

konrad
... technologies ...

WE KNOW **HOW TO TEST!**

LabVIEW & TestStand Better Together

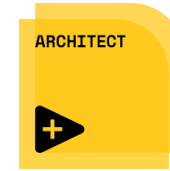
Chris Roebuck

Chief Technology Officer
Konrad Technologies UK Ltd

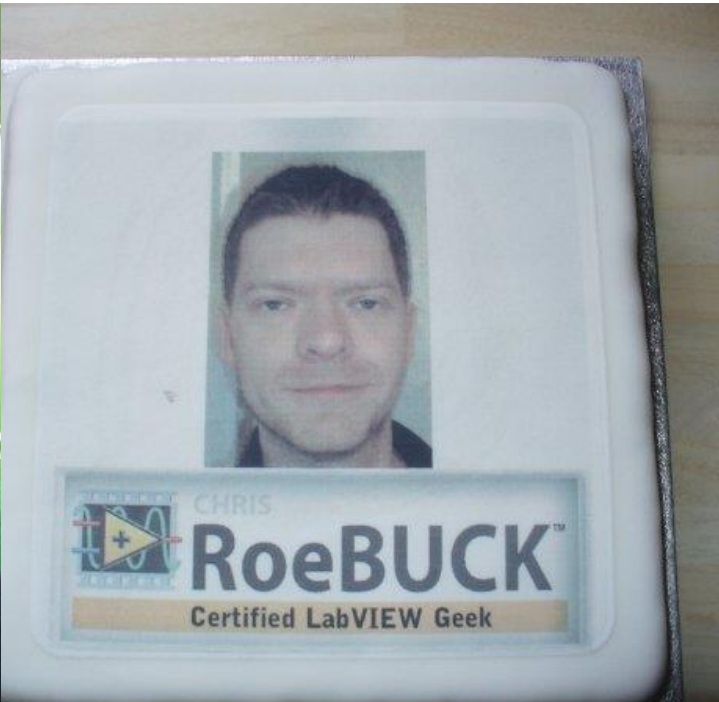


Chris Roebuck

Chief Technology Officer



BSc Physics
LabVIEW / TestStand evangelist



LabVIEW & TestStand Better Together



*The complementary roles of LabVIEW and TestStand
throughout the test lifecycle*

But I don't need TestStand, right?

"I don't need user management"

"I don't need fancy test reports"

"My test limits will never change"

"I don't need multi-UUT support"

"I don't need database results storage"

"I don't need to provide anyone other than software developers the capability to develop tests"

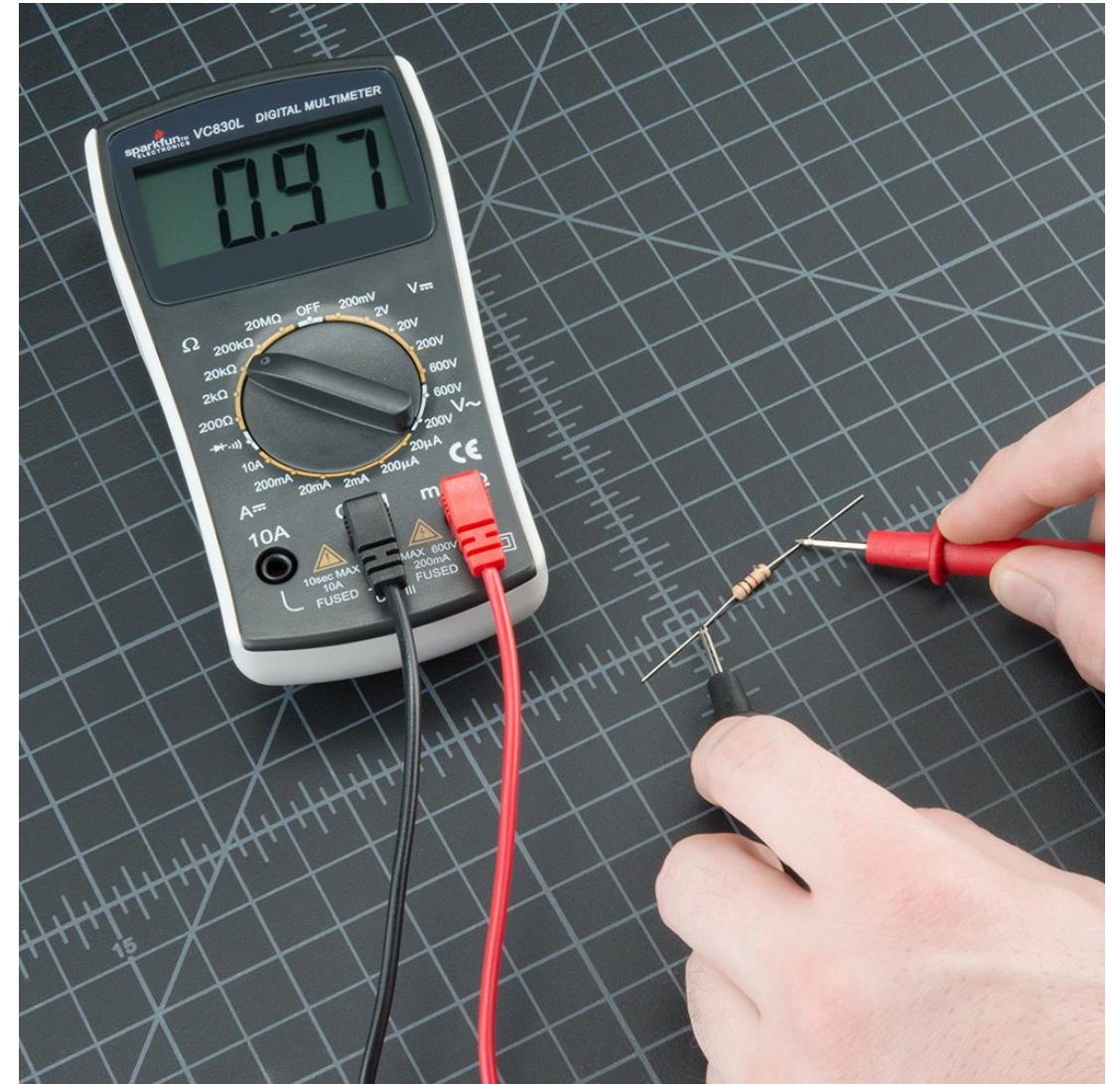
"I will never need to run operations in parallel"

"I don't want to pay for Base Deployment Engine"

A long time ago in a test lab far,
far away....

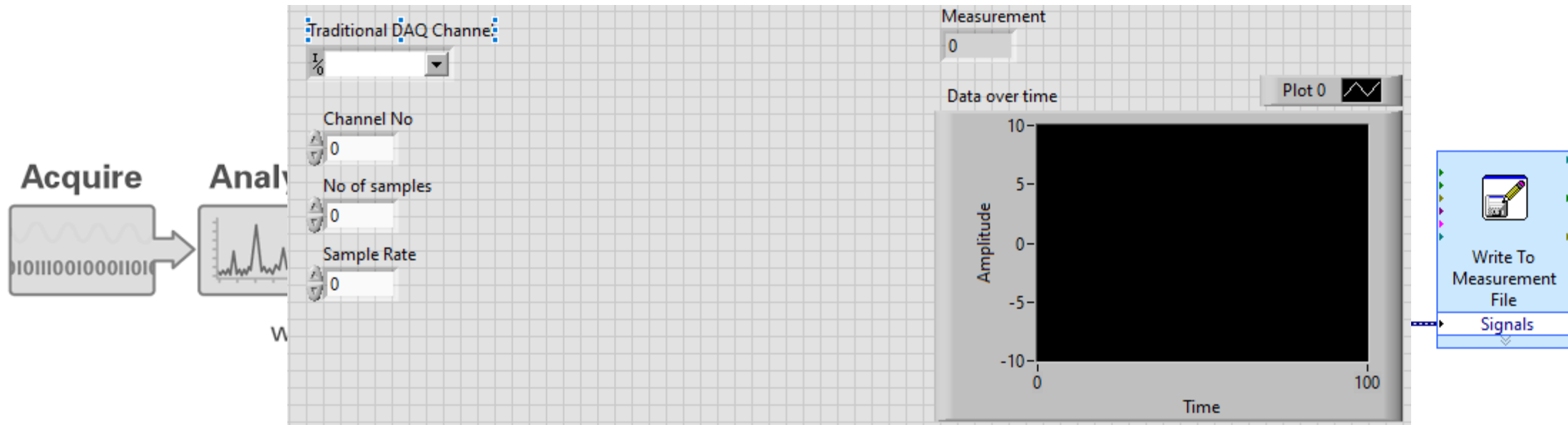
Somebody is automating a manual test

- Connecting an instrument to a PC (or even better, use PXI)
- Use LabVIEW to communicate with instrument using appropriate APIs
 - VISA
 - DAQmx



