

Using IBM Rational Tools for ISO 26262 and Automotive SPICE

Kim, Dong Young
Watson IoT Connected Products
IBM Korea

Agenda

1. ISO 26262 and Automotive SPICE
2. Key concepts for the standard compliance
3. Traceability in Requirements
4. Task and Process
5. Out-of-the-Box template for ISO 26262
6. Model based development and testing
7. Summary

ISO 26262 and Automotive SPICE

- ***ISO 26262***

- Focused on all aspects of ***functional safety*** including ***hardware*** and ***software***.
- Based on broader standard of IEC 61508, a general standard on electronic and programable devices for functional safety.
- ***Certification*** on ***process***, ***product*** and ***tools***

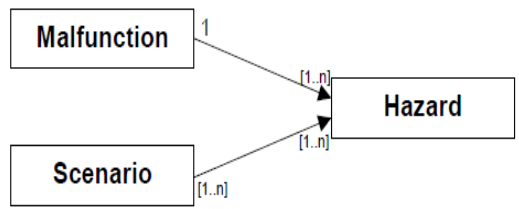
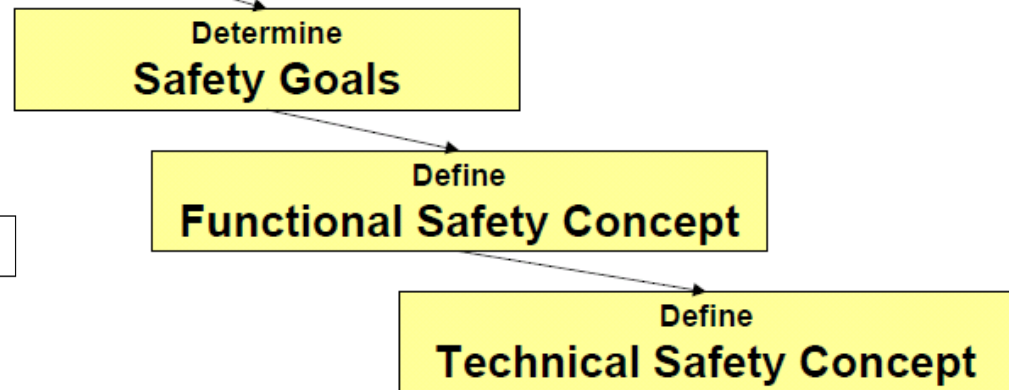
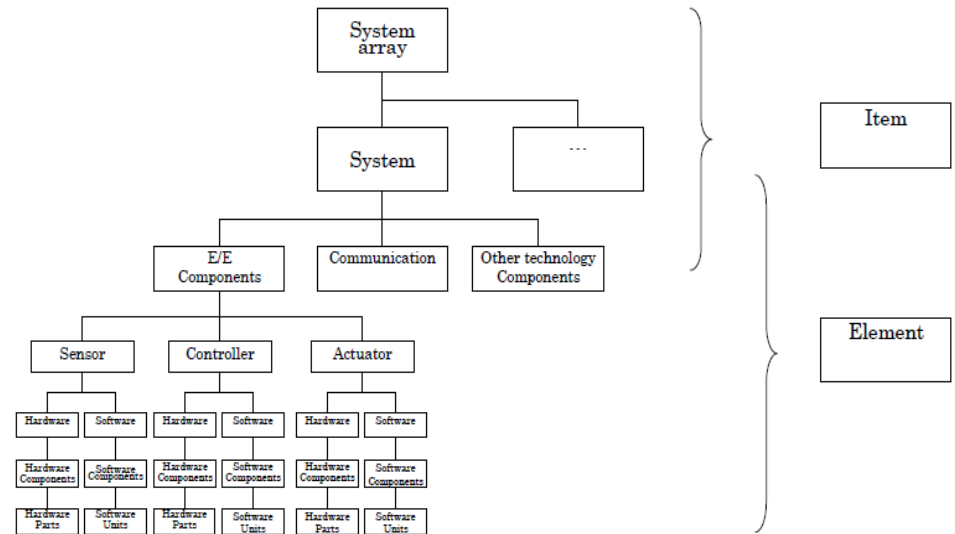
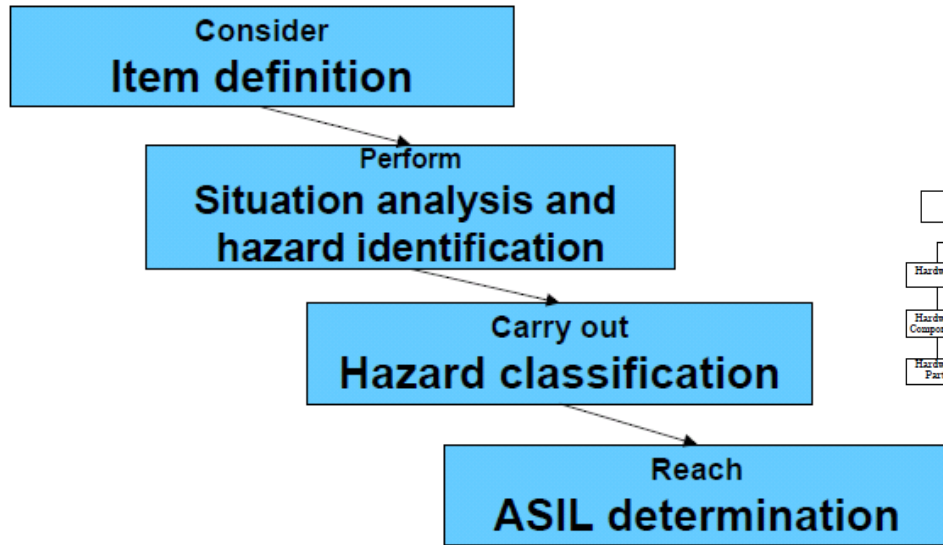
- ***Automotive SPICE***

- Main focus is ***software***.
- It has two dimensions of ***process*** model and ***capability*** level.
 - Level 0-5

ISO 26262

- Functional Safety for *automotive industry*.
- It became an official ISO standard.
 - It is automotive application of IEC 61508, a more general standard on electronic and programmable devices for functional safety.
- Key concepts
 - *ASIL* (Automotive Safety Integrity Level)
 - *Safety lifecycle management*

ASIL Determination and Lifecycle Concepts



Safety Lifecycle

- Safety Goal
 - Safety requirements
 - Functional safety requirements
 - Technical functional safety requirements
 - Hardware and software design and implementation
 - Testing
-
- All these need to be managed with the *traceability* among these.

Agenda

1. ISO 26262 and Automotive SPICE
2. Key concepts for the standard compliance
3. Traceability in Requirements
4. Task and Process
5. Out of the Box Template for ISO 26262
6. Model based development and testing
7. Summary

Key Concepts for Compliance with the Standards

- *Traceability*
- *Process*
- *Accountability*

Typical Situation

- *Requirements*

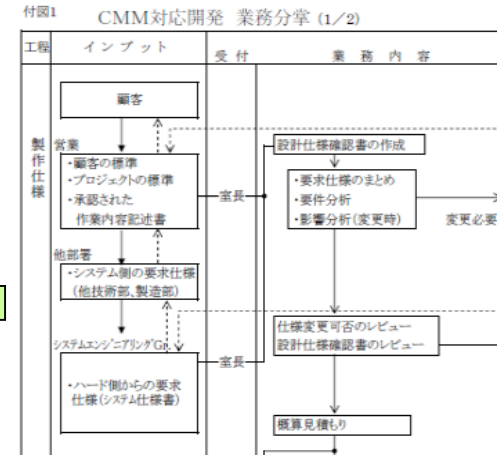
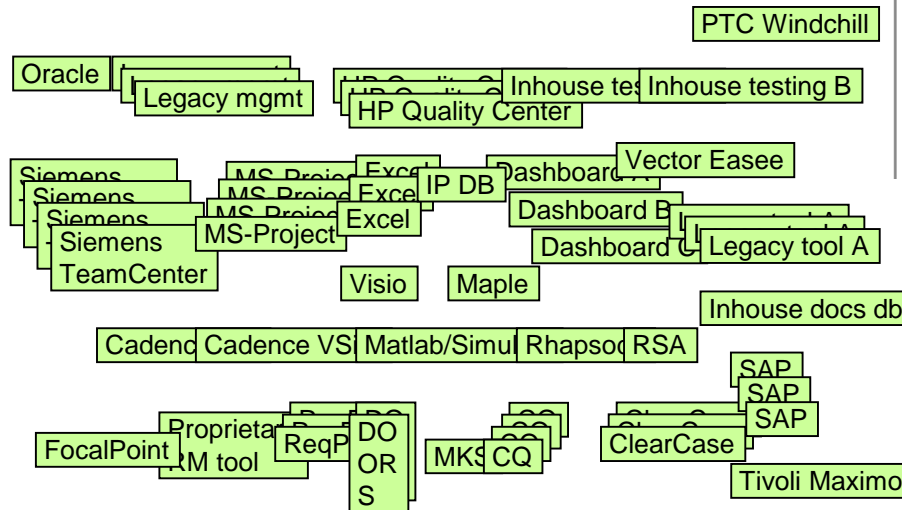


- *Process*

- Only by document

- *Tools*

- Silos
- Ad hoc integration

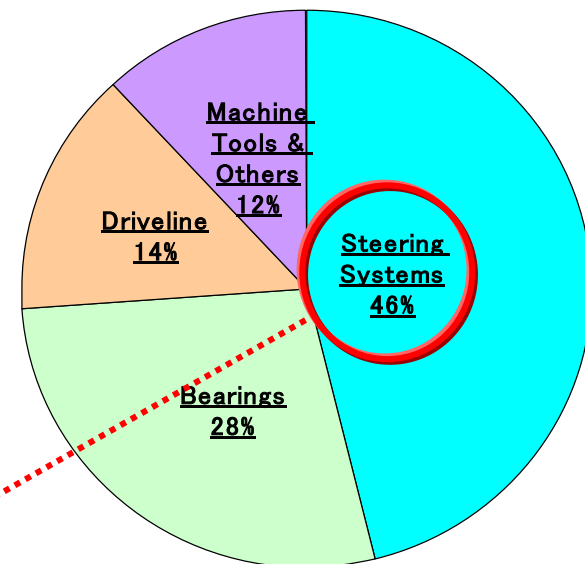


Agenda

1. ISO 26262 and Automotive SPICE
2. Key concepts for the standard compliance
3. Traceability in Requirements
4. Task and Process
5. Out of the Box Template for ISO 26262
6. Model based development and testing
7. Summary

JTEKT

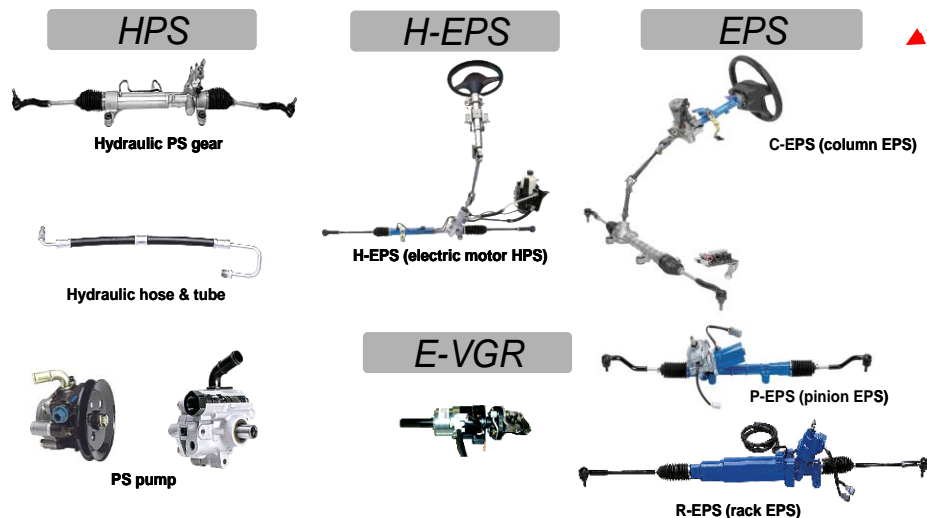
- Founding: January 2006
- Sales : 955,400 M yen
- Number of employees: 36,572
- Number of domestic plants: 12
- Number of overseas companies: 74



Consolidated Sales by product (FY 2010)

Steering Systems	3,875
Bearings	3,168
Driveline	1,246
Machine Tools & Others	1,265
Total	9,554

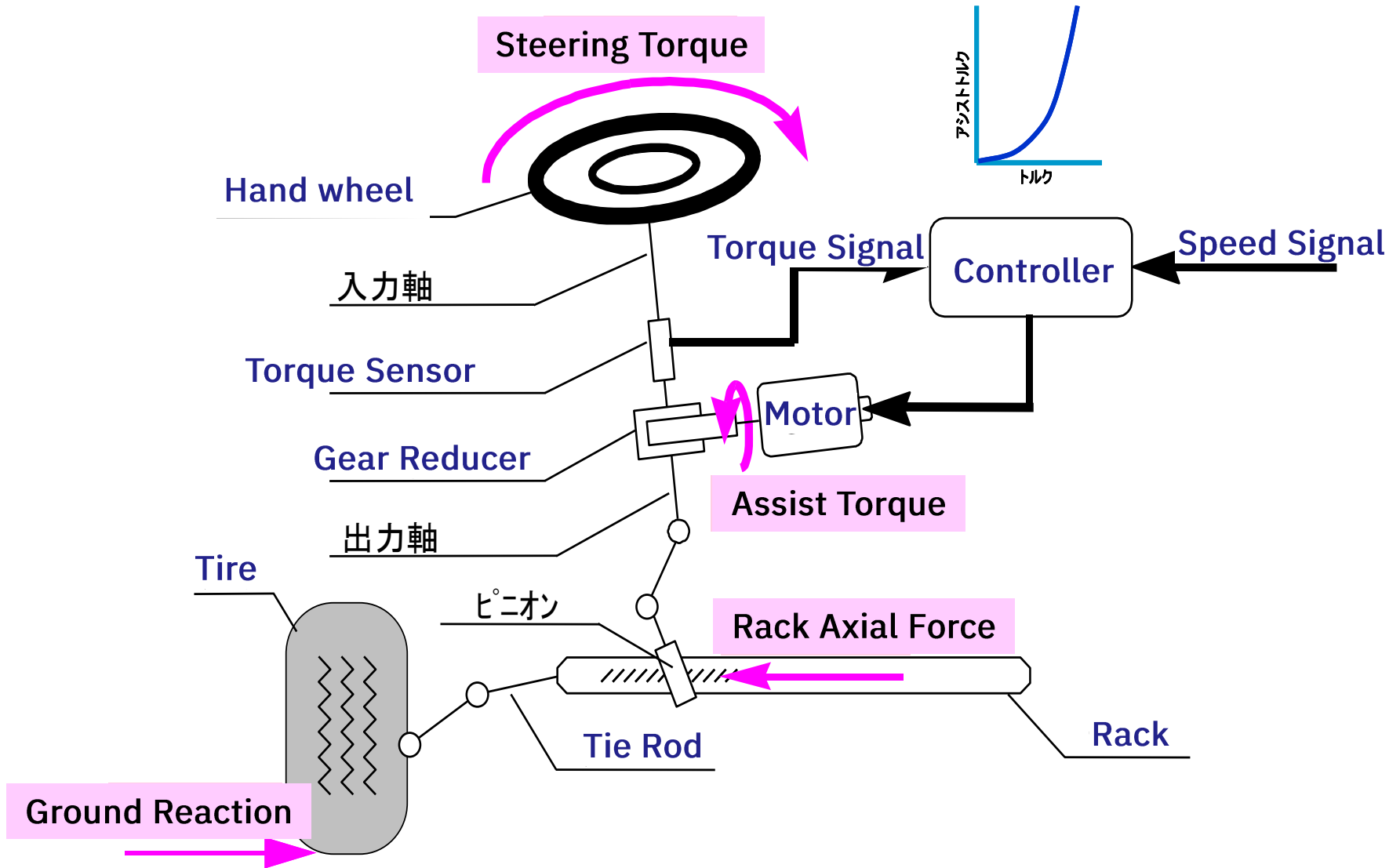
(100 M yen)



