Bruce Powel Douglass
Mats Gothe

IBM Rational Workbench for Systems and Software Engineering

A smarter planet starts with the innovation of smarter products. Engineering teams driving this innovation need new, collaborative ways to develop and deliver the right demands on time, on budget, and with the right quality. IBM® Rational® Workbench for Systems and Software Engineering (Workbench) software provides integrated and collaborative capabilities for such systems delivery teams.

Systems delivery is often divided into multiple categories, for example, systems engineering, project management, software engineering, and quality management. These categories interconnect across the systems delivery life cycle, performing distinct tasks and creating distinct life cycle work products. The IBM Rational Workbench for Systems and Software Engineering software is a life cycle management solution that supports such collaborative tasks and linking of the life cycle work products. The Workbench also enacts the systems delivery workflows and provides task management capabilities to effectively run the systems delivery project.

This IBM Redpapers™ publication is intended for systems delivery teams who require an integrated solution for systems life cycle management for systems and software engineering.

An integrated solution for systems life cycle management

Without integration across the systems delivery life cycle, systems and software teams are left to operate in silos. When silos form, product delivery effectiveness suffers. In order to deliver smarter products that respond to changing market needs, it is necessary to allow systems and software engineering teams to work efficiently and to manage all the life cycle work products through collaboration. The IBM Rational Workbench for Systems and Software Engineering software, shown in Figure 1, provides this integrated systems life cycle management solution.

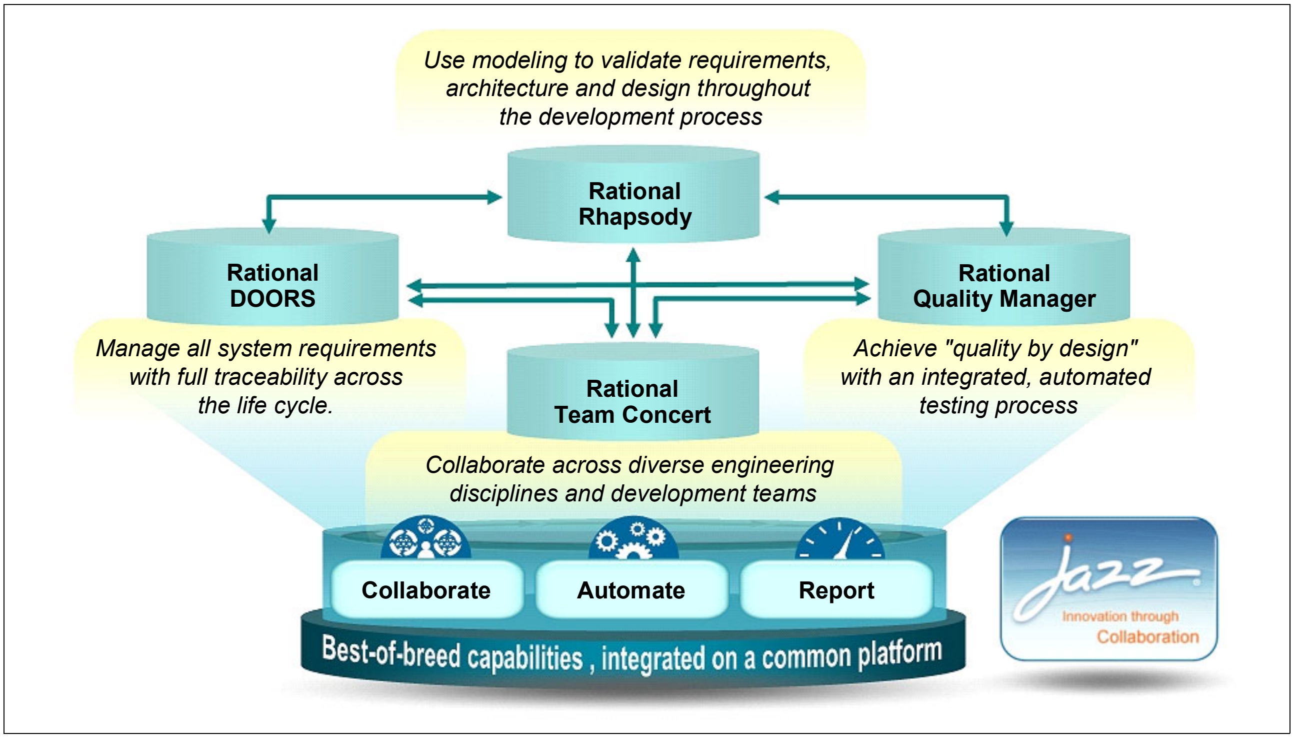

Use modeling to validate requirements, architecture and design throughout the development process
Rational Rhapsody
Rational DOORS
Rational Quality Manager
Manage all system requirements with full traceability across the life cycle.
Rational Team Concert
Achieve "quality by design" with an integrated, automated testing process
Collaborate across diverse engineering disciplines and development teams
Collaborate
Automate
Report
Best-of-breed capabilities , integrated on a common platform
Jazz
Innovation through Collaboration

Figure 1 The Rational Workbench for Systems and Software Engineering software

The IBM Rational Workbench for Systems and Software Engineering software provides an integrated solution for the following systems and software engineering team roles:

- Systems engineers

  The Workbench provides an integrated and collaborative environment for requirements analysis, architecture management, and work, change and configuration management for teams of systems engineers. The leading products are the IBM Rational DOORS® software and the IBM Rational Rhapsody® software for systems engineering tasks, integrated with the IBM Rational Team Concert™ software for life cycle management of the work products. The integration with the IBM Rational Quality Manager software provides for strong collaboration with system validation teams from the start of the project.

- Project, development, and test team leads

  The IBM Rational Team Concert software and the IBM Rational Quality Manager software provide work and plan management for systems delivery teams across the project life cycle and enable live transparency through collaboration, automation, and reporting on the systems delivery work products and project health.

- Software engineers

  The Workbench, with the IBM Rational Rhapsody software integrated with the IBM Rational Team Concert software in the Eclipse IDE, provides a software development solution for software engineers. This integrates model-driven development with the capabilities of the IBM Rational Team Concert software for team collaboration, like model configuration management, work items, change sets, and continuous software build support. The Workbench also provides traceability to upstream systems engineering work products in the IBM Rational DOORS software and the IBM Rational Rhapsody software, or downstream traceability to systems integration and validation.

- Software and system testers

  The IBM Rational Quality Manager software provides a collaborative environment for test planning, construction, and execution, supporting continuous testing as part of the software engineering teams, as well as test management of system validation and acceptance testing. The IBM Rational Test Lab Manager software improves the efficiency of system test labs and manages how test resources are requested and provided.

Integration enables collaboration

The Workbench is integrated on the IBM Rational Jazz™ platform. Uniquely attuned to global and distributed teams, the Jazz platform transforms systems and software delivery by making it more collaborative, productive, and transparent. You can think of the Jazz platform as an extensible framework that dynamically integrates and synchronizes people, processes, and assets associated with development projects. Unlike the monolithic, closed solutions of the past, the Jazz platform is an open platform that supports the Open Services for Lifecycle Collaboration (OSLC) initiative for improving tool interoperability. Products built on the Jazz platform can use a rich set of capabilities for team-based systems and software delivery.

The Workbench is integrated on the Jazz platform, providing collaboration, automation, and reporting for the systems delivery life cycle, as shown in Figure 2.

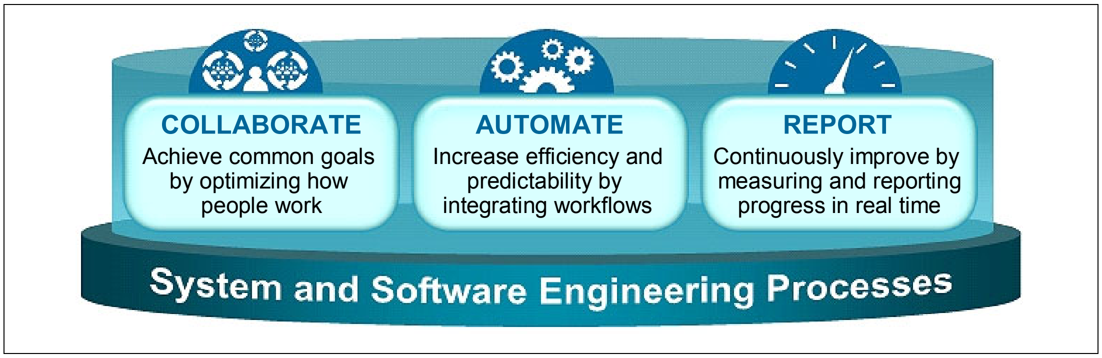

COLLABORATE
Achieve common goals by optimizing how people work
AUTOMATE
Increase efficiency and predictability by integrating workflows
REPORT
Continuously improve by measuring and reporting progress in real time
System and Software Engineering Processes

Figure 2   Jazz platform integration capabilities enable collaboration, automation, and reporting

Rather than providing a “one-size fits no one” solution, the Jazz Integration Architecture follows the models and standards of the World Wide Web, where life cycle work products are loosely coupled through the linking of resources, thereby making the repository boundaries of systems delivery disappear.

Therefore, the architectural principles of the Jazz platform represent a key departure from traditional repository approaches taken in the past. Data integration is achieved via linked artifacts across repositories using RESTful interfaces. Browsing may be performed using either web interfaces or rich clients. These approaches will allow organizations to seamlessly integrate life cycle work products, processes, and teams.

As a key part of the Jazz platform, OSLC is an industry initiative to enable interoperability of tools and resources across disciplines, life cycle solutions, and vendors. The ability to provide and consume OSLC specifications enables the Workbench to integrate openly across a system life cycle management solution. It also allows the flexibility for customers to use and integrate tools from multiple vendors to capture and manage data across the system life cycle management solution. This ability is of significant value when establishing openness and collaboration across projects delivering systems of systems, where smarter products integrate with IT-based operations or management applications.

Systems development workflows

The systems development space consists of multiple integrated activities: systems engineering, project planning, software development, and quality management. System engineering is responsible for clarification of requirements and architecture, but is not focused on electronics or mechanical or software components *per se*. Software engineering takes the specifications from system engineering and implements the software that is embedded in devices or smart products.

Software engineering also implements systems of systems with operational software solutions that interface with and manage smart products. Planning capabilities help define the project schedule and ensure that the teams are working on the right tasks. These capabilities also provide control and visibility of the project status, progress, and dependencies on external component providers. Quality management provides continuous validation of software integration, test planning, traceability, tracking, and metrics reporting on system verification and acceptance.

