NATEP Project: asureSign Aero

Workshop Report

Identification of Needs for Tool Support in Meeting Aircraft Avionics Systems, Hardware & Software Certification Standards

Project Ref.: NATEP Grant MA005
Project Task: WP2 DO-178/DO-254 and ISO26262 Comparison

Prepared by: Dr. Chris Harper (contractor)
Project Partner: University of Bristol
Date of Issue: 30th July 2014
List of Delegates

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
<th>Name</th>
<th>Affiliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chris Harper</td>
<td>University of Bristol</td>
<td>Duncan Brown</td>
<td>Controls &amp; Data Services</td>
</tr>
<tr>
<td>Kerstin Eder</td>
<td>University of Bristol</td>
<td>Phil Hall</td>
<td>Thales Quintec</td>
</tr>
<tr>
<td>Dejanira Araiza Illan</td>
<td>University of Bristol</td>
<td>Robin Cook</td>
<td>Thales Quintec</td>
</tr>
<tr>
<td>Piotr Trojanek</td>
<td>University of Bristol</td>
<td>Jean-Luc Johnson</td>
<td>Airbus</td>
</tr>
<tr>
<td>Jim Thomas</td>
<td>TVS</td>
<td>Ian Murphy</td>
<td>Creative Intellect Consulting</td>
</tr>
<tr>
<td>Mike Bartley</td>
<td>TVS</td>
<td>Dave Anders</td>
<td>Cobham Mission Equipment</td>
</tr>
<tr>
<td>Serrie Chapman</td>
<td>TVS</td>
<td>Daniel Scott</td>
<td>Cobham Mission Equipment</td>
</tr>
<tr>
<td>Nigel Charman</td>
<td>Rolls Royce</td>
<td>Jim Desa</td>
<td>Ultra Electronics</td>
</tr>
</tbody>
</table>

Introduction

This report is a summary of a workshop of the project “asureSign Aero”, which is funded by the National Aerospace Technology Exploitation Programme (NATEP) and constitutes the output of Work Package 2 of the project.

NATEP is a £40M programme funded by the UK Government’s Advanced Manufacturing Supply Chain Initiative, and managed by ADS through the regional aerospace alliances in the UK. The programme is aimed at SME companies in the aerospace supply chain, with the objective of providing support for development of their own innovative technologies and technology management capabilities. The programme’s aim is to deliver one hundred novel technologies through collaborative projects within the UK aerospace supply chain.

The asureSign Aero project is to explore whether the tool asureSign, developed originally for requirements traceability management for hardware in the Automotive industry, could be extended to provide better support for Aerospace projects. asureSign Aero is led by Test & Verification Solutions Ltd (TVS), the suppliers of asureSign, with support from the University of Bristol (UoB) as the Academic Partner, and Rolls Royce plc (RR) as End User Partner.

The project work programme will consist of a comprehensive analysis of avionics safety standards, a process of identification of new product features (led by UoB), an “agile process” to deliver the features selected for development, and finally an evaluation on real avionics development project (to be decided).

The work programme consists of the following work packages (WPs):

- **WP1**: DO-178/DO-254 Assessment
- **WP2**: DO-178/DO-254 and ISO26262 Comparison
- **WP3**: asureSign New Features Identification
- **WP4**: asureSign New Features First Iteration
- **WP5**: asureSign New Features Second Iteration
- **WP6**: asureSign New Features Third Iteration
- **WP7**: asureSign Evaluation

This report constitutes the output of WP2.
The agenda for the workshop was as follows:

**Morning session:**
1. Introduction
   a. asureSign Aero NATEP Project (TVS)
   b. Project Workshop Programme (UoB)
   c. System, HW & SW Assurance Standards (UoB)

2. Examples of Problems with Tool-chain Integration
   a. Rolls Royce
   b. TVS

**Afternoon session:**
3. Round Table Discussions – current issues and future requirements
4. Overview of asureSign (TVS)

The aim of the workshop was to invite the delegates to share their experiences of requirements traceability management in aerospace projects, so that real user needs for traceability management functionality could be identified. Therefore, delegates were invited to make comments at any time.

The following pages provide a summary review of the comments that were made in each presentation, and the summary of the afternoon round-table discussion lists the major issues and functional features that were suggested.
Overview of Presentations
The following reviews cover the presentations made in the morning session.