A Case Study by JTEKT

- This case examines requirement management on EPS (Electric Power Steering).
- It followed steps and created artifacts as defined by the ISO 26262 standard.
 - Preliminary Hazard Analysis (PHA)
 - Safety Goal Definition
 - Hazard And Risk Analysis (H&R)
 - Functional Safety Requirement (FSR)
 - Functional Safety Concept (FSC)

Assist Control for EPS



Preliminary Hazard Analysis

Specify potential hazards in EPS systems

ID	Item	Function	Potential failure mode	Failure mode	Direct effect	Causes	Failure mode distribution	System hazards
PHA_1	F001	Torque sensor (TSM,TSS)	No	Not delivery the torque value	torque value = init value	No Run time	Fixed to 0	Under assistance
PHA_2	F001	Torque sensor (TSM,TSS)	No	Not delivery the torque value	torque value = init value	No Run time	Fixed to High,Low	Self steer
PHA_3	F001	Torque sensor (TSM,TSS)	No	Not delivery the torque value	torque value = init value	No Run time	Fixed to small value	Asymmetrical assistance
PHA_4	F001	Torque sensor (TSM,TSS)	No	Not delivery the torque value	torque value = init value	No Run time	Fixed to small value	Under assistance
PHA_5	F001	Torque sensor (TSM,TSS)	Loss	Stop delivery the torque value	torque value fixed	Run time error / Task over run	Fixed to 0	Under assistance
PHA_6	F001	Torque sensor (TSM,TSS)	Loss	Stop delivery the torque value	torque value fixed	Run time error / Task over run	Fixed to High,Low	Self steer
PHA_7	F001	Torque sensor (TSM,TSS)	Loss	Stop delivery the torque value	torque value fixed	Run time error / Task over run	Fixed to small value	Asymmetrical assistance
PHA_8	F001	Torque sensor (TSM,TSS)	Loss	Stop delivery the torque value	torque value fixed	Run time error / Task over run	Fixed to small value	Under assistance
PHA_9	F001	Torque sensor (TSM,TSS)	Delayed	Delivery the torque value delayed	torque value delayed	Task over run	Delayed	Small valiation of assistance
PHA_10	F001	Torque sensor (TSM,TSS)	Increased	torque value increased	torque gain increased	Run Bad design Data corrupted T/S malfunction Bad design	Increased	Over assistance
PHA_11	F001	Torque sensor (TSM,TSS)	Decreased	torque value reduced	torque gain decreased	Data corrupted T/S malfunction Bad design	Decreased	Under assistance

ex. "Self steering" caused by "torque sensor failure" (causing a "fixed signal to be output")

Hazard and Risk Analysis

Specify the risks regarding each hazard and assign ASILs

ID	Condition1	Condition2	Condition3	Condition4	E	C(Se)	S(Se)	ASIL(Self steer)	C(Mot)	S(Motor)	ASIL(Motor locking)	C(Mot)	S(Mot)	ASIL(Motor breaking)	C(Su)
RA_1	City driving	free driving	Straight line	Dry(~30Km/h)	E4	C3	S2	C	C1	S0	QM	C1	S0	QM	C1
RA_2	City driving	free driving	Straight line	Dry(~70Km/h)	E4	C3	S3	D	C1	S0	QM	C1	S0	QM	C1
RA_3	City driving	free driving	Straight line	Wet roads(~30Km/h)	E3	C3	S2	B	C1	S0	QM	C1	S0	QM	C1
RA_4	City driving	free driving	Straight line	Wet roads(~70Km/h)	E3	C3	S3	C	C1	S0	QM	C1	S0	QM	C1
RA_5	City driving	free driving	Gentler curve	Dry(~30Km/h)	E4	C3	S2	C	C3	S2	C	C3	S2	C	C2
RA_6	City driving	free driving	Gentler curve	Dry(~70Km/h)	E4	C3	S3	D	C3	S3	D	C3	S3	D	C2
RA_7	City driving	free driving	Gentler curve	Wet roads(~30Km/h)	E3	C3	S2	B	C3	S2	B	C3	S2	B	C2
RA_8	City driving	free driving	Gentler curve	Wet roads(~70Km/h)	E3	C3	S3	C	C3	S3	C	C3	S3	C	C2
RA_9	City driving	free driving	Sharp curve	Dry(~30Km/h)	E4	C3	S2	C	C3	S2	C	C3	S2	C	C3
RA_10	City driving	free driving	Sharp curve	Dry(~70Km/h)	E4	C3	S3	D	C3	S3	D	C3	S3	D	C3
RA_11	City driving	free driving	Sharp curve	Wet roads(~30Km/h)	E3	C3	S2	B	C3	S2	B	C3	S2	B	C3
RA_12	City driving	free driving	Sharp curve	Wet roads(~70Km/h)	E3	C3	S3	C	C3	S3	C	C3	S3	C	C3
RA_13	City driving	free driving on the road which there are 2 or more lanes	Straight line	Dry(~30Km/h)	E4	C3	S2	C	C1	S0	QM	C1	S0	QM	C1
RA_14	City driving	free driving on the road which there are 2 or more lanes	Straight line	Dry(~70Km/h)	E4	C3	S3	D	C1	S0	QM	C1	S0	QM	C1
RA_15	City driving	free driving on the road which there are 2 or more lanes	Straight line	Wet roads(~30Km/h)	E3	C3	S2	B	C1	S0	QM	C1	S0	QM	C1
RA_16	City driving	free driving on the road which there are 2 or more lanes	Straight line	Wet roads(~70Km/h)	E3	C3	S3	C	C1	S0	QM	C1	S0	QM	C1
RA_17	City driving	free driving on the road which there are 2 or more lanes	Gentler curve	Dry(~30Km/h)	E4	C3	S2	C	C3	S2	C	C3	S2	C	C2
RA_18	City driving	free driving on the road which there are 2 or more lanes	Gentler curve	Dry(~70Km/h)	E4	C3	S3	D	C3	S3	D	C3	S3	D	C2
RA_19	City driving	free driving on the road which there are 2 or more lanes	Gentler curve	Wet roads(~30Km/h)	E3	C3	S2	B	C3	S2	B	C3	S2	B	C2
RA_20	City driving	free driving on the road which there are 2 or more lanes	Gentler curve	Wet roads(~70Km/h)	E3	C3	S3	C	C3	S3	C	C3	S3	C	C2
RA_21	City driving	free driving on the road which there are 2 or more lanes	Sharp curve	Dry(~30Km/h)	E4	C3	S2	C	C3	S2	C	C3	S2	C	C3

Judge the "impact" of self steering in driving scenarios where it might conceivably occur (ex. When "self steering" occurs while the driver is driving "straight line", maneuverability and stability are lost and a serious accident can occur)

E4
C3
S3
E4:C3:S3 => ASIL D

Safety Goal Definition

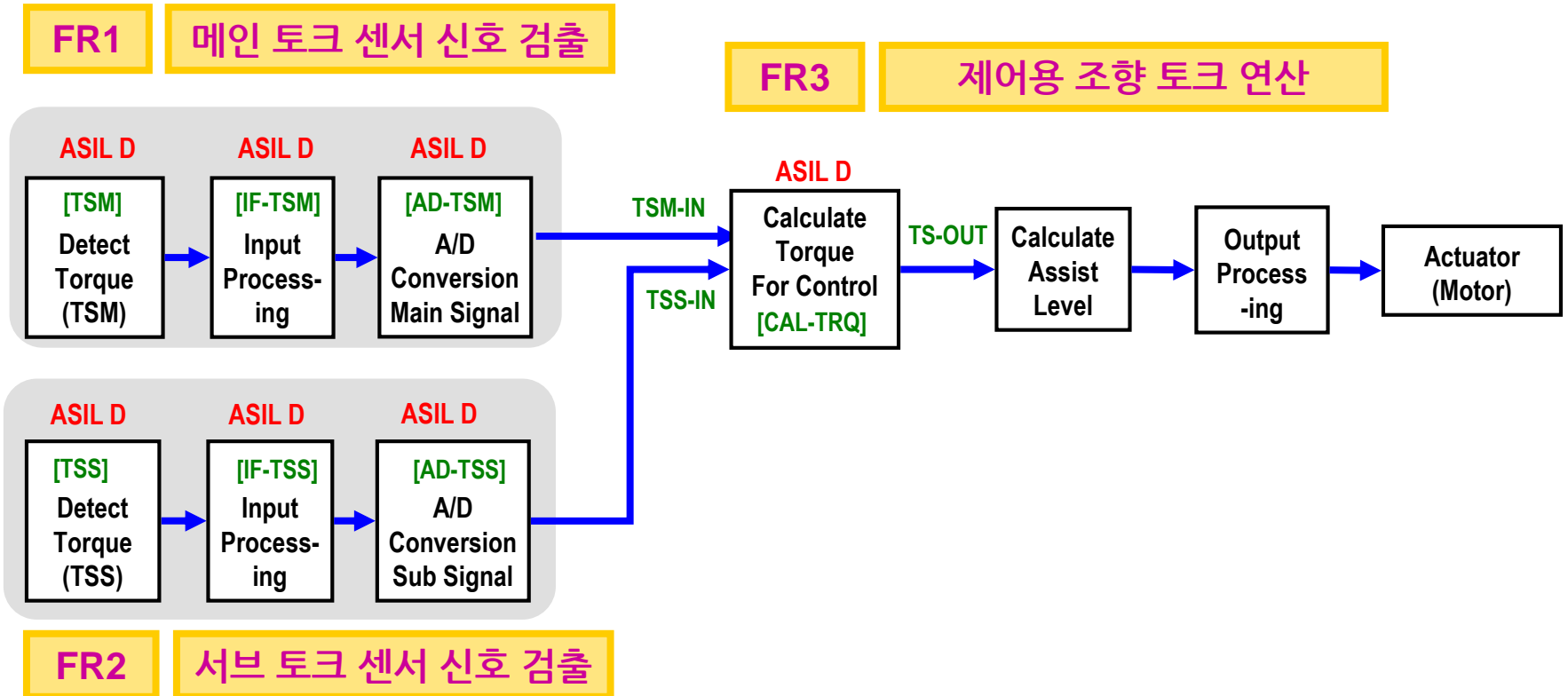
Specify safety goals regarding each hazard

ID	Safety goals	Required ASIL	Safe state
SSG_2	Steering system shall avoid the selfsteer	D	EPS shut down with failure warning
SSG_3	Steering system shall avoid the large variation of assistance	D	EPS shut down with failure warning
SSG_4	Steering system shall avoid the steering lock	D	EPS shut down with failure warning
SSG_5	Steering system shall avoid the opposite assistance	D	EPS shut down with failure warning
SSG_6	Steering system shall avoid the over assistance	C	EPS shut down with failure warning

ex. Set safety goals to “preventing self steering”.
 Assign ASIL: D
 Safe State : Turn on a warning lamp & Shutdown the EPS

EPS function

Preliminary architectural assumption



Functional Safety Requirement

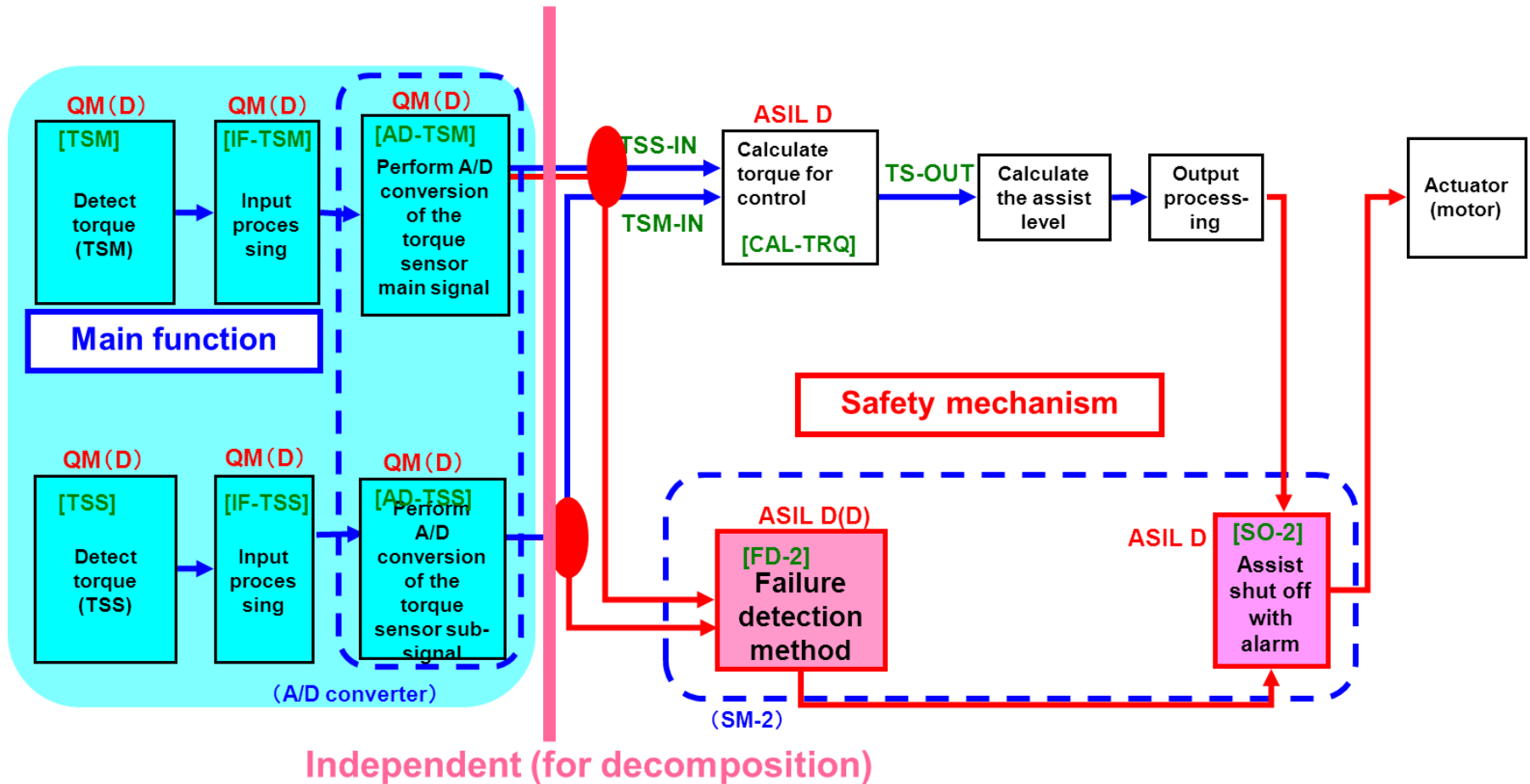
Specify safety goals regarding each hazard

