

CM Generations and a Vision for the Future

Copyright Information

CM Generations and a Vision for the Future

CM+ is a trademark of Neuma Technology Inc.

Neuma Technology provides this document "as is" without warranty of any kind, either expressed or implied, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. Neuma reserves the right to make changes to the product described in this document at any time and without prior notice.

CM Generations and a Vision for the Future

by: Neuma Technology Inc.
#51 - 5450 Canotek Rd.
Ottawa, ON K1J 9G3
Canada

Tel: (613) 749-9450

<http://www.neuma.com>
email: sales@neuma.com

©2004 Neuma Technology Inc. All rights reserved.

Printed in Canada

DEC VAX, DEC VMS, DEC ALPHA are trademarks of Compaq Computer Corporation. Intel is a registered trademark of Intel Corporation. Microsoft Windows, Microsoft Windows XP, Microsoft Windows 2000, Microsoft Windows NT, Microsoft Windows 98, Microsoft Windows 95 and MS-DOS are trademarks of Microsoft Corporation. Sun/Solaris is a trademark of Sun Microsystems, Inc. UNIX is a registered trademark, licensed exclusively through X/Open Company, Ltd. UNIX and XWindow System are registered trademarks of X/Open Company Ltd. PostScript is a registered trademark of Adobe Systems Incorporated.

About Neuma Technology Inc.

Neuma Technology Inc. provides an integrated tool suite that helps manage the automation of the software development lifecycle. Customers can rely on Neuma's wealth of experience and understanding of the complexities of software lifecycle management to assist them in delivering quality products to their markets.

The company's flagship product, CM+ is a high-performance Software Configuration Management System which provides an automated environment for the management of quality software projects. Based on a process-oriented database and workflow technology, CM+ provides reliable configuration management capabilities, within a tightly integrated set of applications. Unique approaches to managing software releases and change packages, and a fully scaleable suite of applications differentiates CM+.

The integrated applications of CM+ include:

- ◆ Version Control
- ◆ Change Control
- ◆ Configuration Management
- ◆ Build and Release Support
- ◆ Problem Tracking
- ◆ Activity Tracking
- ◆ Requirements Management
- ◆ Document Management

Contacting Neuma Technology Inc.

For additional information, e-mail Neuma at support@neuma.com or visit our Web site at www.neuma.com.

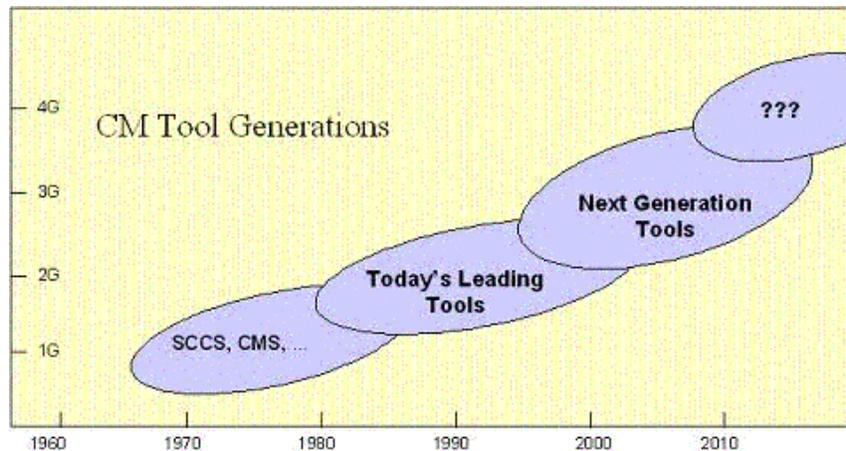
This article was originally published in the January 2005 issue of *CM Crossroads Journal*.

CM Generations and a Vision for the Future

It was 1978 when I first introduced change packages (a.k.a. Updates) as the central feature of an in-house CM system - still in use today supporting a 40 million LOC project. It was 1982 when I introduced the concept of streams to rationalize branching along the product roadmap. Little did I know that a quarter century later, these concepts would just be starting to move to the forefront.

