

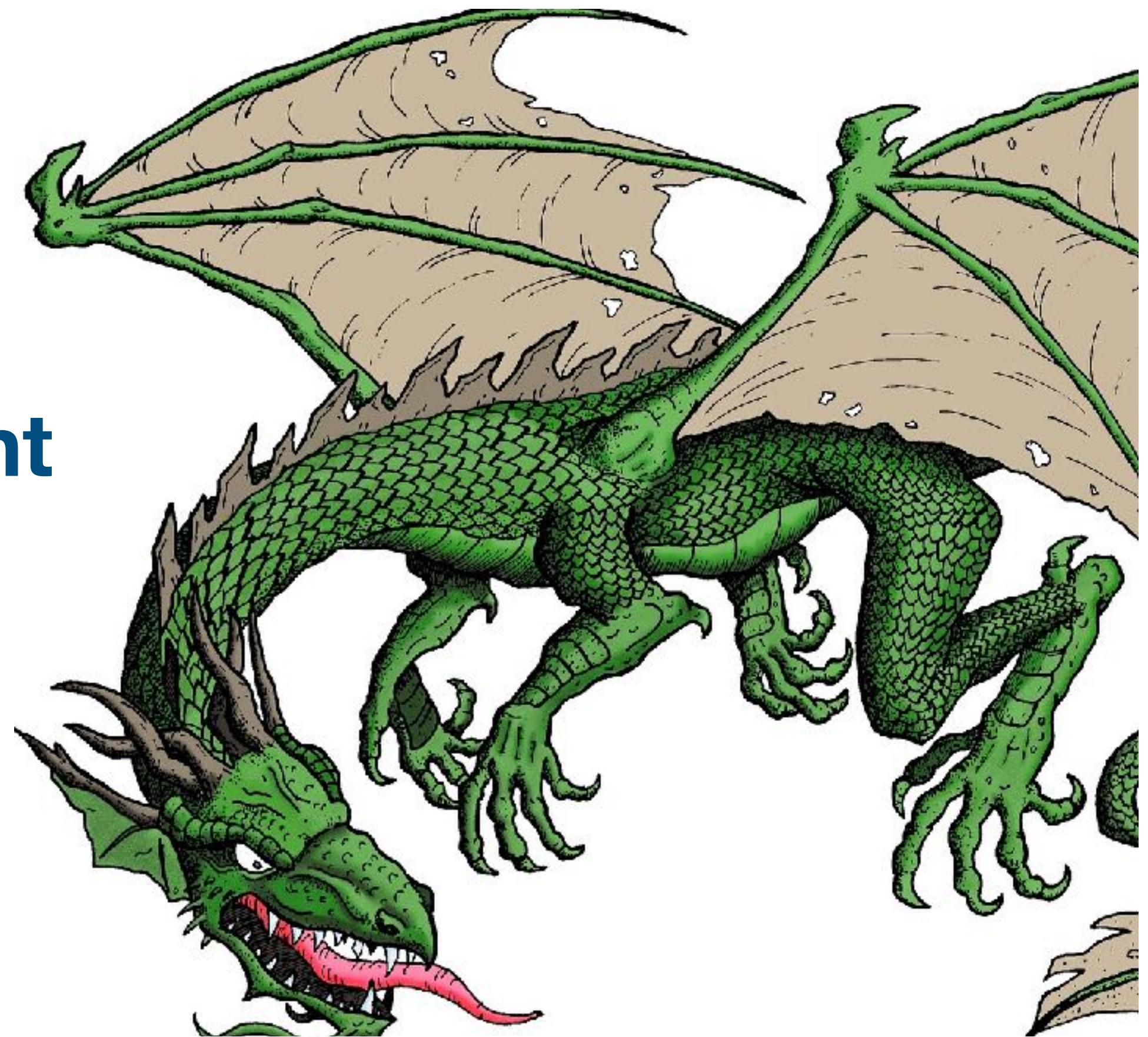
This slide deck for offline viewing:  
<https://github.com/ewbarnard/dragons>

