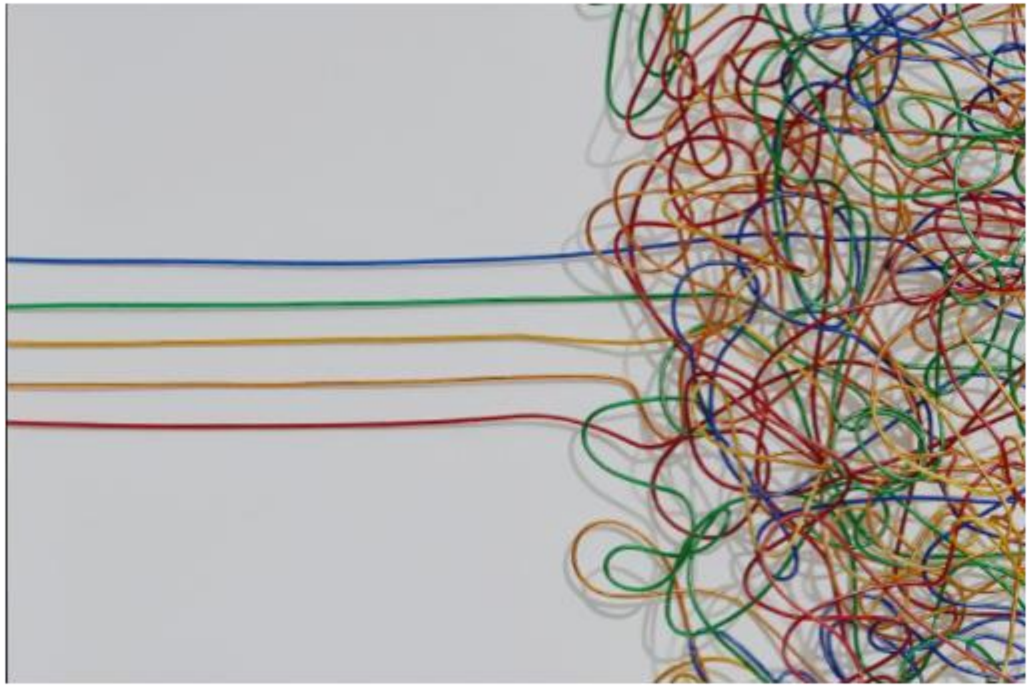
Myth #2: More technology is better

We Don’t Value Simplicity

- We can’t define and design software well
- Complexity is killing us
- Legacy is a huge part of the problem
- We are stuck in a loop, fixing broken things and building on top of software that is fundamentally unsound
- Leads us to avoid investigating fundamental issues

