IBM Software Group

Accelerating Safety-Critical Development with IBM Rational - ISO 26262 and DO-178 B

Graham Bleakley
Graham.bleakley@uk.ibm.com

Rational. software

Go to IBM

Innovation for a smarter planet

Safety standards

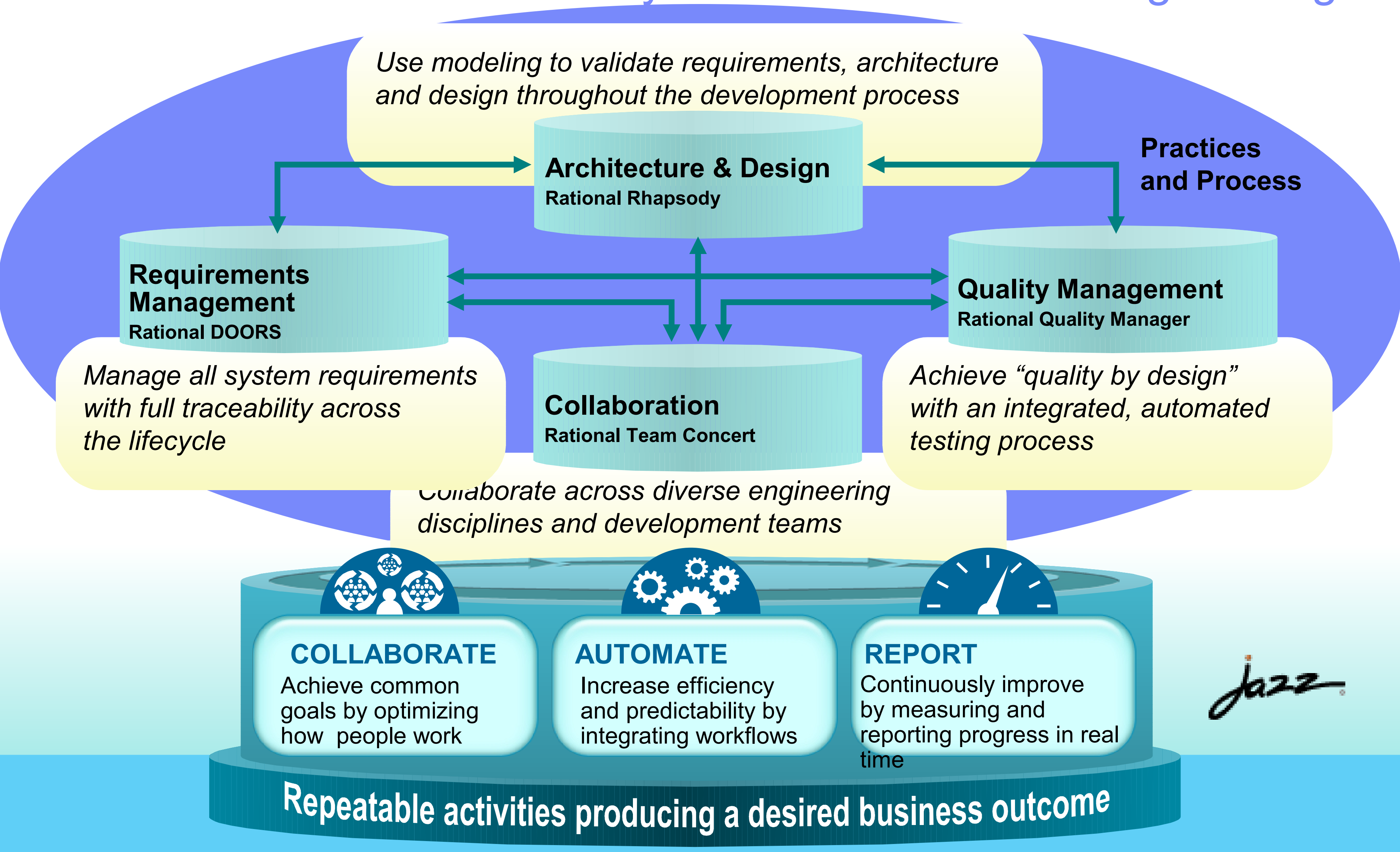
- Avionics/aerospace
  - **DO-178B** / ED-12B (RTCA/EUROCAE)
  - DO-178B is a widely accepted standard often used as a baseline for other certification efforts outside of avionics
- Medical
  - FDA 510(k) and IEC 60601

- Functional safety in process industry
  - IEC 61508
- Automotive
  - **ISO-26262** and MISRA-C
- Railway systems
  - EN50128 and EN50129
- Nuclear power plants
  - IEC 880

Development tools for Safety Critical Systems

- Support for:-
  - ISO 26262 (Functional Safety for Road Vehicles)
  - DO178B (Safety critical software for aerospace)
- Both support an integrated environment for the development of products that rely on these standards
  - Even though the standards are different in terms of application areas the tools that they use are identical
- Practice content on Rational Method Composer
  - Web-based content for describing processes
  - Provides guidance on the workflows, tasks and activities, required to produce the work products to help with compliance to these standards
  - Highly customizable to fit with the process and workflows of the organisation it is being deployed in
  - Provides guidance and tool mentors describing how to use Rational tools to aid in the development process.
- Process content in Rational Team Concert
  - Provides project management and governance of the process
  - Uses process templates derived from the practice content developed in Rational Method Composer
  - Standardizes the development process for these standards
  - Integrates seamlessly with various rational tools to enable and automate the process

IBM Rational Workbench for Systems and Software Engineering



Use modeling to validate requirements, architecture and design throughout the development process

Architecture & Design
Rational Rhapsody

Practices and Process

Requirements Management
Rational DOORS

Quality Management
Rational Quality Manager

Manage all system requirements with full traceability across the lifecycle

Collaboration
Rational Team Concert

Achieve "quality by design" with an integrated, automated testing process

Collaborate across diverse engineering disciplines and development teams

COLLABORATE
Achieve common goals by optimizing how people work

AUTOMATE
Increase efficiency and predictability by integrating workflows

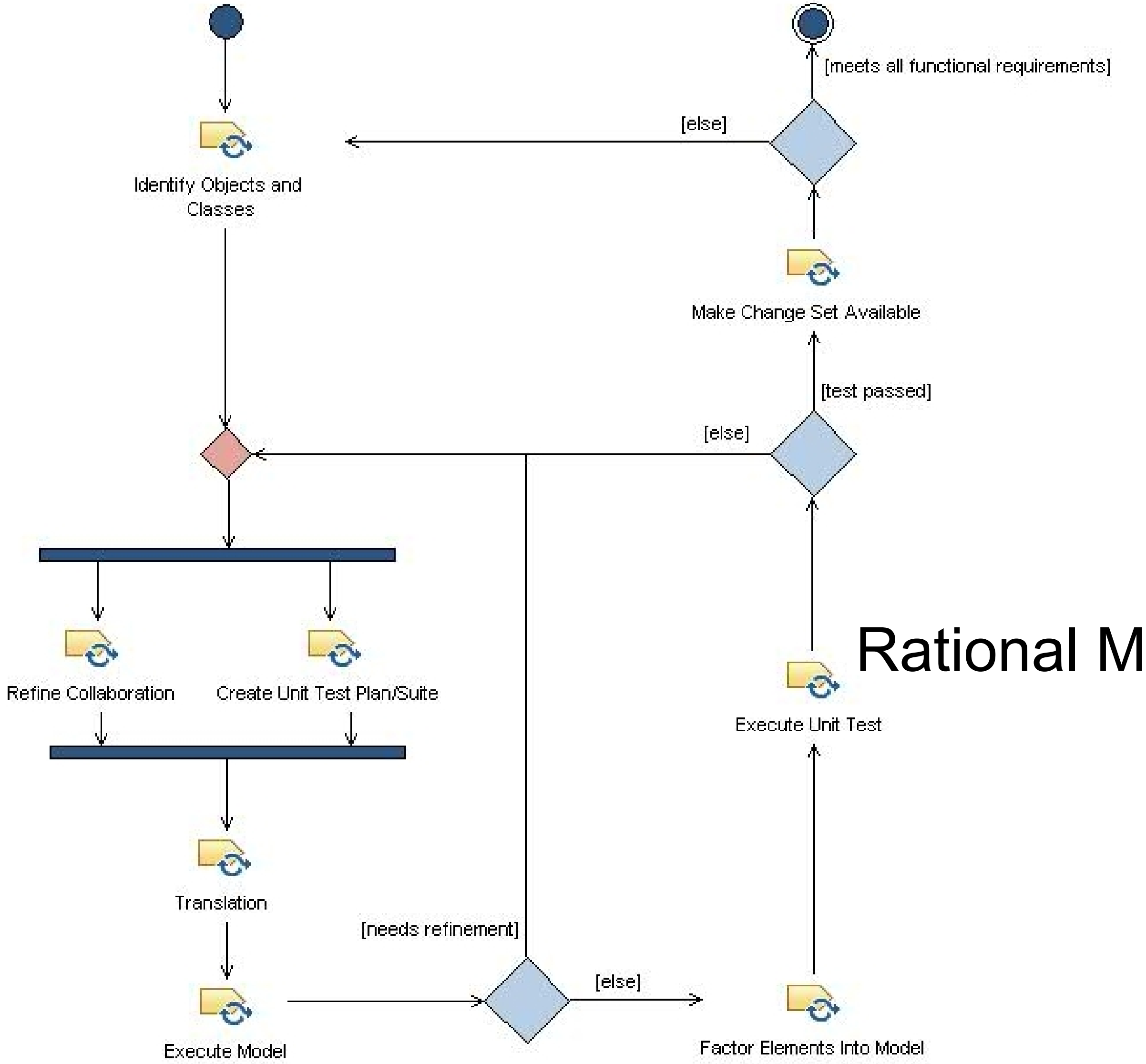
REPORT
Continuously improve by measuring and reporting progress in real time

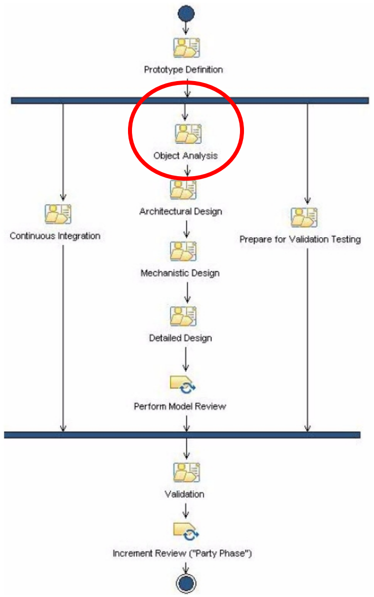
Repeatable activities producing a desired business outcome

Rational software Process and Collaboration support

- Rational Team Concert is the enabler for controlling process and managing change
    - Process template for ISO 26262 and DO-178B
        - Helps with project management
        - Team management
        - Task allocation
    - Integrates with practices that give guidance on the application of tools to support the standard
    - Configuration management and Collaboration platform
- Integrates with multiple Rational Tools
    - Rational Method Composer (RMC) for process guidance
    - Rational DOORS for requirements management
    - Rational Rhapsody for Model Based Systems Engineering
        - Removes system design errors early in the development process
        - Has a safety profile to aid in FMEA, FTA and hazard analysis
        - Developing an Automotive Safety profile specifically for ISO 26262
    - Rational Quality Manager (RQM) to plan tests
    - Rational Test Conductor to automate tests

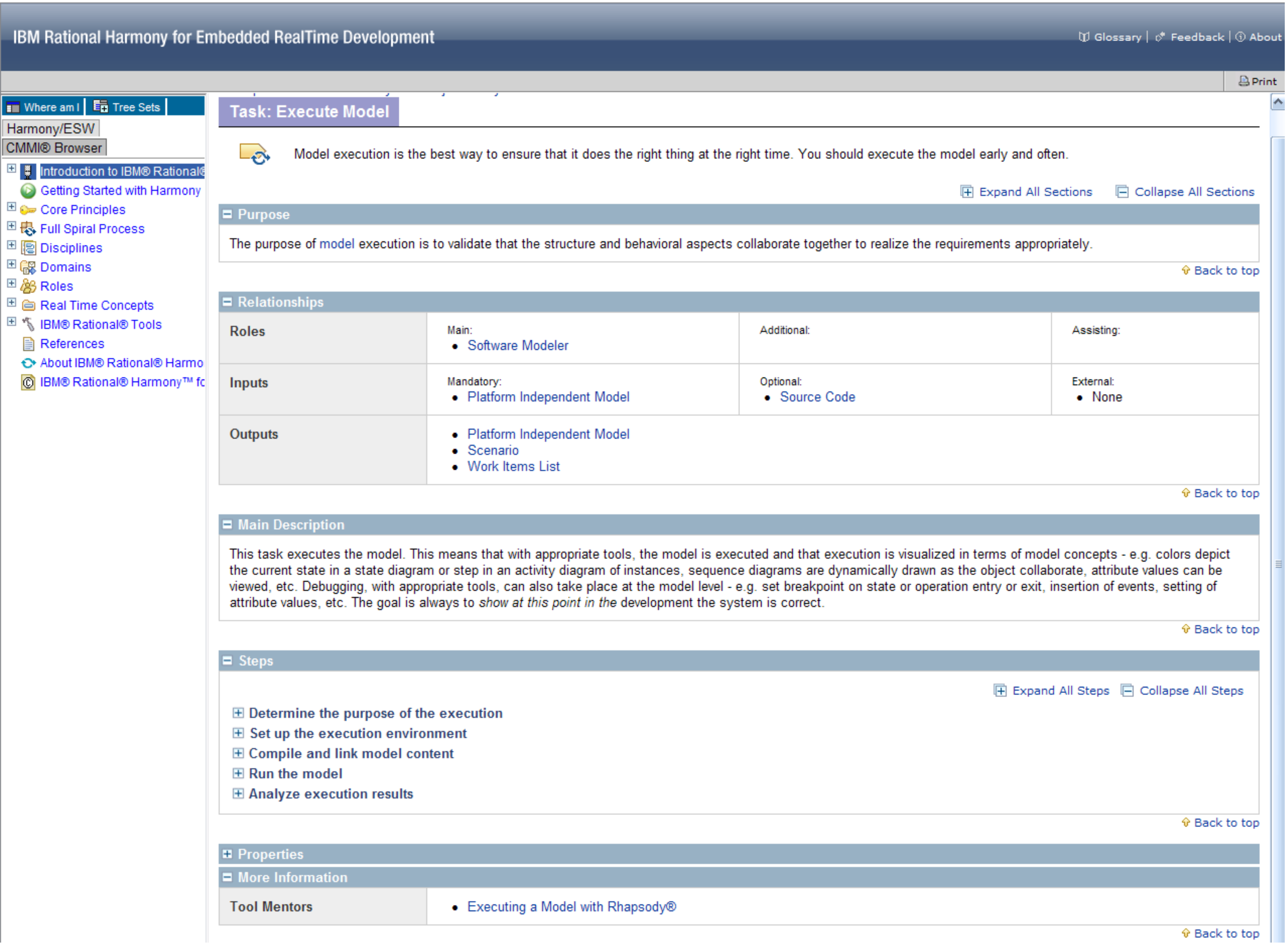
RMC: Capture Workflows (e.g. Harmony/ESW)


[meets all functional requirements]
[else]
Identify Objects and Classes
Make Change Set Available
[test passed]
[else]
Refine Collaboration
Create Unit Test Plan/Suite
Execute Unit Test
Translation
[needs refinement]
[else]
Execute Model
Factor Elements Into Model



Rational Method Composer

RMC: Workflows published as website



IBM Rational Harmony for Embedded RealTime Development

Task: Execute Model

Model execution is the best way to ensure that it does the right thing at the right time. You should execute the model early and often.

Purpose

The purpose of model execution is to validate that the structure and behavioral aspects collaborate together to realize the requirements appropriately.

Relationships

Roles	Main: • Software Modeler	Additional:	Assisting:
Inputs	Mandatory: • Platform Independent Model	Optional: • Source Code	External: • None
Outputs	• Platform Independent Model • Scenario • Work Items List		

Main Description

This task executes the model. This means that with appropriate tools, the model is executed and that execution is visualized in terms of model concepts - e.g. colors depict the current state in a state diagram or step in an activity diagram of instances, sequence diagrams are dynamically drawn as the object collaborate, attribute values can be viewed, etc. Debugging, with appropriate tools, can also take place at the model level - e.g. set breakpoint on state or operation entry or exit, insertion of events, setting of attribute values, etc. The goal is always to *show at this point in the* development the system is correct.

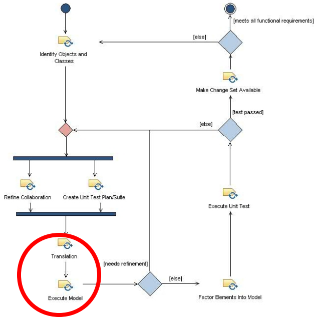
Steps

- **Determine the purpose of the execution**
- **Set up the execution environment**
- **Compile and link model content**
- **Run the model**
- **Analyze execution results**

Properties

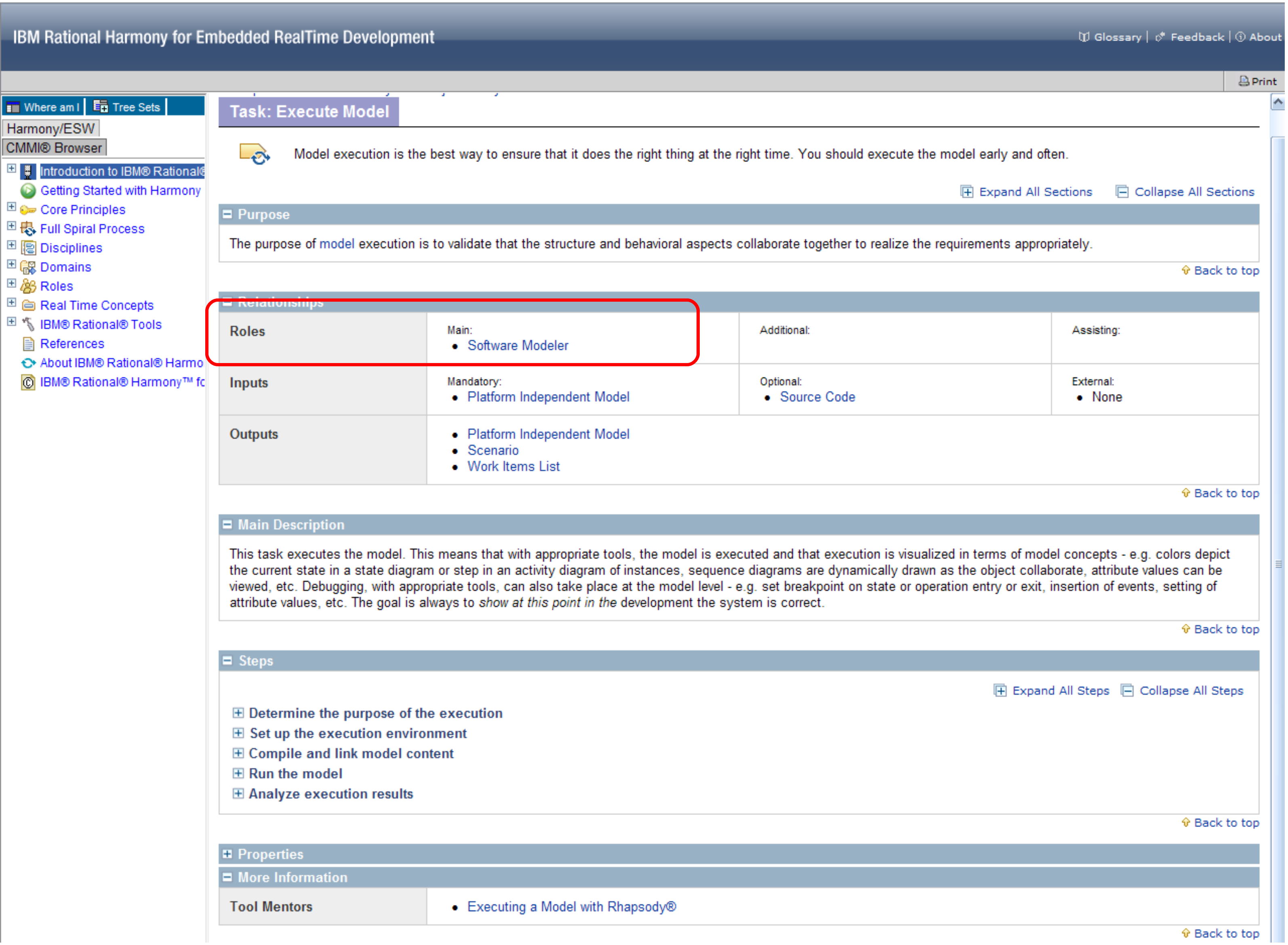
More Information

Tool Mentors	• Executing a Model with Rhapsody®



[meets all functional requirements]
[else]
Identify Objects and Classes
Make Change Set Available
[test passed]
[else]
Refine Collaboration
Create Unit Test Plan/Suite
Execute Unit Test
Translation
[needs refinement]
Execute Model
[else]
Factor Elements Into Model

RMC: RMC: Workflows published as website



IBM Rational Harmony for Embedded RealTime Development
Glossary | Feedback | About
Print
Where am I | Tree Sets
Harmony/ESW
CMMI® Browser
Introduction to IBM® Rational®
Getting Started with Harmony
Core Principles
Full Spiral Process
Disciplines
Domains
Roles
Real Time Concepts
IBM® Rational® Tools
References
About IBM® Rational® Harmo
IBM® Rational® Harmony™ fc

Task: Execute Model

Model execution is the best way to ensure that it does the right thing at the right time. You should execute the model early and often.

⊞ Expand All Sections ⊟ Collapse All Sections

⊟ Purpose

The purpose of model execution is to validate that the structure and behavioral aspects collaborate together to realize the requirements appropriately.

Back to top

⊟ Relationships

Roles	Main: • Software Modeler	Additional:	Assisting:
Inputs	Mandatory: • Platform Independent Model	Optional: • Source Code	External: • None
Outputs	• Platform Independent Model • Scenario • Work Items List		

Back to top

⊟ Main Description

This task executes the model. This means that with appropriate tools, the model is executed and that execution is visualized in terms of model concepts - e.g. colors depict the current state in a state diagram or step in an activity diagram of instances, sequence diagrams are dynamically drawn as the object collaborate, attribute values can be viewed, etc. Debugging, with appropriate tools, can also take place at the model level - e.g. set breakpoint on state or operation entry or exit, insertion of events, setting of attribute values, etc. The goal is always to *show at this point in the* development the system is correct.