# Here Be Dragons

Finding the Joy in Software Development

Edward Barnard, 2021 Midwest PHP

*Book coming Summer 2021*



# Our Agenda

- Gate keepers
- Dragons
- Imposters
- Wizards
- Product Managers



*Here Be Dragons: Finding the Joy in Software Development* coming Summer 2021

 @ewbarnard

# Here Be Dragons

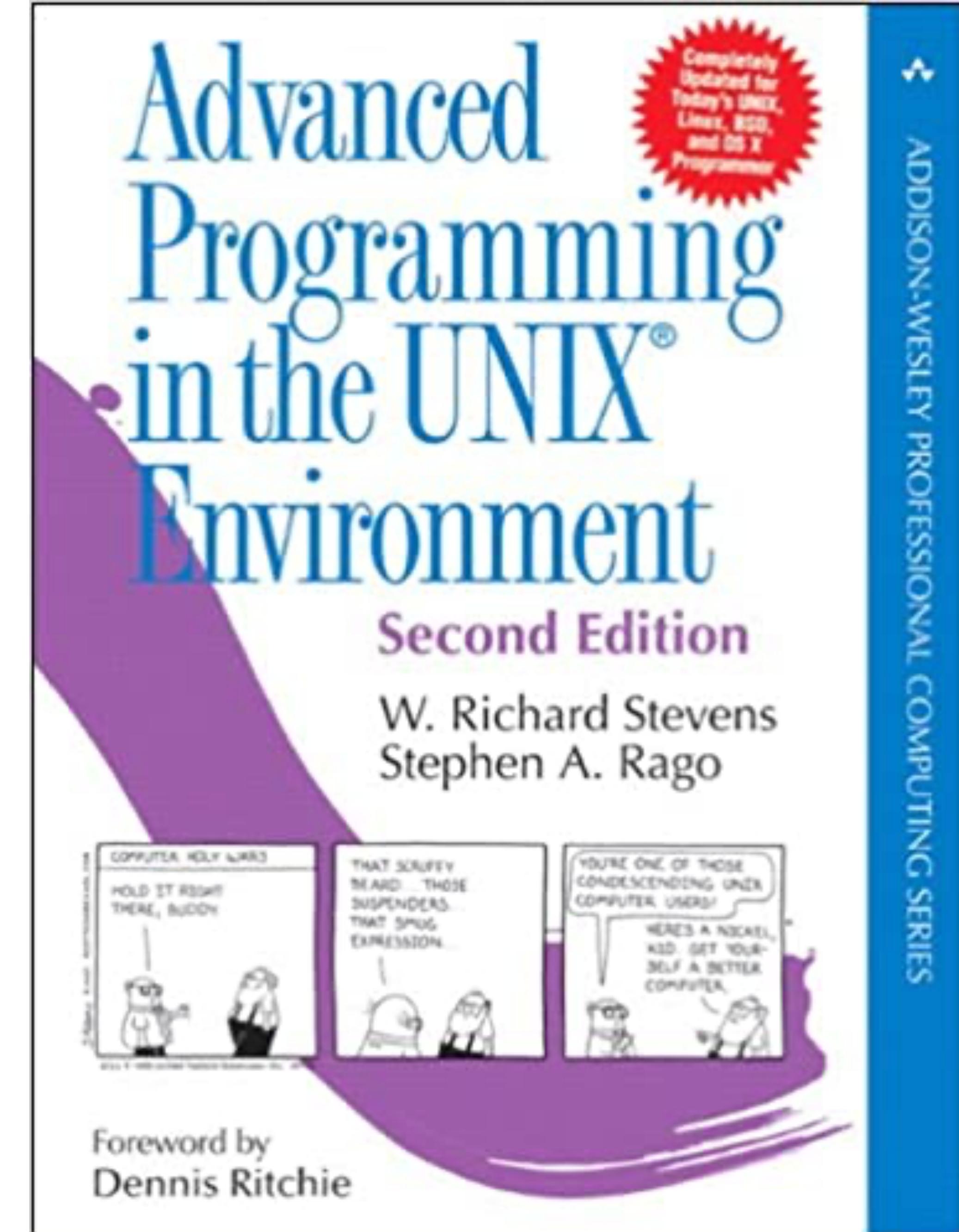
## So Say the Gatekeepers

- Strong tendency toward “gate keeping” in 2020s
- Senior developers warn “there be dragons” while announcing a “hands off” policy toward crucial areas of code base
- You and I need to get past that
- Where’s the fun when you can’t touch the dragons?



# Gate Keeping The UNIX Guru

- *Computer Holy Wars:* Hold it right there, buddy
- That scruffy beard... those suspenders, that smug expression
- You're one of those condescending UNIX computer users!
- Here's a nickel, kid. Get yourself a better computer.



# The UNIX Guru

## My Visits

- First visit — “it’s in the *man* page”
- Filesystem checks after editing the root filesystem in single-user mode — “it’s in the *man* page”
- Empty data structure passed to a system call and eventually passed to my device driver — “it’s in the *man* page”
  - But it wasn’t in the man page — “it’s in the code”
  - I showed him the code

# The UNIX Guru

Do you see the problem here?

- Our Guru did not want to “dumb down” the discussion for his audience
- If you want to work with the Guru, you need to rise to his or her level, not the other way around
- You’re expected to *work* for it, just like the Guru did in the first place
- Show a willingness to “pay your dues”

**With respect, I disagree.**

**“If I have seen further than others, it is by standing on the shoulders of giants.”**

**— Sir Isaac Newton (1675), likely quoting Bernard of Chartres (1159)**

# The Modern Guru

## Yours is the shoulder

- *Bring* people up to your level, in the small things and big things
- It's on *you* to explain over and over
- Work within the *other* person's context
- Example – helping a family member with phone privacy settings, explaining over and over, helping develop changed habits

# When you ARE the Guru

## Yours is the shoulder

- We don't extend the craft by expecting everyone to “pay their dues” like we did
- We extend the craft by helping others **exceed** our own mastery
- Yours is the shoulder of giants — it's on you to help others see further

**But what if I'm not the Guru?**

# *Dune* by Frank Herbert (1965), pp. 65-66

# Learn that you can learn

“Many marked the speed with which Paul learned the necessities of the planet Dune. The teachers, of course, know the basis for this speed. For the others, we can say that Paul learned rapidly because **his first training was in how to learn**. And the first lesson of all was **the basic trust that he could learn**. It is shocking to find how many people do not believe that they can learn, and how many more believe learning to be difficult. Paul knew that every experience carries its lesson.”

