How VectorCAST supports ASPICE and ISO 26262
Practical Examples

Vector Congress North America 2019
Agenda

- Brief History of Software Development Standards
  - Solutions for Software Development
  - Implementing Workflows with VectorCAST
  - Mapping VectorCAST Features
  - Conclusions
Brief History of Software Development Standards

30 Years of Evolution

Note: [xxxx] is the initial release date
Brief History of Software Development Standards

Automotive SPICE®
Process Reference Model
Process Assessment Model

ISO 26262
Road vehicles - Functional safety -
Part 6:
Product development at the software level
Brief History of Software Development Standards

Large & Challenging

Automotive SPICE®

ISO 26262

1. Vocabulary
2. Management of functional safety
3. Concept phase
4. Product development at the system level
5. Product development at the hardware level
6. Product development at the software level
7. Production and operation
8. Supporting processes
9. ASIL-oriented and safety-oriented analyses
10. Guideline on ISO 26262

1 Part, 128 Pages

Figure 1. Automotive SPICE v3.1 2017; Page 12

10 Parts, 486 Pages

Figure 2. ISO 26262-6 2011; Page viii
Brief History of Software Development Standards

- Solutions for Software Development
  - Implementing Workflows with VectorCAST
  - Mapping VectorCAST Features
  - Conclusions
PREEvision supports the development of E/E architectures, including key process areas according to ASPICE.

- Includes the Design of Safety Relevant Systems according to ISO 26262.
Solutions for Software Development

CANdelaStudio, vFlash, CANoe.DiVa, & Indigo - Diagnostics

CANdelaStudio
Author Diagnostic Specification

ODXStudio
Edit ODX Data

CANdela Diagnostic Data

CANdela Template

MICROSAR DCM/DEM
Generate ECU SW Diagnostics

CANoe
Test Functions and Diagnostics

CANoe.DiVa
Validate ECU SW Diagnostics

vFlash
Update ECU SW

Indigo
Vehicle and System Diagnostics

Engineering Services for perfect fit
Solutions for Software Development

CANoe, vTESTstudio, VT System, & vVIRTUALtarget

Benefits

- Full support in the development process, from SW unit test to system validation
- Uniform test management, test automation (CI), result analysis and traceability
Agenda

Brief History of Software Development Standards
Solutions for Software Development

- **Implementing Workflows with VectorCAST**
  - Mapping VectorCAST Features
  - Conclusions
Implementing Workflows with VectorCAST

Software Development

Automotive SPICE®

ISO 26262

Figure 5. Automotive SPICE v3.1 2017; Page 12

Figure 6. ISO 26262-6 2011; Page vii
Implementing Workflows with VectorCAST

Refine Further

Automotive SPICE®

ISO 26262
Implementing Workflows with VectorCAST

Focus on Software

Automotive SPICE®

ISO 26262
Implementing Workflows with VectorCAST

VectorCAST Support

**Automotive SPICE®**
- Software Req Analysis
- Software Arch Design
- Software & Unit Construction

**ISO 26262**
- Specification of Software Safety Req
- Software Arch Design
- Software Integration & Test

- VectorCAST Supports
## Remarkably Similar

<table>
<thead>
<tr>
<th><strong>Automotive SPICE®</strong></th>
<th><strong>ISO 26262-6</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5  Initiation of product development at the software level</td>
</tr>
<tr>
<td>SWE.1 Software Requirements Analysis</td>
<td>6  Specification of software safety requirements</td>
</tr>
<tr>
<td>SWE.2 Software Architectural Design</td>
<td>7  Software architectural design</td>
</tr>
<tr>
<td>SWE.3 Software Detailed Design and Unit Construction</td>
<td>8  Software unit design and implementation</td>
</tr>
<tr>
<td>SWE.4 Software Unit Verification</td>
<td>9  Software unit testing</td>
</tr>
<tr>
<td>SWE.5 Software Integration and Integration Test</td>
<td>10  Software integration and testing</td>
</tr>
<tr>
<td>SWE.6 Software Qualification Test</td>
<td>11 Verification of software safety requirements</td>
</tr>
<tr>
<td></td>
<td>Annex C  Software configuration</td>
</tr>
</tbody>
</table>
Three Workflows

9 Software unit testing

Individual units or modules are tested. It involves testing of source code by developers.

10 Software integration and testing

Individual modules are grouped together and tested. The purpose is to determine that modules are working as expected once they are integrated.

11 Verification of software safety requirements

Testing is performed on the whole system by checking whether the system or application meets the requirement specification document.
Agenda

Brief History of Software Development Standards
Solutions for Software Development
Implementing Workflows with VectorCAST

*Mapping VectorCAST Features*

Conclusions
What we see at most customers is a software architect or safety manager role. The person(s) in this ‘role’ owns the responsibility to define, specify, develop, monitor, and improve the software development process.

The mapping will show how various features in VectorCAST could support your software development process.

