



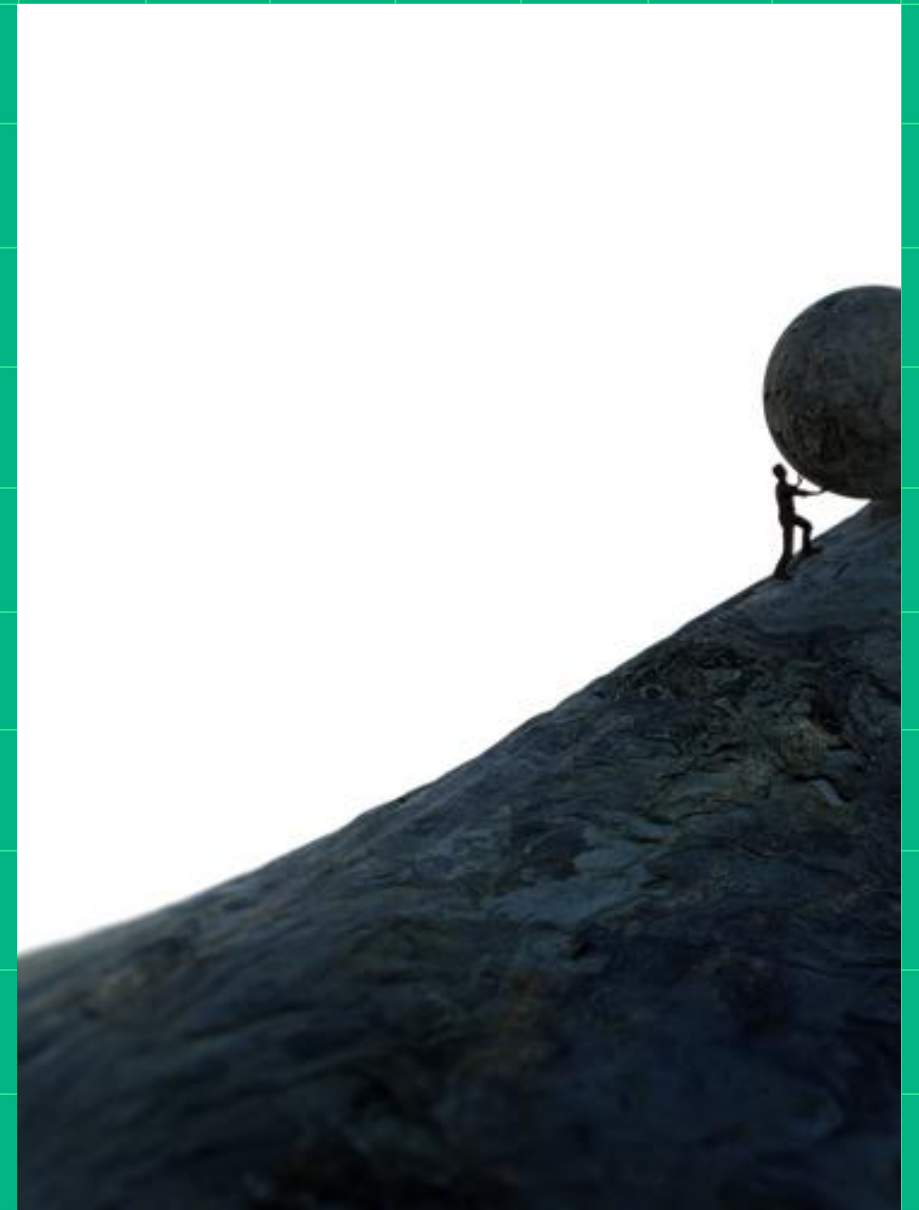
# CONNECT

# DFD – Design For Reality

Norm!