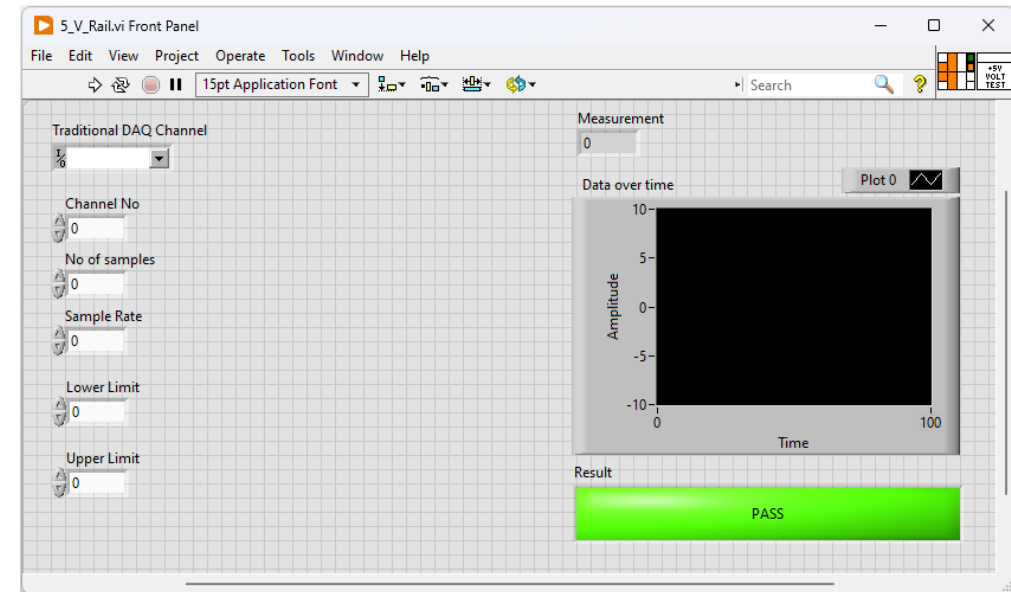
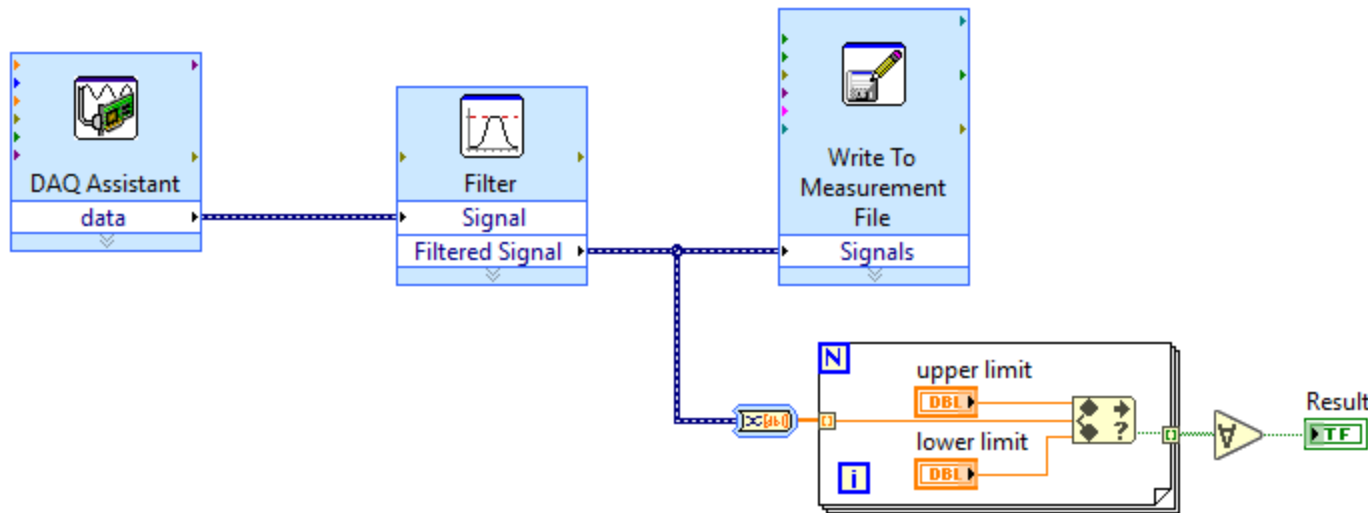
3 VI's Acquire, Analyze, Present

- The starting point for many automated test solutions is the ubiquitous 3 VI solution
- We acquire, analyze and present. Often the presentation is in the form of a graph or chart or a simple file I/O operation resulting in a measurement data file of some description



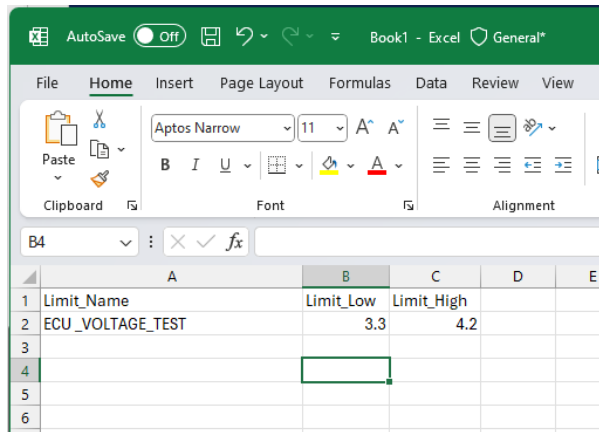
A measurement quickly becomes a test

- Pass Fail criteria applied to a measured value gives us our first test producing a Boolean result
- We may provide upper and / or lower limits
- An array of measurements may be compared to a single value and the results aggregated or compared to multiple limits such as a mask test



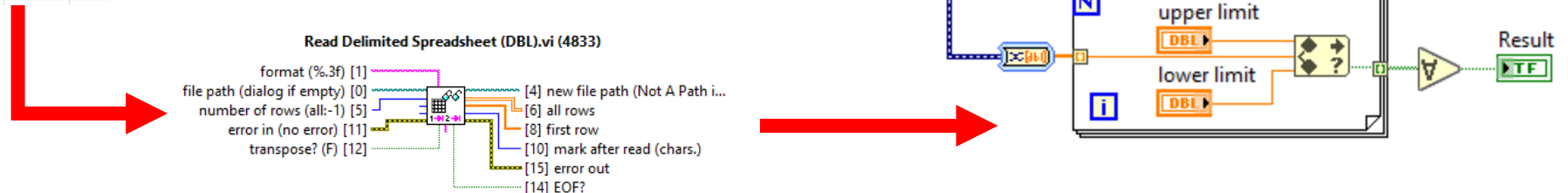
Eventually we externalise parameters?

- We realise that 'hard coded' limits or instrument / UUT configuration results in considerable code churn as products evolve or specifications change
- Parameters / Limits are often read in dynamically from CSV or Text files meaning the test configuration can change without software modification



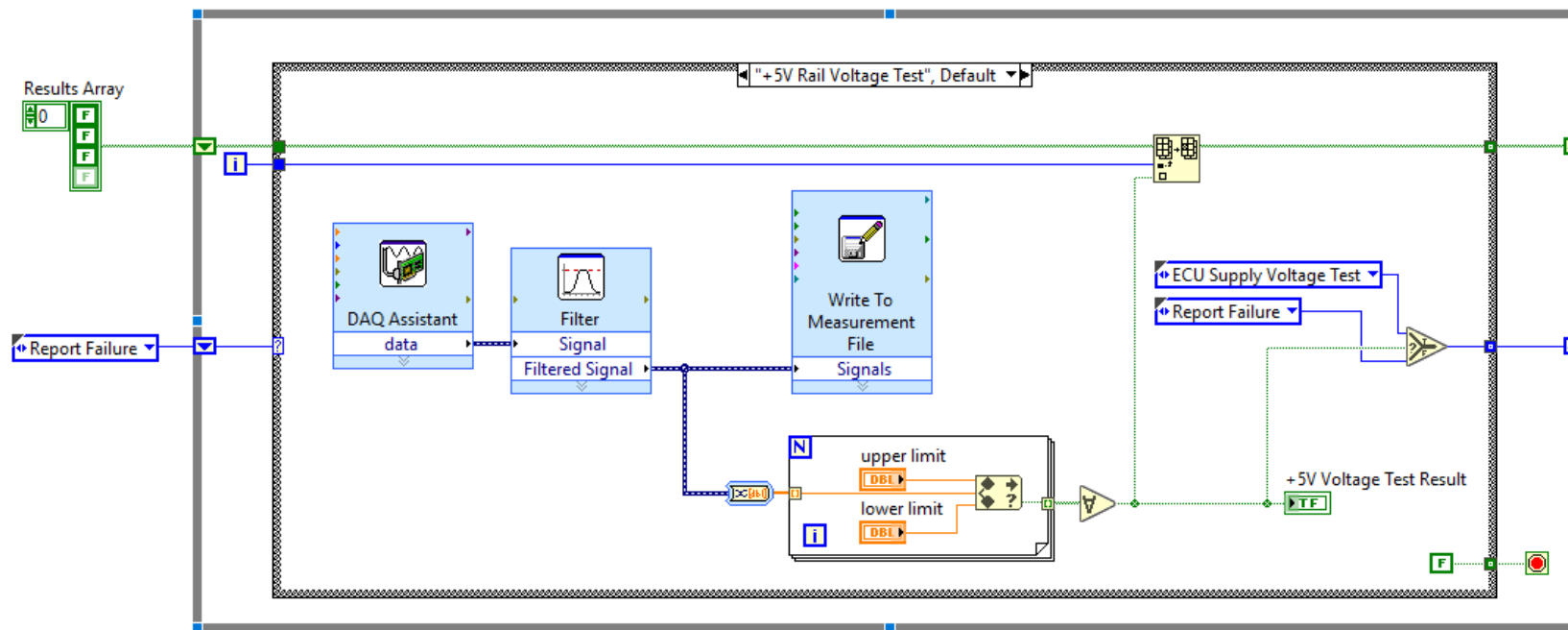
The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E
1	Limit_Name	Limit_Low	Limit_High		
2	ECU_VOLTAGE_TEST	3.3	4.2		
3					
4					
5					
6					