Back to top

⊟ Steps

⊞ Expand All Steps ⊟ Collapse All Steps

- ⊞ **Determine the purpose of the execution**
- ⊞ **Set up the execution environment**
- ⊞ **Compile and link model content**
- ⊞ **Run the model**
- ⊞ **Analyze execution results**

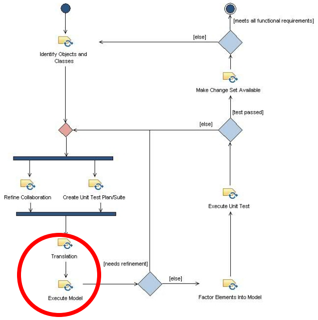
Back to top

⊞ Properties

⊟ More Information

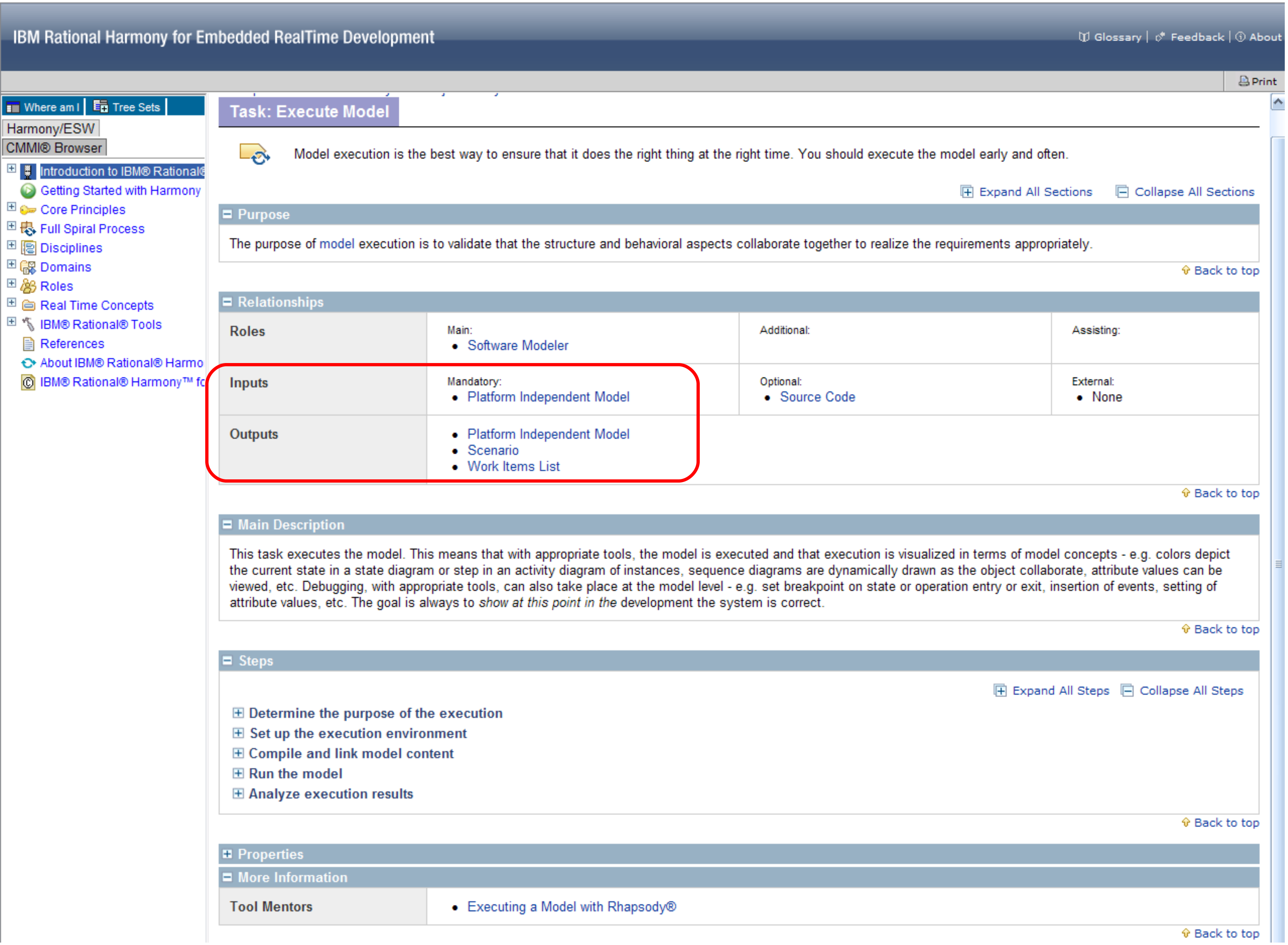
Tool Mentors	• Executing a Model with Rhapsody®

Back to top



[meets all functional requirements]
[else]
Identify Objects and Classes
Make Change Set Available
[test passed]
[else]
Refine Collaboration
Create Unit Test Plan/Suite
Execute Unit Test
Translation
[needs refinement]
[else]
Execute Model
Factor Elements Into Model

RMC: RMC: Workflows published as website



IBM Rational Harmony for Embedded RealTime Development
Glossary | Feedback | About
Print
Where am I
Tree Sets
Harmony/ESW
CMMI® Browser
Introduction to IBM® Rational®
Getting Started with Harmony
Core Principles
Full Spiral Process
Disciplines
Domains
Roles
Real Time Concepts
IBM® Rational® Tools
References
About IBM® Rational® Harmo
IBM® Rational® Harmony™ fc

Task: Execute Model

Model execution is the best way to ensure that it does the right thing at the right time. You should execute the model early and often.

Expand All Sections   Collapse All Sections

Purpose

The purpose of model execution is to validate that the structure and behavioral aspects collaborate together to realize the requirements appropriately.

Back to top

Relationships

Roles	Main: • Software Modeler	Additional:	Assisting:
Inputs	Mandatory: • Platform Independent Model	Optional: • Source Code	External: • None
Outputs	• Platform Independent Model • Scenario • Work Items List		

Back to top

Main Description

This task executes the model. This means that with appropriate tools, the model is executed and that execution is visualized in terms of model concepts - e.g. colors depict the current state in a state diagram or step in an activity diagram of instances, sequence diagrams are dynamically drawn as the object collaborate, attribute values can be viewed, etc. Debugging, with appropriate tools, can also take place at the model level - e.g. set breakpoint on state or operation entry or exit, insertion of events, setting of attribute values, etc. The goal is always to *show at this point in the* development the system is correct.

Back to top

Steps

Expand All Steps   Collapse All Steps

- **Determine the purpose of the execution**
- **Set up the execution environment**
- **Compile and link model content**
- **Run the model**
- **Analyze execution results**

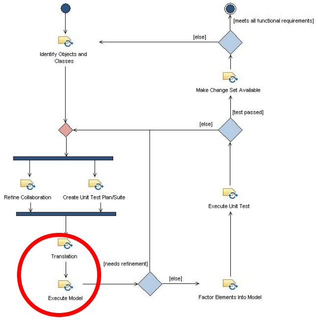
Back to top

Properties

More Information

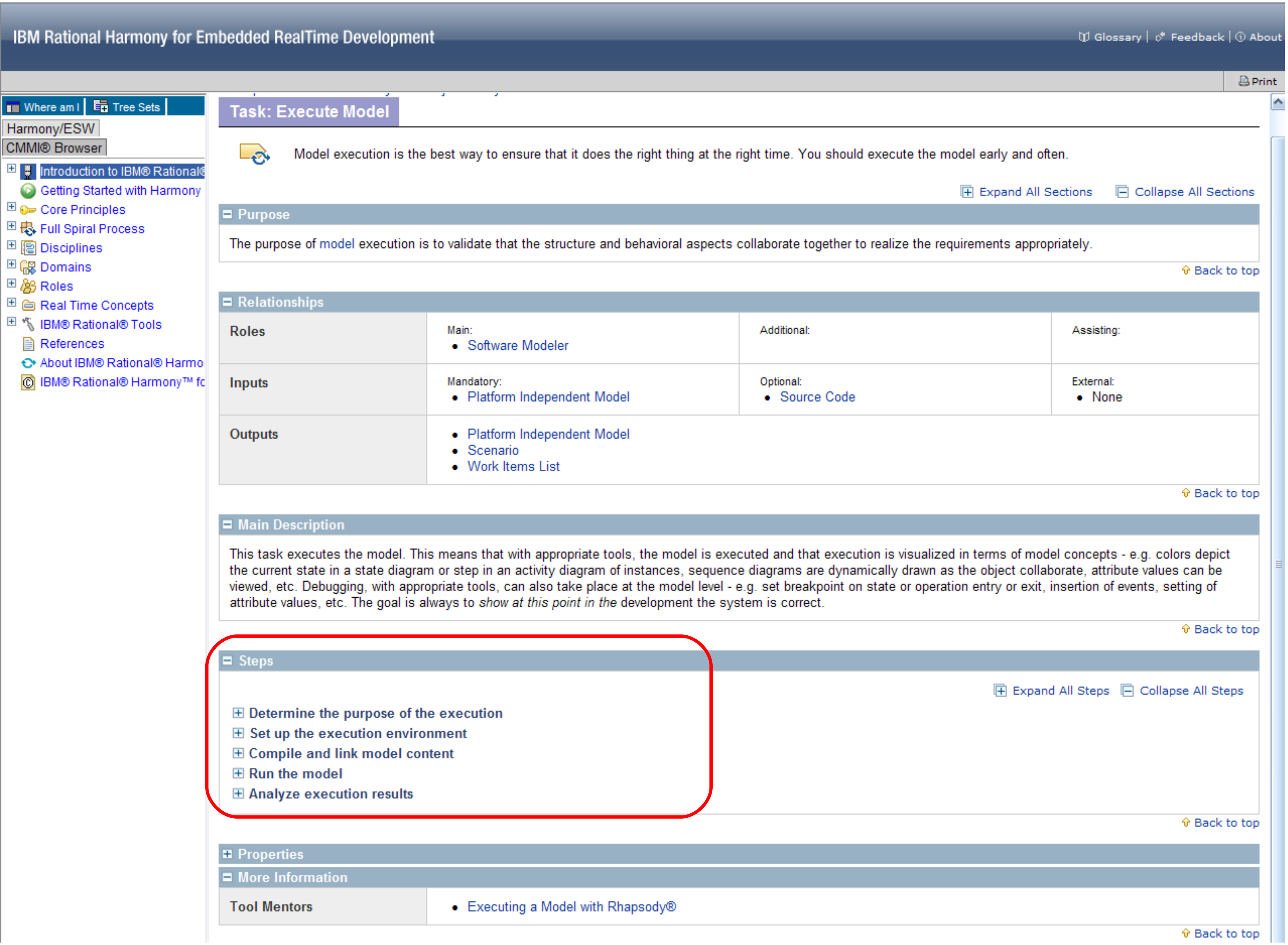
Tool Mentors	• Executing a Model with Rhapsody®

Back to top



[meets all functional requirements]
[else]
Identify Objects and Classes
Make Change Set Available
[test passed]
[else]
Refine Collaboration
Create Unit Test Plan/Suite
Execute Unit Test
Translation
[needs refinement]
[else]
Execute Model
Factor Elements Into Model

RMC: RMC: Workflows published as website


IBM Rational Harmony for Embedded RealTime Development
Glossary | Feedback | About
Print
Where am I | Tree Sets
Harmony/ESW
CMMI® Browser
Introduction to IBM® Rational®
Getting Started with Harmony
Core Principles
Full Spiral Process
Disciplines
Domains
Roles
Real Time Concepts
IBM® Rational® Tools
References
About IBM® Rational® Harmo
IBM® Rational® Harmony™ fc

Task: Execute Model

Model execution is the best way to ensure that it does the right thing at the right time. You should execute the model early and often.

Expand All Sections | Collapse All Sections

Purpose

The purpose of model execution is to validate that the structure and behavioral aspects collaborate together to realize the requirements appropriately.

Back to top

Relationships

Roles	Main: • Software Modeler	Additional:	Assisting:
Inputs	Mandatory: • Platform Independent Model	Optional: • Source Code	External: • None
Outputs	• Platform Independent Model • Scenario • Work Items List		

Back to top

Main Description

This task executes the model. This means that with appropriate tools, the model is executed and that execution is visualized in terms of model concepts - e.g. colors depict the current state in a state diagram or step in an activity diagram of instances, sequence diagrams are dynamically drawn as the object collaborate, attribute values can be viewed, etc. Debugging, with appropriate tools, can also take place at the model level - e.g. set breakpoint on state or operation entry or exit, insertion of events, setting of attribute values, etc. The goal is always to *show at this point in the* development the system is correct.

Back to top

Steps

Expand All Steps | Collapse All Steps

- Determine the purpose of the execution
- Set up the execution environment
- Compile and link model content
- Run the model
- Analyze execution results

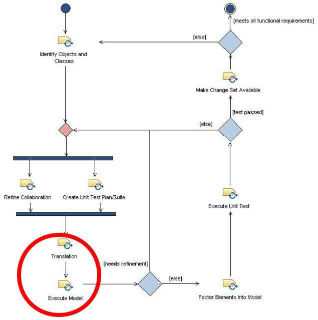
Back to top

Properties

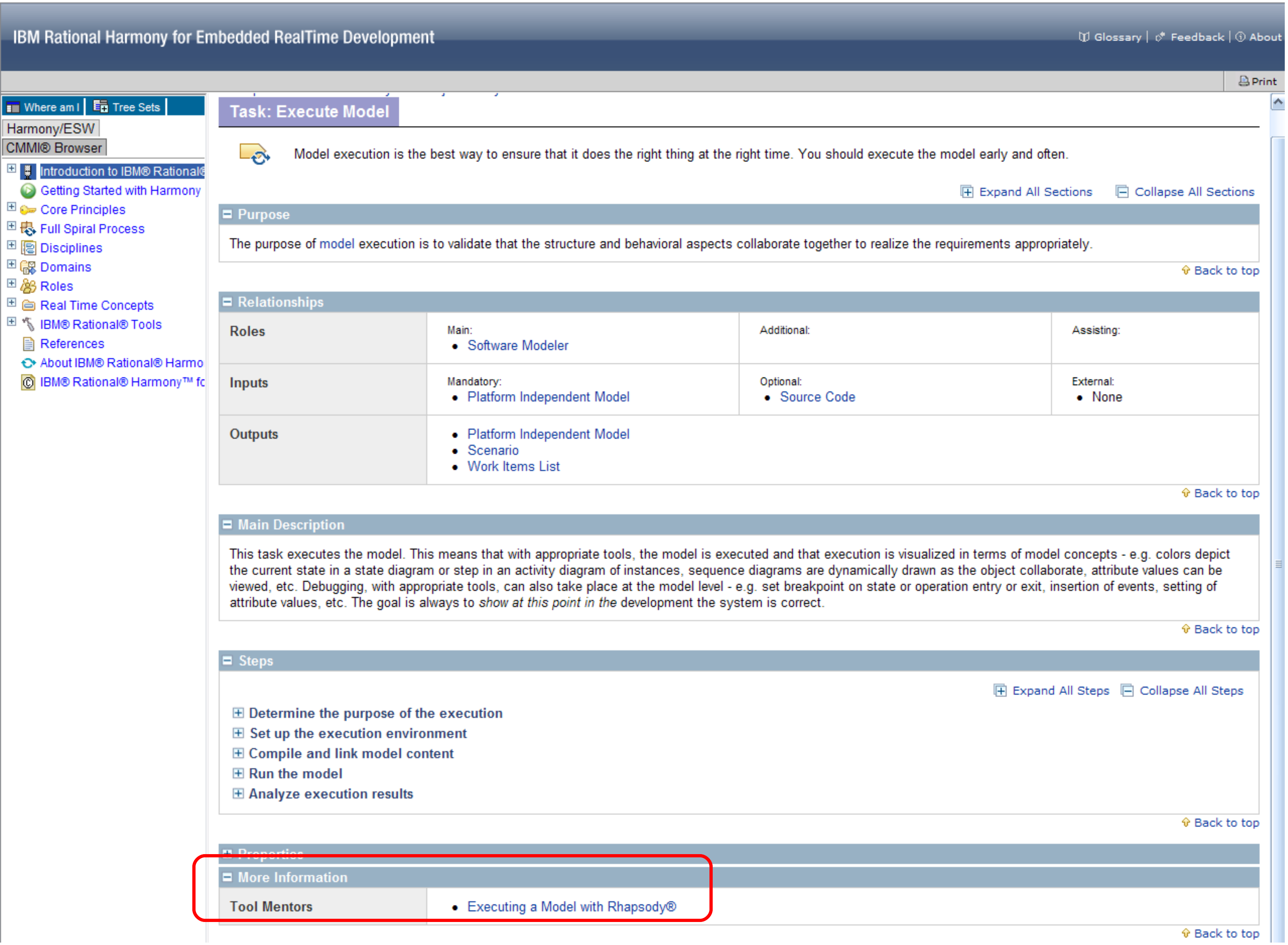
More Information

Tool Mentors	• Executing a Model with Rhapsody®

Back to top


[meets all functional requirements]
[else]
Identify Objects and Classes
Make Change Set Available
[test passed]
[else]
Refine Collaboration
Create Unit Test Plan/Suite
Execute Unit Test
Translation
[needs refinement]
[else]
Execute Model
Factor Elements Into Model

RMC: RMC: Workflows published as website



IBM Rational Harmony for Embedded RealTime Development
Glossary | Feedback | About
Print
Where am I
Tree Sets
Harmony/ESW
CMMI® Browser
Introduction to IBM® Rational®
Getting Started with Harmony
Core Principles
Full Spiral Process
Disciplines
Domains
Roles
Real Time Concepts
IBM® Rational® Tools
References
About IBM® Rational® Harmo
IBM® Rational® Harmony™ fo

Task: Execute Model

Model execution is the best way to ensure that it does the right thing at the right time. You should execute the model early and often.

Expand All Sections Collapse All Sections

Purpose

The purpose of model execution is to validate that the structure and behavioral aspects collaborate together to realize the requirements appropriately.

Back to top

Relationships

Roles	Main: - Software Modeler	Additional:	Assisting:
Inputs	Mandatory: - Platform Independent Model	Optional: - Source Code	External: - None
Outputs	- Platform Independent Model - Scenario - Work Items List		

Back to top

Main Description

This task executes the model. This means that with appropriate tools, the model is executed and that execution is visualized in terms of model concepts - e.g. colors depict the current state in a state diagram or step in an activity diagram of instances, sequence diagrams are dynamically drawn as the object collaborate, attribute values can be viewed, etc. Debugging, with appropriate tools, can also take place at the model level - e.g. set breakpoint on state or operation entry or exit, insertion of events, setting of attribute values, etc. The goal is always to *show at this point in the* development the system is correct.

Back to top

Steps

Expand All Steps Collapse All Steps

- Determine the purpose of the execution
- Set up the execution environment
- Compile and link model content
- Run the model
- Analyze execution results

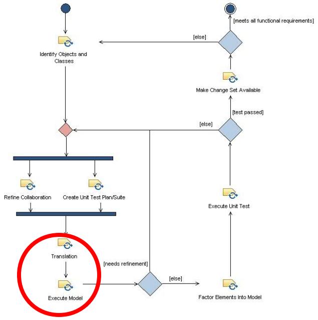
Back to top

Properties

More Information

Tool Mentors	- Executing a Model with Rhapsody®

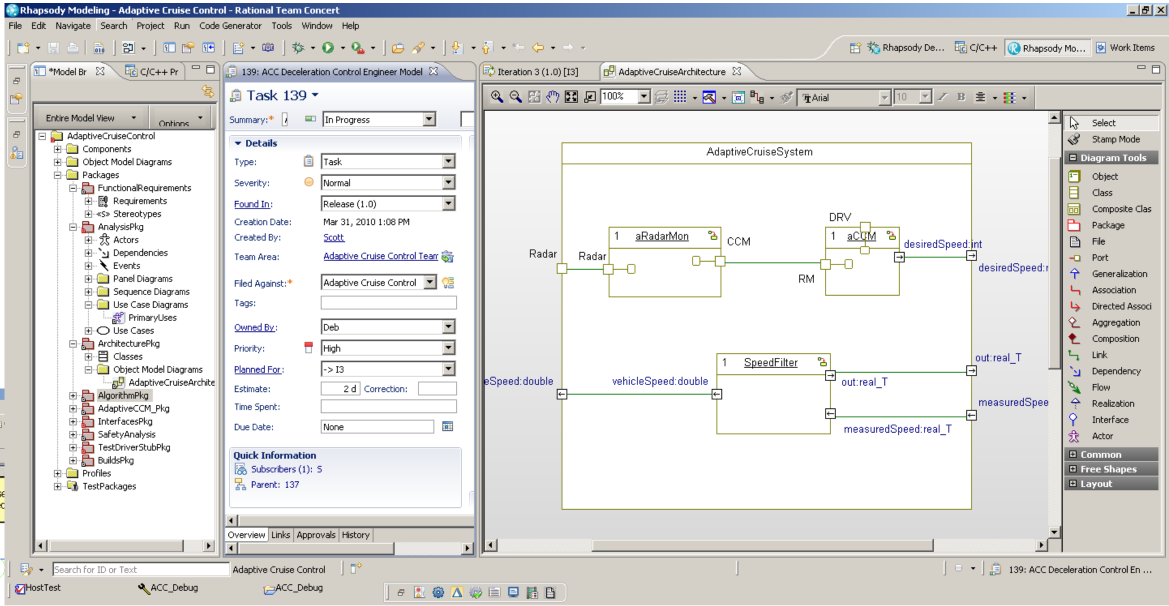
Back to top



[meets all functional requirements]
[else]
Identify Objects and Classes
Make Change Set Available
[test passed]
[else]
Refine Collaboration
Create Unit Test Plan/Suite
Execute Unit Test
Translation
[needs refinement]
Execute Model
[else]
Factor Elements Into Model

RTC Tool Integrations

Rational Team Concert


Rhapsody Modeling - Adaptive Cruise Control - Rational Team Concert

139: ACC Deceleration Control Engineer Model

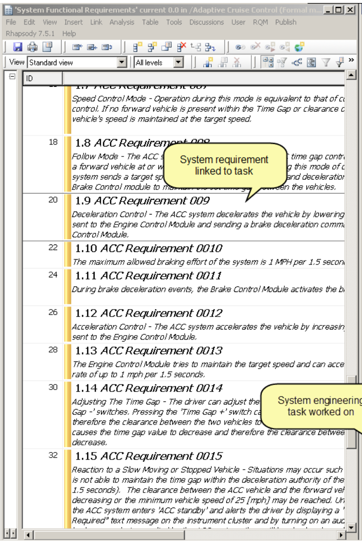
Task 139

Summary: In Progress

Details

Type:	Task
Severity:	Normal
Found In:	Release (1.0)
Creation Date:	Mar 31, 2010 1:08 PM
Created By:	Scott
Team Area:	Adaptive Cruise Control Team
Filed Against:	Adaptive Cruise Control
Tags:	
Owned By:	Deb
Priority:	High
Planned For:	-> I3
Estimate:	2 d Correction:
Time Spent:	
Due Date:	None

Quick Information
Subscribers (1): S
Parent: 137


'System Functional Requirements' current 0.0 in /Adaptive Cruise Control (Formal m...

