



---

# **Tools and Implementation Strategies for Process Improvement via CMMI for Comprehensive Software Lifecycle Management**

---

**Denise Padilla**  
Sandia National Laboratories

September 2009



# Verification, Validation, and Accreditation (VV&A)

---

- **Sandia National Laboratories has developed advanced modeling and simulation tools to support the Army's Brigade Combat Team Modernization program**
- **The Army requires formal VV&A of the simulation tools to ensure their capabilities are acceptable, reliable, and accurate for their intended use**
- **Capability Maturity Model Integration (CMMI) is listed as an acceptability criteria for accreditation within the Army**
- **Therefore, we have employed CMMI as our process improvement model to ensure**
  - **Alignment with the Army's software quality assurance requirements**
  - **Alignment with Sandia's corporate software quality assurance policies**
  - **Confidence in the integrity, quality, and performance of our software**



# CMMI Tie to VV&A

---

- **Rather than a broad attempt to improve in all process areas, initial efforts were focused on pursuing the specific process areas where most beneficial and directly related to VV&A**
  - **Configuration Management (CM)**
  - **Requirements Development (RD)**
  - **Requirements Management (REQM)**
  - **Verification (VER)**
  - **Validation (VAL)**
- **Employing the continuous representation of the CMMI model is most advantageous for this approach to process improvement**



# Tools for Comprehensive Software Lifecycle Management

---

- **We are using Serena products for software lifecycle management**
  - Dimensions RM
  - Dimensions CM
  - Business Mashups
- **The Serena tools are highly configurable**
  - Allows for implementation of our CMMI-based processes
  - Allows for bidirectional data flow between the tools



# Requirements Management using Dimensions RM

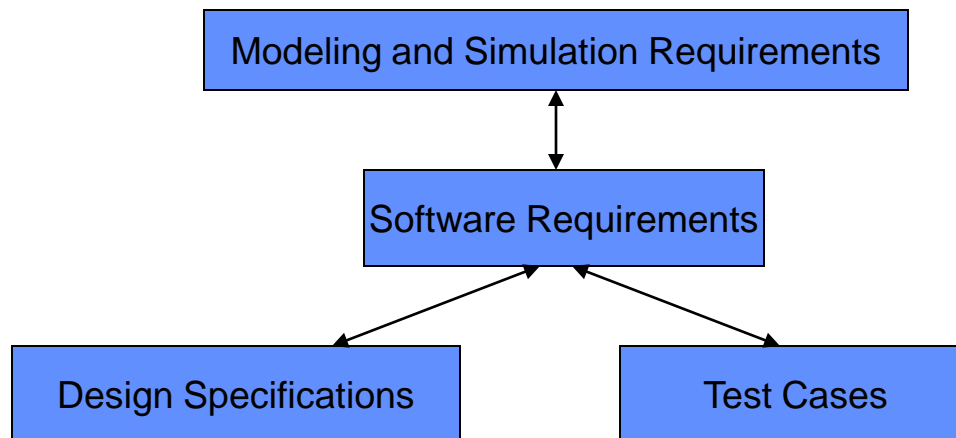
---

- **RM is used for managing and tracking the following requirements/objects throughout the project lifecycle:**
  - Modeling and simulation requirements
  - Software requirements
  - Design specifications
  - Verification test cases
  - Validation test cases
- **We are using RM to**
  - Control and track changes to requirements/objects
  - Track the type, status, and other attributes of a requirement/object
  - Baseline requirements/objects
  - Determine the impact of a changing requirement/object on
    - Other requirements
    - Design specifications
    - Testing
    - Schedule
  - Create and maintain traceability between requirements and work products such as design specifications and test cases
    - To ensure all requirements are allocated and tested
    - To ensure the requirements and design elements are comprehensive
  - Create and publish the following documents
    - Software Requirements Document
    - Detailed Design Document
    - Verification Test Digest
    - Validation Test Digest