By my reckoning, the past 15 to 20 years of CM technology and process advances have produced, what I would call, the second generation of CM and many strides have been made by various parties toward the third generation. Over the remainder of the decade, we'll see some CM vendors bring forth their third generation solutions - indeed one or two already have. The fourth generation lies beyond that.

The CM industry is always in need of some vision, set forth by the innovators, the practitioners and the user-community. As with most technology advances, CM will only reach the bar we set for it. How do you recognize next generation tools, that is, apart from the marketing hype? And what will the CM landscape look like over the next 2 generations?



If you're building a Missile Defense System, a new generation of Aircraft, or a tamper-proof Financial System, you'll want to position yourself to best take advantage of new technology. Lets take a brief look at the past and present to prepare us to look into the future.

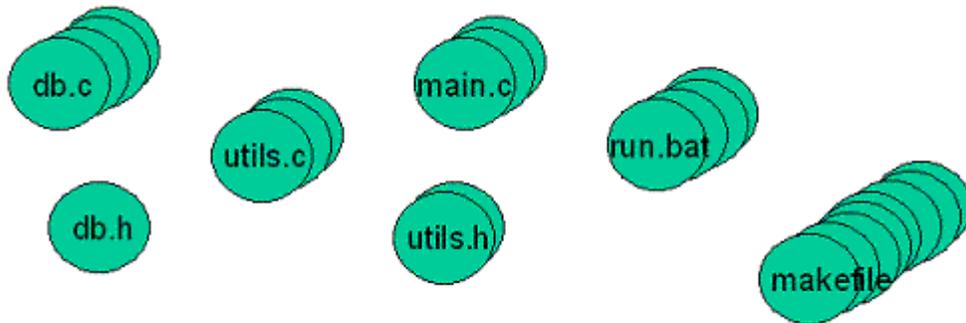
First Generation CM

The first generation of Configuration Management tools dates back to the late 1960s and the 1970s, with SCCS (Unix), then RCS and a number of mainframe capabilities such as CMS (VMS). Through the 1970s, CM began to take a formal foothold. Most solutions were still in-house solutions, but usually based on a basic version control program such as SCCS.

The initial set of CM tools provided the most basic Version Control and Build facilities. These were typically a collection of tools, sometimes formally grouped, sometimes informally grouped. First Generation CM tools are identified by the following properties.

- ◆ Configuration Item/Unit Identification, including Revision identification schemes
- ◆ Check-out and Check-in of files, with exclusive check-out support

- ◆ Basic Branching and Branch Identification capability
- ◆ The ability to retrieve a file without reserving it for check-out
- ◆ The ability to compare revisions of a file and perform basic merge operations
- ◆ The ability to define and capture forever a baseline definition
- ◆ Basic Integration with basic Build/Make tools
- ◆ Some level of scripting support, perhaps through the OS environment



1st Generation CM: manage revisions, checkouts & simple branching

First generation tools were not user-friendly. Developers were used to the command-line and a few extra commands to safeguard their source code was not a big deal, as long as the system was reliable. The CM manager's job was anything but defined. The one clear goal was to be able to produce nightly/weekly builds and to be able to reproduce them as necessary.

Second Generation CM

During the 1980s and 1990s, ideas within the CM community pushed capabilities through a second generation (2G) with some key advances. One of the key advances, still being absorbed by the some CM vendors and consultants, is the concept of managing changes rather than files. Most other advances dealt with optimization of the basic CM capabilities. 2G CM advances include:

- ◆ Change packaging and basic change management
- ◆ Revisioning of directory structure
- ◆ Support for semi-automatic Baseline definition
- ◆ Sharing of Revisions across Development Streams/Releases/Branches
- ◆ Support for Makefiles and build scripting
- ◆ Source Code Bulk-Loading Capability, usually through scripting
- ◆ Exclusive and Parallel Checkout Support
- ◆ Branch and Label Management or equivalent.
- ◆ Distributed Build Capability, especially since computers were significantly slower
- ◆ Integration with SCC API Compliant IDEs, late in the '90s
- ◆ Basic workspace support