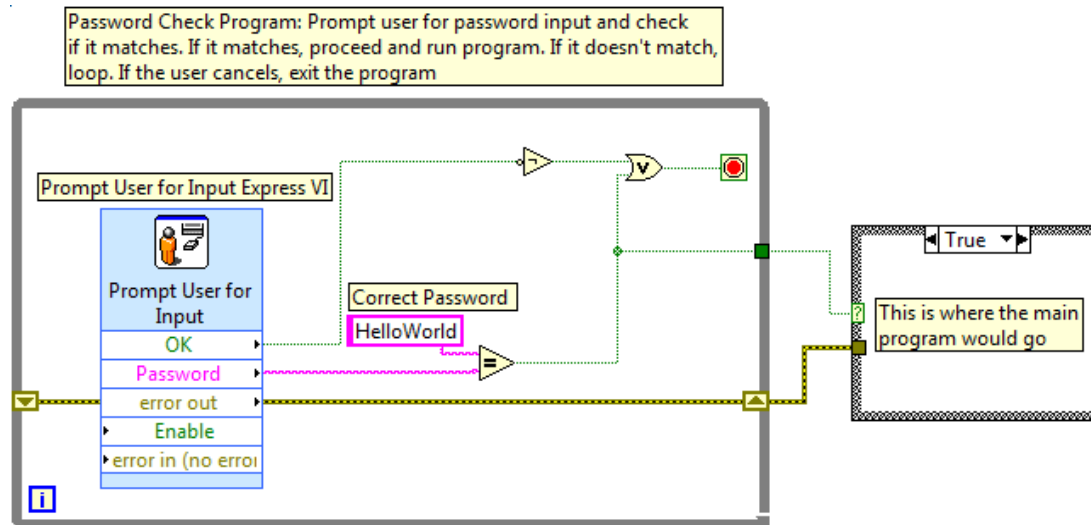
Test Sequencing

- We add multiple tests, decision making, control logic and error handling
- Results of test may be used to determine later actions
- Steps may be repeated
- A results list is maintained throughout



User Management

- We add the option for 'some' users or types of user to perform certain protected actions
- We implement our own authentication mechanism
- Maybe we authenticate the Windows user, maybe we maintain a database of users



Login

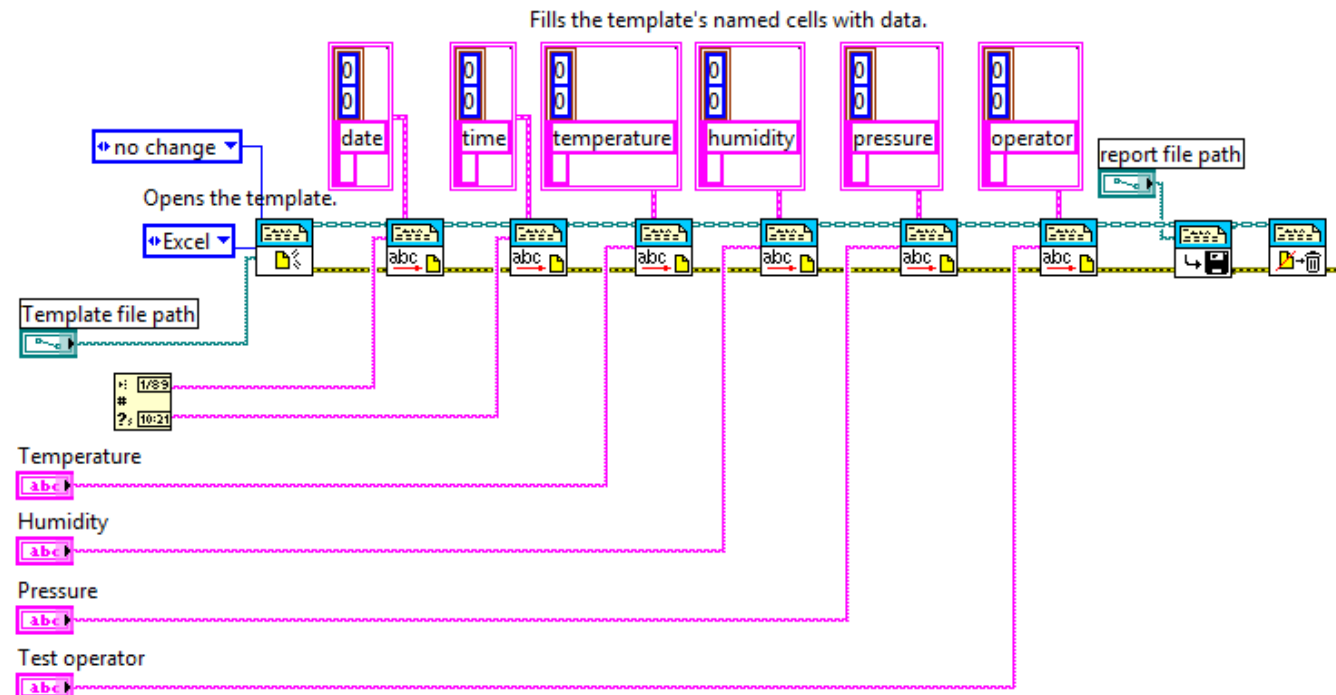
User name:

Password:

OK Cancel

Test Reporting

- Reporting needs move beyond writing to text / CSV / TDMS file
- Addition of headers, corporate logos, images and captured waveforms
- Hyperlinks to additional files and resources (calibration data or data plots)
- Multiple report formats



Becoming difficult to manage

- Adding more customers means more reports, configurations and limit files
- Increase in product variants increases the number of limits and configuration files
- Data storage requires data analysis, now you're potentially dealing with SQL
- Product variants have different test flows, decision making logic becoming more complex
- R&D want to use same solution as production but with more interactive capabilities, but how?
- Fragile?

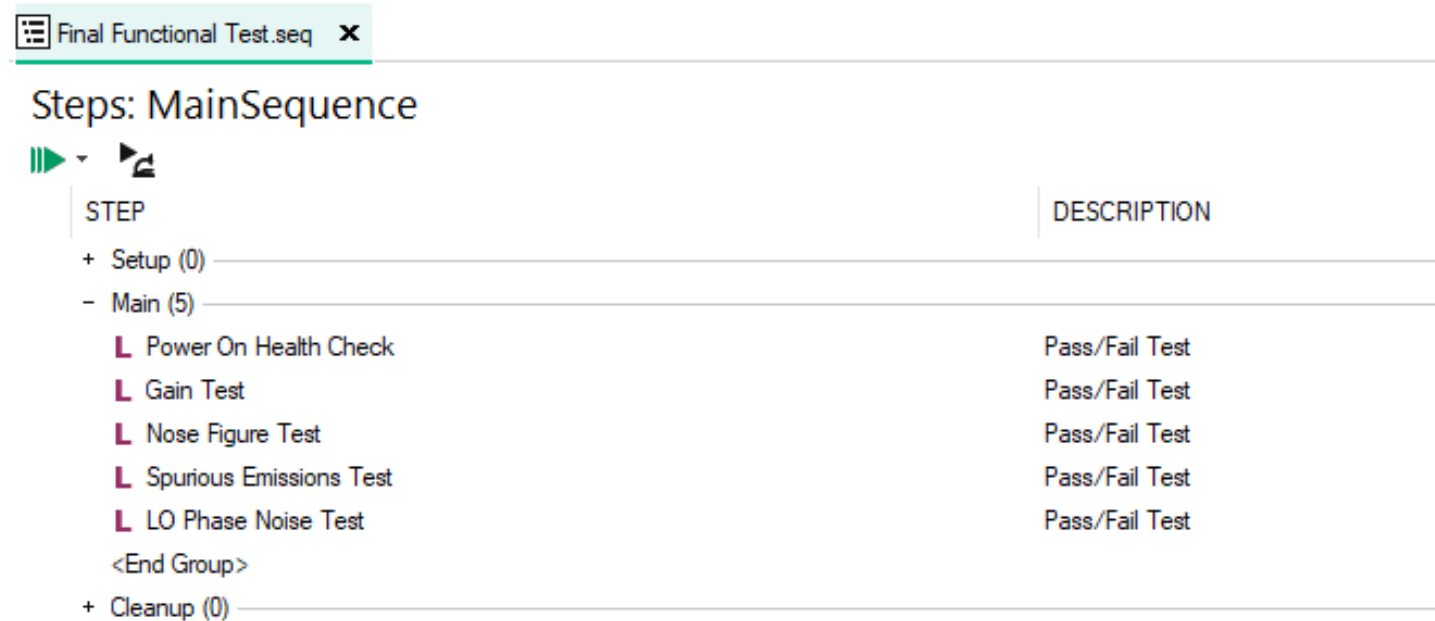
Maybe we should have selected a Test Executive before things became complex?

A long time ago in a *different* test lab far,
far away....

Customer purchases TestStand



Implements TestStand + “The VI”



STEP	DESCRIPTION
+ Setup (0)	
- Main (5)	
L Power On Health Check	Pass/Fail Test
L Gain Test	Pass/Fail Test
L Noise Figure Test	Pass/Fail Test
L Spurious Emissions Test	Pass/Fail Test
L LO Phase Noise Test	Pass/Fail Test
<End Group>	
+ Cleanup (0)	

- Single or small number of VI calls
- Complex or long running
- Minimal user feedback
- Limited interaction opportunities

WHAT IF TITANIC MISSED THE ICEBERG?