Introductory Presentation and Overview of Aviation Standards
Presenters: Chris Harper (UoB) and Jim Thomas (TVS)

The first presentation provided an introduction to the asureSign Aero project, to explain the context of the workshop, covering the NATEP programme, history of the project proposal, and the workshop programme planned for WP2 and WP3.

The presentation then went on to provide a brief overview of the main international aerospace standards for design assurance of systems, namely SAE ARP 4754A, RTCA DO-178C and RTCA DO-254.

The aim of this presentation was to summarise the standards to serve as a launching point for subsequent discussions.

Each standard was reviewed in terms of the general models they provide of aircraft system design processes, and the specific guidance that each standard offers on the subject of traceability.

A copy of the presentation slides is provided in Annex A.

Comments:
During the presentation various comments and suggestions were noted:

General comments about Traceability Management:
- Management of product variations is important – this is generally done by maintaining a superset of requirements with requirements selection for particular variants
- Traceability Reviews - it is important that traceability data can be presented for review (not confined to viewing in a tool)
- Interaction with stakeholders in all levels of the design flow is needed. This is perceived as a weakness of how design is done now.
- High-level requirements are hard to translate between people that make them happen (hardware & software designers). However a design problem is solved, people working at all levels of design need to understand.
- Traceability to other processes – requirements flow between the main design processes to other processes, in particular safety assessment.

More Specific Comments about Traceability Issues in Software (from the review of DO-178C):
- Variable requirements – how do you deal with code that is designed for variable requirements? Possible options might include:
  - Verifying the same generic code and then having a database per variant
  - Defining a superset of requirements that has some “tailoring” so that you can configure into the variants
- Variability issues in code – management of requirements by the tool to identify issues with code (e.g., for inherited code that nobody knows what it does anymore)
- Software function overloading causes traceability issues
- Traceability management does not provide decision coverage, but with object-oriented software the two topics might overlap (e.g., overloading a function in a package, and requirements as packages definitions)
- Formalised practices for OOP (languages, processes, methodologies, etc.) are very recent in standards/guidelines (only in DO-178C not earlier versions)

**Examples of problems with Tool-chain Integration**

*Presenters: Nigel Charman (RR), Serrie Chapman (Infineon/TVS)*

Two presentations were made by asureSign Aero project members Rolls Royce and TVS, as examples of the issues that the workshop was intended to identify. The intention was to encourage delegates to contribute their own experiences in the afternoon session, but the talks produced several on-the-spot discussions as well.

**Presentation from Rolls Royce**

Nigel Charman gave a talk (verbal only, no presentation slides) on experiences at Rolls Royce and their issues regarding requirements traceability management.

Rolls Royce manufacture gas turbines, primarily for aircraft but also for other industry sectors such as marine products, and have divisions involved in related fields such as tidal generation and fuel cells.

Rolls have joined with other supplier companies to form the engine control systems manufacturer AEC. The company is adopting PTC Integrity for requirements capture, change management, and traceability. The principal products developed by AEC are:

- Engine Management systems – DAL A
- Health Monitoring systems – DAL C and DAL E (so is Integrity too expensive/too feature-rich for these units?)

In the past, requirements management used to mean “putting them on a list and linking to tests”. Every development unit managed their requirements different tools such as DOORS, Lifespan, and Documentum. Traceability was usually done manually (“man-draulically”) using spreadsheets (Excel), MS Access databases, and occasionally some very old tools. Requirements management both at high and low levels was done by assigning a unique number and then assigning tests to each one.

What RR/AEC wants is a common method of working and to be able to manage software tests, hardware tests, and system tests. The following management processes need to be supported:

- Changes to design for efficiency but no requirements change;
- Changes to a rig affecting some tests which can then affect other tests;
- Any software that is sub-contracted out must be managed to ensure that it is linked in to in-house work, and that change management in these situations is controlled;
- Impact analysis when changes are proposed – when requirements-based tests change, how do they modify the whole chain of verification, and how does information modification affect the testing environment?
**Presentation from TVS**

Serrie Chapman from TVS made a presentation on some of the experiences that TVS has had working in conjunction with Infineon on the use of asureSign in the automotive sector, to serve as a point of comparison with the aerospace sector examples presented/raised by the other delegates.