ID	Function and Failure mode	Hazard	ASIL level	Safe state	Operating modes	Fault tolerance time interval
FSR_1	1 Torque sensor (TSM)					
FSR_2	Fixed to High	Self steer	D (TBD)	EPS Shutdown with Failure warning	All	20(TBD)
FSR_3	Fixed to Low	Self steer	D (TBD)	EPS Shutdown with Failure warning	All	20(TBD)
FSR_4	Fixed to small	Asymmetrical assistance	Availability target	-	All	-
FSR_5	Fixed to small	Under assistance	Availability target	-	All	-
FSR_6	Fixed to 0	Under assistance	Availability target	-	All	-
FSR_7	Increased	Over assistance	C (TBD)	EPS Shutdown with Failure warning	All	200(TBD)
FSR_8	Decreased	Under assistance	Availability target	-	All	-
FSR_9	Offset (High)	Self steer	D (TBD)	EPS Shutdown with Failure warning	All	20(TBD)
FSR_10	Offset (Low)	Self steer	D (TBD)	EPS Shutdown with Failure warning	All	20(TBD)
FSR_11	Offset (Small)	Asymmetrical assistance	Availability target	-	All	-

Define FSR for Every Failure mode
(ex. Set safety goals to “preventing self steering”)
Assign ASIL : D
Safe State : Turn on a warning lamp & Shut down the EPS

Functional Safety Concept

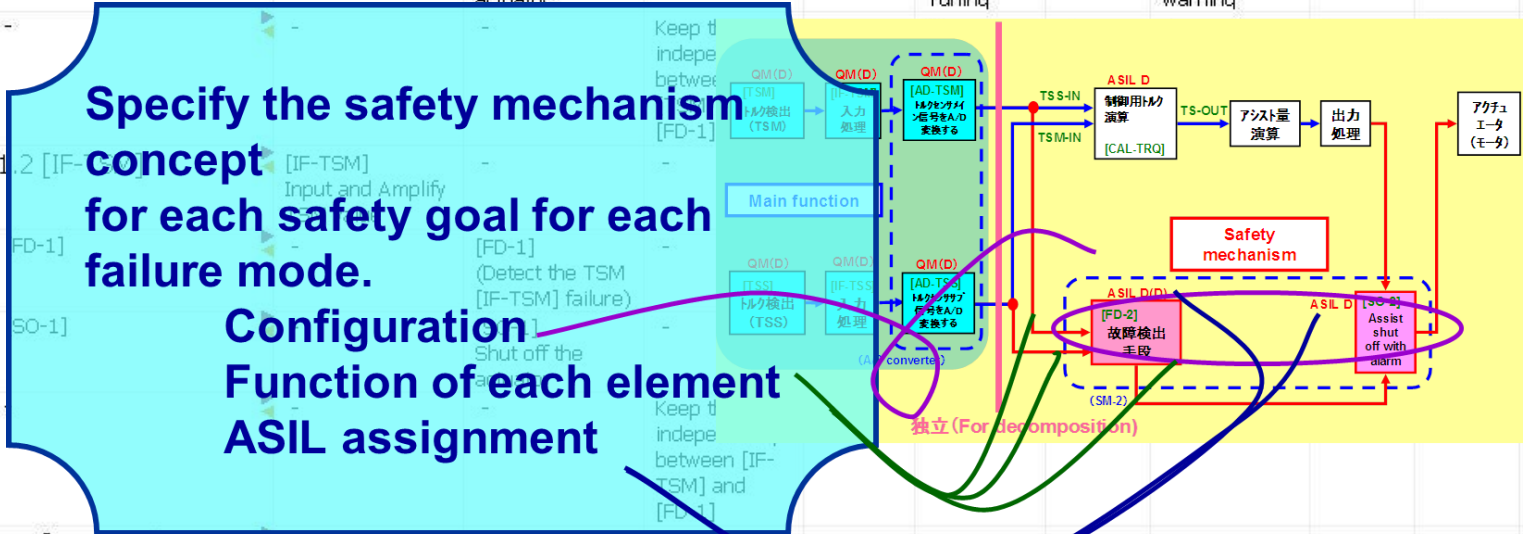
Review the safety mechanism



Functional Safety Concept

Create safety requirement specifications for the allocation of safety requirements to the system.

ID	Hazard and Function	Allocated functional s	Allocated functional	Another functional	Allocated Fu	Operating mod	Fault tolera	Safe state (in cas	Emergency	Function re	Action need
FSC_TS_1	1 Self steer										
FSC_TS_2	1.1 Torque sensor (TSM)										
FSC_TS_3	1.1.1 [TSM]	[TSM] Detect the TSM value	-	--	QM(D)	Running/ Initialize/ Tuning	20(TBD)	EPS Shutdown with Failure warning	TBD	No	No
FSC_TS_4	[FD-1]	-	[FD-1] (Detect the TSM [TSM] failure)	--	D(D)	Running/ Initialize/ Tuning	20(TBD)	EPS Shutdown with Failure warning	TBD	No	No
FSC_TS_5	[SO-1]	-	[SO-1] Shut off the actuator	--	D	Running/ Initialize/ Tuning	20(TBD)	EPS Shutdown with Failure warning	TBD	No	No
FSC_TS_6	-	-	-	-	-	-	-	-	-	-	-
FSC_TS_7	1.1.2 [IF-TSM]	[IF-TSM] Input and Amplify	-	-	-	-	-	-	-	-	No
FSC_TS_8	[FD-1]	-	[FD-1] (Detect the TSM [IF-TSM] failure)	-	-	-	-	-	-	-	No
FSC_TS_9	[SO-1]	-	[SO-1] Shut off the actuator	-	-	-	-	-	-	-	No
FSC_TS_10	-	-	-	-	-	-	-	-	-	-	-
FSC_TS_11	1.1.3 [AD-TSM]	[AD-TSM] Convert TSM value from Analog to digital (TSM-IN)	-	-	QM(D)	Running/ Initialize/ Tuning	20(TBD)	EPS Shutdown with Failure warning	TBD	No	No



Specify the safety mechanism concept for each safety goal for each failure mode.

Configuration Function of each element ASIL assignment

Results

PHA

H&R

FSR

FSC

Identifying hazard in EPS

Item	Item ID	Item Name	Item Description	Item Category	Item Status	Item Date	Item Author	Item Reviewer	Item Approved
1	1.1	1.1.1	1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1

Assigning ASIL for each hazard.

Item	Item ID	Item Name	Item Description	Item Category	Item Status	Item Date	Item Author	Item Reviewer	Item Approved
1	1.1	1.1.1	1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1

Defining FSR to achieve safety goal

Item	Item ID	Item Name	Item Description	Item Category	Item Status	Item Date	Item Author	Item Reviewer	Item Approved
1	1.1	1.1.1	1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1

Creating safety requirement specification

Item	Item ID	Item Name	Item Description	Item Category	Item Status	Item Date	Item Author	Item Reviewer	Item Approved
1	1.1	1.1.1	1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1	1.1.1.1.1

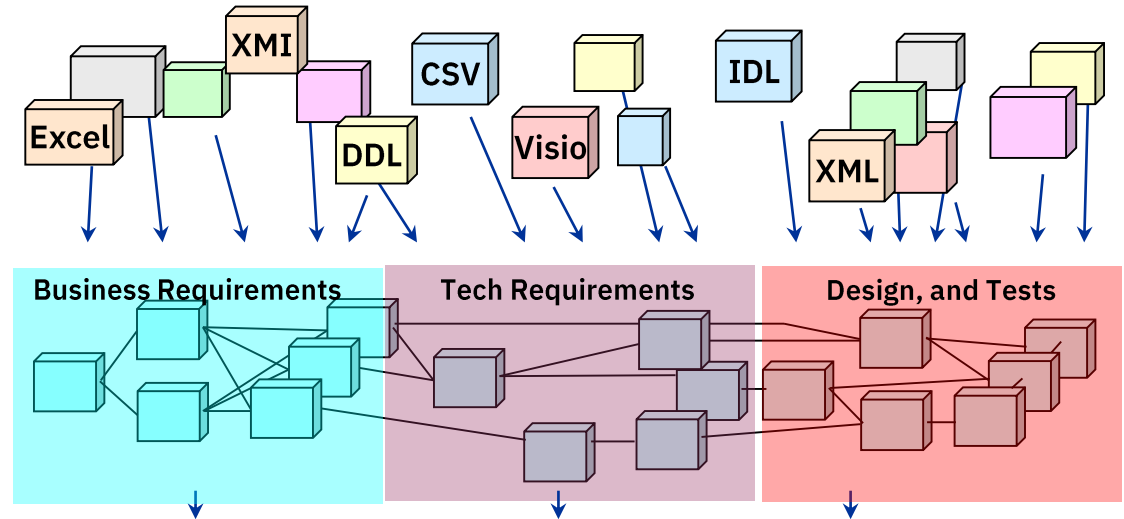
safety goal for each hazard

Safety goal ref.	Top level safety goal	Required ASIL	Safe state
SSG-001	Steering system shall avoid the selfsteer	D	EPS shut down with failure warning
SSG-002	Steering system shall avoid the large variation of assistance	D	EPS shut down with failure warning
SSG-003	Steering system shall avoid the steering lock	D	EPS shut down with failure warning
SSG-004	Steering system shall avoid the opposite assistance	D	EPS shut down with failure warning
SSG-005	Steering system shall avoid the over assistance	C	EPS shut down with failure warning

Maintaining traceability is impossible.

Rational DOORS

- Used by *major automotive OEM's*, especially in Europe.
- Supports *Traceability* in requirements



'User Requirements' current 1.0 (Issue1) in /Sub-compact (Formal module) - DOORS

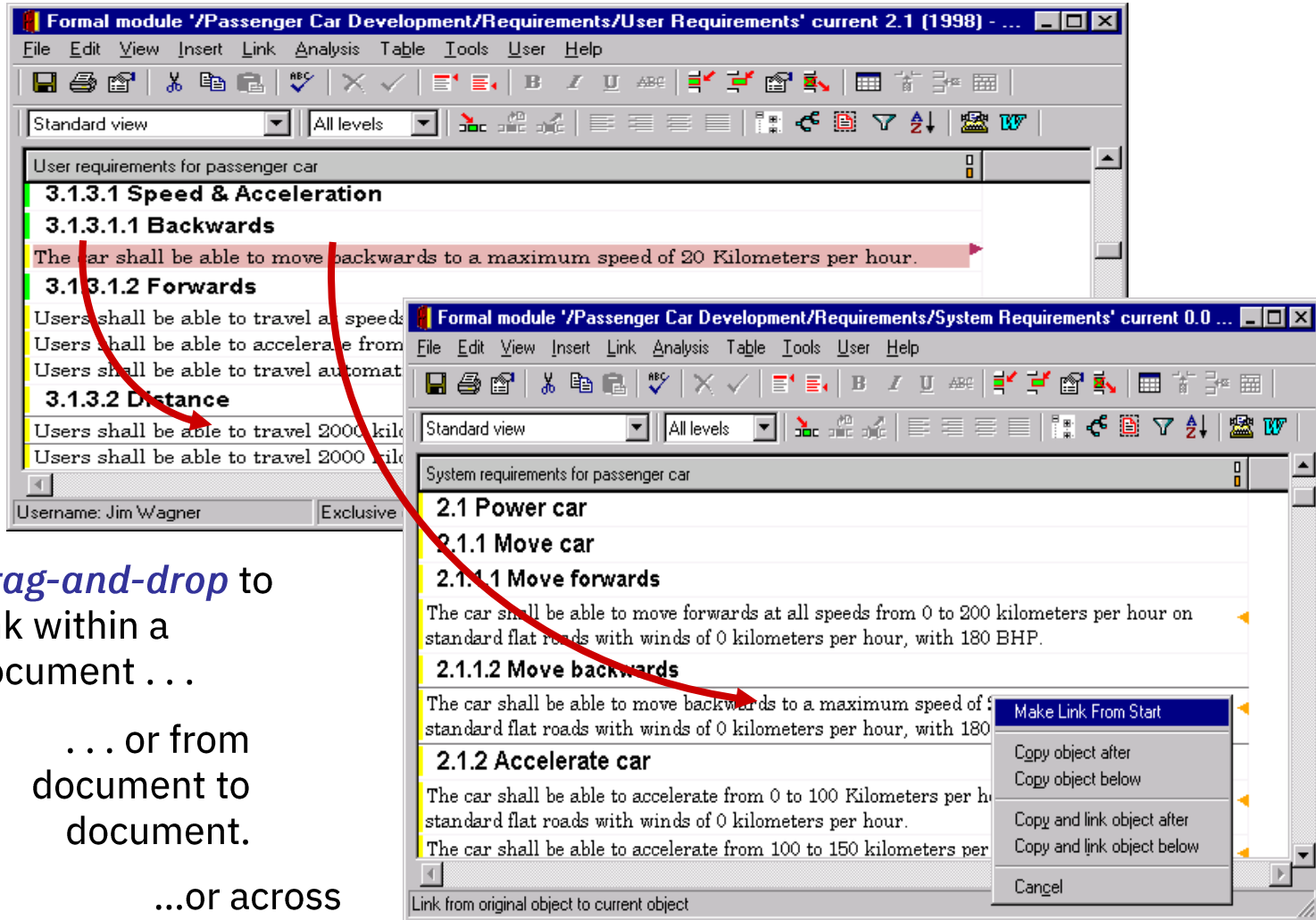
File Edit View Insert Link Analysis Table Tools User Help

View Full Trace View All levels

ID	User Requirements	Functional Requirements	Design	Test Plan
TRN-CSR-35	3.1.2.3 Stopping			
TRN-CSR-36	Users shall be able to stop safely.	<p>FR-23</p> <p>The car shall be able to stop from 10 kilometers per hour to 0 kph in 2 seconds.</p> <p>FR-24</p> <p>The car shall be able to stop from 30 kilometers per hour to 0 kph in 6 seconds.</p>	<p>TRN-AD-48 Disc brakes</p> <p>TRN-AD-48 Disc brakes</p> <p>TRN-AD-48 Disc brakes</p>	<p>TRN-TP-34 High Speed Braking Test</p> <p>TRN-TP-35 Low Speed Braking Test</p> <p>TRN-TP-34 High Speed Braking Test</p> <p>TRN-TP-35 Low Speed Braking Test</p> <p>TRN-TP-34 High Speed Braking Test</p>

Username: Dave Mason Exclusive edit mode

Linking & Traceability



Drag-and-drop to link within a document ...

... or from document to document.

...or across projects.

Use traceability to perform analysis

- Coverage analysis
- Impact analysis
- Derivation analysis

Tracking changes

Formal module 'Sports utility vehicle 4x2/Requirements/User Requirements' baseline 2.0 - DOORS

File Edit View Insert Link Analysis Table Tools User DOORSrequireIT Help

A - Basic view | All levels

User requirements for passenger car

1 Introduction

These are the initial user requirements for t
Follow this Internet link to view the market

1.1 Schedule

This module contains the user requirement
September 2001.