Let us look at the tasks within these systems development activities and how the IBM Rational Workbench for Systems and Software Engineering software supports these workflows.

System engineering workflows

System engineering has several primary tasks to perform in the systems development process:

- Stakeholder requirements definition

  Definition of the stakeholder needs as a set of requirements, which is clustered into use cases.

- Requirements analysis

  Elaboration of the use case requirements.

- System functional analysis

  Detailed black box analysis of the requirements, including the construction of high-fidelity executable requirements models.

- System design synthesis

  Identification of a system architecture, including the responsibilities of each subsystem, allocation of requirements to each, and the definition of the subsystem interfaces.

The overall workflow for systems engineering is shown in Figure 3.

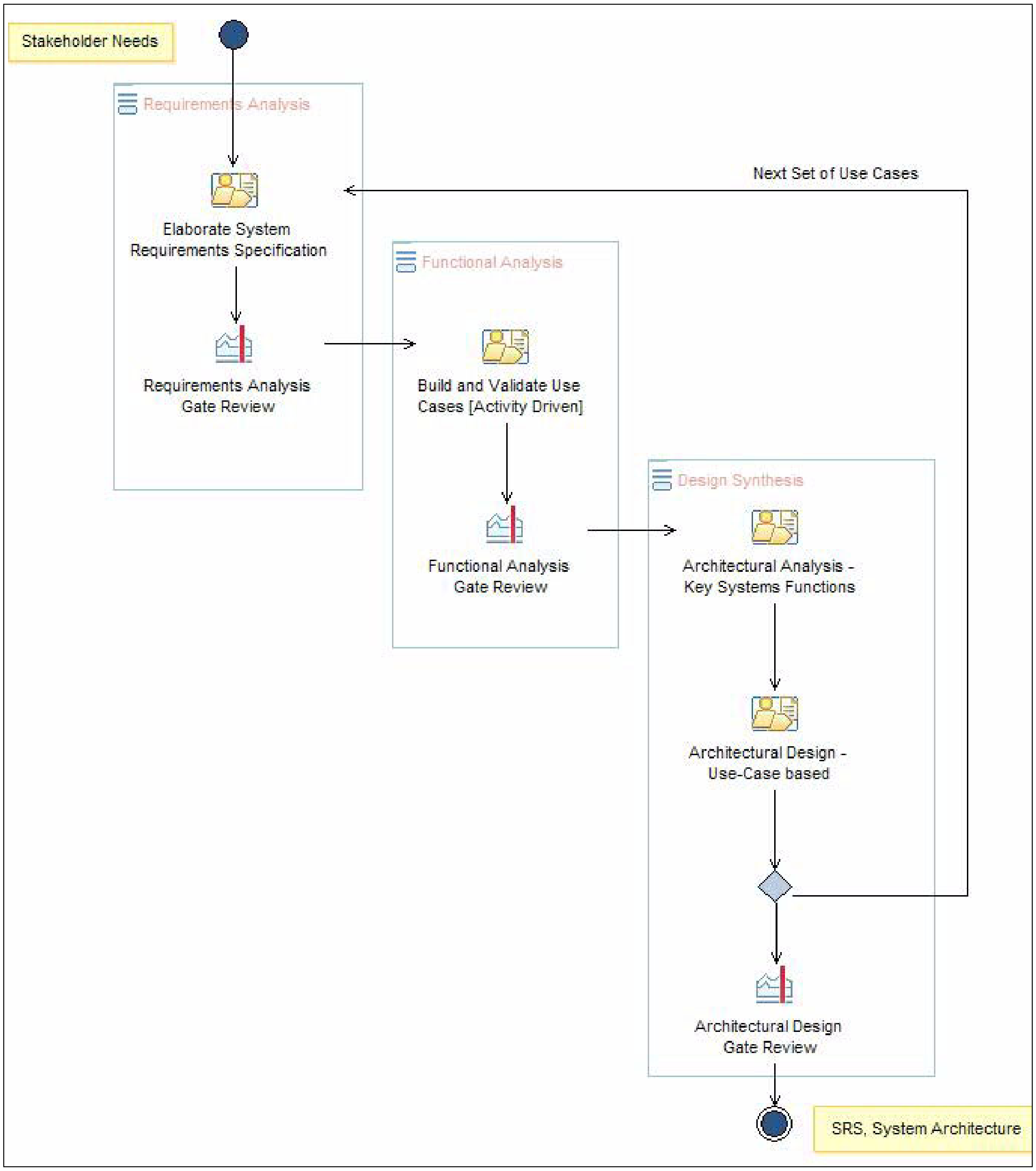
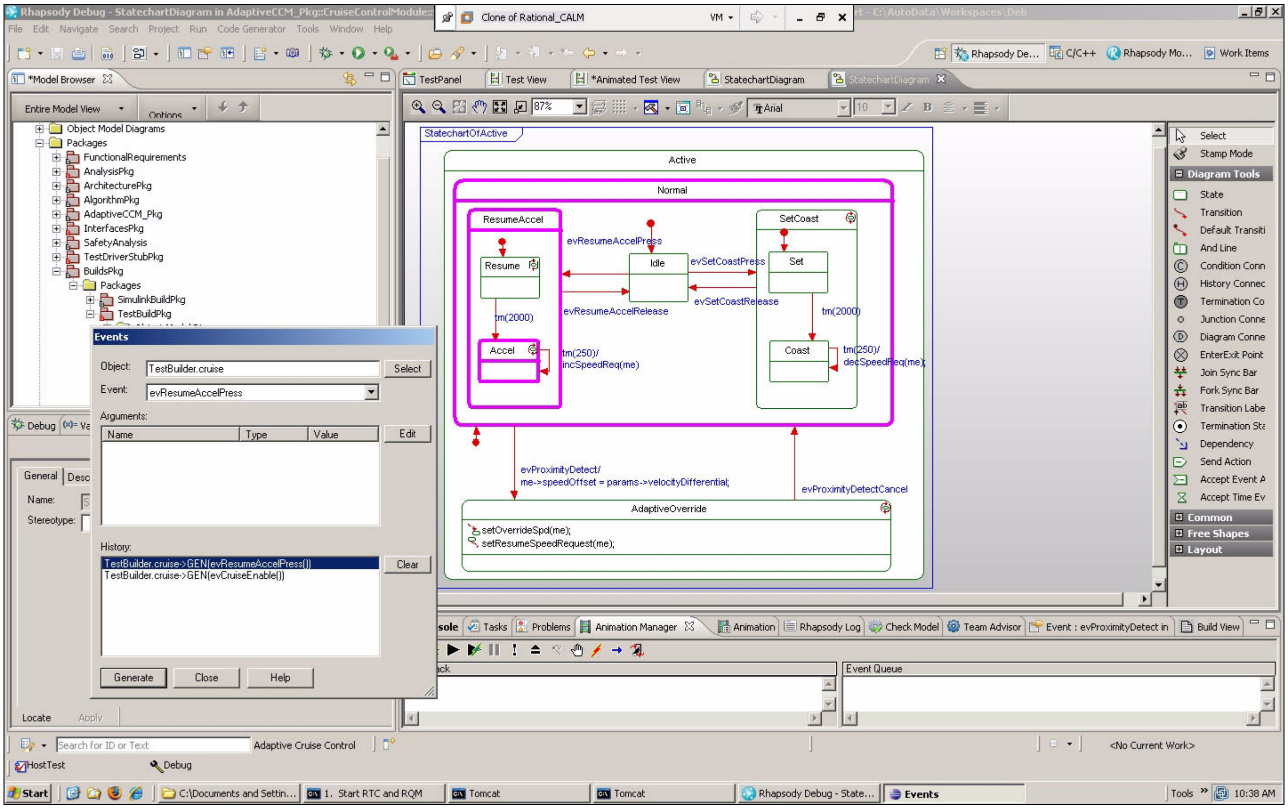

Stakeholder Needs
Requirements Analysis
Elaborate System Requirements Specification
Requirements Analysis Gate Review
Next Set of Use Cases
Functional Analysis
Build and Validate Use Cases [Activity Driven]
Functional Analysis Gate Review
Design Synthesis
Architectural Analysis - Key Systems Functions
Architectural Design - Use-Case based
Architectural Design Gate Review
SRS, System Architecture

Figure 3   The system engineering workflow

The first two activities within systems engineering identify and manage requirements. Stakeholder requirements are statements of stakeholder need. In requirements analysis, system requirements (statements of system behavior and performance) that meet the stakeholder needs are identified. These requirements are, for the most part, textual in nature and consumable by all relevant stakeholders. Even though textual requirements frequently suffer from imprecision and ambiguity, these steps are nonetheless crucial in virtually all system development organizations. The Workbench includes the IBM Rational DOORS software to support identification and management of stakeholder and system requirements.

The next systems engineering activity, system functional analysis, moves from textual requirements to a more precise, unambiguous language to analyze functional and quality of service (performance) requirements, such as the Systems Modeling Language (SysML). System functional analysis builds an elaborated model of each system use case that can be validated through simulation. This allows systems engineers to perform early validation of the correctness of system behavior, such as outputs for various inputs. It also allows for collaborative interactions with stakeholders about requirements that might not be apparent from the textual system requirements.

The Workbench includes the IBM Rational Rhapsody software, shown in Figure 4, which supports linking and modeling of service requirements to system requirements in the IBM Rational DOORS software. The collaborative capabilities of the solution increase the efficiency and quality of systems delivery; for example, in addition to building the executable use case models, the system engineers also analyze the safety and reliability of the system, and update both the textual system requirements and the verification plan.



