LDRA 50 1975–2025

A TASKING Company

Developing Safe Software for Autonomous Systems

Alex Lim - Principal Field Application Engineer and Multi-core Lead

Speaker

Principal field application engineer and Multicore Lead, LDRA

Alex Lim

Alex is a Principal Field Application engineer, Multi-core lead at LDRA. Over the years he has worked closely with industry leading companies in automotive, aerospace, and other safety critical domains. Alex has driven innovative solutions with LDRA's customers in ADAS systems, helped real-time operating system vendors and silicon vendors achieve safety and security goals, and worked closely with avionics suppliers to meet the latest standards. Alex also works with LDRA customers and distributors to bring LDRA solutions and international safety and security standards to emerging markets. He represents LDRA on industry bodies and has delivered presentations at numerous events including several autonomous vehicles conferences, Digital Avionics Systems Conferences, and Future Airborne Capabilities Environment Technical Interchange meetings. Over the course of his career, Alex has worked at the Space and Missiles System Center on safety and mission critical Flight Safety and Flight Management software, and designed autopilot simulations for UAVs.

https://autonomousvehicletechnologyexpo-usa.com/speakers/alex-lim-1?&searchTerm=803&filters.conference_id=__hasValue&sortby=personCompany%20asc&searchgroup=libraryentry-speakers

Automotive Industry Trends

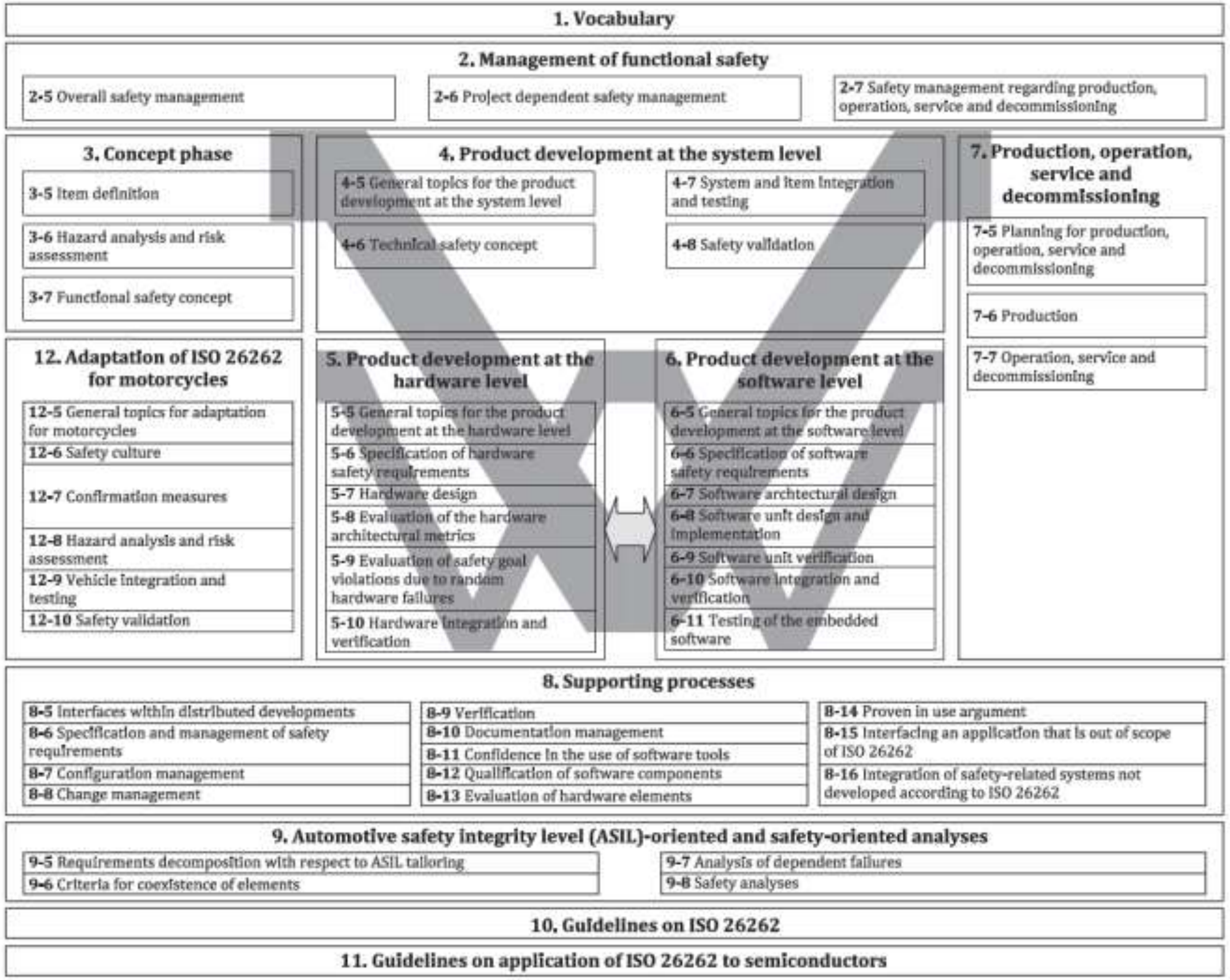
- Innovation is causing market disruption
  - Increasingly, a product's DNA is its software
  - Innovation requires learning fast, deciding fast, acting fast, delivering fast
- Products are becoming part of connected IoT solutions
  - More partners, more standards, more interfaces, more emergent behavior
  - Inherently more failure modes, including OTA update failures
- Products are becoming much more autonomous
  - More software, more technology, more 'intelligent'
  - Advanced driver-assistance systems: lane departure warning, blind spot monitoring, adaptive cruise control, automatic parking, collision avoidance
- The global advanced driver-assistance system (ADAS) market is expected to grow 19% annually, reaching USD 67.4B by 2025 - Grand View Research
  - https://www.grandviewresearch.com/press-release/global-advanced-driver-assistance-systems-adas-market February 2018

Agenda

- Automotive Industry Trends
- Challenges to Achieving Software Compliance
- Overcoming the Challenges

Goals of ISO 26262

Meeting the goals of ISO 26262 for ADAS systems with their high level of complexity and high safety integrity level is a huge challenge!



1. Vocabulary

2. Management of functional safety

2-5 Overall safety management

2-6 Project dependent safety management

2-7 Safety management regarding production, operation, service and decommissioning

3. Concept phase

3-5 Item definition

3-6 Hazard analysis and risk assessment

3-7 Functional safety concept

4. Product development at the system level

4-5 General topics for the product development at the system level

4-6 Technical safety concept

4-7 System and item integration and testing

4-8 Safety validation

7. Production, operation, service and decommissioning

7-5 Planning for production, operation, service and decommissioning

7-6 Production

7-7 Operation, service and decommissioning

12. Adaptation of ISO 26262 for motorcycles

12-5 General topics for adaptation for motorcycles

12-6 Safety culture

12-7 Confirmation measures

12-8 Hazard analysis and risk assessment

12-9 Vehicle integration and testing

12-10 Safety validation

5. Product development at the hardware level

5-5 General topics for the product development at the hardware level

5-6 Specification of hardware safety requirements

5-7 Hardware design

5-8 Evaluation of the hardware architectural metrics

5-9 Evaluation of safety goal violations due to random hardware failures

5-10 Hardware integration and verification

6. Product development at the software level

6-5 General topics for the product development at the software level

6-6 Specification of software safety requirements

6-7 Software architectural design

6-8 Software unit design and implementation

6-9 Software unit verification

6-10 Software integration and verification

6-11 Testing of the embedded software

8. Supporting processes

8-5 Interfaces within distributed developments

8-6 Specification and management of safety requirements

8-7 Configuration management

8-8 Change management

8-9 Verification

8-10 Documentation management

8-11 Confidence in the use of software tools

8-12 Qualification of software components

8-13 Evaluation of hardware elements

8-14 Proven in use argument

8-15 Interfacing an application that is out of scope of ISO 26262

8-16 Integration of safety-related systems not developed according to ISO 26262

9. Automotive safety integrity level (ASIL)-oriented and safety-oriented analyses

9-5 Requirements decomposition with respect to ASIL tailoring

9-6 Criteria for coexistence of elements

9-7 Analysis of dependent failures

9-8 Safety analyses

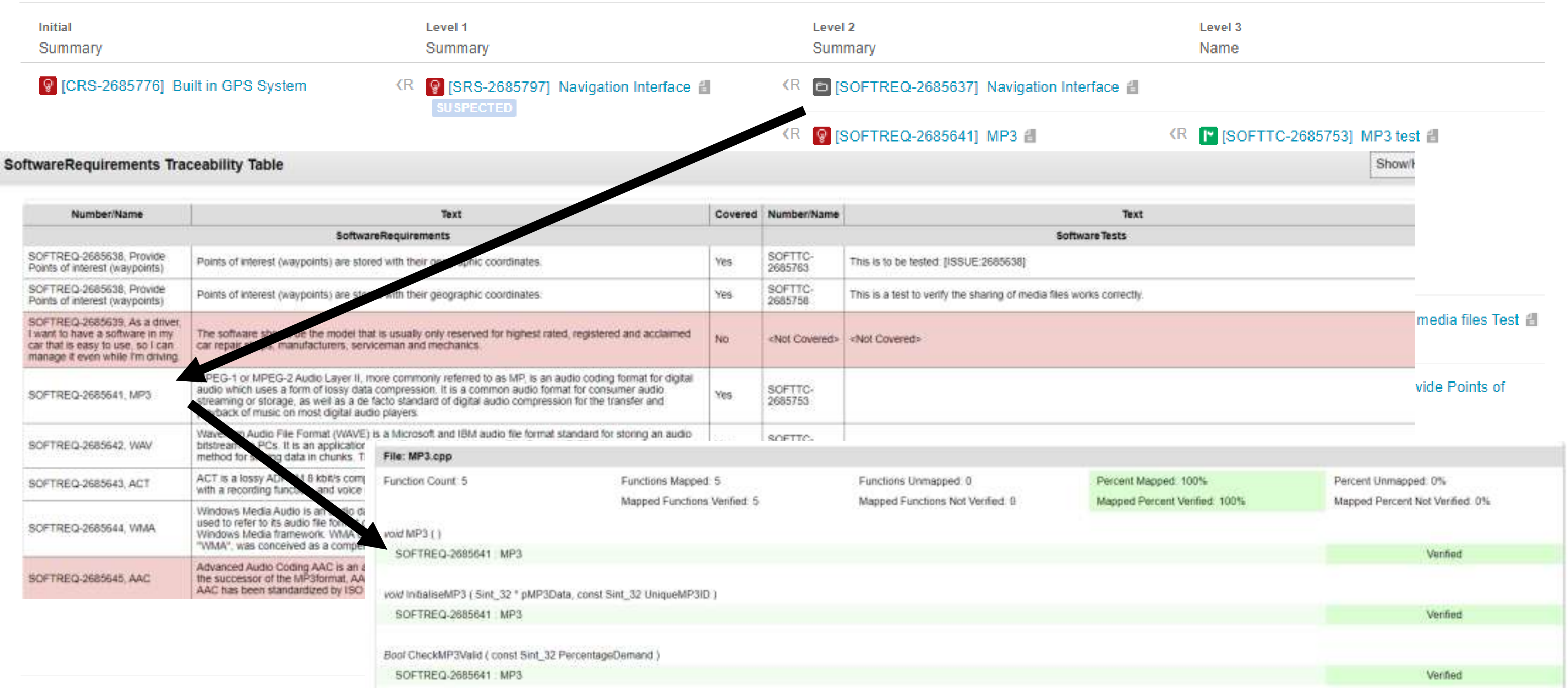
10. Guidelines on ISO 26262

11. Guidelines on application of ISO 26262 to semiconductors

Essential Capabilities for Overcoming Software Compliance Challenges

- End to End Traceability and Transparency
- Insightful Impact Analysis and Change Management
- Cross Discipline Collaboration
- Rapid Iterative Development
- Efficient and Predictable Tool Qualification and Software Certification

Traceability and Transparency
Requirements – Code



Initial Summary		Level 1 Summary		Level 2 Summary		Level 3 Name
[CRS-2685776] Built in GPS System	‹R	[SRS-2685797] Navigation Interface SUSPECTED	‹R	[SOFTREQ-2685637] Navigation Interface		
			‹R	[SOFTREQ-2685641] MP3	‹R	[SOFTTC-2685753] MP3 test
						media files Test
						vide Points of

SoftwareRequirements Traceability Table

Number/Name	Text	Covered	Number/Name	Text
SoftwareRequirements			Software Tests	
SOFTREQ-2685638, Provide Points of interest (waypoints)	Points of interest (waypoints) are stored with their ge phic coordinates.	Yes	SOFTTC-2685763	This is to be tested: [ISSUE:2685638]
SOFTREQ-2685638, Provide Points of interest (waypoints)	Points of interest (waypoints) are st with their geographic coordinates.	Yes	SOFTTC-2685758	This is a test to verify the sharing of media files works correctly.
SOFTREQ-2685639, As a driver, I want to have a software in my car that is easy to use, so I can manage it even while I'm driving.	The software sh be the model that is usually only reserved for highest rated, registered and acclaimed car repair , manufacturers, serviceman and mechanics.	No	<Not Covered>	<Not Covered>
SOFTREQ-2685641, MP3	PEG-1 or MPEG-2 Audio Layer II, more commonly referred to as MP, is an audio coding format for digital audio which uses a form of lossy data compression. It is a common audio format for consumer audio streaming or storage, as well as a de facto standard of digital audio compression for the transfer and back of music on most digital audio players.	Yes	SOFTTC-2685753	
SOFTREQ-2685642, WAV	Wave Audio File Format (WAVE) is a Microsoft and IBM audio file format standard for storing an audio bitstream PCs. It is an application method for ng data in chunks. T		SOFTTC-	
SOFTREQ-2685643, ACT	ACT is a lossy AD 8 kbit/s com with a recording func and voice			
SOFTREQ-2685644, WMA	Windows Media Audio is an dio d used to refer to its audio file fo Windows Media framework. WMA "WMA", was conceived as a compe			
SOFTREQ-2685645, AAC	Advanced Audio Coding AAC is an the successor of the MP3format, AA AAC has been standardized by ISO			