# Requirements Traceability in Dimensions RM

---

- **Explicit links created in RM for bidirectional traceability**





# Configuration Management using Dimensions CM

---

- **Dimensions CM is used for change and configuration management of all project artifacts**
- **We are using CM to**
  - **Track, authenticate, and version changes**
  - **Access all project artifacts**
  - **Manage the interdependencies of all project artifacts**
  - **Ensure backup and retrieval of baselined product artifacts over the project lifetime**
  - **Manage the software development lifecycle (Development, Quality, Production)**
  - **Provide baseline management, build and release management**
  - **Provide visibility into project status**



# Change/Work Request and Process Management using Business Mashups

---

- **Mashups is used for issue management and process management and enforcement**
- **We are using Mashups for**
  - **Change requests**
    - Issue and defect tracking
  - **Work/task requests**
    - New feature requests
    - Analysis requests
    - Baseline requests
    - Build requests
    - Release requests
    - Test requests
    - Process Improvement requests
  - **Monitoring the status of requests through to resolution**
  - **Tracking histories of and managing relationships among requests**
  - **Analyzing trends and activity reports to improve processes**
  - **Providing comprehensive process control (process management)**
  - **Automating the development process to ensure repeatability, predictability, and accountability**





# Integration of Mashups, CM, and RM

---

- **Mashups has been integrated with RM and CM to manage the complexity of application development processes throughout the lifecycle**
- **The integration provides traceability between**
  - change/work requests
  - modeling and simulation requirements
  - software requirements
  - verification test cases
  - validation test cases
  - design specifications
  - and all versions of project files and artifacts
- **All changes to configuration items (in RM and CM) must be linked to a request from Mashups and approved by the appropriate change control board (CCB)**