In the second generation, CM advances were not limited to CM tool capabilities. Perhaps even more so, focus was placed on other CM issues: Process, Useability, Reliability and Administration and Performance issues. Tool administration saw two key advances

- ◆ Concurrent Unix and Windows platform support

- ◆ Scalability of the solution to support hundreds of users

From a reliability perspective, the key item was the ability to maintain data security for the CM repository. Rules for when and what could be accessed gave way to automatic enforcement and some tailoring of the rules. The repository had to be safe, especially from the development team itself - from new users, from finger problems, etc.

From a Process perspective, the concepts of traceability began to take hold, as the CMM and Total Quality concepts began to emerge. Process-oriented advances for a 2G CM tool include:

- ◆ Integrated Problem (or Issue) and/or Task Tracking
- ◆ Traceability between the files changed and the tracked problems and activities/tasks
- ◆ Basic customization capabilities, typically through scripting
- ◆ Basic Rules and Triggers capabilities to allow some workflow automation, traceability and communication
- ◆ Introduction of a state-based promotion model for changes

The remaining key area of 2G CM tool progress dealt with improvements to the usability, primarily through a graphical user interface (GUI). Advances in this area include:

- ◆ Provision of a GUI for all basic end-user CM functions
- ◆ The ability to navigate the source tree and file history graphically
- ◆ Ability to look at a consistent collection of related files by specifying a Context and/or View
- ◆ The addition of Basic Reporting Capabilities, especially for Changes and Problems/Tasks
- ◆ Improved presentation for File Compares and Merges
- ◆ The addition of Remote Access Capabilities

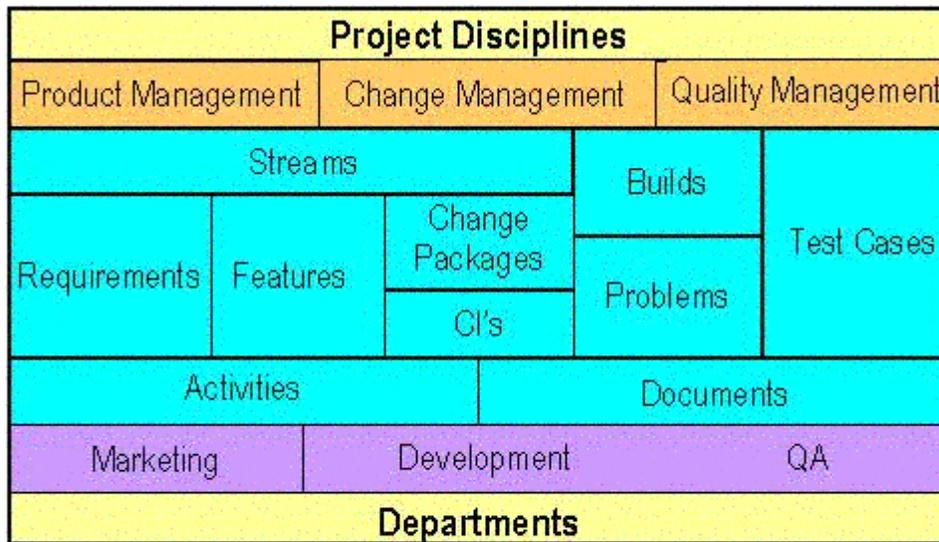
This set of capabilities describes, for the most part, the 2nd generation of CM tools. Although some of the freeware tools are still properly categorized as 1G CM tools, 2G CM tools account for a large chunk of today's CM marketplace. They combine CM Capabilities with Process Capabilities and Friendly User Interfaces to support small to medium size or even large projects (several hundred users). Today, many tools have begun to reach further towards the Third Generation.