The presentation introduced the CRYSTAL project (Critical SYSTem Engineering AcceLeration) which is researching the problem of tool-chain integration, primarily in support of the automotive sector although aerospace companies are also involved in the project.

Industrial practices and standards for design assurance in the automotive sector are years behind their aviation counterparts in terms of maturity. In the past, requirements have been managed by use of structured text tools, such as HTML, Wiki, or Framemaker. Test plans were held in a bespoke database that had to be extended for requirements traceability. A diverse range of tools have been used, with interoperability achieved by means of XML schema.

In the past, design assurance standards have sometimes been applied rather selectively by manufacturers, to suit their capabilities and weaknesses. CRYSTAL is developing formats, specifications and prototype tools for improved design assurance to support compliance with ISO 26262, which is transforming the automotive industry.

A copy of the presentation is provided in Annex B, and some additional PDF files reviewing the CRYSTAL project are available.
Round-table Discussion – Current Issues and Future Requirements

The afternoon session was reserved for a round-table discussion in which the delegates could express their most pressing concerns regarding RTM tools, and specify their preferences for features that may not exist in tools currently available on the market. The results were compiled into the following table:

<table>
<thead>
<tr>
<th>Item</th>
<th>Issue</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Report Generation</td>
<td>“How do you use the evidence if you don’t have the tool?” – Report generation from the central database, views, etc.</td>
</tr>
</tbody>
</table>
| 2.   | Version Management | • Keeping track of versions – what is being done on which version at any given time?  
• Merging vs. locking strategies – breaking before branching in software companies, to solve the control of versions – use of ‘streams’ to manage multiple versions (too flexible for SCS?)  
• Partially a CM problem...?  
For audit purposes → reproduce the proof/trail |
| 3.   | Identifying Toolsets | What tools are being used?:  
RTM: DOORS, Cradle, PTC Integrity, Reqtify, Vector Gateway  
CM: All Change, PCMS, PVCS, Seapine ALM, IBM Clearcase, Documentum, CMS, Lifespan, SVN  
Design: Artisan, Matlab/Simulink, IBM Rational, Auto-coding from Matlab with Rhapsody  
Safety: Medini Analyse, Isograph Fault Tree+, RWB, Adelard ASCE, Cassandra |
| 4.   | Correctness Checks on Requirements | • Natural language parsing, for searching etc.  
• Requires specification of semi-formal language rules, but if done could be used to perform correctness checks on requirements  
• More of a “Nice to have” feature: technically difficult |
| 5.   | Test Coverage | • Mapping of requirements to tests is often difficult due to lack of tool integration  
• Development of tests in close association to requirements is highly desirable  
• Changes to tests (e.g. due to improvements/updates to test tools) are often difficult to track back to the requirements to identify the degree of regression testing that is required – impact analysis issue; automatic update if tests are changed would be desirable  
• A natural test environment to do mapping of requirements to tests (as these are written by hand, a machine makes them, etc.), more easily and automatically would be desirable |
<table>
<thead>
<tr>
<th>Item</th>
<th>Issue</th>
<th>Remarks</th>
</tr>
</thead>
</table>
| 6.   | Management of evidence | • Management of evidence, especially weight of evidence available to support different requirements / arguments / coverage goals  
• Measurement metrics for how under- or over-verified/specification a given requirement might be  
• Structural measurement metrics for ‘gearing’ or ‘fan-out’ of high-level reqs to low-level reqs, etc.  
• Visual representations (tree views etc.) may be a useful feature. Current tools (DOORS, Rectify) do not quite do this the right way, perhaps improvements can be made |
| 7.   | Overlap between RTM and CM | • May lead to unnecessary duplication of information  
• Commercial toolsets tend to duplicate functions because both types of tool try to do RTM and CM  
• The interoperability specifications/interfaces being developed in the CRYSTAL project are intended to help with this |
| 8.   | Tool information interchange / interoperability: | If no one tool does the complete job, how can information be shared?  
Three levels:  
• direct connection with database  
• exporting into an importable format  
• export, then translate before importing (hard work)  
• pre-process before exporting (really hard work – may govern how you define/structure the data in the first place) |
| 9.   | Tool qualification | If the RTM/CM tool can corrupt the results of tests (or other safety evidence) or give false positives then qualification of the tool is required to DO-178C |
| 10.  | Licensing mechanisms | Variable/flexible licensing is desirable to prevent persons being prevented from use during peak-workload periods on a project |
| 11.  | User Process Models | The problem may not actually be with the RTM/CM tools, it may be with the lack of common meta-models/process models/ontologies and the tool users structuring these items  
- often the tools are configured directly by end-users not support departments or suppliers and hence are used very inefficiently  
- achieving a consensus among users about process models etc. is often as important as the features of the tool |
<p>| 12.  | Interaction between process and product in design assurance | Requirements for design assurance are on the process not just the product – the development process itself is part of the requirement set; some issue, e.g. partitioning can cross the boundary between process and product and cause major problems if not satisfied; many projects have been lazy about specifying process requirements (tend to just specify the DAL without considering the implications) |</p>
<table>
<thead>
<tr>
<th>Item</th>
<th>Issue</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.</td>
<td>Implicit traceability</td>
<td>Tracking of implicit cross-connections between requirements, e.g. timing requirements, or where they have been defined as being related (e.g. UI reqs), or where historically groups of requirements have been changed at the same time (i.e. RTM tool learns which groups are related), or non-atomic requirements that get split out into atomic requirements (i.e. refinement at the same design level)</td>
</tr>
<tr>
<td>14.</td>
<td>Use of open data formats for interoperability</td>
<td>Use of open data formats as much as possible is encouraged, and RTM/CM tools should be able to store/manage/make available any such data, to help integration with other tools and to allow flexibility in changing project processes without losing the ability to support projects with existing tools.</td>
</tr>
</tbody>
</table>