Scope Creep

- What if my customer now wants:
 - Different functional behaviors for different users
 - Different Report Formats for individual projects
 - Multiple report outputs from a single test run
 - Multi-UUT testing
 - Control of shared resources across multiple test sockets
 - Ability to interactively execute and re-run single steps in a sequence or entire tests
 - Ability to interact with instruments or UUT using soft front panels whilst carry out investigations or optimizations
 - Ability to add pre-conditions for test execution in the interests of test throughput
 - Ability to re-order tests programmatically based on anticipated yield in-order to reduce time to find first failure

These are all things we can reasonably expect a maturing product test to require eventually

TestStand uh?

*This is the stuff we end up needing
anyway*

*"I don't need user
management"*

*"I don't need fancy test
reports"*

*"My test limits will never
change"*

*"I don't need multi-UUT
support"*

*"I don't need database
results storage"*

*"I don't need to provide
anyone other than software
developers the capability to
develop tests"*

*"I will never need to run
operations in parallel"*

*"I don't want to pay for
Base Deployment Engine"*

777774-3505

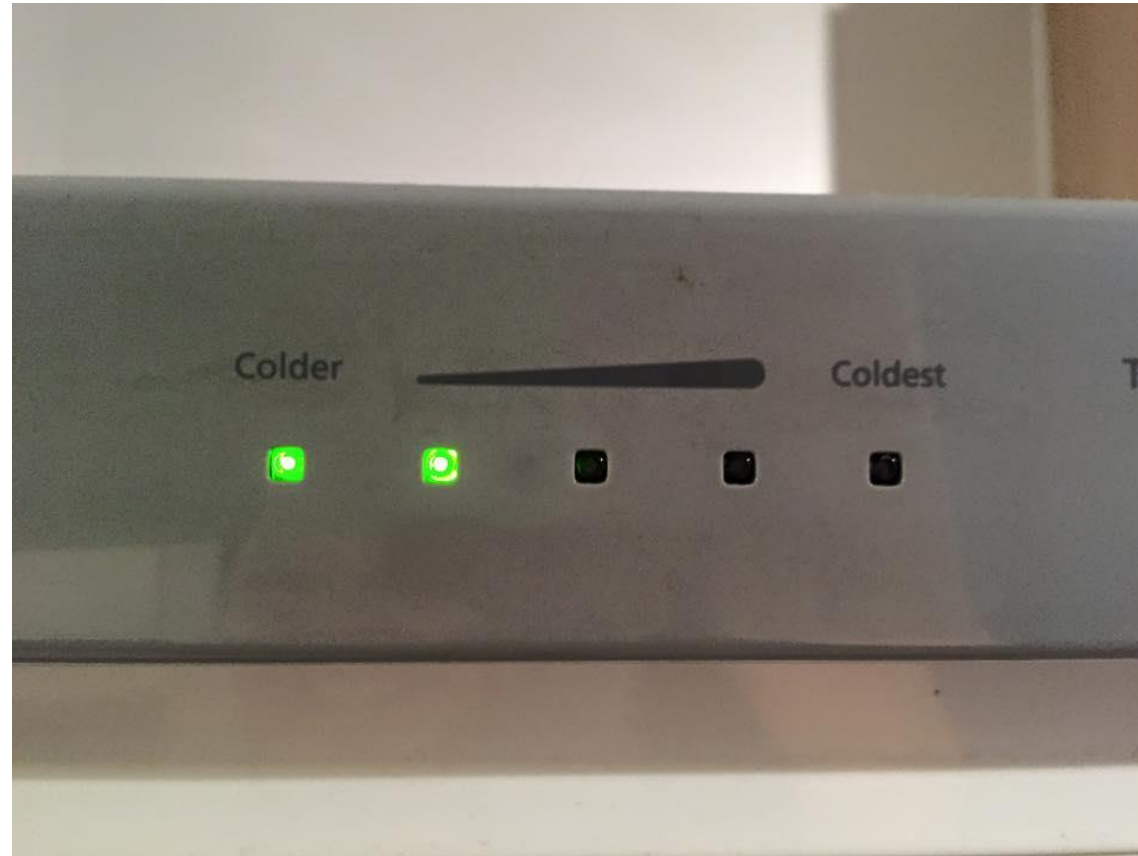
There's a better, more complementary way

(and it allows for parallel development and head start work)

Parallel Activity 1 – UI Development

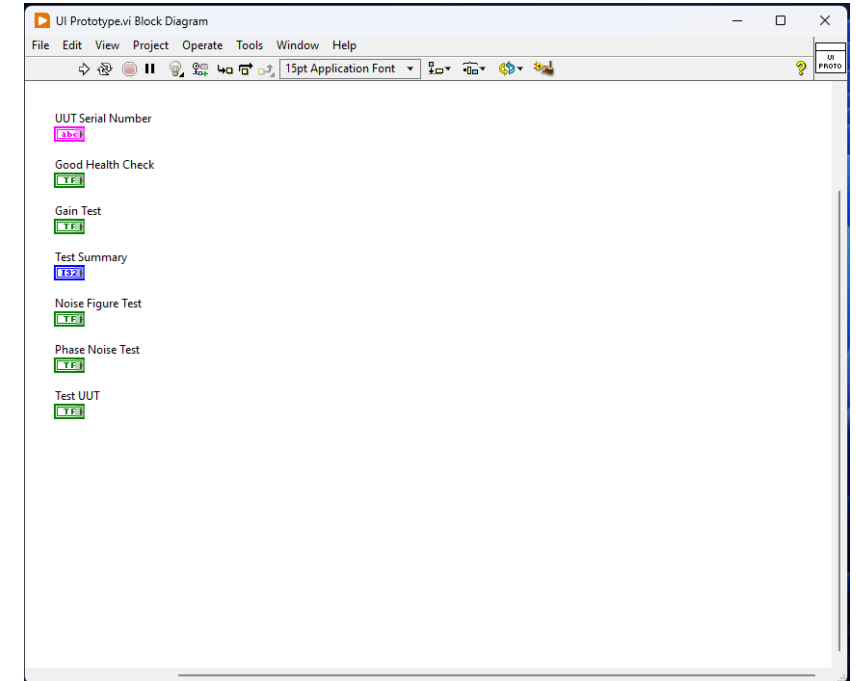
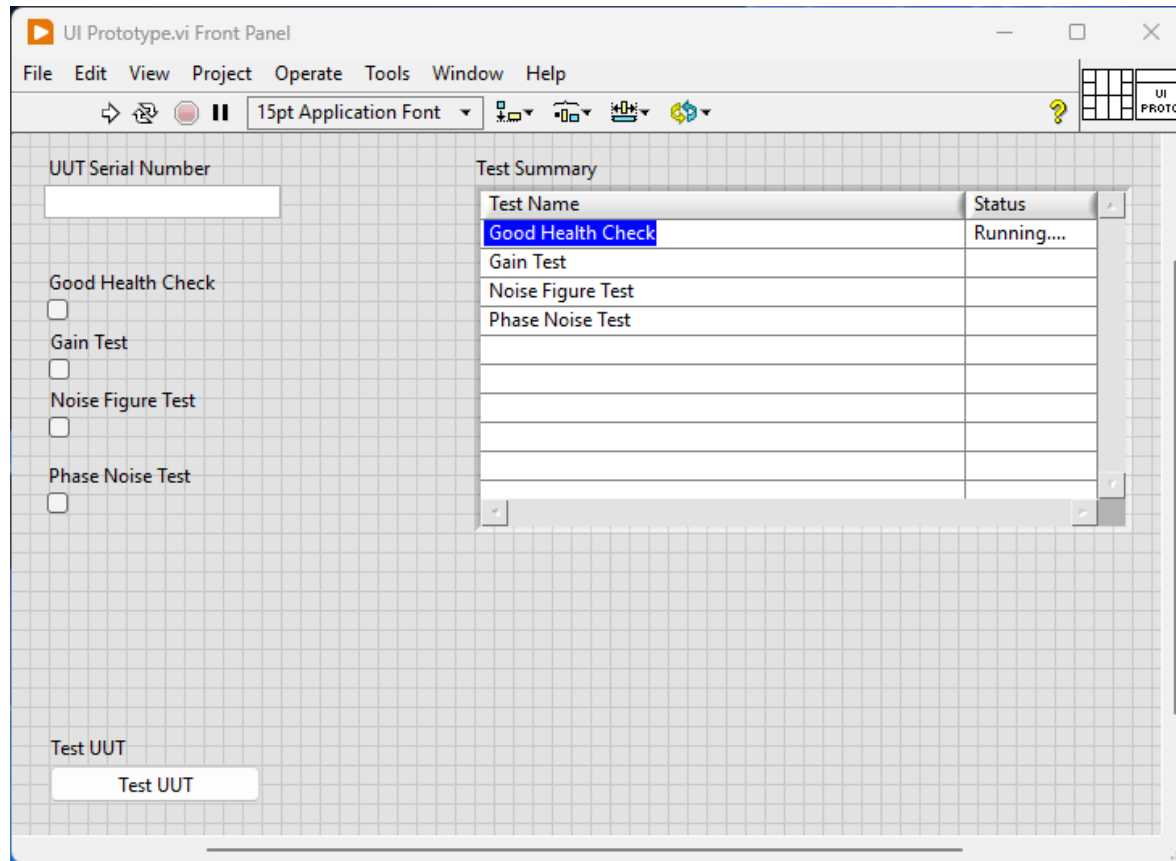
UI Prototyping

- Get early buy-in or approval by the end users of the UI
- Debug any usability issues
- Identify potentially different use cases