Events

Object: TestBuilder.cruise

Event: evResumeAccelPress

History:

```
TestBuilder.cruise->GEN(evResumeAccelPress())
TestBuilder.cruise->GEN(evCruiseEnable())
```

Figure 4 Using the IBM Rational Rhapsody software to analyze and validate system requirements

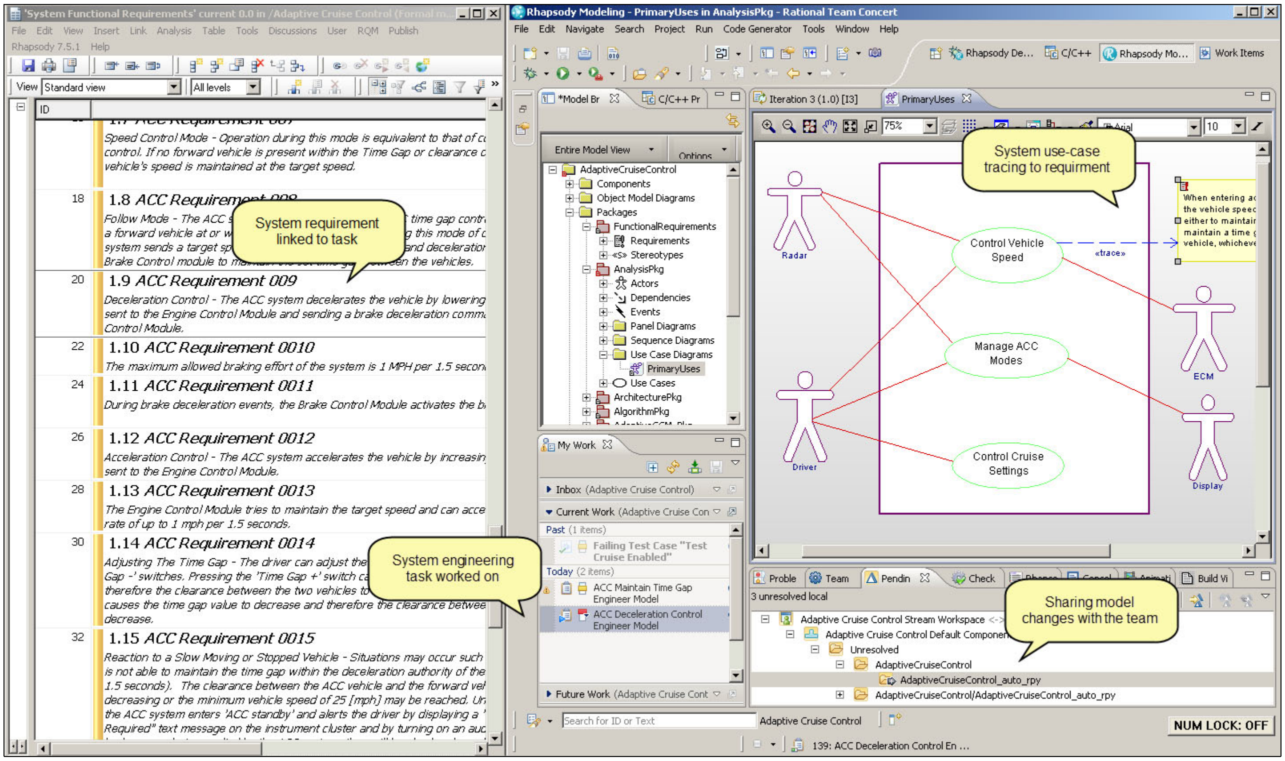
During design synthesis, the systems engineering teams perform the following key tasks:

- A set of trade studies to make the architectural selection.
- Identification of the architectural subsystems and allocation of system requirements to the subsystems.
- Updating the safety and reliability analysis to take into account the system architecture.
- Identification of subsystem interfaces and state-based behavior.
- Creation of subsystem-level requirements.
- Developing the system control laws and allocating parts of them to subsystems.
- Updating system verification test plans.

Managing system and stakeholder requirements

The IBM Rational DOORS software is the leading requirements management application that provides a comprehensive requirements management capability to the IBM Rational Workbench for Systems and Software Engineering software. It manages stakeholder requirements, system requirements, and decomposed subsystem requirements that engage stakeholders, system engineers, and software engineers and testers in a collaborative requirements process.

Together with the IBM Rational Rhapsody software and the IBM Rational Team Concert software, the IBM Rational DOORS software supports the system engineering teams’s task to capture and link system requirements to stakeholder requirements. When integrated with the IBM Rational Rhapsody software, the system requirements are linked to elaboration and specification of use case requirements and executable requirements models, as shown in Figure 5.



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

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 m … 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

Rhapsody Modeling - PrimaryUses in AnalysisPkg - Rational Team Concert

When entering a
the vehicle speed
either to maintain
maintain a time g
vehicle, whicheve

System requirement linked to task
System engineering task worked on
System use-case tracing to requirment
Sharing model changes with the team
File Edit View Insert Link Analysis Table Tools Discussions User RQM Publish
Rhapsody 7.5.1 Help
View Standard view
All levels
ID
File Edit Navigate Search Project Run Code Generator Tools Window Help
Rhapsody De...
C/C++
Rhapsody Mo...
Work Items
*Model Br
C/C++ Pr
Entire Model View
Options
AdaptiveCruiseControl
Components
Object Model Diagrams
Packages
FunctionalRequirements
Requirements
«S» Stereotypes
AnalysisPkg
Actors
Dependencies
Events
Panel Diagrams
Sequence Diagrams
Use Case Diagrams
PrimaryUses
Use Cases
ArchitecturePkg
AlgorithmPkg
Iteration 3 (1.0) [I3]
PrimaryUses
75%
Arial
10
Radar
Driver
Control Vehicle Speed
Manage ACC Modes
Control Cruise Settings
«trace»
ECM
Display
My Work
Inbox (Adaptive Cruise Control)
Current Work (Adaptive Cruise Con
Past (1 items)
Failing Test Case "Test Cruise Enabled"
Today (2 items)
ACC Maintain Time Gap Engineer Model
ACC Deceleration Control Engineer Model
Future Work (Adaptive Cruise Cont
Search for ID or Text
Adaptive Cruise Control
Proble
Team
Pendin
Check
Build Vi
3 unresolved local
Adaptive Cruise Control Stream Workspace
Adaptive Cruise Control Default Componen
Unresolved
AdaptiveCruiseControl
AdaptiveCruiseControl_auto_rpy
AdaptiveCruiseControl/AdaptiveCruiseControl_auto_rpy
NUM LOCK: OFF
139: ACC Deceleration Control En ...

Figure 5 The Workbench provides access for systems engineers to tasks and work products

The IBM Rational Team Concert (RTC) software coordinates the system engineering tasks and workflows, which are all governed by your team’s associated process. Requirements from the IBM Rational DOORS software can be linked to work items in the RTC tool. This linkage enables developers or stakeholders to review the requirements linked to a work item. The work item type might be a requirements change request, an implementation task, a defect, or another standard or custom work item type.

System requirements are the basis for a systems delivery project and managing change to those requirements is critical. The IBM Rational DOORS software provides requirements change management capabilities with RTC. This workflow integrates stakeholders, project leads, and systems engineers. Requested changes to requirements can be reviewed against current requirements and delivery plans, inspected for impact, and a consensus reached through the reviews process in RTC before the changes are applied to the IBM Rational DOORS software.

Project planning and tracking workflows

At the end of a systems engineering phase, a set of detailed subsystem specifications exist that form the basis of project planning and downstream engineering. These specifications are still multidisciplinary. Project planning supports the allocation of requirements as plan items into iteration plans and association to validation test cases, as shown in Figure 6. Plan items are decomposed into engineering team plans and work dependencies between teams.

The IBM Rational Team Concert software, included in the Workbench, is a collaborative team environment that combines project planning, workflow enactment, work item management, and transparent project health and reporting to system teams.

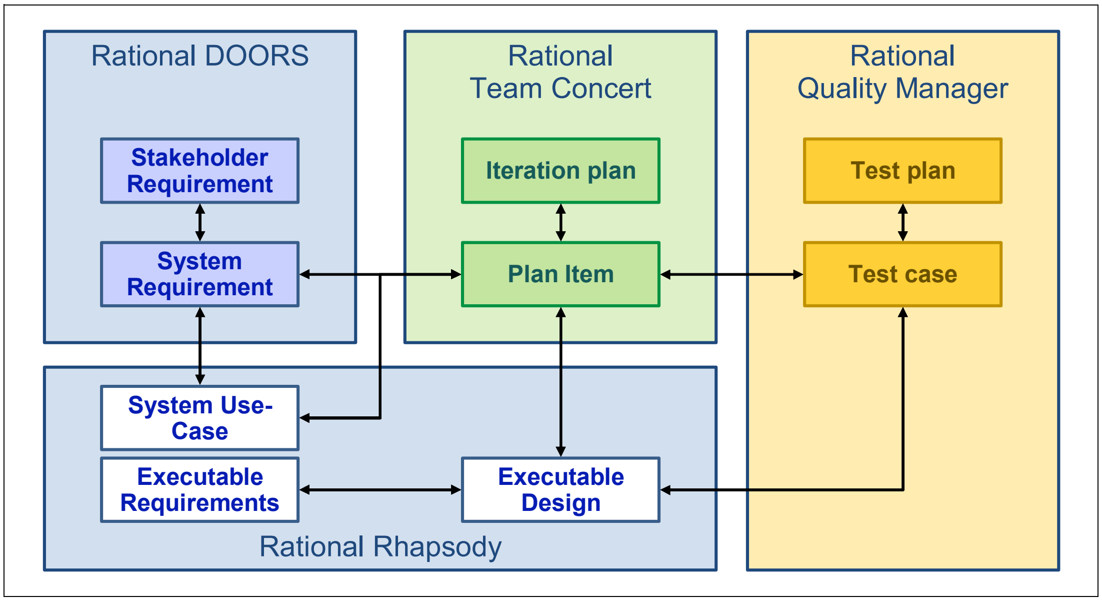
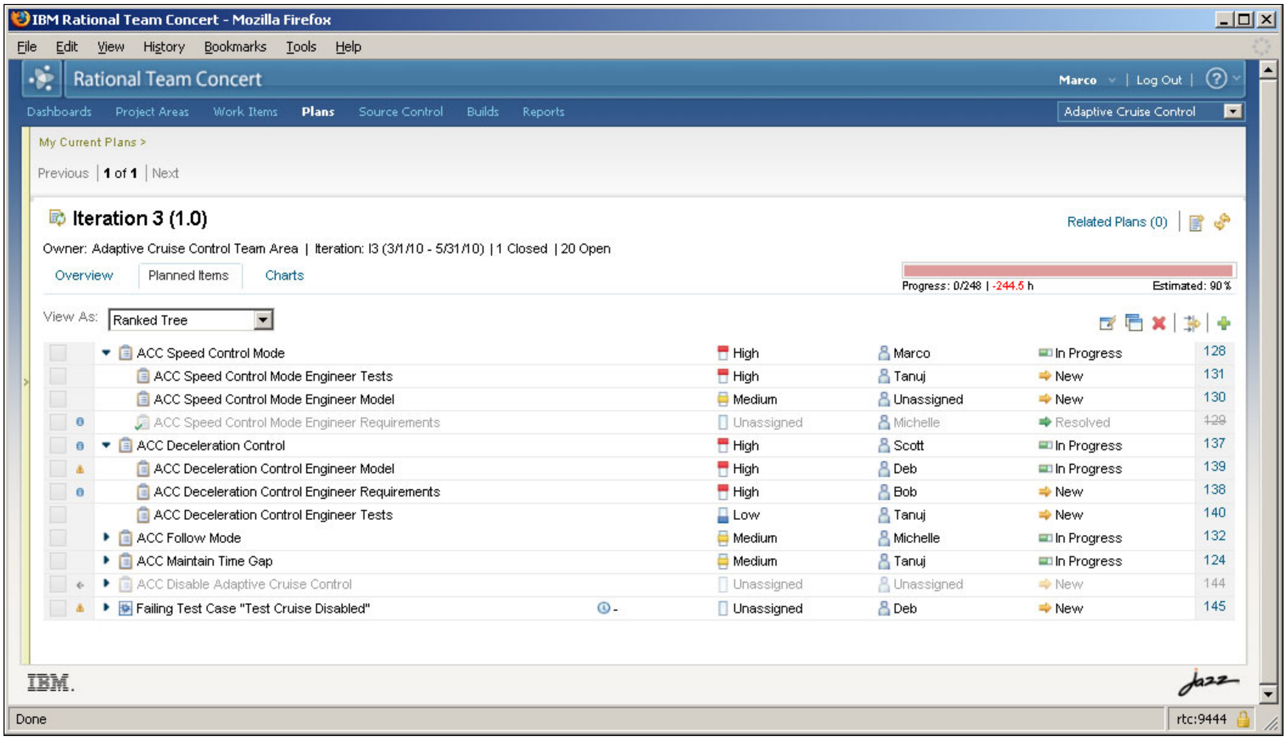