The Future Begins Now: Third Generation CM

The next generation of CM tools has been evolving through the 1990s and the early years of the new millennium. Third Generation (3G) CM tools include significant advances over their 2G counterparts. Although there are very few tools which might deserve to be called 3G tools, any tool that meets more than 80% of the 2G criteria is bound to meet at least a small percentage to 3G criteria.

It's more difficult to categorize 3G CM Tool criteria. Why? Because technology will change before there are many 3G tools. Will 64-bit support and Instant Messaging be important CM factors? This is a bit difficult to predict. What about basic advances in repository technology such as journalling and backup? As data sizes continue to grow, so must advances in repository technology.

The 3G CM tool criteria are more evenly divided along the five key areas: CM Capabilities, Administration and Performance, CM Process, Reliability and Availability, and Usability.



Third Generation CM: a single unified solution.

3G CM Capabilities

3G CM tools must focus not simply on doing more, but doing so more simply - better supporting agile development. Workspace Management, Build and Release Management, and minimization of Branching, Merging and Labelling are key areas that are being addressed. Key advances include:

- ◆ Easy bulk loading of source from the user's workspace
- ◆ Support bulk loading of multiple revisions of a baseline or subsystem
- ◆ Formal support of development Streams and Releases
- ◆ Introduction of first class objects to eliminate labelling and minimize branching and merging
- ◆ Automated stream-based branching
- ◆ Change promotion replacing promotion branches, and driving the CM process
- ◆ Automated Build, ANT and Makefile Generation
- ◆ Synchronization of user workspace with a context view
- ◆ Interactive comparisons of Builds and/or Releases - Traceability Data and Code
- ◆ Queued Checkout to support the Exclusive Checkout Model
- ◆ Integration with other common IDE APIs and perhaps emergence of a generic Change-based IDE API.

In this generation we start to see new capabilities, perhaps a few that we question or simply don't understand. Yet side by side we see capabilities that have been around in some tools for ages (e.g. IBM ClearCase's Integration of Views with the OS File System; Neuma's CM+ Change-based promotion). Still many questions and concerns: Can I really

eliminate labelling and why do I want to? How can we reduce branching and merging without compromising our CM process? It can't be done! - can it?

The key to developing successful 3G tool capabilities is first to recognize why different CM patterns and strategies are used, and then to automate these. 2G tools provided a lot of new capabilities, but these were not properly guided. Triggers, labels, branches, permissions, views - all fine, but all requiring significant guidance. To develop a 3G tool, a new mindset is required. Although the capability is important, the use of the capabilities is more so. In a 2G world, every shop decides how it's going to do branching, labelling, views, etc. In a 3G world, the tool determines how and guides you based on your required use cases, and then allows you to customize.

First class objects are crucial for items such as changes, baselines, builds, and branches. It's no longer sufficient to identify a change as all file revisions which referenced a particular task. Change workflow is not the same as task workflow. It's insufficient to identify a build by all the revisions with a particular label. A build needs its own workflow, and needs to be promoted through integration, testing and perhaps the release process. Branches are not just another way of numbering a revision, they have a number of properties peculiar to them which do not need to be associated with revisions.

Use of an appropriate set of first class objects eliminates so many complexities that the configuration management task can be automated and replaced with a change management task: What changes are we promoting today? Which changes do I need to include this feature in my build? Which changes fix the "gating" problems for the next beta release? These decisions replace CM tasks such as: How do I label these files to promote them and do I have to do merges? Do I have to label all of the files for this build or can I use a set of labels to specify it, and add a label for the new changes?

Tedious planning and setup tasks should also be left to the CM tool. What is our branching strategy and how should it change as we begin work on the next major release? How do I define my configuration view to work on a particular build or development stream? Yes, these all appear to be valid questions and concerns, but in a 3G CM tool, the concern is left for the CM tool, while developers and CM managers focus more on change management. The key shift is to guided automation and this will grow over the next two or three generations of CM tools.