The applicability of these features to your software development process will have to be evaluated on a project by project basis.
<table>
<thead>
<tr>
<th>ISO 26262-6</th>
<th>VectorCAST Features</th>
</tr>
</thead>
</table>
| 5 Initiation of product development at the software level | Unit Testing  
System Testing  
Data Flow Static  
Data Flow Dynamic | Static Analysis  
Requirements Gateway  
Control Flow Static  
Control Flow Dynamic |
| 6 Specification of software safety requirements  | Unit Testing  
System Testing  
Data Flow Static  
Data Flow Dynamic | Static Analysis  
Requirements Gateway  
Control Flow Static  
Control Flow Dynamic |
| 7 Software architectural design                 | Unit Testing  
System Testing  
Data Flow Static  
Data Flow Dynamic | Static Analysis  
Requirements Gateway  
Control Flow Static  
Control Flow Dynamic |
| 8 Software unit design and implementation       | Unit Testing  
System Testing  
Data Flow Static  
Data Flow Dynamic | Static Analysis  
Requirements Gateway  
Control Flow Static  
Control Flow Dynamic |
| 9 Software unit testing                         | Unit Testing  
System Testing  
Data Flow Static  
Data Flow Dynamic | Static Analysis  
Requirements Gateway  
Control Flow Static  
Control Flow Dynamic |
| 10 Software integration and testing             | Unit Testing  
System Testing  
Data Flow Static  
Data Flow Dynamic | Static Analysis  
Requirements Gateway  
Control Flow Static  
Control Flow Dynamic |
| 11 Verification of software safety requirements  | Unit Testing  
System Testing  
Data Flow Static  
Data Flow Dynamic | Static Analysis  
Requirements Gateway  
Control Flow Static  
Control Flow Dynamic |
| Annex C Software configuration                 | Unit Testing  
System Testing  
Data Flow Static  
Data Flow Dynamic | Static Analysis  
Requirements Gateway  
Control Flow Static  
Control Flow Dynamic |
### 5 Initiation of product development at the software level

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>T.1</td>
<td>UT, ST, SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
</tbody>
</table>
### Mapping VectorCAST Features

#### 7 Software architectural design (SWE.2)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>T.2</td>
<td>UT, ST</td>
<td>UT, ST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T.3</td>
<td>DFS, CFS</td>
<td>UT, ST, SA</td>
<td>UT, SA, DFS, CFS</td>
<td>SA, CFS, DFS</td>
<td>CFS, DFS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T.4</td>
<td>UT</td>
<td>UT</td>
<td>UT, ST</td>
<td>UT, ST</td>
<td>UT, CFS</td>
<td>UT, ST</td>
<td></td>
</tr>
<tr>
<td>T.5</td>
<td>UT, ST</td>
<td>UT, ST</td>
<td>UT, ST</td>
<td>UT, ST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CFS, DFS</td>
</tr>
</tbody>
</table>
# Mapping VectorCAST Features

## 8 Software unit design and implementation (SWE.3)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
<th>j</th>
</tr>
</thead>
<tbody>
<tr>
<td>T.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T.8</td>
<td>UT, ST</td>
<td>UT, ST, SA</td>
<td>UT, ST, DFD</td>
<td>SA</td>
<td>UT, SA</td>
<td>SA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T.9</td>
<td>UT, ST</td>
<td></td>
<td>UT, ST, CFS, CFD</td>
<td>UT, ST, DFS, DFD</td>
<td>UT, ST, SA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## 9 Software unit testing (SWE.4)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>T.10</td>
<td>UT, RGW</td>
<td>UT</td>
<td>UT</td>
<td>UT</td>
<td>UT</td>
</tr>
<tr>
<td>T.11</td>
<td>UT, RGW</td>
<td>UT</td>
<td>UT</td>
<td>UT</td>
<td>UT</td>
</tr>
<tr>
<td>T.12</td>
<td>UT</td>
<td>UT</td>
<td>UT</td>
<td>UT</td>
<td>UT</td>
</tr>
</tbody>
</table>
## Mapping VectorCAST Features

### 10 Software integration and testing (SWE.5)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>T.13</td>
<td>UT, ST, RGW</td>
<td>UT, ST</td>
<td>UT, ST</td>
<td>UT, ST</td>
<td>UT, ST</td>
</tr>
<tr>
<td>T.14</td>
<td>UT, ST, RGW</td>
<td>UT, ST</td>
<td>UT, ST</td>
<td>UT, ST</td>
<td>UT, ST</td>
</tr>
<tr>
<td>T.15</td>
<td>UT, ST</td>
<td>UT, ST, CFD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## 11 Verification of software safety requirements (SWE.6)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>T.16</td>
<td>UT, ST</td>
<td>UT, ST</td>
<td>UT, ST</td>
</tr>
</tbody>
</table>
## Annex C Software configuration

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1</td>
<td>UT, ST</td>
<td>UT, ST</td>
<td>UT, ST</td>
</tr>
</tbody>
</table>
Conclusions

VectorCAST can generate reports and artifacts to cover most of the software development processes in ASPICE and ISO-26262

- ASPICE SWE.1-6 and ISO-26262 Part 6 are remarkably similar
- Contact us to learn more about applying VectorCAST to your workflows
- Consider using vVIRTUALtarget to start developing your AUTOSAR applications before your micro is selected
- How to learn more? Come to our Software Testing Factory TechDay on November 19th
For more information about Vector and our products please visit

www.vector.com

Author:
Krueger, Kurt
Vector North America