*Quote slightly edited for clarity.*

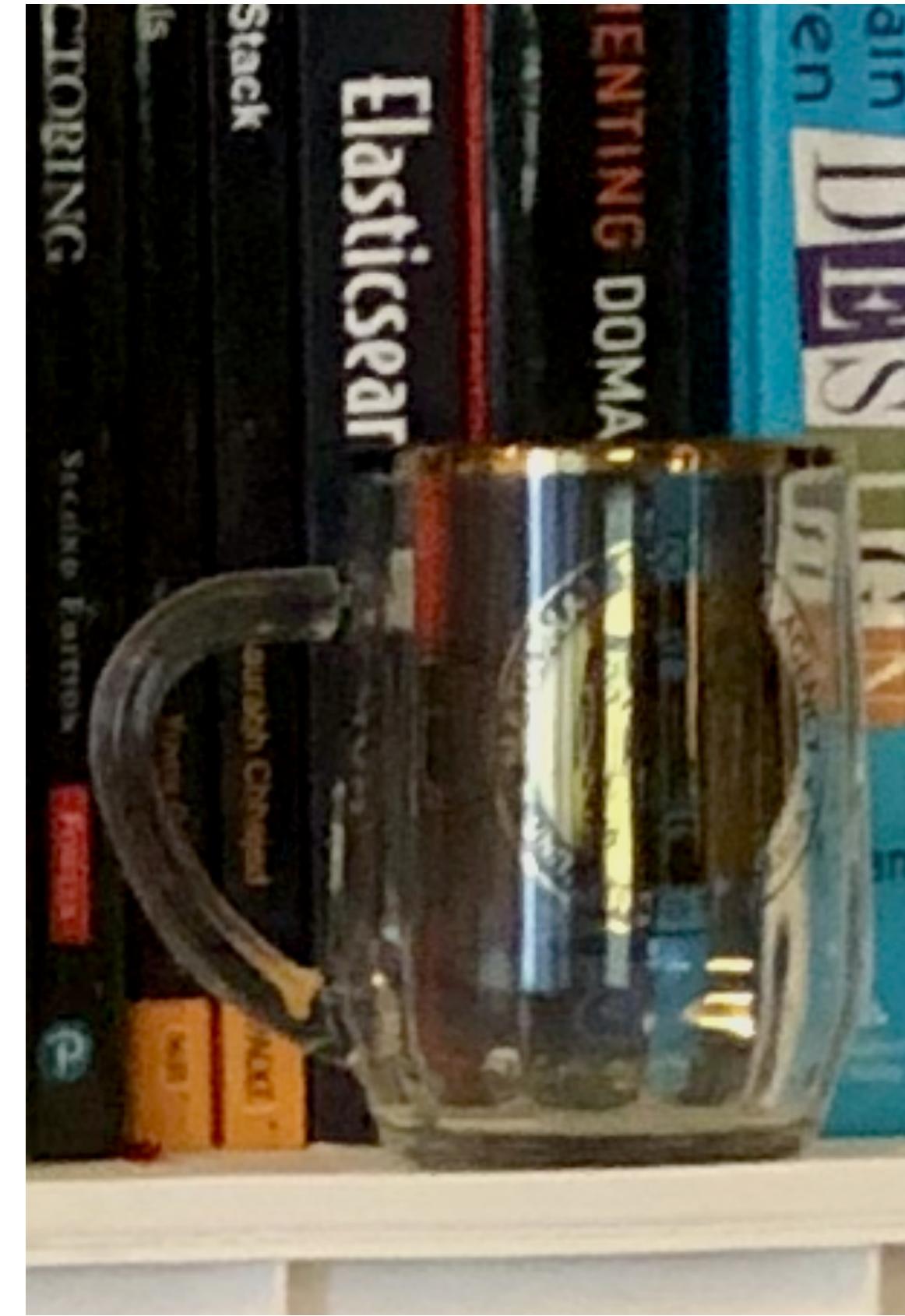
# *Here Be Dragons: Finding the Joy in Software Development* coming Sum