Figure 6 Plan items and relationship to systems delivery work products

When using the Rational Team Concert software, the system and software engineering teams are:

- Managing their task assignments as part of the milestone plans.
- Using task boards to overview individual task assignments and to plan work activities.
- Using systems requirements and system use cases to drive the content of project release and iteration plans.
- Aligning the system delivery project plan, engineering team plans, and system test plans.
- Tracking project health through transparency and collaboration.

Figure 7 shows the iteration planning capability in the IBM Rational Team Concert software. The plan view in the figure shows all work items for the iteration as a ranked hierarchical list.


IBM Rational Team Concert - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Rational Team Concert
Marco | Log Out |
Dashboards Project Areas Work Items Plans Source Control Builds Reports
Adaptive Cruise Control
My Current Plans >
Previous | 1 of 1 | Next
Related Plans (0)
Overview Planned Items Charts
View As: Ranked Tree
IBM.
Jazz
Done
rtc:9444

Iteration 3 (1.0)

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

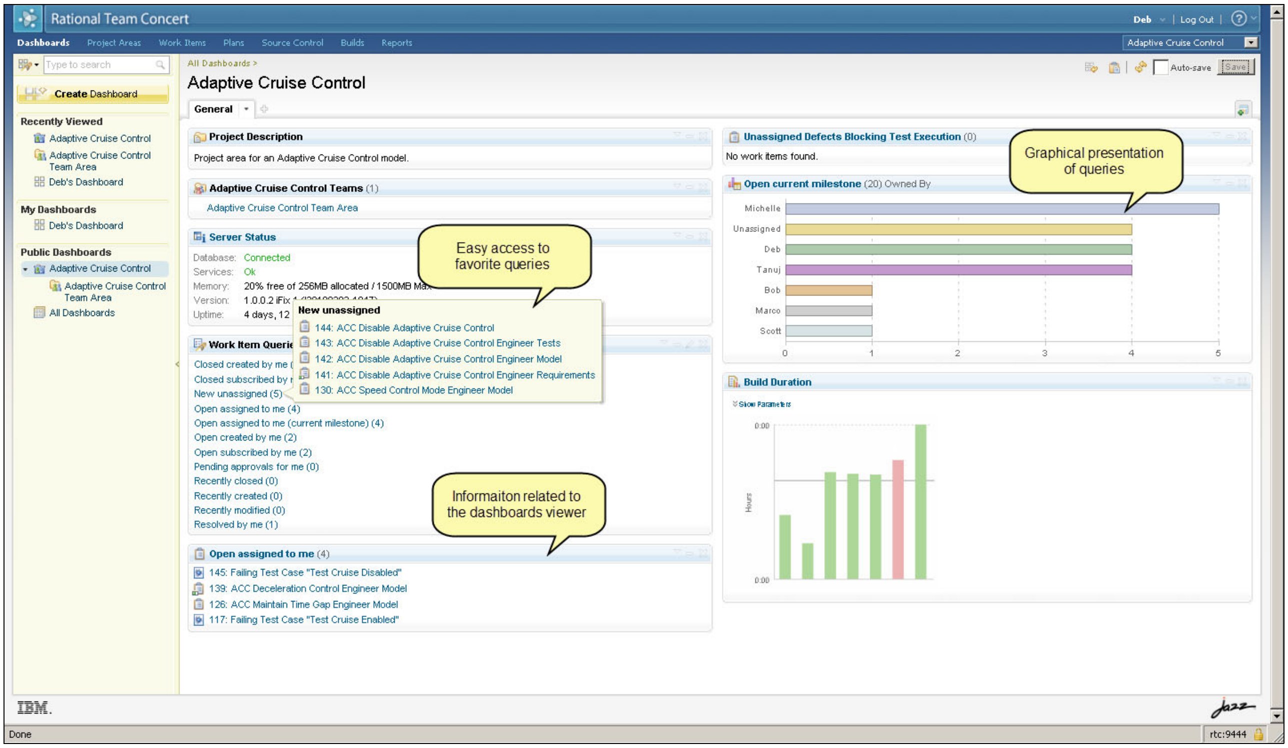
Progress: 0/248 | -244.5 h Estimated: 90 %

ACC Speed Control Mode	High	Marco	In Progress	128
ACC Speed Control Mode Engineer Tests	High	Tanuj	New	131
ACC Speed Control Mode Engineer Model	Medium	Unassigned	New	130
ACC Speed Control Mode Engineer Requirements	Unassigned	Michelle	Resolved	~~129~~
ACC Deceleration Control	High	Scott	In Progress	137
ACC Deceleration Control Engineer Model	High	Deb	In Progress	139
ACC Deceleration Control Engineer Requirements	High	Bob	New	138
ACC Deceleration Control Engineer Tests	Low	Tanuj	New	140
ACC Follow Mode	Medium	Michelle	In Progress	132
ACC Maintain Time Gap	Medium	Tanuj	In Progress	124
ACC Disable Adaptive Cruise Control	Unassigned	Unassigned	New	144
Failing Test Case "Test Cruise Disabled" -	Unassigned	Deb	New	145

Figure 7 Iteration plan in the IBM Rational Team Concert software with plan items

Project and team plans are the live view of the work items assigned to the teams. As the teams progress with the delivery of the work, the plans are continuously updated and provide transparency with regard to team status and project health, which encourages and increases collaboration among teams. With change management integrated into work items, the team members manage all the work, such as tasks, change requests, and defects, in the same live plan.

Project dashboards in the IBM Rational Team Concert software and the IBM Rational Quality Manager software further improve project tracking through transparency and reporting of the team's status and the project's health. Dashboards are configured as shared dashboards for all project members or for individual use by any user. Figure 8 shows an example of a project tracking dashboard. Dashboards present live project and plan information in tabular or graphical form using viewlets, queries, or event feeds. Viewlets tracking cross repository information, using the rich semantics of the life cycle work product traceability links, further improve the project tracking capabilities.



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 Querie

Closed created by me (
Closed subscribed by r
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

Figure 8   Project dashboards improve project tracking through transparency and reporting

The IBM Rational Team Concert software is deployed centrally and connects globally dispersed systems delivery teams using web clients and Internet standards to limit the impact from global development challenges and boundaries.

Software engineering workflows

The IBM Rational Workbench for Systems and Software Engineering software provides integrated workflows for software engineering teams practicing an incremental development process, such as the *IBM Rational Harmony for Embedded RealTime* process, which encompasses the concerns of the development and implementation of high-reliability and safety-critical software. In such processes, every iteration usually lasts between four and six weeks, and develops the software for one or a small number of use cases. Figure 9 shows the workflow for a single iteration.

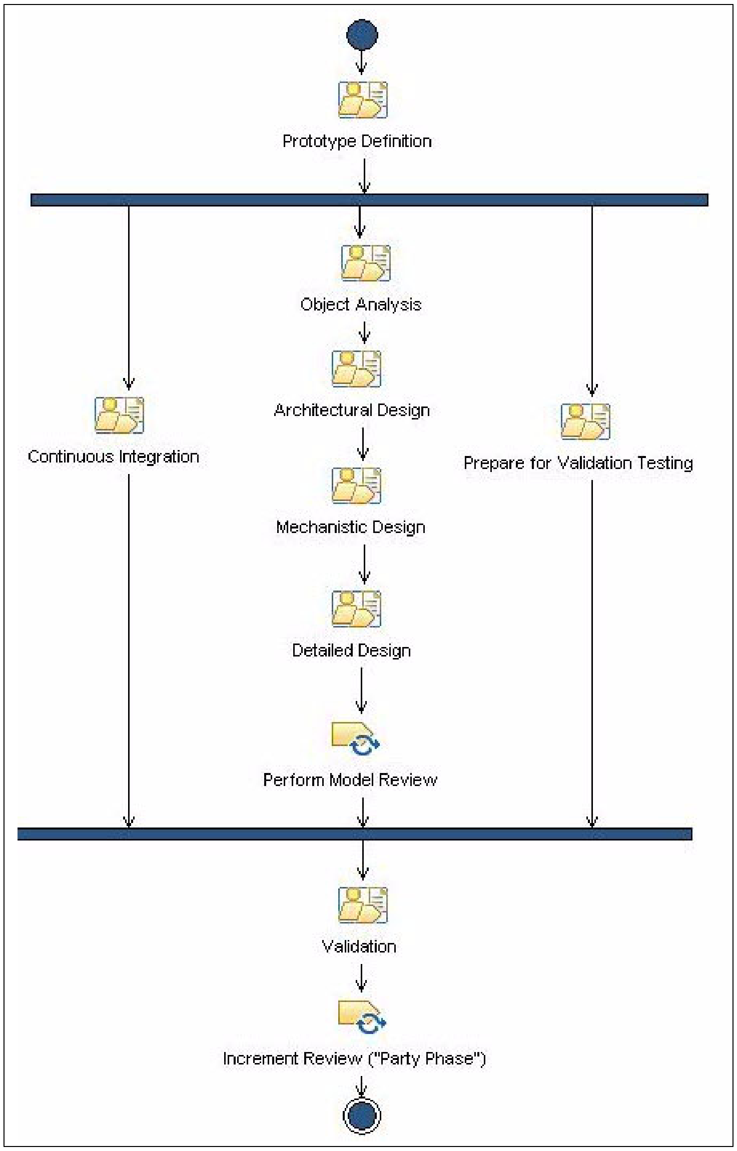


Figure 9 IBM Workbench for Systems and Software Engineering

You can see that every iteration consists of a sequence of a number of key activities, including:

- Prototype definition (plan iteration)

  This activity defines the iteration work in terms of an iteration plan of work items in the IBM Rational Team Concert software. Some work items are derived as plan items from the use cases. Other work items are managed as lists of defects and enhancements, or captured as risk items. The iteration plan elaborates on the software requirements and use cases to be implemented in the iteration.