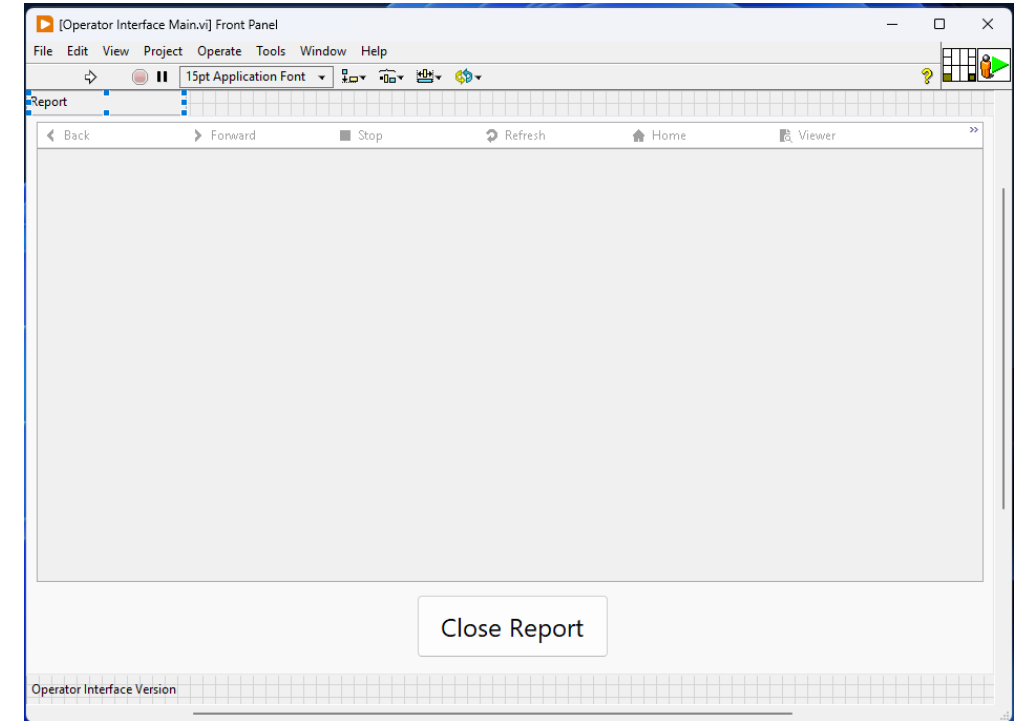
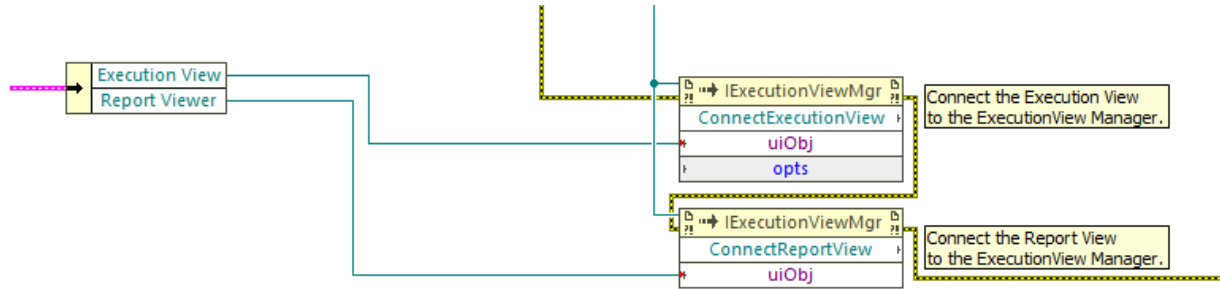
LabVIEW UI Prototyping

- LabVIEW UI prototyping ok for mock-up or layout proofing
- Behavioural examples requires coding
- Test execution requires implementation



LabVIEW + TestStand Controls UI Prototyping

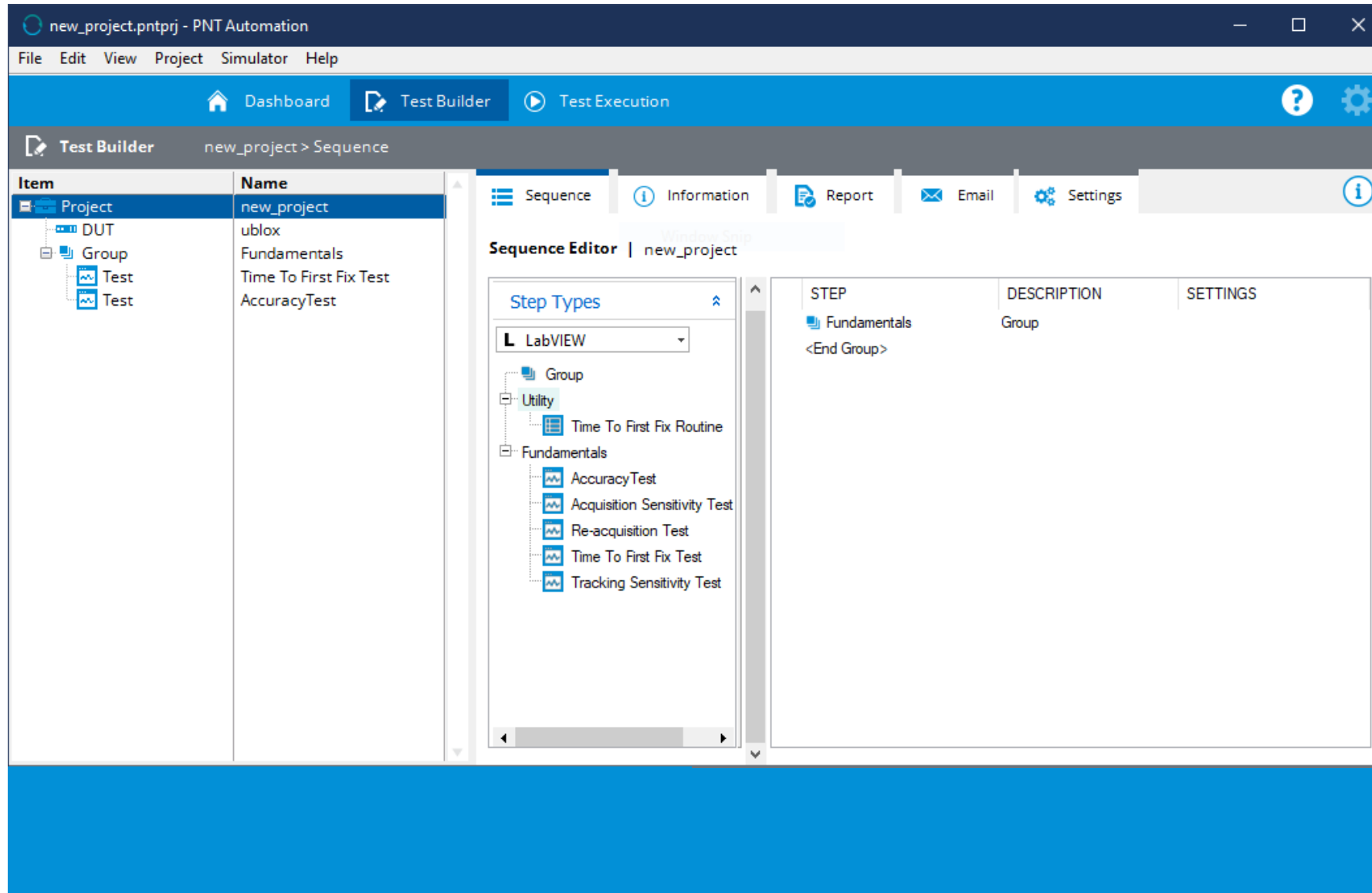
- TestStand Controls have behaviours out of the box
- TestStand API makes sequence execution easy
- TestStand controls allow test progress and reporting with very little code



Demo

UI Prototyping

LabVIEW + TestStand API = Advanced UI



Parallel Activity 2 – Test Sequence Development

Test Sequence Prototyping – Manual Sequence

- A simple sequence of prompts guiding the user through a manual test
 - We get to debug order of operations
 - We get to debug control flow
 - We can use Test Types with None Adapter to record test outcomes
 - We get a fully-fledged report
 - Fast and Easy to create – No programming required

Manual Sequence.seq

Steps: Power On

STEP	DESCRIPTION	SETTINGS
+ Setup (0)		
- Main (3)		
Set PSU Current to 0.5A	NameOf(Step)	
Set PSU Voltage to 5V	NameOf(Step)	
Set PSU output State On	NameOf(Step)	
<End Group>		
+ Cleanup (0)		

Demo

Manual TestStand Sequence

Test Sequence Prototyping – Work Instructions

- A simple sequence of advanced prompts guiding the user through a manual test
 - Fast and Easy to create – Little programming required
 - We can display images to guide the operations

The screenshot displays the NI TestStand Sequence Editor interface. The 'Steps' pane shows a sequence: Setup (0), Main (2), Measure U... Action, Numeric Data, UUT Posti... Numeric Limit Test, <End Group>, and Cleanup (0). The 'Call Stack' pane shows the current execution path: UUT Good Health Check - Main (Manual Se) and MainSequence - Main (Manual Sequence w). A 'DMM Measurement' dialog box is overlaid on the right, featuring an image of a Sparkfun VC830L Digital Multimeter. Below the image, text instructs the user to connect the multimeter and enter the value shown on its screen into the input box, which is highlighted by a red arrow. The input box is labeled 'Numeric Value' and contains the number '0'. A 'Submit Data Entry' button is at the bottom of the dialog box. The status bar at the bottom indicates 'User: administrator' and 'Environment: <Global>'.