File: MP3.cpp

Function Count: 5	Functions Mapped: 5	Functions Unmapped: 0	Percent Mapped: 100%	Percent Unmapped: 0%
	Mapped Functions Verified: 5	Mapped Functions Not Verified: 0	Mapped Percent Verified: 100%	Mapped Percent Not Verified: 0%

void MP3 ( )

SOFTREQ-2685641 : MP3	Verified

void InitialiseMP3 (Sint_32 * pMP3Data, const Sint_32 UniqueMP3ID)

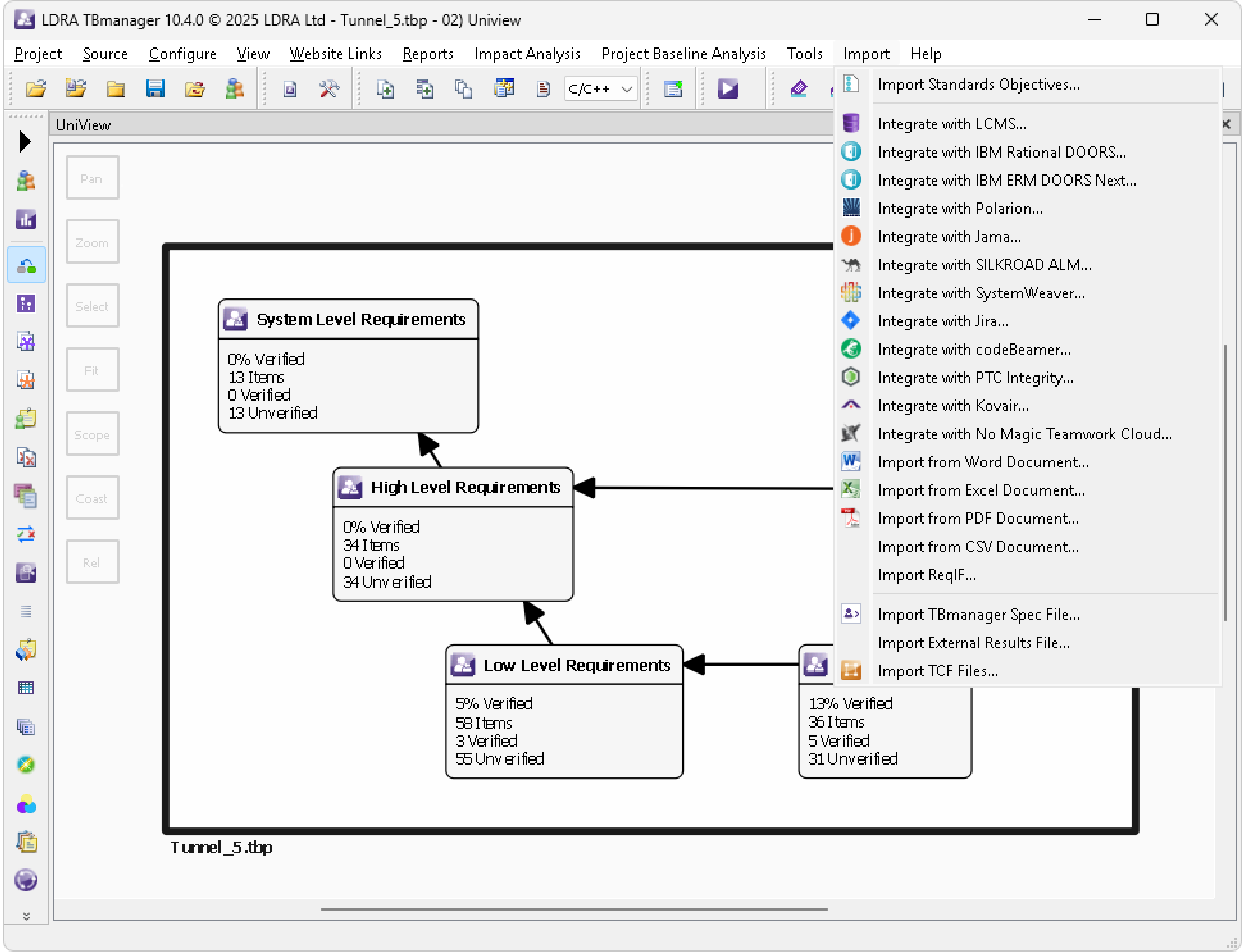
SOFTREQ-2685641 : MP3	Verified

Bool CheckMP3Valid (const Sint_32 PercentageDemand)

SOFTREQ-2685641 : MP3	Verified

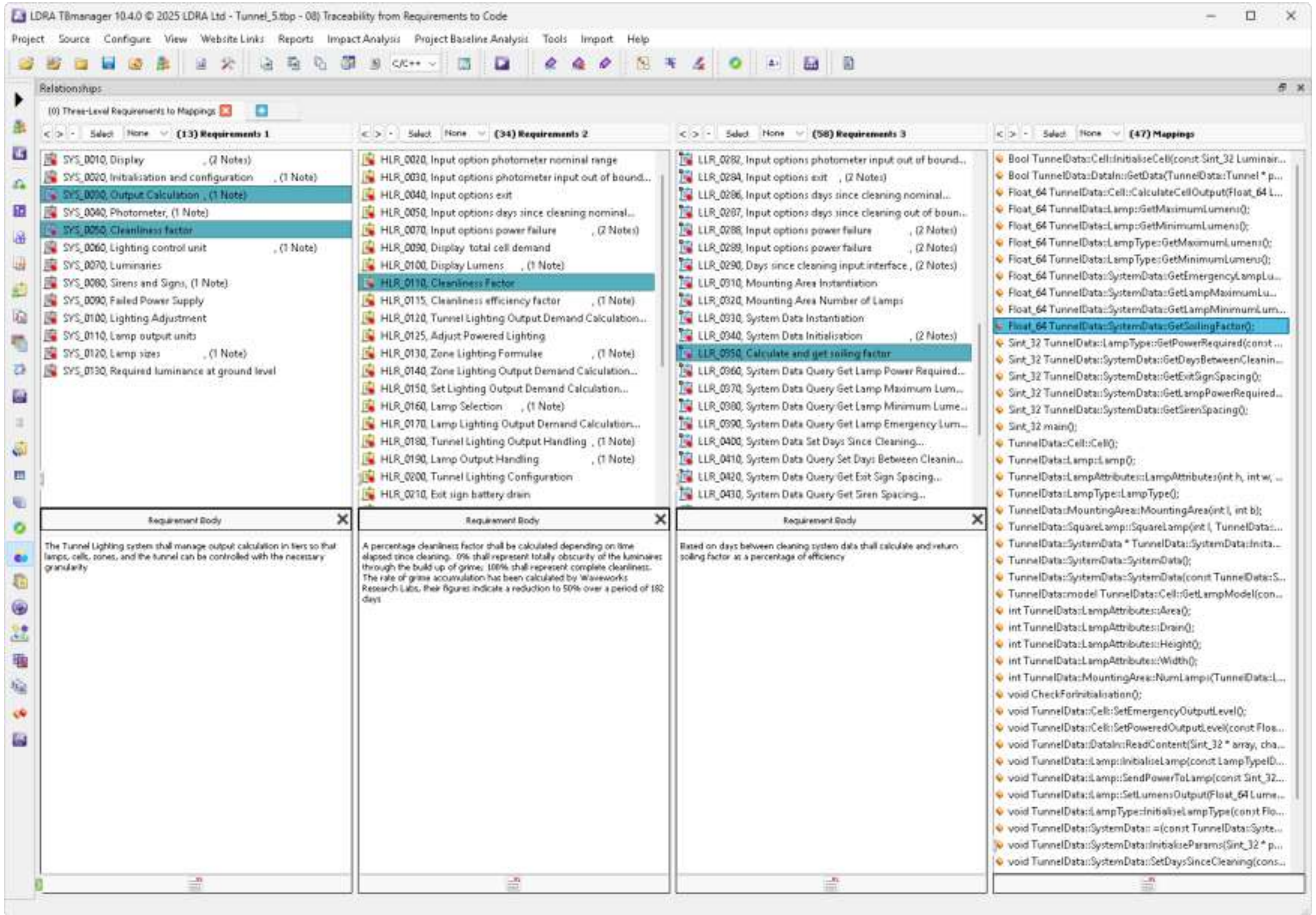
Traceability in TBmanager with ALM tools

- Integration with many ALM tools
  - Also word, excel, pdf, csv, ReqIF


LDRA TBmanager 10.4.0 © 2025 LDRA Ltd - Tunnel_5.tbp - 02) Uniview
Project
Source
Configure
View
Website Links
Reports
Impact Analysis
Project Baseline Analysis
Tools
Import
Help
C/C++
UniView
Pan
Zoom
Select
Fit
Scope
Coast
Rel
Import Standards Objectives...
Integrate with LCMS...
Integrate with IBM Rational DOORS...
Integrate with IBM ERM DOORS Next...
Integrate with Polarion...
Integrate with Jama...
Integrate with SILKROAD ALM...
Integrate with SystemWeaver...
Integrate with Jira...
Integrate with codeBeamer...
Integrate with PTC Integrity...
Integrate with Kovair...
Integrate with No Magic Teamwork Cloud...
Import from Word Document...
Import from Excel Document...
Import from PDF Document...
Import from CSV Document...
Import ReqIF...
Import TBmanager Spec File...
Import External Results File...
Import TCF Files...
System Level Requirements
0% Verified
13 Items
0 Verified
13 Unverified
High Level Requirements
0% Verified
34 Items
0 Verified
34 Unverified
Low Level Requirements
5% Verified
58 Items
3 Verified
55 Unverified
13% Verified
36 Items
5 Verified
31 Unverified
Tunnel_5.tbp

Traceability

- What is the big advantage with LDRA tools.
  - Deep analysis
  - Dynamically trace all the way down to the function level
  - Map tests to requirements



LDRA TBmanager 10.4.0 © 2025 LDRA Ltd - Tunnel_5.tbp - 08) Traceability from Requirements to Code

(13) Requirements 1

```
SYS_0010, Display , (2 Notes)
SYS_0020, Initialisation and configuration , (1 Note)
SYS_0030, Output Calculation , (1 Note)
SYS_0040, Photometer, (1 Note)
SYS_0050, Cleanliness factor
SYS_0060, Lighting control unit , (1 Note)
SYS_0070, Luminaries
SYS_0080, Sirens and Signs, (1 Note)
SYS_0090, Failed Power Supply
SYS_0100, Lighting Adjustment
SYS_0110, Lamp output units
SYS_0120, Lamp sizes , (1 Note)
SYS_0130, Required luminance at ground level
```

Requirement Body

The Tunnel Lighting system shall manage output calculation in tiers so that lamps, cells, zones, and the tunnel can be controlled with the necessary granularity

(34) Requirements 2

```
HLR_0020, Input option photometer nominal range
HLR_0030, Input options photometer input out of bound...
HLR_0040, Input options exit
HLR_0050, Input options days since cleaning nominal...
HLR_0070, Input options power failure , (2 Notes)
HLR_0090, Display total cell demand
HLR_0100, Display Lumens , (1 Note)
HLR_0110, Cleanliness Factor
HLR_0115, Cleanliness efficiency factor , (1 Note)
HLR_0120, Tunnel Lighting Output Demand Calculation...
HLR_0125, Adjust Powered Lighting
HLR_0130, Zone Lighting Formulae , (1 Note)
HLR_0140, Zone Lighting Output Demand Calculation...
HLR_0150, Set Lighting Output Demand Calculation...
HLR_0160, Lamp Selection , (1 Note)
HLR_0170, Lamp Lighting Output Demand Calculation...
HLR_0180, Tunnel Lighting Output Handling , (1 Note)
HLR_0190, Lamp Output Handling , (1 Note)
HLR_0200, Tunnel Lighting Configuration
HLR_0210, Exit sign battery drain
```

Requirement Body

A percentage cleanliness factor shall be calculated depending on time elapsed since cleaning. 0% shall represent totally obscurity of the luminaires through the build up of grime; 100% shall represent complete cleanliness. The rate of grime accumulation has been calculated by Waveworks Research Labs, their figures indicate a reduction to 50% over a period of 182 days

(58) Requirements 3

```
LLR_0282, Input options photometer input out of bound...
LLR_0284, Input options exit , (2 Notes)
LLR_0286, Input options days since cleaning nominal...
LLR_0287, Input options days since cleaning out of boun...
LLR_0288, Input options power failure , (2 Notes)
LLR_0289, Input options power failure , (2 Notes)
LLR_0290, Days since cleaning input interface , (2 Notes)
LLR_0310, Mounting Area Instantiation
LLR_0320, Mounting Area Number of Lamps
LLR_0330, System Data Instantiation
LLR_0340, System Data Initialisation , (2 Notes)
LLR_0350, Calculate and get soiling factor
LLR_0360, System Data Query Get Lamp Power Required...
LLR_0370, System Data Query Get Lamp Maximum Lum...
LLR_0380, System Data Query Get Lamp Minimum Lume...
LLR_0390, System Data Query Get Lamp Emergency Lum...
LLR_0400, System Data Set Days Since Cleaning...
LLR_0410, System Data Query Set Days Between Cleanin...
LLR_0420, System Data Query Get Exit Sign Spacing...
LLR_0430, System Data Query Get Siren Spacing...
```