2 User types

This section describes the nature of the user:

2.1 New Nationalities

The car will be used in the countries, UK, U

2.2 User sizes

The car shall be suitable for people mini
kilograms to 130 kilograms.

3 Requirements

This section contains the user requirements.

3.1 Capability Requirements

3.1.1 Carrying Capacity

3.1.1.1 Number of People

Four average size adults shall be able to tr
comfort is defined as being equivalent to th
produced in 1996.

The top level of cars are those in the price ra

Username: Paul Raymond | Read-only mode - b

Previous
Baseline

Formal module 'Sports utility vehicle 4x2/Requirements/User Requirements' current 2.1 (1998) - ...

File Edit View Insert Link Analysis Table Tools User DOORSrequireIT Help

A - Basic view | All levels

User requirements for SUV 4x2

1 Introduction

These are the initial user requirements for the development of a new sports utility vehicle.
All requirements ©2000 QSS, Inc. For details on QSS visit: <http://www.qssinc.com>

1.1 Schedule

This module contains the user requirements for a new car to be co
September 2002.

2 User types

This section describes the nature of the users of the proposed vehicle.

2.1 Nationalities

The car will be used in the countries, UK, North America, Northern
Zealand.

2.2 User sizes

The car shall be suitable for people minimum and maximum sizes
kilograms to 175 Kilograms.

3 Requirements

This section contains the user requirements.

3.1 Capability Requirements

3.1.1 Carrying Capacity

3.1.1.1 Number of People

Four average size adults shall be able to travel in comfort for a per
comfort is defined as being equivalent to the standard of comfort pro
produced in 1999.

Username: Paul Raymond | Exclusive edit mode

Current Version

Change
History

Object 17 properties - DOORS

General | Access | History | Attributes | Links

User	Session	Date	Modification
Paul Raymond	452	07/02/01 08:40:25	Modify Object Attribute: Long list
Paul Raymond	452	07/02/01 08:40:52	Modify Object Attribute: Object Text
Paul Raymond	452	07/02/01 08:41:11	Modify Object Attribute: Long list
Paul Raymond	452	07/02/01 08:41:11	Modify Object Attribute: Long list
Paul Raymond	452	07/02/01 08:41:19	Modify Object Attribute: Risk
Paul Raymond	452	07/02/01 08:41:50	Modify Object Attribute: Verification M...
Paul Raymond	452	07/02/01 08:42:04	Modify Object Attribute: Object Text
Paul Raymond	452	07/02/01 08:42:10	Modify Object Attribute: Object Text

Details of selected history record

From: Four average size adults shall be able to travel in comfort for a period of 3 hours. This level of comfort is defined as being

To: Four average size adults shall be able to travel in comfort for a period of 3 hours. This level of comfort is defined as being

Restore

Only show entries with:

Dates: from: to:

User:

Details... Refresh Export...

Previous Next OK Cancel Apply Help

Verification

PHA

H&R

Safety Goal

FSC

FSR

Out Link (Depth 2)	Out Link (Depth 1)	Object Identifier	Safety goals	In Link (Depth 1)	In Link (Depth 2)	In Link (Depth 3)
Self steer	D	SSG_2	Steering system shall avoid the selfsteer	[PI-CTRL]	[PI-CTRL]	Opposite value
Self steer				[FD-3]	[FD-3]	Fixed to 0
Self steer				[SO-3]	[SO-3]	Oscillation(Low freq)
Self steer				[VMOD-3]	[VMOD-3]	Oscillation(Low freq)
Self steer				[FD-3]	[FD-3]	Fixed to High
Self steer				[SO-3]	[SO-3]	Fixed to Low
Self steer				[VMOD-2]	[VMOD-2]	Offset (High)
Self steer				[FD-3]	[FD-3]	Offset (Low)
Self steer				[SO-3]	[SO-3]	Opposite value
Self steer				[TSM]	[TSM]	Oscillation(Low freq)
Self steer				[FD-1]	[FD-1]	Offset (Low)
Self steer				[SO-1]	[SO-1]	Offset (High)
Self steer				-	-	Fixed to Low
			Fixed to High			
			Detect Microprocessor (Main-CPU)malfun.			
			Detect Microprocessor (Main-CPU)malfun.			
			Detect Microprocessor (Main-CPU)malfun.			
			Detect Microprocessor (Main-CPU)malfun.			
			Detect Microprocessor (Main-CPU)malfun.			
			Opposite value			
			Fixed to 0			
			Oscillation(Low freq)			
			Oscillation(Low freq)			
			Fixed to High			
			Fixed to Low			
			Offset (High)			
			Offset (Low)			
			Opposite value			
			Oscillation(Low freq)			
			Offset (Low)			
			Offset (High)			
			Fixed to Low			
			Fixed to High			
			Detect Microprocessor (Main-CPU)malfun.			
			Detect Microprocessor (Main-CPU)malfun.			
			Detect Microprocessor (Main-CPU)malfun.			
			Detect Microprocessor (Main-CPU)malfun.			
			Detect Microprocessor (Main-CPU)malfun.			
			Opposite value			
			Fixed to 0			
			Oscillation(Low freq)			
			Oscillation(Low freq)			

Agenda

1. ISO 26262 and Automotive SPICE
2. Key concepts for the standard compliance
3. Traceability in Requirements
4. Task and Process
5. Out of the Box Template for ISO 26262
6. Model based development and testing
7. Summary

Process Requirements

- Governance with enforcement for *accountability*
 - Publishing process definition will help for transparency.
- *Linking between process and work products* is needed for ISO 26262 and A-SPICE.
 - Measurement will be needed for process improvement for higher level capability of A-SPICE.

Rational Team Concert - Components

Work Items

- Defects, Enhancements
- Task tracking, planning
- Queries
- Approvals
- Iteration planning

SCM

- Change sets
- Streams
- Components
- Parallel development
- GDD
- *git Connector*

Build

- Work Item and Change Set Traceability
- Supports Ant, Maven and Command line Tools
- Personal Builds
- *Integration with Hudson/Jenkins*

- Project, Teams, Users
- Process enactment, customization, permissions
- In-context Collaboration, Events, Presence
- Reporting, Dashboards (project, team, personal)

Jazz Team Server

Rational Team Concert

Task Work Item: Workflow

Start action: Initialize Resolve action: Resolve Reopen action: Reopen

▼ Transitions

From	To	Closed	In Progress	New	Reopened	Resolved	Triaged	Verified
Closed		<None>	None <input type="text"/>	None <input type="text"/>	Reopen <input type="text"/>	None <input type="text"/>	None <input type="text"/>	None <input type="text"/>
In Progress		None <input type="text"/>	<None>	Stop Working <input type="text"/>	None <input type="text"/>	Resolve <input type="text"/>	None <input type="text"/>	None <input type="text"/>
New		None <input type="text"/>	Start Working <input type="text"/>	<None>	None <input type="text"/>	Resolve <input type="text"/>	Triage <input type="text"/>	None <input type="text"/>
Reopened		None <input type="text"/>	Start Working <input type="text"/>	None <input type="text"/>	<None>	Resolve <input type="text"/>	Triage <input type="text"/>	None <input type="text"/>
Resolved		Close <input type="text"/>	None <input type="text"/>	None <input type="text"/>	Reopen <input type="text"/>	<None>	None <input type="text"/>	Verify <input type="text"/>
Triaged		None <input type="text"/>	Start Working <input type="text"/>	Untriage <input type="text"/>	None <input type="text"/>	Resolve <input type="text"/>	<None>	None <input type="text"/>
Verified		Close <input type="text"/>	None <input type="text"/>	None <input type="text"/>	Reopen <input type="text"/>	Set Resolved <input type="text"/>	None <input type="text"/>	<None>

▼ Actions

Name	Target State
Close	Closed
Initialize	New
Reopen	Reopened
Resolve	Resolved
Set Resolved	Resolved
Start Working	In Progress
Stop Working	New
Triage	Triaged
Untriage	New
Verify	Verified

▼ States

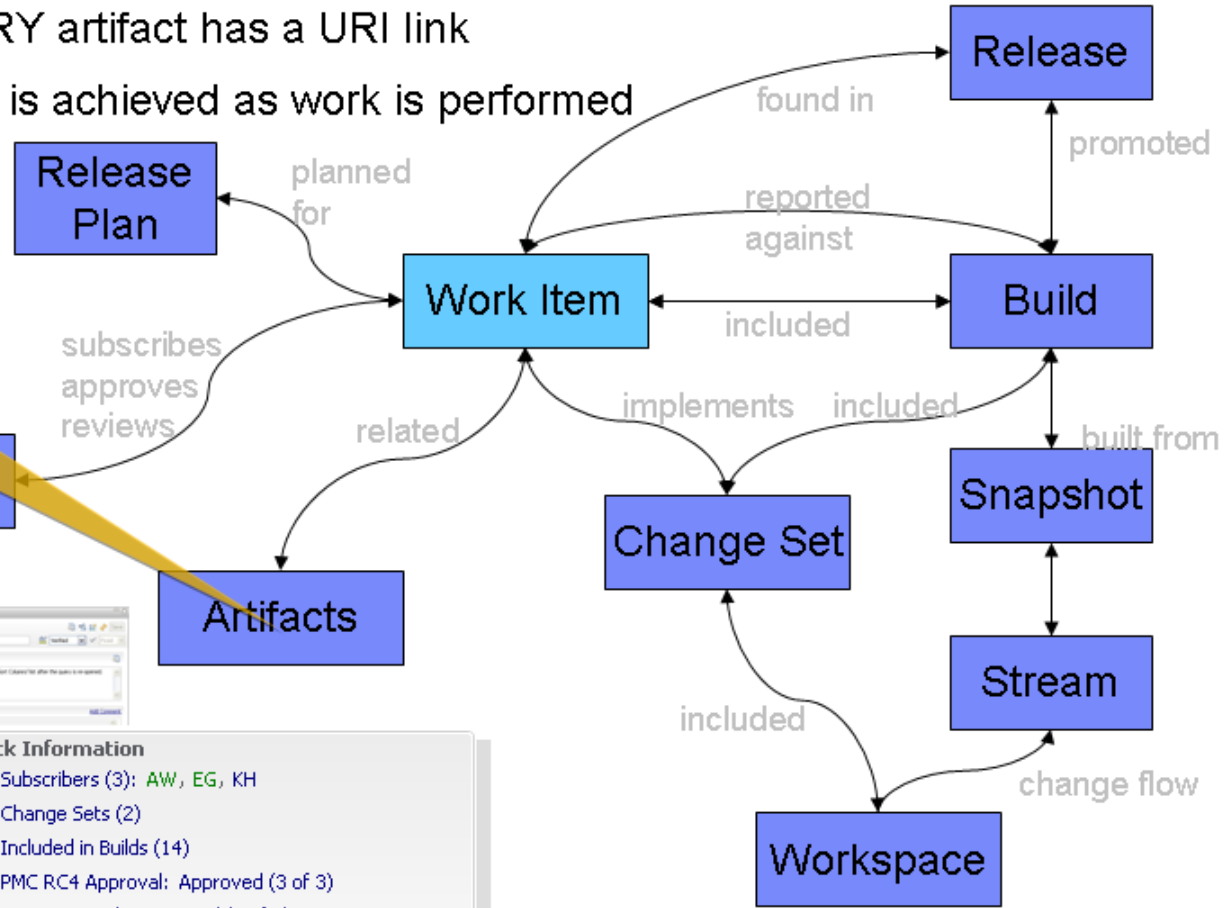
Name	Group	Show resolution
Closed	closed	<input checked="" type="checkbox"/>
In Progress	inprogress	<input type="checkbox"/>
New	open	<input type="checkbox"/>
Reopened	open	<input type="checkbox"/>
Resolved	closed	<input checked="" type="checkbox"/>
Triaged	open	<input type="checkbox"/>
Verified	closed	<input checked="" type="checkbox"/>

Automate traceability centered around work items

- In Team Concert EVERY artifact has a URI link
- Automated traceability is achieved as work is performed

Wind River or HW Artifacts related to software tasks or change requests

User



Quick Information

- Subscribers (3): AW, EG, KH
- Change Sets (2)
- Included in Builds (14)
- PMC RC4 Approval: Approved (3 of 3)
- RC4 Approval: Approved (1 of 1)
- RC4_Review: Approved (2 of 2)
- 0.6rc4_Verification: Approved (1 of 1)

Plans are composed of work items created by project managers and team members to manage work

The screenshot displays the JUnit Plan application interface. At the top, the title bar shows 'JUnit Plan [4.4 m2]'. Below the title bar, the application name 'JUnit Plan' is visible. The main area shows the 'Team Area: JUnit Team | Iteration: 4.4 m2 (8/22/08 - 9/21/08) | 10 Closed | 13 Open'. A progress bar indicates 'Progress: 13 / 186 | -51 h' and 'Estimated: 69%'. The interface is organized into three sections, one for each team member:

- Bill Cassavelli**: Closed items: 3 | Open items: 2. Progress: 9 / 11 | +8 h. Estimated: 100%.
 - Runner throws confusing exception "no runnable methods"...
 - Improve documentation for 4.4
 - 4.1 missing in maven-metadata.xml on ibiblio
 - Ignored method fails
 - timeout doesn't work properly for >=2 cases in junit4.3?
- Jason Mitchell**: Closed items: 2 | Open items: 5. Progress: 4 / 140 | -42 h. Estimated: 60%.
 - assertEquals throws NPE while comparing null elements
 - zxzxzx
 - Improve documentation for 4.4
 - JUnit version on the website is still 4.1
 - assertEquals array comparison doesnt handle nulls
 - assertThat signature does not match Matcher
 - CompositeRunner.filter incorrect if child throws NoTestsRema
- Markus Kent**: Closed items: 4 | Open items: 6. Progress: 0 / 35 | -14 h. Estimated: 67%.
 - assertArrayEquals misses differences
 - testCount hard-coded to 1 for childless Description
 - Provide improved Assertion syntax
 - Improve documentation for 4.4
 - shows green bar while assert false

The right sidebar contains several sections:

- Group by**: Owner
- Sort By**: Priority
- Bars**: Progress
- Exclude**: Future and past items, Resolved items
- More Filters...**: Tags...
- Related Work Items**: Show Backlog, Unplanned Closed Items
- Next Plans**: 4.4 rc0, 4.4 rc1

At the bottom, there are tabs for 'Overview', 'Planned Items', and 'Charts'.

Collaborate using Workitems and Plans

The image displays the Rational Team Concert (RTC) interface, illustrating collaboration through Work Items and Plans. Three callout boxes highlight key features:

- Various levels of work planning:** A callout pointing to the 'FASL Sprint 6 Development Plan' window, which shows a hierarchical view of tasks and their progress. The plan includes tasks like 'Dependency metadata collection in RTCz' and 'Need to know the function sets of FA to finalize FA SCM Config GUI on Build Definition Editor', each with associated progress bars and estimated completion times.
- Discuss work with members:** A callout pointing to the 'Defect 4273' window, which shows a detailed view of a work item. The 'Discussion' section contains comments from team members, such as 'So every file we create will have an own set things up. I suspect that the BF ag directory for IBMUSER in RACF we will...'. The 'Quick Information' section lists subscribers and dependencies.
- Collaborate in context:** A callout pointing to a chat window titled 'Jean-Yves Rigolet [started: 2:33:01 PM]'. The chat shows a message: 'Jean-Yves RL... JY, did you see the workitem 4392?'. This demonstrates how team members can communicate directly within the context of a specific work item.