Change the first-class objects and the developer's life is made easier too. A developer can select a product and development stream, or perhaps a specific build identifier, rather than defining or using configuration view. Once this context is established it can be modified by including changes (i.e. specific change packages), if necessary, or by specifying a specific test/promotion level the user wishes to work to - the latest build level instead of the most recently submitted files, for example. The developer doesn't worry about when to branch, the CM tool will announce when branching is required. Changes are promoted and the CM tool identifies if there are any dependant changes that also need promotion. The CM tool identifies the alignment of revisions for a baseline or for a user context - no need to do branching and labelling for this, as the change packages contain all the necessary information.

3G Administration and Performance

A 3G tool, even though it is more complex than a 2G tool, must bring down the level of administration, significantly. And it must permit the prospective customer to perform a rapid evaluation and roll-out of the solution without having to commit significant budget - time, infrastructure or money. Key advances for a 3G solution include:

- ◆ Low administration operation
- ◆ Full interoperability between big and little endian architectures, without loss of functionality
- ◆ Platform independent customization capabilities
- ◆ Fast roll-out, for evaluation and for production use
- ◆ Painless upgrade capability with virtually no down-time
- ◆ Basic multi-site capability for working among a few different sites
- ◆ Easy scalability to support thousands of users

Show me a CM tool that meets these criteria, and I'll show you a CM Admin team that is happy. It doesn't consume their time, they can upgrade their hardware and switch from Windows to Linux, or vice versa with no down time. It wasn't a big deal to evaluate or to put into production - if they decide they don't have the budget right now, no big deal to back it out. And they don't have to worry about infrastructure upgrades until they go over a few hundred users. Some of the users can even work off site. A few tools out there can pretty well cover these criteria, largely because of their architecture. Others are bogged down in the complexity of the infrastructure and the associated administration that goes with the infrastructure, whether hardware (e.g. highly tuned servers) or software (e.g. Database).

Process Advances

Next up is the Process advances

A 3G CM tool will have a sophisticated state-based workflow capability tightly integrated with a task-based project management capability to give a fully unified process model. A state-based workflow is not just a way of implementing rules and triggers. It provides a means of specifying states and transitions, of putting rules and triggers on each transition, of specifying the permissions and/or roles required for a transition.

Project management capabilities provide a living Work Breakdown Structure (WBS) capability where activities can be broken down into tasks and sub-tasks. When integrated with state-based workflow, some of the tasks will automatically appear in the WBS, and be assigned to the appropriate resources, because of a feature or a build reaching a specific state. Prioritized to-do lists (task lists) should drive each team member forward, especially in an agile development shop using an iterative integration approach.

Process advances for 3G tools include:

- ◆ Integrated Process Workflow Capability
- ◆ Access permissions beyond those offered by the file/network system
- ◆ A seamless integration of Configuration, Change and Product Management functionality
- ◆ Extensive and easy customization of process, user interface and data schema
- ◆ Support for end-to-end traceability, interactively and through reports. From products and requirements to builds and test cases.
- ◆ Real-time metrics to support timely decisions and process improvement
- ◆ Advanced data import, export and reporting capabilities
- ◆ Integrated Project Management with WBS (Work Breakdown Structures) Support and Planning and Projection capabilities
- ◆ Integrated Requirements Management and Test Suite Management with Traceability

3G Process affects a number of management applications encompassing a significant portion of the development cycle. In a 3G CM tool, it is not sufficient to glue together these applications. They must be fully integrated, seamlessly, so that the user is not aware of multiple management tool interfaces or the delays which occur by having queries which span multiple repositories.

It's also imperative that the processes reflect the customer's processes. Yet the customer does not want to be burdened with defining the correct process up front before using the CM tool. In fact, the customer won't understand the CM problem domain until well along the development and release road map. So it will be imperative that incremental changes can be made quickly and easily without any down time.