File Edit View Insert Link Analysis Table Tools Discussions User RQM Publish
Rhapsody 7.5.1 Help
View Standard view
All levels
ID

Speed Control Mode - Operation during this mode is equivalent to that of c control. If no forward vehicle is present within the Time Gap or clearance c vehicle's speed is maintained at the target speed.

18 1.8 ACC Requirement 008
Follow Mode - The ACC s … time gap contr a forward vehicle at or w … g this mode of c system sends a target sp … and deceleratio Brake Control module to … en the vehicles.

20 1.9 ACC Requirement 009
Deceleration Control - The ACC system decelerates the vehicle by lowering sent to the Engine Control Module and sending a brake deceleration comm Control Module.

22 1.10 ACC Requirement 0010
The maximum allowed braking effort of the system is 1 MPH per 1.5 secon

24 1.11 ACC Requirement 0011
During brake deceleration events, the Brake Control Module activates the b

26 1.12 ACC Requirement 0012
Acceleration Control - The ACC system accelerates the vehicle by increasin sent to the Engine Control Module.

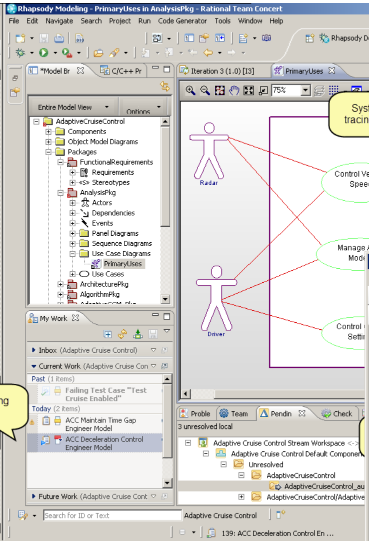
28 1.13 ACC Requirement 0013
The Engine Control Module tries to maintain the target speed and can acce rate of up to 1 mph per 1.5 seconds.

30 1.14 ACC Requirement 0014
Adjusting The Time Gap - The driver can adjust the … Gap -' switches. Pressing the 'Time Gap +' switch ca … therefore the clearance between the two vehicles to … causes the time gap value to decrease and therefore the clearance betwee decrease.

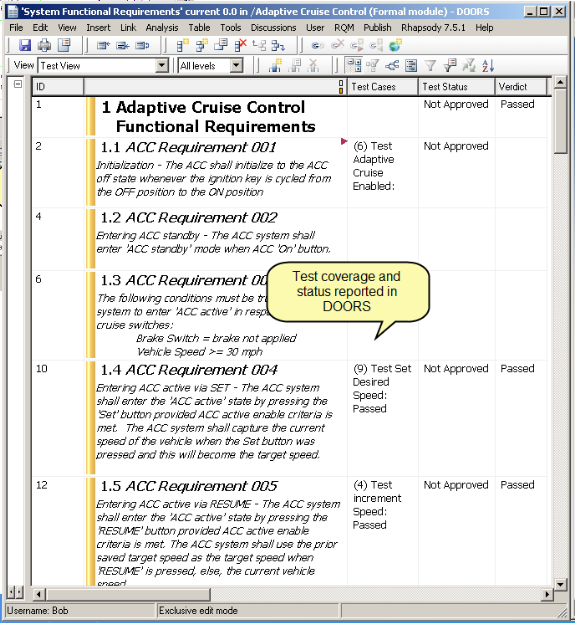
32 1.15 ACC Requirement 0015
Reaction to a Slow Moving or Stopped Vehicle - Situations may occur such is not able to maintain the time gap within the deceleration authority of the 1.5 seconds). The clearance between the ACC vehicle and the forward vel decreasing or the minimum vehicle speed of 25 [mph] may be reached. Ur the ACC system enters 'ACC standby' and alerts the driver by displaying a ' Required" text message on the instrument cluster and by turning on an auc

System requirement linked to task

System engineering task worked on


Rhapsody Modeling - PrimaryUses in AnalysisPkg - Rational Team Concert

System use tracing to rec

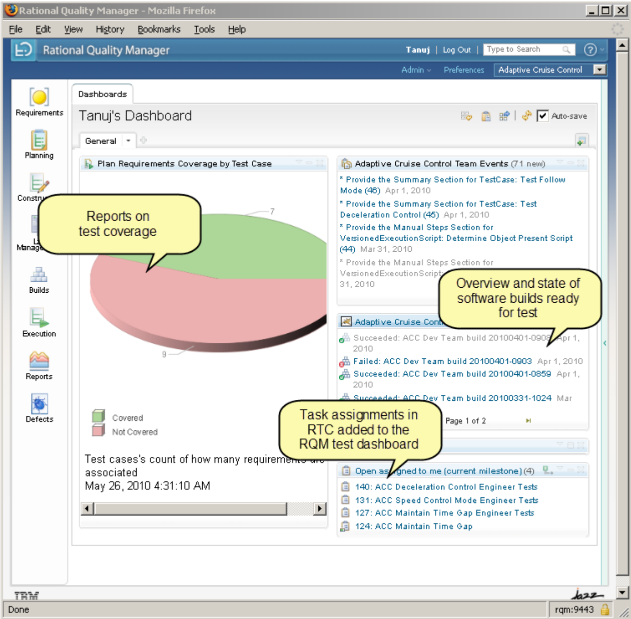

'System Functional Requirements' current 0.0 in /Adaptive Cruise Control (Formal module) - DOORS

File Edit View Insert Link Analysis Table Tools Discussions User RQM Publish Rhapsody 7.5.1 Help
View Test View
All levels

ID		Test Cases	Test Status	Verdict
1	1 Adaptive Cruise Control Functional Requirements		Not Approved	Passed
2	1.1 ACC Requirement 001 Initialization - The ACC shall initialize to the ACC off state whenever the ignition key is cycled from the OFF position to the ON position	(6) Test Adaptive Cruise Enabled:	Not Approved	
4	1.2 ACC Requirement 002 Entering ACC standby - The ACC system shall enter 'ACC standby' mode when ACC 'On' button.			
6	1.3 ACC Requirement 00 The following conditions must be tr system to enter 'ACC active' in resp cruise switches: Brake Switch = brake not applied Vehicle Speed >= 30 mph			
10	1.4 ACC Requirement 004 Entering ACC active via SET - The ACC system shall enter the 'ACC active' state by pressing the 'Set' button provided ACC active enable criteria is met. The ACC system shall capture the current speed of the vehicle when the Set button was pressed and this will become the target speed.	(9) Test Set Desired Speed: Passed	Not Approved	Passed
12	1.5 ACC Requirement 005 Entering ACC active via RESUME - The ACC system shall enter the 'ACC active' state by pressing the 'RESUME' button provided ACC active enable criteria is met. The ACC system shall use the prior saved target speed as the target speed when 'RESUME' is pressed, else, the current vehicle speed	(4) Test increment Speed: Passed	Not Approved	Passed

Test coverage and status reported in DOORS

Username: Bob
Exclusive edit mode


Rational Quality Manager - Mozilla Firefox

Tanuj's Dashboard

Plan Requirements Coverage by Test Case

Test cases's count of how many requirements are associated
May 26, 2010 4:31:10 AM

Adaptive Cruise Control Team Events (71 new)

- Provide the Summary Section for TestCase: Test Follow Mode (46) Apr 1, 2010
- Provide the Summary Section for TestCase: Test Deceleration Control (45) Apr 1, 2010
- Provide the Manual Steps Section for VersionedExecutionScript: Determine Object Present Script (44) Mar 31, 2010
- Provide the Manual Steps Section for VersionedExecutionScript … 31, 2010

Adaptive Cruise Con…

- Succeeded: ACC Dev Team build 20100401-0908 Apr 1, 2010
- Failed: ACC Dev Team build 20100401-0903 Apr 1, 2010
- Succeeded: ACC Dev Team build 20100401-0859 Apr 1, 2010
- Succeeded: ACC Dev Team build 20100331-1024 Mar

Page 1 of 2

Open assigned to me (current milestone) (4)

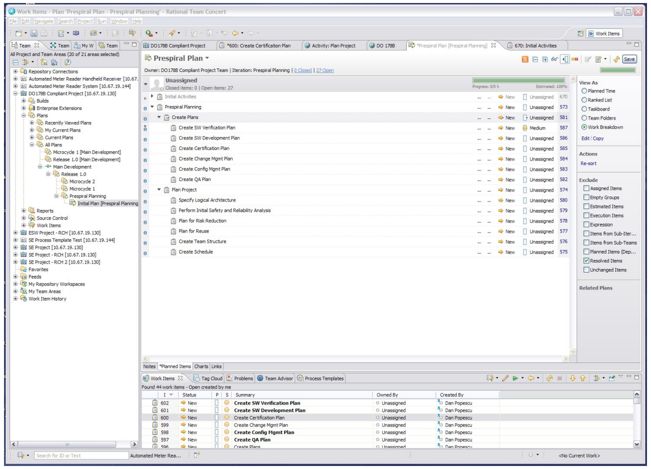
- 140: ACC Deceleration Control Engineer Tests
- 131: ACC Speed Control Mode Engineer Tests
- 127: ACC Maintain Time Gap Engineer Tests
- 124: ACC Maintain Time Gap

Reports on test coverage

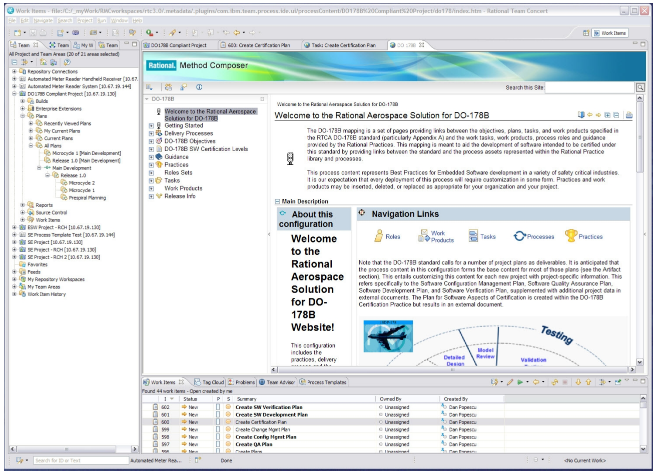
Overview and state of software builds ready for test

Task assignments in RTC added to the RQM test dashboard

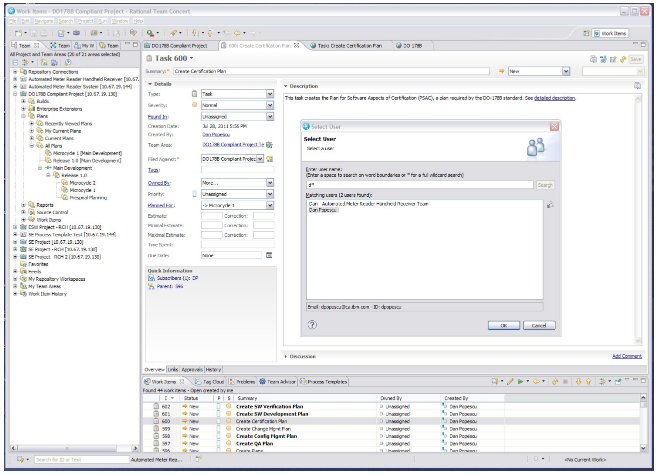
RTC: Process definition & enactment integration



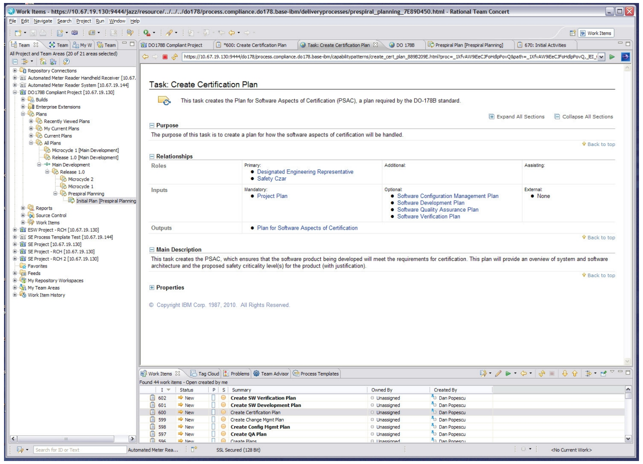
Select task templates



View process guidance

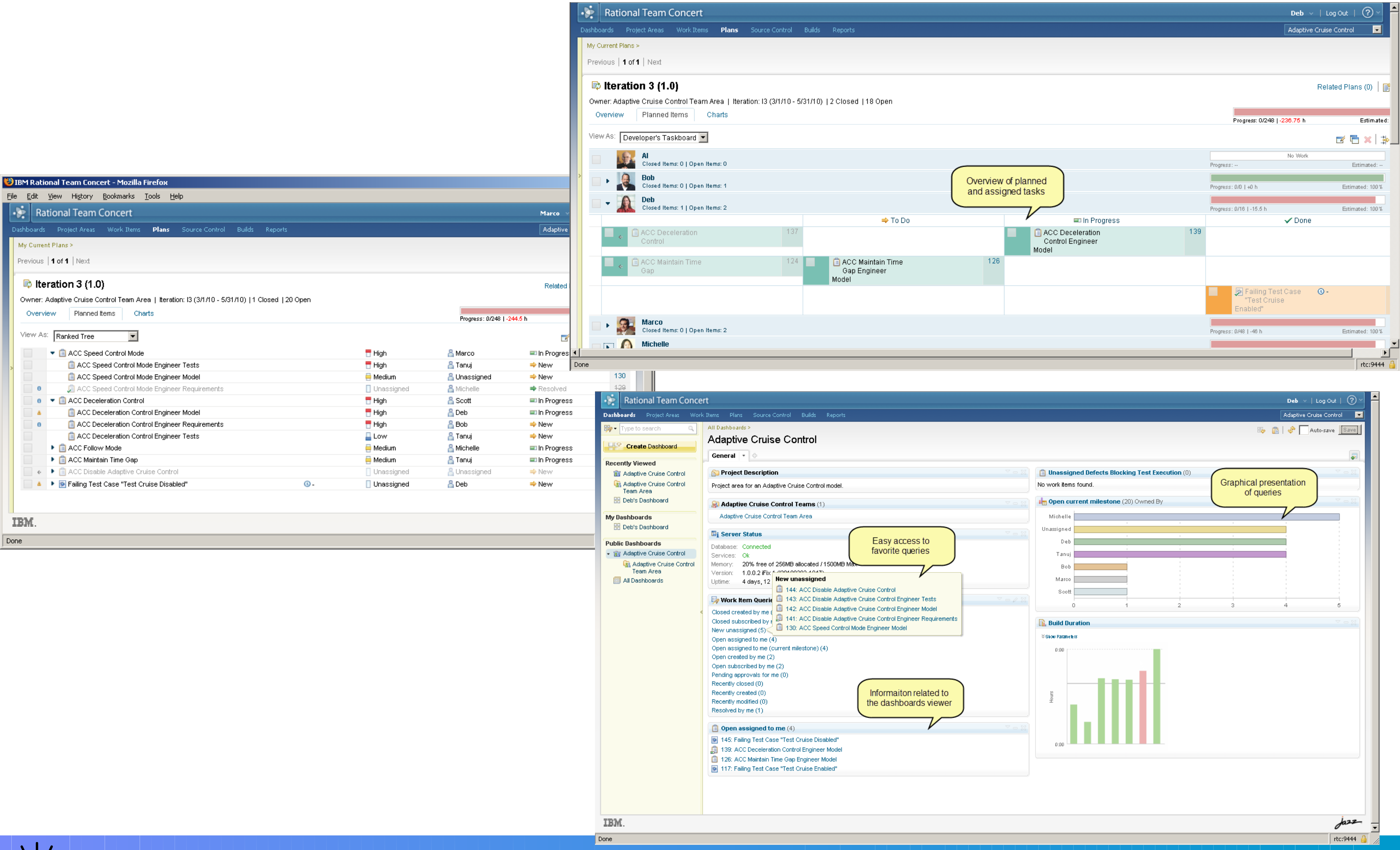


Assign tasks to workers



View Task-specific guidance

RTC: Tracking via stakeholder-specific dashboards



Overview of planned and assigned tasks

Easy access to favorite queries

Graphical presentation of queries

Informaiton related to the dashboards viewer

IBM Rational Team Concert - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Rational Team Concert
Marco
Dashboards Project Areas Work Items Plans Source Control Builds Reports
Adaptive
My Current Plans >
Previous | 1 of 1 | Next
Related
Overview Planned Items Charts
IBM.
Done

Iteration 3 (1.0)

Owner: Adaptive Cruise Control Team Area | Iteration: I3 (3/1/10 - 5/31/10) | 1 Closed | 20 Open

Progress: 0/248 | -244.5 h

View As: Ranked Tree

ACC Speed Control Mode	High	Marco	In Progres
ACC Speed Control Mode Engineer Tests	High	Tanuj	New
ACC Speed Control Mode Engineer Model	Medium	Unassigned	New
ACC Speed Control Mode Engineer Requirements	Unassigned	Michelle	Resolved
ACC Deceleration Control	High	Scott	In Progress
ACC Deceleration Control Engineer Model	High	Deb	In Progress
ACC Deceleration Control Engineer Requirements	High	Bob	New
ACC Deceleration Control Engineer Tests	Low	Tanuj	New
ACC Follow Mode	Medium	Michelle	In Progress
ACC Maintain Time Gap	Medium	Tanuj	In Progress
ACC Disable Adaptive Cruise Control	Unassigned	Unassigned	New
Failing Test Case "Test Cruise Disabled"	Unassigned	Deb	New

Iteration 3 (1.0)

Owner: Adaptive Cruise Control Team Area | Iteration: I3 (3/1/10 - 5/31/10) | 2 Closed | 18 Open

Progress: 0/248 | -236.75 h Estimated:

View As: Developer's Taskboard

Al
Closed Items: 0 | Open Items: 0
No Work
Progress: -- Estimated: --

Bob
Closed Items: 0 | Open Items: 1
Progress: 0/0 | +0 h Estimated: 100 %

Deb
Closed Items: 1 | Open Items: 2
Progress: 0/16 | -15.5 h Estimated: 100 %

	To Do	In Progress	Done
ACC Deceleration Control 137		ACC Deceleration Control Engineer Model 139	
ACC Maintain Time Gap 124	ACC Maintain Time Gap Engineer Model 126		
			Failing Test Case "Test Cruise Enabled"

Marco
Closed Items: 0 | Open Items: 2
Progress: 0/48 | -48 h Estimated: 100 %

Michelle

Recently Viewed
- Adaptive Cruise Control
- Adaptive Cruise Control Team Area
- Deb's Dashboard

My Dashboards
- Deb's Dashboard

Public Dashboards
- Adaptive Cruise Control
  - Adaptive Cruise Control Team Area
- All Dashboards

Adaptive Cruise Control

Project Description

Project area for an Adaptive Cruise Control model.

Adaptive Cruise Control Teams (1)

Adaptive Cruise Control Team Area

Server Status

Database: Connected
Services: Ok
Memory: 20% free of 256MB allocated / 1500MB Max
Version: 1.0.0.2 iFix
Uptime: 4 days, 12

Work Item Queri

- Closed created by me
- Closed subscribed by
- New unassigned (5)
- Open assigned to me (4)
- Open assigned to me (current milestone) (4)
- Open created by me (2)
- Open subscribed by me (2)
- Pending approvals for me (0)
- Recently closed (0)
- Recently created (0)
- Recently modified (0)
- Resolved by me (1)

New unassigned

- 144: ACC Disable Adaptive Cruise Control
- 143: ACC Disable Adaptive Cruise Control Engineer Tests
- 142: ACC Disable Adaptive Cruise Control Engineer Model
- 141: ACC Disable Adaptive Cruise Control Engineer Requirements
- 130: ACC Speed Control Mode Engineer Model

Open assigned to me (4)

- 145: Failing Test Case "Test Cruise Disabled"
- 139: ACC Deceleration Control Engineer Model
- 126: ACC Maintain Time Gap Engineer Model
- 117: Failing Test Case "Test Cruise Enabled"

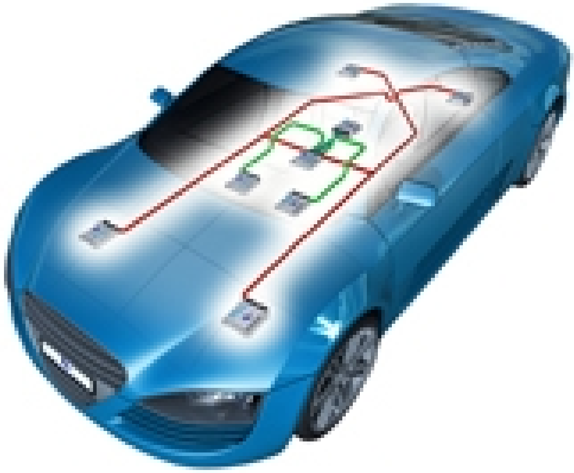
Unassigned Defects Blocking Test Execution (0)

No work items found.