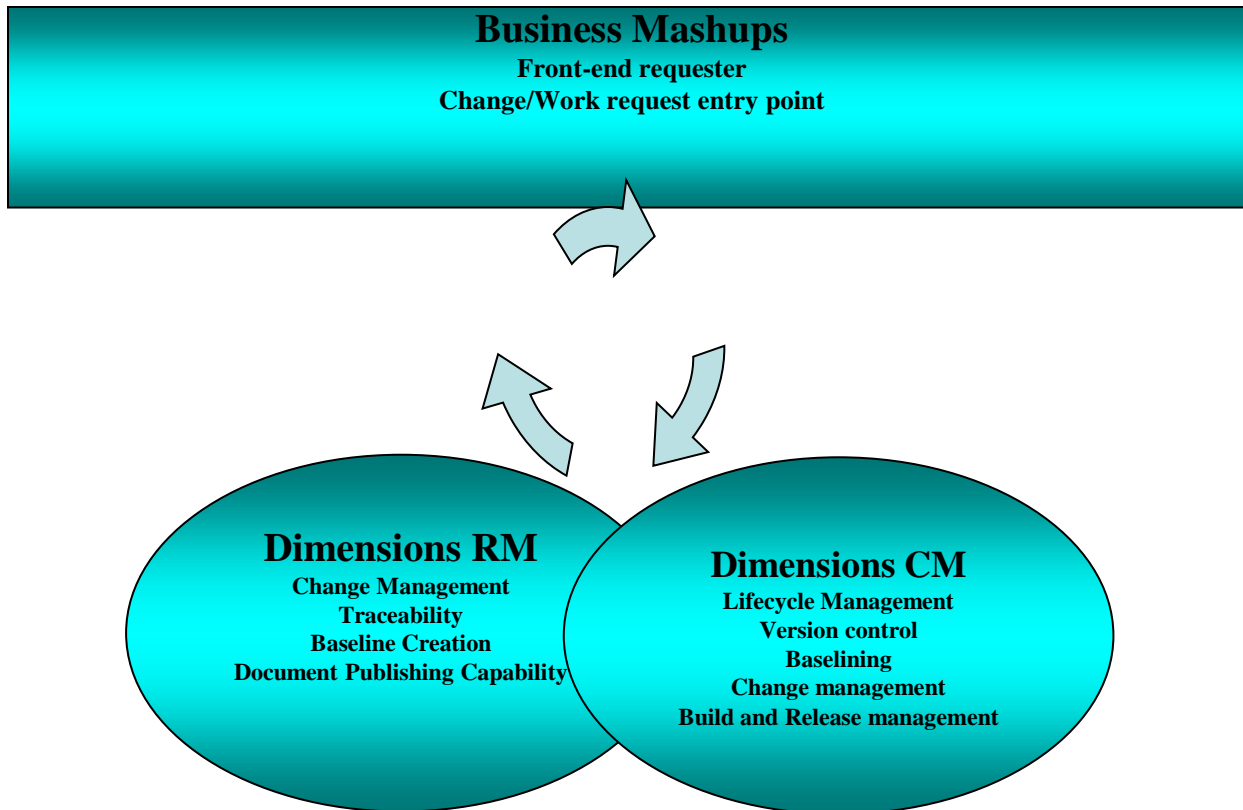


# Integration of Mashups, CM, and RM

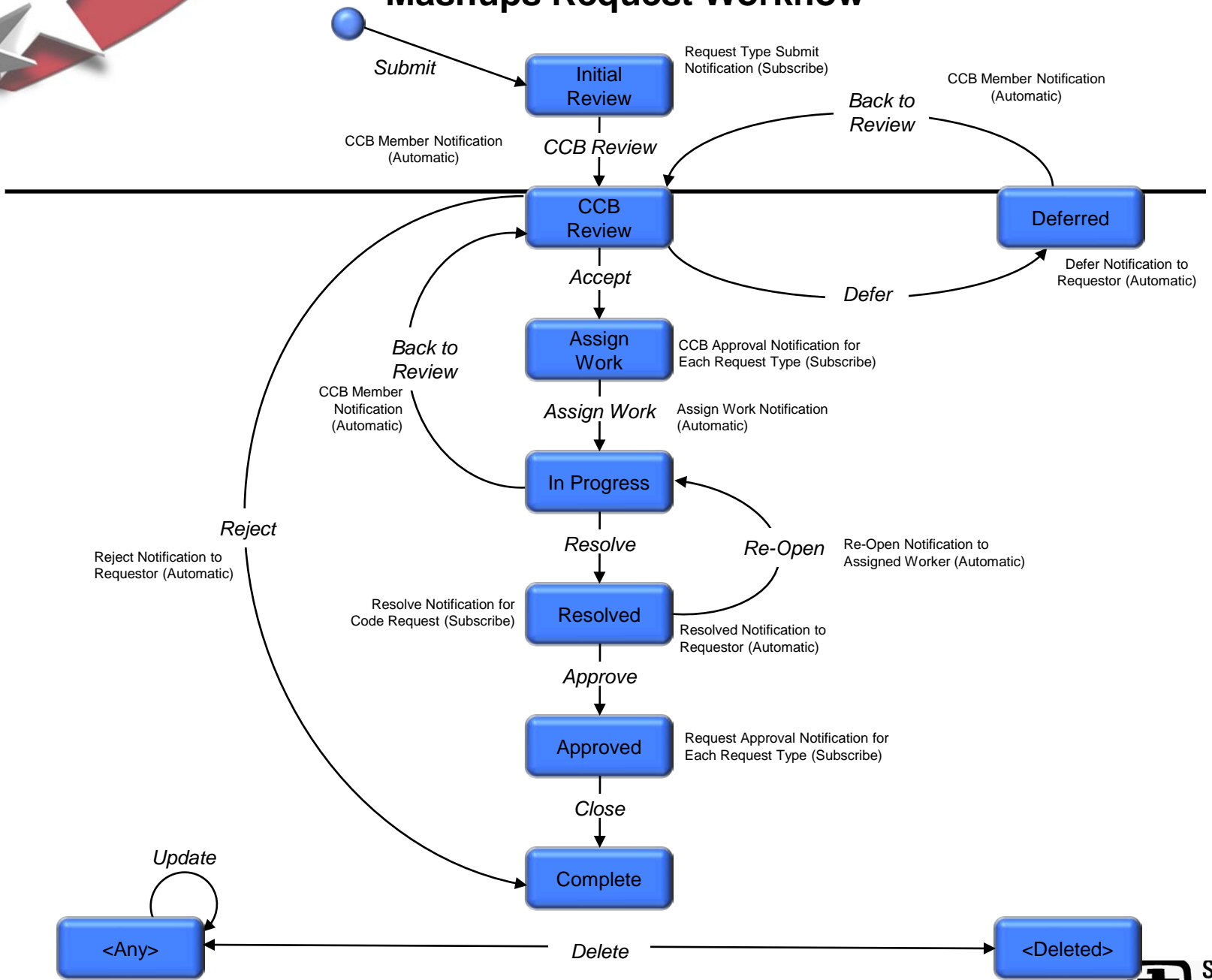
---

- **The integrated solution provides**
  - Bidirectional data flow across all three tools
  - Tight integration between requirements management, change and configuration management, build and release management, baseline management, and deployment readiness through a common process model
  - Visibility into the project status
  - Repeatability
  - Accountability
  - Comprehensive application lifecycle management from conception through delivery and maintenance
  - Process-centric software development
  - Process enforcement
  - The infrastructure for producing formal VV&A artifacts
- **The Mashups workflow**
  - Incorporates various CMMI practices
  - Eliminates the need for a dedicated configuration manager
  - Has been designed to provide process enforcement with flexible degrees of formality for projects of various
    - Sizes
    - Types
    - Funding levels
    - Risk levels
    - Turn around times

# Integration of Dimensions RM, Dimensions CM, and Business Mashups



# Mashups Request Workflow



# Mashups

[Back to Results](#)

**SoSAT Systems Sustainment** Bug00027: Diagnostic checkbox error can yield no diagnostic results (Inactive)

## State Change History



## REQUEST INFORMATION

<b>Issue Id:</b>	Bug00027	<b>Item Type:</b>	Bug
<b>Title:</b>	Diagnostic checkbox error can yield no diagnostic results	<b>Affected Model:</b>	N/A