Chief Technical Support Engineer

The Tactics and Principles  
of “Design for Debug”



# The Promise

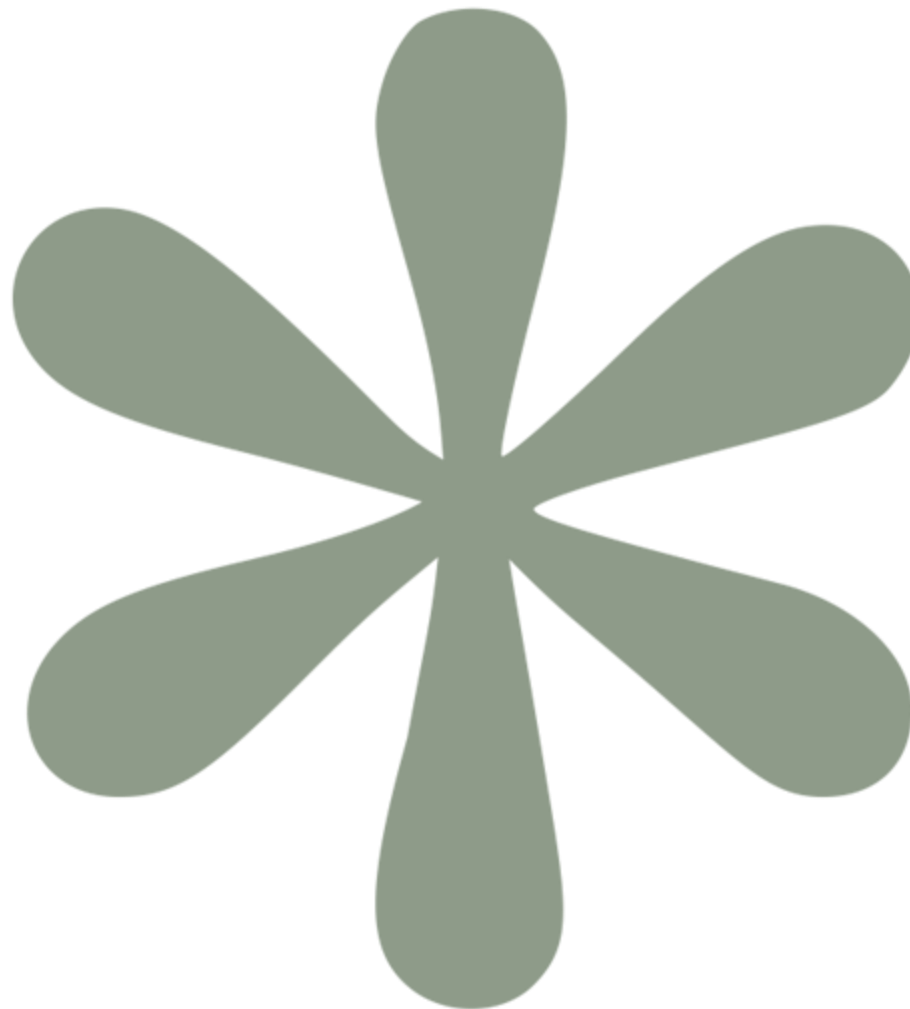
You will\* walk away with:

Newfound **enthusiasm** and skills  
for **proactively** augmenting your development;  
with a mindful **compassion** towards  
the **person** who needs to debug your code  
(who might be **you**)

# “Debugging is a Waste of Time”



# Clarification



# Design For Debug Definition



“It is inevitable”  
~Agent Smith

Explicit **design** choices,

which intentionally enable or **accelerate**,

the inevitable process of **debugging**.

Design for  
Debug

This is fine



Design for  
Reality |



# Why do we waste time?



- Program to the 'happy path'
- Accelerated development cycle



- We're not perfect
- If it runs, it's good

"It's all bloody well and good..."

till something stops working"

~DCorney





~~What do you want to do?~~

What do you want to give someone else to do?



# The Time Suck

Why does identifying the cause take time?  
Why does it take longer for some than others?