This table forms the primary output of the workshop, as it contains ideas for new features of RTM Tools that can be assessed and developed within the asureSign Aero project. This information will be carried forward into the second workshop to be undertaken in WP3.
Annex A: Copy of Initial Workshop Presentation

asureSign Aero (NATEP Grant MA005)

WP2 Workshop:
Identification of Needs for Tool Support in Meeting Aircraft Avionics Systems, Hardware & Software Certification Standards

Dr Chris Harper
Systems & Safety Assurance Consultant
representing University of Bristol

Workshop Agenda

10.00am Introduction
   – asureSign Aero NATEP Project (TVS)
   – Project Workshop Programme (UoB)
   – System, HW & SW Assurance Standards (UoB)

11:00 Break

11:15 Examples of Problems with Tool-chain Integration
   – Rolls Royce
   – TVS

12:00 Lunch

After Lunch: Round Table Discussions – current issues and future requirements
Overview of asureSign (TVS)
**National Aerospace Technology Exploitation Programme (NATEP)**

- Aimed at small and medium sized aerospace supply chain companies
- Support for development of their own innovative technologies & technology management capabilities
- £40m programme over the next 3 years
- **Objective:** to deliver 100 novel technologies through collaborative projects within the UK aerospace supply chain

- Managed by ADS & regional aerospace alliances
- Sponsored by the UK aerospace primes and Tier 1’s
- Funded by the government’s Advanced Manufacturing Supply Chain Initiative (AMSCI)
- URL: [http://www.natep.org.uk/](http://www.natep.org.uk/)

---

**asureSign Aero (NATEP Grant MA005)**

- **Objective:** to explore whether the tool asureSign, developed originally for requirements traceability management for hardware in the Automotive industry, could be extended to provide better support for Aerospace projects
- **Project Team:**
  - TVS – project lead – developers of asureSign
  - University of Bristol – team member
  - Rolls Royce – project partner
- Support for compliance with DO-178C, DO-254 and ARP4754 is seen as essential
- Experience of practical issues with RTM is sought from aerospace companies
Workshop Programme

- **Workshop #1:**
  Identification of Needs for Tool Support in Meeting Aircraft Avionics Systems, Hardware & Software Certification Standards
  - Perform small survey of experience from aerospace companies about problems and challenges in RTM, based on achieving compliance with standards
  - Identify the principal problems and issues that companies may have

- **Workshop #2:**
  Identification of Problems Addressable by asureSign Software
  - Review capabilities of asureSign to address the identified problems
  - Identify possible new requirements and design changes

ARP 4754  Certification Considerations for Highly Integrated or Complex Aircraft Systems

