An Effective Test Strategy is More Than A Killer HAL/MAL

Removing Roadblocks to Making Your Great Idea a Reality

Tie idea to test strategy
Center of Excellence Requirements

- Standard Libraries and Templates
- Internal User Group
- Designated Technical Lead
- Engineer Good Software
- Requirements Gathering and Tracking
- Internal Onboarding
- Software Deployment and Distribution
- Engage in Community Learning
- Engineer Good Software
- Ensure Technical Leadership
- Software Testing and Release
- Requirements Gathering and Tracking
- Standard Libraries and Templates
- Code Management
- Design and Code Reviews
- Development Environment and Core Concepts
- Developer Mastery
- Architecture Mastery
- Software Deployment and Distribution
- Designated Technical Lead
- Organizational Proficiency Plan
- Learning and Development Plan
- Internal Onboarding
- Internal User Group
- External and Global Community
NI Methodology Consulting Services

Accelerating our customer’s test transformation initiatives to deliver business and operational outcomes.

Offerings from NI MCS.

- Benchmarking
- Total Cost Ownership
- Symptom Diagnosis

- CoE Planning
- Project Architecture
- Cost & Resource Est

- CoE Coaching
- Custom Systems
- Skills & On-boarding
Test Strategy
Introduction – Painting the Picture

Selling Your Ideas Management

Garnering Buy-In From You Peers and Other Teams

Ensuring You Have the Processes and Onboarding for Success
Facts vs. Design & Art

**Objective**
(facts, science, there is a right answer)

**Subjective**
(design, art, no right answer, convince others)

- Designing a HAL/MAL
- Learning OO in LabVIEW
Art of Technology

“Our brains evolved to receive a pleasant shot of dopamine when we learn something new and classify it”
Daniel J Levitin – The Organized Mind
p. 32
How Many Times Do You Have That A-Ha Moment When You are In Front of the Laptop?
Why is it that I always get my best ideas while shaving?” - Einstein
The Brain is On or Off?
Actually Different Parts of the Brain

Central Executive (stay-on-task)

Mind Wandering (default state)
Actually Different Parts of the Brain

Mind Wandering
(default state)
Actually Different Parts of the Brain

Central Executive
(stay-on-task)
You Have the Great Idea for a Test Strategy… now what?
Getting a Meeting with the Boss

“Hey Alex, I have the great idea for a Test Strategy… we could…”
Introduction – Painting the Picture

Selling Your Ideas Management

Garnering Buy-In From You Peers and Other Teams

Ensuring You Have the Processes and Onboarding for Success
Industry Experience … High Mortality

Not staffed/resourced.

Project stalls (repeating efforts)

Investments Made…
Fails to reach original objective
“Let’s invest in new reuse ‘standard’ for HW and SW to increase efficiency & margins.”

Program Management
- Win Bids
- Launch Product
- Higher Upfront Cost
- Schedule & Quality Risk
- Low/No Reward? (Savings after launch.)
- Low Motivation
- Risk > Reward

Design & Test Eng
- Create Quality
- Test Equipment On-time
- Initial Pilot Schedule
- Higher $ = Visibility
- More Resources
- Reduce Obsolescence/Support
- Increase Efficiency (Multi Program)
- High Motivation
- Reward > Risk

Factory
- On-time Delivery
- Reliable Quality Process
- Schedule Impact
- Unknown Maintainability
- Forced Upgrades
- Large Operation Savings
- Improved Uptime & Quality
- Undecided
- Risk <> Reward

Perspective

Complex/Competing Nature of Our Environment – Silo’s
WHAT kind of Change?

Test has wide range of potential affects = unclear objective. Engineering effort without good specifications is difficult at best.

- Clear definition of pain points in business terms, not technology or solution.
  - Prioritize, which pain is #1. Is it strong enough to warrant change?
  - Expect different answers from various groups.
- TCO (Total cost of Ownership) framework can provide excellent resource
  - Begins with inclusive model to compare/quantify multiple aspects
  - Neutral approach
- Sponsorship & Assistance
Agenda

Introduction – Painting the Picture

Selling Your Ideas Management

Garnering Buy-In From You Peers and Other Teams

Ensuring You Have the Processes and Onboarding for Success
You Have the your Great Idea for a Test Strategy… now what?
Getting Buy-In: The Balancing Act
Peer Pressure

