Developing Safe Automotive Electric and Electronic (E/E) Systems: Achieving Compliance with ISO 26262
Agenda

- Trends in Automotive Product Development & Delivery
  - The importance of “safety”
  - Introduction to ISO 26262

- IBM Rational software platform for automotive systems
  - An integrated support ISO 26262
  - Process and collaboration support
  - Requirements Engineering
  - A modelling approach for a safety based development lifecycle
  - Quality Management

- Summary
**Business and Market Drivers**
*Are Leading to Increased Electronic and Embedded (E/E) Software Content*

<table>
<thead>
<tr>
<th>Green</th>
<th>Product Differentiation</th>
<th>Growing Customer Demands</th>
<th>Competition</th>
<th>Legal Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybrid, Electric</td>
<td>Innovation, Brand Image</td>
<td>Comfort, Telematics</td>
<td>Overcapacity, Price Wars</td>
<td>Safety, Environment</td>
</tr>
</tbody>
</table>

*Increased usage of electronics and embedded software; Highly sophisticated in-vehicle electric/electronic (E/E) systems*

*INSTRUMENTED, INTERCONNECTED, INTELLIGENT*
Electric and Electronic (E/E) systems are realizing automotive product innovation.. BUT

- Issues with the complexity of automotive Electrical and Electronic systems
  - Typically 50 + ECUs, 8 Buses of 4 different types, Network controllers
  - Diagram schematics for most vehicles now resemble small modern aircraft with a similar amount of code and its inherent complexity

- Embedded systems are being used to
  - Implement more and more advanced features
  - Improve safety (although this adds complexity and leads to other issues)
Electric and Electronic (E/E) systems are realizing automotive product innovation

- Issues with the complexity of automotive Electrical and Electronic systems

How can you ensure that a possible malfunction will not harm the driver, occupants or road users?

→ You’ll need to take care about „safety“
What is Safety?

Safety in the context of automotive embedded systems is about the prevention, detection, and response to unintended behavior that can lead to harm for the vehicle occupants and other road user.
What is ISO 26262?

ISO 26262 FDIS („Road vehicles – Functional safety“) is a new upcoming ISO standard for safety relevant electronic and electric (E/E) systems in passenger cars up to 3.5 tons.
Drivers for ISO 26262

- Numerous cases of recalls and problems with automotive software (just some examples, many others)
  - **March 2004: Chrysler Pacifica (34,561 vehicles)**
    - Software protocol used to test the vehicle exhaust gas recirculation (EGR) system may lead to engine stalling under certain circumstances, increasing the risk of a crash.
  - **Apr 2004: GM recalls 12,329 Cadillac SRXs**
    - One-second delay in brake activation “The problem, due to a software anomaly, only occurs during the first few seconds of driving when the SUV is moving slowly”
  - **Dec 2004: Hyundai recalls 120,000 Elantras**
    - Airbag software problem detected in Insurance Institute crash tests (driver side airbag didn’t deploy in crash test)
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
Scope of ISO 26262

- ISO 26262 (derived from principles in IEC 61508)
  - Functional Safety for EE Systems in passenger vehicles, i.e. Automotive as opposed to Trucks and Buses
  - Covers the management and technical aspects of the complete safety development lifecycle
  - Takes 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
  - Design safety into the system from the outset
  - ISO 26262 covers the complete development lifecycle from Lust to Dust
What this means for Automotive

- All Automotive System development for Electronic and Electrical components need to comply to ISO 26262
  - Consists of 10 parts

<table>
<thead>
<tr>
<th>DIS 26262</th>
<th>1 Vocabulary</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Management of functional safety</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3 Concept phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Product development on system level</td>
</tr>
<tr>
<td>5 Hardware development</td>
</tr>
<tr>
<td>6 Software development</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7 Production and operation</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>8 Supporting processes</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>9 ASIL-oriented and safety-oriented analysis</th>
</tr>
</thead>
</table>

| 10 Guideline on ISO 26262 (informative) |

- Due to the reuse of a large number of existing elements and pre-existing vehicle architectures the development cycle is seldom a pure top down V-cycle
- Means that the safety-lifecycle must be adaptable and flexible
  - Heavily influenced by the V-model
  - Made more modular
IBM Rational Software Platform for Automotive Systems

provide the means to implement Automotive Best Practices for Functional Safety, complying with ISO 26262