- Provides guidance on certification issues for complex aircraft systems (primarily electronic)
- “Parent” standard of DO-178B and DO-254, covering certification guidance for the complete system design integrating all E/EE/PE & Mech. Subsystems
- Complementary to guidelines for safety assessment in ARP 4761
ARP 4754 Process Model

ARP 4754 and Traceability Management

Traceability is discussed in ARP 4754 with respect to:

- **System Development**
  - Flow-down: A/C → System → Item → HW/SW
  - Derived requirements flow up

- **Validation**
  - Traceability of validation checks to source requirements (completeness & correctness checking)
  - Traceability of assumptions (to supporting data)
  - Traceability of requirements to non-requirement sources (design decisions, standards, data)
  - Validation matrix is a method of traceability

- **Verification**
  - Requirements coverage analysis
  - Verification matrix is a method of traceability

- **Safety Assessment**
  - Flow-down: FHA → A/C or Sys Fns.
  - Tracking of safety analysis of derived reqs.

- **Configuration Management**
  - Traceability between successive configuration baselines (in terms of the changes applied)
Software Lifecycles in DO-178C

- **DO-178C** identifies a set of basic software development process tasks:
  - Requirements
  - Design
  - Coding
  - Integration

- But it is flexible on how they are combined into an overall software development lifecycle.

---

DO-178C Software Considerations in Airborne Systems and Equipment Certification

Traceability is discussed in DO-178C with respect to:

- **Information Flow between Software Processes and System Processes**
  - Flow-down: System → SW and Safety → SW
  - Derived requirements flow up to System Design & Safety Assessment

- **Software Development (Requirements, Design, Coding)**
  - Downwards traceability: HL Reqs → SW Arch → LL Reqs → Source Code
  - Upwards traceability:
    - To give visibility to the derived reqs & architectural design decisions
    - To verify the complete implementation of all high-level requirements
    - To detect dead or undocumented source code (deactivated code → structural coverage analysis)

- **Verification**
  - Requirements coverage analysis: [HL-reqs/LL-reqs/Code] to test cases and results
  - Traceability must be checkable by review

- **Configuration Management**
  - Traceability data must be associated with defined Configuration Items
  - Traceability between successive configuration baselines (in terms of the changes applied to Configuration Items)
  - Traceability must be established either to a process or its output document (depending on context)

- **Change Control**
  - Software changes should be traced to their origin and the software life cycle processes repeated from the point at which the change affects their outputs.
The DO-254 HW Lifecycle model assumes a basic linear pattern, with supporting processes:

- Planning
- Requirements Capture
- Conceptual Design
- Detailed Design
- Implementation
- Production Transition
- Manufacturing Process
- System Development Process

Supporting Processes:
- Verification & Validation
- Configuration Management
- Process Assurance
- Certification Liaison

Most DO-254 guidance on traceability processes for HW is the same as the guidance in DO-178C for SW.

However, there are a few specific differences:
- Detailed Hardware Design: It is not intended to require traceability to detailed components, such as resistors, capacitors or gates, unless required for safety considerations.
- Functional Failure Path Analysis: Traces must be established between the FFPs and associated hardware requirements and derived requirements.
- Product Service Experience: Traceability data is required showing the explicit relationship of the service experience data and verification that provides design assurance coverage of each FFP.
- Elemental Analysis & Safety-specific HW Analysis: traceability data must show the explicit relationship of the verification procedures to the elements in the analysis.
- Formal Methods: Traceability data shall correlate proofs or proof scripts with component models and their associated formal statement of requirements and parent requirements.
Recent Experience: Use of Traceability Tools for Safety Management

- **Use of RTM Tools for unconventional purposes:**
  Safety management documents: hazard logs, safety analyses, safety cases (argument-evidence traceability)

- **Maintaining consistency in diverse documents:** Most RTM tools identify links, but do not update information (which must be done manually)

- **Capture of information from specialised analysis tools:**
  - Safety analysis toolsets, e.g. fault trees, RBDs, FMEA tables
  - Sometimes the analysis is actually documented within the RTM tool

For the rest of the day…

- **Example problems:**
  Case studies from project partners

- **Lunch**

- **Discussions:**
  Delegates are invited to share their experiences and issues with RTM and the associated tools
Welcome to asureSign™ Roadmap