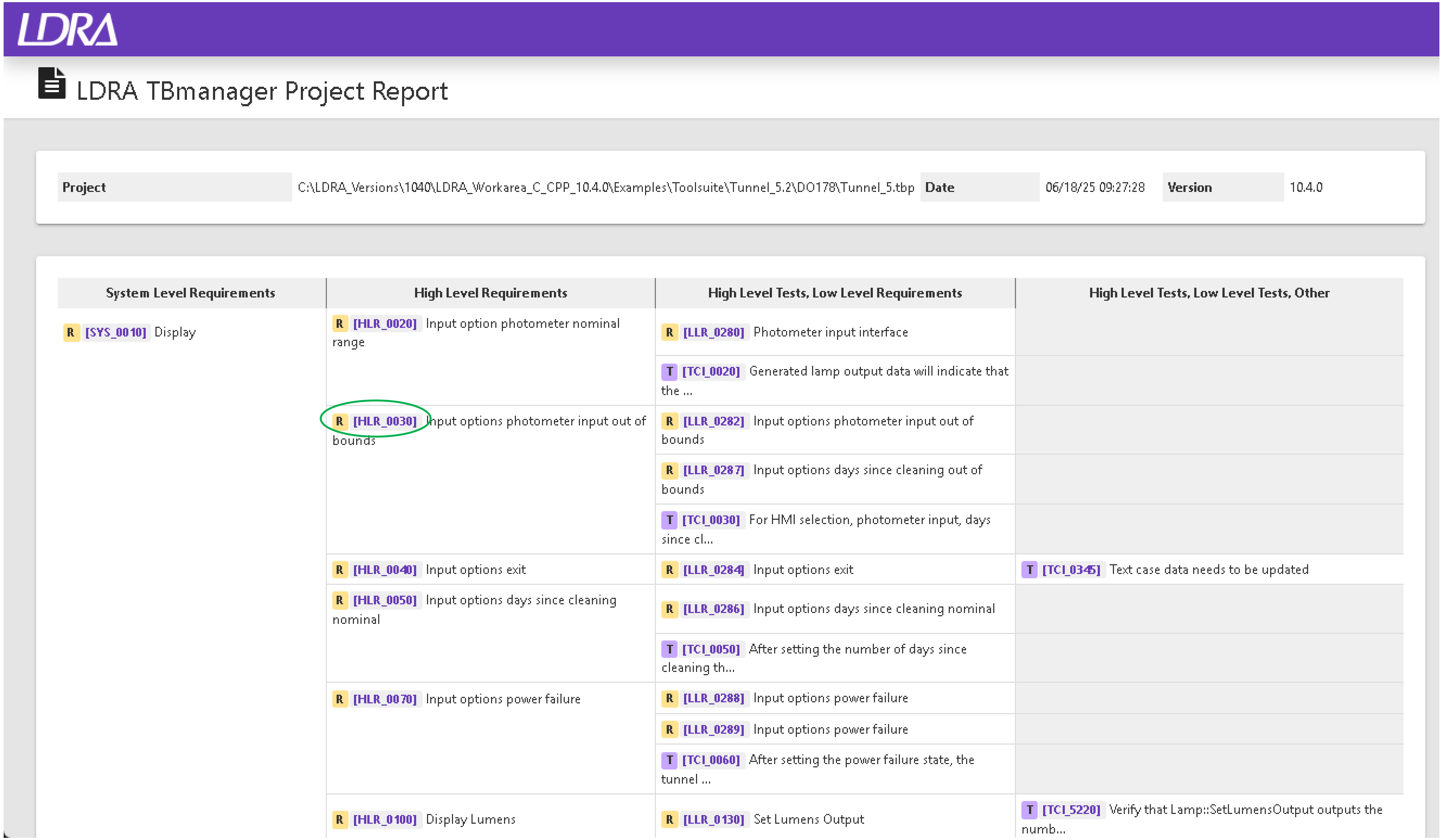
Requirement Body

Based on days between cleaning system data shall calculate and return soiling factor as a percentage of efficiency

(47) Mappings

```
Bool TunnelData::Cell::InitialiseCell(const Sint_32 Luminair...
Bool TunnelData::DataIn::GetData(TunnelData::Tunnel * p...
Float_64 TunnelData::Cell::CalculateCellOutput(Float_64 L...
Float_64 TunnelData::Lamp::GetMaximumLumens();
Float_64 TunnelData::Lamp::GetMinimumLumens();
Float_64 TunnelData::LampType::GetMaximumLumens();
Float_64 TunnelData::LampType::GetMinimumLumens();
Float_64 TunnelData::SystemData::GetEmergencyLampLu...
Float_64 TunnelData::SystemData::GetLampMaximumLu...
Float_64 TunnelData::SystemData::GetLampMinimumLum...
Float_64 TunnelData::SystemData::GetSoilingFactor();
Sint_32 TunnelData::LampType::GetPowerRequired(const ...
Sint_32 TunnelData::SystemData::GetDaysBetweenCleanin...
Sint_32 TunnelData::SystemData::GetExitSignSpacing();
Sint_32 TunnelData::SystemData::GetLampPowerRequired...
Sint_32 TunnelData::SystemData::GetSirenSpacing();
Sint_32 main();
TunnelData::Cell::Cell();
TunnelData::Lamp::Lamp();
TunnelData::LampAttributes::LampAttributes(int h, int w, ...
TunnelData::LampType::LampType();
TunnelData::MountingArea::MountingArea(int l, int b);
TunnelData::SquareLamp::SquareLamp(int l, TunnelData::...
TunnelData::SystemData * TunnelData::SystemData::Inita...
TunnelData::SystemData::SystemData();
TunnelData::SystemData::SystemData(const TunnelData::S...
TunnelData::model TunnelData::Cell::GetLampModel(con...
int TunnelData::LampAttributes::Area();
int TunnelData::LampAttributes::Drain();
int TunnelData::LampAttributes::Height();
int TunnelData::LampAttributes::Width();
int TunnelData::MountingArea::NumLamps(TunnelData::L...
void CheckForInitialisation();
void TunnelData::Cell::SetEmergencyOutputLevel();
void TunnelData::Cell::SetPoweredOutputLevel(const Floa...
void TunnelData::DataIn::ReadContent(Sint_32 * array, cha...
void TunnelData::Lamp::InitialiseLamp(const LampTypeID...
void TunnelData::Lamp::SendPowerToLamp(const Sint_32...
void TunnelData::Lamp::SetLumensOutput(Float_64 Lume...
void TunnelData::LampType::InitialiseLampType(const Flo...
void TunnelData::SystemData:: =(const TunnelData::Syste...
void TunnelData::SystemData::InitialiseParams(Sint_32 * p...
void TunnelData::SystemData::SetDaysSinceCleaning(cons...
```

Full Project Tree Report

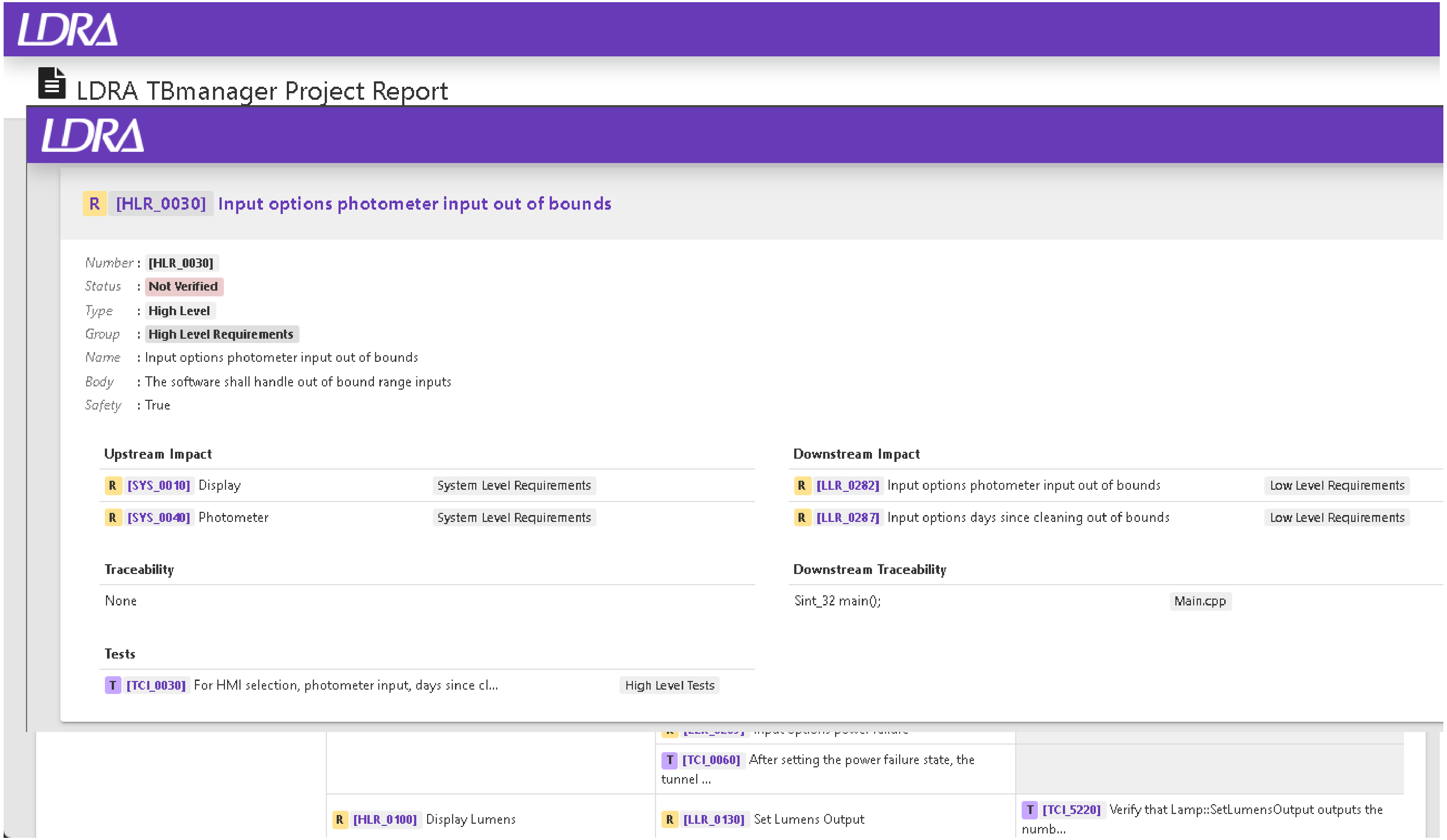


LDRA TBmanager Project Report

Project	C:\LDRA_Versions\1040\LDRA_Workarea_C_CPP_10.4.0\Examples\Toolsuite\Tunnel_5.2\DO178\Tunnel_5.tbp	Date	06/18/25 09:27:28	Version	10.4.0

System Level Requirements	High Level Requirements	High Level Tests, Low Level Requirements	High Level Tests, Low Level Tests, Other
R [SYS_0010] Display	R [HLR_0020] Input option photometer nominal range	R [LLR_0280] Photometer input interface	
		T [TCI_0020] Generated lamp output data will indicate that the ...	
	R [HLR_0030] Input options photometer input out of bounds	R [LLR_0282] Input options photometer input out of bounds	
		R [LLR_0287] Input options days since cleaning out of bounds	
		T [TCI_0030] For HMI selection, photometer input, days since cl...	
	R [HLR_0040] Input options exit	R [LLR_0284] Input options exit	T [TCI_0345] Text case data needs to be updated
	R [HLR_0050] Input options days since cleaning nominal	R [LLR_0286] Input options days since cleaning nominal	
		T [TCI_0050] After setting the number of days since cleaning th...	
	R [HLR_0070] Input options power failure	R [LLR_0288] Input options power failure	
		R [LLR_0289] Input options power failure	
		T [TCI_0060] After setting the power failure state, the tunnel ...	
	R [HLR_0100] Display Lumens	R [LLR_0130] Set Lumens Output	T [TCI_5220] Verify that Lamp::SetLumensOutput outputs the numb...