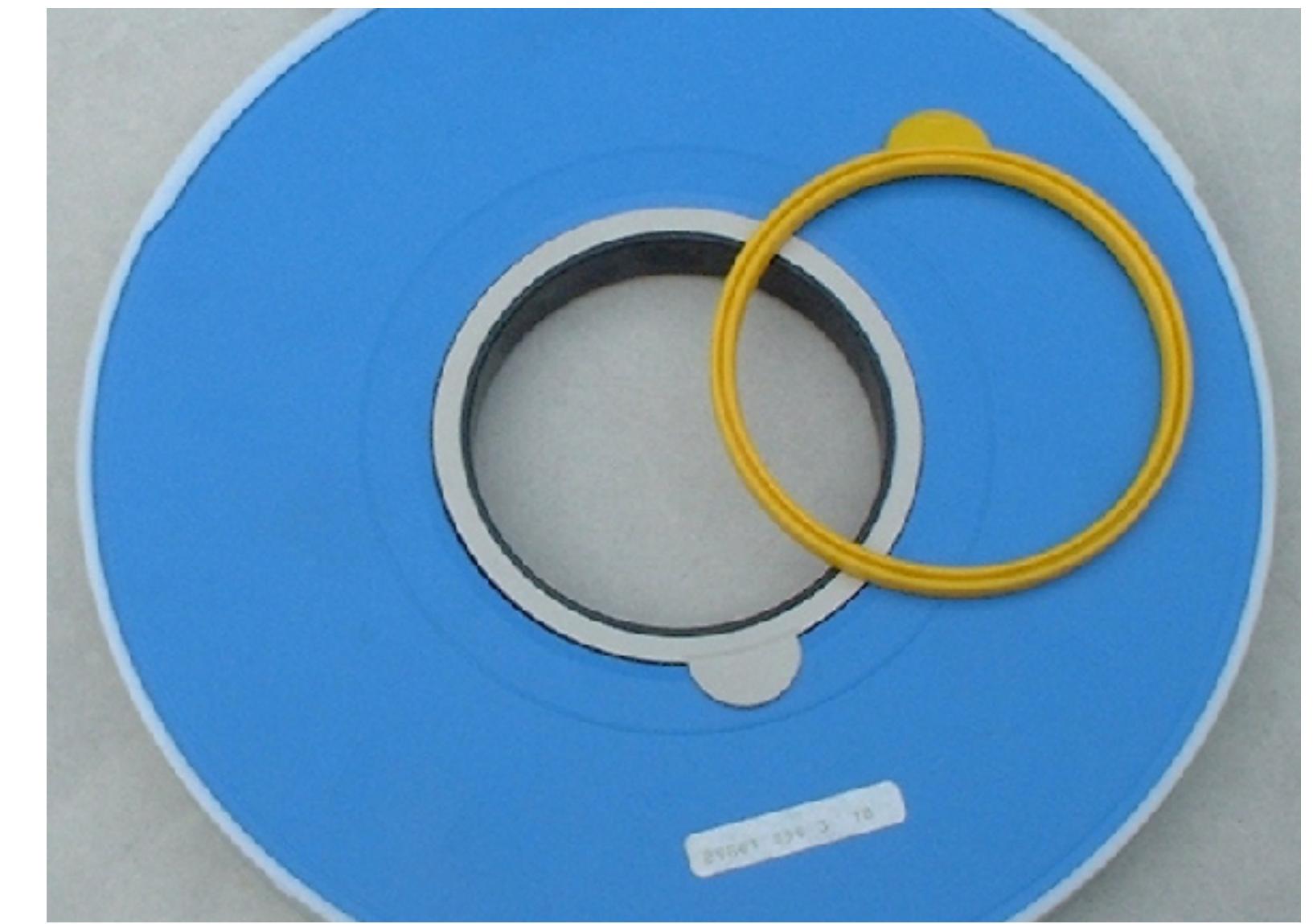


# Let's Have Fun With the Dragons!

## My NSA “Mug Shot”

The safe-arrival call





*Here Be Dragons: Finding the Joy in Software Development* coming Summer 2021

 @ewbarnard

**MINNEAPOLIS** — Network Systems Corp. (NSC) this week will announce a series of enhancements to its Remote Device System (RDS) channel extender line that will enable the units to support remote mainframe storage devices as if they were locally attached to a host.

The company also plans to unveil a new line of channel extenders to support links to tape drives.

Mainframe channel extension for direct-access storage devices (DASD) will enable users to augment disaster recovery strategies by deploying IBM or third-party DASD or tape drives at remote locations and using them to back up data as it is written to local drives.

the remote DASD unit a full set of command sequences, instead of forcing the unit to process one sequence and echo back to the host for additional instructions, thereby creating delays.

"We behave as if we were the true mainframe channel," Corneille said.

DASD support on the RDS lines is being offered as a free upgrade to existing users, while base prices for the RDS models will remain unchanged for new customers.

Corneille also said it added a multihop support feature to the RDS line that enables the channel extenders to route data over sev-

**2** The software emulates an IBM channel control program over the wide-area connection, giving

**3**

channel speeds. The RDS Printer Support Application is available  
*(continued on page 29)*

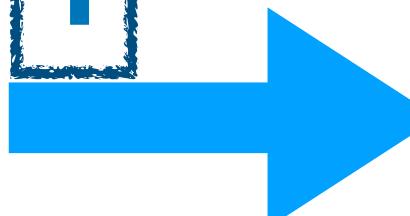
---

“**B**y the very nature of a disaster recovery plan, you should want your critical data volumes accessible at a remote location.”