Serrie-justine Chapman
Requirements Engineering Consultant
Test and Verification Solutions Ltd

FIRST REQ ENGINEERING SOLUTION

Requirements Database

Product Requirement Document

Internal Target Specification

Safety Concept

Test Plan

logos
username
defense
test
**CHALLENGES**

- Requirement interpretation may change through flow
- Manually mapping of documents with identifiers is time-consuming and subject to error
- Translation or moving of data may cause corruption or error
- Link to proofs of implementation/results is manual
- Visibility of Requirements through the entire tree is complex
- Communication across domains (pre-silicon/post-silicon) complex

**Under Verification:**
- Requirements not implemented
- Incorrect testing due to miscomprehension
- Re-spin due to late realisation

**Over Verification:**
- Resource cost
- Duplication of effort
- Incorrect product (size/speed etc)

**TIGHTLY INTEGRATED RE & TEST**

![Diagram](image-url)
AsureSgn™

Allows Requirements driven Verification methodology
Requirements can be imported from external tool: ‘top-level test plan’
Requirements can be refined and matched to goals: ‘bottom-level test plan’
Mapping of the Requirements to test plan and results
Accumulates data over a period of time
- Analyses and presents the status of tests
- Collects and presents coverage metrics
Works across multiple test types (random, formal, directed, manual, assertions etc)
Control of pass/fail criteria on a per milestone/per group level
Allows easy control and evaluation of the progress of the project.

MAPPING GOALS TO COVERAGE OR TESTS
COMPLIANCE:
HIERARCHICAL SET OF REQUIREMENTS

REFINING THE REQUIREMENTS TO TEST
DESCRIPTION LEVEL

Measurable goals

- Goal1.1.1
- Goal1.1.1.2
- Goal1.2.1
- Goal1.2.1.1

Refined requirements (sub-features and goals)

- Req1.1.1
- Req1.1.2
- Req1.2.1
- Goal1.2.2

Refined requirements (sub-features)

- Req1.1
- Req1.2

Feature Level
Requirements
(Top-Level test Plan)
MAPPING GOALS TO COVERAGE OR TESTS

- Each goal can be mapped to one or more metrics (and each metric to one or many goals).
- During the lifecycle of the project, goals relating to milestones may change their percentage coverage required for a pass.
- Assigning milestones to goals allows management of early releases, easily identifying maturity of the project and identifying resource or time risks.
- Unmapped goals = test plan with no test.
  - New test/metric required.
- Unmapped metric = test but no plan or requirements.
  - Implicit or explicit requirement missing for requirements list (completeness).
  - Test/metric no longer needed – saves maintenance.
- Obsolete features and goals identify variant differences.
  - Don't run those mapped metrics/tests for this variant (saves regression size/runs and debug).
asureSgnTM captures all the metrics and test data based on a unique Regression.

asureSgnTM can generate a unique identifier for a regression based on:
• A Version Control System
• A Time Stamp

**Version Control System (VCS):**
Regressions can be generated based on the state of VCS. This identifies the version (config/label/tag) of the code against which the tests were run and can also track the changes between two versions of source code.

Supported version control systems are:
• Git
• SVN
• Clearcase

**Timestamp:** Regressions can also be generated based on the current timestamp. This is helpful if there is no VCS or the user doesn’t wish to link the results to any VCS version.

1. Proper verification steps are always executed with each regression.
2. Verification is performed strictly on the basis of Requirements, thus both Over verification and Under verification are properly handled.
3. Project progress graph and milestones are the best indicator of how project is progressing, and both Engineers and Managers can benefit from it.
4. Release dates can be properly identified.
PROJECT HISTORY GRAPH

- Is a key indicator of the overall progress of the project.
- Any dip or peaks indicate debugging and corrective actions may be required.

UNMAPPED GOALS EXAMPLE

Every section on the analyzer window is informative and each section can play an important role in identifying issues:

- Unmapped metric = test but no plan or requirements.
  - Implicit or explicit requirement missing for requirements list (completeness)
  - Test/metric no longer needed – saves maintenance
REQUIREMENTS HIERARCHY

- A pass criteria can be assigned by milestone, project, feature, sub-feature or goal
- Pass criteria may be related to the importance of test (Safety level)
- Keeping the hierarchy within one tool simplifies Requirements Traceability
**MILESTONES**

Milestones are **user defined significant events** in the course of a project and are key indicator to **identify the maturity of the project**. Various milestones can be created e.g. alpha ready, beta ready etc.