Full Project Tree Report



LDRA TBmanager Project Report

R [HLR_0030] Input options photometer input out of bounds

Number	: [HLR_0030]
Status	: Not Verified
Type	: High Level
Group	: High Level Requirements
Name	: Input options photometer input out of bounds
Body	: The software shall handle out of bound range inputs
Safety	: True

Upstream Impact

R [SYS_0010] Display	System Level Requirements
R [SYS_0040] Photometer	System Level Requirements

Downstream Impact

R [LLR_0282] Input options photometer input out of bounds	Low Level Requirements
R [LLR_0287] Input options days since cleaning out of bounds	Low Level Requirements

Traceability

None

Downstream Traceability

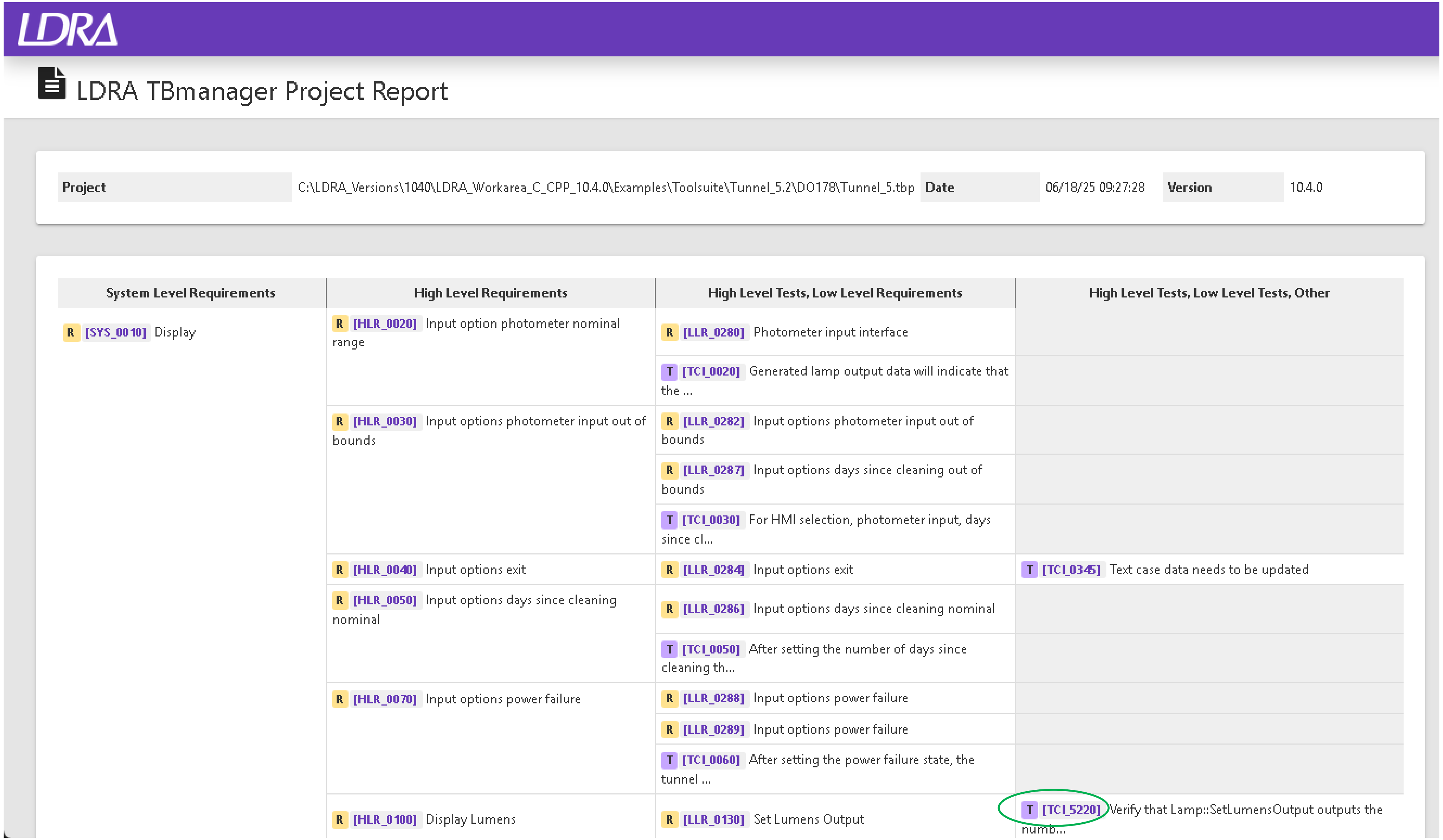
Sint_32 main();	Main.cpp

Tests

T [TCI_0030] For HMI selection, photometer input, days since cl...	High Level Tests

	R [illegible] Input options power failure	
	T [TCI_0060] After setting the power failure state, the tunnel ...	
R [HLR_0100] Display Lumens	R [LLR_0130] Set Lumens Output	T [TCI_5220] Verify that Lamp::SetLumensOutput outputs the numb...

Full Project Tree Report

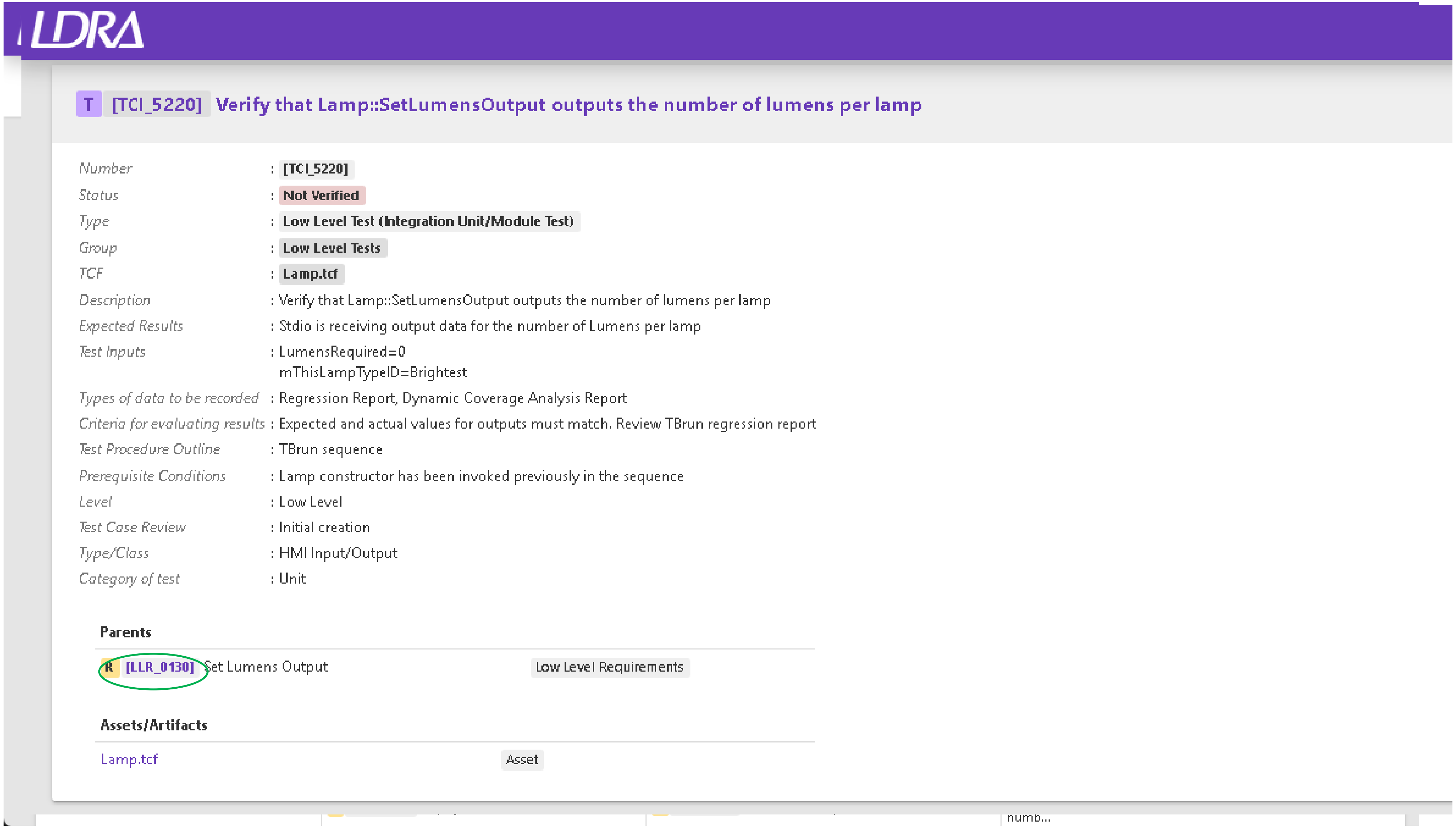


LDRA TBmanager Project Report

Project	C:\LDRA_Versions\1040\LDRA_Workarea_C_CPP_10.4.0\Examples\Toolsuite\Tunnel_5.2\DO178\Tunnel_5.tbp	Date	06/18/25 09:27:28	Version	10.4.0

System Level Requirements	High Level Requirements	High Level Tests, Low Level Requirements	High Level Tests, Low Level Tests, Other
R [SYS_0010] Display	R [HLR_0020] Input option photometer nominal range	R [LLR_0280] Photometer input interface	
		T [TCI_0020] Generated lamp output data will indicate that the ...	
	R [HLR_0030] Input options photometer input out of bounds	R [LLR_0282] Input options photometer input out of bounds	
		R [LLR_0287] Input options days since cleaning out of bounds	
		T [TCI_0030] For HMI selection, photometer input, days since cl...	
	R [HLR_0040] Input options exit	R [LLR_0284] Input options exit	T [TCI_0345] Text case data needs to be updated
	R [HLR_0050] Input options days since cleaning nominal	R [LLR_0286] Input options days since cleaning nominal	
		T [TCI_0050] After setting the number of days since cleaning th...	
	R [HLR_0070] Input options power failure	R [LLR_0288] Input options power failure	
		R [LLR_0289] Input options power failure	
		T [TCI_0060] After setting the power failure state, the tunnel ...	
	R [HLR_0100] Display Lumens	R [LLR_0130] Set Lumens Output	T [TCI_5220] Verify that Lamp::SetLumensOutput outputs the numb...

Full Project Tree Report



T [TCI_5220] Verify that Lamp::SetLumensOutput outputs the number of lumens per lamp

Number	: [TCI_5220]
Status	: Not Verified
Type	: Low Level Test (Integration Unit/Module Test)
Group	: Low Level Tests
TCF	: Lamp.tcf
Description	: Verify that Lamp::SetLumensOutput outputs the number of lumens per lamp
Expected Results	: Stdio is receiving output data for the number of Lumens per lamp
Test Inputs	: LumensRequired=0 mThisLampTypeID=Brightest
Types of data to be recorded	: Regression Report, Dynamic Coverage Analysis Report
Criteria for evaluating results	: Expected and actual values for outputs must match. Review TBrun regression report
Test Procedure Outline	: TBrun sequence
Prerequisite Conditions	: Lamp constructor has been invoked previously in the sequence
Level	: Low Level
Test Case Review	: Initial creation
Type/Class	: HMI Input/Output
Category of test	: Unit

Parents

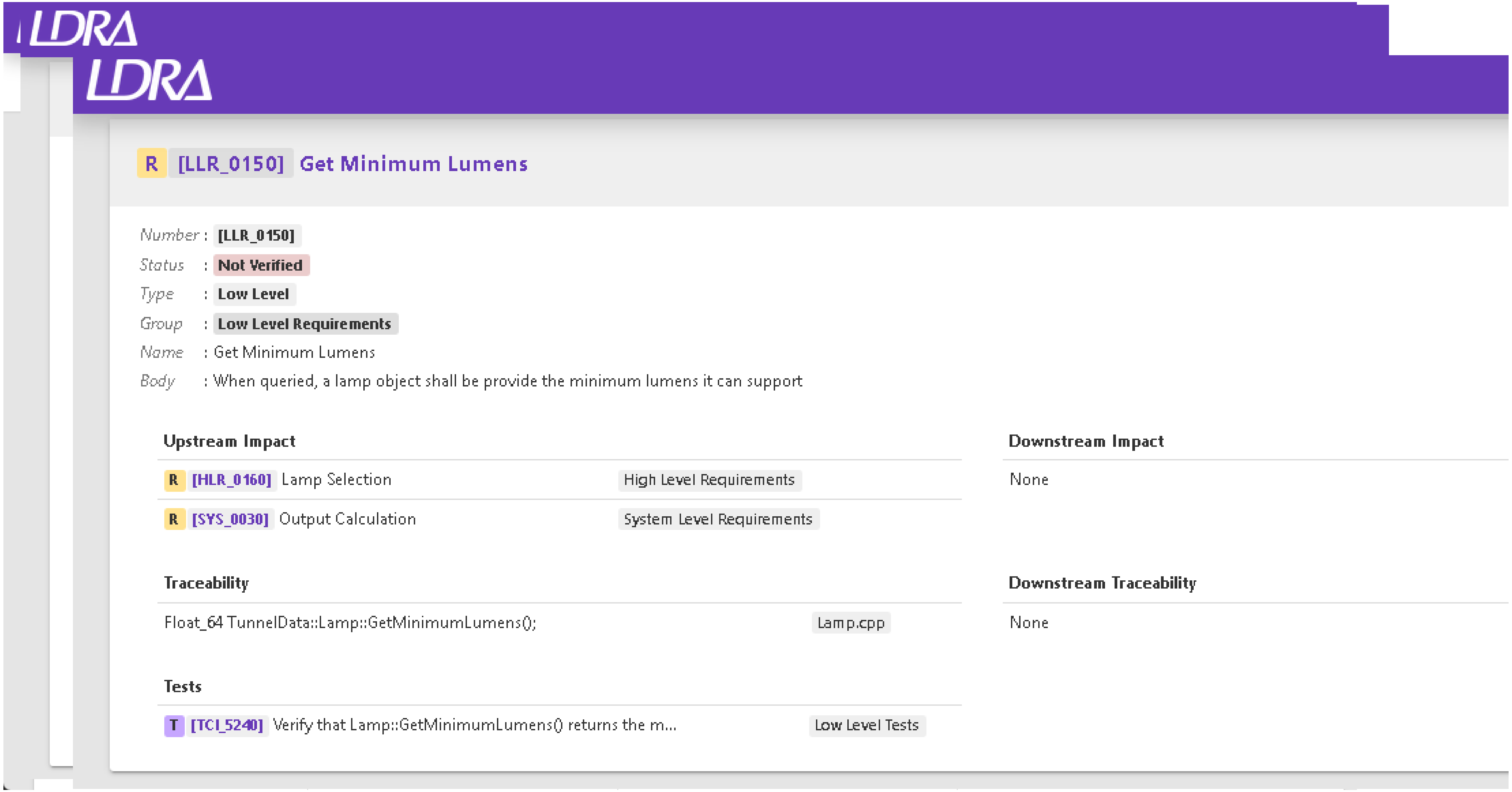
R [LLR_0130]	Set Lumens Output	Low Level Requirements

Assets/Artifacts

Lamp.tcf	Asset

numb...