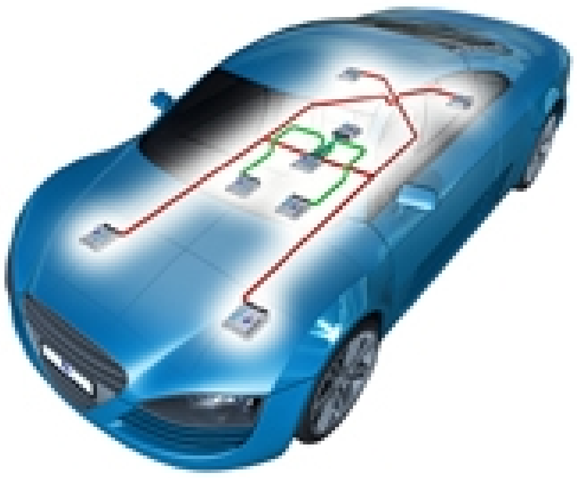
Open current milestone (20) Owned By

Build Duration

ISO-26262

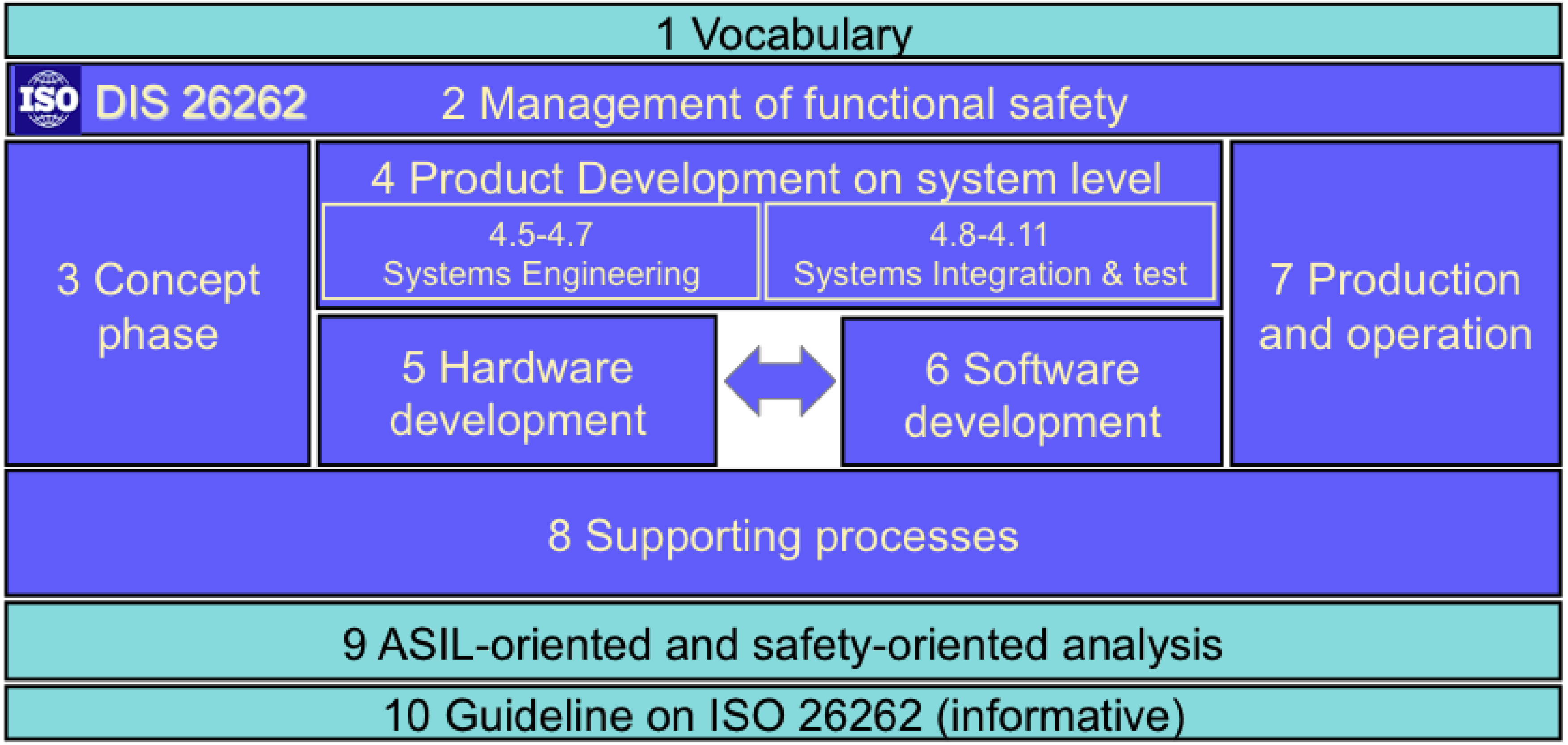


What is 26262?



- Automotive Safety Standard Under Development
  - Technical name is ISO 26262
    - ISO is International Organization for Standardization
- Which parts of vehicle does 26262 affect?
  - Electrical/Electronic (E/E) “that provides safety or safety-related functions”
  - Obvious examples:
    - anti-lock brakes, air bags, traction control, electronic cruise control, adaptive cruise control, collision avoidance, lane change control
  - Less obvious examples:
    - front windshield defroster/defogger, rear windshield (backlite) defroster, auto-on headlamps, auto-on running lights, seat-belt pre-tensioners, low tire pressure warning system, engine, electric-assist power steering.

A look inside ISO 26262 "Road vehicles -- Functional safety


1 Vocabulary
ISO
DIS 26262
2 Management of functional safety
3 Concept phase
4 Product Development on system level
4.5-4.7
Systems Engineering
4.8-4.11
Systems Integration & test
5 Hardware development
6 Software development
7 Production and operation
8 Supporting processes
9 ASIL-oriented and safety-oriented analysis
10 Guideline on ISO 26262 (informative)

Defines an automotive safety lifecycle including

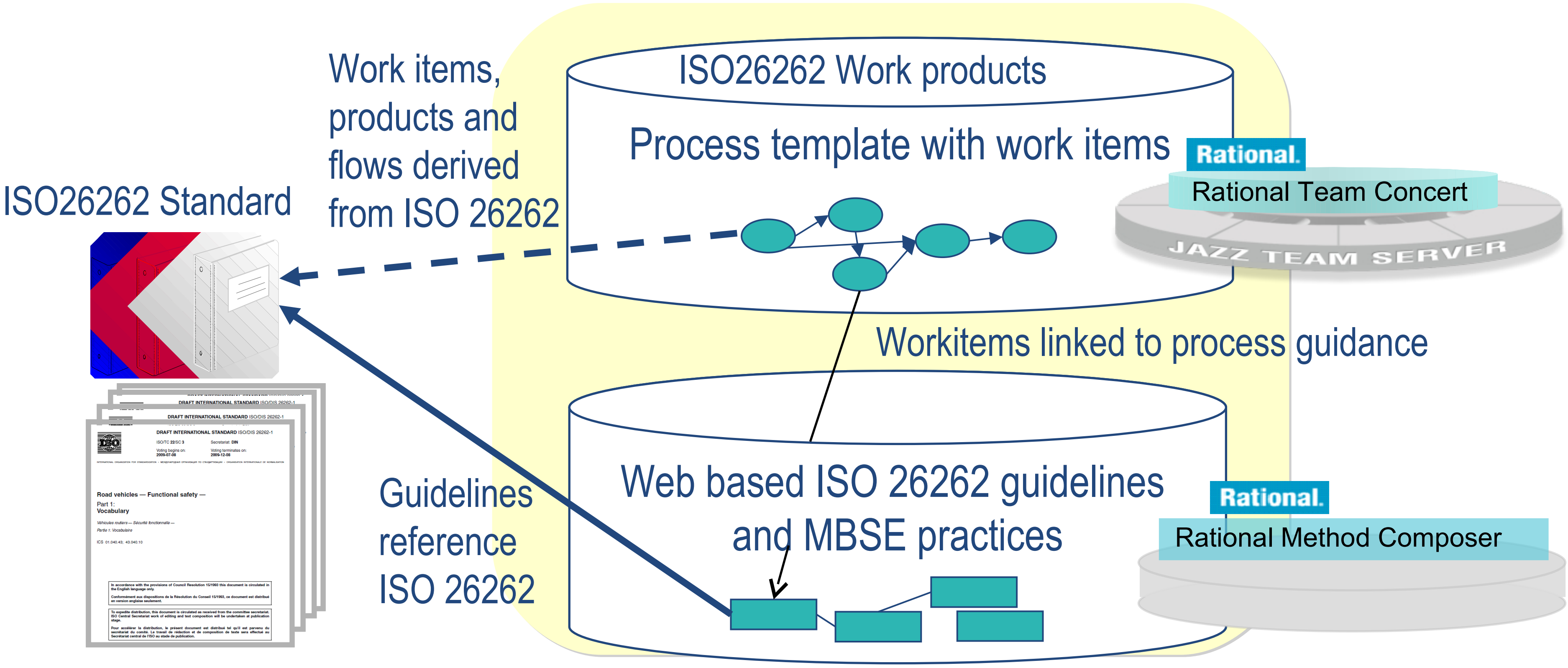
- a risk-based approach for determining risk classes (Automotive Safety Integrity Levels, ASILs).
- definition of optional, recommended and highly recommended methods for development activities within system-, hardware and software development depending on defined ASIL

Drivers for ISO 26262

- **German Legislature requires, that safe cars are developed according to state-of the-art technology**
- **You need a defensible process for creating safe software**
  - Consider adopting documented best practices instead of inventing your own
  - If everyone else adopts MISRA, IEC 61508 or ISO 26262 and you don’t, you might be considered negligent (failure to follow “standard practices”)
- ISO 26262 currently draft standard (DIS)
  - Published June 2009
- Currently delivery rumours sometime between September and December 2011

ISO 26262 RTC and RMC

- 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



ISO26262 Standard

Work items, products and flows derived from ISO 26262

ISO26262 Work products

Process template with work items

Rational.

Rational Team Concert

Workitems linked to process guidance

Web based ISO 26262 guidelines and MBSE practices

Rational.

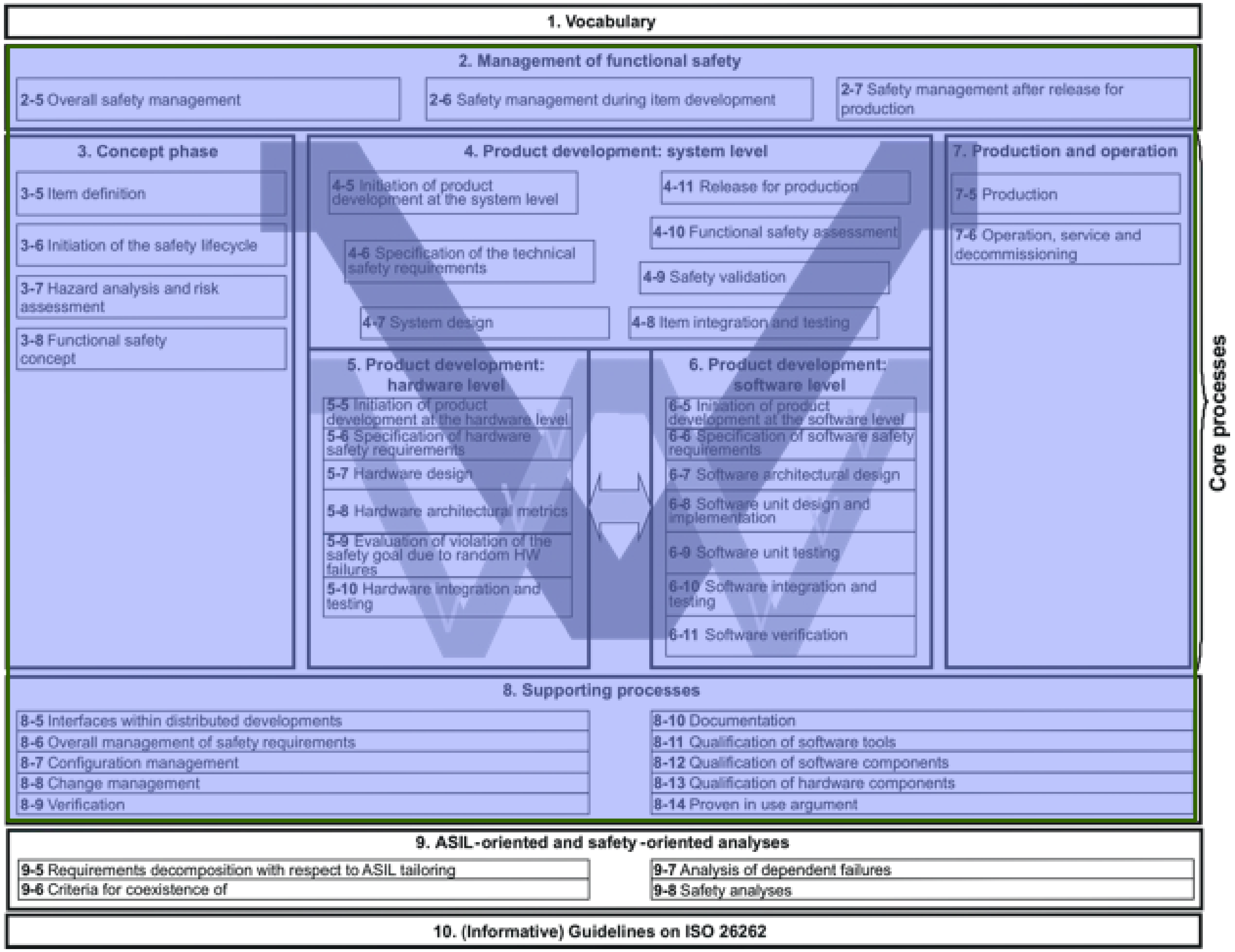
Rational Method Composer

Guidelines reference ISO 26262

JAZZ TEAM SERVER
DRAFT INTERNATIONAL STANDARD ISO/DIS 26262-1
Road vehicles — Functional safety —
Part 1:
Vocabulary

RTC ISO 26262 Process and Practice templates

- Scope of Process template and guidance covers 95%, phases 2-8*



1. Vocabulary

2. Management of functional safety

2-5 Overall safety management

2-6 Safety management during item development

2-7 Safety management after release for production

3. Concept phase

3-5 Item definition

3-6 Initiation of the safety lifecycle

3-7 Hazard analysis and risk assessment

3-8 Functional safety concept

4. Product development: system level

4-5 Initiation of product development at the system level

4-6 Specification of the technical safety requirements

4-7 System design

4-8 Item integration and testing

4-9 Safety validation

4-10 Functional safety assessment

4-11 Release for production

5. Product development: hardware level

5-5 Initiation of product development at the hardware level

5-6 Specification of hardware safety requirements

5-7 Hardware design

5-8 Hardware architectural metrics

5-9 Evaluation of violation of the safety goal due to random HW failures

5-10 Hardware integration and testing

6. Product development: software level

6-5 Initiation of 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 testing

6-10 Software integration and testing

6-11 Software verification

7. Production and operation

7-5 Production

7-6 Operation, service and decommissioning

Core processes

8. Supporting processes

8-5 Interfaces within distributed developments

8-6 Overall management of safety requirements

8-7 Configuration management

8-8 Change management

8-9 Verification

8-10 Documentation

8-11 Qualification of software tools

8-12 Qualification of software components

8-13 Qualification of hardware components

8-14 Proven in use argument

9. ASIL-oriented and safety-oriented analyses

9-5 Requirements decomposition with respect to ASIL tailoring

9-6 Criteria for coexistence of

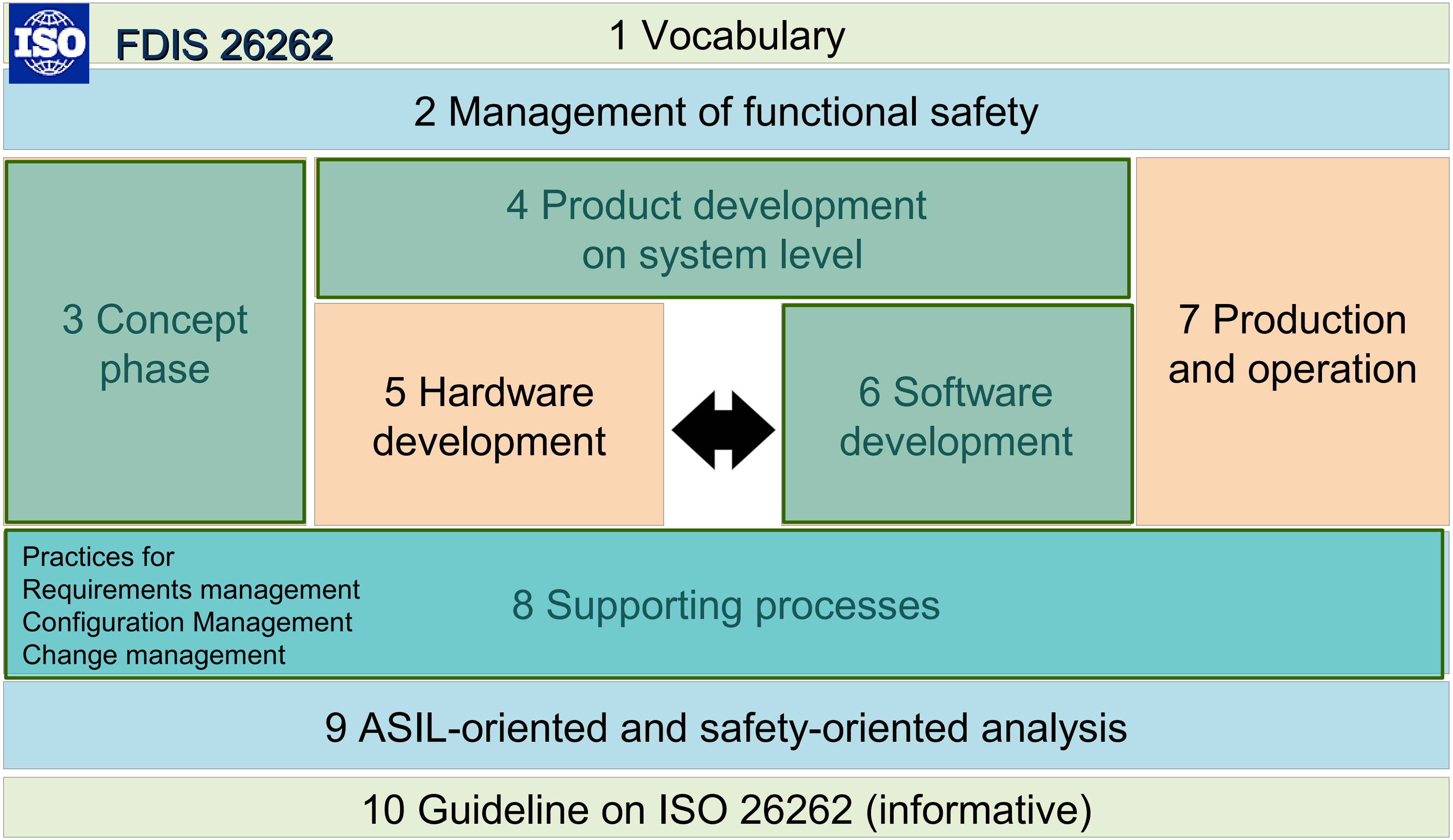
9-7 Analysis of dependent failures

9-8 Safety analyses

10. (Informative) Guidelines on ISO 26262

Available practices for ISO 26262

- Mainly in the areas of supporting practices and around MBSE, SW and test
- Work going on with Embedded HW and SW integration



FDIS 26262

1 Vocabulary

2 Management of functional safety

3 Concept phase

4 Product development on system level

5 Hardware development

6 Software development

7 Production and operation

Practices for
Requirements management
Configuration Management
Change management

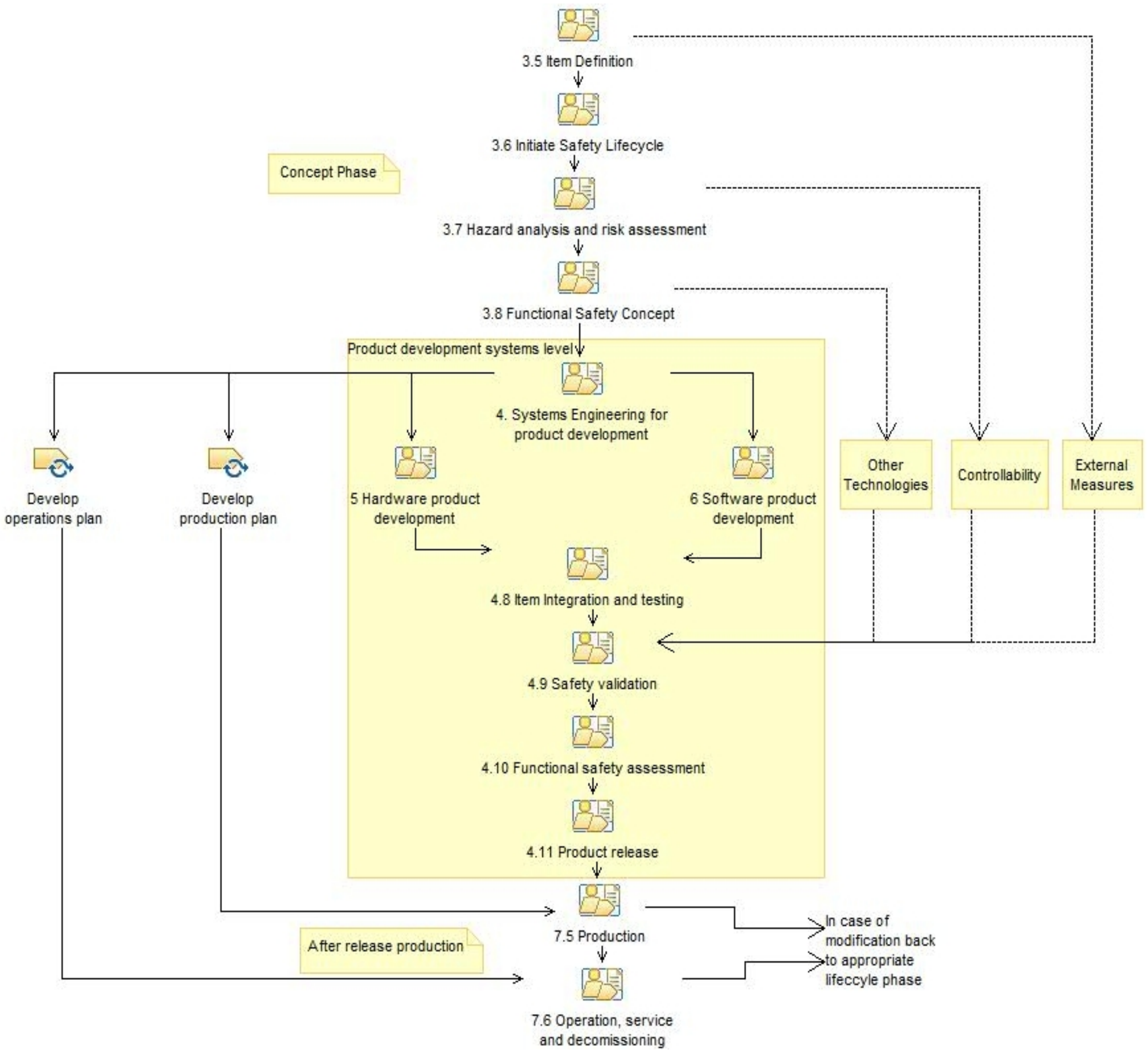
8 Supporting processes

9 ASIL-oriented and safety-oriented analysis

10 Guideline on ISO 26262 (informative)

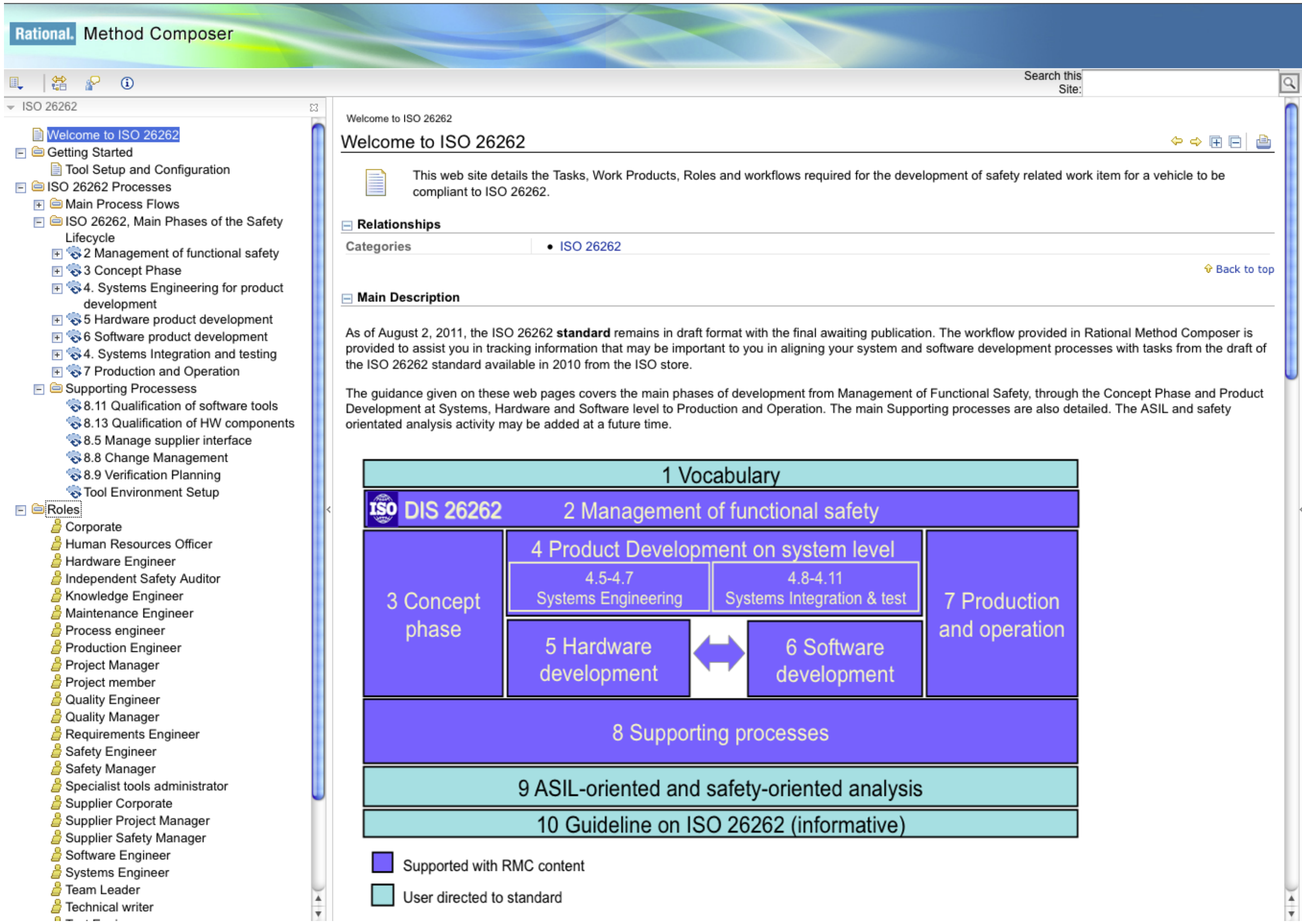
ISO 26262 in Rational Method Composer

- RMC captures activities and flows
- Flows are generic and reflect ISO 26262
  - Can be customised to fit your process
- 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 workproducts
  - Applicable roles