- Object analysis

  This activity constructs a functionally correct model in the IBM Rational Rhapsody software and a source code baseline for realizing the use cases implemented within this iteration.

- Architectural design

  The architectural design activity adds in zero or more architectural decisions, clustered primarily around the five key architectural views (component, concurrency, safety and reliability, distribution, and deployment) and relevant secondary views. This activity is done primarily through the application of architectural design patterns modeled in the IBM Rational Rhapsody software.

- Mechanistic design

  This activity focuses on the optimization of the collaborations that realize each use case. This is mostly done through the application of mechanistic design patterns.

- Detailed design

  This activity involves the detailed design that optimize the smallest implementation units, that is, class, object, function, and data structure. Teams adopting a UML/SysML based model-driven development practice, or teams developing the application in, for example, C, C++, or Java™, will use the IBM Rational Rhapsody software and the IBM Rational Team Concert software, and the integrated IDE in Eclipse, to design, build, and validate the implementation.

- Model review

  This activity reviews software analysis and design content to ensure consistency with architectural intent, adherence to the organizational quality standards, and to disseminate knowledge about the software structure and behavior. Teams using the IBM Rational Team Concert software can configure the development workflows to enforce model reviews as part of their delivery process.

- Continuous build and integration

  Continuous build is done in parallel with most of the design activities in the iteration. Builds integrate the source code baseline to ensure its consistency and freedom from defects. The team build component in the IBM Rational Team Concert software implements the team's build process and provides build awareness, control, and traceability to the software engineering team and the software test team.

- Preparation for validation

  The Workbench integrates development and test activities and enables software teams to start early with preparing for validation. These planning activities are done in the IBM Rational Quality Manager software in parallel with the development activities and create test fixtures, test plans, and test cases.

- Validation

  Validation tests the software at a black box level to ensure that it properly implements the requirements; this includes both the newly added functionality and regression tests to ensure that the software that worked in previous iterations is not broken. The integration of the IBM Rational Quality Manager software and the IBM Rational Rhapsody software enables test teams to reuse and run model test cases.

- Increment review

  The increment review (also known as a retrospective) compares actual against planned progress, evaluates the efficiency and effectiveness of the work, and updates plans as necessary and appropriate to reflect the reality of the situation. The IBM Rational Team Concert software provides work items specifically tailored for retrospectives.

We will look at three of the software engineering activities in more detail:

- Prototype definition
- Object analysis
- Architectural design

Prototype definition (plan iteration)

Prototype definition plans out the details of the coming iteration in terms of the schedule, requirements (organized as use cases), and test milestones (organized as test cases). Figure 10 shows the overall workflow. If the software development activity is preceded by system engineering, then system requirements are already done, but there is usually still some derivation of software-specific requirements and allocation to the software component structure to be done. Additionally, the effect of the allocation on the safety and reliability of the system must also be examined. This is all done within the prototype definition workflow.

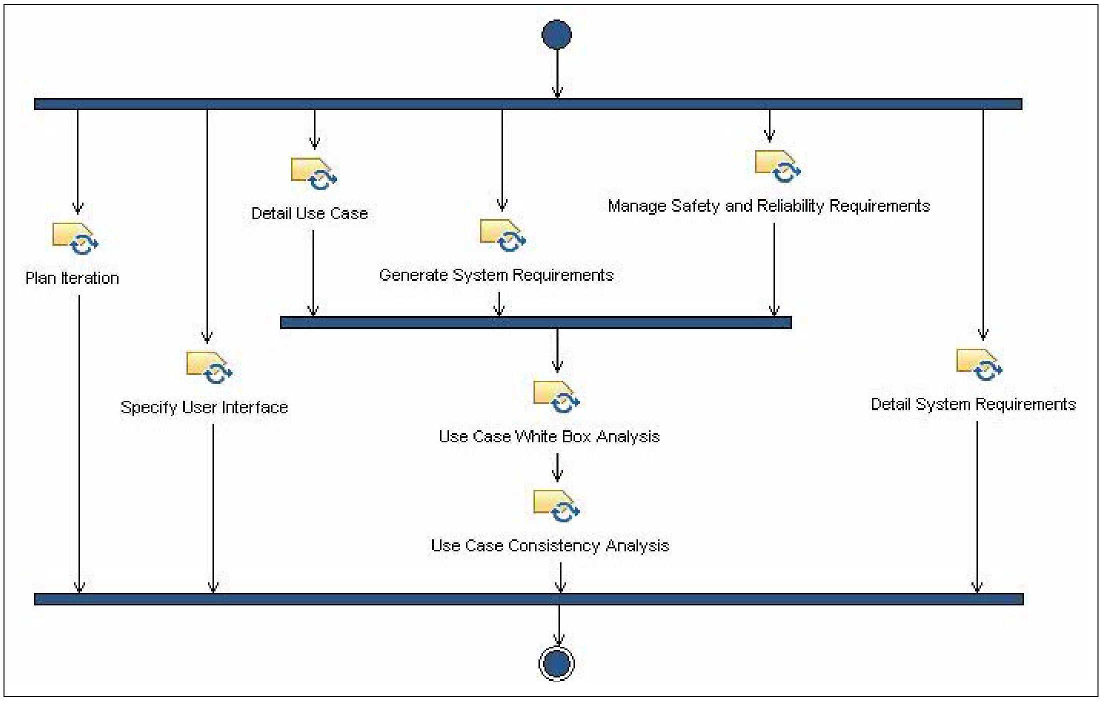
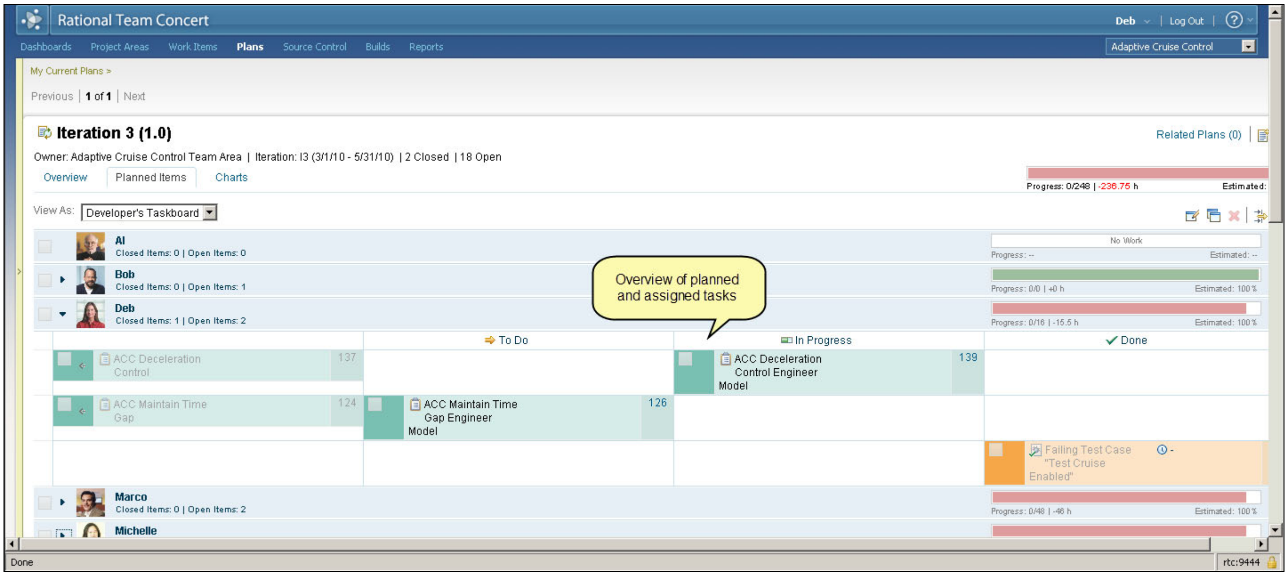

Detail Use Case
Manage Safety and Reliability Requirements
Plan Iteration
Generate System Requirements
Specify User Interface
Detail System Requirements
Use Case White Box Analysis
Use Case Consistency Analysis

Figure 10 Prototype definition activity

The IBM Rational Team Concert software contains a powerful planning environment that enables team leads to assign work, in context of requirements and use cases, balance the plan from a work estimate, and track the process from the live work items. Figure 11 shows the Developer's Taskboard in the IBM Rational Team Concert software.



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 | -46 h Estimated: 100 %

Michelle

Figure 11 Developer's Taskboard in the IBM Rational Team Concert software

Object analysis

Figure 12 shows the workflow for object analysis. In this workflow and the design workflows, high quality source code is incrementally developed. The workflow encompasses the notion of test driven development, the practice of creating developer (unit) tests, and applying them at the same time as the software is developed. The figure also indicates how periodically the developer's work is released to the configuration builder for daily (or more frequent) integration and test, a practice known as *continuous integration*.