“Clues Watson, Clues...” ~Sherlock Holmes

- Instinct
- Sniff test
- Observations
- "What happens if I wiggle this"
- Historical Knowledge
- Systematic Understanding



## Still Not Sold?

Even with new techniques in development and architectures we still debug a lot

If you're spending ample time trying to track down a problem...you can improve your system

(and likely should)

Not facilitating good debugging is a **technical debt** issue

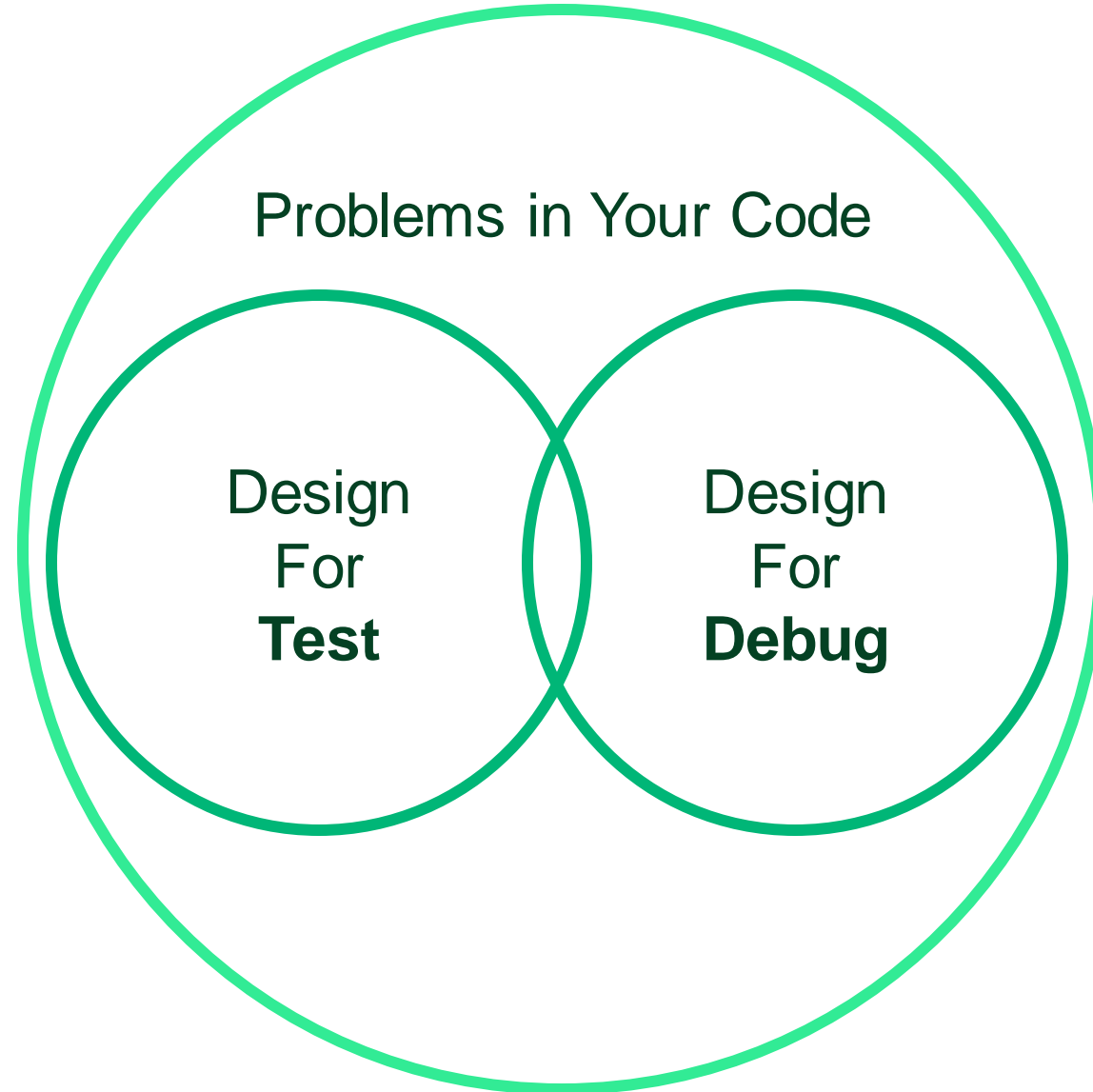
Bugs = Technical Debt

Time spent '*finding the problem*' = Paying off interest

Fixing the issue = Paying off Principal

Bad Debug Design = High Interest Rates

# What about ‘Test Driven Design’/ ‘Design for Test’?





# Example Comparing Difficulty (none vs good) design

Continuous Measurement and Logging

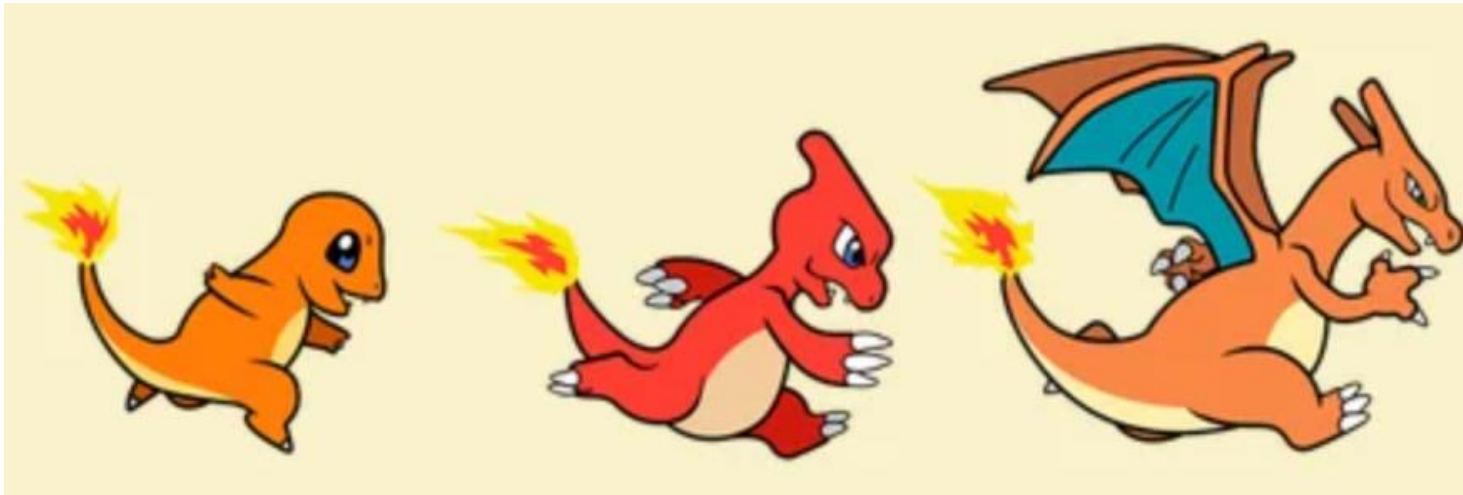
DQMH

So now what?