Yes, But Can I Get At My Data

Reliability and availability are crucial factors when deadlines are tight or teams are large. An outage of even a couple of hours can be costly. A 3G CM tool is not at the mercy of a server or network link. Nor is it acceptable that the tool presents poor performance as the number of concurrent users climb. 3G criteria includes:

- ◆ High reliability and availability (less than 24 hrs outage/year including scheduled upgrades)
- ◆ Full transaction journaling and data recovery capabilities
- ◆ Availability of on-line user forums and support centers
- ◆ Advanced Backup and/or Redundancy Capabilities

"Smart client" technology and other such techniques are used to off-load the server and network. Local caching and, where necessary, data replication are used to ensure point-and-click performance capabilities. Data redundancy, transaction journaling and automated data recovery capabilities ensure that outages do not occur across the entire development team, and can be recovered from in minutes.

As CM repositories grow in size, and with a high transaction rate, advanced backup capabilities are required which require no down time and which are quick to backup and easy to restore. It is not feasible to do nightly backups of hundreds of gigabytes of project data. Nor is it reasonable to require dozens of incremental backups to be retrieved before a project's data can be restored in response to a disk crash. Reasonable disaster recovery support must be provided, not just by the IT department, but by the CM tool itself, as it is now a key component of the IT strategy.

It Keeps Getting Better

The thing about 3G CM is that, although it is handling an ever increasingly complex set of problems and capabilities, it is easier to use, and user acceptance becomes much less of an issue than with a 2G tool. What are the barriers? Is it slow, complex, non-intuitive, insufficiently flexible? Key criteria in the area of Useability include:

- ◆ Accessibility from all major platforms, especially Unix and Windows
- ◆ Rapid performance supporting point-and-click navigation of data
- ◆ Flexible reporting and interactive query, with well-designed pre-packaged reports
- ◆ Support for advanced merge capabilities such as change propagation/yank and product-wide operations
- ◆ Provision of context-sensitive on-line help, preferably in the form of process guidance

- ◆ User-selectable differencing and merge tools
- ◆ Good remote access performance using IP-based protocols (not requiring disk mounting)
- ◆ Operation across multiple sites

A 3G CM tool must support, at a minimum, 3 types of user interfaces: Unix, Windows and Web. Ideally, these are all the same or at least very similar in look and feel. In particular, Web based access should bridge legacy platforms where native interfaces are otherwise unavailable. And please don't tell me that I have to learn a new interface because I've moved from one platform to another.

The 3G tool provides good management level reports: metrics, change summaries, release note support, SVD/VDD generation. Reports are needed which are specific to each area of the tool: requirements, project management, problem tracking, change management, build and release management, test suite management, and perhaps other areas. Even more important is the need for good interactive queries. Not just single point data retrievals, but multi-axis summaries which can easily be drilled into for more detail. How about graphs which are interactive so that they too can be queried - I really want to know what is making that bar of the chart so big. This is the generation where paper and softcopy reports largely give way to interactive queries. Why? Because they are more flexible - you choose what you want to see on the fly. You change data as you identify the need, for example, to raise the priority of a problem. Instead of coming away from your Change Control Board (CCB) meetings with a set of actions, implement the actions at the CCB and interactively show the results. Use interactive metrics to learn about your project behaviour, to identify your process problems and successes, or just to satisfy your interests.

Interactive query will not catch on unless you have point-and-click performance. Technology advances must be incorporated into a 3G tool so that the computer is waiting for the user, not vice versa. Otherwise, the benefits of the tool will accrue only to the very patient, and perhaps not for long. This applies to both native interfaces and web interfaces.

Data navigation capabilities, such as source trees, work breakdown structure trees, etc. are a minimum level of data access capability that must be provided, along with the means to navigate traceability information in an easy manner.

The big element of useability in the user interface must be object-oriented. You have a report - click on an object to change it, to promote it, to check it out, etc. Basic GUI style guidelines which did not exist in the 1G and 2G tools, have already gone through several iterations in the past few years. These must be harnessed into the CM tool, but even more so, the CM tool must present new ways of navigating data history, traceability information, and metrics.

The Fourth Generation CM Tool

Early adoptors of Fourth Generation (4G) CM tools will be those who are dealing with long term projects. It's to your advantage to understand now what a 4G system will do for you. Here's why.

