



# MELCOR Quality Assurance Practices

**Larry Humphries and John Reynolds**

## **MELCOR Software Quality Assurance Training**

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,  
for the United States Department of Energy's National Nuclear Security Administration  
under contract DE-AC04-94AL85000.

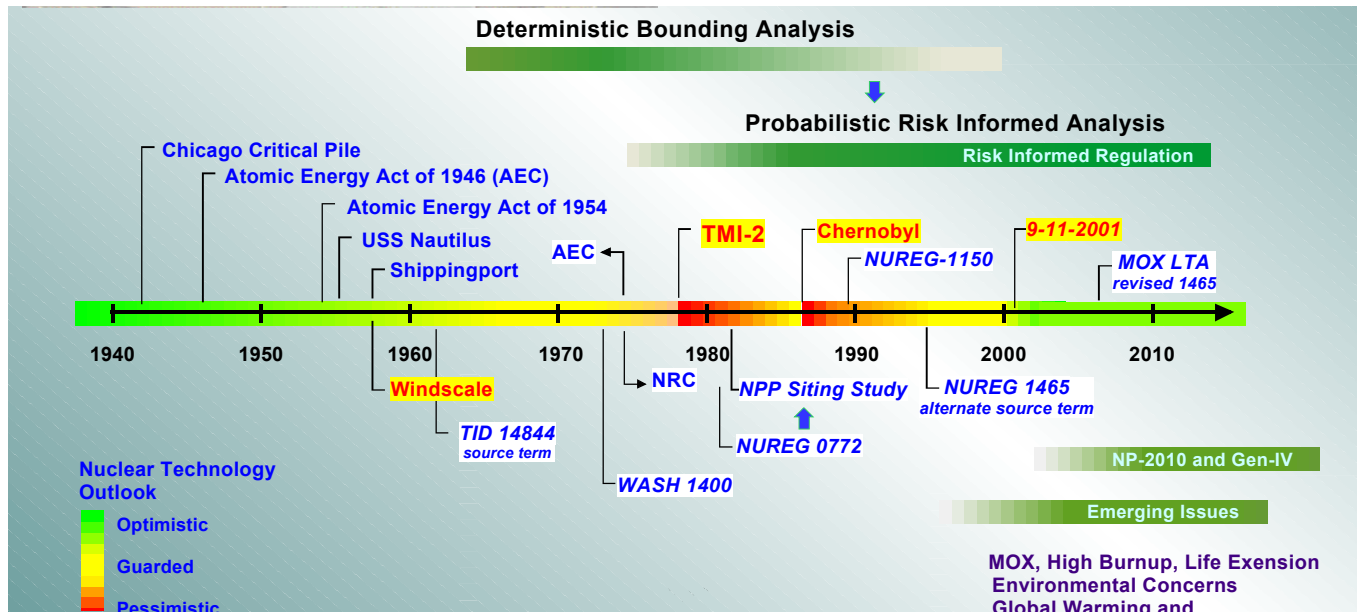




# MELCOR Code Development Project

- **MELCOR Development**

- MELCOR has been developed at SNL for the USNRC
  - Project began in 1982
  - Development of new capabilities still underway
- MELCOR Development is also strongly influenced by US NRC Severe Accident Cooperative Research Program (CSARP)





# MELCOR Code Development Thrust Areas

## Code reliability

- ◆ Validation
- ◆ QA
- ◆ Numerical stability

## User Utilities

- ◆ Converter
- ◆ Post-processors
- ◆ Uncertainty Engine
- ◆ DEF

## Code Enhancements

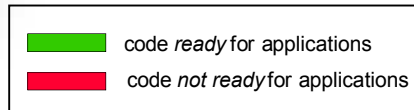
- ◆ New/improved modeling
- ◆ Code performance

**MELCOR  
Code  
Development**

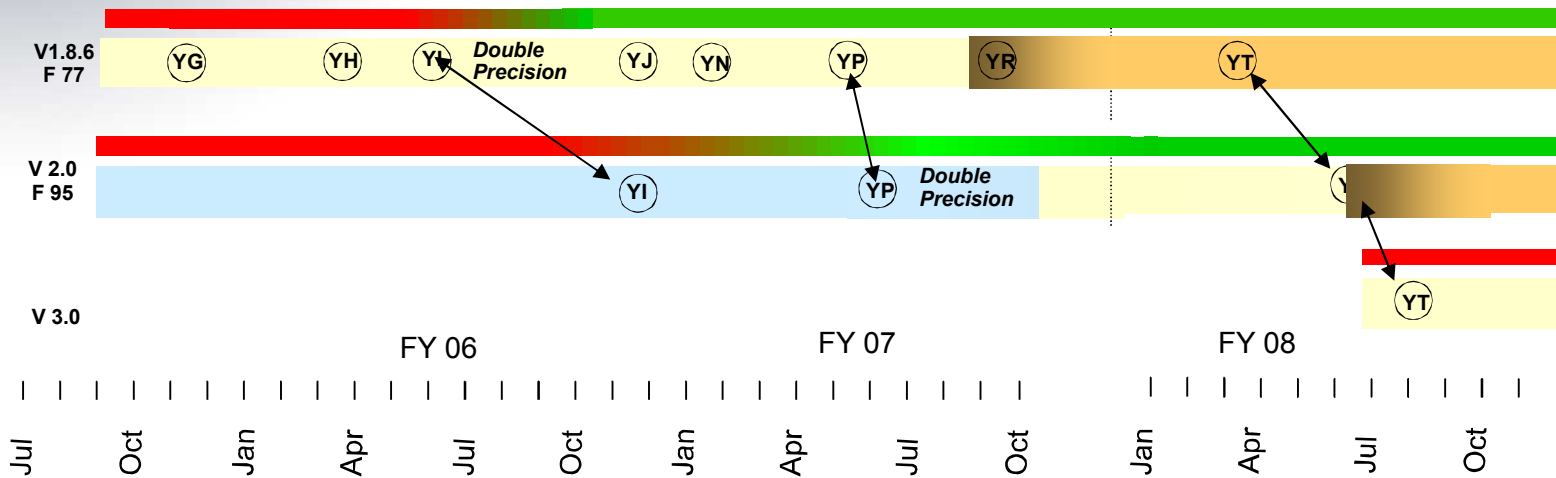
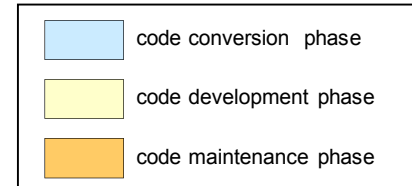
# MELCOR Code Development Timeline