Full Project Tree Report



R [LLR_0150] Get Minimum Lumens

Number : **[LLR_0150]**

Status : **Not Verified**

Type : **Low Level**

Group : **Low Level Requirements**

Name : Get Minimum Lumens

Body : When queried, a lamp object shall be provide the minimum lumens it can support

Upstream Impact

R	[HLR_0160]	Lamp Selection	High Level Requirements
R	[SYS_0030]	Output Calculation	System Level Requirements

Downstream Impact

None

Traceability

Float_64 TunnelData::Lamp::GetMinimumLumens(); Lamp.cpp

Downstream Traceability

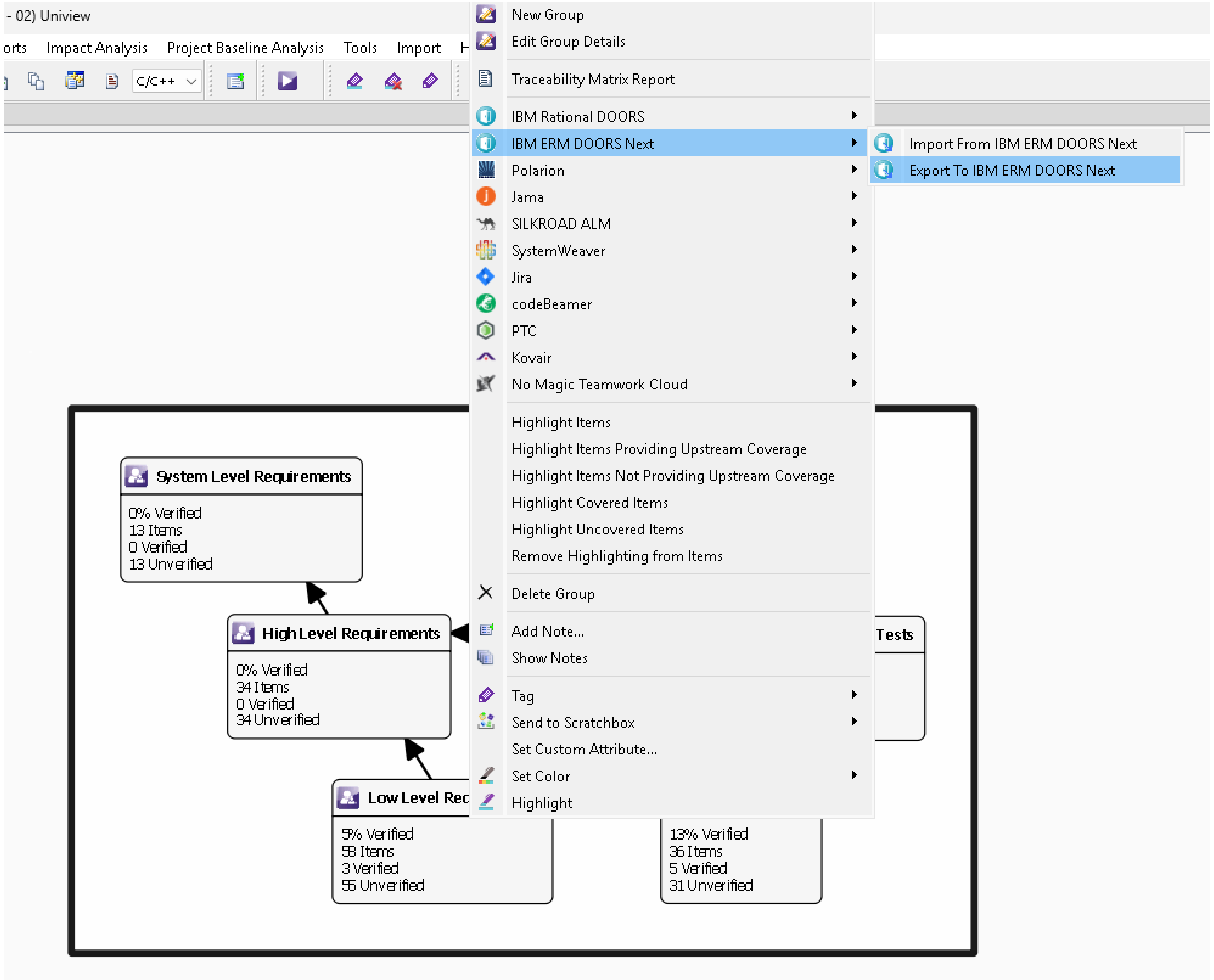
None

Tests

T	[TCI_5240]	Verify that Lamp::GetMinimumLumens() returns the m...	Low Level Tests

Import and Export to ALM

- Can import and export to ALM tools
  - Can export back test case status
  - Can export traceability all the way down to the function level



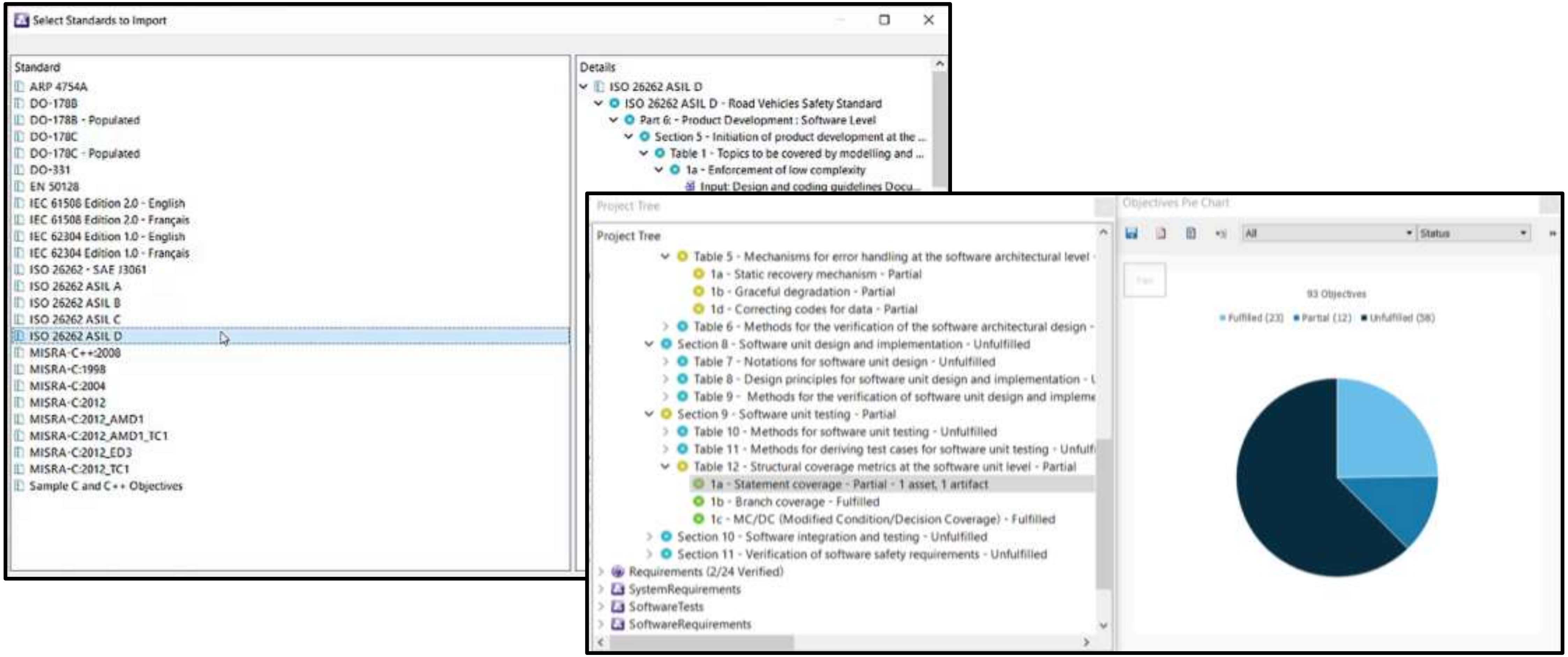
Impact Analysis and Change Management

What do we mean by Impact Analysis and Change Management?

- **Impact analysis** provides insights into the up/down stream impact of a potential change, enabling more informed design decisions to be made

- **Change management** shows up/down stream items suspected to be affected by a change, guiding the development team in making sure all related artifacts are reviewed and updated appropriately.

Objectives Tracking



Select Standards to Import

```
Standard
ARP 4754A
DO-178B
DO-178B - Populated
DO-178C
DO-178C - Populated
DO-331
EN 50128
IEC 61508 Edition 2.0 - English
IEC 61508 Edition 2.0 - Français
IEC 62304 Edition 1.0 - English
IEC 62304 Edition 1.0 - Français
ISO 26262 - SAE J3061
ISO 26262 ASIL A
ISO 26262 ASIL B
ISO 26262 ASIL C
ISO 26262 ASIL D
MISRA-C++:2008
MISRA-C:1998
MISRA-C:2004
MISRA-C:2012
MISRA-C:2012_AMD1
MISRA-C:2012_AMD1_TC1
MISRA-C:2012_ED3
MISRA-C:2012_TC1
Sample C and C++ Objectives
```

```
Details
ISO 26262 ASIL D
  ISO 26262 ASIL D - Road Vehicles Safety Standard
    Part 6: - Product Development : Software Level
      Section 5 - Initiation of product development at the ...
        Table 1 - Topics to be covered by modelling and ...
          1a - Enforcement of low complexity
            Input: Design and coding guidelines Docu...
```

Project Tree

```
Project Tree
        Table 5 - Mechanisms for error handling at the software architectural level -
          1a - Static recovery mechanism - Partial
          1b - Graceful degradation - Partial
          1d - Correcting codes for data - Partial
        Table 6 - Methods for the verification of the software architectural design -
      Section 8 - Software unit design and implementation - Unfulfilled
        Table 7 - Notations for software unit design - Unfulfilled
        Table 8 - Design principles for software unit design and implementation - U
        Table 9 -  Methods for the verification of software unit design and impleme
      Section 9 - Software unit testing - Partial
        Table 10 - Methods for software unit testing - Unfulfilled
        Table 11 - Methods for deriving test cases for software unit testing - Unfulfi
        Table 12 - Structural coverage metrics at the software unit level - Partial
          1a - Statement coverage - Partial - 1 asset, 1 artifact
          1b - Branch coverage - Fulfilled
          1c - MC/DC (Modified Condition/Decision Coverage) - Fulfilled
      Section 10 - Software integration and testing - Unfulfilled
      Section 11 - Verification of software safety requirements - Unfulfilled
  Requirements (2/24 Verified)
  SystemRequirements
  SoftwareTests
  SoftwareRequirements
```

Objectives Pie Chart

All

Status

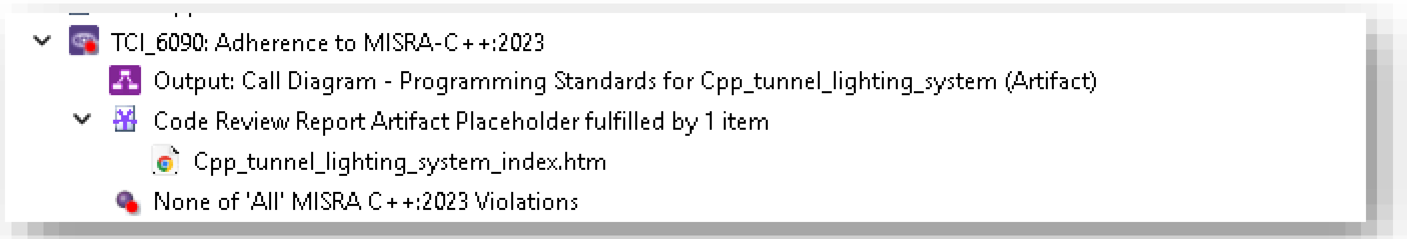
93 Objectives

■ Fulfilled (23) ■ Partial (12) ■ Unfulfilled (58)

Artifacts and assets can be linked to objectives to reduce risk and cost during the audit process