- Focusing on collaborative
  1. Product Portfolio Management
  2. Requirement Engineering
  3. Model Driven Systems Development
  4. Software Development for Systems
  5. Integrated Product Change Management
  6. Quality Management

- Supporting Automotive Standards such as
  - EAST-ADL2, RIF, MISRA, CMMI/SPICE, ISO26262 and AUTOSAR

- Based on JAZZ technology

- Foundation for an end-to-end Engineering Lifecycle Management solution
A look to the inside: How IBM Rational Software Platform for Automotive Systems supports ISO 26262

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

Manage collaborative systems lifecycle management across development teams and engineering disciplines with Automotive data model based on AUTOSAR & ISO26262 process template and compliance
A look to the inside: How IBM Rational Software Platform for Automotive Systems supports ISO 26262

An integrated support ISO 26262

- Rational DOORS
- Rational Rhapsody
- Rational Method Composer
- Rational Team Concert
- Rational Quality Manager
- Rational Test Conductor
- Rational Test RealTime
A look to the inside: How IBM Rational Software Platform for Automotive Systems supports ISO 26262

Process and collaboration support

Rational DOORS

Rational Rhapsody

Rational Method Composer
Rational Team Concert

Rational Quality Manager
Rational Test Conductor
Rational Test RealTime

Innovation for a smarter planet

IBM Software Group | Rational software

© 2011 IBM Corporation
Rational software Process and Collaboration support

- Rational Team Concert is the enabler for controlling process and managing change
  - Process template for ISO 26262
    - Helps with project management
    - Team management
    - Task allocation
  - Integrates with practices that give clear guidance on the application of ISO 26262
  - Configuration management and Collaboration platform
- Integrates with multiple Rational Tools
  - Rational Method Composer (RMC) for process management
  - 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
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

Workitems linked to process guidance

Web based ISO 26262 guidelines and MBSE practices

Guidelines reference ISO 26262

Rational Team Concert

Rational Method Composer
RTC ISO 26262 Process and Practice templates

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

*Clauses 8.12 Qualification of Software components and 8.14 Proven in use argument, not currently supported
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

<table>
<thead>
<tr>
<th>FDIS 26262</th>
<th>1 Vocabulary</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Management of functional safety</td>
<td></td>
</tr>
<tr>
<td>3 Concept phase</td>
<td></td>
</tr>
<tr>
<td>4 Product development on system level</td>
<td></td>
</tr>
<tr>
<td>5 Hardware development</td>
<td></td>
</tr>
<tr>
<td>6 Software development</td>
<td></td>
</tr>
<tr>
<td>7 Production and operation</td>
<td></td>
</tr>
</tbody>
</table>

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
ISO 26262 scalability and modularity

- Process flow is very modular, componentised across
  - Systems, SW or HW or OEM and Subcontractor relationships
- Process template also modular to reflect major tasks or phases
ISO 26262 Process flow

- The modular nature of the time lines means that the flow can be adapted
  - Duplicate the available flows or
  - Further levels of detail can be applied at the iteration level
- The process flow is controlled in the project environment by start and end dates
  - Make the relevant work task the active iteration in a timeline
ISO RTC 26262 Roles

- Only two roles specifically defined
  - Project manager
  - Safety manager
- A number of other roles can be inferred by the nature of the tasks e.g
  - Software engineer
  - Safety engineer
- Roles defined in the template allow project managers to assign them to team members
  - Affects what they do
  - Controlled by permissions (pre set)
- Task and work item assignment can be further controlled using teams
ISO 26262 Teams

- Set of predefined teams that map to the concept phases
  - Need to be populated with team members
- Teams can be linked to project phases
  - Makes it easy for a project manager to allocate the contents of work item template to team
- Team leaders assign work items to team members.
ISO 26262 work item templates