## Code Readiness



## Code life-cycle



### • MELCOR 1.8.6

- Molten pool models
- Core Package upgrade
- Released Fall 2005
- Code Maintenance
- Current Workhorse

### • MELCOR 2.1

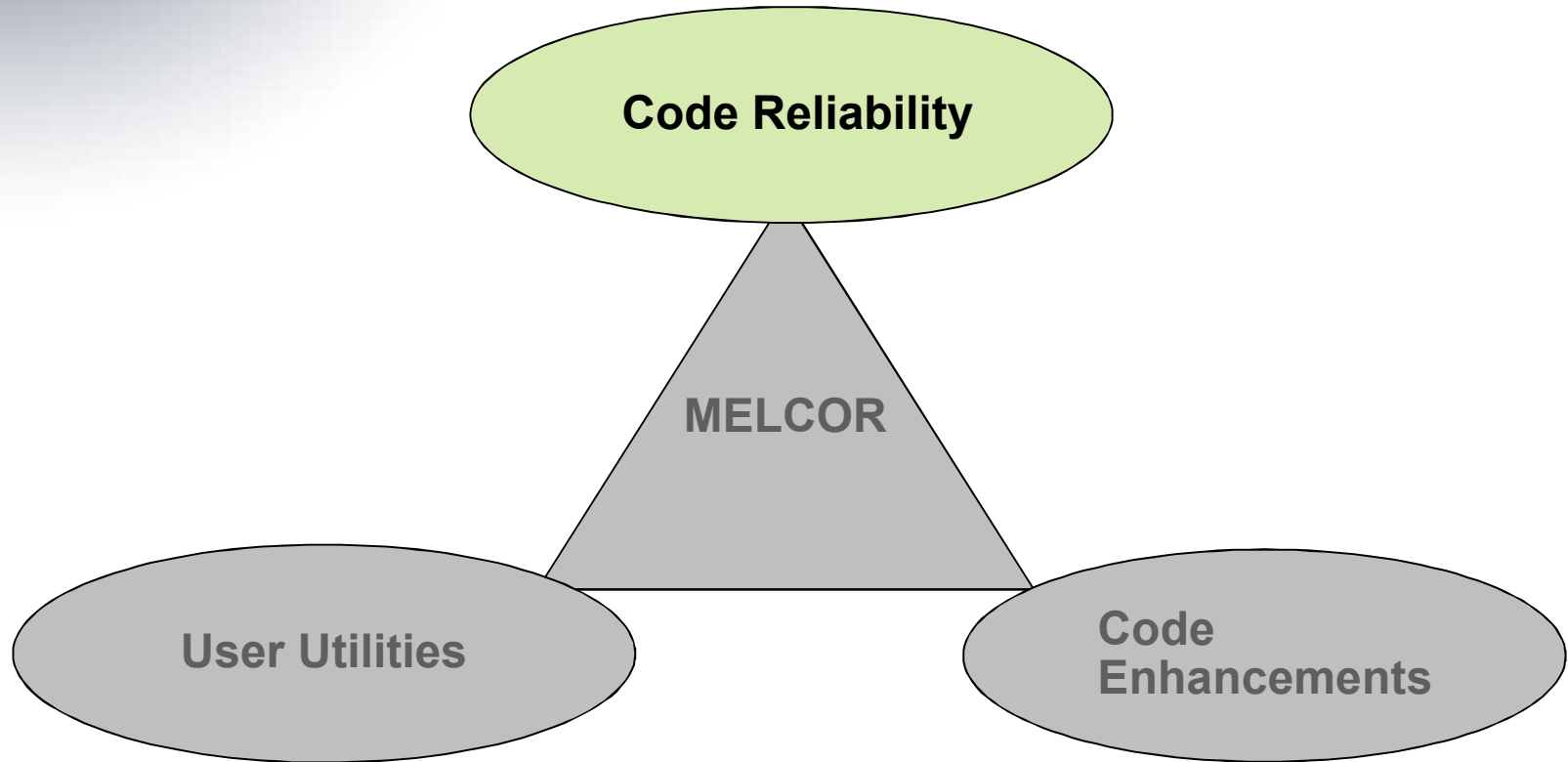
- FORTRAN 95
- New input
- 2.0 beta version released Sept 2006
- 2.1 Release Sept 2008

### • MELCOR 3

- Current developmental version



# Code Development: Code Reliability





# DOE Safety Software Software Quality Assurance

- A Sandia (Org 00421) assessment was performed on September 6, 2007 with the following objectives:
  - Provide MELCOR software team and management with a perspective of areas of software engineering/quality strength and areas of improvement - relative to the self-assessment instrument and requirements of DOE O 414.1C.
  - Provide a comparison of the assessment results and gaps for MELCOR with the gap analyses conducted by DOE in 2004.
  - Provide MELCOR with a potentially sustaining capability to conduct their own self-assessments and manage identified improvement areas.
- Sandia Corporate Process Requirement 001.3.6 (CPR 001.3.6)
  - The software management framework adapted from two internationally recognized standards
    - the Capability Maturity Model Integration (CMMI) ®
    - and ISO 9001
    - These standards provide elements of traceability, repeatability, visibility, accountability, roles and responsibilities, and objective evaluation