The interface also shows a list of team members and their work item counts, such as 'Liam Doherty' with 15 closed and 6 open items, and 'Nicolas Dangeville' with 3 closed and 5 open items. The bottom of the screen shows a 'No Current Work' status.

Check the project status and health

The screenshot displays the IBM Rational Team Concert dashboard for the 'FASL Scrum' project. The interface includes a navigation bar with options like 'Dashboards', 'Project Areas', and 'Work Items'. A central panel shows the 'FASL Scrum' overview with a 'Welcome to FASL Scrum' message and a list of team members. A 'Burndown' chart is visible, showing the remaining work over time. A 'Project Activity' bar chart shows changes in work items. A 'Team communication' callout points to a 'Sprint 6 Development phase in progress' section. A 'Various project health dashboards' callout points to the overall dashboard layout. A 'Burndown charts' callout points to the 'Sprint Burndown' chart.

Team communication

Various project health dashboards

Burndown charts

Sprint 6 Development phase in progress

- Dates: 5/29-6/8
- Beta 3 preparation 6/23 - DCUT
- 6/24 - 6/30 FVT
- 6/29 - 6/30 Documentation Review
- 7/1 - Beta code delivery to jazz.net
- 7/8 - Beta 3 customer availability

Sprint Burndown

Team Area: FASL Scrum | Iteration: Sprint 6 Development (5/29/09 - 6/23/09) | 27 Open

Hours

Date	Hours
5/29/09	1500
6/5/09	1450
6/12/09	1350
6/19/09	1200
6/26/09	500

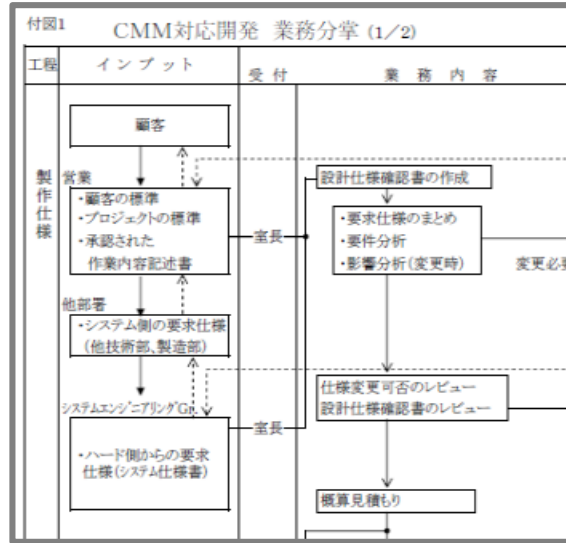
Project Activity

Change set additions, modifications and deletions

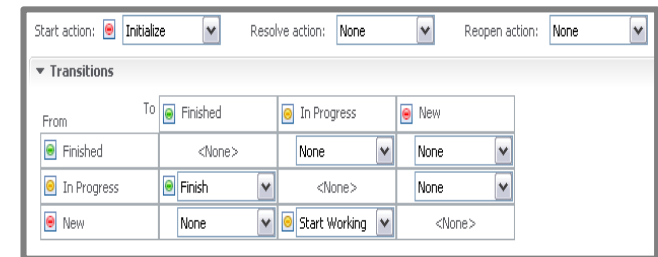
Date	Count
5/29/09	11
6/5/09	11
6/12/09	18
6/19/09	8
6/26/09	8

How did they do it?

Validate a task for ISO 26262 and A-SPICE



- Create task as work item



- Define a task for each step.
- Identify a set of input work products
- Identify a set of output work products

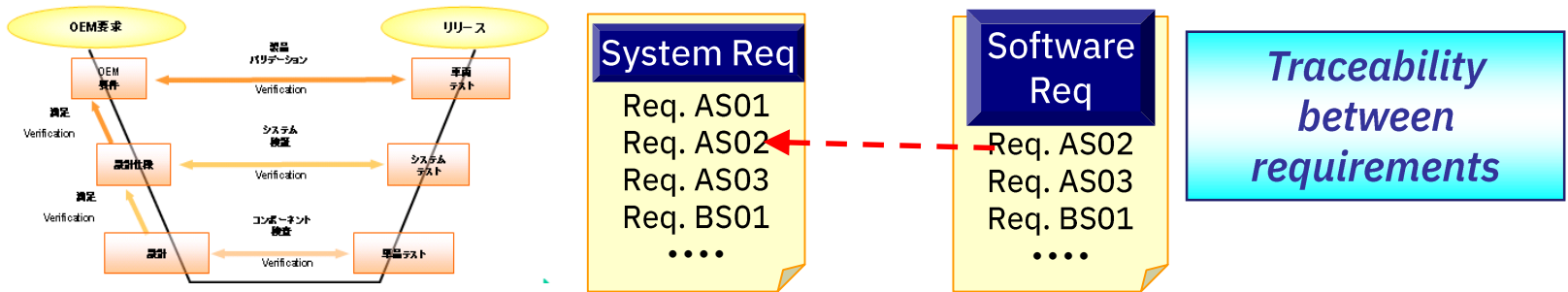


- Define where work product is stored
- Create link to each work product from the task

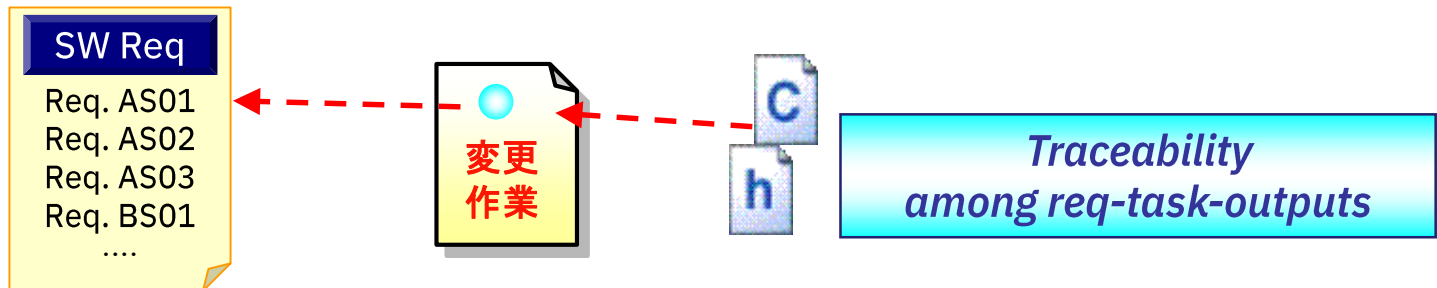
Meeting ISO 26262 and A-SPICE

1. Traceability between work products

- Req \leftrightarrow Design \leftrightarrow Testing, artifacts in the *safety lifecycle*.

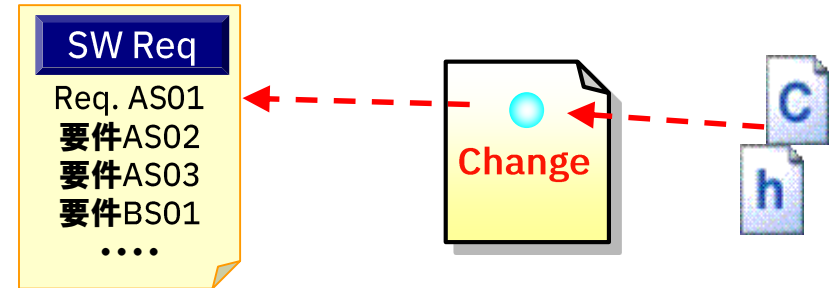


2. Linking tasks with work products



Linking to work products from Task

- Connectivity between,
tasks - requirements
tasks - outputs



開発リリース1 - フェーズ1 [フェーズ1] x

開発リリース1 - フェーズ1

所有者: ジェイテクト | 反復: フェーズ1 (10/12/14 - 11/01/14) | 完了: 9 | 未完了: 60

完了項目: 3 | 未完了項目: 60

進行状況: 24/130時間

見込み残り: 38%

製作仕様	0/338時間	新規	割り当てなし	778		
外部仕様	24/488時間	新規	割り当てなし	779		
外部仕様書作成	--	2日	新規	梅田	ID67	839
外部仕様書ピアレビュー	--	4時間	新規	梅田	ID68	840
マイルストーンレビュー-22	--	4時間	新規	豊住	ID69	799
機能仕様	0/498時間	新規	割り当てなし	800		
機能仕様書作成	--	2日	新規	割り当てなし	ID101	806
機能仕様書ピアレビュー	--	1日	新規	割り当てなし	ID102	807
マイルストーンレビュー-23	--	4時間	新規	割り当てなし	ID103	808
設計レビュー(マイルストーン)	--	4時間	新規	割り当てなし	ID104	809

編集 | コピー

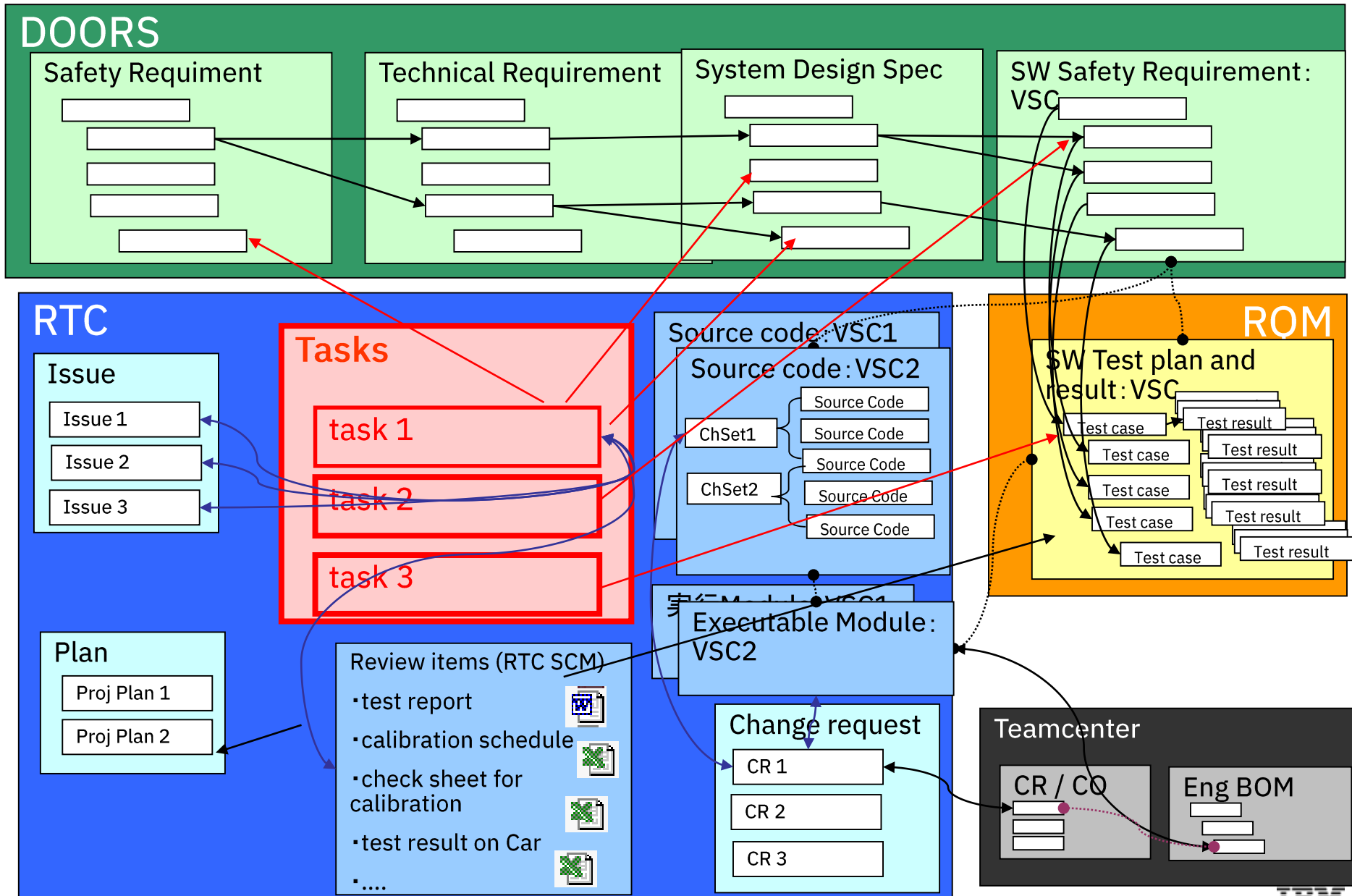
アクション

再ソート

除外

Connection between Req ID and Requirement

Data linkage among Tools



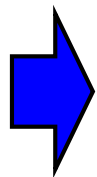
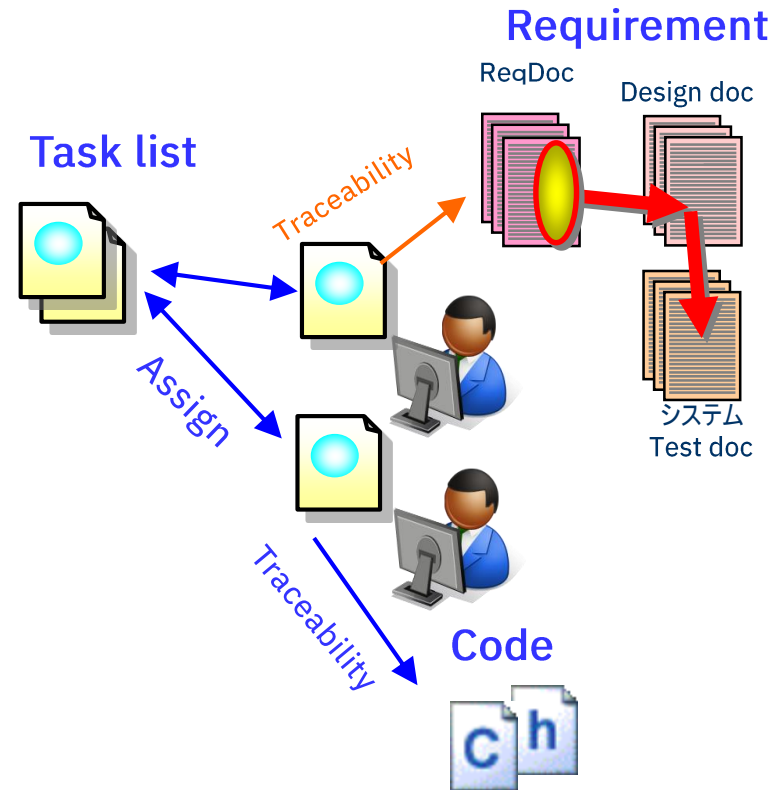
Validation

Existing process can be mapped to a set of tasks of RTC.

1. Assigning a task to each person is natural.
2. Accessing work products from a task is easy and efficient.

Process created is in compliance with ISO 26262 and Automotive SPICE.

1. RTC can *keep traceability*.
 - Requirements and task items
 - Task items and outputs
2. RTC enables to *execute right tasks in right process*
 - Follow the rules
 - Status can be captured in the process



Shift from artifact focus to task focus.