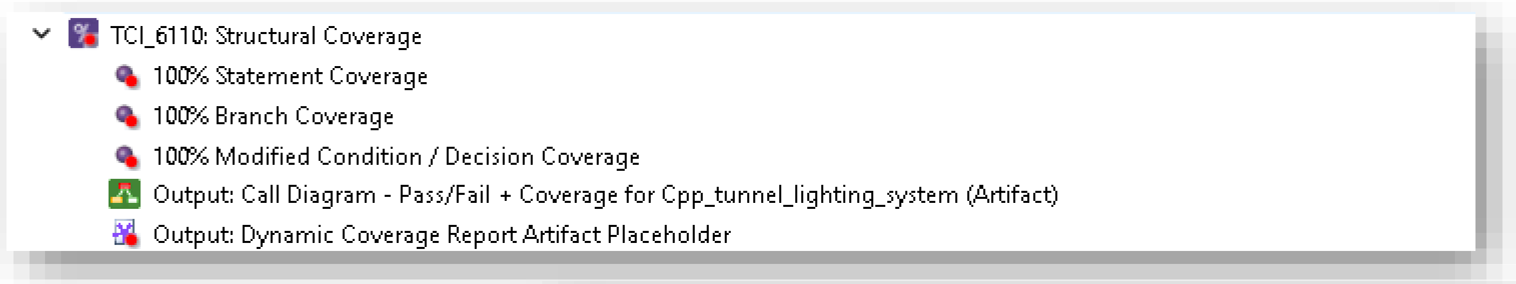
Test management

- **Static Analysis**



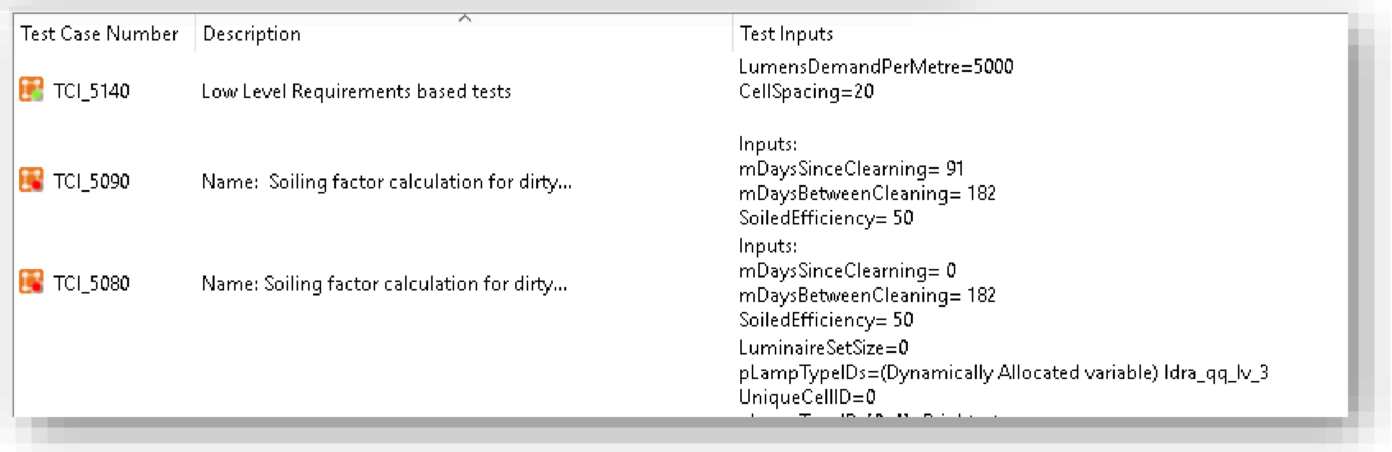
TCI_6090: Adherence to MISRA-C++:2023
- Output: Call Diagram - Programming Standards for Cpp_tunnel_lighting_system (Artifact)
- Code Review Report Artifact Placeholder fulfilled by 1 item
  - Cpp_tunnel_lighting_system_index.htm
- None of 'All' MISRA C++:2023 Violations

- **Code Coverage**



TCI_6110: Structural Coverage
- 100% Statement Coverage
- 100% Branch Coverage
- 100% Modified Condition / Decision Coverage
- Output: Call Diagram - Pass/Fail + Coverage for Cpp_tunnel_lighting_system (Artifact)
- Output: Dynamic Coverage Report Artifact Placeholder

- **Unit Testing**



Test Case Number	Description	Test Inputs

TCI_5140	Low Level Requirements based tests	LumensDemandPerMetre=5000
		CellSpacing=20

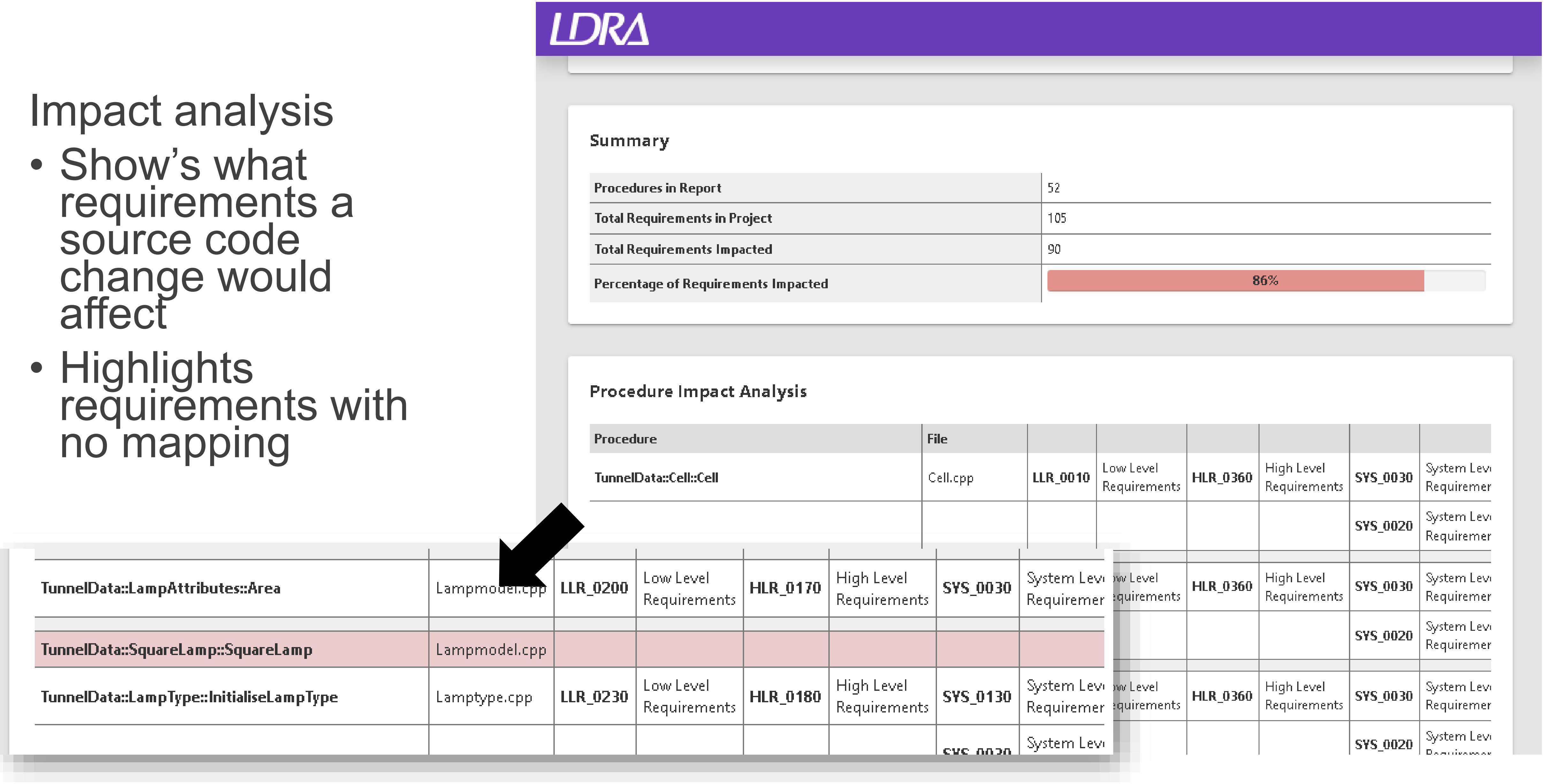
TCI_5090	Name: Soiling factor calculation for dirty...	Inputs:
		mDaysSinceClearning= 91
		mDaysBetweenCleaning= 182
		SoiledEfficiency= 50

TCI_5080	Name: Soiling factor calculation for dirty...	Inputs:
		mDaysSinceClearning= 0
		mDaysBetweenCleaning= 182
		SoiledEfficiency= 50
		LuminaireSetSize=0
		pLampTypeIDs=(Dynamically Allocated variable) ldra_qq_lv_3
		UniqueCellID=0

Impact Analysis

Impact analysis

- Show’s what requirements a source code change would affect
- Highlights requirements with no mapping



Summary

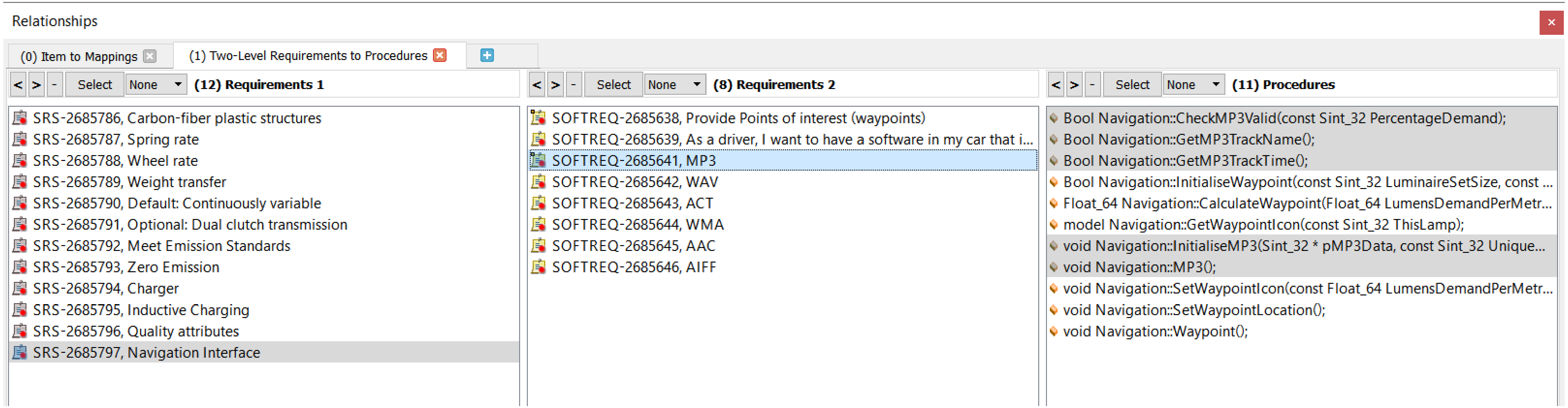
Procedures in Report	52
Total Requirements in Project	105
Total Requirements Impacted	90
Percentage of Requirements Impacted	86%

Procedure Impact Analysis

Procedure	File						
TunnelData::Cell::Cell	Cell.cpp	LLR_0010	Low Level Requirements	HLR_0360	High Level Requirements	SYS_0030	System Levi Requiremer
						SYS_0020	System Levi Requiremer
			ow Level equirements	HLR_0360	High Level Requirements	SYS_0030	System Levi Requiremer
						SYS_0020	System Levi Requiremer
			ow Level equirements	HLR_0360	High Level Requirements	SYS_0030	System Levi Requiremer
						SYS_0020	System Levi Requiremer

TunnelData::LampAttributes::Area	Lampmodel.cpp	LLR_0200	Low Level Requirements	HLR_0170	High Level Requirements	SYS_0030	System Levi Requiremer
TunnelData::SquareLamp::SquareLamp	Lampmodel.cpp						
TunnelData::LampType::InitialiseLampType	Lamptype.cpp	LLR_0230	Low Level Requirements	HLR_0180	High Level Requirements	SYS_0130	System Levi Requiremer
						SYS_0020	System Levi

Improved Decision Making



(12) Requirements 1

- SRS-2685786, Carbon-fiber plastic structures
- SRS-2685787, Spring rate
- SRS-2685788, Wheel rate
- SRS-2685789, Weight transfer
- SRS-2685790, Default: Continuously variable
- SRS-2685791, Optional: Dual clutch transmission
- SRS-2685792, Meet Emission Standards
- SRS-2685793, Zero Emission
- SRS-2685794, Charger
- SRS-2685795, Inductive Charging
- SRS-2685796, Quality attributes
- SRS-2685797, Navigation Interface