**Apprehension**



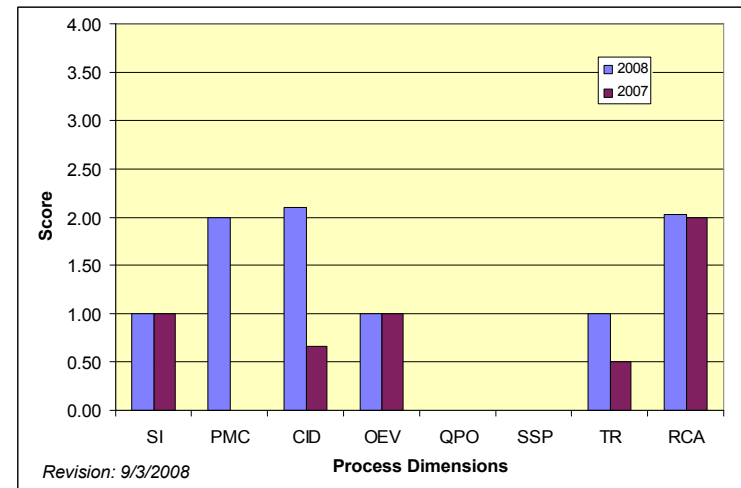
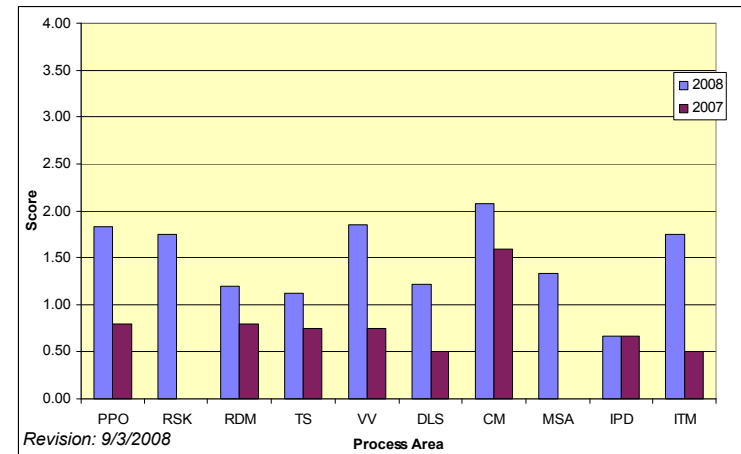
# Software Quality Assurance Annual Re-evaluation

- ◆ Internal review annually
- ◆ External review every 5 years

- Process areas

- Project planning and oversight
- Requirements Development and Management
- Risk Management
- Configuration Management
- Technical Solution
- Integrated Teaming
- Integrated Product
- Verification and Validation
- Measurement and Analysis
- Development and Lifecycle Support

**Problem areas**  
**Strengths**





# Continuous Integration

- **Advantages**

- When unit tests fail, or a bug is discovered, developers might revert the codebase back to a bug-free state, without wasting time debugging.
- Integration problems are detected and fixed continuously - no last minute hiatus before release dates;
- Early warning of broken/incompatible code;
- Early warning of conflicting changes;
- Immediate unit testing of all changes;
- Constant availability of a "current" build for testing, demo, or release purposes;
- The immediate impact of checking in incomplete or broken code acts as an incentive to developers to learn to work more incrementally with shorter feedback cycles.

- **Disadvantages**

- When a user checks in code modifications that breaks the code, all other developers are severely impacted



**Avoid “Cowboy”  
Programming**



# MELCOR Software Quality Assurance Best Practices

## Emphasis is on Automation

**Affordable solutions**

**Consistent solutions**

- **MELCOR Wiki**
  - Archiving information
  - Sharing resources (policies, conventions, information, progress) among the development team.
- **Code Configuration Management (CM)**
  - ‘Subversion’
  - TortoiseSVN
  - VisualSVN integrates with Visual Studio (IDE)
- **Code Review**
  - Code Collaborator
- **Nightly builds & testing**
  - DEF application used to launch multiple jobs and collect results
  - HTML report
  - Regression test report
- **Regression testing and reporting**
  - More thorough testing for code release
  - Target bug fixes and new models for testing
- **Bug tracking and reporting**
  - Bugzilla online
- **Assessment calculations**
- **Documentation**
  - Available on Subversion repository with links from wiki
  - Latest PDF with bookmarks automatically generated from word documents under Subversion control
    - Links on MELCOR wiki
- **Sharing of information with users**
  - External web page
  - MELCOR workshops
  - Possible user wiki



# MELCOR Developers Wiki Site Not Open to Public

## MELCOR Developers Wiki

### – Archive records

- Requirements, design, and testing
- Regression tests
- Assessment work

### – Information Sharing

- Debugging Policies
- Testing Policies
- Code Development practices
- Coding Conventions
- Lessons Learned
- Software Risk Management
- Version Changes
- Reference Library

The screenshot shows the MELCOR Wiki main page. At the top, there are navigation tabs for 'article', 'discussion', 'edit', and 'history'. Below this is the 'Main Page' header and the 'MELCORWIKI' title. A central table of contents lists various topics such as 'MELCOR Vision', 'MELCOR Project', 'Policies', 'MELCOR Contacts', 'Current Activities', 'Code Development Projects', 'Future Model Improvements', 'Current Model Development Activities', 'Current Assessment Activities', 'MELCOR 2.1 Release Activities', 'MELCOR Assessments', 'MELCOR Code Development', 'MELCOR Code Development: 62842 (Y8602)', 'Advanced Reactors: 62862 (Y8681)', 'MASCOT: 105224', 'mediawiki', 'Wiki User's Guide', 'Community portal', 'Current events', 'Recent changes', 'Random page', 'Help', 'Donations', 'search', and 'toolbox'. On the right side, there is a box titled 'The Melcor Code' with a description and a small image. Below that is a box titled 'Important Dates' listing several key events from 2006, including meetings and releases. At the bottom, there is a paragraph stating the goal of the MELCOR project: 'We will position MELCOR as the gold standard for severe accident modeling of new and existing reactors by developing new models and capabilities that are tested for robustness and have been extensively assessed against the world's available experimental data.'