- ◆ It's beneficial to know where the industry is going before establishing your corporate architecture. Perhaps you're looking at going with a market leader; maybe you're looking at an Open Source solution. Understanding 4G technology will help you to identify your longer term ongoing costs - costs which 4G CM technology will likely slash.

- ◆ Although there are no commercial 4G systems today, many of the 4G capabilities can be found across today's commercial CM landscape. Some examples: a few newer tools (and some not so new) have very small footprints - without sacrificing functionality; IBM's ClearCase has long supported a view of revisions integrated with the file system; Neuma's CM+ already supports "Fully Synchronous MultiSite" and "Checkpoint/Recovery" capabilities.
- ◆ At least one tool vendor is expected to release a 4G version of its CM tool as early as next year!

If nothing else, early feature availability, and perhaps an early 4G tool, will help do two things: educate the market place; and help form requirements for CM process definition and tool acquisition.

3G CM tools significantly cut the cost of CM when compared to 2G tools. 4G tools do the same, just differently. Although there's more mileage to be found by 4G systems in improving useability and reducing administration and infrastructure requirements, much of the cost savings will come from expanding the scope of functionality.

Built-in disaster recovery will augment and help to simplify IT department plans. Improved scalability of client to server ratios will reduce server complexity and costs. Integration of Resource Management and Customer Tracking capabilities will help reduce overall tool costs, and will take the tool integration burden off the back of the customer.

What else can we expect to see in the 4G timeframe? Some of the technology directions are wildcards. But we have enough to put generic road map together. The most likely difficulty here is separating 3G and 4G criteria.

4G CM Capabilities

CM capabilities will support more natural ways of working. The end-user will no longer need to be trained on how to do something. Training will consist more of what you can do - and the tool will help you do it. Expect to see most of these capabilities in a 4G CM tool:

- ◆ Drag and Drop Bulk Loading of Source and of Multiple Baseline Revisions
- ◆ Configuration Management Full Automated, Giving way to Product and Change Management
- ◆ Active Workspace Management - the tool keeps you informed
- ◆ Promotable directory structure changes - no more directory checkouts or strict ordering of structure changes
- ◆ Integration of Views with the OS File System - like ClearCase has been doing for years
- ◆ Dynamic Variants - a subset of changes will be managed as persistent changes on top of your view to support your variant
- ◆ Product/Sub-product management - look at/work on a product from a product or sub-product perspective
- ◆ Context view-based dependency analysis with architectural layering and partitioning support
- ◆ Automatic Change Generation based on changes made in a developer's workspace
- ◆ File revisioning augmented with full data revisioning

4G CM Administration

Administration will support flexibility within the IT environment. Fewer server issues, fewer technology migration issues, fewer data backup and recovery issues. Look for:

- ◆ Small Footprint Solutions - without sacrificing functionality
- ◆ Zero Administration Operation
- ◆ Scalability to thousands of users per server, allowing a single server per site
- ◆ Fully Synchronous MultiSite - the same view from any site at any time
- ◆ Full interoperability between 32- and 64-bit platforms
- ◆ Automated checkpoint backups - ensuring backups happen even if when they don't

4G CM Process

The CM process capabilities will be extended to support a dynamically configurable unified process. When additional data or sister application support is required, the integrated RAD capability will allow the application set to be expanded easily, with seamless integration, using a common repository and a common user interface.