NI TestStand - Sequence Editor [Running...]

File Edit View Execute Debug Configure Source Control Tools Window Help

Windows

Sequence Files (1)

Manual Sequence with Wor...

Executions (1)

MainSequence - Manual Se...

Other (0)

MainSequence - Manual Sequence with...

Steps

STEP	DESCRIPTION
+ Setup (0)	
- Main (2)	
Measure U...	Action, Numeric Data
UUT Posti...	Numeric Limit Test, 3
<End Group>	
+ Cleanup (0)	

Steps Report

Call Stack

- UUT Good Health Check - Main (Manual Se)
- MainSequence - Main (Manual Sequence w)

User: administrator Environment: <Global>

DMM Measurement

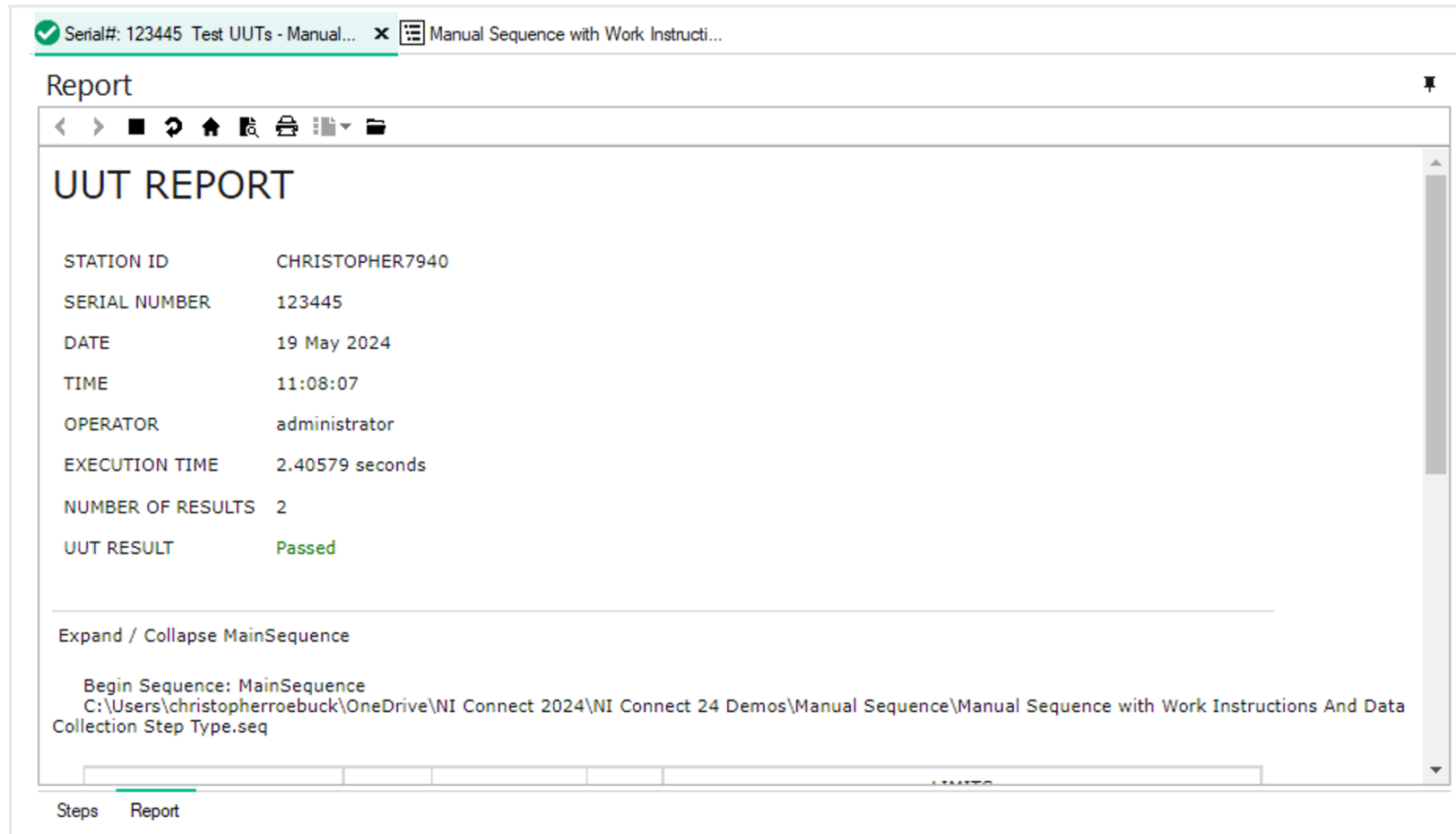
Connect the multimeter as shown in the image and then enter the value shown on its' screen into the input box (highlighted by red arrow) below

Enter value here → Numeric Value 0

Submit Data Entry

Test Sequence Prototyping – Report Prototyping

- Test Types using the None Adapter allow us to provide measurement data and apply limits / comparisons – Results appear in report allowing early validation of format



The screenshot shows a software window titled "Report" with a tab labeled "Serial#: 123445 Test UUTs - Manual...". The window displays a "UUT REPORT" with the following details:

STATION ID	CHRISTOPHER7940
SERIAL NUMBER	123445
DATE	19 May 2024
TIME	11:08:07
OPERATOR	administrator
EXECUTION TIME	2.40579 seconds
NUMBER OF RESULTS	2
UUT RESULT	Passed

Below the report details, there is a section titled "Expand / Collapse MainSequence" with the following text:

Begin Sequence: MainSequence
C:\Users\christopherroebuck\OneDrive\NI Connect 2024\NI Connect 24 Demos\Manual Sequence\Manual Sequence with Work Instructions And Data Collection Step Type.seq

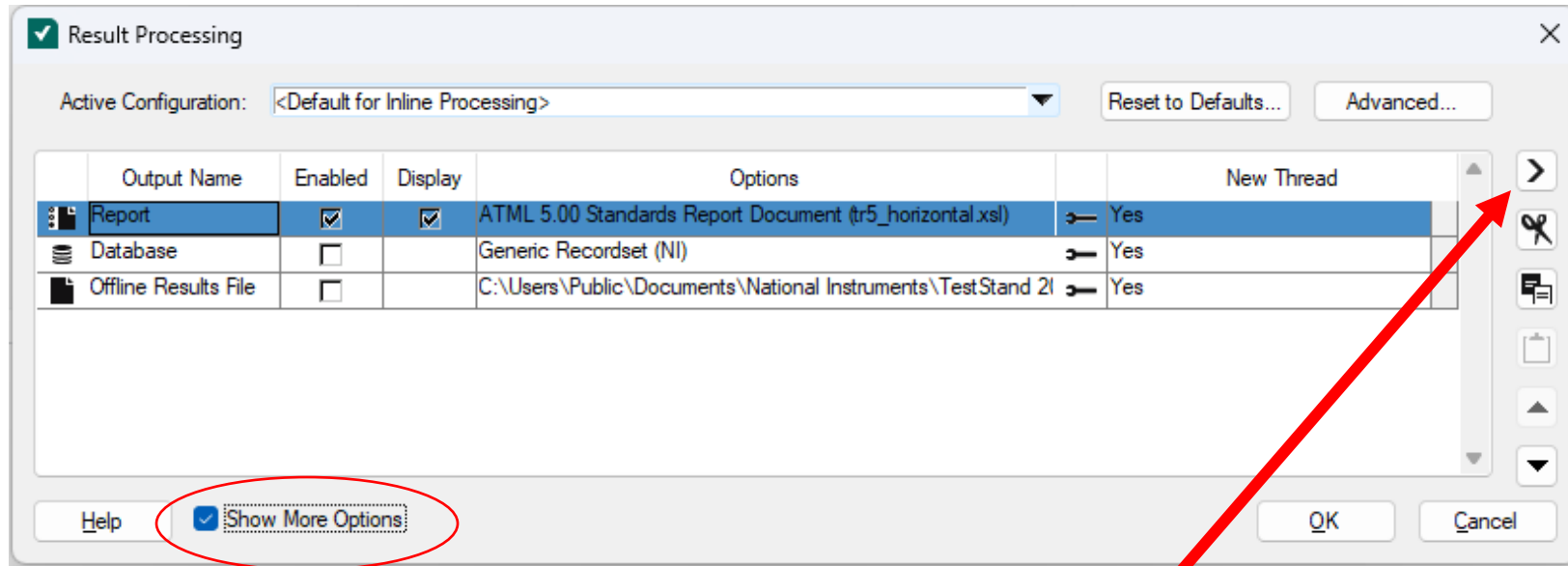
At the bottom of the window, there are two tabs: "Steps" and "Report", with "Report" being the active tab.