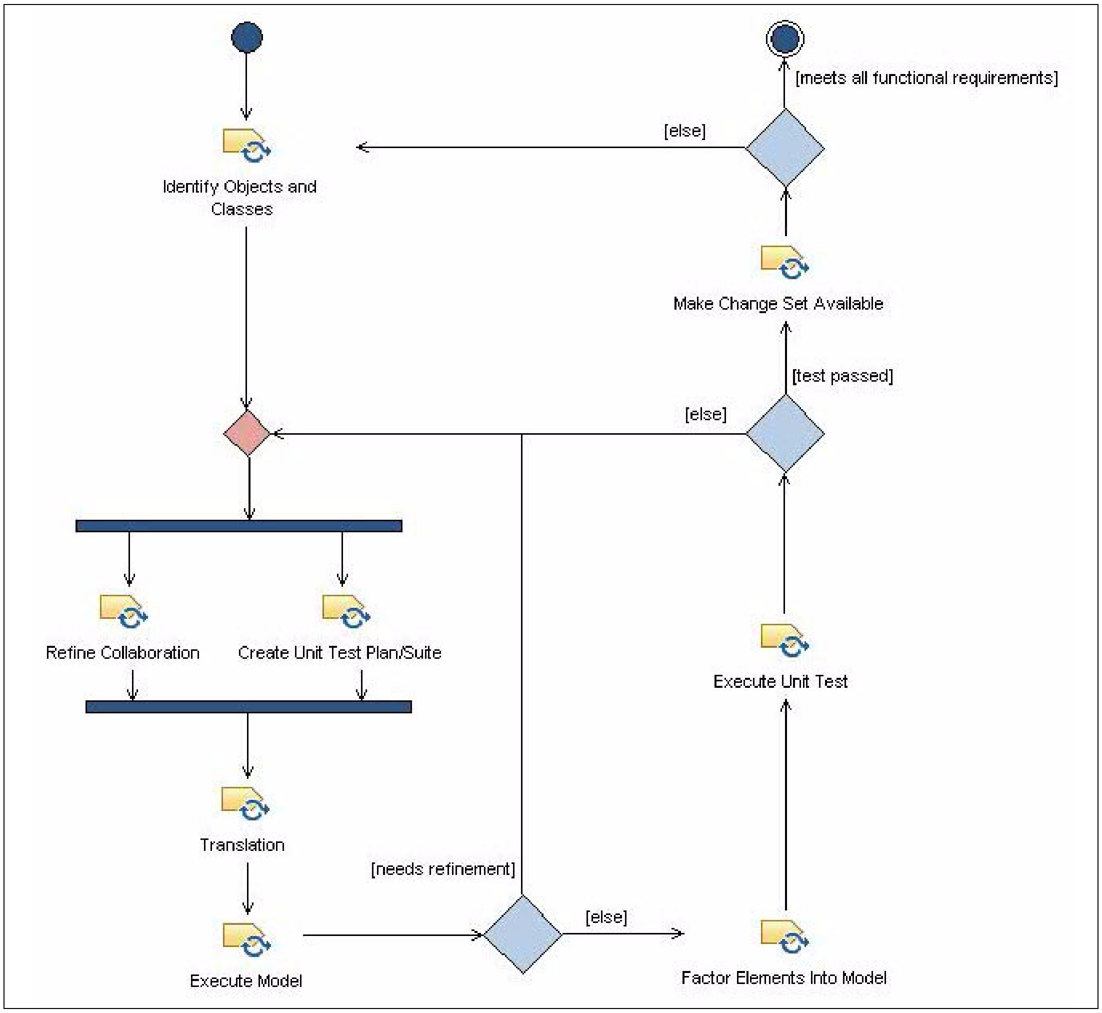

[meets all functional requirements]
[else]
Identify Objects and Classes
Make Change Set Available
[test passed]
[else]
Refine Collaboration
Create Unit Test Plan/Suite
Execute Unit Test
Translation
[needs refinement]
[else]
Execute Model
Factor Elements Into Model

Figure 12 Object analysis activity

Systems delivery processes have strong notions of architecture, based on five key views of architecture. These views include:

- Subsystem and component architecture

  This view of architecture focuses on the large scale units of system organization and their responsibilities and interfaces.

- Concurrency and resource architecture

  This view of architecture identifies the concurrency units and allocates the functional software elements within those concurrency units. This view also identifies how tasks are scheduled and synchronized, and how they share resources.

- Distribution architecture

  The distribution view of architecture identifies how software elements are distributed across processing units (such as CPUs and CPU cores), and the policies and patterns used for them to collaborate to achieve system goals.

- Safety and reliability architecture

  This view of architecture defines how the system identifies, isolates, and handles faults at runtime to ensure safe and reliable operation.

- Deployment architecture

  The deployment view focuses on the distribution of responsibility across engineering disciplines, such as software, electronics, and mechanical.

Architectural design

In the IBM Rational Workbench for Systems and Software Engineering software, the architectural decisions are added to the architectural design activity supported by the IBM Rational Rhapsody software. However, not all architectural decisions are made in every iteration. You might, for example, add the subsystem and component and deployment architectures in iteration 1, the safety and reliability architecture in iteration 3, the distribution in iteration 6, and the concurrency architecture in iteration 7. The overall workflow for software architectural design is shown in Figure 13.

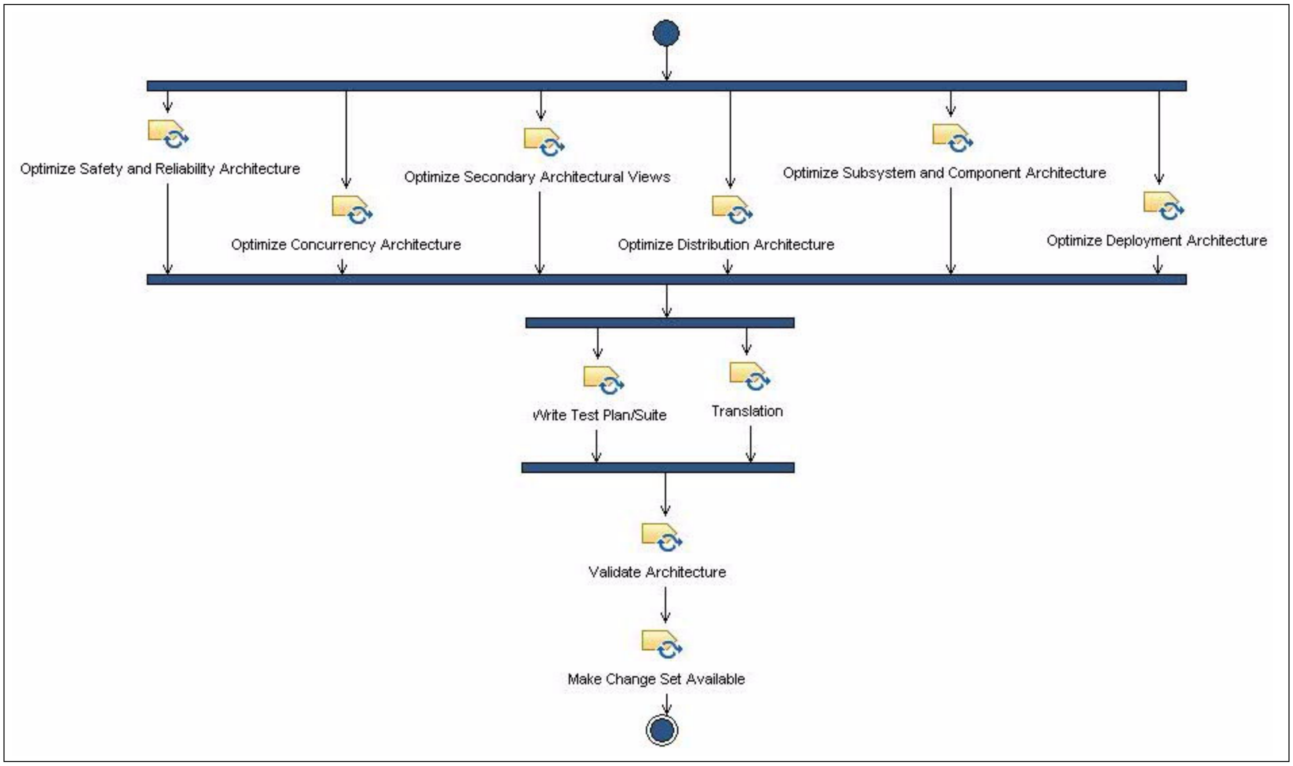
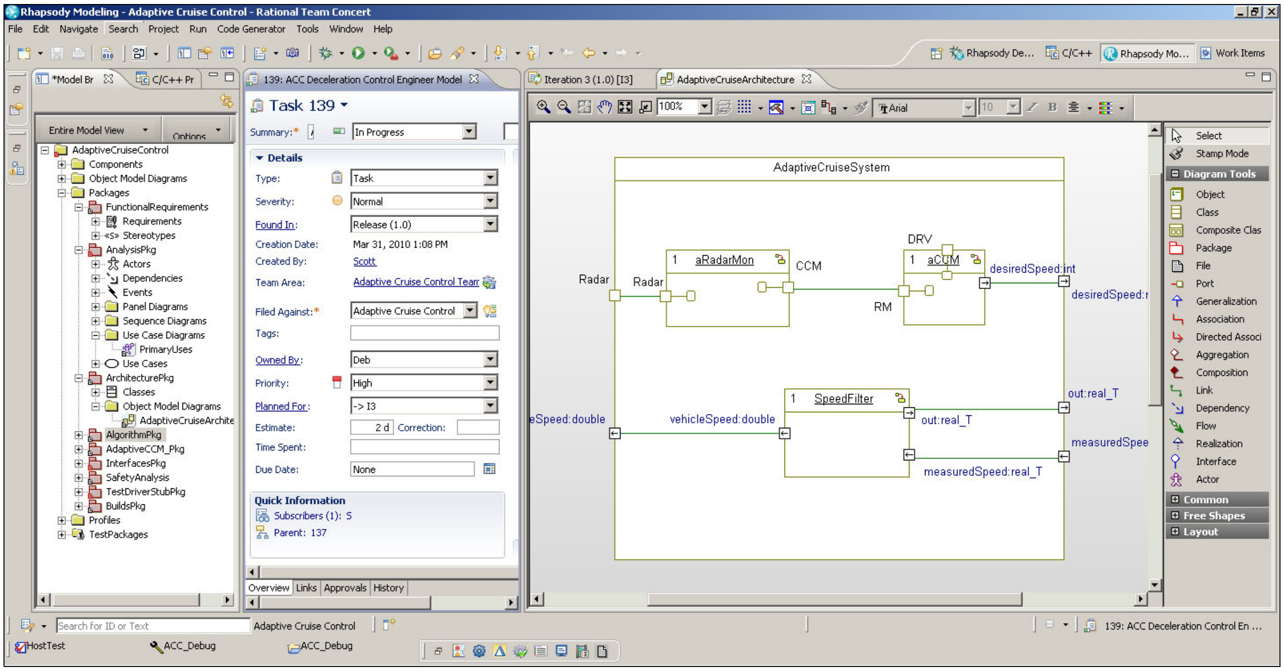

Optimize Safety and Reliability Architecture
Optimize Concurrency Architecture
Optimize Secondary Architectural Views
Optimize Distribution Architecture
Optimize Subsystem and Component Architecture
Optimize Deployment Architecture
Write Test Plan/Suite
Translation
Validate Architecture
Make Change Set Available

Figure 13 Software architectural design views

The IBM Rational Rhapsody software provides the capabilities for modeling system and software requirements, as well as developing and delivering the software components. It provides an easy to use yet rigorous environment for the creation, management, and execution of both SysML and UML models. Integrating with the IBM Rational Rhapsody software, the system requirements are linked to elaboration and specification of use case requirements and executable requirements models, as illustrated in Figure 14.