- Work item templates are modularised, it covers
  - Separate safety management section
  - Main concept phases
  - Separation of production and operation activities
  - Aspects of supporting processes
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
A look to the inside: How IBM Rational Software Platform for Automotive Systems supports ISO 26262

Requirements Engineering

Rational DOORS

Rational Rhapsody

Rational Method Composer
Rational Team Concert

Rational Quality Manager
Rational Test Conductor
Rational Test RealTime
Quality begins with Requirements: IBM Requirements Engineering Solution

- 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**
Requirements Management Must Provide Lifecycle Traceability
From Idea through End of Life

- **Traceability is the key to compliance with ISO 26262**
  - Initial requirements will be decomposed, which creates traceability relationships
  - Other relationships can also be traced such as “consists of”, “verifies”, etc.
  - Traceability must be enforced in order to ensure consistency and completeness

- Traceability from customer requirements through product development to test and delivery enables organizations to:
  - Know which requirements are implemented and tested vs. those which are not
  - Manage and defend against scope creep
IBM Rational DOORS
Manage All Requirements Across the Lifecycle and Across Disciplines

- Combined document and spreadsheet views
- Simple, intuitive interfaces for easy adoption
- History and baselines

End-to-end visual validation in a single view
- Input and output from/to various common formats

Solve the right problem because the requirements are visible at all times

Writing Requirements within Context
A look to the inside: How IBM Rational Software Platform for Automotive Systems supports ISO 26262

A modelling approach for a safety based development lifecycle
Modern Approaches for Describing Systems Are Evolving
To Better Manage Complexity and Reduce Time-to-market

Past

- Specifications
- Interface requirements
- System design
- Analysis & trade-off
- Test plans

Future

Moving from Document-centric to Model-centric
What is Model Driven Systems Development (MDSD)?

A structured approach for the development of complex systems across the mechanical, electronic and software disciplines

- Ensures that all requirements are fulfilled
- Employs models as the primary artifacts throughout systems development
- Facilitates improved communication among all stakeholders
- Provides a disciplined way to manage complexity through abstraction
- Improves quality through integration of testing with development
- Allows specification and development of software that controls the system and enables its use
Allowing abstraction, hierarchies and modularization with domain-focused, standards-based languages

- **SysML – Systems Modeling**
  Language for modeling high-level
  - vehicle functions
  - logical and technical architecture
  - vehicle and E/E system behavior

- **UML – Unified Modeling Language**
  for modeling
  - ECU and SW architecture
  - Client-specific profiles

- **AUTOSAR**
  - Detailed E/E System and ECU HW and SW architecture
Maximize your budget & boost productivity with effective Systems Engineering

**IBM Rational Rhapsody® software Family**

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

**Model Driven Testing:**
- Bring the benefits of abstraction and automation to testing
- Deliver products that meet customer expectations faster, cheaper

**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

**Requirements Driven Testing:**
- Reduce overall dev costs by dramatically reducing time in the testing phase
- Automatic regression testing, Change impact and analysis

**Sequence Diagrams**

**Finding & Correcting Errors**

**Host based**

**Target based**

**Automated unit testing**
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
- Also work going on with KVI
  - Medini tool
  - Safety analysis
  - Integrate with Rhapsody and RTC
Auto-generation of safety-relevant summary data

Fault Source Matrix, Fault Detection Matrix, Fault-Requirement Matrix, Hazard Analysis…

- Traceability improves your ability to make your safety case
- Safety metadata guides downstream engineering work
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