3.5 Item Definition
3.6 Initiate Safety Lifecycle
Concept Phase
3.7 Hazard analysis and risk assessment
3.8 Functional Safety Concept
Product development systems level
4. Systems Engineering for product development
Develop operations plan
Develop production plan
5 Hardware product development
6 Software product development
Other Technologies
Controllability
External Measures
4.8 Item Integration and testing
4.9 Safety validation
4.10 Functional safety assessment
4.11 Product release
7.5 Production
After release production
In case of modification back to appropriate lifeccyle phase
7.6 Operation, service and decomissioning

ISO 26262 Published Website

- Contains content covering
  - Main workflows and activityites for each part of 26262
- Each activity and task has links to the relevant
  - Roles
  - Input work products
  - Output work products
  - Relationships to other tasks



Rational. Method Composer

Search this Site:

ISO 26262

- Welcome to ISO 26262
- Getting Started
  - Tool Setup and Configuration
- ISO 26262 Processes
  - Main Process Flows
  - ISO 26262, Main Phases of the Safety Lifecycle
    - 2 Management of functional safety
    - 3 Concept Phase
    - 4. Systems Engineering for product development
    - 5 Hardware product development
    - 6 Software product development
    - 4. Systems Integration and testing
    - 7 Production and Operation
  - Supporting Processess
    - 8.11 Qualification of software tools
    - 8.13 Qualification of HW components
    - 8.5 Manage supplier interface
    - 8.8 Change Management
    - 8.9 Verification Planning
    - Tool Environment Setup
- Roles
  - Corporate
  - Human Resources Officer
  - Hardware Engineer
  - Independent Safety Auditor
  - Knowledge Engineer
  - Maintenance Engineer
  - Process engineer
  - Production Engineer
  - Project Manager
  - Project member
  - Quality Engineer
  - Quality Manager
  - Requirements Engineer
  - Safety Engineer
  - Safety Manager
  - Specialist tools administrator
  - Supplier Corporate
  - Supplier Project Manager
  - Supplier Safety Manager
  - Software Engineer
  - Systems Engineer
  - Team Leader
  - Technical writer

Welcome to ISO 26262

Welcome to ISO 26262

This web site details the Tasks, Work Products, Roles and workflows required for the development of safety related work item for a vehicle to be compliant to ISO 26262.

Relationships

Categories	• ISO 26262

Back to top

Main Description

As of August 2, 2011, the ISO 26262 **standard** remains in draft format with the final awaiting publication. The workflow provided in Rational Method Composer is provided to assist you in tracking information that may be important to you in aligning your system and software development processes with tasks from the draft of the ISO 26262 standard available in 2010 from the ISO store.

The guidance given on these web pages covers the main phases of development from Management of Functional Safety, through the Concept Phase and Product Development at Systems, Hardware and Software level to Production and Operation. The main Supporting processes are also detailed. The ASIL and safety orientated analysis activity may be added at a future time.

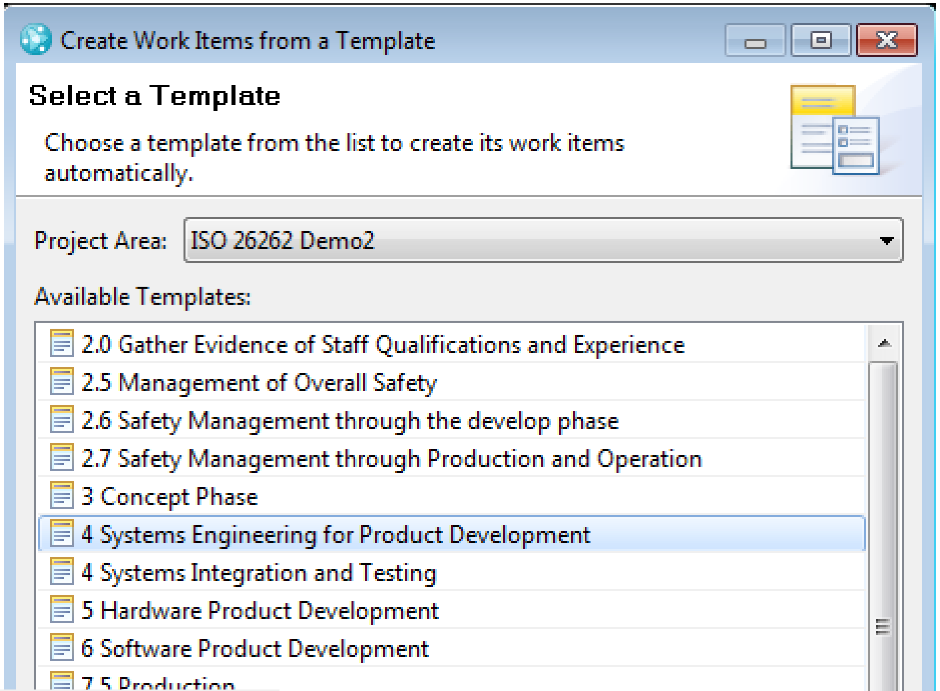
1 Vocabulary
ISO DIS 26262
2 Management of functional safety
3 Concept phase
4 Product Development on system level
4.5-4.7 Systems Engineering
4.8-4.11 Systems Integration & test
5 Hardware development
6 Software development
7 Production and operation
8 Supporting processes
9 ASIL-oriented and safety-oriented analysis
10 Guideline on ISO 26262 (informative)

■ Supported with RMC content

□ User directed to standard

ISO 26262 work item templates

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



Create Work Items from a Template

Select a Template

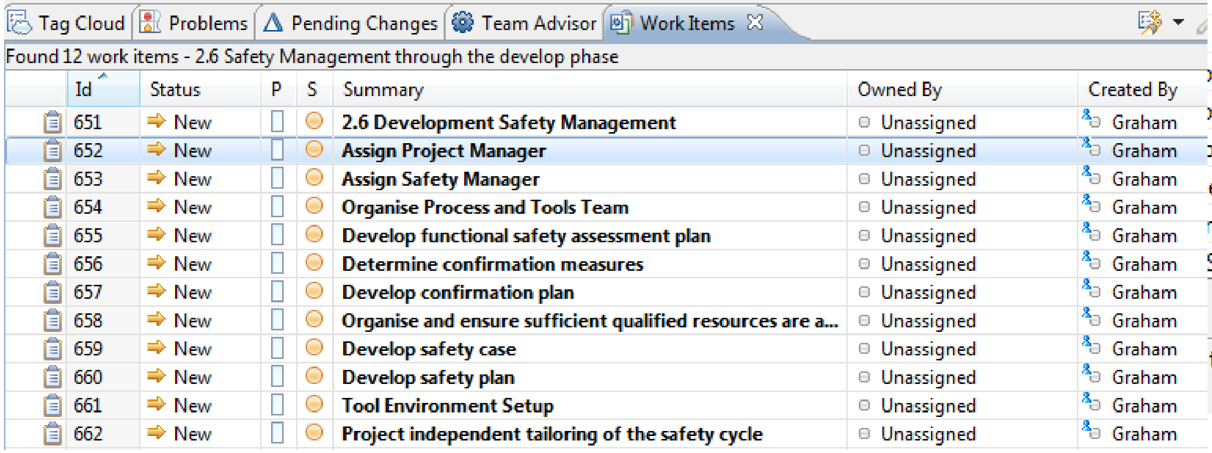
Choose a template from the list to create its work items automatically.

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
- of SW tools
- of HW components
- pplier relationship
- ement
- nning
- Set Up

the functional safety concept, the item is developed frc

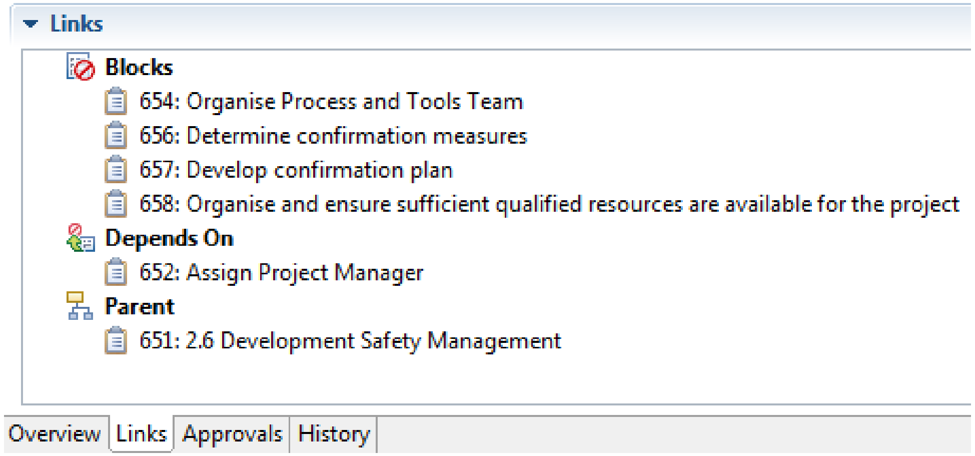


Found 12 work items - 2.6 Safety Management through the develop phase

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

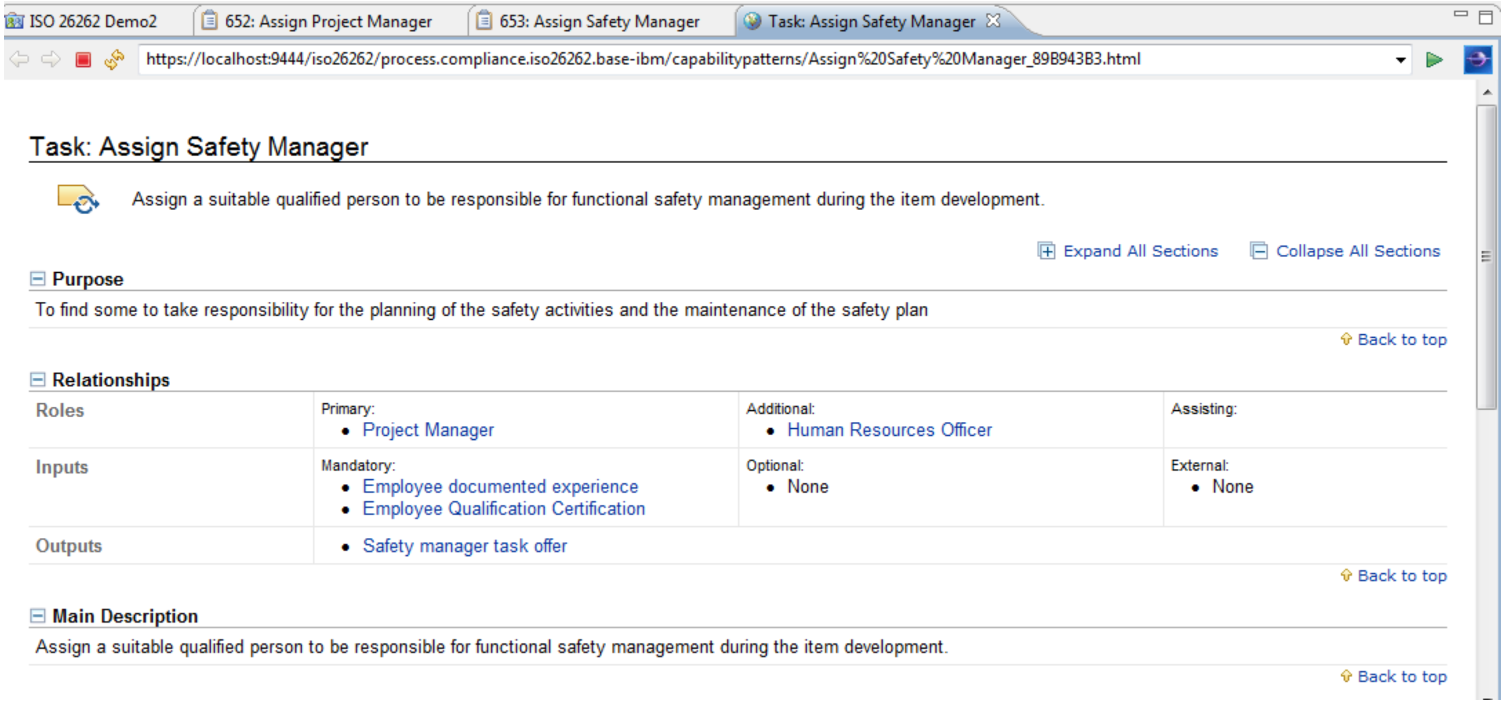


Links

Blocks
- 654: Organise Process and Tools Team
- 656: Determine confirmation measures
- 657: Develop confirmation plan
- 658: Organise and ensure sufficient qualified resources are available for the project

Depends On
- 652: Assign Project Manager

Parent
- 651: 2.6 Development Safety Management



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

Relationships

Roles	Primary: • Project Manager	Additional: • Human Resources Officer	Assisting:
Inputs	Mandatory: • Employee documented experience • Employee Qualification Certification	Optional: • None	External: • None
Outputs	• Safety manager task offer		

Main Description

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

Tool Qualification for ISO 26262

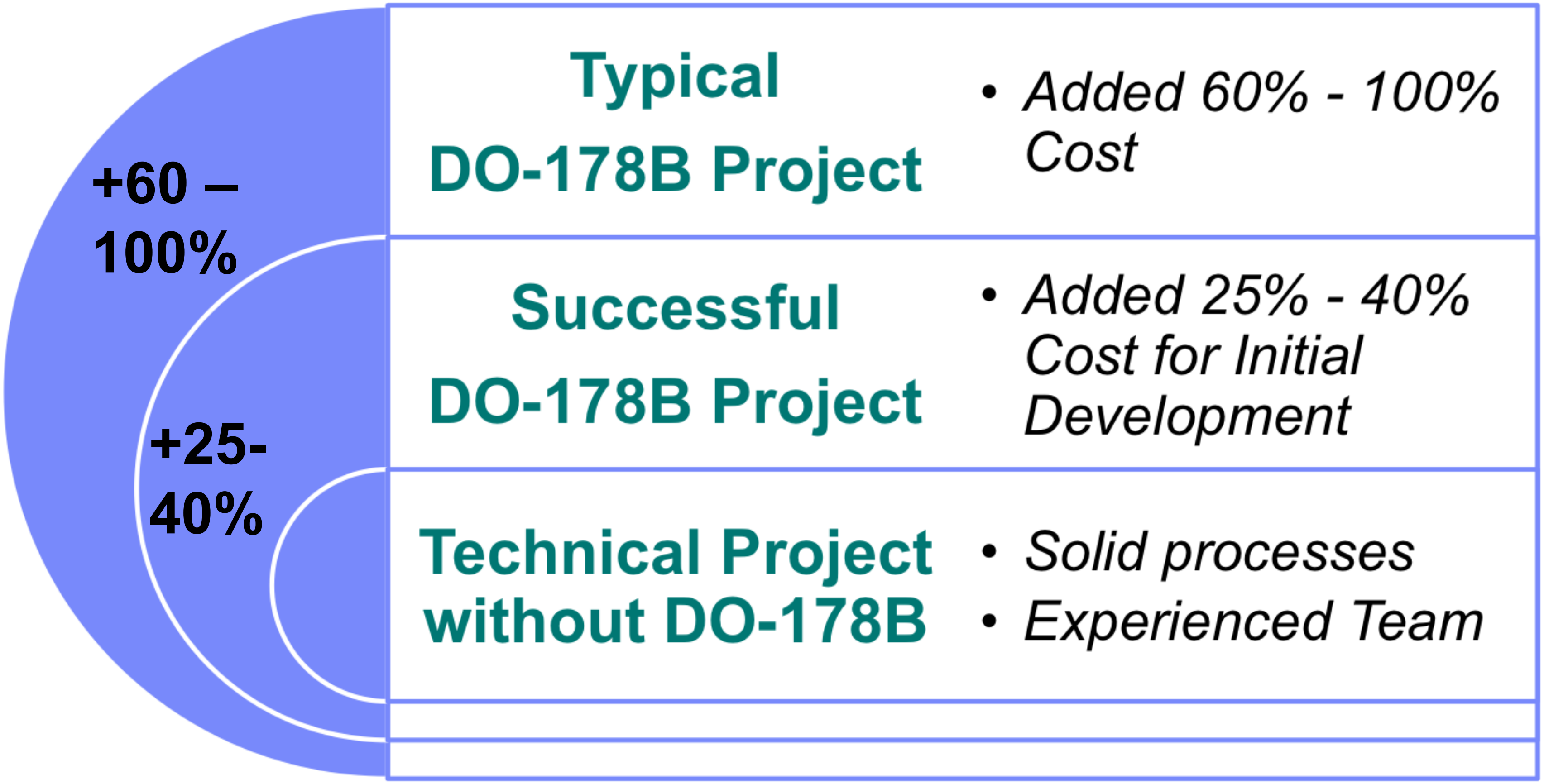
- ISO 26262 requires tools "used in the development" of safety related software be qualified
  - Unlike standards such as DO-178B, ISO 26262 spans tools used across the entire development cycle. (RM, CM, etc).
- Within the ISO 26262 standard, there is detailed guidance on tool qualification
  - Use cases for tools first documented and analyzed
  - Analysis will evaluate if malfunctioning software tool (or output from tool) can lead to violation of safety requirement
  - Probability of preventing or detecting such errors is determined.
  - This leads to Tool Confidence Level (TCL) determination
  - TCL + ASIL guides how you qualify a tool.
    - E.g. increased confidence of use is a possible tool qualification method
- The ISO/DIS 26262 tool qualification process requires the creation of the following tool qualification work products (ISO/DIS 26262-8, 11.5; see the appendix for a summary) by our customers:
  - Software Tool Qualification Plan
  - Software Tool Documentation
  - Software Tool Classification Analysis .. .For TCL determination.
  - Software Tool Qualification Report
- One step further is to have a independent authority, e.g. TUV (Technical Inspector Association) .. to audit development process to develop tool along with qualification work products.
- ***IBM Rational plans to provide tool qualification kits for ISO 26262 in the future.***

DO-178B

DO-178B at 30,000 feet

- DO-178B defines detailed guidelines for development of aviation software that performs intended functions
- The FAA accepts use of DO-178B as a means of certifying software in avionics
- DO-178B outlines the *objectives* to be met, the work activities to be performed for each objective, and the evidence (output documents) to be supplied for each objective
- Objectives are organized into process areas
  - Planning
  - Development
  - Verification
  - Configuration Management
  - Quality Assurance