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 Tear

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

Overview | Links | Approvals | History

Adaptive Cruise Control

Figure 14 Software engineering with the IBM Rational Rhapsody software

The IBM Rational Rhapsody software also interfaces with several tools for various purposes that extend the Workbench and deepen the engineering capabilities. These capabilities include:

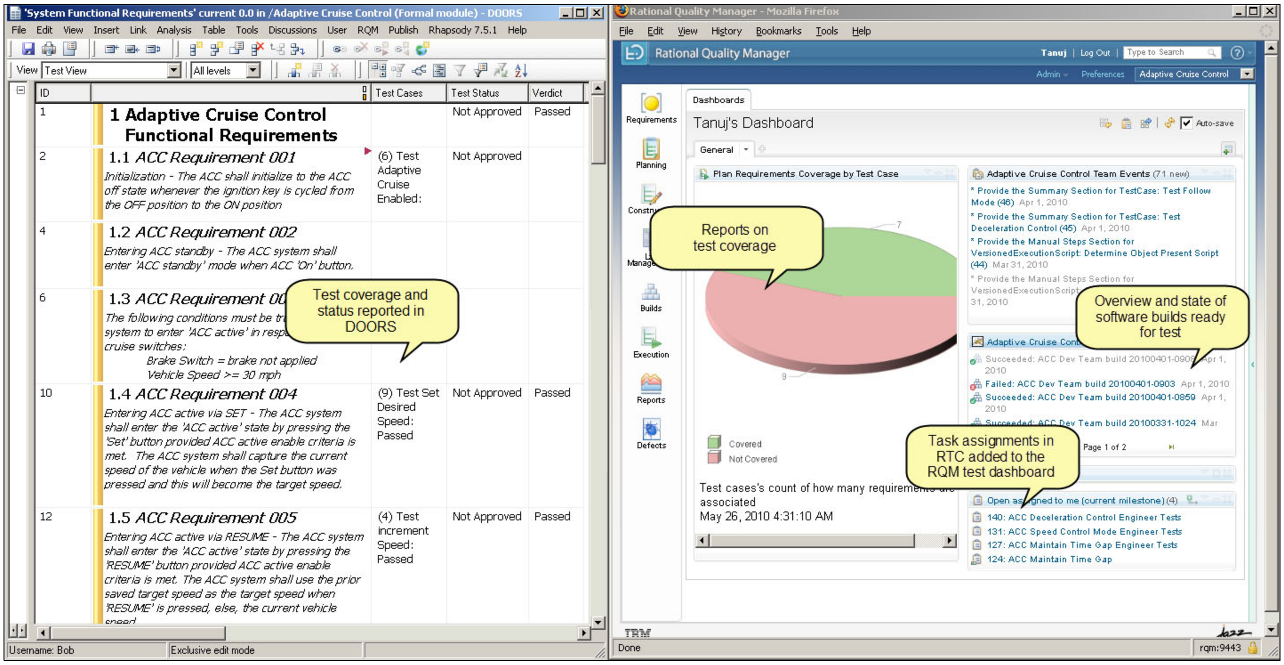
- Configuration and change management (with the IBM Rational Team Concert software)
- Requirements management (with the IBM Rational DOORS software)
- Quality management (with the IBM Rational Quality Manager software)
- Partner integrations, such as with Wind River
- Mathematical tools, such as Simulink

The IBM Rational Rhapsody software also has a large number of special purpose add-ons, such as the UML profile for DoDAF and MoDAF (UPDM) for DoD and MOD architectures, a safety analysis profile for analysis of safety critical and high-reliability systems, a profile for development of functionally-oriented embedded C programs, and support for the UML profile for test, including the IBM Rational Test Conductor software and the Automatic Test Generator (ATG) tool. These tools automate the ability to capture test cases from the model, create test architectures to support them, and even automatically run the test suites. The ATG tool can even parse the model and ensure that all model elements and features are tested by creating any missing test cases.

Quality management

The IBM Rational Workbench for Systems and Software Engineering software includes the IBM Rational Quality Manager software, which is an integrated environment for test planning, test construction, and test execution. The IBM Rational Quality Manager software also provides effective workflow control, tracking, and traceability features for test and verification teams. Because it is a web-based test solution, testers can easily access test assets and start or run test executions from any web browser.

Using the IBM Rational Quality Manager software dashboard, shown in Figure 15, testers can review the systems quality metrics and project status, which might include tasks assigned in the IBM Rational Team Concert software, or software integration builds ready to be tested.



File Edit View Insert Link Analysis Table Tools Discussions User RQM Publish Rhapsody 7.5.1 Help
View
File Edit View History Bookmarks Tools Help
Tanuj | Log Out | Type to Search
Admin
Preferences
Requirements
Planning
Constru
Manage
Builds
Execution
Reports
Defects
Dashboards
General
Auto-save
7
9
IBM
Jazz

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

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

Username: Bob — Exclusive edit mode

Test coverage and status reported in DOORS

Rational Quality Manager - Mozilla Firefox

Rational Quality Manager

Adaptive Cruise Control

Tanuj's Dashboard

Plan Requirements Coverage by Test Case

🟩 Covered
🟥 Not Covered

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

Reports on test coverage

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 Cont

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

Overview and state of software builds ready for test

Open assigned to me (current milestone) (4)

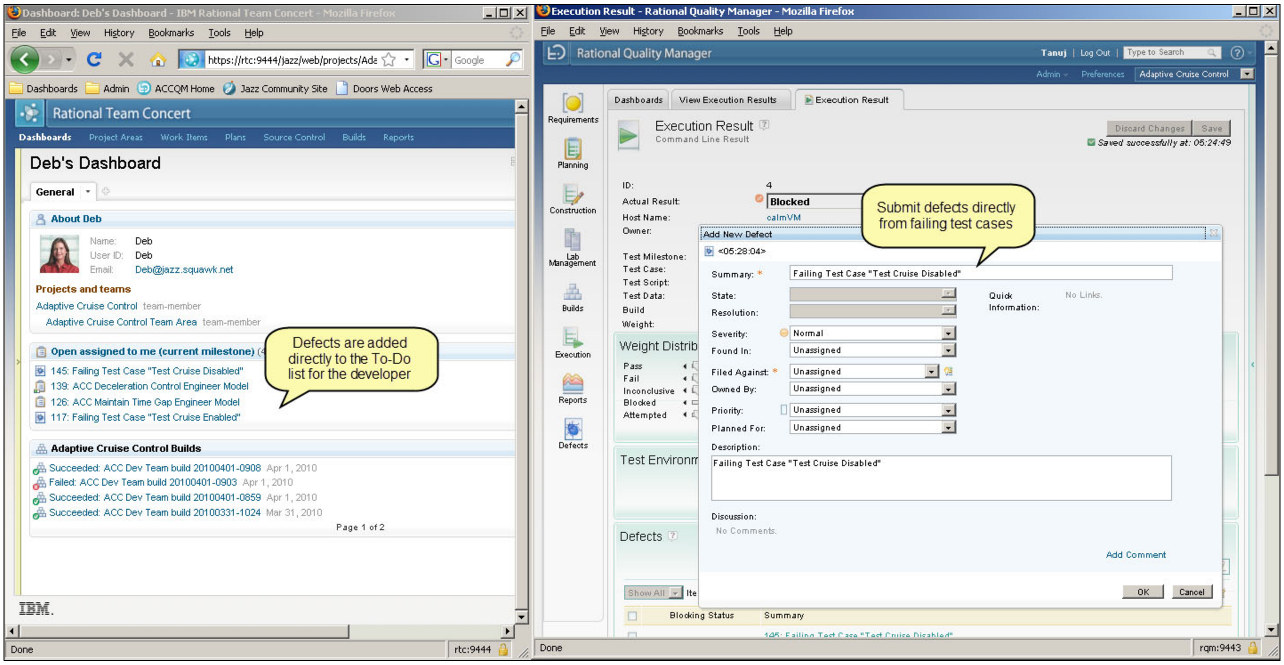
140: ACC Deceleration Control Engineer Tests
131: ACC Speed Control Mode Engineer Tests
127: ACC Maintain Time Gap Engineer Tests
124: ACC Maintain Time Gap

Task assignments in RTC added to the RQM test dashboard

Done — rqm:9443

Figure 15 Test management integrates test planning and execution with requirements

Like other products in the Workbench, the Rational Quality Manager software is integrated using the Jazz platform and OSLC. This integration enables testers to browse the web of work products and navigate from test cases to plan items, requirements, and model elements. OSLC integrations also enable testers to create and link defects directly into the development repository, as shown in Figure 16.



Dashboard: Deb's Dashboard - IBM Rational Team Concert - Mozilla Firefox

Deb's Dashboard

About Deb

Name: Deb
User ID: Deb
Email: Deb@jazz.squawk.net

Projects and teams

Adaptive Cruise Control team-member
Adaptive Cruise Control Team Area team-member

Open assigned to me (current milestone)

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"

Adaptive Cruise Control Builds

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 31, 2010

Page 1 of 2

Execution Result - Rational Quality Manager - Mozilla Firefox

Execution Result
Command Line Result

Saved successfully at: 05:24:49

ID: 4
Actual Result: Blocked
Host Name: calmVM

Add New Defect

<05:28:04>

Summary: Failing Test Case "Test Cruise Disabled"
Severity: Normal
Found In: Unassigned
Filed Against: Unassigned
Owned By: Unassigned
Priority: Unassigned
Planned For: Unassigned
Quick Information: No Links.
Description: Failing Test Case "Test Cruise Disabled"
Discussion: No Comments.