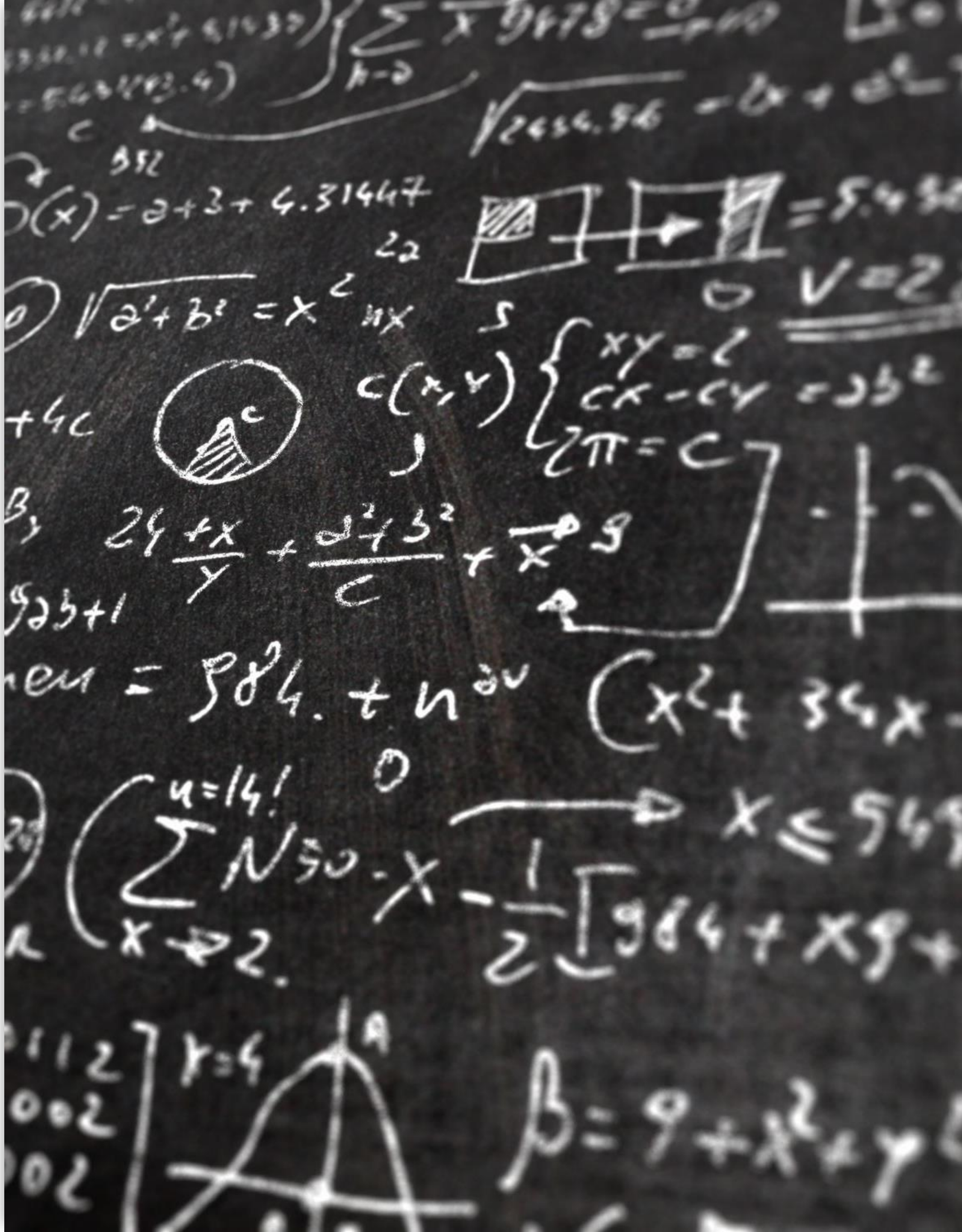


Metaphors for Current Software



Complexity Leads to Emergent Failures

- It is not the *sum* of the components – it is the *product*.
- Once we pass the point of understanding and modeling, we cannot be assured of the outcomes.



Myth #3: Technology is the solution

We Focus Too Much on Technology

Cybersecurity is a very specialized, technical field:

Attackers

Malware

IDS

Firewalls

Network traffic

Forensics

And not everyone using computing is technically-saavy.

Relying on Technology for Solutions Leads to Greater Complexity!

Courtney's third law:

There are no technical solutions to management problems, but there are management solutions to technical problems.

People Are Part of the System!

- We tend to design for our peers, not for the public
- We fail to understand concerns and limitations of the downstream users and operators
- We then blame the users when things go wrong

Consider

- How are systems designed for people with physical limitations?
- Design for an aging population?
- Design to accommodate different biometric profiles?
- Design for national differences in culture and language?
- Design for differences in literacy?

ALL Users Should be Part of the System!

- Education
- Awareness
- Non-punitive reporting
- Simplified interfaces
- Diversity of views
- Encouraging feedback

