Continuous Integration within a Model Based Workflow

MathWorks Automotive Conference

Nick Mazzilli, P.E.

10/28/2020

©2020 A123 Systems LLC. All rights reserved.
A123 Systems is a global leader in designing and manufacturing lithium-ion battery cells and systems. Built upon the pursuit of bringing innovation to the market, A123 offers an ever growing portfolio of world class lithium-ion batteries from a full line of high power mild-hybrid systems to energy dense solutions for the transportation and energy storage markets.
Transportation Solutions

**Low Voltage Hybrids**
World-class technology delivers a lightweight option that supports fuel economy improvements through high charge rate capability

**Plug-In & Hybrid Electric Vehicles**
With very long cell life, a modular component strategy, and strong cost focus, A123 quickly provides complete battery systems

**Commercial Vehicles**
Customizable solutions help A123 dominate the market for large commercial vehicles that demand high power
Technology Drives the Energy Storage

Renewable Integration
Large form factor LFP and NMC cells, flexible module solutions and strong focus on cost provides unique technical and commercial solutions

Critical Power
Uninterruptable Power Systems powered by A123 long life and reliable cells offers competitive advantage for back-up power systems

Frequency Regulation
A123 expertise in high power LFP chemistry supports power demand of grid-tied frequency regulation systems

©2020 A123 Systems LLC. All rights reserved.
Project Background

- A123 looking to develop new MBD core for Autosar production programs (from legacy C-code)
- Developed between small team (<2) but how to roll out to larger team (10+) and maintain consistency and quality?
- Partnered with MathWorks to help automate this through a Jenkins build server
ISO 26262 Reference Workflow

Figure 2. Verification and validation activities specified in IEC Certification Kit.

CI Workflows

Individual SWCs
- Check model over standards compliance
- *Generate* Code
- Run test cases on model (MIL vs SIL)
- Polyspace reports
- Traceability reports
- HTML template of model with linked requirements and descriptions
- SVN commit

Integration of SWCs
- External links to all individual SWCs
- Combine shared utilities
- Get all necessary .c and .h files for each SWC
- Composition testing (Application Layer Testing)
- Binary creation, ready to test on hardware
SWC Example

Review Results in MATLAB/Simulink or Jenkins

Tests
1) Simulink Check
2) Code Generation
3) Simulink Test
4) Polyspace
5) Simulink Report Generation

Make changes to your files
Commit your changes in SVN
SVN Post Build Hook Triggers Jenkins
Jenkins Runs Tests

Test Passed?

Yes
Jenkins commits over SVN branch

No
Jenkins sends email to program members with failure report
Simulink version: 9.2
System: COOL_SWC

Run Summary

<table>
<thead>
<tr>
<th>Pass</th>
<th>Fail</th>
<th>Warning</th>
<th>Not Run</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>482</td>
<td>0</td>
<td>166</td>
<td>0</td>
<td>648</td>
</tr>
</tbody>
</table>

- **Identify unconnected lines, input ports, and output ports**
  - Identify unconnected lines, input ports, and output ports in the model
  - **Passed**
    - There are no unconnected lines, input ports, and output ports in this model.

- **Check root model Import block specifications**
  - **Passed**

- **Check diagnostic settings ignored during accelerated model reference simulation**
  - The configuration parameter settings passed the check.

**Static model analysis at each branch commit!**
HTML of Model

"t_max_t_spread_reset_thrsh" (Subsystem)

1. 91711: 1) (Tmax (48VG2CORE_COOL_##30))

Link to Requirement Set
Document: 48VG2CORE_COOL_slnreq
Location/Item: 91711

1) (Tmax

1) (Tmax <= K_COOL_BATTEMPRESET_T) AND (Tspread <= K_COOL_TEMPSPRDRFSET_T)

Model HTML with requirements clearly identified for review
Author(s): Jenkins_Server_Admin
Published on: 01-Sep-2020

Chapter: Requirement Set: 48VG2CORE_COOL_

Requirements analysis of implemented and Verified requirements at each commit!

<table>
<thead>
<tr>
<th>Description</th>
<th>Requirements analysis of implemented and verified requirements at each commit!</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filepath</td>
<td>E:\Jenkins_Worksace\CodeGenration_Simulink_COOL\branches\G248NBOSW-766\COOL\Inputs\Requirements\48VG2CORE_COOL__sreqx</td>
</tr>
<tr>
<td>Revision</td>
<td>6</td>
</tr>
<tr>
<td>Created by</td>
<td>nmazzilli</td>
</tr>
<tr>
<td>Created on</td>
<td>02-Apr-2020 13:35:10</td>
</tr>
<tr>
<td>Modified by</td>
<td>nmazzilli</td>
</tr>
<tr>
<td>Modified on</td>
<td>02-Apr-2020 15:30:38</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Implementation Status</th>
<th>Total</th>
<th>Implemented</th>
<th>Justified</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>44</td>
<td>39</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Verification Status</th>
<th>Total</th>
<th>Passed</th>
<th>Justified</th>
<th>Failed</th>
<th>Unexecuted</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>44</td>
<td>20</td>
<td>4</td>
<td>3</td>
<td>14</td>
<td>3</td>
</tr>
</tbody>
</table>
Integration Example

Make reference to SWCs to integrate
Commit your changes in SVN
SVN Post Build Hook Triggers Jenkins
Jenkins Runs Tests
Tests
1) Shared Utilities regenerated in shared folder
2) Get .c .h files
3) Application Testing
4) Binary Creation
5) Polyspace Reports

Review Results in MATLAB/Simulink or Jenkins

Jenkins commits over SVN branch. Ready for HIL Testing
Jenkins sends email to program members with failure report

Test Passed?

Yes
No
Project Value

• Build server acts as sheriff

• Model reviews are based on generated report metrics not emotions
  + Model Advisor
  + Cyclomatic complexity
  + Polyspace
  + Simulink Test
  + Linked Requirements

• Bug tickets are created based on these reports in Jira

• Integration can still be automated
Next Steps

- Integration testing / composition testing needs to be improved upon
- Add HIL testing to this automation process
- Add PIL testing to this automation process
- Continue to refactor models with best practices and refine library usage
Conclusion

• One tool to develop software, link requirement, write test cases and create reports for ISO compliant workflow

• Built in reports necessary for reviews

• Reviews can be done by team members over Jenkins

• Write Jenkins test steps so that failures at any individual test can trigger a build failure

• 90% of steps that can be automated are

• Recommend top-down approach to control .arxml files for build process