<b>Description:</b>	02/24/2009 08:07:38 AM - Hatcher, Jesse: In the Diagnostics Options pop-up, if a top-level checkbox is marked with a green box (indicating that some, but not all, of the sub-items are checked) and the running the diagnostic will yield no results.		
<b>Request Type:</b>	Code	<b>Components:</b>	Input Editor
<b>Affected Version:</b>	SoSAT 1.5	<b>CM Project:</b>	SOSAT:DEVELOP_SOSAT_2.0
<b>Priority:</b>	Minor		
<b>Environment:</b>	Windows XP		
<b>CONTACT INFORMATION:</b>			
<b>Requester Name:</b>	Hatcher, Jesse		
<b>Requester Phone Number:</b>	844-5169		
<b>Requester E-mail Address:</b>	jphatch		
<b>RTM Rqmt ID:</b>			
<b>RTM Status:</b>	Accepted		
<b>RTM Requirement Text:</b>			

# Dimensions RM

Microsoft Internet Explorer - Edit Attributes - Dimensions RM

## Test\_Case\_Details: TC\_122

Action:  Edit attributes  Add a comment  Submit a change request

**Rqmt ID:** TC\_122      **Test Case Name:** Test Case: Verify Rescale works

**Test Case Description:** Test Case: Verify Rescale works properly  
This test will verify that systems rescale properly, including checking that changes to elements of a System Type will propagate to the Systems and that change to the Time-to-Fail and Multiplier Distributions for an element of a System Type will propagate to the Systems Time-to-Fail Distributions. (Note that these tests ensure that the UI is updated, as the Systems change. To verify that the values are correct, see the "System Uncertainties" test cases.)  
This test case is comprised of 3 sub-test cases.  
1. Verify Scaling works properly when properties of a Primary Element are changed.