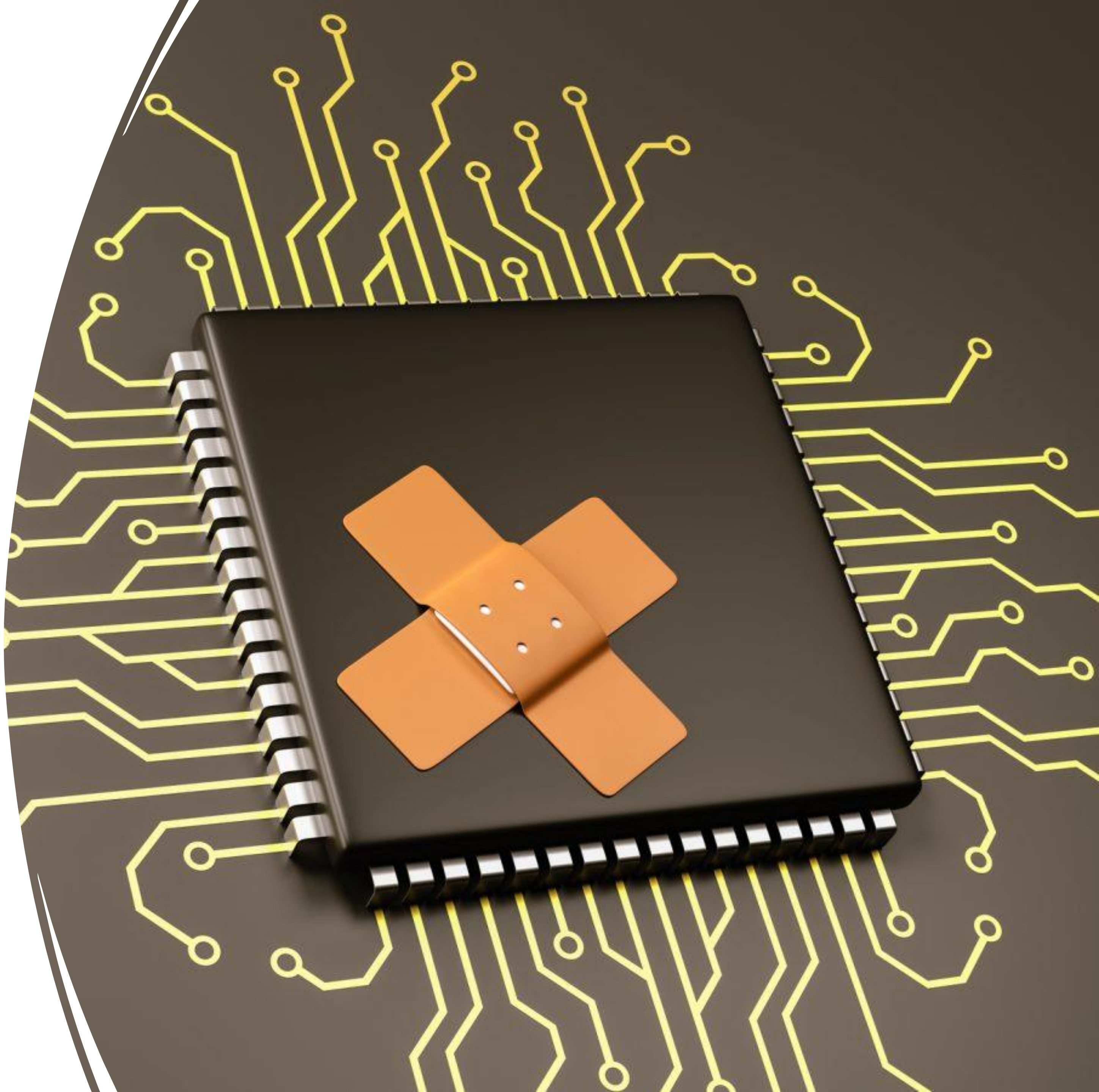
Myth #4: Patching is somehow security

Patching

Building correctly the first time is better than applying a patch – even applying one quickly.

Especially if the patch is after a security incident.

The market doesn’t demand correct code but does support pen testing and patching.