For early releases from one group (sub-system -> system) this can assist with communication – features passing/failing at each milestone.

---

**REGRESSION COMPARISON**

Regression comparison helps debug and identify differences over time.
OTHER FEATURES

- Multiple user access to databases
- Supports UCIS standard (multiple tools)
- Supports locking of requirements for protection
- Tailored Report generation for audit, with description for goals (test plan within the tool and easily updated)
- Supports batch and bulk operations.
- XML support allows easy integration with other tooling
- Features and goals may be edited, manipulated and enhanced inside the tool.
- Internal comments allowed to help communication within a project
- Requirements can be signed-off manually – if by manual inspection e.g. waveform

Variant Management
Requirements completeness

Change management

Requirements Database

Variant x
xml

Variant x
Target Spec

Domain Analyser Dashboard

<table>
<thead>
<tr>
<th>Req</th>
<th>Pre-silicon IP</th>
<th>Pre-silicon SOC</th>
<th>Validation IP</th>
<th>Validation SOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req_x</td>
<td>pass</td>
<td>No</td>
<td>fail</td>
<td>No</td>
</tr>
<tr>
<td>Req_y</td>
<td>no</td>
<td>pass</td>
<td>No</td>
<td>pass</td>
</tr>
</tbody>
</table>

API Interface

Pre-silicon IP
Pre-silicon SOC
Validation IP
Validation SOC
Firmware, software etc
## Dashboard Information

<table>
<thead>
<tr>
<th>Hierarchy</th>
<th>Variant</th>
<th>System</th>
<th>IP</th>
<th>SOC</th>
<th>IP Val</th>
<th>SOC Val</th>
<th>Test</th>
<th>SW</th>
<th>Requirements Traceability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variant 1</td>
<td>System</td>
<td>4 x moduleB</td>
<td><a href="mailto:soc1@tvs.com">soc1@tvs.com</a></td>
<td><a href="mailto:val1@tvs.com">val1@tvs.com</a></td>
<td>Pass</td>
<td>pass</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variant 2</td>
<td>System</td>
<td>3 x moduleB</td>
<td><a href="mailto:soc1@tvs.com">soc1@tvs.com</a></td>
<td><a href="mailto:val1@tvs.com">val1@tvs.com</a></td>
<td>bronze</td>
<td>Gold</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variant 3</td>
<td>moduleA</td>
<td>Feature A</td>
<td><a href="mailto:ip1@tvs.com">ip1@tvs.com</a></td>
<td><a href="mailto:soc2@tvs.com">soc2@tvs.com</a></td>
<td><a href="mailto:iv1@tvs.com">iv1@tvs.com</a></td>
<td><a href="mailto:val2@tvs.com">val2@tvs.com</a></td>
<td>pass</td>
<td>pass</td>
<td>pass</td>
</tr>
<tr>
<td>Variant 3</td>
<td>moduleB</td>
<td>Feature X</td>
<td><a href="mailto:ip2@tvs.com">ip2@tvs.com</a></td>
<td><a href="mailto:iv2@tvs.com">iv2@tvs.com</a></td>
<td>pass</td>
<td>pass</td>
<td>gold</td>
<td>bronze</td>
<td></td>
</tr>
<tr>
<td>Variant 3</td>
<td>moduleC</td>
<td>Feature Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Asil D?**

- pass
- pass
- pass
- fail

**Why?**

**Not tested?**

**Measure Of Metrics**

- **Same as Dashboard but for quick verification analysis**
  - All History Graphs
  - All History Tables
  - Change management issues (from Jira)
  - Coverage overview per ‘Goal’ (test requirement)
- **Replaces in-depth weekly reports**
CONTACT:

For further information and questions please contact:

Serrie-Justine Chapman  
TVS Requirements Engineering Consultant  
Mail: serrie@testandverification.com  
Tel: ++44 (0) 7854 785747

TVS http://testandverification.com/  
Engine Shed  
Station Approach  
Temple Meads  
Bristol  
BS1 6QH  
United Kingdom  
T: +44 (0)117 903 1100  
F: +44 (0)117 903 9001