# Techniques for Everyone!

L1 User  
I create systems  
for me

L3 Architect  
I make frameworks  
for others



L2 Developer  
I build on the shoulders  
of others  
to build big systems



So now what?

# What to provide the future?

Because they might know where you live...

# Insight

# Accelerators



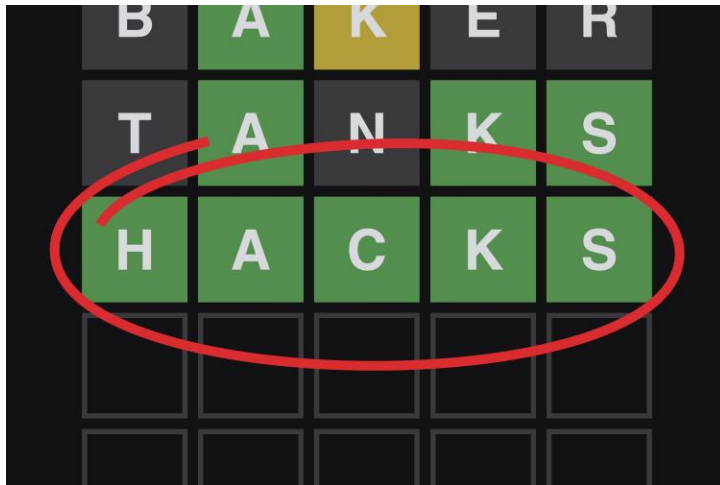


# Insight & Accelerators

Simplified Example

## Insight

Information  
which leads to identification



## Accelerators

Tools  
which lead to access



# Logging :: (Insight)

## General Progress and System Status

- L1
  - Custom LabVIEW probes
  - UDP Broadcast
  - Log Files / IO Trace
- L2
  - DQMHA API Tester
  - HSE Logging
  - AF Debug Conditional Flag - DETT
- L3
  - Configurable Levels of Information
  - Retaining last XX number of system states and actions
  - Bolt-on remote debugging