... Somebody forgot to use VI Analyzer
The Workshop (from Peter Horn’s Session)

- Introduction
- Idea collection
- Decide priorities
- Assign actions

Key Takeaway

Ensure that the entire team is focused on the goal and motivated to achieve success
A Few Workshop Guidelines

1. Identify the Problem to Solve
2. Get the Right People in the Room
3. Get the Right People in the Room for “a Day”
4. Ensure Someone Owns the Followup Plan
No involvement generally means no commitment…

...people generally protect what they build
User Groups

Bach LabVIEW User Group

Vision

Bach is a general purpose LabVIEW User Group. So much music in today's popular culture is really based on the foundations of classical music composers. And the granddaddy of them all is none other than Johann Sebastian Bach. As our first LabVIEW User Group, "Bach" concentrates on LabVIEW as an IDE and how to employ good software principles to build scalable and maintainable software.
Who Is Your Customer?

Plug-In Architecture Aligned the Module to the Requirement

Test Engineering Creates One more Module
Who Is Your Customer?
The Plan – Where Is It
Own the Plan
Someone Needs to Drive the Effort
AGENDA

Introduction – Painting the Picture

Selling Your Ideas Management

Garnering Buy-In From You Peers and Other Teams

Ensuring You Have the Processes and Onboarding for Success
## Where Does the Account Start with Center of Excellence

<table>
<thead>
<tr>
<th>START HERE</th>
<th>BUILD PROFICIENT PRACTICES</th>
<th>ESTABLISH BEST PRACTICES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5. Organizational Proficiency Plan</td>
<td>4. Software Deployment and Distribution</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. Requirements Gathering and Tracking</td>
</tr>
</tbody>
</table>

- Engineer Good Software
- Engage in Community Learning
- Ensure Technical Leadership

**BUILD PROFICIENT PRACTICES**
- Standard Libraries and Templates
- Code Management
- Design and Code Reviews
- Organizational Proficiency Plan
- Learning and Development Plan
- Internal Onboarding