(8) Requirements 2

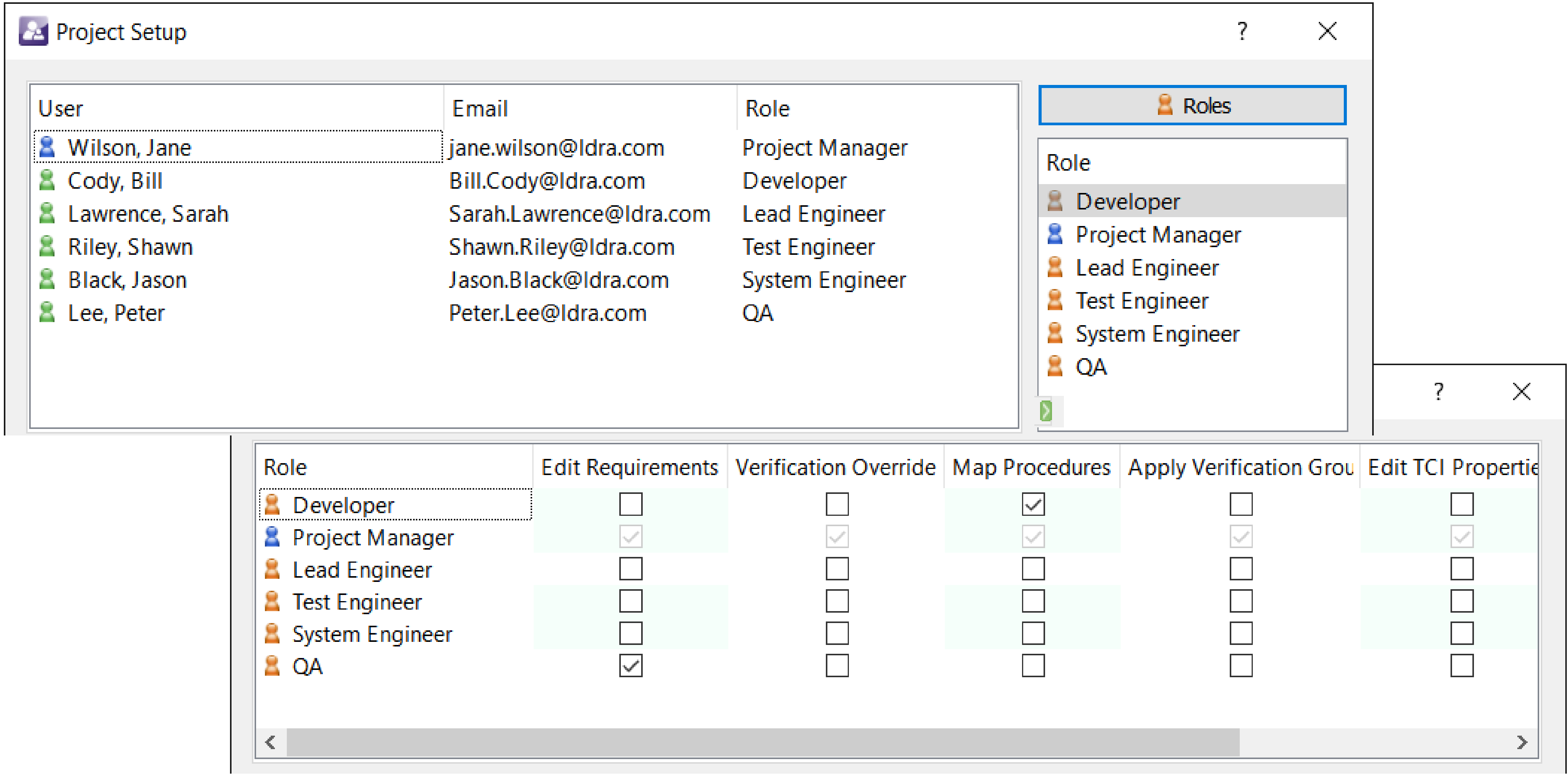
- SOFTREQ-2685638, Provide Points of interest (waypoints)
- SOFTREQ-2685639, As a driver, I want to have a software in my car that i...
- SOFTREQ-2685641, MP3
- SOFTREQ-2685642, WAV
- SOFTREQ-2685643, ACT
- SOFTREQ-2685644, WMA
- SOFTREQ-2685645, AAC
- SOFTREQ-2685646, AIFF

(11) Procedures

- Bool Navigation::CheckMP3Valid(const Sint_32 PercentageDemand);
- Bool Navigation::GetMP3TrackName();
- Bool Navigation::GetMP3TrackTime();
- Bool Navigation::InitialiseWaypoint(const Sint_32 LuminaireSetSize, const ...
- Float_64 Navigation::CalculateWaypoint(Float_64 LumensDemandPerMetr...
- model Navigation::GetWaypointIcon(const Sint_32 ThisLamp);
- void Navigation::InitialiseMP3(Sint_32 * pMP3Data, const Sint_32 Unique...
- void Navigation::MP3();
- void Navigation::SetWaypointIcon(const Float_64 LumensDemandPerMetr...
- void Navigation::SetWaypointLocation();
- void Navigation::Waypoint();

Cross Discipline Collaboration

Collaboration Management



Project Setup

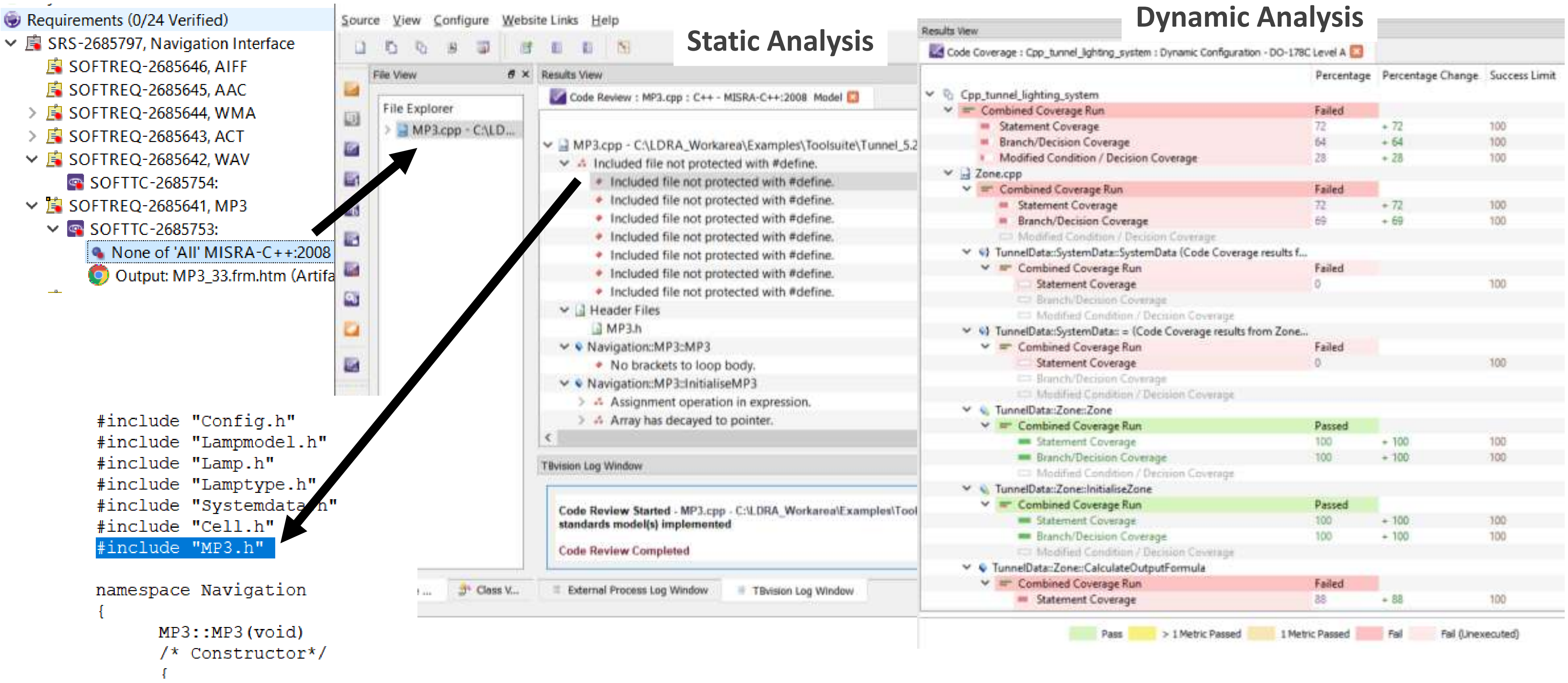
User	Email	Role
Wilson, Jane	jane.wilson@ldra.com	Project Manager
Cody, Bill	Bill.Cody@ldra.com	Developer
Lawrence, Sarah	Sarah.Lawrence@ldra.com	Lead Engineer
Riley, Shawn	Shawn.Riley@ldra.com	Test Engineer
Black, Jason	Jason.Black@ldra.com	System Engineer
Lee, Peter	Peter.Lee@ldra.com	QA

Role
Developer
Project Manager
Lead Engineer
Test Engineer
System Engineer
QA

Role	Edit Requirements	Verification Override	Map Procedures	Apply Verification Grou	Edit TCI Propertie
Developer	□	□	☑	□	□
Project Manager	☑	☑	☑	☑	☑
Lead Engineer	□	□	□	□	□
Test Engineer	□	□	□	□	□
System Engineer	□	□	□	□	□
QA	☑	□	□	□	□

Rapid Iterative Development

Shift Left – Test Early and Often, Apply Principles of CI and TDD



Requirements (0/24 Verified)
- SRS-2685797, Navigation Interface
  - SOFTREQ-2685646, AIFF
  - SOFTREQ-2685645, AAC
  - SOFTREQ-2685644, WMA
  - SOFTREQ-2685643, ACT
  - SOFTREQ-2685642, WAV
    - SOFTTC-2685754:
  - SOFTREQ-2685641, MP3
    - SOFTTC-2685753:
      - None of 'All' MISRA-C++:2008
      - Output: MP3_33.frm.htm (Artifa

Static Analysis

Code Review : MP3.cpp : C++ - MISRA-C++:2008 Model

- MP3.cpp - C:\LDRA_Workarea\Examples\Toolsuite\Tunnel_5.2
  - Included file not protected with #define.
    - Included file not protected with #define.
    - Included file not protected with #define.
    - Included file not protected with #define.
    - Included file not protected with #define.
    - Included file not protected with #define.
    - Included file not protected with #define.
    - Included file not protected with #define.
  - Header Files
    - MP3.h
  - Navigation::MP3::MP3
    - No brackets to loop body.
  - Navigation::MP3::InitialiseMP3
    - Assignment operation in expression.
    - Array has decayed to pointer.

TBvision Log Window

```
Code Review Started - MP3.cpp - C:\LDRA_Workarea\Examples\Tool
standards model(s) implemented

Code Review Completed
```

```
#include "Config.h"
#include "Lampmodel.h"
#include "Lamp.h"
#include "Lamptype.h"
#include "Systemdata.h"
#include "Cell.h"
#include "MP3.h"

namespace Navigation
{
      MP3::MP3(void)
      /* Constructor*/
      {
```

Dynamic Analysis

Results View

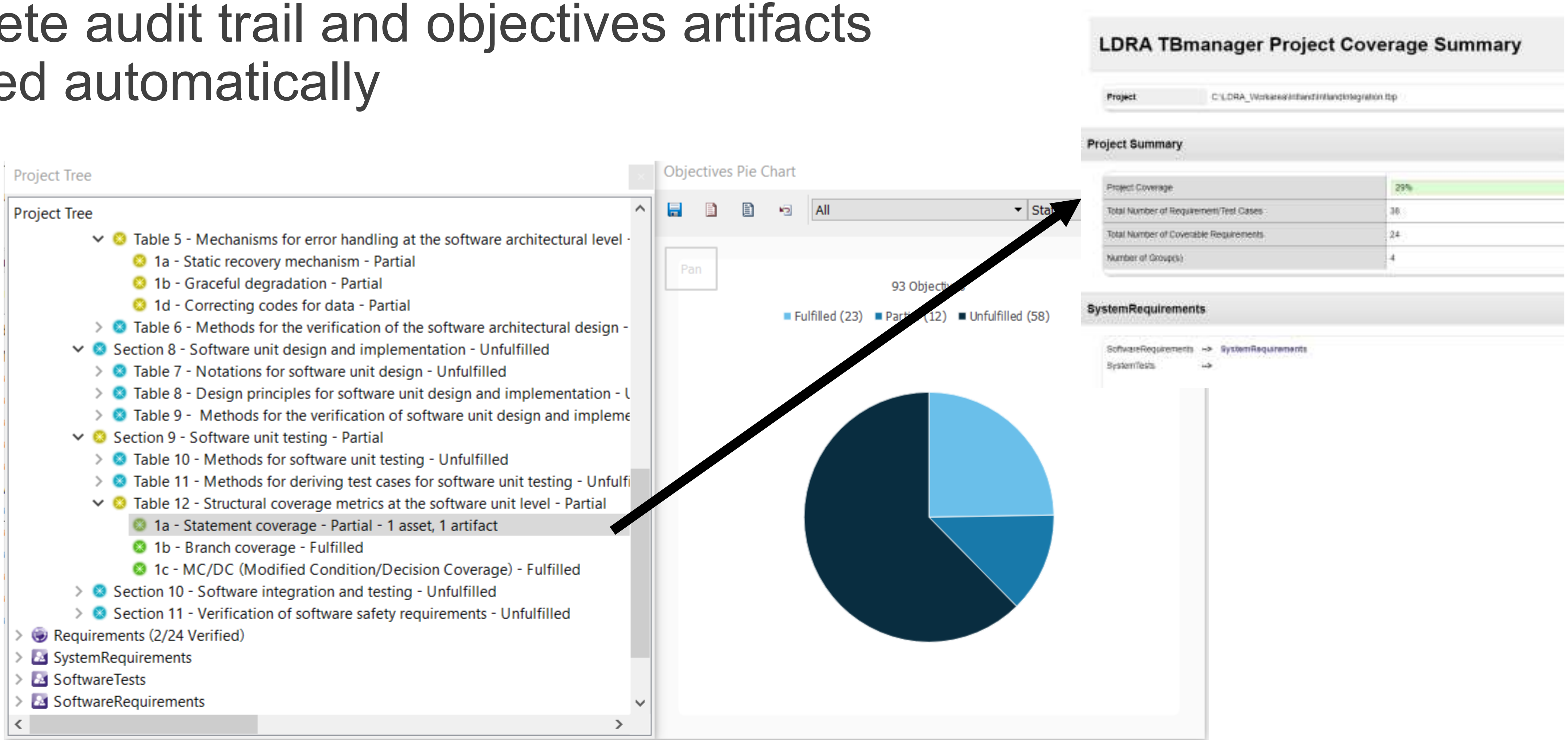
Code Coverage : Cpp_tunnel_lighting_system : Dynamic Configuration - DO-178C Level A