# Code Stack Transporters :: (Accelerator)

No...having 100 windows open to debug, is not ok

- L1
  - Corkboard / VI Server Shortcuts
  - Custom Projects
  - #Bookmarks (Automatic Error Handling)
- L2
  - DQMHAPI Tester
  - Panther Dashboard - PantherLab
  - AF BowZer - Zayah
- L3
  - Module / Re-entrant VI Registration
  - Custom Framework Browsers
  - Emulation of Familiar Tools - Memory Allocation – VI Analyzer

# Data Dumps :: (Insight)

The only bad information, is no information

- L1
  - Errors that provide actionable insight
  - Logging Probes
  - Unceremonious Mother-Cluster to Binary File
- L2
  - AF Actor Extension
  - HSE (maybe)
  - Kraken
- L3
  - Configuration, Results AND Intermediate Data
  - Diffing tools
  - On-Demand / Periodic / On-Error Klaxon

# Issue Recreation :: (Accelerator)

Like a time-machine into your awkward teenage years

- L1
  - Message playback
  - Recorded data as simulation data
  - Functional containment
- L2
  - DQMHAPI Tester
  - Class Extension Injection
- L3
  - Ingest log / dump -> output macro-script / debug VI
  - Debug harness for modules / algorithms



So Now What?

# What to provide the future?

Because they might know where you live...

# Insight

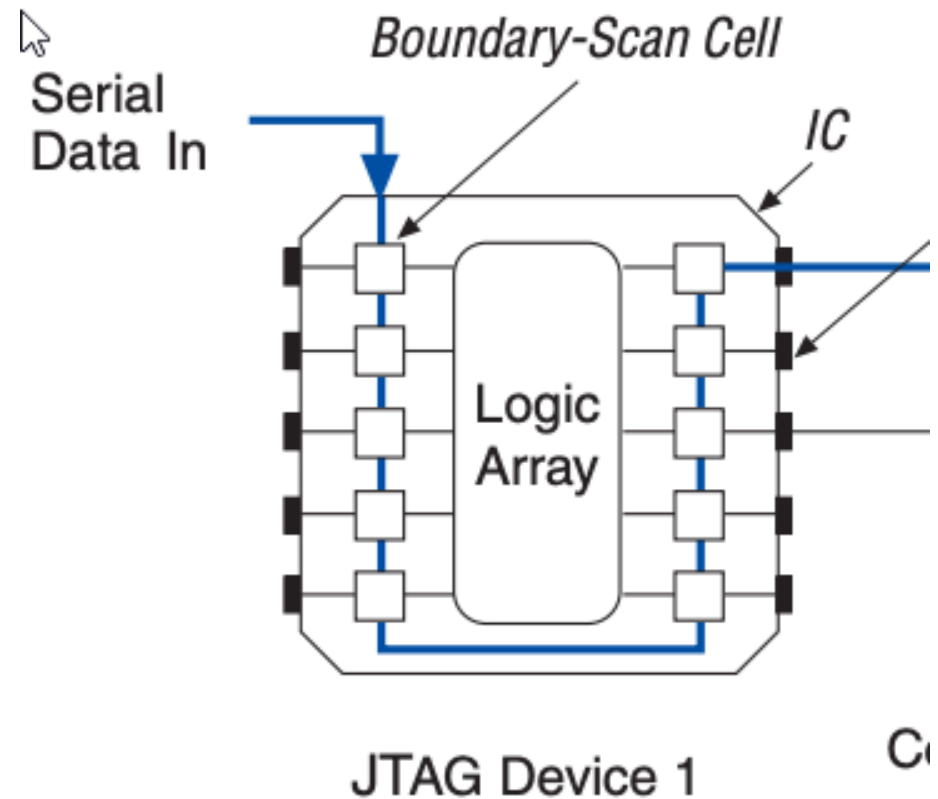
# Accelerators



# Wait...doesn't this exist already?

Design for Debug – a voice crying out in the desert

JTAG Boundary Scan – Joint Test Action Group



**27,300,000**  
**Results Later**



# Wait...doesn't this exist already?

Design for Debug – a voice crying out in the desert

## 2 Hits from the LV Community

### 1 general SW blog

### Remainder is TDD



~~What do you want to do?~~

What do you want to give someone else to do?