File Edit View History Bookmarks Tools Help
https://rtc:9444/jazz/web/projects/Ada
Dashboards Admin ACCQM Home Jazz Community Site Doors Web Access
Rational Team Concert
Dashboards Project Areas Work Items Plans Source Control Builds Reports
General
IBM.
Done rtc:9444
Defects are added directly to the To-Do list for the developer
Rational Quality Manager
Tanuj Log Out Type to Search
Admin Preferences Adaptive Cruise Control
Requirements Planning Construction Lab Management Builds Execution Reports Defects
Dashboards View Execution Results Execution Result
Discard Changes Save
Owner: Test Milestone: Test Case: Test Script: Test Data: Build Weight:
Weight Distrib
Pass Fail Inconclusive Blocked Attempted
Test Environm
Defects
Show All
State: Resolution:
Add Comment
OK Cancel
Blocking Status Summary
145: Failing Test Case "Test Cruise Disabled"
Submit defects directly from failing test cases
Done rqm:9443

Figure 16 Test execution integrates with defects and engineering workflows

The Jazz platform integrations in the Workbench, based on OSLC, enable testers and engineers to collaborate in-context on test results and submitted defects. Queries help testers find completed engineering tasks ready to be tested, or fixed defects ready to be re-validated. Testers also submit defects directly into the IBM Rational Team Concert software so engineers are immediately informed if testing is blocked. Links in the defect to the test execution result and test logs provide engineers important information to quickly uncover software issues and deliver defect resolutions.

The IBM Rational Test Lab Manager software integrates fully with the IBM Rational Quality Manager software and helps improve the efficiency of system test labs. The IBM Rational Test Lab Manager software optimizes how resources are requested and provided. It also integrates with system and software engineering test solutions.

Extending the IBM Rational Workbench for Systems and Software Engineering software

While the IBM Rational Workbench for Systems and Software Engineering software provides a complete solution for systems development teams, it can be extended with additional capabilities from the IBM Rational solutions portfolio shown in Figure 17. These capabilities include:

- Systems of systems

  For integration of manufactured products with the Internet and enterprise IT systems
- Business planning and alignment

  Capabilities for project portfolio management and enterprise architecture
- Product life cycle management

  Integrations with mechanical, electric, and electronic tools and data
- Reporting and metrics

  Dashboard mashups and automated document generation
- Service management

  Linkages to operations and capabilities for enterprise asset management

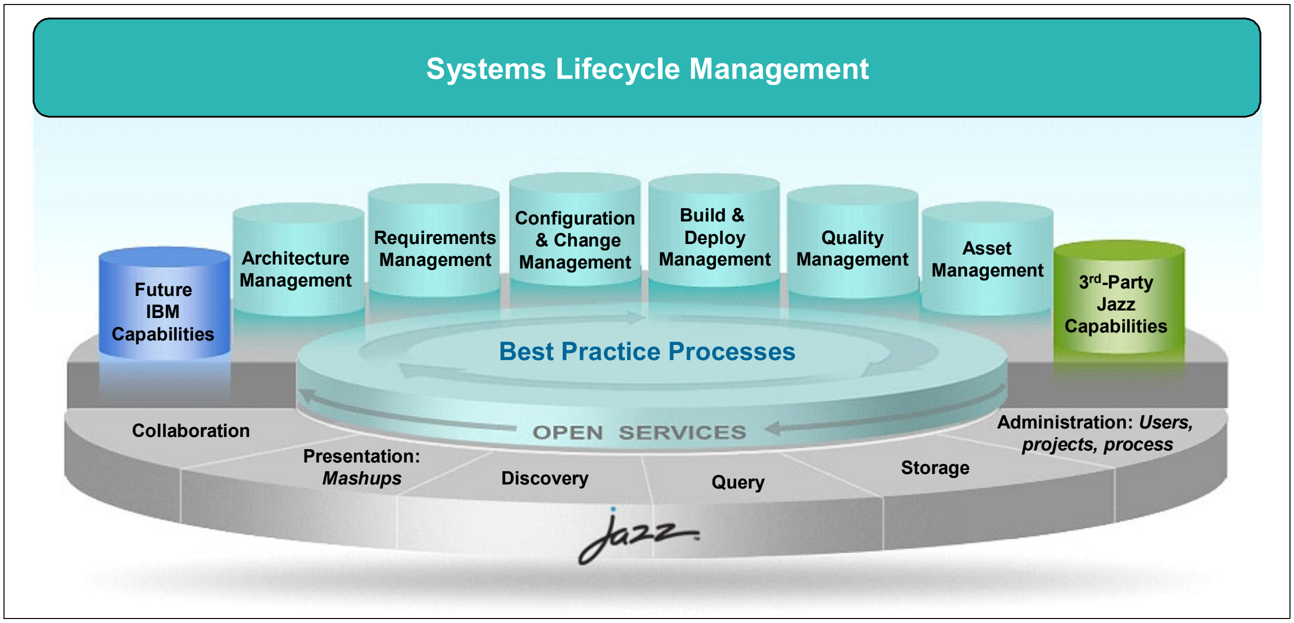


Figure 17 Extending the Workbench solution with additional capabilities

IBM Rational Publishing Engine software and IBM Rational Insight software

The IBM Rational Publishing Engine software automates the generation of documents that can assist the organization in formal reviews or in providing documentation for regulatory compliance requirements. This can help improve productivity and reduce risk and cost. The IBM Rational Publishing Engine software integrates with the IBM Rational Workbench for Systems and Software Engineering software and generates composite documents from the systems life cycle repositories.

The IBM Rational Insight software adds performance measurement and analysis to the management and reporting solution. It helps projects improve process performance through reports and dashboards.

IBM Rational Method Composer software

Systems delivery processes, like the IBM Rational Harmony process, provide a framework for process adoption and development capability improvements. The IBM Rational Method Composer software provides a modeling environment for the creation, management, and publishing of configured process content. The IBM Rational Method Composer software includes collections of practices that use leading process expertise and experience in systems engineering, governance and compliance, requirements management, quality management, and change and release management. Configured processes are published to the web and used in the enactment of the systems delivery processes in the Workbench.

Summary

The ability to engineer complex systems is a key enabler for the design and delivery of smarter products. Such products, or devices in systems of systems, include an increasing amount of software that integrates with hardware and electronics.

This paper has illustrated the systems delivery workflows supported by the IBM Rational Workbench for Systems and Software Engineering software. There are many individual work tasks that compose these activities, which results in the creation and modification of a potentially large number of work products. These work products include the stakeholder, system, and software requirements, systems engineering and software model(s), project release and milestone plans, integration, and validation test plans, and software components, source code, and builds. Developing highly reliable and safe systems is a complex and difficult undertaking. The IBM Rational Workbench for Systems and Software Engineering software targets these challenges by providing an integrated life cycle solution for the engineering teams.

Systems engineering focuses on the creation of requirements that are correct, complete, consistent, and at the right level of fidelity, the creation of an optimal architecture that supports the necessary functionality while optimizing various design criteria, and the allocation of those requirements into the identified architectural elements. Embedded and technical software continues with the creation of high-quality software that provides much of the functionality and personality of the system, creating high value for its consumers.

The IBM Rational Workbench for Systems and Software Engineering software is a systems delivery solution that assists with collaboration, automation, and reporting. The Workbench helps create and manage the life cycle work products necessary to create today’s complex systems. The IBM Rational Team Concert software provides a core platform for systems and software engineering teams, enabling not only engineering collaboration, but also an unprecedented level of transparency of project progress and quality for management. The IBM Rational DOORS software is the preeminent requirements management tool providing such key capabilities as traceability, coverage analysis, and change impact analysis. The IBM Rational Rhapsody software is the best-of-breed modeling tool for both systems engineering and embedded and technical software development. The IBM Rational Quality Manager software provides test management, test execution, and reporting. The IBM Rational Workbench for Systems and Software Engineering software is integrated to provide a set of best-of-breed capabilities for systems development and delivery on a common platform.

For more information

To learn more about how IBM can help you develop and deliver better software, contact your IBM representative or IBM Business Partner, or visit the following address:

`http://ibm.com/rational/solutions/systems`

In addition, financing solutions from IBM Global Financing can enable effective cash management, protection from technology obsolescence, improved total cost of ownership, and return on investment. Also, our Global Asset Recovery Services help address environmental concerns with new, more energy-efficient solutions. For more information about IBM Global Financing, visit the following address:

`http://ibm.com/financing`

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper:

- Douglass, *Design Patterns for Embedded Systems in C: An Embedded Software Engineering Toolkit*, Elsevier Science & Technology Books, 2010, ISBN 1856177076
- Douglass, *Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns*, Addison-Wesley, 1999, ISBN 0201498375
- Douglass, *Real Time UML Workshop for Embedded Systems*, Elsevier Science, 2006, ISBN 0750679069
- Douglass and Booch, *Real-Time Agility: The Harmony/ESW Method for Real-Time and Embedded Systems Development*, Addison-Wesley, 2009, ISBN 0321545494
- Douglass and Harel, *Real Time UML: Advances in the UML for Real-Time Systems*, Addison-Wesley, 2004 ISBN 0321160762
- Douglass and Jensen, *Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems*, Addison-Wesley, 2002, ISBN 0201699567

The team who wrote this paper

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center:

Bruce Powel Douglass has a PhD. in Neurocybernetics, and over 30 years of experience designing safety-critical and real-time applications in a variety of hard real-time environments. He has designed and taught courses in agile methods, object-orientation, MDA, real-time systems, and safety-critical systems development, and is the author of a number of technical books, including the ones listed in “Related publications”. He is the Chief Evangelist at IBM Rational, where he mentors IBM professional staff, consults with many customers all over the world, represents IBM at many different conferences, and authors tools and processes for the embedded real-time industry.

Mats Gothe is a solution architect in the Collaborative Life Cycle Management team with a focus on solutions for IT and systems. He has been with the IBM Rational brand for over 19 years and has held various positions, which have included manager for IBM Rational Rose® development in Sweden and IBM Rational technical sales and service manager in the Nordic region, as well as for the Ericsson Corporate Account team. He is an author of several publications, including the IBM Redbooks® publication *Collaborative Application Lifecycle Management with IBM Rational Products*, SG24-7622. He has a PhD. in Physics from Uppsala University and is based in Sweden.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author - all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document REDP-4681-00 was created or updated on June 30, 2010.

Send us your comments in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbooks@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DOORS®
IBM®
Jazz™
Rational Rose®
Rational Team Concert™
Rational®
Redbooks®
Redpaper™
Redpapers™
Redbooks (logo) ®
Rhapsody®

The following terms are trademarks of other companies:

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.