**Custom Attributes**

**System Attributes**

**Created By:** MGENDREAU      **Current Status:** Current  
**Modified By:** TLGREENE      **Object ID:** 133  
**Suspect:** No      **Time Created:** 21-SEP-2007@15:09:53  
**Time Modified:** 09-FEB-2008@11:36:15

**Links**

**Project\_Requirements (2 links)** [Expand all/Collapse all](#)

Rqmt ID	Requirement Title
RQMT_20	SoSAT shall allow the user to
RQMT_35	SoSAT shall provide the user the

**Test\_Case\_Header (1 link)**

**Test\_Case\_Details - Secondary Requirements (3 links)** [Properties](#) [\\*](#) [X](#)

Rqmt ID	Test Case Name
TC_123	Sub-Test Case: Verify Scaling
TC_124	Sub-Test Case: Verify Scaling
TC_125	Sub-Test Case: Verify Scaling

**Test\_Case\_Details - Primary Requirements (0 links)**

[Show navigation bar](#)

Update Replace Cancel

# Dimensions CM Baseline View

The screenshot displays the Serena Dimensions software interface. The main window is titled "[dim\_intermediate-dev] Serena Dimensions - [Baseline Catalog]". The interface includes a menu bar (File, Edit, View, Item, Request, Baseline, Release, Project, Tools, Window, Help), a toolbar, and a left-hand navigation pane. The navigation pane shows a tree structure under "My Current Project" with folders like "Folders and Items", "Parts and Items", "Work Areas", "Deployment Areas", "Inbox", "Catalog", and "Sub-Projects". The "Catalog" folder is selected, and the "Baseline Catalog" window is open. This window shows a table with columns "Specification", "Type", and "Template". One entry is visible: "PAYROLL:BL\_VBGUI\_REL\_1\_A" with Type "Release" and Template "ALL\_ITEMS\_LATE".

Below the "Baseline Catalog" window, there are sections for "5 Related Items" and "1 Related Request". The "5 Related Items" section contains a table with columns "Filename", "Revision", "Status", "Update Date", and "Specific". The "1 Related Request" section contains a table with columns "Title", "ID", "Stage ID", and "Status".

Annotations with arrows point to the "Baseline Catalog" window and the "1 Related Request" table. The text "What's in the Baseline" is positioned above the "5 Related Items" table, and "Work/Change Request Tied to the Baseline" is positioned below the "1 Related Request" table.

Specification	Type	Template
✓ [icon] PAYROLL:BL_VBGUI_REL_1_A	Release	ALL_ITEMS_LATE

Filename	Revision	Status	Update Date	Specific
✓ [icon] vb_make.mak	win95#1	UNDER WORK	Jun/20/2001 17:37:44	PAYROL
[icon] payroll_vb.vbp	win95#1	UNDER WORK	Jun/20/2001 17:37:44	PAYROL
[icon] payroll_vb.vb	win95#1	UNDER WORK	Jun/20/2001 17:37:44	PAYROL
[icon] Form1.frx	win95#1	UNDER WORK	Jun/20/2001 17:37:43	PAYROL
[icon] Form1.frm	win95#1	UNDER WORK	Nov/20/2001 17:05:33	PAYROL

Title	ID	Stage ID	Status
[icon] Error in breakdown calculations	PAYROLL_CR_17		RAISED

For Help, press F1

[ 1 ] ENTERPRISE 1.5 [ 5 ] WS\_DEV\_WIN

dim\_intermediate-dev [dim\_intermediate@cmdv1] (no profile) NUM



# Software Plans and Supporting Documentation

---

- The following documents are results of both our VV&A efforts and the implementation of CMMI practices
  - Verification, Validation, and Accreditation Master Plan
  - Verification and Validation Plan
  - Configuration Management Plan
  - Requirements Management Plan
  - Project Management Plan
  - Customer Support Plan
  - Release Management Plan
  - Test Plan
  - Software Quality Plan
  - Software Requirements Document
  - Master Test Digest
  - Master Design Document
  - Conceptual Model
  - SoSAT User Manual





# How CMMI Facilitates our Software Development

---

- **Provides a structured development environment**
  - Structured practices that cover the entire lifecycle, from conception through delivery and maintenance
  - Well-defined, stable, and repeatable processes
- **Helps our team to set improvement objectives and priorities**
- **Provides project continuity in the event of personnel change**
- **Promotes improved quality**
  - Defined quality assurance practices
  - Thorough and formal verification and validation practices