Integrated Code Review

- Annotated review with review comments
- Add *review comments* that can be stored along with the *change set* and attached to a *work item*

The screenshot illustrates the integrated code review workflow in an IDE. On the left, the 'Issues' sidebar lists several items, including '3: add copyri...', '2: Use our testCase class ...', and '1: add vector to the t...'. The main editor displays Java code for 'AllTests.java', with line 4, '`import junit.framework.TestSuite;`', highlighted in pink and marked with a red 'X' icon. A 'Code Review Summary' popup is open, showing the following details:

- Code Review Summary**
- Issue Summary**
 - Open: 3
 - Open & Must Fix: 3
 - Resolved: 0
- Issues (showing only mustfix issues)**
 - General Issues
 - add copyright
 - /JUnit Examples/src/com/ibm/team/junit/examples/AllTests.java
 - Seen by: Jason Mitchell, Markus Kent, Rick Maludzinski
 - Use our testCase class (line 4)
 - add vector to the test suite (line 14)
 - /JUnit Examples/src/com/ibm/team/junit/examples/Math.java
 - Seen by: Jason Mitchell


A 'Save' button is visible at the bottom right of the popup.


Agenda

1. ISO 26262 and Automotive SPICE
2. Key concepts for the standard compliance
3. Traceability in Requirements
4. Task and Process
5. Out of the Box Template for ISO 26262
6. Model based development and testing
7. Summary

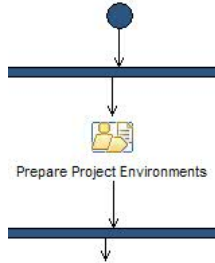
Rational Method Composer

- **Work product:** what is produced 

- **Task:** how to perform the work 

- **Role:** who performs the work 

- **Activity:** WBS and workflow 



The screenshot displays the Rational Method Composer interface. The left sidebar shows a tree view of the project structure, including 'Team (IBM)', 'Practices', 'Roles Sets', 'Enterprise', 'Software Development', 'Architecture', 'Configuration and Change Management', 'Development', 'Project Management', 'Team Lead', 'Requirements', 'Testing', 'Environment', 'Tasks', 'Work Products', 'Guidance', 'Tools', and 'Release Info'. The main content area is titled 'Role: Team Lead' and provides a detailed description of the role's responsibilities. It includes a 'Main Description' section with a list of tasks and a 'Relationships' section showing the role's interactions with other tasks and work products.

Role: Team Lead

This role leads the planning of the project, coordinates interactions with the stakeholders, and keeps the project team focused on meeting the project objectives.

Role Sets: Basic Roles, Project Management

Relationships

The role performs the following tasks:

- Assess Results
- Assign Change Request
- Identify and Assess Risks
- Manage Iteration
- Plan Iteration
- Plan Project

The role is responsible for the following work products:

- Iteration Plan
- Project Plan
- Risk List
- Work Items List

Additionally Performs

- Develop Technical Vision
- Envision the Architecture
- Refine the Architecture

Modifies

- Risk List

Main Description

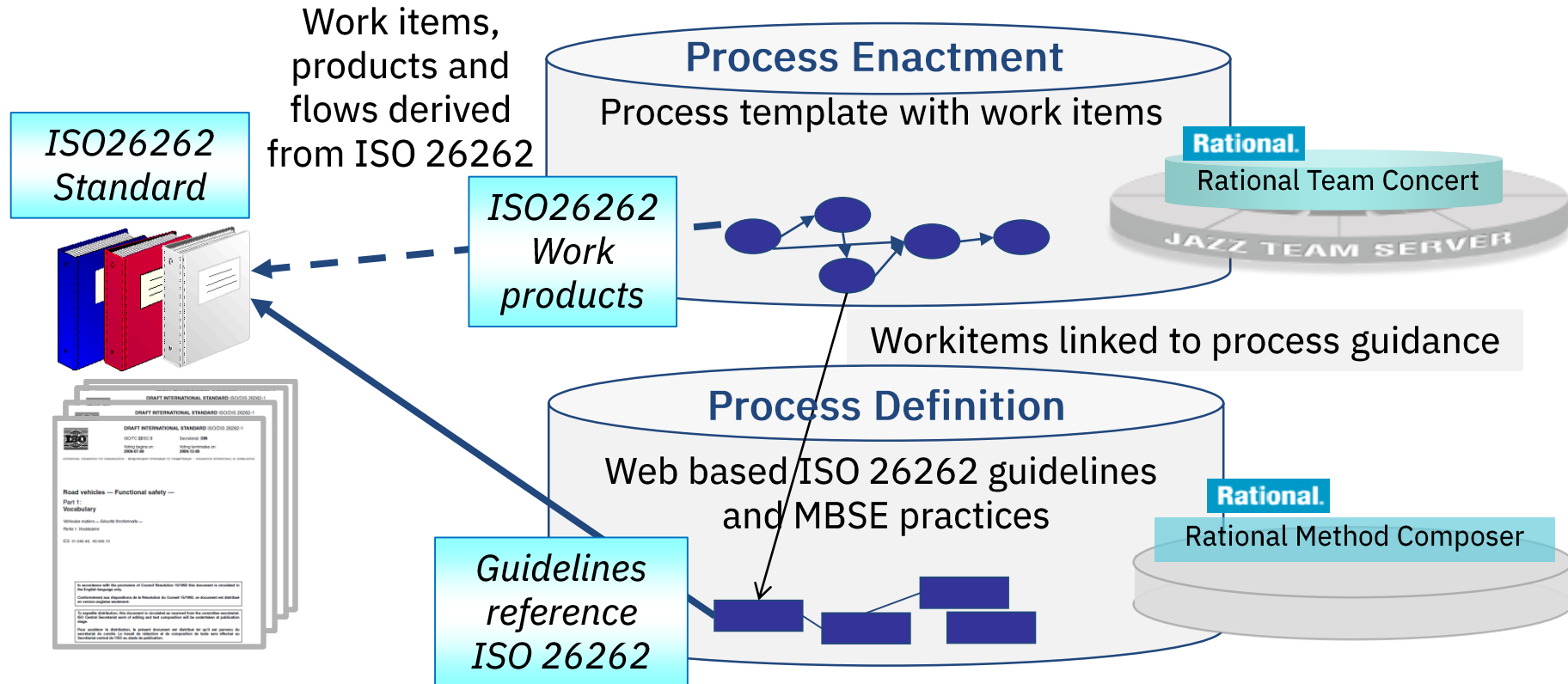
The person in this role:

- Coaches the team to drive a successful outcome of the project and the acceptance of the product by the customer
- Is accountable for the outcome of the project and the acceptance of the product by the customer
- Is responsible for the evaluation of project's risks and for controlling those risks through mitigation strategies
- Applies management knowledge, skills, tools, and techniques to a broad range of tasks to deliver the desired result for a particular project in a timely manner

Innovation for a smarter planet

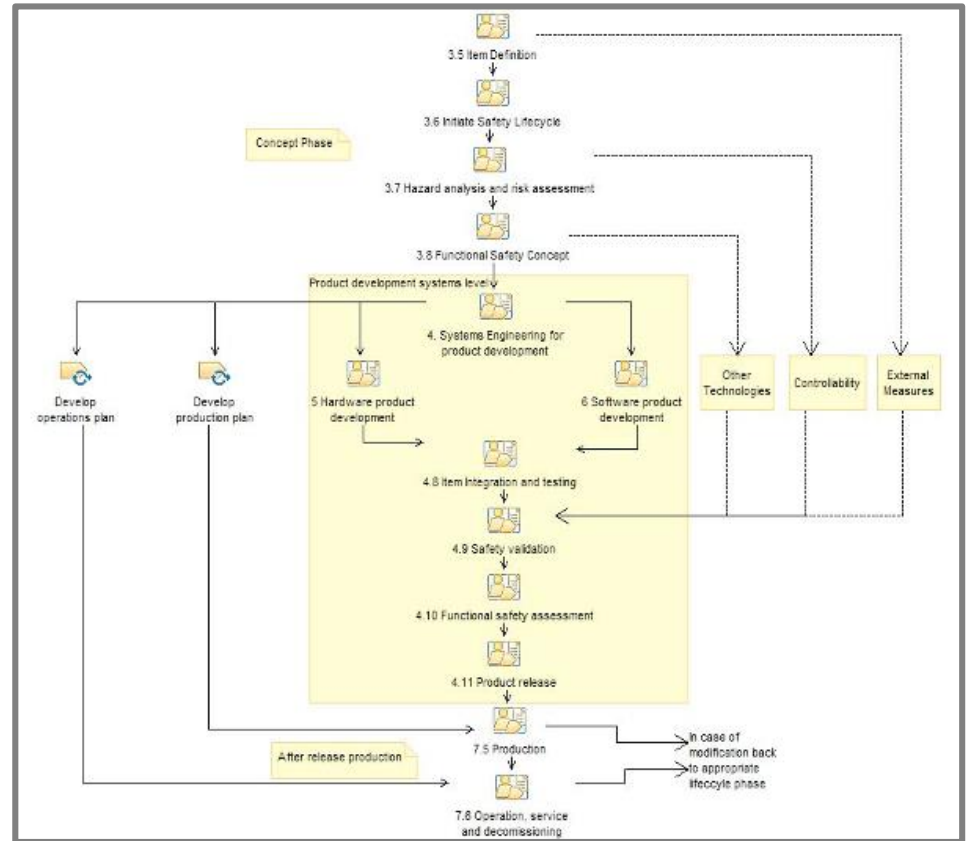
Out-of-the-box ISO 26262 Project Workflows

- Supports all core *processes* and *work products* defined in the *standard*
- *Process template* implemented in Rational Team Concert
- *Guidance* and *practices* implemented in Rational Method Composer



ISO 26262 in Rational Method Composer

- **RMC** captures activities and flows
 - Flows reflect **ISO 26262**
- Activities and flows Reflected in **RTC process template**
 - RTC allows project managers to plan the work and assign tasks to teams
- Drill down through activities for more detail
 - Workflows
 - Task descriptions
 - Incoming and outgoing work products
 - Applicable roles



ISO 26262 work item templates

- *Work item templates* are modularized, it covers
 - Separate safety management section
 - Main concept phases
 - Separation of production and operation activities
 - Aspects of supporting processes

Create Work Items from a Template

Select a Template

Choose a template from the list to create its work items automatically.

Project Area: ISO 26262 Demo2

Available Templates:

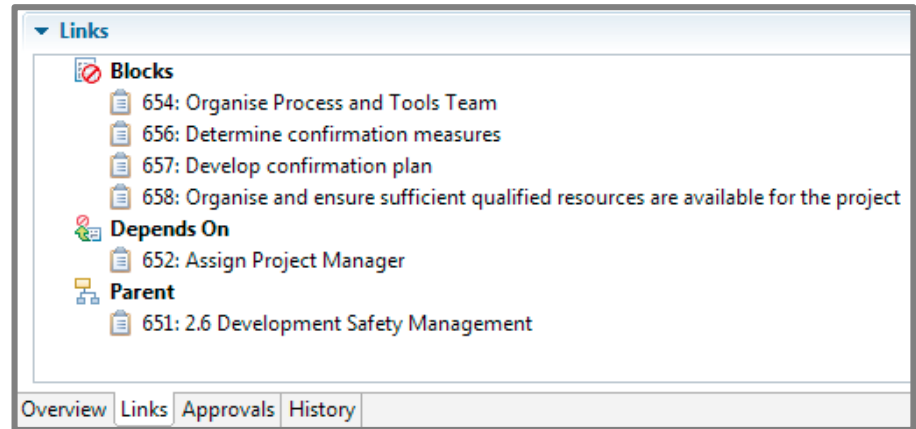
- 2.0 Gather Evidence of Staff Qualifications and Experience
- 2.5 Management of Overall Safety
- 2.6 Safety Management through the develop phase
- 2.7 Safety Management through Production and Operation
- 3 Concept Phase
- 4 Systems Engineering for Product Development
- 4 Systems Integration and Testing
- 5 Hardware Product Development
- 6 Software Product Development
- 7.5 Production

Found 12 work items - 2.6 Safety Management through the develop phase

Id	Status	P	S	Summary	Owned By	Created By
651	New			2.6 Development Safety Management	Unassigned	Graham
652	New			Assign Project Manager	Unassigned	Graham
653	New			Assign Safety Manager	Unassigned	Graham
654	New			Organise Process and Tools Team	Unassigned	Graham
655	New			Develop functional safety assessment plan	Unassigned	Graham
656	New			Determine confirmation measures	Unassigned	Graham
657	New			Develop confirmation plan	Unassigned	Graham
658	New			Organise and ensure sufficient qualified resources are a...	Unassigned	Graham
659	New			Develop safety case	Unassigned	Graham
660	New			Develop safety plan	Unassigned	Graham
661	New			Tool Environment Setup	Unassigned	Graham
662	New			Project independent tailoring of the safety cycle	Unassigned	Graham

ISO 26262 work items

- Individual activities are children of main task
- Individual activities are linked together in flows
- Contain basic description that links to details of task



ISO 26262 Demo2 652: Assign Project Manager 653: Assign Safety Manager Task: Assign Safety Manager

https://localhost:9444/iso26262/process.compliance.iso26262.base-ibm/capabilitypatterns/Assign%20Safety%20Manager_89B943B3.html

Task: Assign Safety Manager

Assign a suitable qualified person to be responsible for functional safety management during the item development.

Expand All Sections Collapse All Sections

Purpose

To find some to take responsibility for the planning of the safety activities and the maintenance of the safety plan

Back to top

Relationships

Roles	Primary: <ul style="list-style-type: none">Project Manager	Additional: <ul style="list-style-type: none">Human Resources Officer	Assisting:
Inputs	Mandatory: <ul style="list-style-type: none">Employee documented experienceEmployee Qualification Certification	Optional: <ul style="list-style-type: none">None	External: <ul style="list-style-type: none">None
Outputs	<ul style="list-style-type: none">Safety manager task offer		

Back to top

Main Description

Assign a suitable qualified person to be responsible for functional safety management during the item development.

Back to top

Agenda

1. ISO 26262 and Automotive SPICE
2. Key concepts for the standard compliance
3. Traceability in Requirements
4. Task and Process
5. Out of the Box Template for ISO 26262
6. Model based development and testing
7. Summary

Modeling recommended for Higher ASIL

Table 1 — Specifying safety requirements

Methods		ASIL			
		A	B	C	D
1a	Informal notations for requirements specification ^a	++	++	+	+
1b	Semi-formal notations for requirements specification ^a	+	+	++	++
1c	Formal notations for requirements specification ^a	+	+	+	+

^a In the case of model-based development, safety requirements at each level need to be described with the same combination of methods used for the software safety requirements specification.