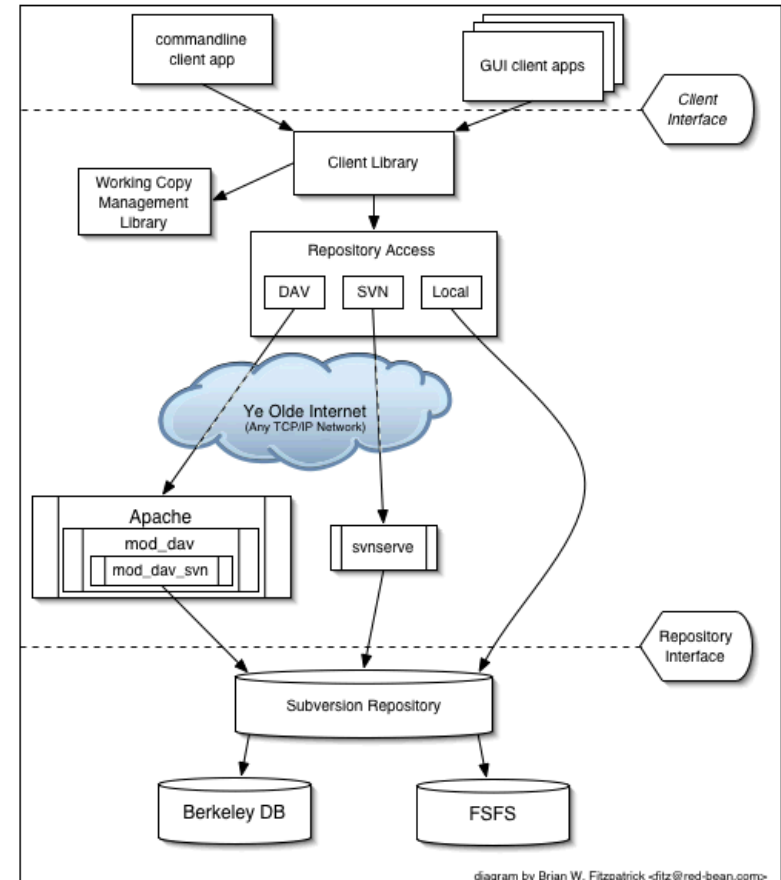
# Software Configuration Management (SCM)

- **Methodology for managing code changes in a team development environment**
- All SCM systems provide the following essential features:
  - Concurrency Management
    - Concurrency refers to the simultaneous editing of a file by more than one person.
      - Resolve places where code changes conflict
  - Versioning
    - Tracks file versions
    - Makes it possible to roll back changes or recreate a version
  - Synchronization
    - Update changes made by other developers



# Subversion (SVN)

- **Free Open Source**
- **Operates across networks allowing different people access to a central repository**
- **Atomic commits.**
  - No part of a commit takes effect until the entire commit has succeeded.
  - Revision numbers are per-commit,
  - Commit's log message is attached to its revision
- **Branching**
  - **Merging branches back into the trunk**
- **Repository can be locked**
  - Especially useful for binary files
- **Command Line Client**
  - Makes it possible to integrate SVN with other applications

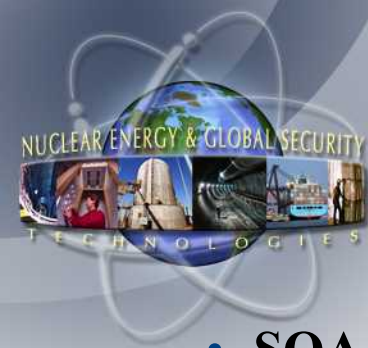




# TortoiseSVN

- **Powerful commit dialog**
  - Forces user to supply comments
  - Integrated spell checker for log messages
  - Auto completion of paths and keywords of the modified files
  - Text formatting with special chars
- **Useful Tools**
  - [TortoiseBlame](#)
  - Patch files
  - [StatSVN](#)
- **Integration with other products**
  - **Integrates Subversion with Windows explorer**
    - All commands are available directly from the windows explorer.
    - See the status of your files directly in the Windows explorer
    - Allows moving files by right-dragging them in the windows explorer
  - **Integration with issue tracking systems**
    - A separate input box to enter the issue number assigned to the commit, or coloring of the issue number directly in the log message itself
    - When showing all log messages, an extra column is added with the issue number. You can immediately see to which issue the commit belongs to.
    - Issue numbers are converted into links which open the web browser directly on the corresponding issue
    - Optional warning if a commit isn't assigned to an issue number
  - **Integration with MSWord compare**
  - **Integration with Visual Studio**
    - All modifications apparent within Visual Studios
    - Easy to see and check in all modified files (still not automatic)

**Tortoise SVN is a Subversion Client that integrates SVN with Windows Explorer**



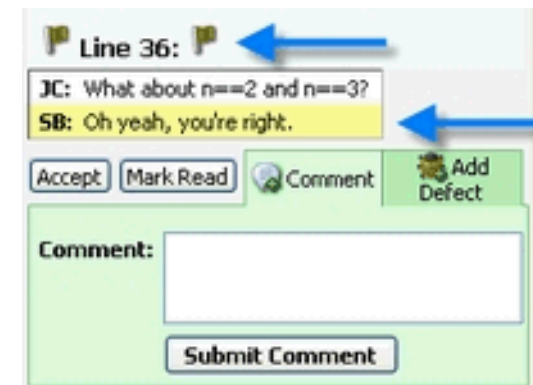
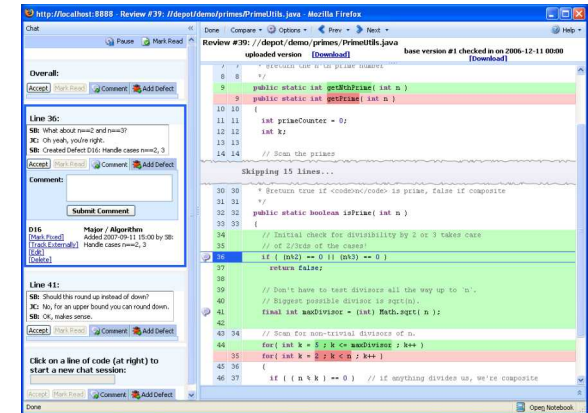
# Code Review