Quality support will be pre-packaged in the CM tool - change and revision control of individual requirements, a clear, key set of metrics, and project forecasting capabilities. These will give management a clear idea of what's happening and how likely they are to meet their goals in the specified time frames. Planning activities and tasks will move from the realm of tracking data into the realm of deliverables. A task will grow from a line item and a description of deliverables/objectives, into the (unrefined) user documentation and test plans - automatically tied directly to the task. Here are some of the key process advances we'll see in 4G CM:

- ◆ Advanced workflow capabilities - integrating state-based and task-based workflow
- ◆ Dynamically Configurable Unified Process - incremental changes will be automatically reflected in on-line guidance
- ◆ Integrated RAD Capability to extend the set of integrated applications
- ◆ Change and revision control of requirement items
- ◆ Project and Quality Metrics and Forecasting Capabilities
- ◆ Evolution of Planning Activity/Tasks into Deliverables
- ◆ Resource management
- ◆ End-to-end impact analysis
- ◆ Test Run Management and Metrics
- ◆ Data Management including Data Security
- ◆ Customer Tracking Capabilities

4G Reliability and Availability

This will be one of the hottest areas of 4G CM systems growth that will enable them to grow into true organizational backbones. The goal is to keep your information available, to keep your processes running and your data safe. Look for:

- ◆ Hot-standby Disaster recovery
- ◆ Checkpointing and recovery capability

- ◆ Ultra High Reliability and Availability (99.95% or < 4 hrs per yr)
- ◆ Information Security Levels
- ◆ Recovery from Malicious/Subtle Data Corruption
- ◆ Proven Longevity of the Tool on projects older than 10 years

4G Useability

Useability will benefit from the improvements in CM functional capabilities. But beyond that, senior levels of management will begin to rely on the CM tools. The CM tools will provide the high levels of reporting and interactive navigating required to make decisions during a meeting. Look for:

- ◆ Extensive Reporting Formats (XML, Spreadsheet, HTML, Text, etc.)
- ◆ Advanced Interactive Browsers (Hyperdata, Tree-browse, form browse, etc)
- ◆ Executive Summary and Interactive Drill-down Capabilities
- ◆ Wide set of Standard IDE/3rd Party Tool Integrations

Beyond the Fourth Generation

Beyond the 4G CM tool, we'll see the CM centric system grow into an Enterprise Management System. Expect to see an expansion of functionality to support a wider definition of product management, including Time Sheet Management and Analysis, Budgeting, Risk Management, Integrated Quality Assessment, Web Site Management, ITIL compliant CM Deployment, HR Management and Sales Tracking, which has no other place to attach itself.

The expansion of CM tools into these areas will occur as XRAD (eXtremely RAD) capabilities are bundled with the CM capabilities. Organizations, large and small, will have the capability to rapidly define and customize their applications and processes using the same repository and tools used in their CM and product management environments.

From a useability perspective, more and more non-technical users will find that they can benefit from Change Control and Configuration Management: lawyers, accountants, media content providers, and so forth. A few simple operations from the Operating System menus (save my work, show my history, show a change, yank a change, etc.) will make this possible. The CM engine will move to a more center stage position within the OS.

Disaster Recovery will give way to Continuous Operation. Take down one site and end-users won't notice - like card and shelf redundancy in high reliability telecom and aerospace products.

So Where Are We Now

So we have painted a grand plan. Is it likely to be followed? It doesn't really matter. What matters is getting the ideas on the table so that the market place starts demanding them. This is what will ultimately drive the industry. Users are too focused on their own shops and problems to be able to paint this plan. Vendors and Consultants see dozens, if not hundreds of sites per year - they must interact with users to form this vision, and not simply to meet today's requirements.

But in the mean time, we need to understand where we are now. There are dozens of tools, and each has their strengths. With input from others, this author will attempt to refine the

definition of CM generations so that we can measure our existing tools and get a real idea of what it means to leap from one generation of CM to another. One goal will be to be able to compare CM tools more objectively.

Perhaps you disagree with some of these projections, or perhaps you see them happening along different timelines. I welcome your input.

Joe Farah is the President and CEO of Neuma Technology. Prior to co-founding Neuma in 1990, Joe was Director of Software Architecture and Technology at Mitel, and in the 1970s a Development Manager at Nortel (Bell-Northern Research) where he developed the Program Library System (PLS) still heavily in use by Nortel's largest projects. A software developer since the late 1960s, Joe holds a B.A.Sc. degree in Engineering Science from the University of Toronto. You can contact Joe at farah@neuma.com