Table 3 — Notations for software architectural design

Methods		ASIL			
		A	B	C	D
1a	Informal notations	++	++	+	+
1b	Semi-formal notations	+	++	++	++
1c	Formal notations	+	+	+	+

IBM Rational Rhapsody

Formal Notation for Requirement Specification

Safety driven Systems Design:

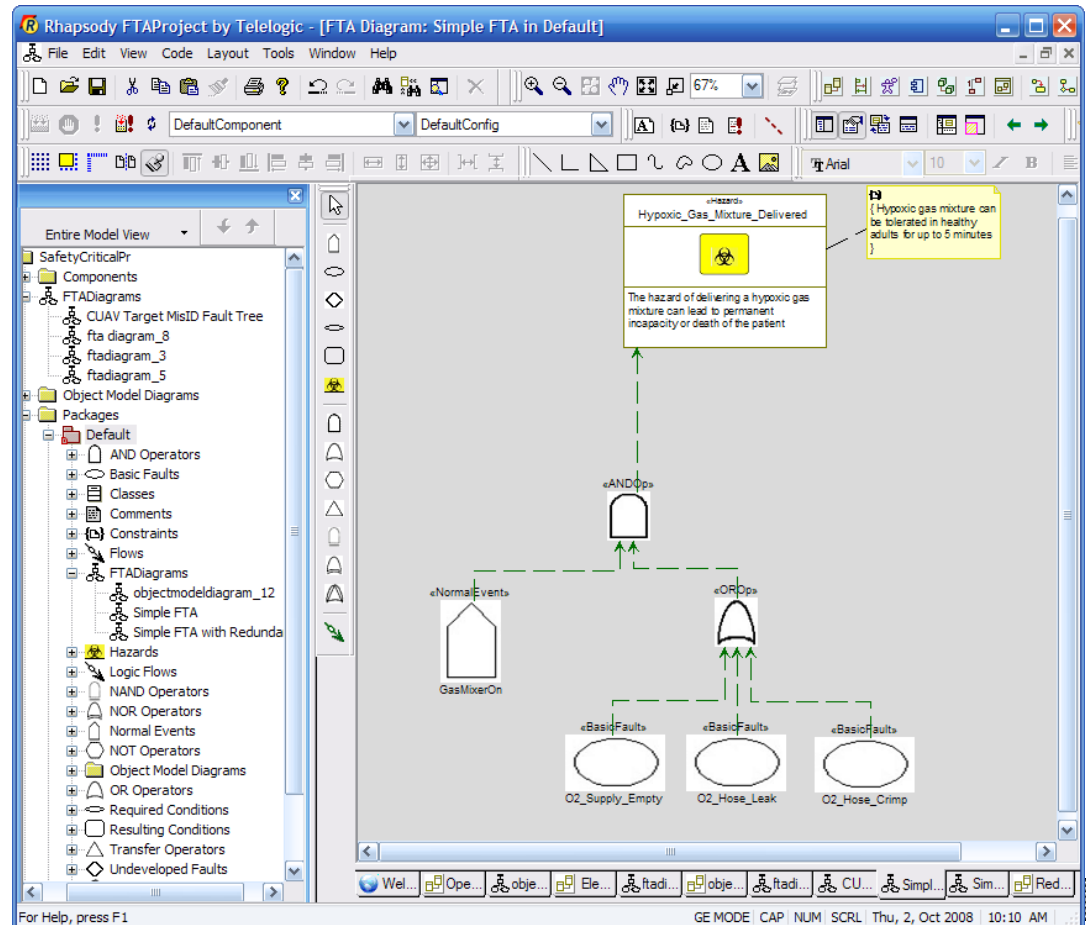
- Understand Safety requirements early in the development cycle
- Design safety into the system to begin with

Simulation, Execution and Automation:

- Identify and eliminate errors early when they are less costly to fix
- Visually communicate intended behavior to customers to deliver the right product
- Perform design level debugging

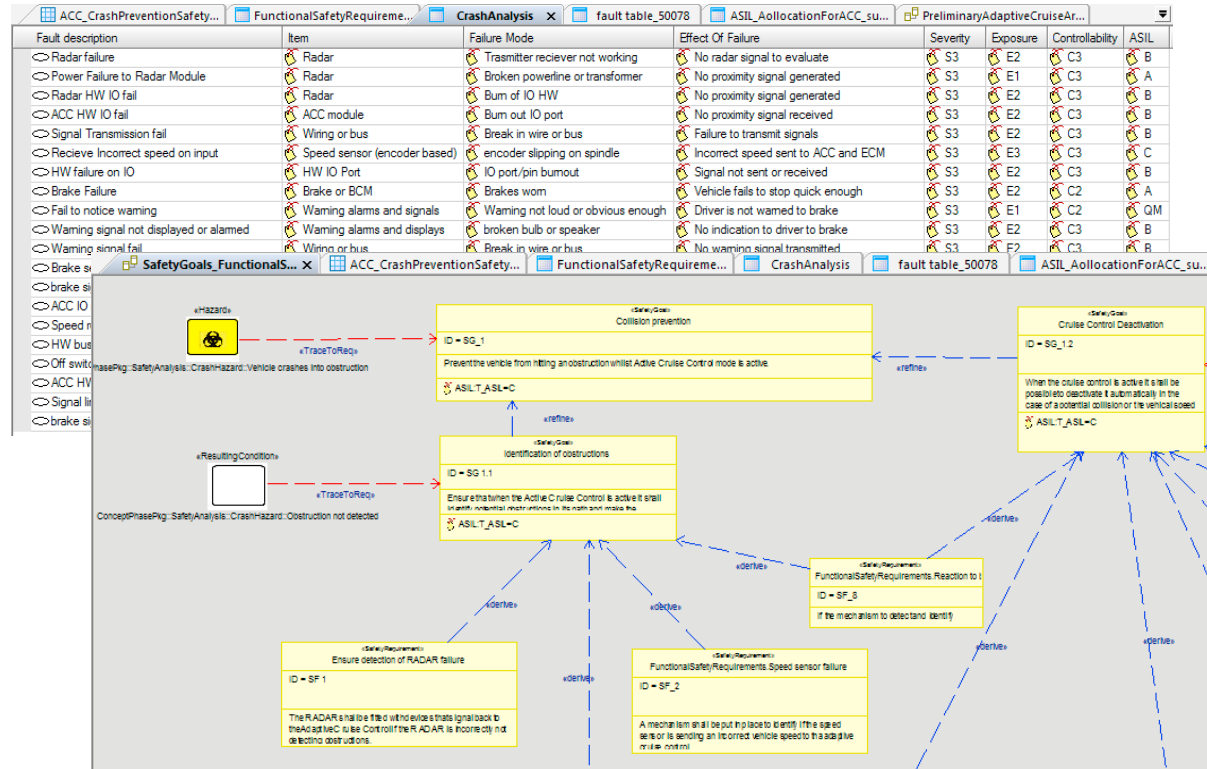
Safety-Critical Profile

- Brings together model based systems and software development with safety analysis
 - Safety Analysis profile in Rhapsody allows safety analysis to be carried out
- Covers
 - FTA diagrams
 - Hazard analysis table view
 - Constraint table view
 - Derived safety based requirements



Automotive Safety Analysis Profile

- Extends the original safety analysis profile
- Extended FMEA table into an ASIL table
- Captures ISO 26262 specific concepts
 - SafetyGoal
 - SafetyRequirement
 - ASILs for elements
- Captures Safety Requirements
 - ASIL
 - System/Subsystem Allocation
 - Requirement type



Summary	Element type	ID	Specification	ASIL	Allocated to	Req Type
Collision prevention	SafetyGoal	SG_1	Prevent the vehicle from hitting an obstruction whilst Active	C		
Identification of obstructions	SafetyGoal	SG_1.1	Ensure that when the Active Cruise Control is active it shall	C		
Cruise Control Deactivation	SafetyGoal	SG_1.2	When the cruise control is active it shall be possible to	C		
Ensure detection of RADAR failure	SafetyRequire...	SF_1	The RADAR shall be fitted with devices that signal back to the	A	RdRArbitration	Functional Safety Requirei
RADAR Signal failure	SafetyRequire...	SF_3	A system shall be developed that will recognise if the RADAR	B	RdRArbitration	Functional Safety Requirei
Loss of signals from Brake and ECM	SafetyRequire...	SF_4	A system shall be developed that will recognise if any signals	B	BrakeControlModule, VehicleDynamics	Functional Safety Requirei
Comupt signals	SafetyRequire...	SF_5	A system shall be developed that will recognise corruption of	A	RdRArbitration,	Functional Safety Requirei
Speed sensor failure	SafetyRequire...	SF_2	A mechanism shall be put in place to identify if the speed sensor	C	VehSpdArbitrater,	Functional Safety Requirei
Driver warnings	SafetyRequire...	SF_6	A mechanism shall be put in place to detect and alarm the driver	B	DrvInpSignalArbitration,	Functional Safety Requirei
Driver signals to ACC	SafetyRequire...	SF_7	A mechanism shall be put in place to ensure that any	B	DrvInpSignalArbitration	Functional Safety Requirei
Reaction to bad signals	SafetyRequire...	SF_8	If the mechanism to detect and identify corrupt or missing signals	C	DrvInpSignalArbitration, RdRArbitration, VehSpdArbitrater	Functional Safety Requirei
If brake sensor pad warnings	SafetyRequire...	SF_9	If the brake is worn of the braking sensor broken a mechanism	C	BrakeControlModule	Functional Safety Requirei
ACC switch Off	SafetyRequire...	SF_10	If the switch to notify the ACC that it has been turned off fails	B	InstrumentCluster	Functional Safety Requirei

Rhapsody Kit for ISO 26262 and IEC 61508

IBM Rational Rhapsody Kit for ISO 26262 and IEC 61508 Overview

IBM Rational Rhapsody Reference Workflow Guide

IBM Rational Rhapsody TestConductor Add On Reference Workflow Guide

IBM Rational Rhapsody TestConductor Add On Safety Manual

TÜV SÜD Certificate for IBM Rational Rhapsody TestConductor Add On

TÜV SÜD Report to the Certificate for IBM Rational Rhapsody TestConductor Add On

IBM Rational Rhapsody TestConductor Add On Validation Suite (optional component of the kit)

SXF Framework (C++) **SMXF Framework (C)**

SXF / SMXF Validation Suites

CERTIFICATE
No. Z10 12 07 81878 001

Holder of Certificate: International Business Machines Corp.
5 Technology Park Dr.
Westford MA 01886
USA

Factory(ies): 73219

Certification Mark: 

Product: Software Tool for Safety Related Development

Model(s): IBM Rational Rhapsody TestConductor Add On

Parameters: The model-based verification tool is classified as T2 offline support tool according to IEC 61508-4. It is qualified to be applied in safety related software development for all SIL levels according to IEC 61508, and for all ASIL Levels according to ISO 26262. The report IW84460C is a mandatory part of this certificate.

Tested according to: IEC 61508-3:2010
ISO 26262-8:2011

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.: IW84460C

Date: 2012-08-13
Page 1 of 1

(Matthias Ramckl)



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

ZERTIFIKAT ◆ **CERTIFICATE** ◆ **증명서** ◆ **СЕРТИФИКАТ** ◆ **CERTIFICADO** ◆ **CERTIFICAT**

Model Driven Testing

IBM Rhapsody Test Conductor

- Common Browser
- Requirements linked to test cases
- Easy navigation between Design and Test artifacts;
- Design and Test - Always in sync
- Automatically generated test execution reports

The screenshot displays the main interface of the IBM Rhapsody Test Conductor. On the left, there is a 'Design Artifacts' tree view showing a hierarchy of packages, classes, and diagrams. In the center, the 'Test Artifacts' tree view shows test cases and test components. On the right, a 'Test Context Result' window displays 'Test Execution Reports' with a table of test results. The interface is designed for easy navigation between design and test artifacts.

Test Execution & Test Reporting

Design & Test Processes Fully Integrated

This screenshot shows the test execution and reporting interface. It includes a 'Test Case Result' window for 'Test Case: SD_BB_TST001' with execution details. A 'Test Execution Report' window shows a table of test results for 'SD Instance: 'check stopwatch start''. A 'Test Case Diagram' window shows a sequence diagram with actors and messages, including a message 'showWaveband(aBand)' and a note 'tune(value=87500): Operation Call - In Parameter values do not match.' The interface also shows a 'Test Case List' window with columns for ID, Name, Status, and Progress.

ID	Name	Status	Progress
SD Instance: 'check stopwatch start'	check stopwatch start	passed	100% (1/1)
SD Instance: 'check stopwatch stop'	check stopwatch stop	passed	100% (1/1)
SD Instance: 'check stopwatch start'	check stopwatch start	passed	100% (1/1)
SD Instance: 'check stopwatch stop'	check stopwatch stop	passed	100% (1/1)
SD Instance: 'check stopwatch start'	check stopwatch start	passed	100% (1/1)
SD Instance: 'check stopwatch stop'	check stopwatch stop	passed	100% (1/1)

- Develop Test Case
- Develop Test Case
- Execute/Report on Test Execution
 - Inputs to SUT and stubs behaviours are played out automatically
 - Unexpected behaviours are highlighted
 - Test Execution Reports can be customized to match company/project standards

Jaguar Land Rover cuts software validation time by 90 percent

IBM Rational software helps manage near-real-time requirements updates for 6,000 developers worldwide

The need:

Jaguar Land Rover plc wanted to implement a standard requirements methodology that would enable it to manage increasingly complex requirements and speed new features to market.

The solution:

The company implemented a suite of IBM® Rational® software to create a new requirements management and modeling system.

The benefits:

- Reduces the time required to fully validate software by more than 90 percent, from six to eight weeks to only three days
- Decreases bug-cause detection time by more than 99 percent, from three days of human intervention to 30 seconds
- Accelerates time to market for in-vehicle entertainment systems and helps increase innovation

Solution components:

- IBM® Rational® ClearCase®
- IBM Rational ClearQuest®
- IBM Rational DOORS®
- IBM Rational Rhapsody®

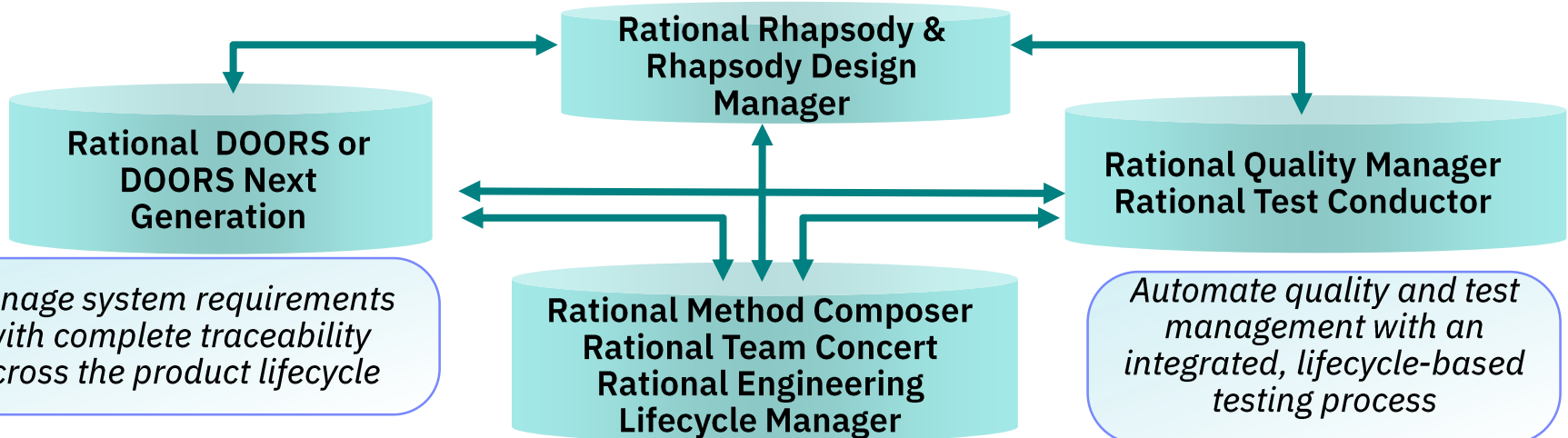


Agenda

1. ISO 26262 and Automotive SPICE
2. Key concepts for the standard compliance
3. Traceability in Requirements
4. Task and Process
5. Out of the Box Template for ISO 26262
6. Model based development and testing
7. Summary