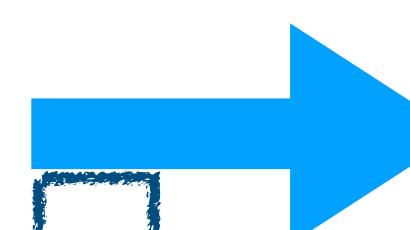
▲▲▲

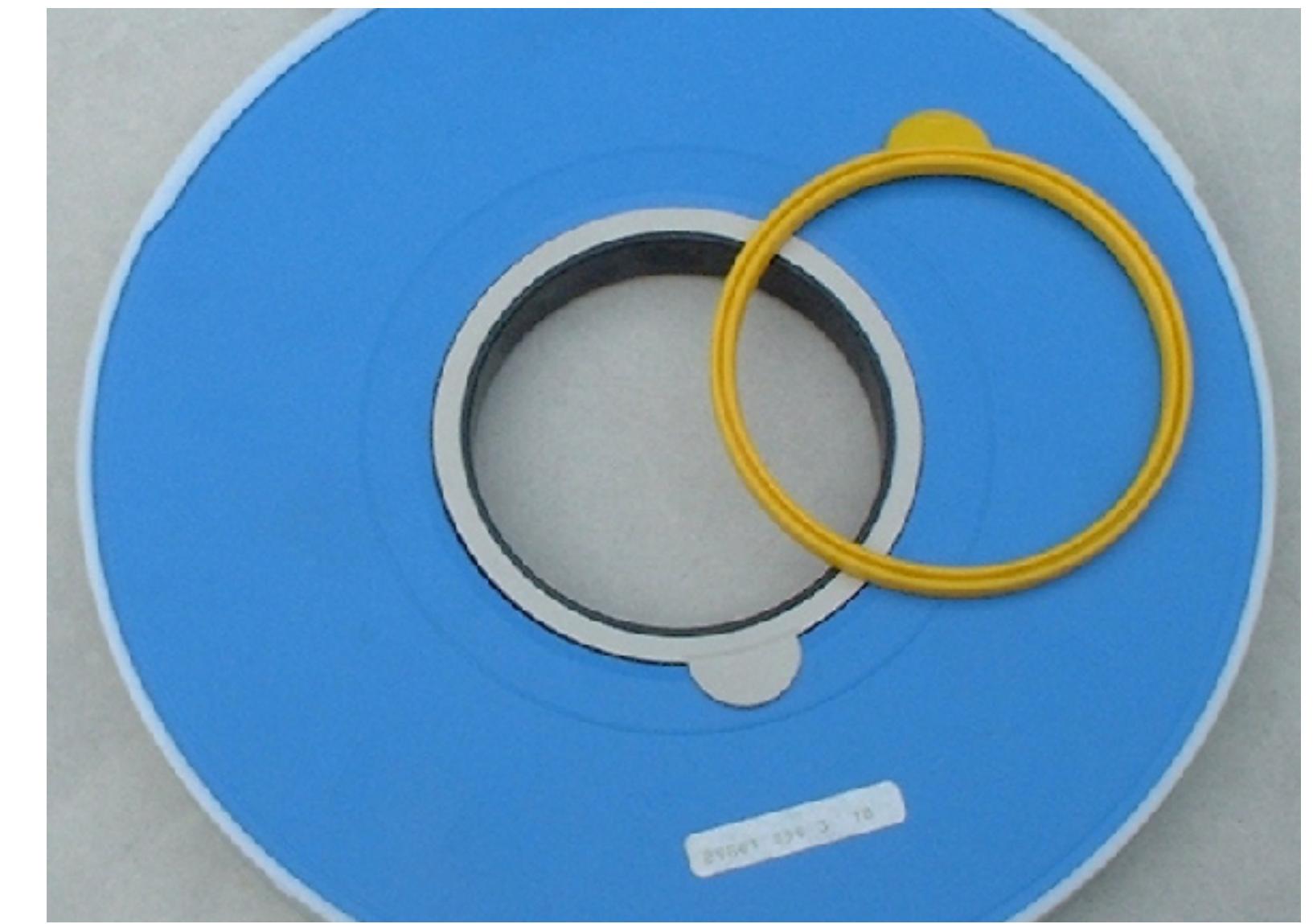
---

**1**



**2**





*Here Be Dragons: Finding the Joy in Software Development* coming Summer 2021

 @ewbarnard

# COMMANDS AND INSTRUCTIONS

## COMMANDS

Commands executed by this subsystem fall into one of the following three categories:

1. Burst Commands
2. Motion Control Commands
3. Non-Motion Control Commands

The table on this page and the one on 40-008 list the subsystem commands and command codes.

Commands not listed will set COMMAND REJECT.

**Programming Note:** The 3803/3420 subsystem has no interlocking to prevent improper sequencing of write- and read-type operations that may result in writing extraneous bits or leaving partial blocks on tape. Avoiding these improper sequences is a program responsibility.

Avoid the following two basic sequences:

1. A write-type operation after a forward read-type operation **except:**
  - a. When the block or Tape Mark (TM) read is known to be followed by a TM. A tape mark is a special block used to separate files.
  - b. When the block or TM read is known to have been followed by erase record gap (ERG) or is known to have been the last block written before a backward operation.

For example: **R R W\*** avoid.

**W B R W\*** allowed.

2. A read forward-type operation following write-type

\* indicates the logical record on which problems may occur.

Because it may be difficult or impossible to ensure the above safe situations, a write after read forward sequence should be used only in applications where a tape format and command sequence exists.

Following a backspace. Assume the tape is formatted with labels where \* is used to

\* EOF \* HDR \* DATA SET \* EOF \*\*

Page 8 of 162

This sequence involves the following steps. After processing the next to last data set header (HDR), read forward to verify the header (HDR) label of the last data set, backspace, write a new HDR, and rewrite the data set. If a new data set is being added, the read forward verifies the second consecutive TM, and thus, the true end of a data set on this tape. A backspace, write new HDR, etc., completes the sequence.