---

<table>
<thead>
<tr>
<th>Requirement</th>
<th>ASIL</th>
<th>Identification of obstructions</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collision prevention</td>
<td>SafetyGoal</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>Identification of obstructions</td>
<td>SafetyGoal</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>Cruise Control Deactivation</td>
<td>SafetyGoal</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>Ensure detection of Radar failure</td>
<td>SafetyRequirement</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>RADAR Signal failure</td>
<td>SafetyRequirement</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>Loss of signals from Brake and ECM</td>
<td>SafetyRequirement</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>Computed acceleration</td>
<td>SafetyRequirement</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>Speed sensor failure</td>
<td>SafetyRequirement</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>Driver warnings</td>
<td>SafetyRequirement</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>Driver signals to ACC</td>
<td>SafetyRequirement</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>Reaction to bad signals</td>
<td>SafetyRequirement</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>ACC switch Off</td>
<td>SafetyRequirement</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>Brake sensor readings</td>
<td>SafetyRequirement</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>Brake failure</td>
<td>SafetyRequirement</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
<tr>
<td>ACC switch</td>
<td>SafetyRequirement</td>
<td>Identification of obstructions</td>
<td>Ensure that the vehicle stops in time before a collision</td>
</tr>
</tbody>
</table>
Benefit from Modeling with IBM Rational Rhapsody

- Extend requirements engineering to development
  - “Traceability and more … “
- Use models for Safety Analysis
- Focus on analysis and design
  - From “system” down to “software”
- Execute models
  - Analyze behavior
  - Find errors early

- Develop highly-optimized embedded C/C++/Java software
  - Model and Code are kept in sync
- Collaborate visually
  - Integrated in the IBM Rational Software Platform for Automotive Systems
Benefit from Modeling with IBM Rational Rhapsody

- Manage Complexity
  - across automotive product development and test
  - collaborative between OEM and supplier
  - aligning the different disciplines, mechanics, electric, electronic and embedded software

- Improve Quality
  - finding errors more early in the development lifecycle

- Increase Re-use
  - Re-use E/E architectural elements – from vehicle product lines down to embedded software
A look to the inside: How IBM Rational Software Platform for Automotive Systems supports ISO 26262

Quality Management

Rational DOORS

Rational Rhapsody

Rational Method Composer
Rational Team Concert

Rational Quality Manager
Rational Test Conductor
Rational Test RealTime
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

Test Execution & Test Reporting

Design & Test Processes Fully Integrated

- 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
A look to the inside: How IBM Rational Software Platform for Automotive Systems supports ISO 26262

An integrated support ISO 26262

- Rational DOORS
- Rational Rhapsody
- Rational Method Composer
- Rational Team Concert
- Rational Quality Manager
- Rational Test Conductor
- Rational Test RealTime
Integrate Safety Design into Design from the beginning

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

Requirements Analysis:
- Functional and Non-Functional Requirements
- Safety Requirements
- Business and Regulatory Requirements

System and Software Design:
- Structural
- Behavioral
- Temporal
Recap: How the elements of our platform relate to ISO 26262

- Requirements (DOORS)
  - 26262 Standard should be placed into a requirements database
  - Drive activities as well as support traceability and verification

- Systems Modeling, Simulation, and Auto-Code Generation (Rhapsody)
  - SysML modeling provides ability to architect overall system – mechanical and E/E and then to execute to verify model

- Configuration and Change Management (Rational Team Concert)
  - Configuration management of E/E In development (baseline and other revisions), as well as configuration management for different option combinations
  - Change Management for control of ECRs to E/E

- Process (Rational Team Concert and Method Composer)
  - 26262 is very process based
  - Non-prescriptive: “what to do” as opposed to “how to do”

- Verification and verification planning (Test Conductor and Rational Quality Manager)
  - Lot of emphasis on validation and verification of Systems, HW and SW
  - Level and type of test dependent upon ASIL of element to be developed.
IBM Rational Software Platform for Automotive Systems

Key Results of an E/E engineering lifecycle management

Key Results

- Improved collaboration along the design chain and the entire lifecycle
- Increased productivity due to comprehensive electrical, electronics and software asset management
- Lower cost of safety and regulatory compliance

ISO FDIS 26262

Effective product development processes can increase productivity by 40% and reduce defects by 75%

Automotive SPICE©
Thank You

www.ibm.com/software/rational

© Copyright IBM Corporation 2011. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM’s sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.