Dollar$ And $ense:
Initial cost increase due to DO-178B



+60 – 100%

+25-40%

Typical DO-178B Project

- *Added 60% - 100% Cost*

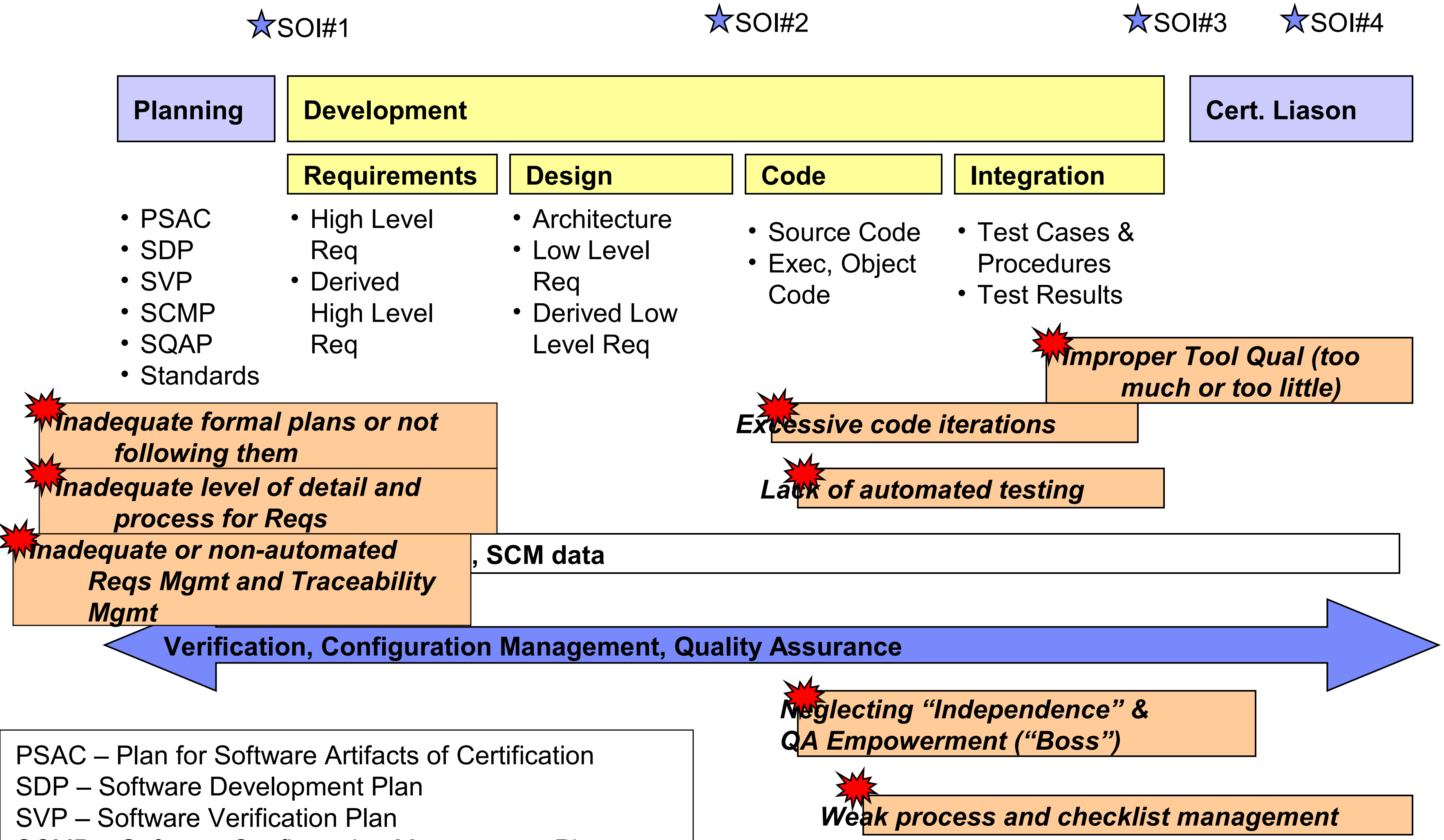
Successful DO-178B Project

- *Added 25% - 40% Cost for Initial Development*

Technical Project without DO-178B

- *Solid processes*
- *Experienced Team*

Efficiency through Automation for DO-178B



SOI#1 SOI#2 SOI#3 SOI#4

Planning

- PSAC
- SDP
- SVP
- SCMP
- SQAP
- Standards

Development

Requirements

- High Level Req
- Derived High Level Req

Design

- Architecture
- Low Level Req
- Derived Low Level Req

Code

- Source Code
- Exec, Object Code

Integration

- Test Cases & Procedures
- Test Results

Cert. Liason

Improper Tool Qual (too much or too little)

Excessive code iterations

Lack of automated testing

Inadequate formal plans or not following them

Inadequate level of detail and process for Reqs

Inadequate or non-automated Reqs Mgmt and Traceability Mgmt

, SCM data

Verification, Configuration Management, Quality Assurance

Neglecting "Independence" & QA Empowerment ("Boss")

Weak process and checklist management

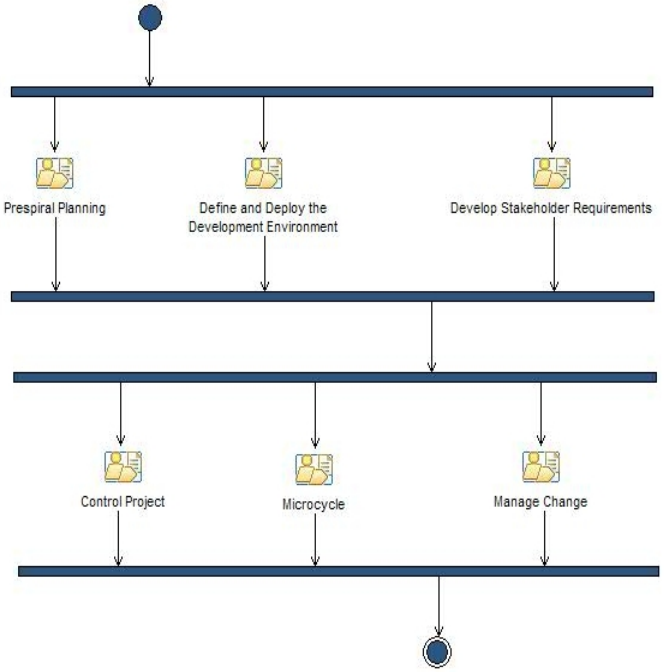
PSAC – Plan for Software Artifacts of Certification
SDP – Software Development Plan
SVP – Software Verification Plan
SCMP – Software Configuration Management Plan
SQAP – Software Quality Assurance Plan

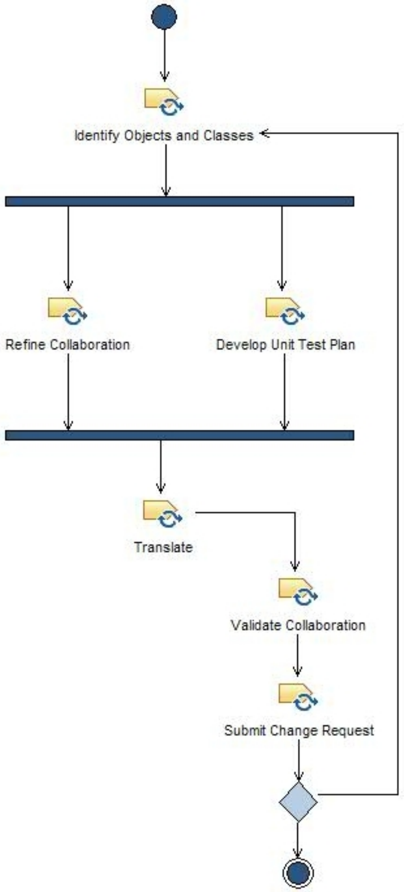
ISDP-178

- The Integrated Software Development Process for DO-178B (ISDP-178) is a set of practices to help organizations developing products for certification under DO-178B
- The process may be applied to any appropriate development tooling but is specifically optimized for the Rational System Accelerator consisting of tools
  - Rational Team Concert
  - Rational DOORS
  - Rational Rhapsody
  - Rational Quality Manager
- The ISDP-178 address three primary needs
  - Process specification
  - Process enactment
  - Specific links from the DO-178B standard to process content to aid in ensuring compliance
    - By Objective
    - By Certification Level
    - By Work Product
    - By Checklist

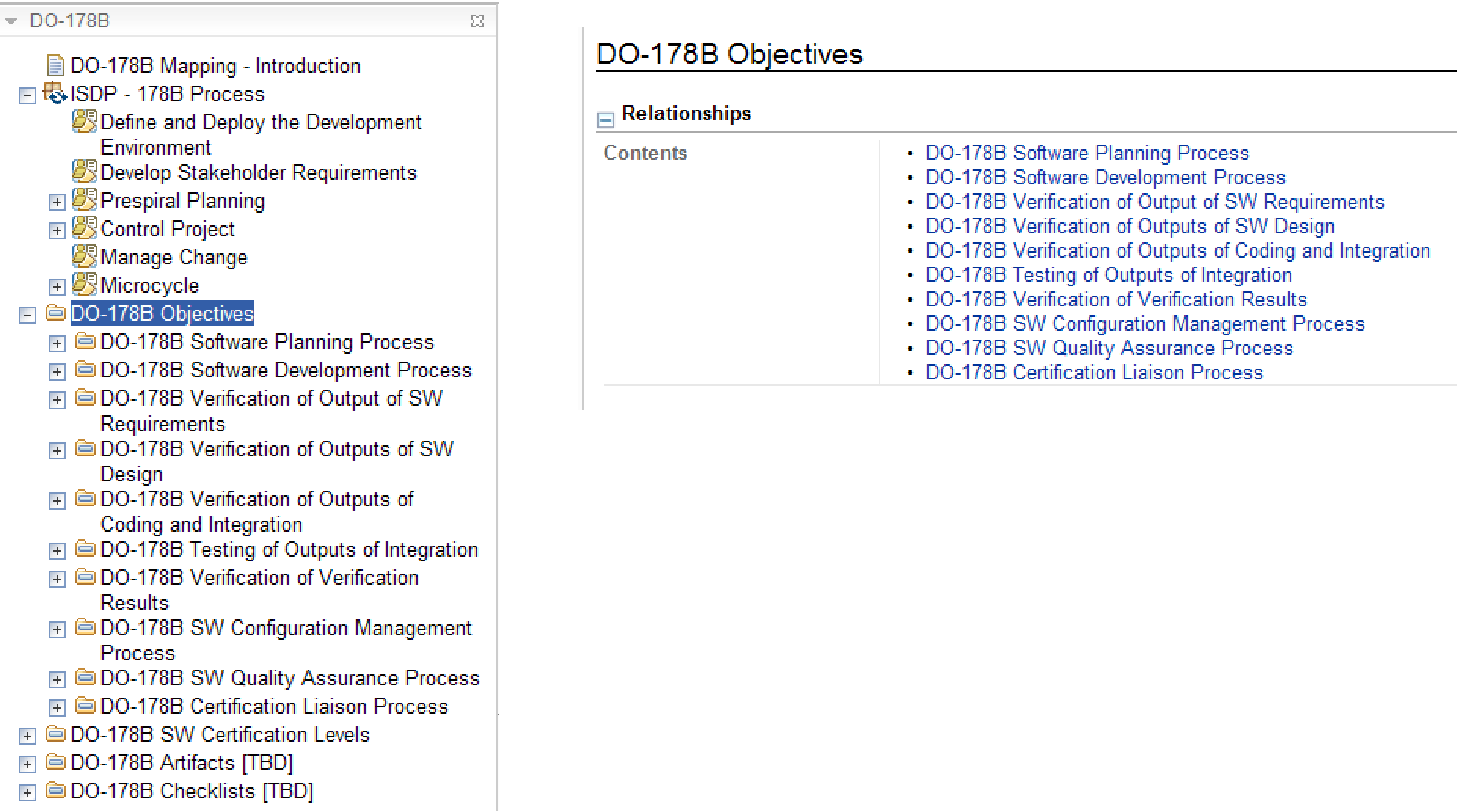
ISDP-178 Process Definition

- The ISDP-178 Process consists of a delivery process composed of a number of best practices, including:
  - Prespiral Planning
  - Developing Requirements
  - Defining and Deploying the Development Environment
  - Project Control (governance)
  - Change Management
  - Configuration Management
  - Incremental Iterative Development
  - High Fidelity Modeling
  - Real-time Embedded Architecture
  - Collaborative Design
  - Continuous Integration
  - Verification and Validation


Prespiral Planning
Define and Deploy the Development Environment
Develop Stakeholder Requirements
Control Project
Microcycle
Manage Change



ISDP-178 Links to DO-178B Standard Content



DO-178B

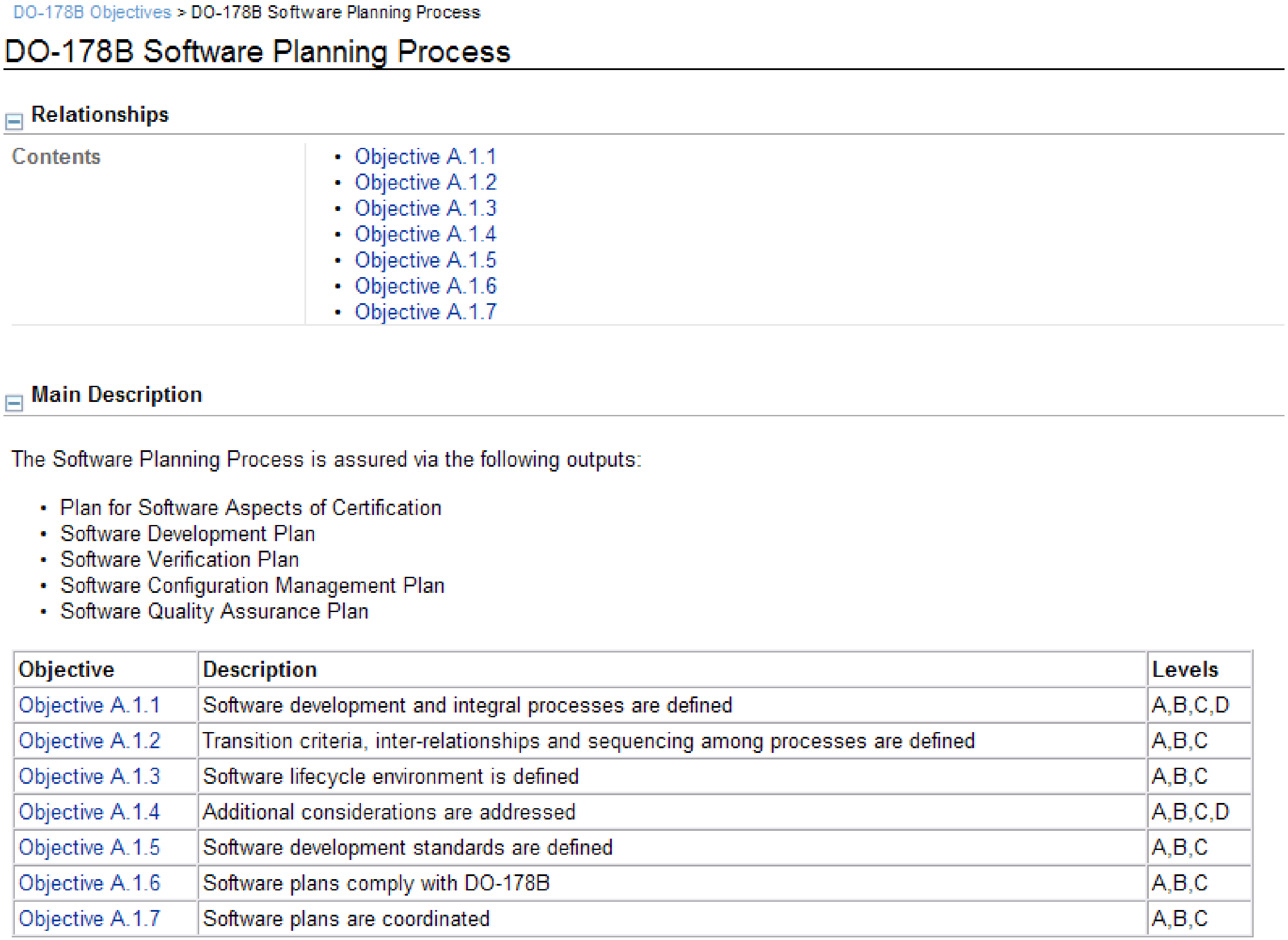
- DO-178B Mapping - Introduction
- ISDP - 178B Process
  - Define and Deploy the Development Environment
  - Develop Stakeholder Requirements
  - Prespiral Planning
  - Control Project
  - Manage Change
  - Microcycle
- DO-178B Objectives
  - DO-178B Software Planning Process
  - DO-178B Software Development Process
  - DO-178B Verification of Output of SW Requirements
  - DO-178B Verification of Outputs of SW Design
  - DO-178B Verification of Outputs of Coding and Integration
  - DO-178B Testing of Outputs of Integration
  - DO-178B Verification of Verification Results
  - DO-178B SW Configuration Management Process
  - DO-178B SW Quality Assurance Process
  - DO-178B Certification Liaison Process
- DO-178B SW Certification Levels
- DO-178B Artifacts [TBD]
- DO-178B Checklists [TBD]

DO-178B Objectives

Relationships

Contents	• DO-178B Software Planning Process • DO-178B Software Development Process • DO-178B Verification of Output of SW Requirements • DO-178B Verification of Outputs of SW Design • DO-178B Verification of Outputs of Coding and Integration • DO-178B Testing of Outputs of Integration • DO-178B Verification of Verification Results • DO-178B SW Configuration Management Process • DO-178B SW Quality Assurance Process • DO-178B Certification Liaison Process

ISDP-178 Links to DO-178B Objectives



DO-178B Objectives > DO-178B Software Planning Process

DO-178B Software Planning Process

Relationships

Contents	• Objective A.1.1 • Objective A.1.2 • Objective A.1.3 • Objective A.1.4 • Objective A.1.5 • Objective A.1.6 • Objective A.1.7

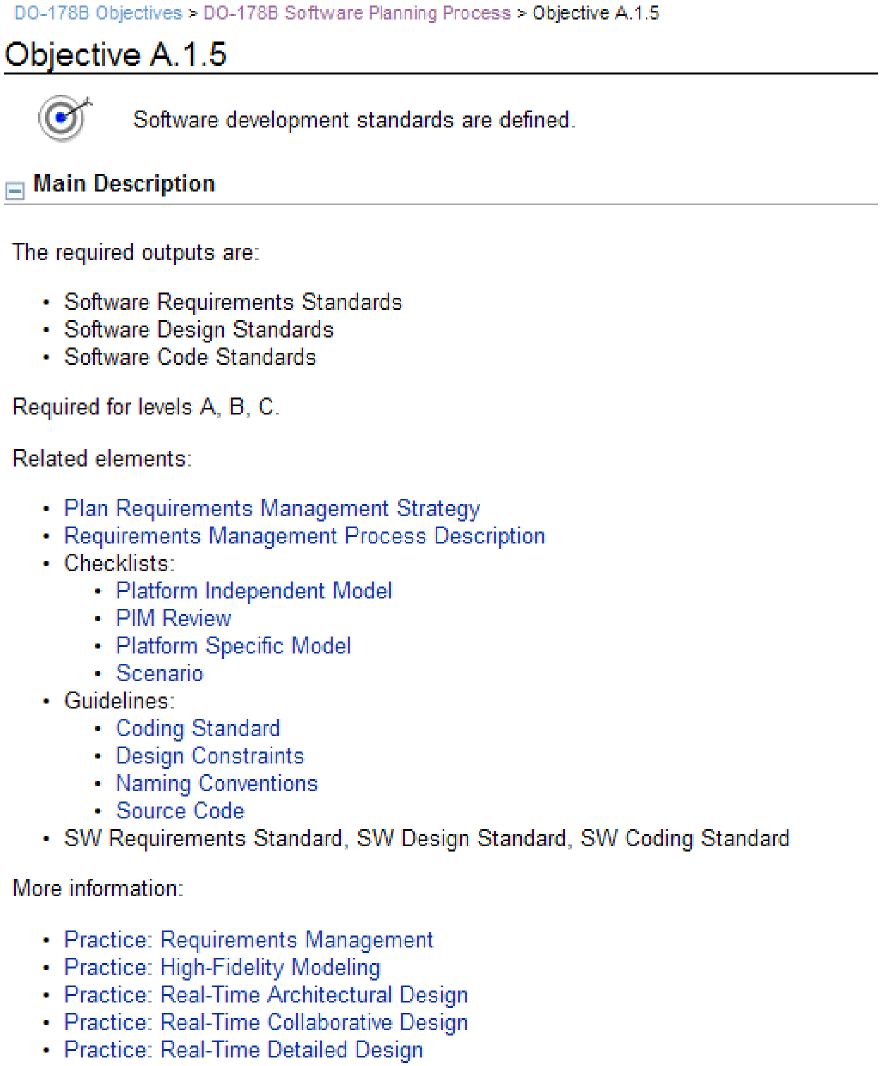
Main Description

The Software Planning Process is assured via the following outputs:

- Plan for Software Aspects of Certification
- Software Development Plan
- Software Verification Plan
- Software Configuration Management Plan
- Software Quality Assurance Plan

Objective	Description	Levels
Objective A.1.1	Software development and integral processes are defined	A,B,C,D
Objective A.1.2	Transition criteria, inter-relationships and sequencing among processes are defined	A,B,C
Objective A.1.3	Software lifecycle environment is defined	A,B,C
Objective A.1.4	Additional considerations are addressed	A,B,C,D
Objective A.1.5	Software development standards are defined	A,B,C
Objective A.1.6	Software plans comply with DO-178B	A,B,C
Objective A.1.7	Software plans are coordinated	A,B,C

ISDP-178 Links to DO-178B Objectives



DO-178B Objectives > DO-178B Software Planning Process > Objective A.1.5

Objective A.1.5

Software development standards are defined.

Main Description

The required outputs are:

- Software Requirements Standards
- Software Design Standards
- Software Code Standards

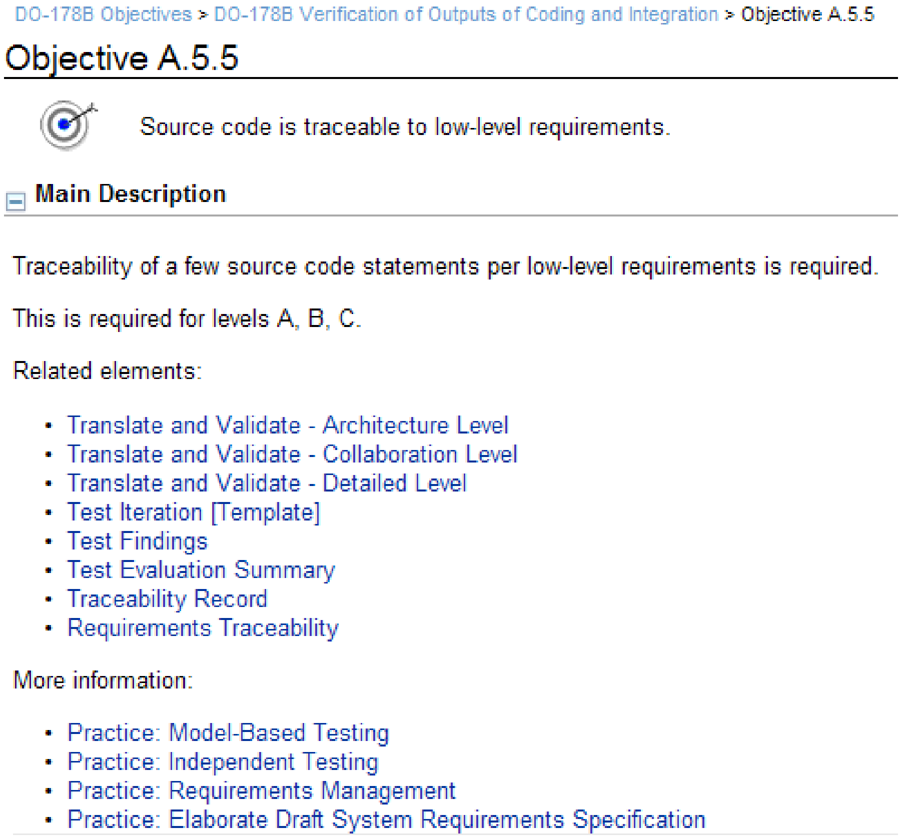
Required for levels A, B, C.