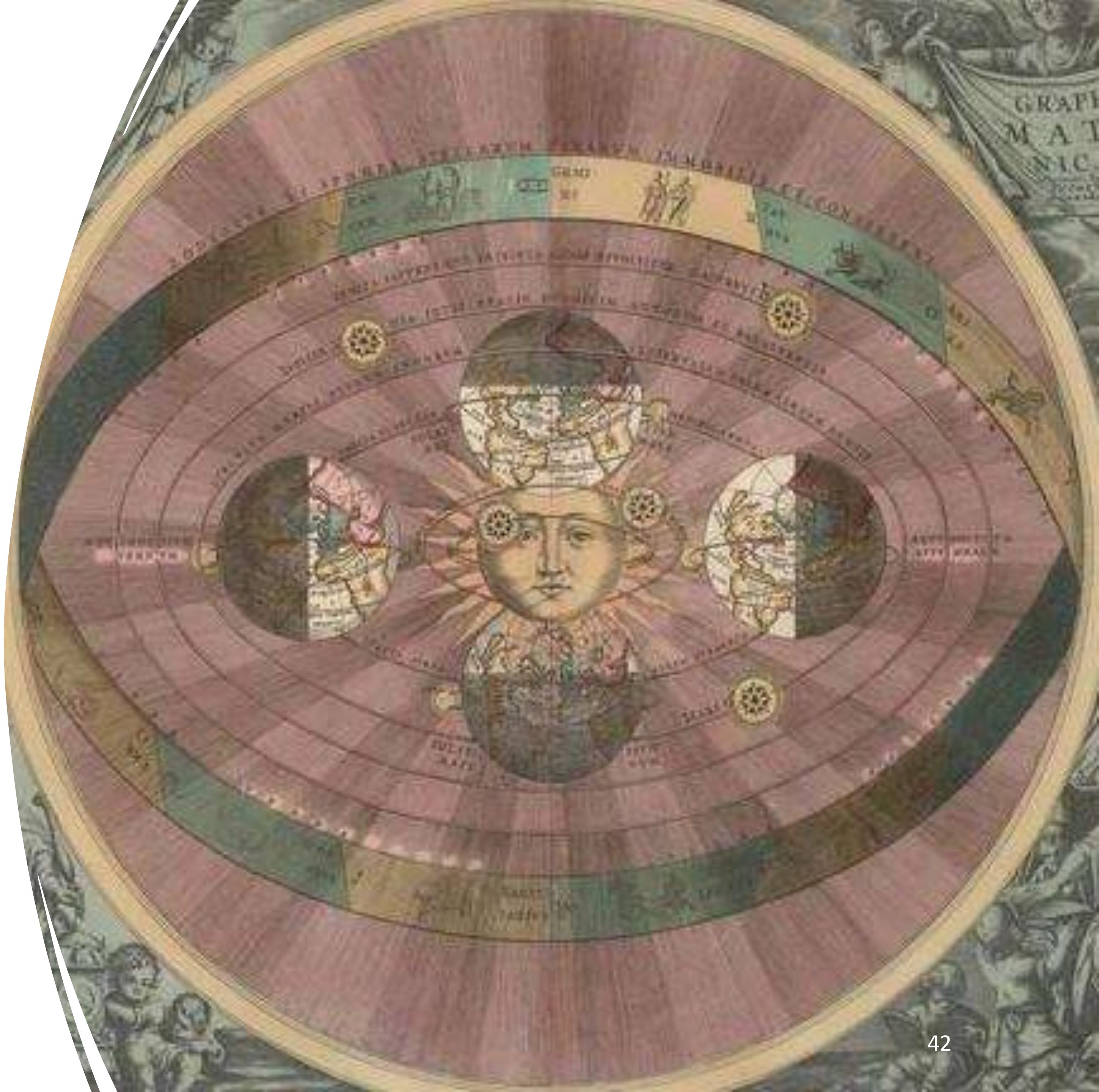


Patching Adds Complexity!

Ptolemy vs. Copernicus

- Ptolemaic view of computing — we continue to patch systems—it seems to work
- Copernican view is not appreciated because it costs money…and may not serve government interests
- However, current system is losing in facing the future. Inside the OODA loop (John Boyd)

Analogy courtesy of Richard Danzig



Misconception: speed and cost are most important

We Value Some of the Wrong Things

Why is time-to-market more important than quality?

Why is speed more important than safety?

Why is easy of patching more important than correct design?