- **SQA Requirements:**
  - As outlined in both Sandia Corporate Procedure:IM100.3.2 and ASME NQA-1, an important aspect of a Software Quality Assurance program involves review and documentation for the entire life cycle of software development, from requirements and design to implementation and testing.
- **Benefits**
  - Code reviews can reduce number of defects in new code
  - Code reviews can lead to better documentation of code & better understanding of new modeling among team members
  - Code reviews provide a process by which seasoned programmers can pass experience and knowledge to less experienced programmers.
  - Improves code readability
- **Problems**
  - Code reviews can be time consuming or cursory
  - Code reviews are not well documented
  - Reviews can be adversarial and not productive



# Code Collaborator

- **HTML based collaboration**
  - Browser is all that is needed to access
  - Shared licenses for multiple users
- **Threaded, contextual chat**
  - File changes, chat conversations, and defects are linked together.
  - Each conversation is threaded by file and line number and can be viewed simultaneously with file content.
  - Choose between multiple views, or download differences to your local machine for further inspection and testing.
  - Hyperlink directly to a file or line numbers
- **Version Control Integration**
  - Integrates with Subversion
- **Asynchronous Review**
  - Perform and manage reviews even when participants are separated by many timezones.
  - Comments are tracked like newsgroups<sup>Vg# 15</sup>





# Code Collaborator (Continued)

- Review PDFs or Image files
  - Comments threaded by pushpin location
- Generate Reports
  - Metrics collected automatically (i.e., person-hours spent in review, LOC inspected, and defect data).
  - Customizable reports show everything from status of pending reviews to detailed audit trails.
- Contextual Defects
  - When a problem is found, create a defect. Customizable fields let you collect as much or as little data as you want.
  - Integration with Bugzilla
  - Defects are threaded by file and line number but are also viewable in an overall review log. Authors correct defects and can upload changes back to the review for verification.
  - CodeCollaborator keeps all chat conversations and defects threaded to the correct logical line of code even if line numbers shift around because of the defect fix.

## 1. Introduction

Radiological Assessment of the Nuclear Regulatory Commission during radiological releases uses various protective atmospheric release models. RASCAL uses NUI to estimate the fractional release of various radionuclides during the duration of various (BWRs).

While the accident :

ID	Review Title	Review Creation Date	Author Full Name	Defect Count	LOC Changed	Total Person-Time
1365	Diff view improvements	2007-09-05 17:01	Brandon DuRette	1	35	00:48:58
1362	Support modifying changelist for addactivity and addtrack	2007-09-04 15:18	Eric Brown	6	444	02:11:19
1358	User version editing for addactivity, addtrack	2007-09-04 15:18	Eric Brown	12	705	03:21:00
1357	auto-detect SCM system	2007-09-04 11:03	Roy Paterson	1	2122	01:17:50
1353	Added system option to disallow auto-reopen of completed reviews	2007-08-30 22:24	Jason Cohen	4	265	01:41:11
1329	Improve performance of diff widget and...	2007-08-22 15:47	Brandon DuRette	1	133	00:22:34



# MELCOR Testing Overview

- **Unit Testing**
  - All input options should be tested
  - Ranges of input should be tested.
    - Values outside of reasonable input should be tested for error messages
- **Automated Build & Test**
  - Test all revisions and ‘catch’ revisions that break the build
  - Build on multiple platforms (currently only Windows) and compilers (CVF & Intel)
  - Suite of fast-running test problems
  - Objective is to correct problems sooner
  - MOE utility for searching output for specified test for success criterion
- **Code Release Testing**
  - Larger general suite of test cases
  - Test that target recent bug fixes
  - Test that target new modeling
- **Code Test Coverage**
- **Code Profiling**



# Unit Testing

- **During development, write tests first—during the initial stages of code development (see [Code Model Development Procedures](#)). In so doing, you:**
  - **Describe what the code is supposed to do, using real input that addresses the specified behavior in concrete, verifiable terms. (Refer to the analyzed use cases for specific test requirements.)**
  - **Provide an example of how the code should be used as a working, tested example.**
  - **Provide a way to verify when the code is finished (that is, when all the tests run correctly).**
  - **Run unit tests every time the code is compiled. Perform this type of alpha testing routinely after every compilation to significantly reduce the accumulation of bugs.**
  - **Verify test coverage for new code added.**
    - **Add test cases to the Integration regression test suites (with an Excel spreadsheet of expected results) so that these new models can be tested after the model is completed.**



## Unit Testing (Continued)

- All input options should be tested
  - **Named Comment Block** feature was added to MELCOR
    - This feature was added to allow the user to include blocks of input records which are selected from a variety of input methods. It can be used for quality assurance purposes to test variations of input records or it can be used in calculations to accommodate variations in a single input deck that are easily activated or deactivated on input. The selection can be made from a command line argument, making it possible to automate testing of multiple options
- Ranges of input should be tested.
  - **Variable Input Fields** was added to MELCOR
    - This feature allows the user to insert variable input fields into records which can then be updated through user input at the time of execution. This can be useful for testing ranges of input fields, managing variant assessment calculations, and could be used to aid interfacing MELCOR to other applications. Users can provide variable values through a variety of methods.
  - Values outside of reasonable input should be tested for meaningful error messages



# MELCOR Automated Build & Testing

- **Automated Build**
  - [Automated Build History](#)
- **Automated Testing**
  - Test Suite run on a distributed computer cluster
    - 10 Dell PowerEdge 2950s (40 runs)
      - 2 Dual Core, Hyperthreading 64-bit Xeon Processors, with 2MB cache, running at 3 GHz
    - DEF (Distributed Executive Framework)
      - App for launching jobs cluster
      - Command line execution for integration with other apps
  - [HTML output](#)
  - Regression Reports
- **Test Cases**
  - Standard test cases chosen for physics coverage ~14 test cases
    - New cases will be added as validation calculations are run
    - Debug & optimized versions tested
    - Unix versions not tested as frequently (will test more frequently in future)
  - Special purpose cases to address particular bug issues



