Formalizing Requirements and Generating Requirements-Based Test Cases

Dalton L'Heureux, MathWorks
Why Model-Based Design Anything?
Model-Based Design: Detect Error Earlier to Minimize Costs

Data gathered by Hewlett Packard referred by XB in 2017
https://xbsoftware.com/blog/why-should-testing-start-early-software-project-development/
Model-Based Design ROI Calculation for Aerospace Applications

<table>
<thead>
<tr>
<th></th>
<th>Requirements</th>
<th>Design</th>
<th>Coding</th>
<th>Analysis</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Savings</td>
<td>$3,720,000</td>
<td>$500,000</td>
<td>$1,000,000</td>
<td>$1,500,000</td>
<td>$1,000,000</td>
</tr>
<tr>
<td>Total Investments</td>
<td>$592,000</td>
<td>$592,000</td>
<td>$592,000</td>
<td>$592,000</td>
<td>$592,000</td>
</tr>
<tr>
<td>ROI</td>
<td>528%</td>
<td>528%</td>
<td>528%</td>
<td>528%</td>
<td>528%</td>
</tr>
</tbody>
</table>
Can we Detect Errors Even Earlier?

Data gathered by Hewlett Packard referred by XB in 2017
https://xbsoftware.com/blog/why-should-testing-start-early-software-project-development/
Let’s Look at an Example!

\[ A \Rightarrow B \]
#1: Manually Author Tests with Simulink Test

- **Inputs**
  - Data file (input)
  - Signal Editor
  - Test Sequence
  - MATLAB Code
  - Stateflow

- **System Under Test**

- **Test Harness**
  - Harness Inputs
  - Signal spec. and routing
  - shift_logic

- **Parameters**
  - .xls

- **Assessments**
  - Data file (baseline)
  - Test Assessment
  - Temporal Assessment
  - MATLAB Code
#2: Generate Tests off Requirement Models Using Simulink Design Verifier Blocks

- Simulink Design Verifier can generate test cases that satisfy coverage objectives for your model, including Decision, Condition, MCDC, Relational Boundary, and Custom Objectives.

- Custom Objectives are modeled using the following constructs:
  - **Test Objective** - Define values of a signal a test case must satisfy
  - **Test Condition** - Constrain values of a signal
  - **Verification Subsystem** - Conceals logic/objectives from analysis

- What do your test cases ACTUALLY mean?
  - Structural Coverage Based Tests?
  - Low-Level Requirements Based Tests used to test generated code?
  - Requirements Based Tests generated from an *Independently* developed requirement model?
  - Requirements Based Tests generated from a Design Model? What about with *Independently* defined expected outputs?
#3: Generate Tests off Requirement Models Using the Requirements Table Block

<table>
<thead>
<tr>
<th>Index</th>
<th>Summary</th>
<th>Precondition</th>
<th>Duration</th>
<th>Postcondition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>getSimulationTime() == 0</td>
<td>true</td>
<td></td>
<td>WaitForComms</td>
</tr>
<tr>
<td>2</td>
<td>false</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>WaitForComms</td>
<td>GCSCmds.WIFIConnected &amp;&amp; ...</td>
<td>Init</td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>Init</td>
<td>GCSCmds.CalibrateCmd</td>
<td>Calibration</td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>Calibration</td>
<td>GCSCmds.GCS_MissionMode == uint8(1) &amp;&amp; ...</td>
<td>ReadyForTO</td>
<td></td>
</tr>
</tbody>
</table>

Inconsistent with requirement 2.7 for inputs:

<table>
<thead>
<tr>
<th>Time</th>
<th>Step</th>
<th>Cmd</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-0.2</td>
<td>1-3</td>
<td>Calibration</td>
</tr>
<tr>
<td>0.3</td>
<td>4</td>
<td>TrackAlt</td>
</tr>
<tr>
<td>0.4</td>
<td>5</td>
<td>Track3D</td>
</tr>
<tr>
<td>0.5</td>
<td>6</td>
<td>LostBall</td>
</tr>
<tr>
<td>0.6</td>
<td>7</td>
<td>LostBall</td>
</tr>
<tr>
<td>0.7</td>
<td>8</td>
<td>LostBall</td>
</tr>
<tr>
<td>0.8</td>
<td>9</td>
<td>LostBall</td>
</tr>
<tr>
<td>0.9</td>
<td>10</td>
<td>LostBall</td>
</tr>
</tbody>
</table>

State variables:

- State V_BODY
- State Altitude
- State Battery Volts

Matlab Expo
Additional Resources

Product Page:
- Requirements Toolbox - MATLAB & Simulink (mathworks.com)

Documentation:
- Requirements Toolbox Documentation (mathworks.com)
- Requirements Definition (mathworks.com)
- Requirements Table Block (mathworks.com)
  - Use a Requirements Table Block to Create Formal Requirements (mathworks.com)

Examples:
- Requirements Toolbox — Examples (mathworks.com)
- Formalize Requirements in Simulink Models — Examples (mathworks.com)
Thank you