	Percentage	Percentage Change	Success Limit
Cpp_tunnel_lighting_system			
Combined Coverage Run	Failed		
Statement Coverage	72	+ 72	100
Branch/Decision Coverage	64	+ 64	100
Modified Condition / Decision Coverage	28	+ 28	100
Zone.cpp			
Combined Coverage Run	Failed		
Statement Coverage	72	+ 72	100
Branch/Decision Coverage	69	+ 69	100
Modified Condition / Decision Coverage			
TunnelData::SystemData::SystemData (Code Coverage results f...			
Combined Coverage Run	Failed		
Statement Coverage	0		100
Branch/Decision Coverage			
Modified Condition / Decision Coverage			
TunnelData::SystemData:: = (Code Coverage results from Zone...			
Combined Coverage Run	Failed		
Statement Coverage	0		100
Branch/Decision Coverage			
Modified Condition / Decision Coverage			
TunnelData::Zone::Zone			
Combined Coverage Run	Passed		
Statement Coverage	100	+ 100	100
Branch/Decision Coverage	100	+ 100	100
Modified Condition / Decision Coverage			
TunnelData::Zone::InitialiseZone			
Combined Coverage Run	Passed		
Statement Coverage	100	+ 100	100
Branch/Decision Coverage	100	+ 100	100
Modified Condition / Decision Coverage			
TunnelData::Zone::CalculateOutputFormula			
Combined Coverage Run	Failed		
Statement Coverage	88	+ 88	100

Pass · > 1 Metric Passed · 1 Metric Passed · Fail · Fail (Unexecuted)

Efficient and Predictable Compliance Management

Complete audit trail and objectives artifacts captured automatically



Project Tree

- Table 5 - Mechanisms for error handling at the software architectural level -
  - 1a - Static recovery mechanism - Partial
  - 1b - Graceful degradation - Partial
  - 1d - Correcting codes for data - Partial
- Table 6 - Methods for the verification of the software architectural design -
- Section 8 - Software unit design and implementation - Unfulfilled
  - Table 7 - Notations for software unit design - Unfulfilled
  - Table 8 - Design principles for software unit design and implementation - U
  - Table 9 - Methods for the verification of software unit design and impleme
- Section 9 - Software unit testing - Partial
  - Table 10 - Methods for software unit testing - Unfulfilled
  - Table 11 - Methods for deriving test cases for software unit testing - Unfulfi
  - Table 12 - Structural coverage metrics at the software unit level - Partial
    - 1a - Statement coverage - Partial - 1 asset, 1 artifact
    - 1b - Branch coverage - Fulfilled
    - 1c - MC/DC (Modified Condition/Decision Coverage) - Fulfilled
- Section 10 - Software integration and testing - Unfulfilled
- Section 11 - Verification of software safety requirements - Unfulfilled
- Requirements (2/24 Verified)
- SystemRequirements
- SoftwareTests
- SoftwareRequirements

LDRA TBmanager Project Coverage Summary

Project	C:\LDRA_Workarea\Inland\InlandIntegration.tbp

Project Summary

Project Coverage	29%
Total Number of Requirement/Test Cases	36
Total Number of Coverable Requirements	24
Number of Group(s)	4

SystemRequirements

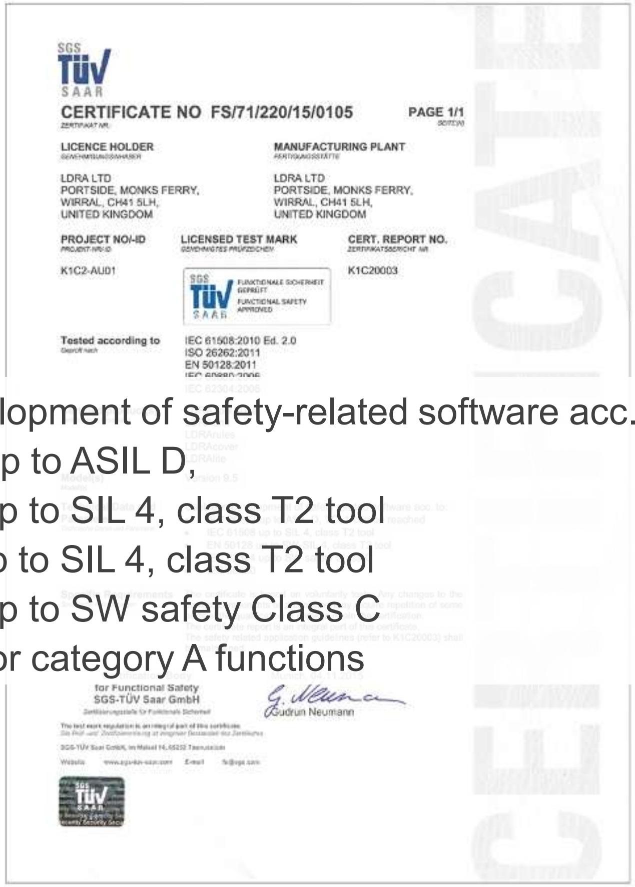
SoftwareRequirements --> SystemRequirements
SystemTests -->

Tool Qualification and Software Certification

Tool vendors provide tool qualification certificates along with artifact packages and qualification services to reduce the effort and risk of the tool qualification process

Suitable for development of safety-related software acc. to:

- ISO 26262 up to ASIL D,
- IEC 61508 up to SIL 4, class T2 tool
- EN 50128 up to SIL 4, class T2 tool
- IEC 62304 up to SW safety Class C
- IEC 60880 for category A functions



SGS TÜV SAAR

CERTIFICATE NO FS/71/220/15/0105 — PAGE 1/1

LICENCE HOLDER

LDRA LTD
PORTSIDE, MONKS FERRY,
WIRRAL, CH41 5LH,
UNITED KINGDOM

MANUFACTURING PLANT

LDRA LTD
PORTSIDE, MONKS FERRY,
WIRRAL, CH41 5LH,
UNITED KINGDOM

PROJECT NO/-ID: K1C2-AU01

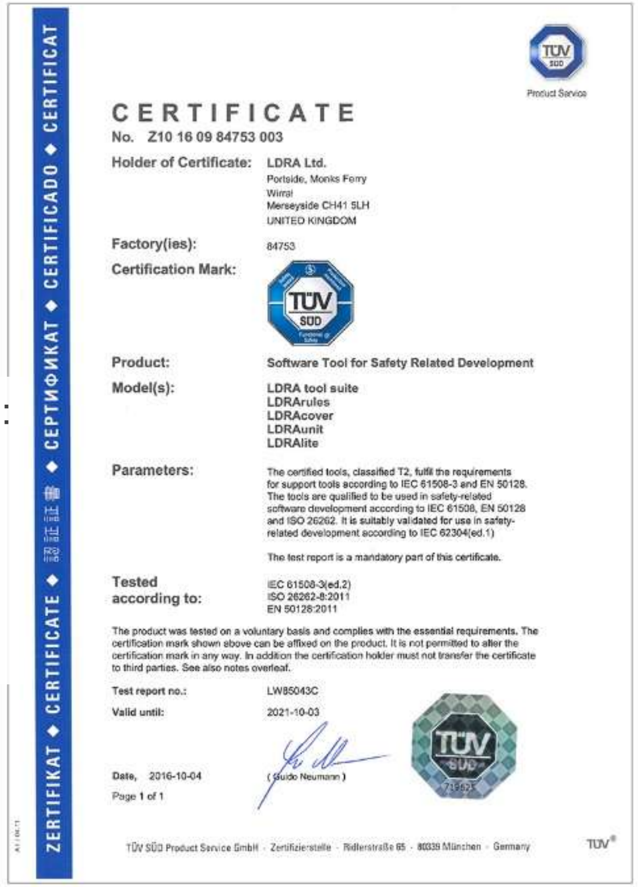
LICENSED TEST MARK: SGS TÜV SAAR FUNKTIONALE SICHERHEIT GEPRÜFT FUNCTIONAL SAFETY APPROVED

CERT. REPORT NO.: K1C20003

Tested according to: IEC 61508:2010 Ed. 2.0
ISO 26262:2011
EN 50128:2011
IEC 60880:2006
IEC 62304:2006

for Functional Safety
SGS-TÜV Saar GmbH

Gudrun Neumann



ZERTIFIKAT ◆ CERTIFICATE ◆ 認証証書 ◆ СЕРТИФИКАТ ◆ CERTIFICADO ◆ CERTIFICAT

TÜV SÜD Product Service

CERTIFICATE

No. Z10 16 09 84753 003

Holder of Certificate: LDRA Ltd.
Portside, Monks Ferry
Wirral
Merseyside CH41 5LH
UNITED KINGDOM

Factory(ies): 84753

Certification Mark: TÜV SÜD

Product: Software Tool for Safety Related Development

Model(s): LDRA tool suite
LDRArules
LDRAcover
LDRAunit
LDRAlite

Parameters: The certified tools, classified T2, fulfil the requirements for support tools according to IEC 61508-3 and EN 50128. The tools are qualified to be used in safety-related software development according to IEC 61508, EN 50128 and ISO 26262. It is suitably validated for use in safety-related development according to IEC 62304(ed.1)

The test report is a mandatory part of this certificate.

Tested according to: IEC 61508-3(ed.2)
ISO 26262-8:2011
EN 50128:2011

The product was tested on a voluntary basis and complies with the essential requirements. The certification mark shown above can be affixed on the product. It is not permitted to alter the certification mark in any way. In addition the certification holder must not transfer the certificate to third parties. See also notes overleaf.

Test report no.: LW85043C

Valid until: 2021-10-03

Date, 2016-10-04

Page 1 of 1

(Guido Neumann)

TÜV SÜD Product Service GmbH · Zertifizierstelle · Ridlerstraße 65 · 80339 München · Germany

https://ldra.com/iso-tuv-certification/

Summary

- The automotive industry is being disrupted by software-driven innovation that leverages autonomy

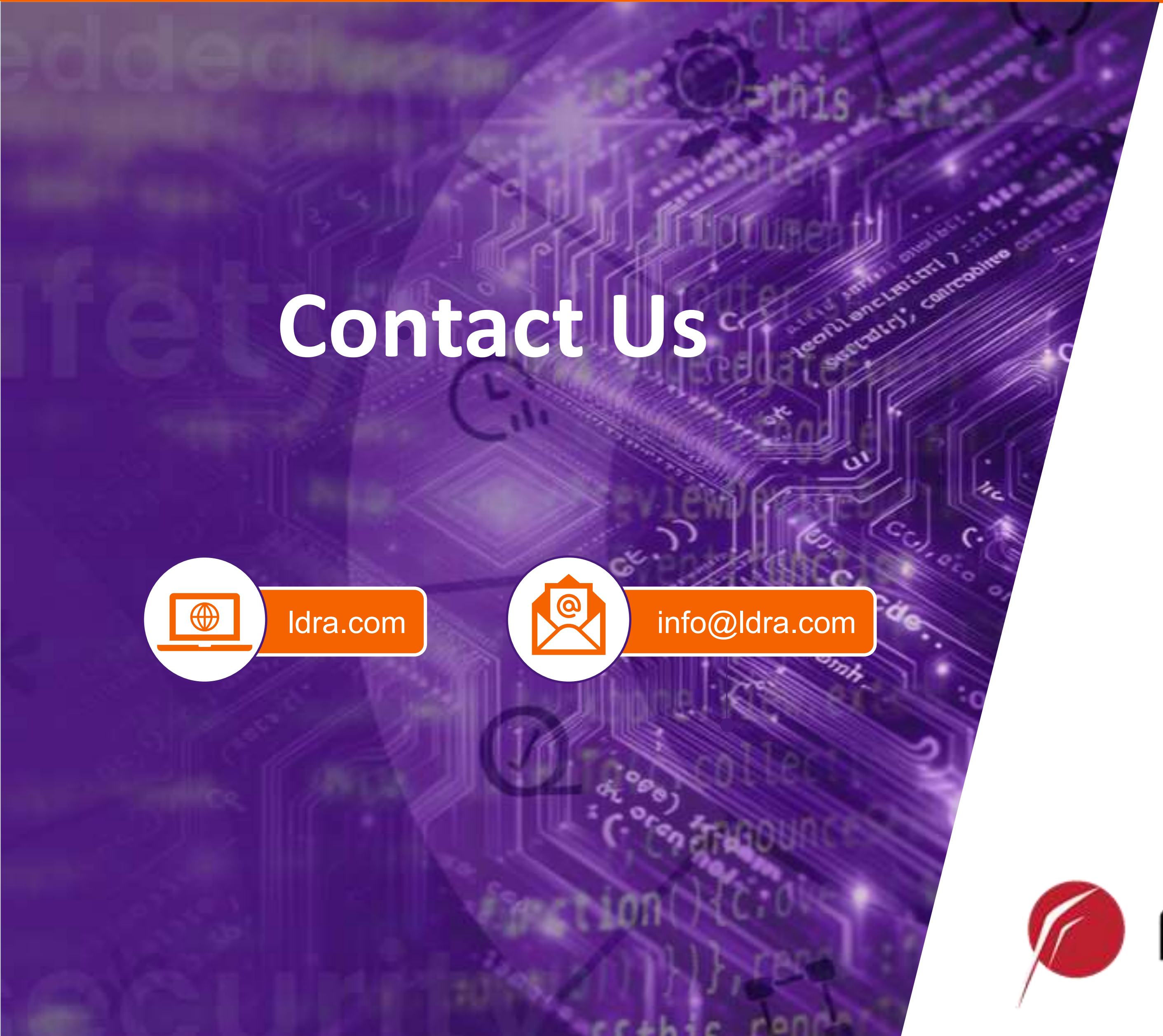
- Product complexity is growing along with the risk of failure, making compliance with functional safety and security standards more challenging

- Using a Code Quality and Verification Management solution simplifies and automates many aspects of system and software development and verification required by ISO 26262, removing cost and risk, allowing companies to accelerate business value by taking advantage of the opportunities in the ADAS landscape

FUJI SETSUBI

LDRA スタンダード認証支援テストツール
https://www.fuji-setsu.co.jp/products/LDRA/

富士設備工業（株）電子機器事業部
https://www.fuji-setsu.co.jp



Contact Us

ldra.com

info@ldra.com

Follow Us

LDRA Limited

LDRA

FUJI SETSUBI