Related elements:

- Plan Requirements Management Strategy
- Requirements Management Process Description
- Checklists:
  - Platform Independent Model
  - PIM Review
  - Platform Specific Model
  - Scenario
- Guidelines:
  - Coding Standard
  - Design Constraints
  - Naming Conventions
  - Source Code
- SW Requirements Standard, SW Design Standard, SW Coding Standard

More information:

- Practice: Requirements Management
- Practice: High-Fidelity Modeling
- Practice: Real-Time Architectural Design
- Practice: Real-Time Collaborative Design
- Practice: Real-Time Detailed Design



DO-178B Objectives > DO-178B Verification of Outputs of Coding and Integration > Objective A.5.5

Objective A.5.5

Source code is traceable to low-level requirements.

Main Description

Traceability of a few source code statements per low-level requirements is required.

This is required for levels A, B, C.

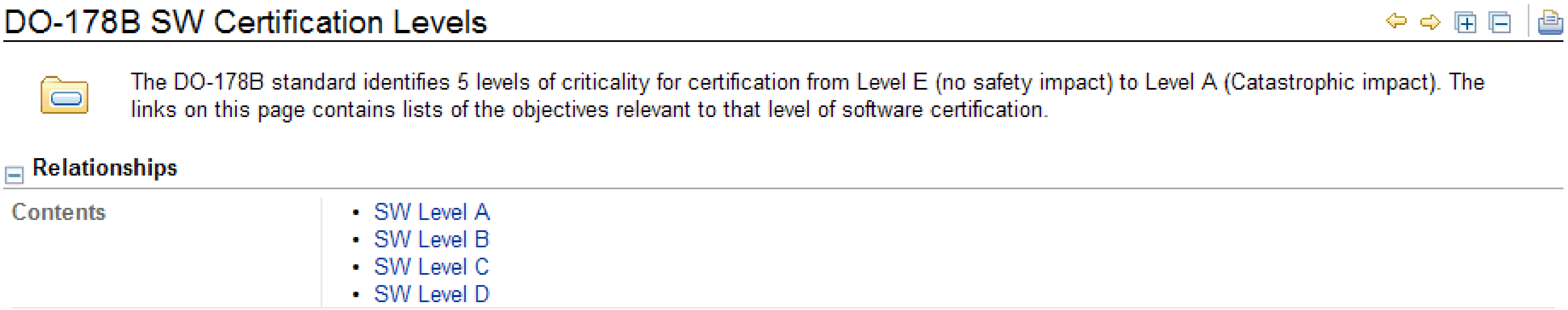
Related elements:

- Translate and Validate - Architecture Level
- Translate and Validate - Collaboration Level
- Translate and Validate - Detailed Level
- Test Iteration [Template]
- Test Findings
- Test Evaluation Summary
- Traceability Record
- Requirements Traceability

More information:

- Practice: Model-Based Testing
- Practice: Independent Testing
- Practice: Requirements Management
- Practice: Elaborate Draft System Requirements Specification

ISDP-178 Links to DO-178B by Certification Level



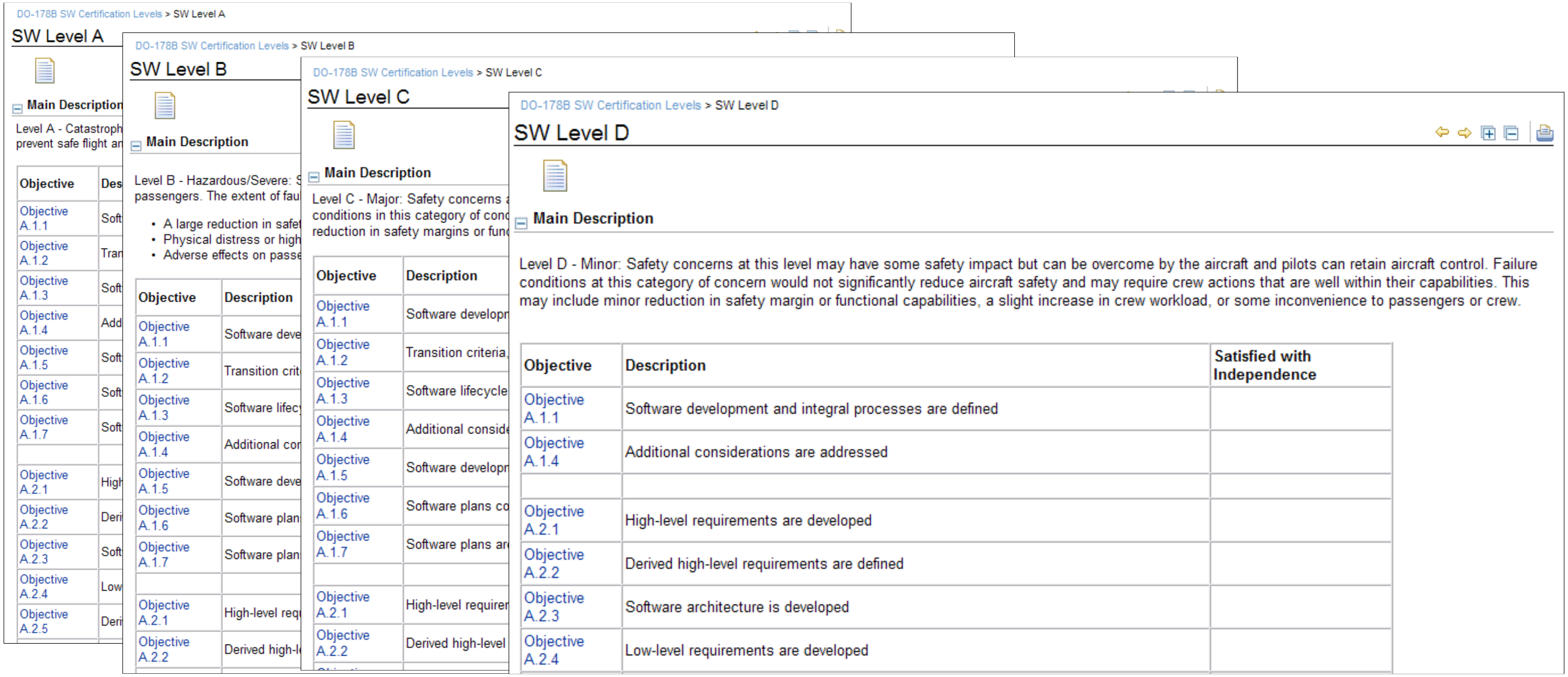
DO-178B SW Certification Levels

The DO-178B standard identifies 5 levels of criticality for certification from Level E (no safety impact) to Level A (Catastrophic impact). The links on this page contains lists of the objectives relevant to that level of software certification.

Relationships

Contents

- SW Level A
- SW Level B
- SW Level C
- SW Level D



DO-178B SW Certification Levels > SW Level A

SW Level A

Main Description

Level A - Catastroph
prevent safe flight an

Objective	Des
Objective A.1.1	Soft
Objective A.1.2	Tran
Objective A.1.3	Soft
Objective A.1.4	Add
Objective A.1.5	Soft
Objective A.1.6	Soft
Objective A.1.7	Soft
Objective A.2.1	High
Objective A.2.2	Deri
Objective A.2.3	Soft
Objective A.2.4	Low
Objective A.2.5	Deri

DO-178B SW Certification Levels > SW Level B

SW Level B

Main Description

Level B - Hazardous/Severe: S
passengers. The extent of faul

- A large reduction in safet
- Physical distress or high
- Adverse effects on passe

Objective	Description
Objective A.1.1	Software deve
Objective A.1.2	Transition crit
Objective A.1.3	Software lifecy
Objective A.1.4	Additional cor
Objective A.1.5	Software deve
Objective A.1.6	Software plan
Objective A.1.7	Software plan
Objective A.2.1	High-level requ
Objective A.2.2	Derived high-l

DO-178B SW Certification Levels > SW Level C

SW Level C

Main Description

Level C - Major: Safety concerns a
conditions in this category of conc
reduction in safety margins or func

Objective	Description
Objective A.1.1	Software developn
Objective A.1.2	Transition criteria,
Objective A.1.3	Software lifecycle
Objective A.1.4	Additional conside
Objective A.1.5	Software developn
Objective A.1.6	Software plans co
Objective A.1.7	Software plans ar
Objective A.2.1	High-level requirer
Objective A.2.2	Derived high-level

DO-178B SW Certification Levels > SW Level D

SW Level D

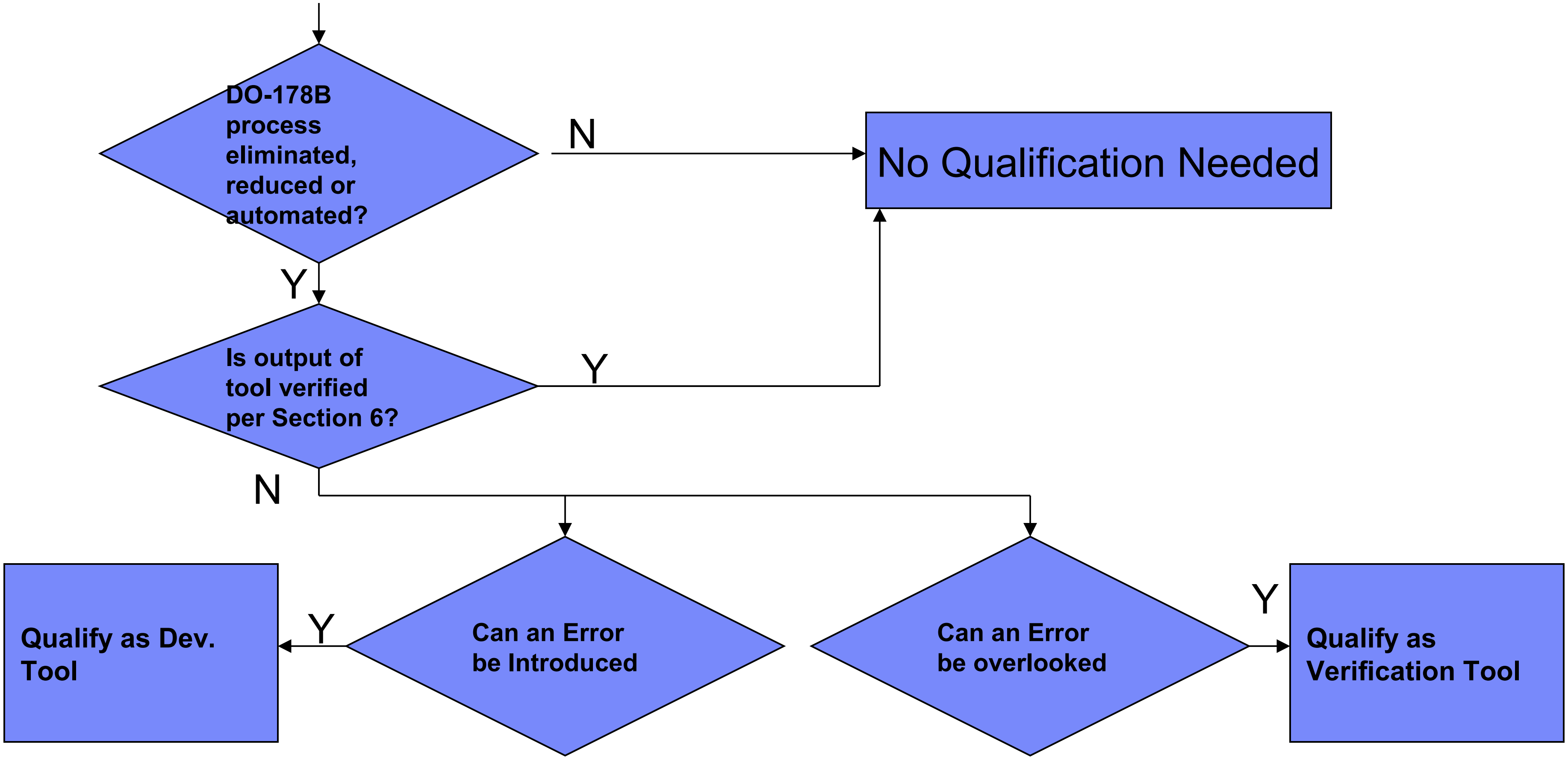
Main Description

Level D - Minor: Safety concerns at this level may have some safety impact but can be overcome by the aircraft and pilots can retain aircraft control. Failure conditions at this category of concern would not significantly reduce aircraft safety and may require crew actions that are well within their capabilities. This may include minor reduction in safety margin or functional capabilities, a slight increase in crew workload, or some inconvenience to passengers or crew.

Objective	Description	Satisfied with Independence
Objective A.1.1	Software development and integral processes are defined	
Objective A.1.4	Additional considerations are addressed	
Objective A.2.1	High-level requirements are developed	
Objective A.2.2	Derived high-level requirements are defined	
Objective A.2.3	Software architecture is developed	
Objective A.2.4	Low-level requirements are developed	

Tool Qualification for DO-178B

- Is Tool Qualification Necessary?
  - Generally not. Ask these questions:

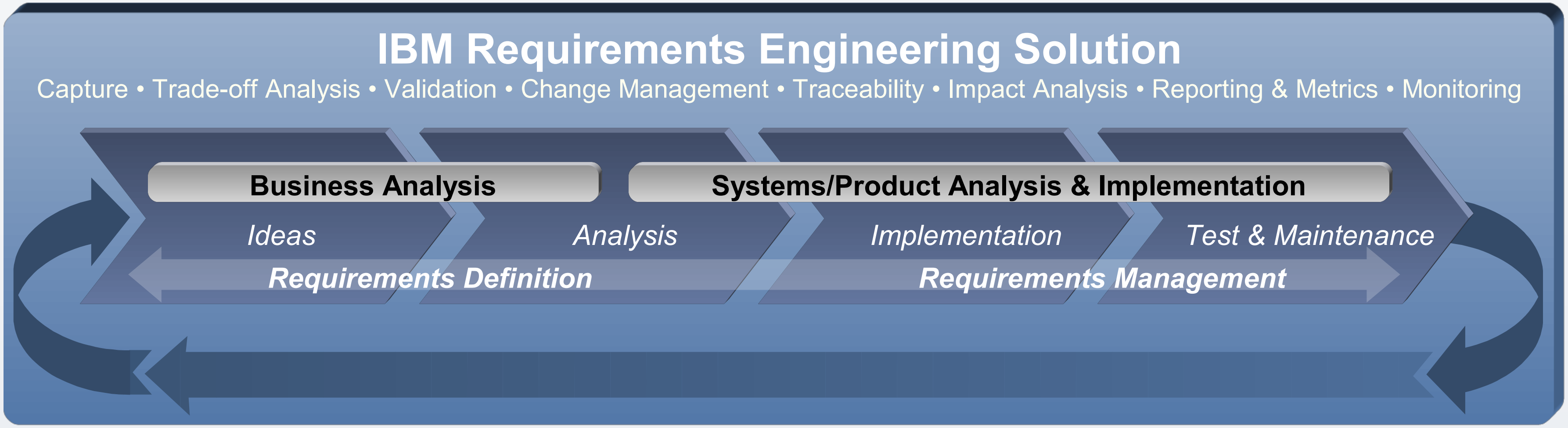

DO-178B process eliminated, reduced or automated?
N
No Qualification Needed
Y
Is output of tool verified per Section 6?
Y
N
Qualify as Dev. Tool
Y
Can an Error be Introduced
Can an Error be overlooked
Y
Qualify as Verification Tool

IBM Rational DO-178B Qualification kits available

IBM Rational Solutions:

- **IBM Rational Test RealTime (System Test, Dynamic Code Coverage for Level A MC/DC & Multiple Decision Coverage, Static Analysis, Memory, Performance & Thread profiling Analysis, Dynamic Trace Capture, Unit Test Automation, Software Metrics, Reporting)**
- **IBM Rational Logiscope (Static Analysis, Dynamic Code coverage for Level A MC/DC & Multiple Decision Coverage, Software Metrics, Helps in code refactoring and identifying duplicate code, Reporting)**

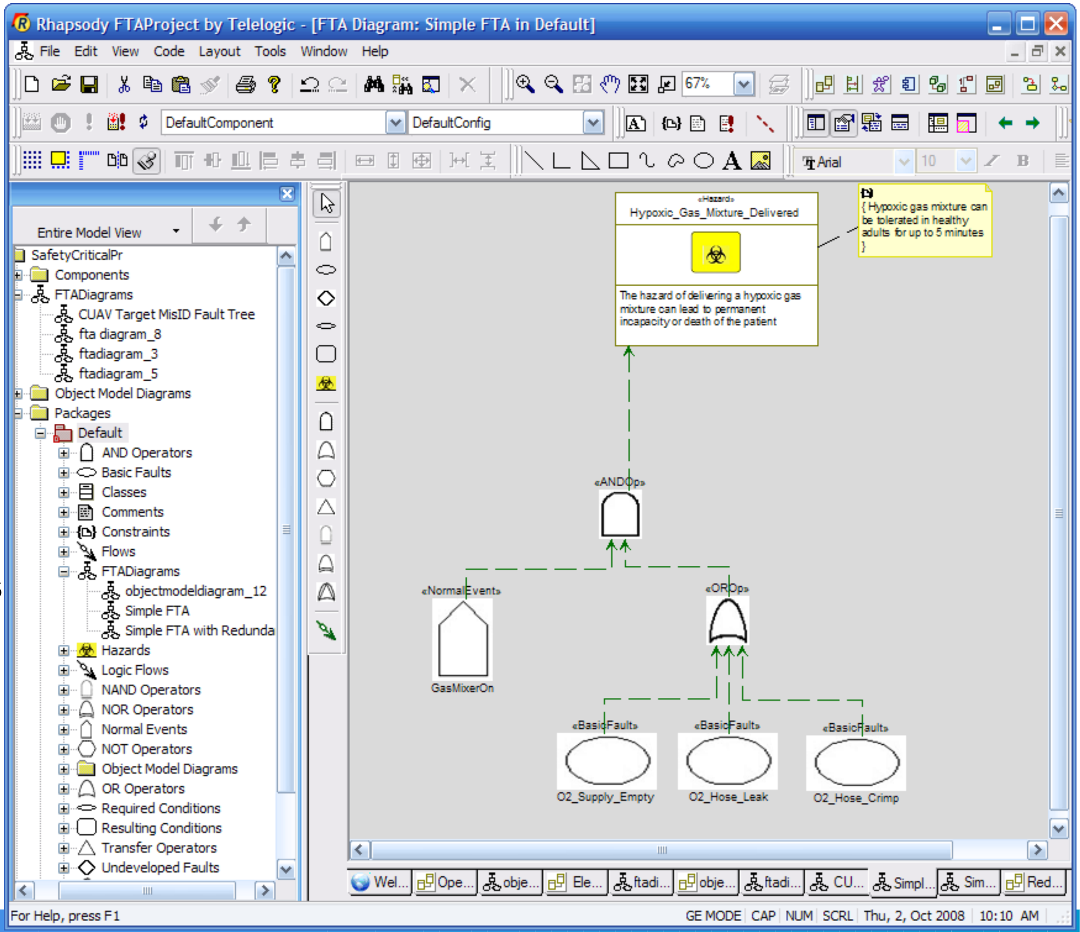
Quality begins with Requirements: IBM Requirements Engineering Solution


IBM Requirements Engineering Solution
Capture • Trade-off Analysis • Validation • Change Management • Traceability • Impact Analysis • Reporting & Metrics • Monitoring
Business Analysis
Systems/Product Analysis & Implementation
Ideas
Analysis
Implementation
Test & Maintenance
Requirements Definition
Requirements Management

- Getting everyone on the same page
  - Includes suppliers and subcontractors
- Managing scope, plus assessing and controlling the impact of change
- Ensuring end-to-end traceability
  - From ideas, feature definitions, product specifications and models…
  - To mechanical, electric/electronic and embedded software implementation, test and maintenance
- Ensuring conformance to contractual agreements
- **Demonstrating compliance to regulations such as ISO 26262**

Safety-Critical Profile in UML for Rhapsody

- 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
- Work going on with KVI
  - Medini tool
  - Safety analysis
  - Integrate with Rhapsody and RTC
- Work going on with Inchron
  - Safety critical performance test analys
  - Integrates with Rhapsody

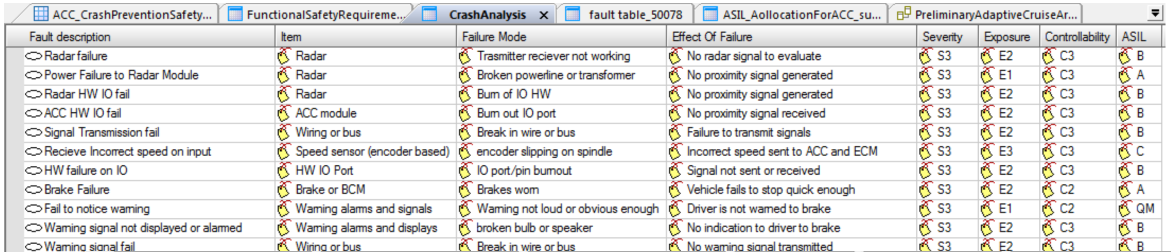


The hazard of delivering a hypoxic gas mixture can lead to permanent incapacity or death of the patient

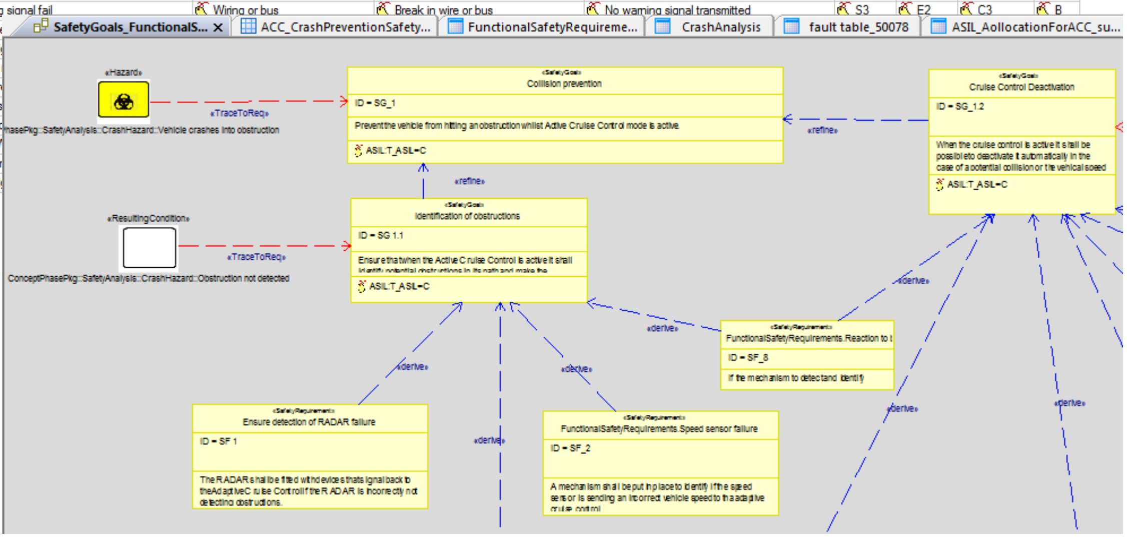
{ Hypoxic gas mixture can be tolerated in healthy adults for up to 5 minutes }

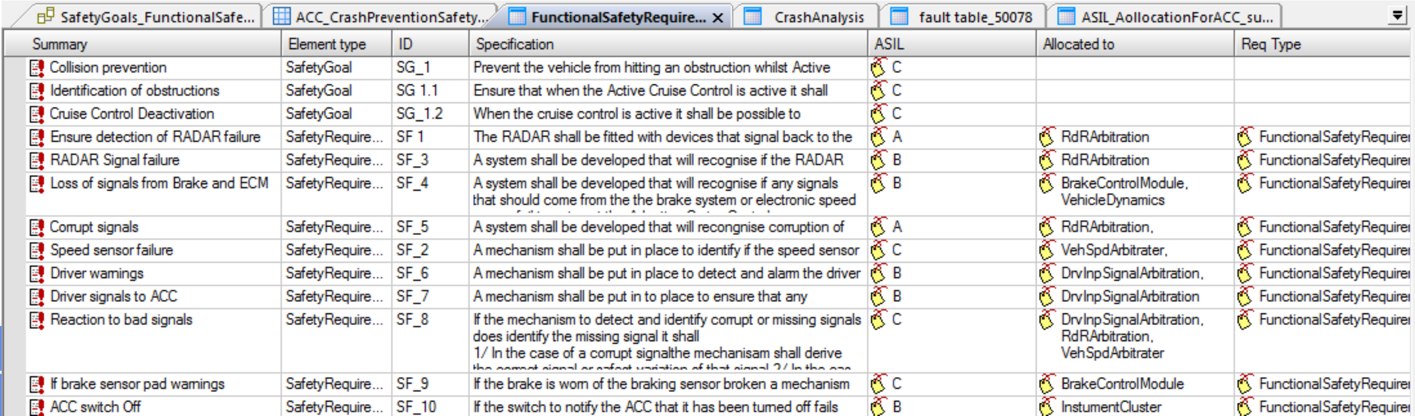
Automotive Safety Analysis Profile

- 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


ACC_CrashPreventionSafety...
FunctionalSafetyRequireme...
CrashAnalysis
fault table_50078
ASIL_AollocationForACC_su...
PreliminaryAdaptiveCruiseAr...