Burst Commands	Command Byte 0 1 2 3 4 5 6 7 Hex
Write	0 0 0 0 0 0 1 01
Read Forward	0 0 0 0 0 0 1 02
Read Backward	0 0 0 0 1 1 0 0 0C
Sense	0 0 0 0 0 1 0 0 04
Sense Reserve	1 1 1 1 0 1 0 0 F4
Sense Release	1 1 0 1 0 1 0 0 D4
Request Track-In-Error	0 0 0 1 1 0 1 1 1B
Loop Write-To-Read	1 0 0 0 1 0 1 1 8B
Set Diagnose	0 1 0 0 1 0 1 1 4B

Motion Control Commands	Command Byte 0 1 2 3 4 5 6 7 Hex
-------------------------	-------------------------------------

Command Byte									
Non-Motion Control Commands	0	1	2	3	4	5	6	7	Hex
No-Operation	0	0	0	0	0	0	1	1	03
Diagnostic Mode Set	0	0	0	0	1	0	1	1	08
Mode Set 1									See 40-008
Mode Set 2									See 40-008

## BURST COMMANDS

Burst commands transfer data across the channel/tape control interface. Channel End and Device End are signaled when the operation is complete (ending status).

The burst commands are:

Write  
 Read Forward  
 Read Backward  
 Sense  
 Sense Reserve  
 Sense Release  
 Request Track-In-Error  
 Loop Write-To-Read (maintenance aid\*)  
 Set Diagnose (maintenance aid\*)

\* Diagnostic programs issue maintenance aid commands via start I/Os (SIOs) that are op-codes in the Channel Command Word (CCW).

## WRITE

Write records data on tape as it moves forward and creates an interblock gap (IBG) at the end of each block. The tape control checks the parity of each data byte received from the I/O interface.

# Key Skill: Finding Patterns in the Noise

Let's examine **142857**

- Multiply by 2 – 285714 (left circular shift of two digits)  $14\ 2857 \rightarrow 2857\ 14$
- Multiply by 3 – 428571 (left circular shift one digit, first digit moved to back)
- Multiply by 4 – 571428 (left circular shift four digits)
- Multiply by 5 – 714285 (left circular shift five digits)
- Multiply by 6 – Guess what? See the pattern? Does it hold? What's left?

# Key Skill: Finding Patterns in the Noise

Let's examine **142857**

- Multiply by 2 – 285714 (left circular shift of two digits)
- Multiply by 3 – 428571 (left circular shift one digit, first digit moved to back)
- Multiply by 4 – 571428 (left circular shift four digits)
- Multiply by 5 – 714285 (left circular shift five digits)
- Multiply by 6 – 857142 (left circular shift three digits)
- Multiply by 7 – 999999 (game over!)

# Key Skill: Finding Patterns in the Noise

Let's examine **142857**

- The pattern comes from decimal fractions
  - $1/2 = .5$ ,  $1/3 = .333\dots$ ,  $1/4 = .25$ ,  $1/5 = .2$ ,  $1/6 = .18666\dots$
  - $1/7 = .142857\ 142857\ 142857\dots$  (there's our number)
  - $2/7 = .285714\ 285714\dots$
  - Is there a point to this weirdness?

# Key Skill: Finding Patterns in the Noise

I solved the NSA problem by spotting a bit of weirdness

- In the noise of continuous data transfers, we spotted a sequence that should not be there
- We noted the *wrong* pattern was in the noise
- This is an extremely useful skill — if you can develop it
- I have a theory about developing that skill...

Achieve weirdness,  
and in the process you get better at  
*spotting* the weirdness,  
and you'll be finding the problems (bugs)  
that nobody else can.

But the process starts by gaining the skills.

# The Process of Mastering a Skill

Azeria, <https://azeria-labs.com/the-process-of-mastering-a-skill/>

- Prodigy or expert? Have you ever wondered what separates experts from everyone else?
- I don't mean the self-proclaimed "experts" who value status over learning
- I mean the "geniuses" of our industry who know their field extraordinarily well and have spent decades honing their skills.
- How do these successful researchers master their field?
- Is it some natural talent they were born with, or do they have a magic trick that only a few know about?

# The Process of Mastering a Skill

Azeria, <https://azeria-labs.com/the-process-of-mastering-a-skill/>

- When performance psychologists studied what distinguishes experts across several different fields, they kept coming across the same answer in each field.
- “Deliberate practice.”
- *Reading her full article at the link above will be time well spent!*

# Dragon Wrangling

## Why Bother?

- Once you're the expert, have fun with it!
- Expect to put in the work to *become* the expert
- Expect to always be learning – our industry evolves (remember *Dune*)
- Gain that “been there done that” confidence as you put in the work to continue learning over and over again

I laid on the “expert” business a  
bit thick for a reason...

# Imposter Syndrome

Common in our industry — including myself



- Pattern of behavior where people doubt their accomplishments
- Persistent, often internalized fear of being exposed as a fraud
- Despite having adequate internal evidence of accomplishments, people with imposter syndrome remain convinced they don't deserve the success they have

# Imposter Syndrome

## My own experience



- Roughly age 10 through age 30
- Imposter syndrome told me that:
  - Since I'd been studying computers longer than anyone else around me, I had to be better than *everyone* else around me — or I was a failure
  - You'll laugh because you know it doesn't work that way!
  - My solution: pay it forward — I could help someone else even if they knew more than me, and I learned more in the process
  - Paying it forward made it “okay”

# Imposter Syndrome

## Software Training



- Due to a car accident, I suddenly had to step in as Operating System Internals instructor, for the first time – and I wasn't ready
- I found out Saturday and the 2-week class started Monday
- No other options!
- I stepped in, as an imposter, said I was the Instructor – for two weeks!