Case	BUR	CAV	CF	COR	CVH	DCH	FCL	FDI	FL	HS	NGO	PAR	RN	SPR
M-8-1 NoMix			X		X				X	X	X			
M-8-1 SYM			X		X				X	X	X			
Lace7			X		X	X			X	X	X		X	
Lace8			X		X	X			X	X	X		X	
Vanam-M3			X		X				X	X	X		X	
Molten Salt			X	X	X				X	X	X			
PHEBUS-B9			X	X	X				X	X	X			
FPT1			X	X	X	X			X	X	X		X	
LOFT			X	X	X	X			X	X	X			
Test Inew	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SURRY (LBLOCA)	X	X	X	X	X	X	X	X	X	X	X		X	X
Zion (SBO)		X	X	X	X	X	X	X	X	X	X	X	X	X
PeachBottom (SBO)	X	X	X	X	X	X			X	X	X		X	X
Grand Gulf (SBO)	X	X	X	X	X	X		X	X	X	X		X	

Table 1-1: Physics Package Coverage



# Regression Tests

- **Regression testing performed with each interim code release**
- **Standard Test Suite**
  - **Qualifies the code for particular application**
    - Analytical results
    - Using ISP or other recognized assessments
    - Baby problems
  - **Formal regression testing report (made available to users)**
  - **Review test cases**
    - Every major code release
    - Coverage testing
- **Special Purpose Testing**
  - **Regression test for each resolved bug**
    - Responsibilities of submitter (owner)
      - supply success criterion
      - review test results for success
    - Test case made inactive after three successes (still available for future testing)
  - **Regression test for new feature or enhancement**
  - **MELGEN (input processing) tests**
    - Summer student (Brad Beeny)



# MELCOR Code Regression Test Report

- Auto-generated regression test report
  - Readable and highly formatted report (PDF)
  - Auto-generation allows report for each interim code release
  - Reports to be made available to users
- Side-by-side comparison of regression test results
  - Comparison plots for two code versions
  - Event time tables
- Test suite coverage tables
- Test case dimension table
- Pedigree information

MELCOR Code Regression Test Report  
Version 2.1 (YP)

Sandia National Laboratories  
8/29/2007

This report compares results from the MELCOR regression tests for Version 2.1 (YP) against results obtained from Version 1.8.6 (YO), using the same test suite. For each member of the test suite, significant variables characterizing the calculation have been plotted for both variants and displayed side-by-side on the same page. These calculations were performed at SNL as part of our general Y2V practices and were chosen to represent a wide, though not exhaustive, range of reactor safety phenomena. Many of these cases are simulations of both separate effects tests as well as integral tests and many are International Standard Problems (ISP) that are characterized by well-qualified test measurements. These benchmarks are provided for verification purposes only, to assess changes between code versions, and may not necessarily represent the best simulation of a given benchmark. Input and output files have been archived and can be made public for users' benefit.

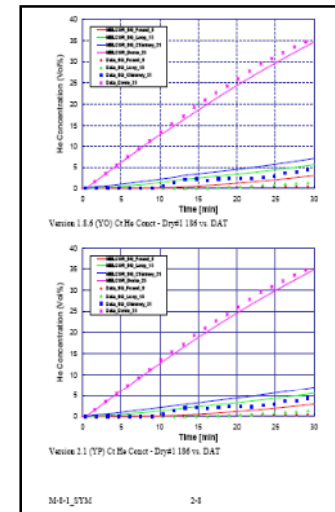
This report was auto-generated by PTFREAD (Version 1.70)

Case	#CVs	#ELs	#ESs	#CAVs	#NCGs	#Changes	NEAD	NAXL	Tend (sec)	CPU (sec)
M-8-1 NoMix	35	80	124	6	0	0	0	1800	282	
M-8-1 SYM	36	63	124	6	0	0	0	1800	263	
Lacc7	3	2	6	5	16	0	0	342011	446	
Lacc8	3	2	6	5	16	0	0	342011	487	
Vnam-M3	13	13	56	5	16	0	0	108000	13788	
Molten Spill	15	3	5	10	16	3	6	3500	2054	
PHIBUS-BP	5	4	13	9	16	3	12	18000	1141	
FP11	31	29	68	9	16	2	31	30000	206257	
LOFT	20	30	56	9	0	3	16	1800	594	
Test Issue	8	11	12	1	9	16	3	10	24950	1120
SURRY (LBLOCA)	143	265	324	1	9	19	6	17	43170	404158
Zion (SBO)	123	209	220	1	9	16	6	19	36000	138786
SwachBonum (SBO)	123	234	143	3	9	16	6	17	86400	185090
Grand Gulf (SBO)	112	224	145	1	9	16	6	17	5800	72393

Table 1-2: Calculation Size (Dimensions)

Event	Version 1.8.6 (YO)	Version 1.86 (YP)
Time (sec)	Time (sec)	Time (sec)
Gap Release - Run 1	12240.28	12255.43
Gap Release - Run 2	12380.07	12372.34
Gap Release - Run 3	12500.47	12480.38
Gap Release - Run 4	12800.27	12785.37
Gap Release - Run 5	13280.07	13278.2
Support Failure - Run 1	13977.66	13970.94
Support Failure - Run 2	14095.5	14090.32
Support Failure - Run 3	15285.52	15270.52
Support Failure - Run 4	15357.34	15330.22
Support Failure - Run 5	21716.79	21712.30
Penetration Failure - by LH Temperature - 1	25890.57	25870.42
Penetration Failure - by LH Temperature - 2	25717.43	25710.50
Penetration Failure - by LH Temperature - 3	25405.02	25403.14

Table 1-3: Event Time Comparison





# MELZILLA Bug Reporting

- Bugzilla site for bug reporting, tracking, and information
  - Available from [SNL web page](#)
- Users submit bugs and details
  - OS, Hardware, affected packages, severity
  - Bug description
  - Attachments
- Comments and attachments can be marked private and not visible to other users
- Utilized more by MELCOR community
  - 90 bugs (M1.8.6) reported last 12 mos.
  - 68 bugs (M1.8.6) resolved
  - 22 bugs (M1.8.6) unresolved

B Sandia National Laboratories
 Bugzilla – Bug 73 Editing of properties of SH component Last modified: 2007-08-28 10:45:57