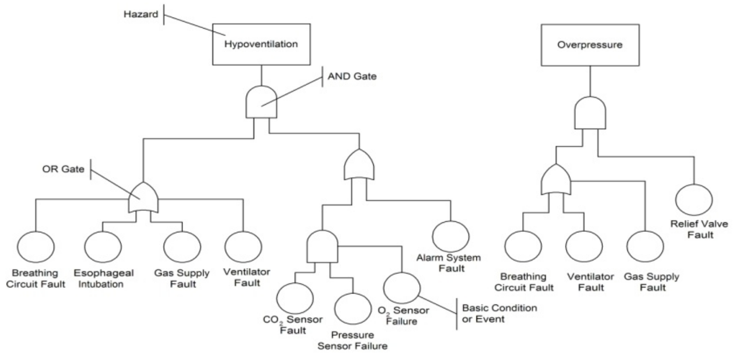
Fault description	Item	Failure Mode	Effect Of Failure	Severity	Exposure	Controllability	ASIL
Radar failure	Radar	Trasmitter reciever not working	No radar signal to evaluate	S3	E2	C3	B
Power Failure to Radar Module	Radar	Broken powerline or transformer	No proximity signal generated	S3	E1	C3	A
Radar HW IO fail	Radar	Bum of IO HW	No proximity signal generated	S3	E2	C3	B
ACC HW IO fail	ACC module	Bum out IO port	No proximity signal received	S3	E2	C3	B
Signal Transmission fail	Wiring or bus	Break in wire or bus	Failure to transmit signals	S3	E2	C3	B
Recieve Incorrect speed on input	Speed sensor (encoder based)	encoder slipping on spindle	Incorrect speed sent to ACC and ECM	S3	E3	C3	C
HW failure on IO	HW IO Port	IO port/pin bumout	Signal not sent or received	S3	E2	C3	B
Brake Failure	Brake or BCM	Brakes wom	Vehicle fails to stop quick enough	S3	E2	C2	A
Fail to notice waming	Waming alarms and signals	Waming not loud or obvious enough	Driver is not wamed to brake	S3	E1	C2	QM
Waming signal not displayed or alarmed	Waming alarms and displays	broken bulb or speaker	No indication to driver to brake	S3	E2	C3	B
Waming signal fail	Wiring or bus	Break in wire or bus	No waming signal transmitted	S3	E2	C3	B
Brake se							
brake si							
ACC IO							
Speed r							
HW bus							
Off switc							
ACC HW							
Signal lir							
brake si							


SafetyGoals_FunctionalS...
ACC_CrashPreventionSafety...
FunctionalSafetyRequireme...
CrashAnalysis
fault table_50078
ASIL_AollocationForACC_su...
«Hazard»
«TraceToReq»
hasePkg::SafetyAnalysis::CrashHazard::Vehicle crashes into obstruction
«SafetyGoal»
Collision prevention
ID = SG_1
Prevent the vehicle from hitting an obstruction whilst Active Cruise Control mode is active
ASIL:T_ASIL=C
«refine»
«SafetyGoal»
Cruise Control Deactivation
ID = SG_1.2
When the cruise control is active it shall be possible to deactivate it automatically in the case of a potential collision or the vehical speed
ASIL:T_ASIL=C
«refine»
«ResultingCondition»
«TraceToReq»
ConceptPhasePkg::SafetyAnalysis::CrashHazard::Obstruction not detected
«SafetyGoal»
Identification of obstructions
ID = SG 1.1
Ensure that when the Active Cruise Control is active it shall identify potential obstructions in its path and make the
ASIL:T_ASIL=C
«derive»
«derive»
«derive»
«derive»
«derive»
«derive»
«derive»
«SafetyRequirement»
FunctionalSafetyRequirements.Reaction to b
ID = SF_8
If the mechanism to detect and identify
«SafetyRequirement»
Ensure detection of RADAR failure
ID = SF 1
The RADAR shall be fitted with devices that signal back to the Adaptive Cruise Control if the RADAR is incorrectly not detecting obstructions.
«SafetyRequirement»
FunctionalSafetyRequirements.Speed sensor failure
ID = SF_2
A mechanism shall be put in place to identify if the speed sensor is sending an incorrect vehicle speed to the adaptive cruise control


SafetyGoals_FunctionalSafe...
ACC_CrashPreventionSafety...
FunctionalSafetyRequire...
CrashAnalysis
fault table_50078
ASIL_AollocationForACC_su...

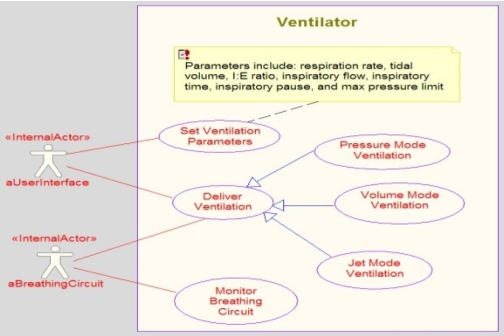
Summary	Element type	ID	Specification	ASIL	Allocated to	Req Type
Collision prevention	SafetyGoal	SG_1	Prevent the vehicle from hitting an obstruction whilst Active	C		
Identification of obstructions	SafetyGoal	SG 1.1	Ensure that when the Active Cruise Control is active it shall	C		
Cruise Control Deactivation	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	FunctionalSafetyRequire
RADAR Signal failure	SafetyRequire...	SF_3	A system shall be developed that will recognise if the RADAR	B	RdRArbitration	FunctionalSafetyRequire
Loss of signals from Brake and ECM	SafetyRequire...	SF_4	A system shall be developed that will recognise if any signals that should come from the the brake system or electronic speed	B	BrakeControlModule, VehicleDynamics	FunctionalSafetyRequire
Comupt signals	SafetyRequire...	SF_5	A system shall be developed that will recongnise comuption of	A	RdRArbitration,	FunctionalSafetyRequire
Speed sensor failure	SafetyRequire...	SF_2	A mechanism shall be put in place to identify if the speed sensor	C	VehSpdArbitrater,	FunctionalSafetyRequire
Driver wamings	SafetyRequire...	SF_6	A mechanism shall be put in place to detect and alam the driver	B	DrvInpSignalArbitration,	FunctionalSafetyRequire
Driver signals to ACC	SafetyRequire...	SF_7	A mechanism shall be put in to place to ensure that any	B	DrvInpSignalArbitration	FunctionalSafetyRequire
Reaction to bad signals	SafetyRequire...	SF_8	If the mechanism to detect and identify comupt or missing signals does identify the missing signal it shall 1/ In the case of a comupt signalthe mechanisam shall derive	C	DrvInpSignalArbitration, RdRArbitration, VehSpdArbitrater	FunctionalSafetyRequire
If brake sensor pad wamings	SafetyRequire...	SF_9	If the brake is wom of the braking sensor broken a mechanism	C	BrakeControlModule	FunctionalSafetyRequire
ACC switch Off	SafetyRequire...	SF_10	If the switch to notify the ACC that it has been tumed off fails	B	InstumentCluster	FunctionalSafetyRequire

Integrate Safety Design into Design from the beginning



Safety Analysis:

- **Fault Tree Analysis (FTA)**
- **Fault Means and Effective Analysis (FMEA)**
- **Hazard Analysis**

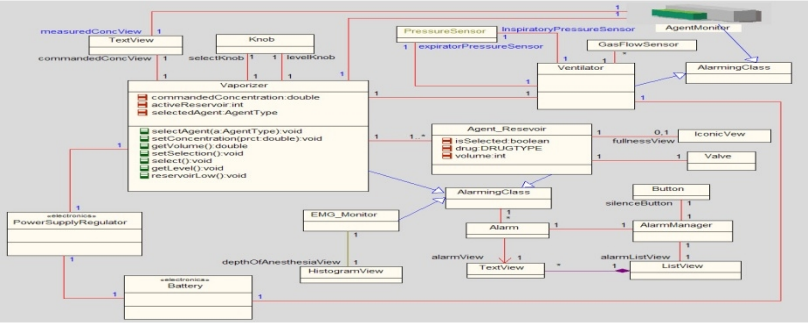

Ventilator
Parameters include: respiration rate, tidal volume, I:E ratio, inspiratory flow, inspiratory time, inspiratory pause, and max pressure limit
«InternalActor»
aUserInterface
«InternalActor»
aBreathingCircuit
Set Ventilation Parameters
Pressure Mode Ventilation
Deliver Ventilation
Volume Mode Ventilation
Jet Mode Ventilation
Monitor Breathing Circuit

The ventilator shall allow settings of inspiratory flow from 1 to 100 L/min with a default of 50 (neonate) or 1 to 180 L/min with a default of 100 (adult).

Inspiratory time shall be settable from 0.1 to 3.0 seconds with a default of 2.0 (neonate) or from 0.1 to 5.0 seconds with a default of 3.0 (adult).

Requirements Analysis:

- **Functional and Non-Functional Requirements**
- **Safety Requirements**
- **Business and Regulatory Requirements**



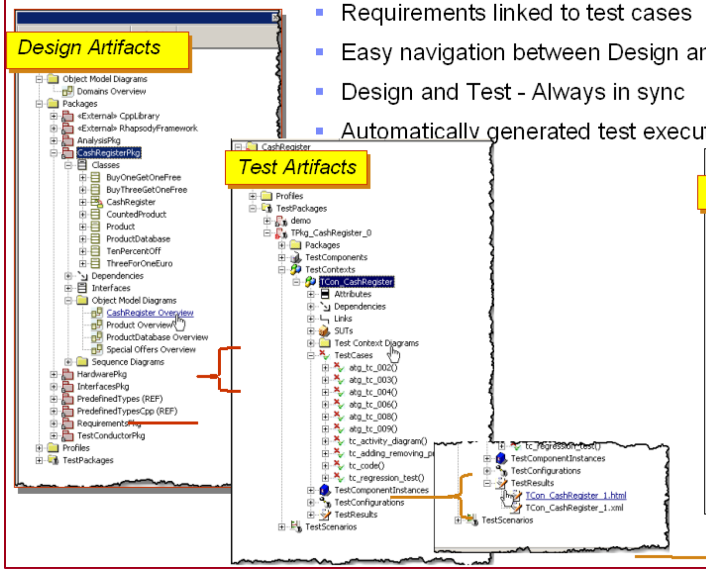
System and Software Design:

- **Structural**
- **Behavioral**
- **Temporal**
- **…**

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



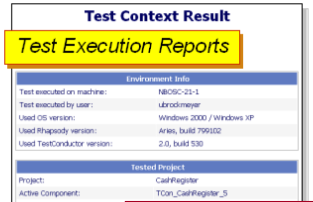
Design Artifacts

```
Object Model Diagrams
    Domains Overview
Packages
    «External» CppLibrary
    «External» RhapsodyFramework
    AnalysisPkg
    CashRegisterPkg
        Classes
            BuyOneGetOneFree
            BuyThreeGetOneFree
            CashRegister
            CountedProduct
            Product
            ProductDatabase
            TenPercentOff
            ThreeForOneEuro
        Dependencies
        Interfaces
        Object Model Diagrams
            CashRegister Overview
            Product Overview
            ProductDatabase Overview
            Special Offers Overview
        Sequence Diagrams
    HardwarePkg
    InterfacesPkg
    PredefinedTypes (REF)
    PredefinedTypesCpp (REF)
    RequirementsPkg
    TestConductorPkg
Profiles
TestPackages
```

Test Artifacts

```
CashRegister
    Profiles
    TestPackages
        demo
        TPkg_CashRegister_0
            Packages
            TestComponents
            TestContexts
                TCon_CashRegister
                    Attributes
                    Dependencies
                    Links
                    SUTs
                    Test Context Diagrams
                    TestCases
                        atg_tc_002()
                        atg_tc_003()
                        atg_tc_004()
                        atg_tc_006()
                        atg_tc_008()
                        atg_tc_009()
                        tc_activity_diagram()
                        tc_adding_removing_p
                        tc_code()
                        tc_regression_test()
                    TestComponentInstances
                    TestConfigurations
                    TestResults
            TestScenarios
```

```
        tc_regression_test()
    TestComponentInstances
    TestConfigurations
    TestResults
        TCon_CashRegister_1.html
        TCon_CashRegister_1.xml
TestScenarios
```



Test Context Result

Test Execution Reports

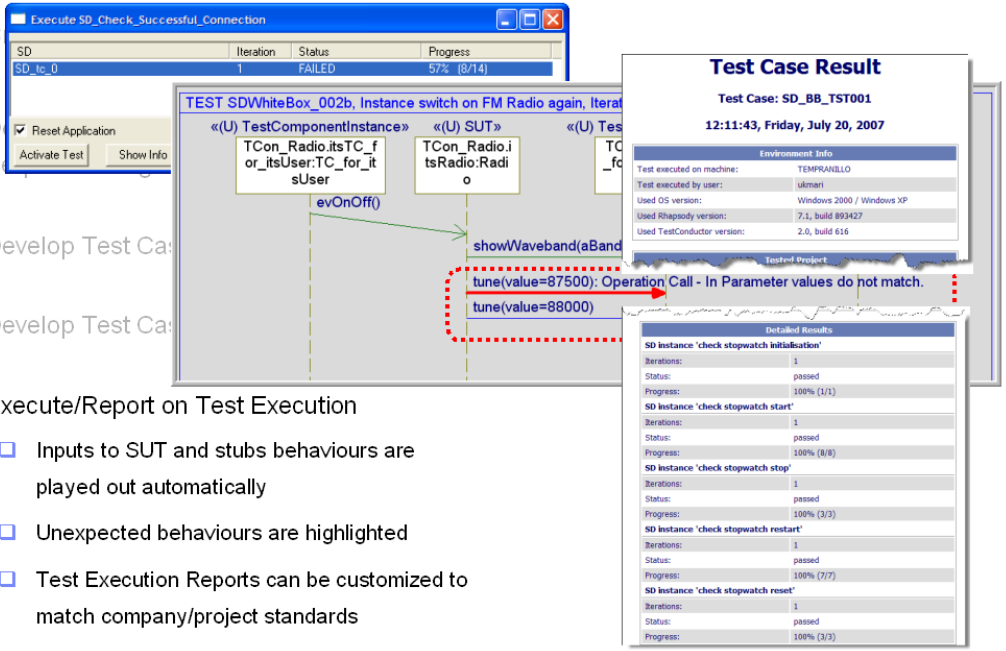
Environment Info	
Test executed on machine:	NBOSC-21-1
Test executed by user:	ubrockmeyer
Used OS version:	Windows 2000 / Windows XP
Used Rhapsody version:	Aries, build 799102
Used TestConductor version:	2.0, build 530

Tested Project	
Project:	CashRegister
Active Component:	TCon_CashRegister_5
Active Configuration:	

Test Context: TCon_C
tc_code
tc_activity_diagram
tc_adding_removing_products
tc_regression_test
atg_tc_008
atg_tc_009
atg_tc_006
atg_tc_002
atg_tc_003
atg_tc_004

Test Execution & Test Reporting

Design & Test Processes Fully Integrated



- C
- D
  Se
- Develop Test Cas
- Develop Test Cas

Execute SD_Check_Successful_Connection

SD	Iteration	Status	Progress
SD_tc_0	1	FAILED	57% (8/14)

☑ Reset Application

TEST SDWhiteBox_002b, Instance switch on FM Radio again, Iterat

«(U) TestComponentInstance»
TCon_Radio.itsTC_for_itsUser:TC_for_itsUser
«(U) SUT»
TCon_Radio.itsRadio:Radio
«(U) Tes
TC_fo
evOnOff()
showWaveband(aBand

tune(value=87500): Operation Call - In Parameter values do not match.

tune(value=88000)

Test Case Result

Test Case: SD_BB_TST001

12:11:43, Friday, July 20, 2007

Environment Info	
Test executed on machine:	TEMPRANILLO
Test executed by user:	ukmari
Used OS version:	Windows 2000 / Windows XP
Used Rhapsody version:	7.1, build 893427
Used TestConductor version:	2.0, build 616

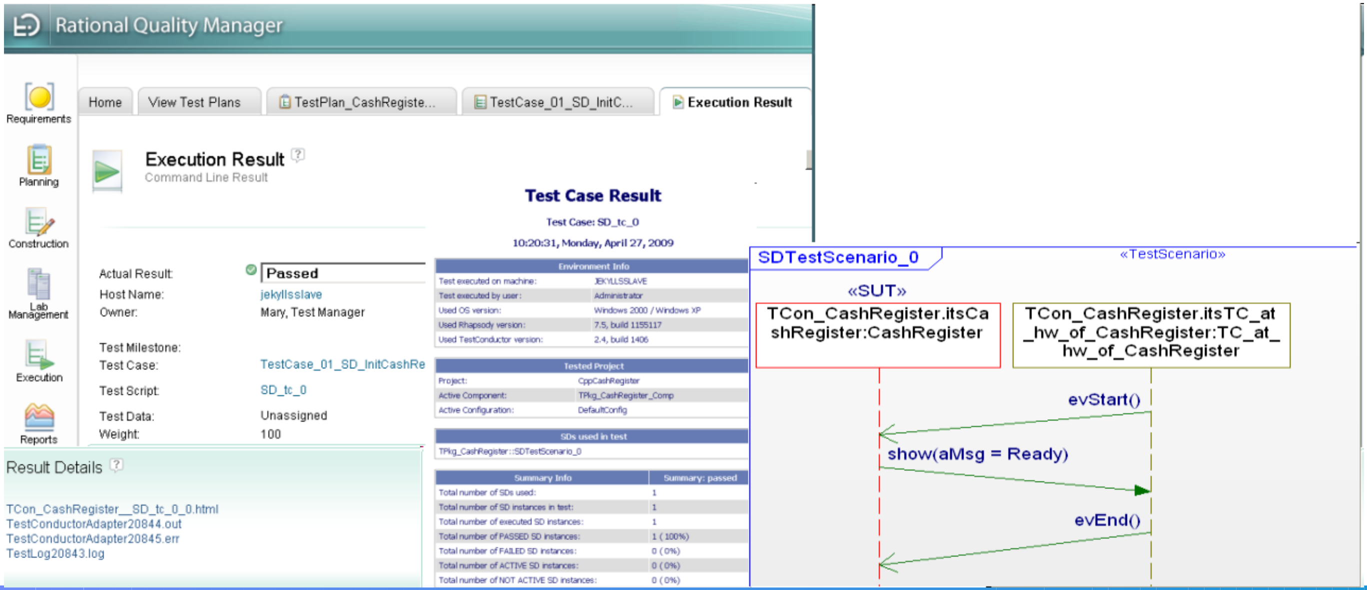
Tested Project

Detailed Results	
SD instance 'check stopwatch initialisation'	
Iterations:	1
Status:	passed
Progress:	100% (1/1)
SD instance 'check stopwatch start'	
Iterations:	1
Status:	passed
Progress:	100% (8/8)
SD instance 'check stopwatch stop'	
Iterations:	1
Status:	passed
Progress:	100% (3/3)
SD instance 'check stopwatch restart'	
Iterations:	1
Status:	passed
Progress:	100% (7/7)
SD instance 'check stopwatch reset'	
Iterations:	1
Status:	passed
Progress:	100% (3/3)

- 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

Rational Rhapsody TestConductor integration with Rational Quality Manager

- Enables full execution control & management of model based Rhapsody TestConductor test cases from RQM
- Execution status (passed/failed) and result reports (Execution Results, Coverage Results) accessible through RQM
- RQM can utilize TestConductor execution results to continuously provide transparent & up to date QA statistics and QA reports



Execution Result
Command Line Result

Actual Result:	Passed
Host Name:	jekyllsslave
Owner:	Mary, Test Manager
Test Milestone:	
Test Case:	TestCase_01_SD_InitCashRe
Test Script:	SD_tc_0
Test Data:	Unassigned
Weight:	100

Result Details

TCon_CashRegister__SD_tc_0_0.html
TestConductorAdapter20844.out
TestConductorAdapter20845.err
TestLog20843.log

Test Case Result

Test Case: SD_tc_0

10:20:31, Monday, April 27, 2009

Environment Info	
Test executed on machine:	JEKYLLSSLAVE
Test executed by user:	Administrator
Used OS version:	Windows 2000 / Windows XP
Used Rhapsody version:	7.5, build 1155117
Used TestConductor version:	2.4, build 1406

Tested Project	
Project:	CppCashRegister
Active Component:	TPkg_CashRegister_Comp
Active Configuration:	DefaultConfig

SDs used in test
TPkg_CashRegister::SDTestScenario_0

Summary Info	Summary: passed
Total number of SDs used:	1
Total number of SD instances in test:	1
Total number of executed SD instances:	1
Total number of PASSED SD instances:	1 (100%)
Total number of FAILED SD instances:	0 (0%)
Total number of ACTIVE SD instances:	0 (0%)
Total number of NOT ACTIVE SD instances:	0 (0%)

www.ibm.com/software/rational