# Imposter Syndrome

## The Last Day



- On the final Friday, I explained the standard procedure — I would step out while the manager came in to solicit course feedback and paper survey
- I explained this was my first time teaching the course, it happened unexpectedly, and I'd greatly appreciate any suggestions at all for better teaching the course next time around
- I unwittingly “stacked the deck” and got maximum survey scores from every student (very unusual)

# Imposter Syndrome

## Fade away – but never completely



- That moment, for me, is when Imposter Syndrome began to lose its grip
  - It was okay that the other instructors were more experienced than I
  - Less experience didn't make me an imposter, though I felt like it
  - I was teaching how things worked even those who *wrote* that software knew more about it than me
  - My students knew more than me, yet I represented my company to them



# Pay It Forward

## How I broke my imposter syndrome

- 20 years to get past, and 30 years later obviously still with me
- That's why I *must* pay it forward:
  - I don't have to worry about whether I am the *best* person to pay it forward
  - I just do it whether I'm the best person or not
- When you've “been there done that” and gained the experience, you can provide the shoulder for others to stand on — even if your brain thinks you're an imposter

# Wizard Thinking

## I lived amongst the Wizards

- When you realize that what you're doing is ***important***, specific details like actually writing the software will take care of themselves
- This attitude might sound dismissive — but it's not
- This is “wizard thinking” based on an expectation of lifelong learning
- “Wizard thinking” expresses love of the craft



***Here Be Dragons: Finding the Joy in Software Development*** coming Summer 2021

 @ewbarnard

# On the Shoulders of Giants

## Build Trust, Earn Trust

- Many newbies want all the answers, NOW. Don't be that person.
- Learn to learn. Learn how you best learn — so that you can put in the work, gain the skills.
- Expect to be continuously learning — and that it's up to you.
- You be “the shoulder” in small things, and when the big things come, you'll know what to do.
- Find ways to help others exceed your own mastery.



Some take off like  
a rocket, never to  
return

# But We Need A Gatekeeper!

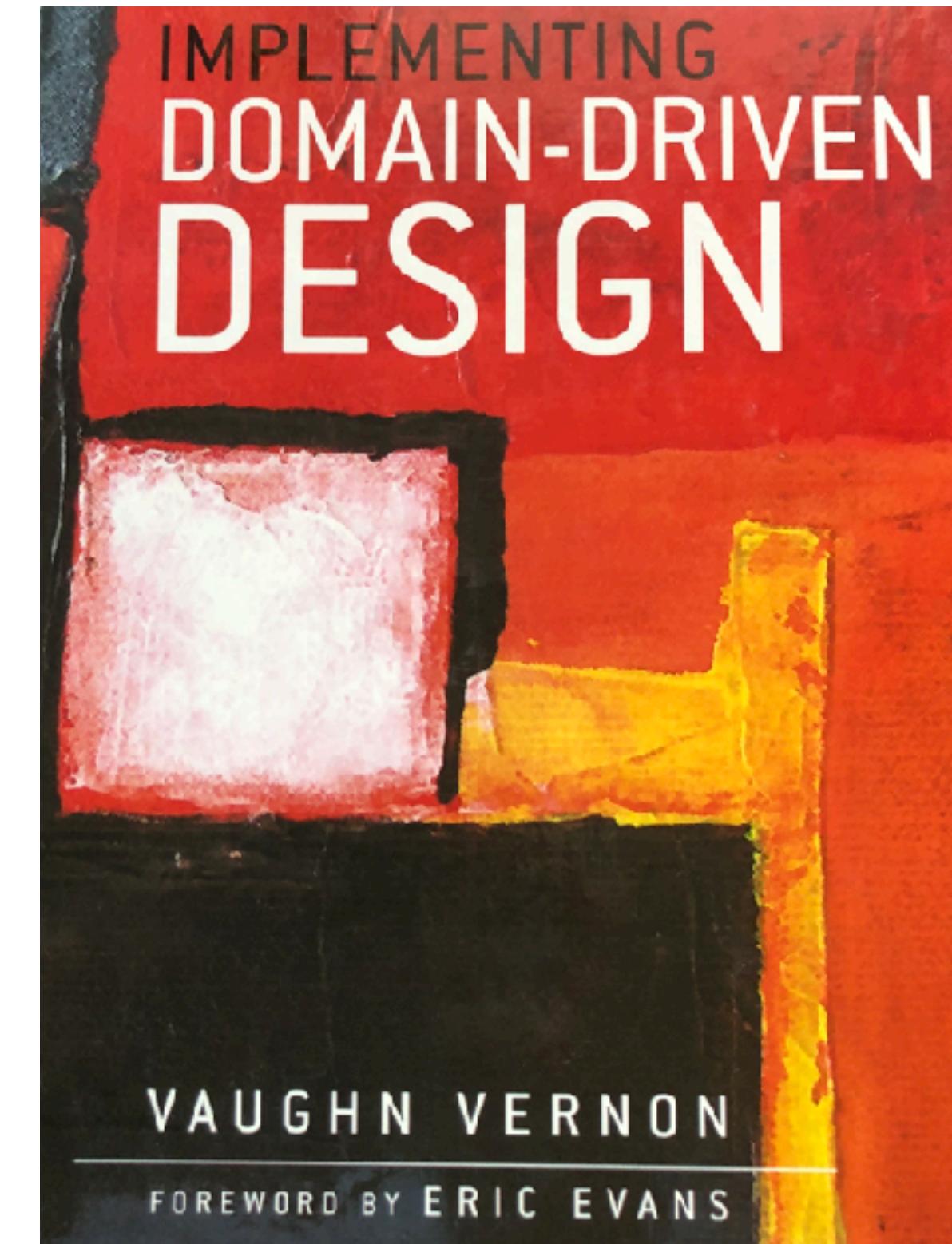
Do you really? Think about why.