Home New Search  Find Reports My Requests Preferences Log out llhumph@sandia.gov

**MELZILLA Bug-Tracking System for MELCOR.**

**Bug List:** (42 of 42) [First](#) [Last](#) [Prev](#) [Next](#) [Show last search results](#)

**Details**

**Summary:** Editing of properties of SH component

**Bug#:** 73

**Hardware:** PC

**OS:** Windows

**Product:** MELCOR 1.86

**Version:** unspecified

**Component:** COR

**Priority:** P5

**Status:** NEW

**Severity:** minor

**Resolution:**

**URL:**

**Whiteboard:**

**Depends on:**

**Blocks:**

[Show dependency tree](#) - [Show dependency graph](#)

**People**

**Reporter:** Randy Cole <rkcole@sandia.gov>

**Assigned To:** Larry Humphries <llhumph@sandia.gov>

**QA Contact:**

**Add CC:**

John Reynolds <jreyno@sandia.gov>  
Katherine McFadden <klmcfat@sandia.gov>  
Larry Humphries <llhumph@sandia.gov>  
William A. Kelly <wakelly@sandia.gov>

Orig. Est.	Current Est.	Hours Worked	Hours Left	%Complete	Gain	Deadline
4.0	4.0	0.0 + 0.0	4.0	0	0.0	(YYYY-MM-DD)

[Summarize time \(including time for bugs blocking this bug\)](#)

**Attachments**

[Add an attachment](#) (proposed patch, testcase, etc.)

**Additional Comments:**  Private

**Related actions**

- [View Bug Activity](#)
- [Format For Printing](#)
- [XML](#)
- [Clone This Bug](#)

Add Larry Humphries <llhumph@sandia.gov> to CC list

Leave as **NEW**  
 Accept bug (change status to **ASSIGNED**)  
 Resolve bug, changing **resolution** to **FIXED**  
 Mark the bug as duplicate of bug #   
 Reassign bug to Larry Humphries <llhumph@sandia.gov>  
 Reassign bug to default assignee and QA contact, and add Default CC of selected component

**Description:** [\[reply\]](#)

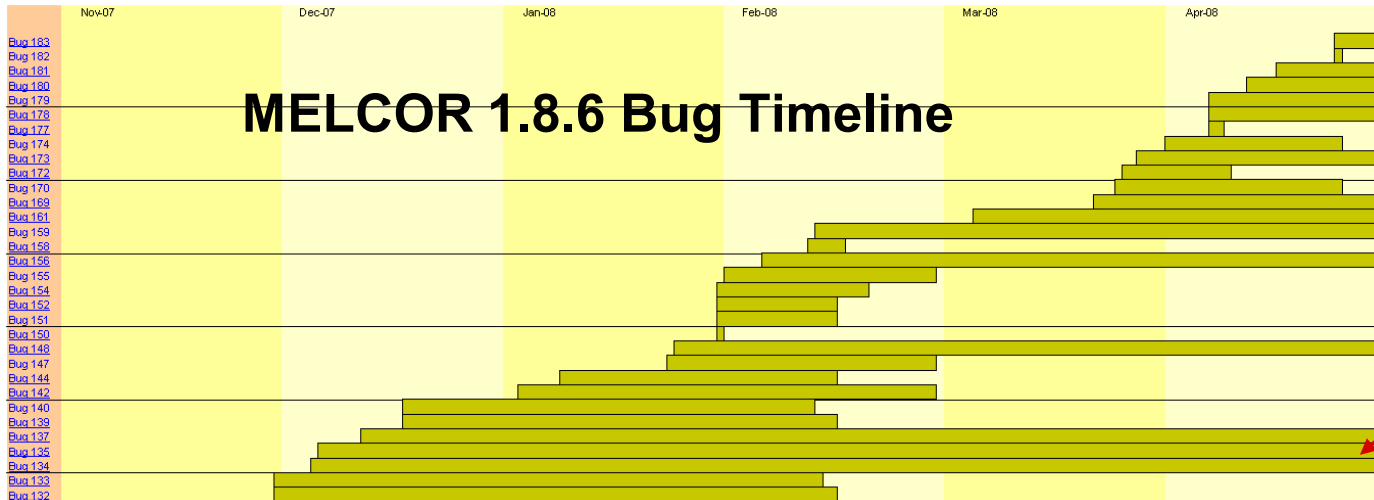
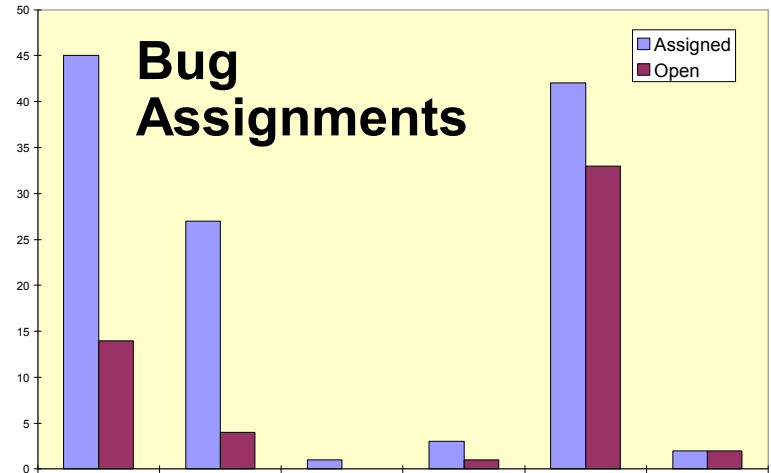
Private

**Opened:** 2007-08-28 10:45

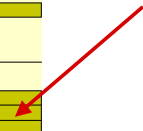


# Bugs Tracked in MELZILLA

- Automated applications for reporting issue management
  - Integrated with MSOffice, Bugzilla, and internal Wiki
  - Charts of open/assigned issues
    - Leveling bugs with developers availability
  - Timeline showing issues entered into bugzilla
    - Trying to reduce the length of time a bug is left open
    - Some issues are harder to resolve than others



**Need to eliminate bugs that are unresolved for more than a month**





# MELCOR Assessments

- **MELCOR 2.1 Assessment Matrix**

- Over 70 calculations currently in the assessment matrix
- Performed by Sandia National Labs and Russian Academy of Science
- Results to be published with MELCOR 2.1 release

- **Test calculations cover a broad range of phenomenon**

- Core uncovering (i.e., TMI-2 and LOFT-FP2),
- Core damage/melt relocation (TMI-2, PHEBUS, LOFT-FP2)
- Core to upper plenum natural circulation (TMI-2)
- Natural circulation within SG tubes (i.e. IIST)
- Core to lower plenum natural circulation (i.e., BACCHUS tests)
- Lower head failure (i.e., LHF & OLHF)
- FP release (i.e., ORNL HI & VI, VERCORS, PHEBUS)
- Containment Thermal Hydraulics (i.e., CVTR, HDR, NUPEC, Marviken blowdown, CSTF ice condenser, PANDA etc.)
- Aerosol Deposition (i.e., ABCOVE, SUPRA pool scrubbing, DEMONA, etc.)