**ESTABLISH BEST PRACTICES**
- Development Environment and Core Concepts
- Developer Mastery
- Architecture Mastery
- Software Deployment and Distribution
- Software Testing and Release
- Requirements Gathering and Tracking
<table>
<thead>
<tr>
<th>Topic</th>
<th>Goal</th>
<th>Actions/Details</th>
<th>Timeframe</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Engineer Good Software</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements Gathering and Tracking</td>
<td>Processes for gathering, tracking and managing requirements exist</td>
<td>None</td>
<td>Complete</td>
</tr>
<tr>
<td></td>
<td>Change cost estimation and sign off processes exist</td>
<td>None</td>
<td>Complete</td>
</tr>
<tr>
<td>Design Reviews</td>
<td>Team conducts iterative design reviews for all projects and has a process for capturing and resolving issues found</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code Reviews</td>
<td>A style guide is used for code development and a code review process exists which makes use of VI Analyser</td>
<td>User group and discussion – see goal slide</td>
<td>End of June 2017</td>
</tr>
<tr>
<td></td>
<td>Team specific additions have been made to the NI Style Guide and VI Analyser test suite</td>
<td>None</td>
<td>Mostly complete</td>
</tr>
<tr>
<td>Standard (Reuse) Libraries and Templates</td>
<td>Opportunities for reuse are identified developed and maintained. Technical lead defines and executes tasks for automating</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The team style guide includes a section which details how the team will document code and deliverables</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code Management (Source Code Control)</td>
<td>Team has selected an SCC solution, and determined a plan to implement it (structure/commit cadence/resolving conflicts)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Team has a test plan which is implemented, and is automated where it is feasible</td>
<td>None</td>
<td>Complete</td>
</tr>
<tr>
<td>Software Testing and Release</td>
<td>Team has a test plan which is implemented, and is automated where it is feasible</td>
<td>Testing processes and code exist for project and reuse code – unit testing, integration testing and DETT procedure</td>
<td>End of December 2017</td>
</tr>
<tr>
<td><strong>Engage in Continuous Community Learning</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organisational Proficiency Plan</td>
<td>Team drafts and executes yearly proficiency plan – includes topics not sufficiently covered by NI training and team specific processes</td>
<td>None</td>
<td>Complete</td>
</tr>
<tr>
<td></td>
<td>Team conducts yearly skills assessment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal Onboarding</td>
<td>Internal training bridges the gap between NI training and productivity on the company LV team. Processes under “Engineer Good Software” are clearly documented</td>
<td>Engineering processes documentation is what is missing, will happen as processes get defined</td>
<td>Ongoing</td>
</tr>
<tr>
<td></td>
<td>A process exists to confirm that the new individual is ready to effectively contribute to the code base</td>
<td>None</td>
<td>Complete</td>
</tr>
<tr>
<td>Internal User Group/Mentoring Program</td>
<td>Meetings are conducted at least 6 times per year and agenda’s are driven by a designated individual</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>None</td>
<td>Complete</td>
</tr>
<tr>
<td>Learning and Development Plan</td>
<td>Team has a training and certification plan for designated individuals and demonstrates execution of the plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>None</td>
<td>Complete</td>
</tr>
<tr>
<td>External and Global Community</td>
<td>Designated individuals attend regional user groups and NI events (NI Week, NI Days, Developer Days)</td>
<td>No regional user groups to attend</td>
<td>Complete</td>
</tr>
<tr>
<td></td>
<td>Designated individuals engage in the forums or monitor relevant discussions</td>
<td>Have attended NI Week multiple times</td>
<td>Complete</td>
</tr>
<tr>
<td>Designated Technical Lead</td>
<td>Individual(s) have been designated to be the technical lead and to drive software process standards</td>
<td>None</td>
<td>Complete</td>
</tr>
<tr>
<td>Development Environment and Core Concepts</td>
<td>Specific roles/team members who only need Core 1-2 understanding are identified, process for proving has been understanding identified and team members have proven such understanding</td>
<td>None</td>
<td>Complete</td>
</tr>
<tr>
<td></td>
<td>Two individuals have attached CLD, team has determined who else needs a CLD skill set</td>
<td>None</td>
<td>Complete</td>
</tr>
<tr>
<td>Architecture Mastery</td>
<td>Designated technical lead has attached a CLA. Existing code has been reviewed by external coach, team has a process for managing/documenting changes and the ability to tackle them</td>
<td>None</td>
<td>Complete</td>
</tr>
<tr>
<td>Software Deployment and Distribution</td>
<td>Team has determined and is using an effective deployment/distribution strategy and defined a regular cadence for builds</td>
<td>Currently have a process but have identified issues so it is under review – see goal slide</td>
<td>End of September 2017</td>
</tr>
</tbody>
</table>
What Are Companies Not Good At?

- Standard Libraries and Templates
- Code Management
- Design and Code Reviews
- Software Testing and Release
- Requirements Gathering and Tracking
- Internal User Group
- External and Global Community
- Organizational Proficiency Plan
- Learning and Development Plan
- Internal Onboarding
- Designated Technical Lead
- Development Environment and Core Concepts
- Developer Mastery
- Architecture Mastery
- Software Deployment and Distribution

Engineer Good Software
Engage in Community Learning
Ensure Technical Leadership
Just Do It’s

- Code Reviews
  - Not hard, just requires a little humility

- User Groups
  - Tip – watch a video at ni.com/coe then meet to discuss and align

- Document and manage requirements
  - Does not have to be perfect! Just do it.
Onboarding

Day 1
1. Quick Start
2. Engineering Manager (EM) skip-level meeting
3. PEER program

Day 2 - 30
1. Engaging assignments
2. Learning our ways
3. Informational onslaught

Day 30+
1. Team integration
2. EM skip-level meeting (again)
3. CTD (continuous training development)
Training Tip: Hire Veteran CPIs to Teach Classes: Mark Ridgley, David Corney, Fabiola De la Cueva
You Have the Great Idea for a Test Strategy… now what?
Blaise Pascal

Mathematician, Inventor, Scientist, Philosopher, Philanthropic, Theologian

... a child prodigy, taught by his father and soon introduced to..
The Salon of the 17th Century

The A-List…

Mathematicians

Philosophers

Scientists

Other

Madem de Stäel (1766-1817)
Someone makes a bold proposition…

… and then defends it.
Building Strong Confident Innovative Teams

—2016 CLA Summit Berlin
“Teamwork Makes the Dream Work”

- Captain Scott Kelly, NIWeek 2019
Stay Connected During and After NIWeek

ni.com/niweekcommunity
facebook.com/NationalInstruments
twitter.com/niglobal
youtube.com/nationalinstruments

ni.com/coe