# The Promise

You will\* walk away with:

Newfound **enthusiasm** and skills  
for **proactively** augmenting your development;  
with a mindful **compassion** towards  
the **person** who needs to debug your code  
(who might be **you**)

Acts 20:35



# Q & A & Ideas!

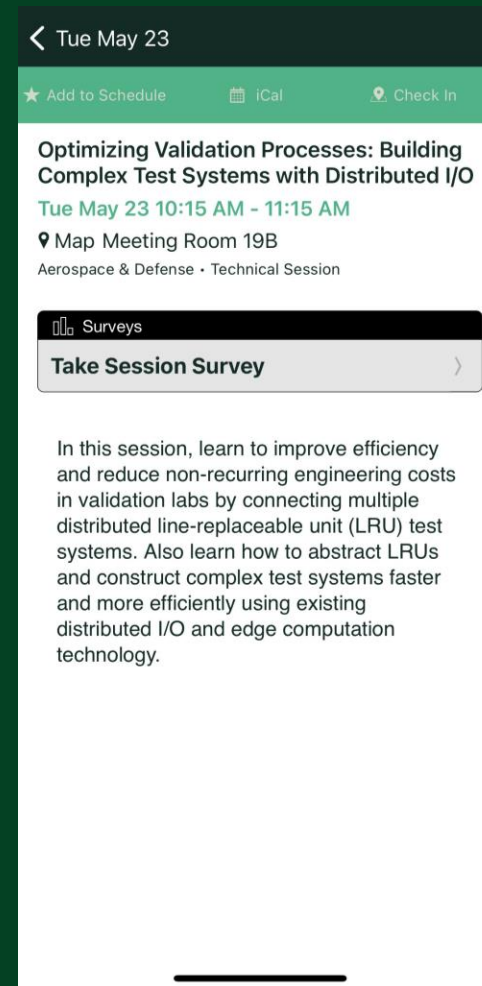
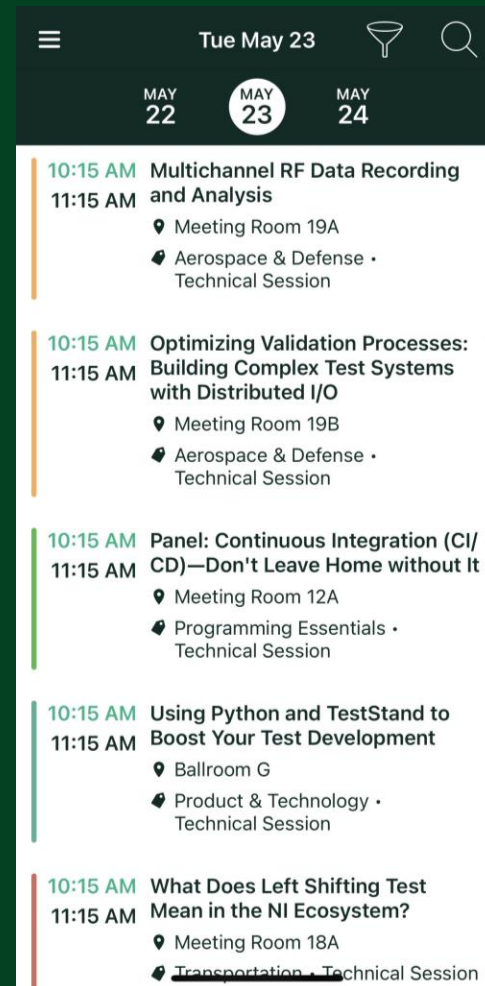
<Insert Thoughts Here>

- x



# Give us your feedback! Quick 2 Question Survey

In the mobile app,  
click into the  
session you would  
like to provide  
feedback for



Click “Take the  
Session Survey”