- It's on you to *teach* the skills, how to understand the dangers
- Do you want to *always and forever* be the one on call?
- But we have a Product Backlog! We're in the middle of a sprint!



Why, yes. Yes, you are. About that...

It seems to me that Scrum and other agile techniques are being used as substitutes for careful modeling, where a product backlog is thrust at developers as if it serves as a set of designs.

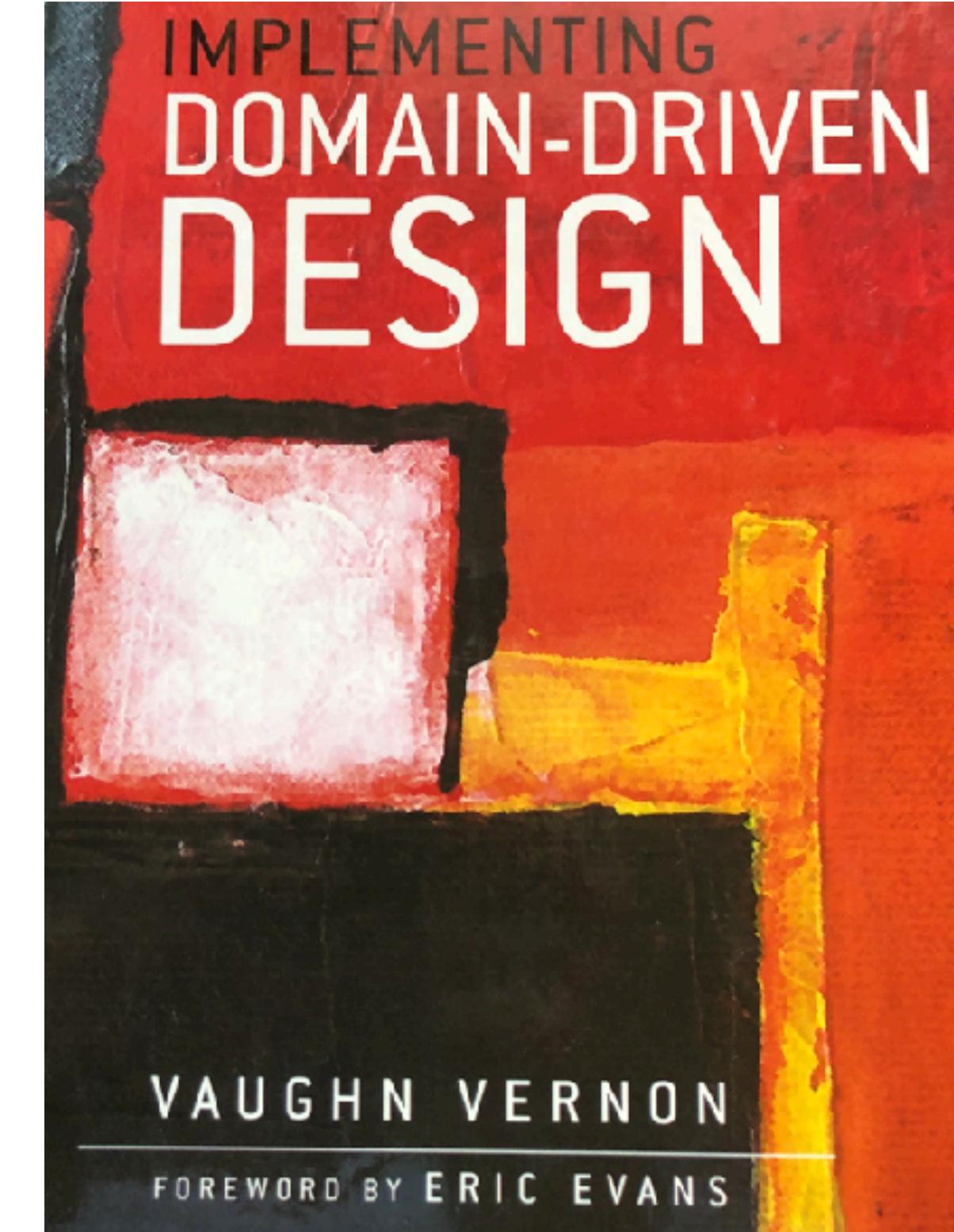


The “Red Book”

Vaughn Vernon, from the Preface, pages xxvii-xxvii

No matter how many project and product managers would like to keep you marching on a relentless path of continuous delivery, Scrum was not meant only as a means to keep Gantt chart enthusiasts happy.

I consider this a big problem...



The “Red Book”

Vaughn Vernon, from the Preface, pages xxvii-xxvii

# Not Going to Happen!

So be it. You can still guide YOUR career.

- Lay out your path to always be learning
- Gain Azeria's skill of *deliberate practice*
- Every experience carries its lesson
- Think like a wizard and *HAVE FUN!*



**“If I have seen further than others, it is by standing on the shoulders of giants.”**

— Future You, just chillin amongst the dragons



*This slide deck and book details:*  
**<https://github.com/ewbarnard/dragons>**

**Here Be Dragons: Finding the Joy in Software Development** coming Summer 2021