IBM Rational tools for Automotive

Use modeling to validate requirements, architecture and design earlier in the development process – including Simulink integration, autocode generation and automated test case generation; use in the same way models for FMEA, FTA

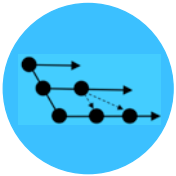


Manage system requirements with complete traceability across the product lifecycle

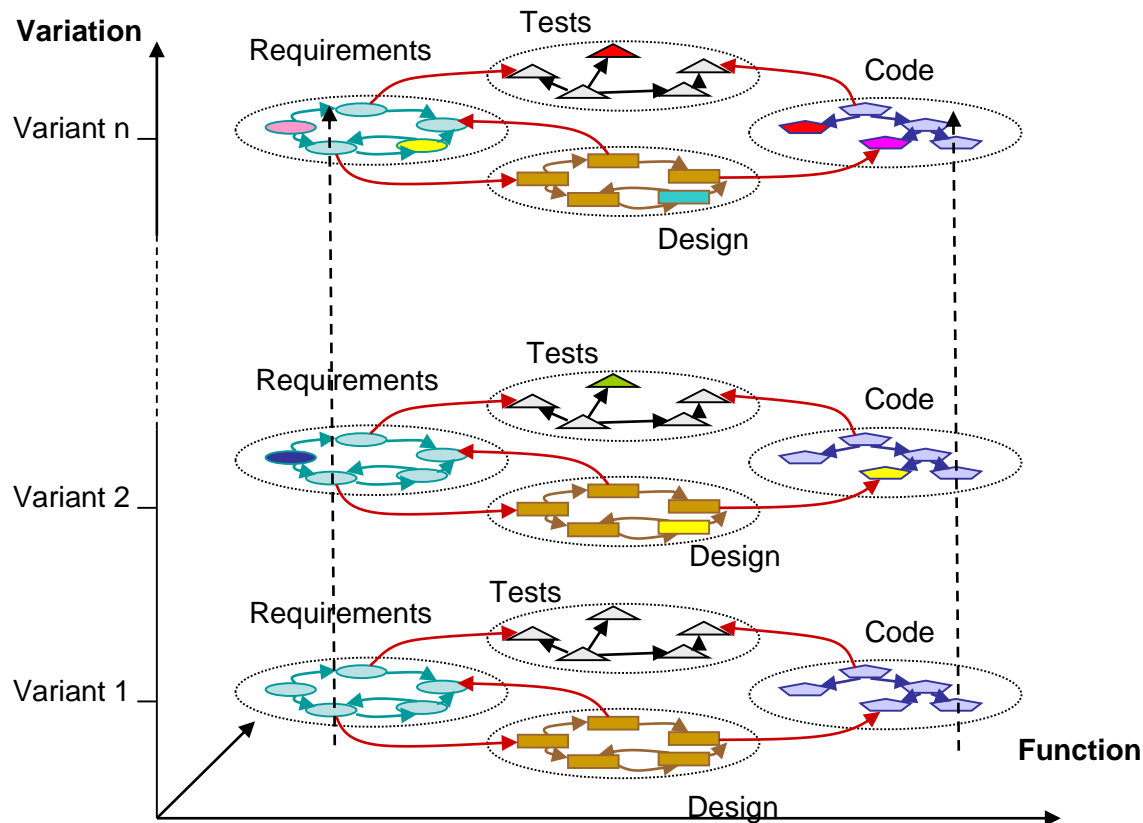
Automate quality and test management with an integrated, lifecycle-based testing process

Manage collaborative systems lifecycle management across development teams and engineering disciplines with Automotive data model based on Automotive SPICE & ISO 26262 process template and compliance

IBM Rational tools for Product Line Engineering (Reuse) using Multi-stream



- Managing product variants as branches of engineering components and artifacts across the lifecycle
- RELM, RDNG, RTC, RQM, Rhapsody + DM



Partnership

- **SPID**
 - Consulting on ISO 26262 and Automotive SPICE
 - Rational Tools and Support
- **IBM**

Summary

- ISO 26262 and Automotive SPICE present challenge for automotive companies, and meeting the challenge can be a major differentiator.
- Key concepts to adopt are traceability, process improvement and accountability.
- Using tools in systematic and coordinated way is critical in meeting the challenge.
- IBM Rational has rich set of tools and services to help customers with partnership with other companies to form ecosystem for solutions.

IBM

Robert Bosch GmbH – Implements continuous engineering platform for smarter, faster and safer vehicles

15% reduction

of effort and latency in communication

20% improvement

in steering through more objective progress and quality insight

Significantly less

effort and time in preparation and delivery of software changes

Solution components

- IBM® Rational® Design Manager
- IBM Rational DOORS®
- IBM Rational Engineering Lifecycle Manager
- IBM Rational Quality Manager
- IBM Rational Team Concert™



The transformation: To more efficiently and accurately develop intelligent, interconnected automotive products, Bosch needed to create a data-driven systems engineering platform. A suite of IBM Rational software helps the organization demonstrate compliance with industry standards such as ISO26262, ASPICE and AUTOSAR.

“By implementing a software and systems design platform founded on continuous integration and strategic reuse we improve our integration speed and increase development quality.”

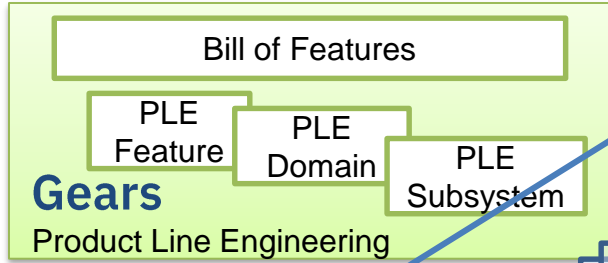
—Nico Maldener, senior project manager,
Robert Bosch GmbH



BOSCH

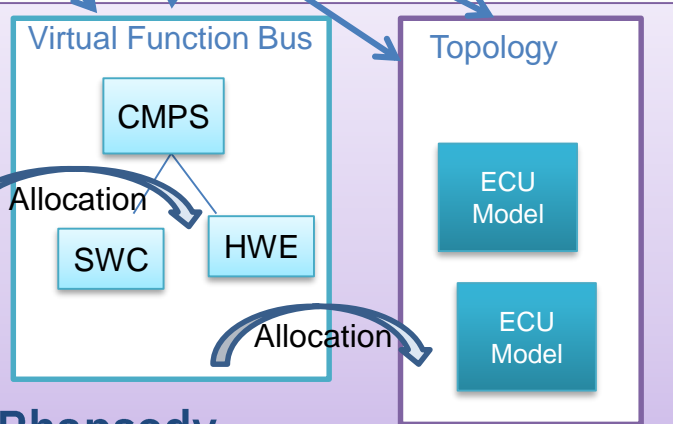
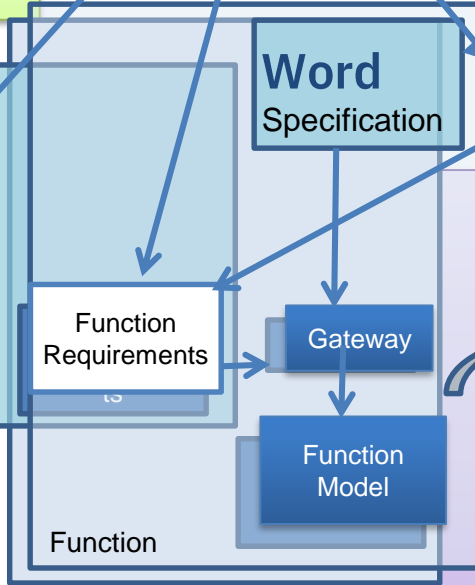
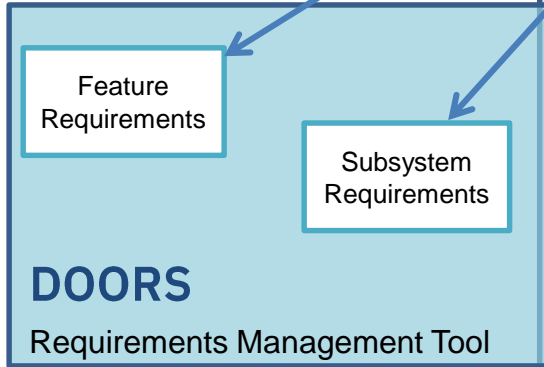
Invented for life
IBM

General Motors



- Feature Requirements
- Performance Requirements
- Functional Requirements
- Serial Data Messages
- Functional Allocation
- Calibrations
- Requirements Allocation Matrix

-
- Specifications**
- Component Requirements
 - Interfaces
 - Serial Data Messages
 - Calibrations



- Requirement Allocation to System Design
- System Design Model Structures
- ECU Publishing Capability

Rational Publishing Engine

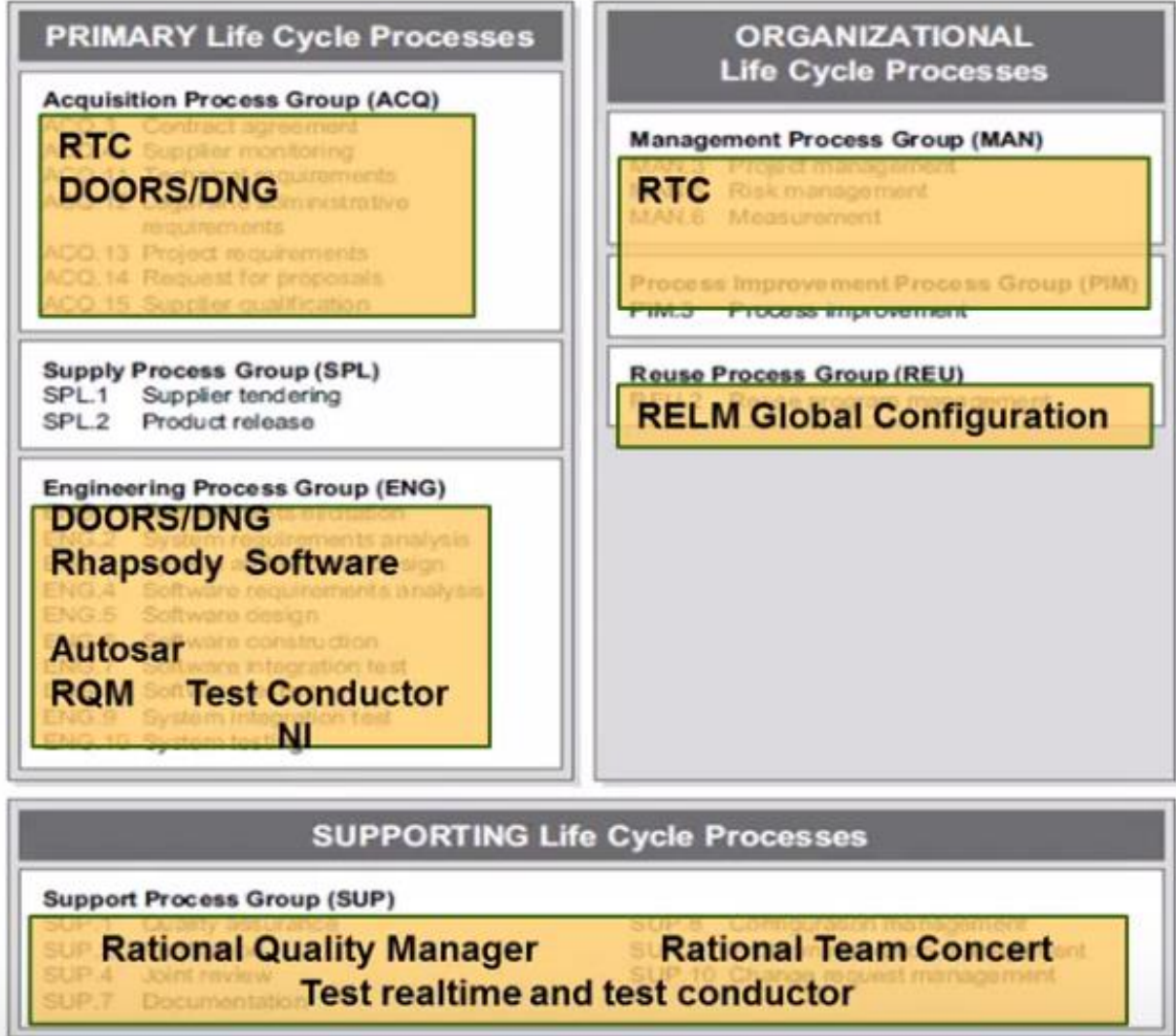
Rational Team Concert / CM Synergy



Tool and Practice coverage for ASPICE



- Have tool support for all of ASPICE base processes
- Have a lot of the "How" as part of the RMC practice libraries
 - SSE
 - DOORS etc
 - ISO 26262 practices
 - Existing CMMI practices



Continental AG

Challenge

- Continental identified a need for increased development efficiency
- There was a growing number of different and mostly independent development tools, contributing to a highly fragmented engineering environment
 - Limiting ability to move people, reuse assets, and increasing training costs
- Geographically dispersed teams have difficulty finding and accessing the latest engineering data
- Decided to establish an advanced platform for the development of automotive electronic systems such as body control units and keyless automobile access control systems.

Results

- Continental has deployed Rational Tokens as a licensing mechanism to provide flexibility in tool usage across the development lifecycle
- Continental is using IBM Rational DOORS for capturing, managing and exchanging requirements with its Clients
- Meeting AUTOSAR requirements in Rhapsody and integrating with Simulink
- The platform has increased the efficiency of the development teams by providing the right information in a structured way, and leveraging the advantages of a model-driven systems and software development approach.
- **Success: 1,000's of users – Growing to 2X current use in 2013**

“IBM Rational DOORS and Rational Rhapsody allow us to better manage the complex architectures of our products, and help us to prevent our development environment from becoming fragmented.”



- Gervin Fadda
- Body and Security business unit
- Continental Automotive Group



Panasonic Automotive

Challenge

- Greater Efficiency
 - Deconstruct project silo's and leverage reuse
- Picking up the Pace
 - Reducing Cycle Time & Increasing Productivity
- Manage Risk of Reducing Staff Per Project
 - Visibility and Action Driven.
- Staff is 100% burdened with no overhead
- Replicated the same tasks across multiple projects
- No real prioritization of projects / activities

Results

- Formalized the release process, tracking iterations and delivery
- Implemented formal change management with resource allocation data, prioritization
- Interlock with OEM via DOORS
- Limited the amount of rework
- Increased quality
- Ability to effectively prioritize projects / activities
- Better ability to estimate projects
- Higher staff moral
- XX% reduction in defects discovered in test
- XX% improvement in development efficiencies

- **Success: Growing > 100% in 2013**

Panasonic

Hella

Summary:

- Hella is using DOORS and Rhapsody successfully (ex StateMate Users)
- IBM Rational Design Partner Program Member
- Utilizing Rhapsody AUTOSAR and Matlab/Simulink Integrations
- Worked jointly with IBM Rational Partner INCHRON in order to add Timing Simulation Solution into the Toolchain