- **MELCOR 2.1 Assessment Matrix**

- **Example Experiments**

- **Phebus**
- **Quench**
- **OLHF/LHF**
- **RASPLAV**
- **LOFT**
- **PANDA**
- **FLECHT/SEASET**
- **CORA13**
- **LACE**
- **NUPEC**

- **Three-Mile Island**



# MELCOR Documentation

- **Reference Manual**
  - Document describing physical modeling and algorithms
- **User Guide**
  - Document describing MELCOR input
- **Assessment Report**
  - Sample problems with input and discussion of output
  - Many assessment tests are evaluated
- **Documentation reviewed at major code release**
  - However, changes to documentation is ongoing
- **Management of document changes**
  - **Wiki** contains new model descriptions as well as descriptions of input and output
    - Undergoes initial review
    - Model description accessible from one place
  - **Documentation stored on SVN repository**
    - **Separate Word document for each MELCOR package**
      - PDF of entire User Guide & Reference Manual available from [wiki](#)
        - » PDF is auto-generated with bookmarks
    - **Document versions can be directly compared**



# MELCOR Web Page

- Downloads
  - MELCOR executables
  - Converter executables
  - 2.1 GUI
  - Documentation
  - PTFREAD
  - Change Documents
- Workshop / CSARP information
- Bug reporting
- FAQs
- More? Regression reports, user wiki...



## MELCOR Online: Downloading MELCOR

- **Account login is your email address**
- **All authorized MELCOR users are given an account**
  - First time users must receive authorization from USNRC
- **Access from MELCOR website or go directly to <https://melcor.sandia.gov/MelcorDownloads/MelcorDownloadStart.aspx>**
- **You can follow its hierarchical links to access every file on the system**

### Current News as of September 1, 2008

MELCOR 2.1 has been released. [View Change Document.](#)

To download the latest version, click [here](#).

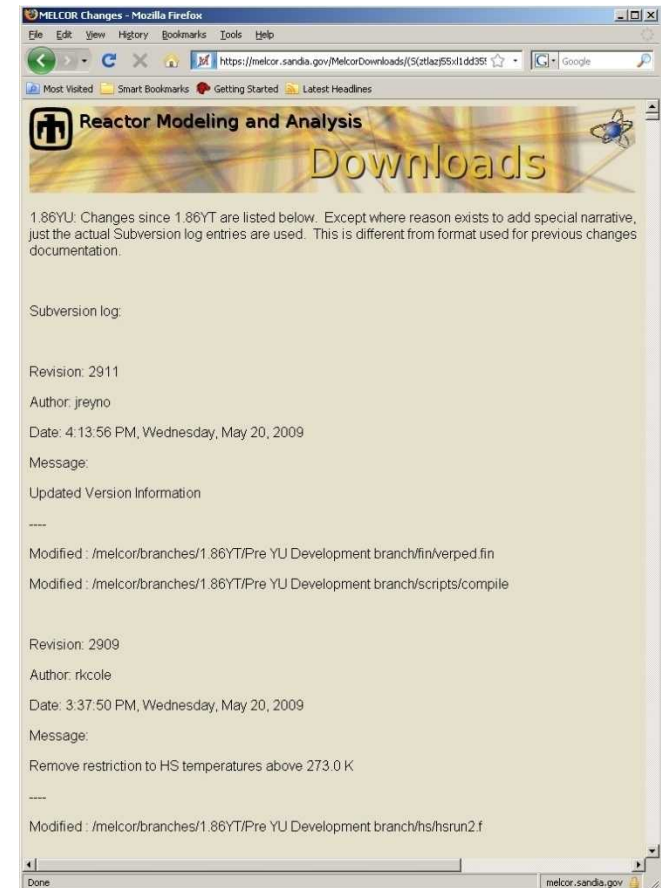
Click on the appropriate link for the [User's Guide](#) and [Reference Manual](#).

### Other Options

[Main Menu](#)

[Old News](#)

To quickly download the latest version of MELCOR, just click the “Latest News” link to see a page like this:





# Strong MELCOR User Community

- **CSARP & MCAP meetings**
  - September 14, 2009
  - Bethesda, MD
- **Workshops (2001, 2002, 2005, 2006, 2008)**



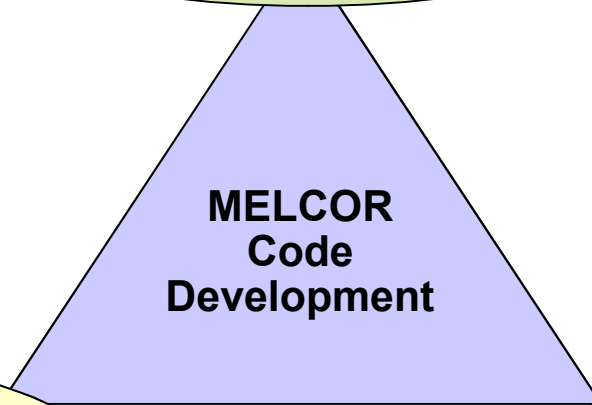
- **European MELCOR User Group (2009)**



# Summary

## Code reliability

- ◆ Validation
- ◆ QA
- ◆ Numerical stability



## User Utilities

- ◆ Converter
- ◆ HTML Output
- ◆ Etc.

## Code Enhancements

- ◆ New/improved modeling
- ◆ Code performance