IBM Rhapsody & Rhapsody Design Manager
for Systems and Software Engineering

April 2016
Agenda

- Engineering challenges, and the benefits of modeling
  - Rhapsody overview
  - Collaboration across the development lifecycle
  - Summary
Complexity creates development challenges

Leading to cost overruns, schedule slips, and quality issues

Poor quality requirements = failed projects

Poor architectures lead to technical debt ($$$)

Incorrect interfaces are a major cause of cost overruns

System performance suffers if functionality is poorly distributed across components

Hardware/software integration is often late

Reinventing engineering artifacts is inefficient
General benefits of modeling

- We model to:
  - Manage complexity through the use of abstraction and separation of concerns
  - Enhance communication
    - Common language promotes common understanding across engineering disciplines
    - Visual approach to understanding and clarifying requirements
    - Clear and consistent behavioral and interface specifications
  - Reduce risk
    - Model execution leads to correctness and reduces uncertainty and risk
    - Models provide traceability and document what you have done
Agenda

- Engineering challenges, and the benefits of modeling
  - **Rhapsody overview**
  - Collaboration across the development lifecycle
  - Summary
IBM Rhapsody – Model Based Systems Engineering (1)

- Collaborative design, development, and test environment
- For **Systems Engineers**:  
  - Model complex systems in SysML, AUTOSAR
IBM Rhapsody – Model Based Systems Engineering (2)

- Collaborative design, development, and test environment
- For Systems Engineers:
  - Model complex systems in SysML, AUTOSAR, etc.
  - Model-based functional analysis to improve quality of requirements
IBM Rhapsody – Model Based Systems Engineering (3)

- Collaborative design, development, and test environment

- For **Systems Engineers**:
  - Model complex systems in SysML, AUTOSAR, etc.
  - Model-based functional analysis to improve quality of requirements
  - Verification requirements and design early with model execution
IBM Rhapsody – Model Based Systems Engineering (4)

- Collaborative design, development, and test environment
- For **Systems Engineers**:  
  - Model complex systems in SysML, AUTOSAR, etc.  
  - Model-based functional analysis to improve quality of requirements  
  - Verification with model execution and model-based test specifications
IBM Rhapsody – Model Based Systems Engineering (5)

- Collaborative design, development, and test environment
- For Systems Engineers:
  - Model complex systems in SysML, AUTOSAR, etc.
  - Model-based functional analysis to improve quality of requirements
  - Verification requirements and design early with model execution
  - Hybrid simulation across multiple engineering domains using the Functional Mockup Interface (FMI) standard
IBM Rhapsody – Software Development

- Collaborative design, development, and test environment
- For Software Engineers:
  - Using UML models to create working and self documented source code
  - Code reuse and/or full code generation in C, C++, Java, Ada
  - Debug source code graphically through model execution
IBM Rhapsody – Model Driven Testing

- Define test cases with sequence diagrams, statecharts, flowcharts or even code
  - OMG UML Testing Profile

- Automate testing tasks
  - Create Test Architecture
  - Execute and monitor tests

- Definition and management of regression tests

- Reporting of results, coverage and traceability
Overview of Safety Critical features in Rhapsody

- Available safety kits include guidance on how to capably develop safety-related software with IBM Rhapsody by meeting the tool qualification objectives described in the safety-related standards.

- Available with ISO 26262 (automotive), IEC 61508 (electrical and/or electronic), IEC 62304 (medical), EN 50128 (railway), DO-178B/C (airborne systems)
Requirements traceability into generated code

- Satisfied requirements are included as comments in the generated code

```java
switch (rootState_active) {
// State Off
// Realizes requirement REQ_SYSSW1:
// Sample shall have an Off state
    case Off:
        // Realizes requirement REQ_SYSSW2:
        // Sample shall be turned on
        if(IS_EVENT_TYPE_OFF(evOn_Default_id))
        {
            ...
        }
        break:
// State On
    case On:
        {
            ...
        }
        break:
    default:
        // Realizes requirement SwitchStatementDefaultClause #SCR1.36:
        // The default clause shall be generated for a switch statement
        break;
}
return res;
```
Agenda

- Engineering challenges, and the benefits of modeling
- Rhapsody overview
- **Collaboration across the development lifecycle**
- Summary
An open, integrated systems approach delivers:

- Market Assessment
- System Verification and Validation
- System Test
- System Requirements
- System Analysis & Design
- Deployment/Release to Mfg.
- System Verification and Validation
- Operations and Maintenance
- Customer Requirements
- Integration and Validation
- Decomposition and Definition
- Access to all engineering information
- Implementation
- Mechanical Design
- Embedded Software Development
- Electrical/Electronics Design
- Collaboration between disciplines

Traceability throughout the lifecycle
Core IBM applications running on the Jazz platform
IBM Rhapsody with DOORS Next Generation

- RM – Requirements Management (aka DOORS Next Generation)
- DM – Design Management (aka Rhapsody Design Manager)
Easy access to model information by stakeholders

- Share, collaborate, and manage models across the lifecycle

- Rhapsody Design Manager is the Rhapsody web client

- A Rhapsody desktop client is used for editing models
UML, SysML, AUTOSAR, Simulink models in Design Manager web client

- View design over web
- Collaborate with stakeholders with commenting
- Mark-up diagrams to elaborate comments
- Browse design information
Design Manager - Collaborative development in Rhapsody client

- View design comments
- Create or view reviews
- Search across design projects
- View details of design review
Design Manager - Traceability to requirements

- Links to requirements can be created in Rhapsody client or web client
- Links can be viewed in Rhapsody or the RM tool (DOORS or DNG)
- Model elements become “suspect” when linked requirements changed
Real-time Planning - integrated with requirements, design, test

- **Free** with Design Manager
- **Rich Hovers** provides insight at your finger tips
- **Plans** provide status of workloads
- **Dashboards** provide real-time metrics and trends

29: Customers can nominate an organization for the program

- Tested By Test Case
  - 6: Organization must identify how much money is desired
  - 7: Allocate dividends by amount and frequency
  - 10: Customers can Nominate an Organization
  - 9: Donors Can Choose to Support an Organization
Design Manager – simplified model management

- Rhapsody models can be stored in a traditional SCM system, and imported periodically into Design Manager. Preferred by software teams.

- Alternatively, models can be **actively managed** by Design Manager and stored in the Jazz repository, eliminating the need for conventional SCM.
  - Multi user model edits are easy! Preferred by Systems Engineers.

Actively Managed mode: Design Manager is managing the model
Summary

- Electronic products are getting smarter, which usually means “more complex”
- A visual approach simplifies the specification and development of complicated systems and their software
  - *With many companies, it has become a necessity*
- Higher quality requirements result from model based functional analysis
- Rhapsody Design Manager web client broadens the accessibility of models, supports design reviews, enables lifecycle traceability, and optionally manages the model
- Rhapsody Developer and Designer provide free access to Design Manager
  - Design Manager provides free access to Jazz planning
- Rhapsody generates MISRA compliant code that can be safety certified
- An integrated workflow can be applied using IBM DOORS Next Generation and IBM Rhapsody
IBM Publishing Engine (RPE)

- Save days. Automate the generation of documentation, from multiple sources.
What is ISO 26262?

- **ISO 26262** ("Road vehicles – Functional safety") is current ISO standard for safety relevant electronic and electric (E/E) systems in passenger cars up to 3.5 tons.

- 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.
Rhapsody Design Manager

- **Rhapsody Architect for Systems Engineers** – core product for Systems Engineers providing requirements visualization, UML, SysML, AUTOSAR authoring, FMI, static analysis, parametric constraint evaluation

- **Rhapsody Designer for Systems Engineers** – includes Rhapsody Architect for SE and adds model execution for requirements validation and trade-off analysis. Supports parametric constraint evaluation, rapid prototyping and model based testing.  
  >> *Includes Rhapsody Design Manager* <<

- **Rhapsody Architect for Software** – core product for embedded software developers working in C, C++, Java. Provides UML, SysML, FMI, DDS, multicore support, structural software development, static analysis, code visualization, reverse engineering, requirements traceability

  >> *Includes Rhapsody Design Manager* <<


  - **Rhapsody Developer for Ada** - version of Rhapsody Developer supporting Ada only.

- **Rhapsody Design Manager** – Web client. Uses Jazz technology to help teams develop, share, search, review and collaborate on their SysML, UML, AUTOSAR, DoDAF, and Simulink models. Optionally provides the model repository for easy multi-user development.