Demo

Manual TestStand Sequence with work instructions

Want a different (or multiple) report(s) – No Problem

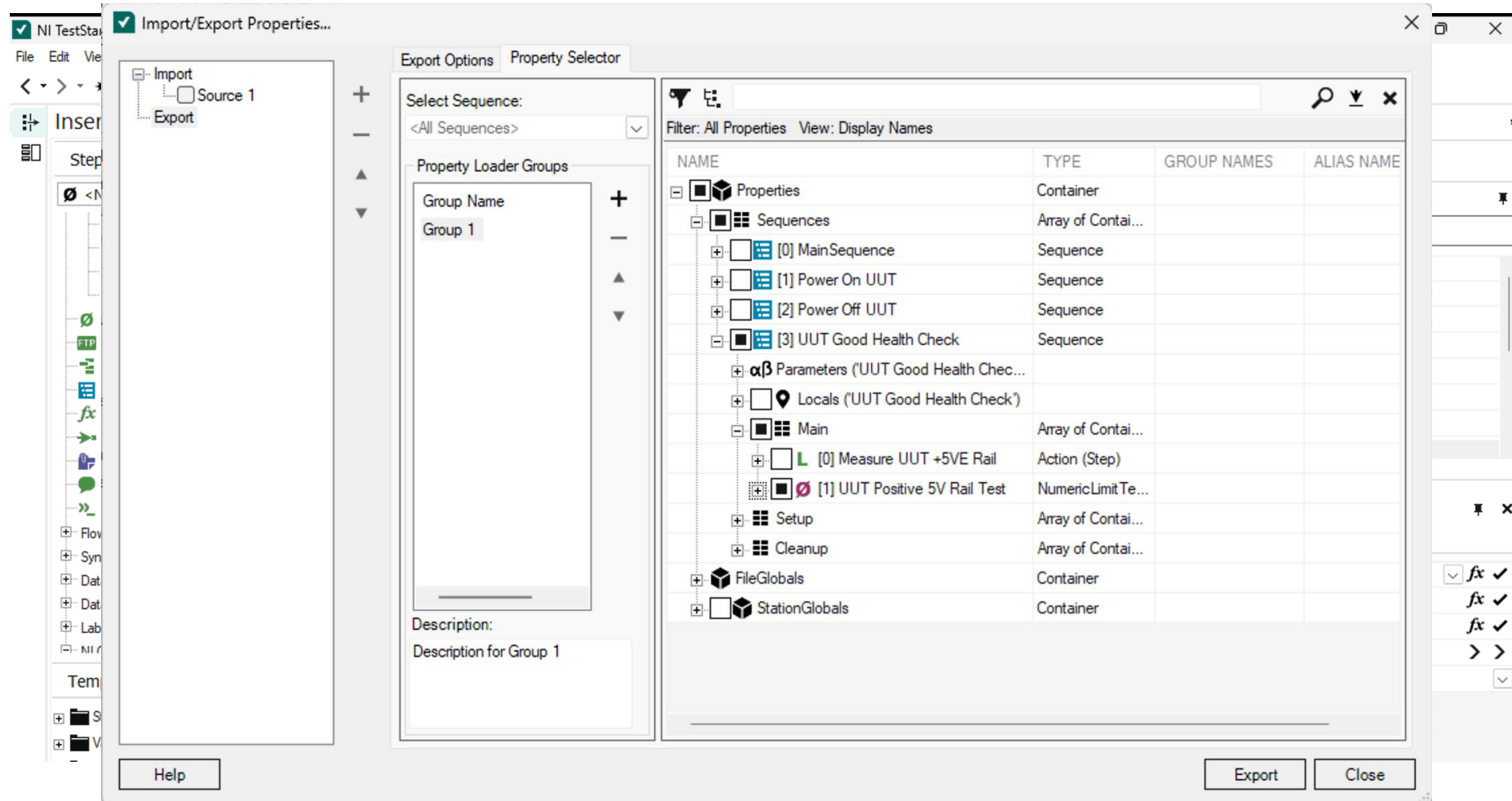
- No need to duplicate report generation code
- TestStand Process model allows multiple report plugins to process the results list
 - Multiple reports of differing formats
- Create your own report plugin for custom report formats



Magic check box makes these options appear

Limits Loading

- Externalise limits by manually setting them up and then exporting them
- Reload using Property loader step type

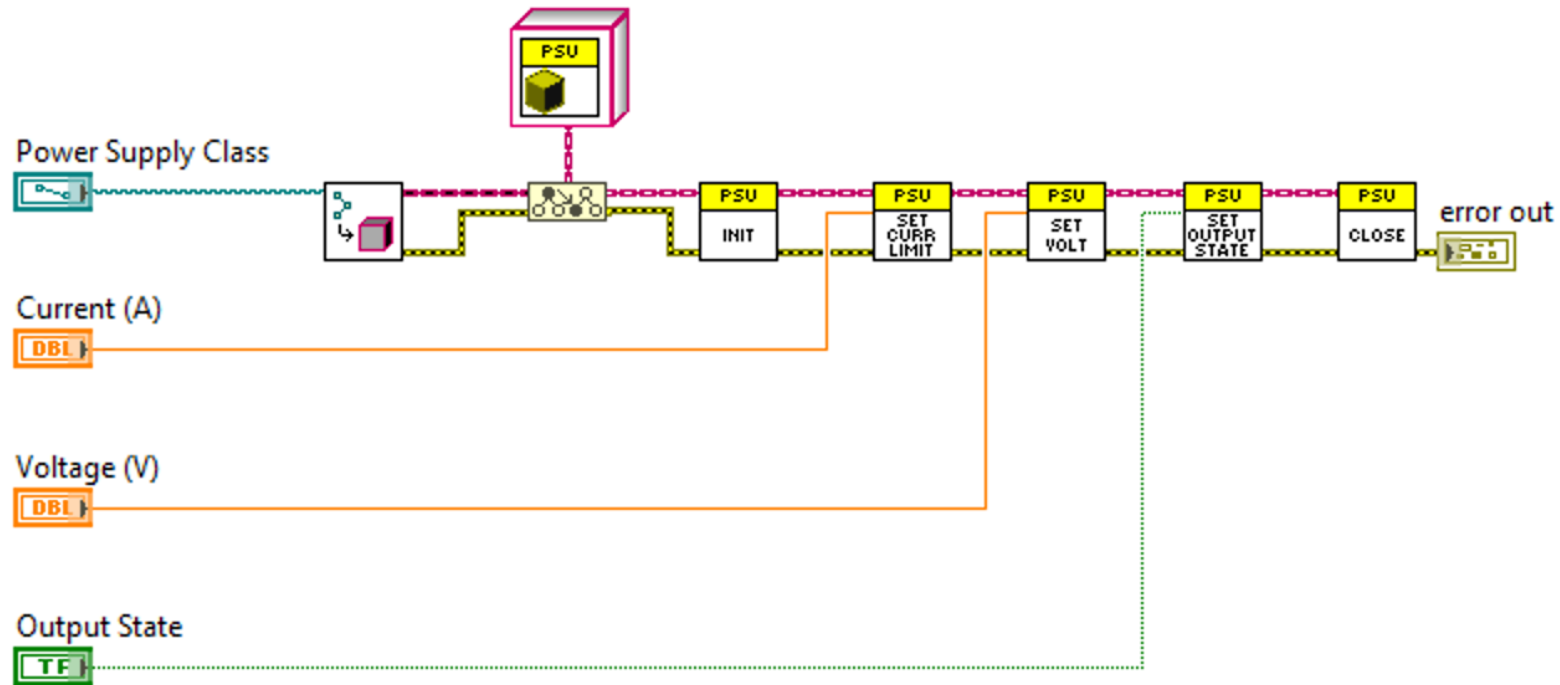


Manual TestStand Sequence with external limits

Parallel Activity 3 – Measurement Code Development

LVOOP Driver Development + Test Harness

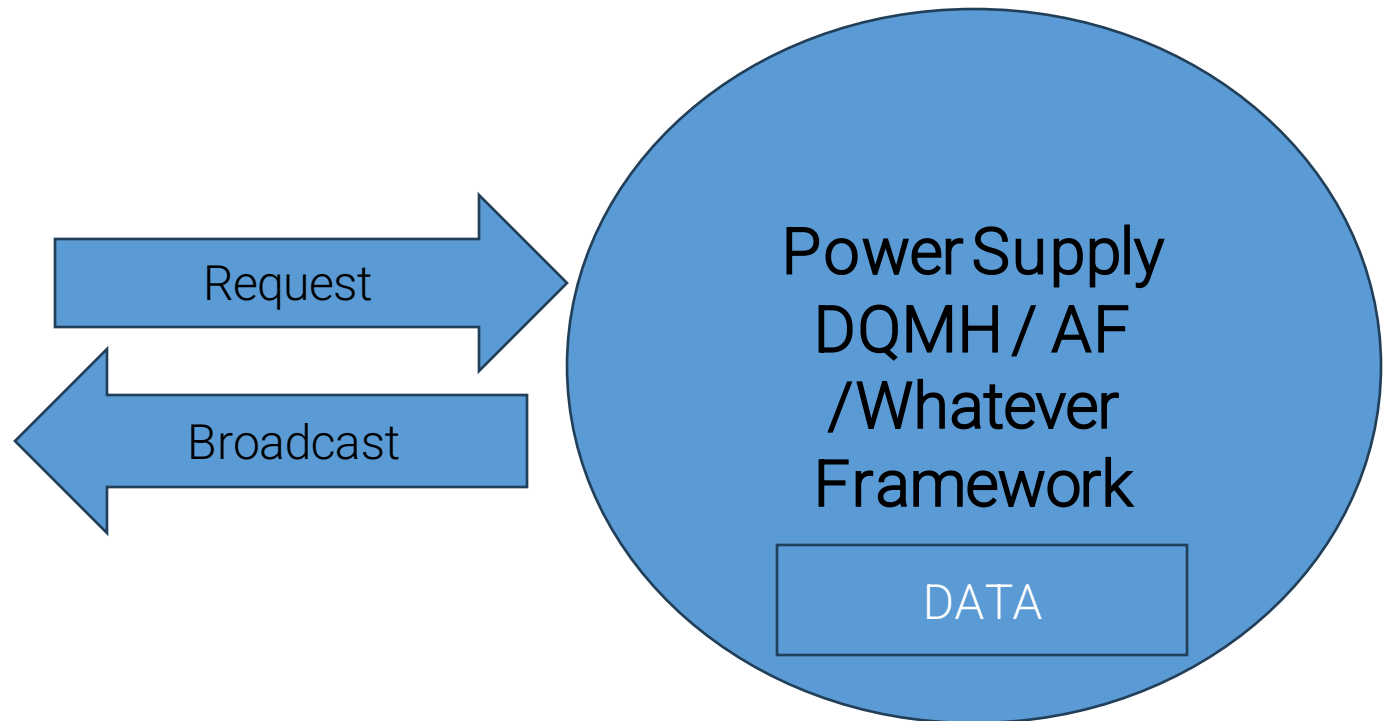
- Simulate Instruments whilst becoming familiar with APIs
- Overload specific instruments
- Scalable / extensible