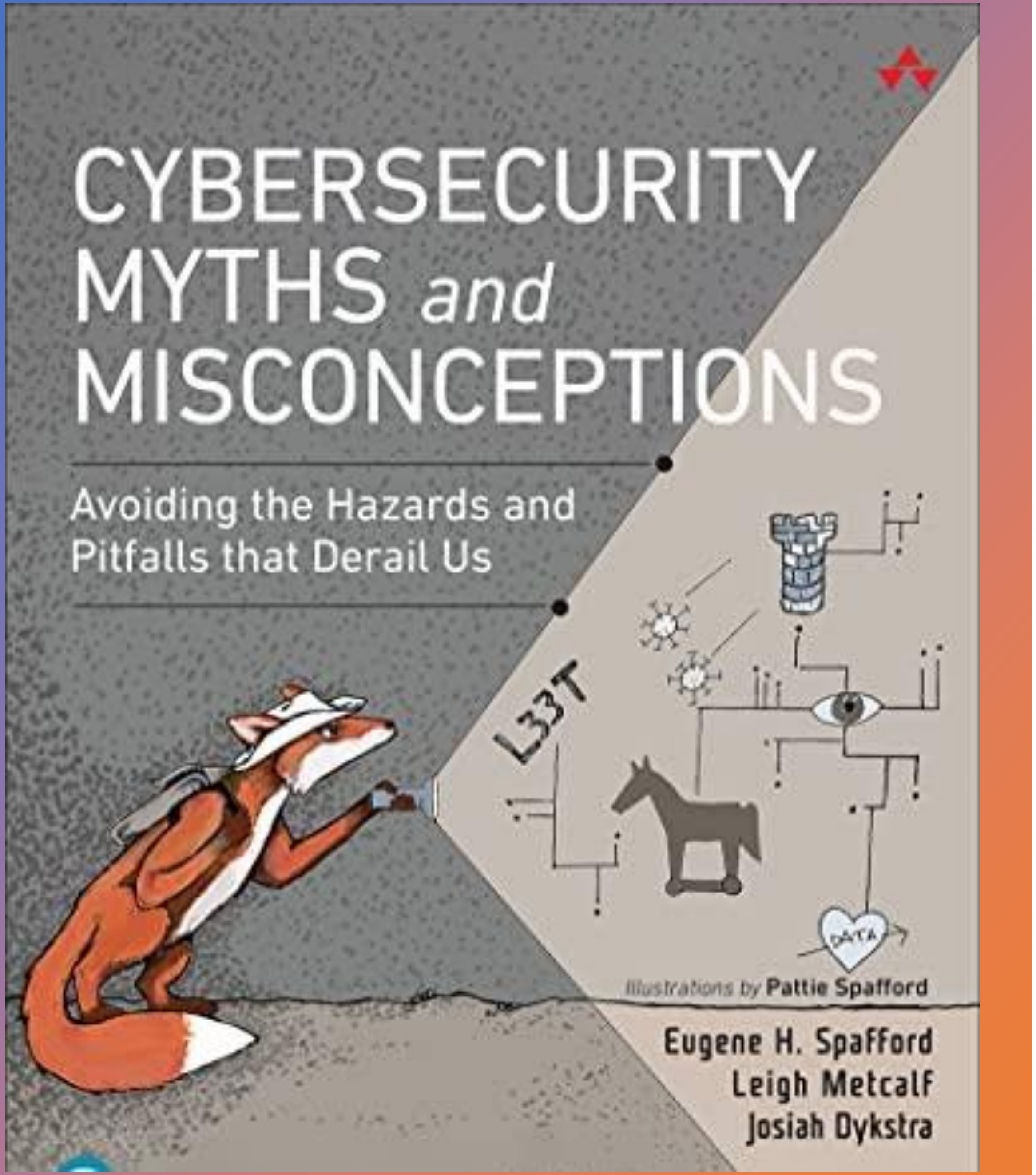
In Part, It Goes Back to Definitions

Not knowing a definition of security means we can't measure it. So, we use what we can measure.

- Cost
- Speed
- Time to change
- Lines of code

How Can We Change For the Better?

- Rethink current conventional wisdom
- Seek simplicity
- Think *whole systems,* including people
- Seek to promote good values


CYBERSECURITY MYTHS and MISCONCEPTIONS
Avoiding the Hazards and Pitfalls that Derail Us
L33T
DATA
Illustrations by Pattie Spafford
Eugene H. Spafford
Leigh Metcalf
Josiah Dykstra

Interested in Learning More?

Available via the usual bookstores and outlets, and

https://informit.com/cybermyths