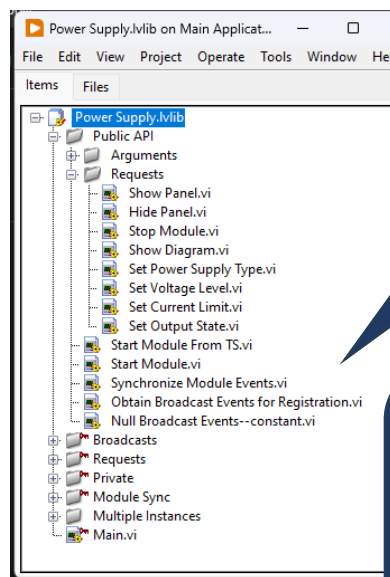
LabVIEW Hardware Abstraction

Instrument / DUT control – QMH / DQMH / AF / WTF etc

- Treat the instrument as an active object
 - An asynchronous entity
 - Request / Broadcast communications
 - Has state
 - Chooses if state suitable for requests



Instrument / DUT control – QMH / DQMH / AF / etc etc



TestStand calls public API = Debugging in LabVIEW

DQMH Common Framework



Power Supply Base.lvclass



Instrument Mimics Manual interaction and debug tools

Extensibility



Agilent 3634A Power Supply.lvclass



Simulated Power Supply.lvclass

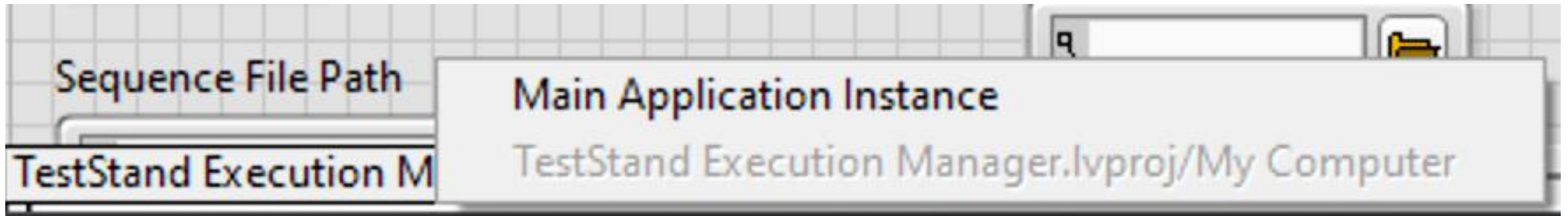


Hardware Abstraction Layer

Demo

PRO TIP

- Set DQMH API Tester to run in Main App Instance



Custom Step Types

- Custom defined Steps
- Available in Insertion Palette
- Define Runtime and Edit time behaviour
- Define actions when new step instance is inserted
- Custom properties allow persistence of data between edit and run time

Demo

Manual TestStand Sequence With Custom Step Types

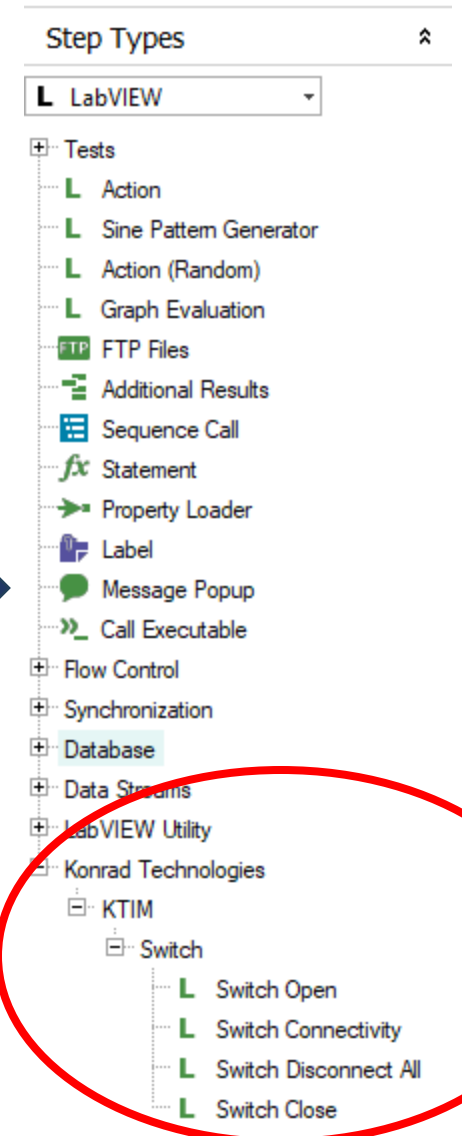
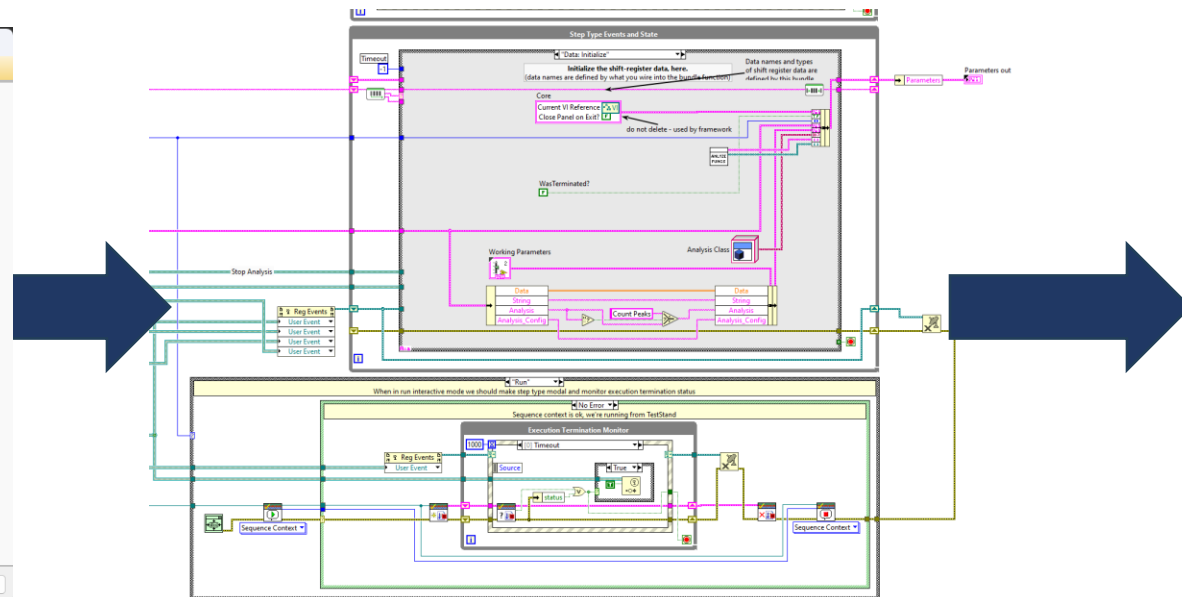
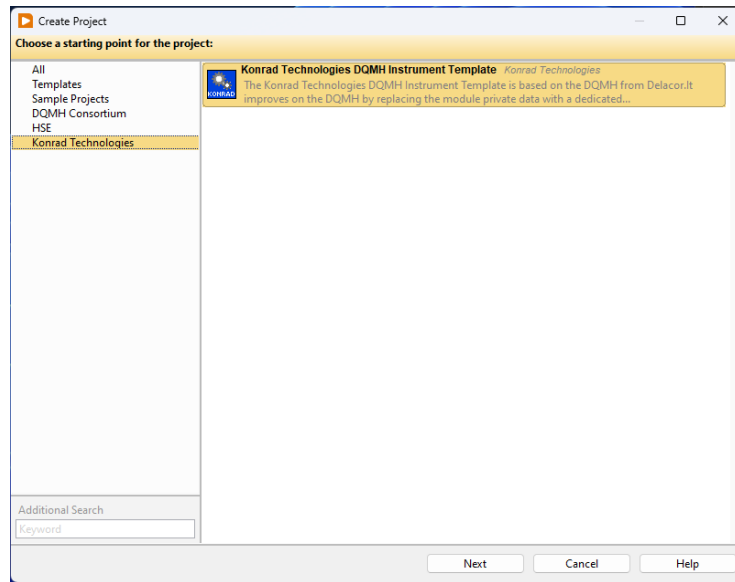
Examples of custom steps



Tooling - Step Templates

Accessibility

Wizards generate code from templates and integrate into TestStand and LabVIEW to make easier to use



In summary

- TestStand provides the features you end up implementing anyway
- Prototype sequences, reports and UIs with real execution much sooner
- Provided guided work examples with interactive LabVIEW steps
- LVOOP allows PoC instrument harnesses
- Asynch LabVIEW 'Module' allows interaction and "sniffing"
- Reduces time to first test

Thank You & Questions?

c.roebuck@Konrad-technologies.co.uk