CMPIC
www.cmpic.com

CMTrends

News and Perspectives for CM Professionals

in this issue:

January 2017

ISSUE 26

Is Location of Work or Job Complexity A CM Qualifier?

by Sarah Dye
SimVentions, Inc.

I have heard time-and-time again in the Department of Defense (DoD) industry, “You are not doing ‘real’ CM if you are not performing CM on a Government location” or “You are not doing ‘CM’ because your project is not a large project.” Does the work location or complexity of the project determine whether CM is being performed or define your value as a CM professional?

The question I initially ask those nay-sayers is, “Do you really know what CM is?” No, it’s not just putting a version number on a document or providing a disc with source code to someone for storage. Let’s take a look at how CM is defined in the standards.

According to ANSI/EIA-649: “CM is a comprehensive process for maintaining consistency of any product's performance, functional and physical attributes with its requirements, design, and operational information.”

According to CMMI: A discipline applying technical and administrative direction and surveillance to (1) identify and document the function and physical characteristics of a configuration item, (2) control changes to those characteristics, (3) record and report change processing and implementation status, and (4) certify compliance with specified requirement.

A summation of both definitions shows that CM consists of processes that are applied to ensure the product is designed and functions in accordance with the documented requirements. These processes should be created based on the foundational elements of CM: Planning, Identification, Change Management, Status Accounting, and Verification & Audit. A process without a strong foundation cannot guarantee the consistency of a product.

CM processes need to be created by the foundational elements, but does the process have to be rigorous in order for it to be substantive?

According to EIA-649: "The degree of rigor and techniques used in implementing CM is commensurate with the type of product and its application environment as defined by the program requirements."

If individuals and companies alike find themselves only applying value and weight to large and rigorous CM processes, then we need to step back and ask why we are doing CM in the first place. I believe these processes apply to all projects.

In the DoD industry, we first and foremost perform CM to ensure the warfighters are receiving the highest quality product to keep them safe. This includes small projects that hold less formal Configuration Control Board (CCB) meetings and may not use as many variance and deviation acronyms due to the size and complexity. Even on the smallest of projects, performing CM is imperative to ensuring the warfighter receives a product that is consistent with requirements.

So, I ask again my initial question, “Does the work location or complexity of the project determine whether CM is being performed or define your value as a CM professional?” The answer is “No” as long as we value the end-user and ensure the best and most recent products are tested and being delivered.

Sarah Dye is the Configuration Management Lead for SimVentions, Inc. Over the last 10 years she has performed CM on many Government projects; has created and implemented corporate CM processes utilized across the company; and, she has maintained Data Management (DM) on Government contracts.

SARAH DYE

Baseline Questions

by **Steve Easterbrook**
CMPIC LLC

I received the following questions from a fellow CM Professional.

> "Can a document that has been declared and released as part of a functional baseline, in later stages of the program, be also part of an allocated baseline or product baseline?"
>
> "Can a document that has been declared and released as part of a allocated baseline, in later stages of the program, be also part of the product baseline?"

Functional, Allocated, and Product baselines are baseline types originally defined in military standards that defined incremental customer control of the configuration from development through production. The concept can also be applied to commercial environments as well, but the terms may, or may not, differ.

My Views: From a CM perspective, the term baseline is associated with a set of work products (documents, parts, code, etc.) that is officially recognized by the organization and/or customer as the starting point for further activities.

A "**Functional**" baseline identifies the approved specification we initially agree to design to (there could be other high level documents as well).

In complex end items, an "**Allocated**" baseline represents the second level of specs that we initially agree to use to design the major components of the end item.

A "**Product**" baseline identifies the official documentation, parts, etc. that we initially agree to build to. If something is "baselined" it implies (in most cases) that these work products are under a more rigorous change control process. e.g. in some DoD contracts, any work products listed in DoD approved Functional, Allocated, and Product baselines would require DoD permission to be changed.

Although the items initially listed in baselines may change over time, the baselines do not. Again they are an initial agreement and never change. From a DoD view you don't put functional baseline items into an allocated baseline. But you can put the current versions of items from initially agreed upon functional and allocated baselines into your product baseline when it is agreed upon.

Baselines are static and do not represent the current configuration information after things change.

NOTE: Some call the CM database / tool that contains the configuration information a "baseline". A database that contains configuration information is not a baseline.

Again, a baseline is only an agreed upon starting point. Changes are then tracked against that agreed starting point.

Analogy: If I go on a diet and weigh myself on day one, I have created a baseline for future weight loss measurement. That weight, which is my "diet baseline", never changes over time. If I subtract the baselined weight from my weight two weeks later, I can determine how much weight I have lost (or gained).

I could also establish a rule that after I create my diet baseline, I cannot eat anything without filling out a FRP (Food Request Proposal) and having it reviewed and approved by my "Diet Control Board" (DCB).

Steve is the President of CMPIC LLC - the Configuration Management Process Improvement Center. Steve has been working in configuration management for over 30 years. He has 12 years of experience as a Configuration Management manager in government and commercial organizations and another 18 years as a CM educator, lead assessor, and consultant. Steve has taught, lectured to, and consulted with thousands of individuals from hundreds of commercial and government organizations on the subject of CM process improvement.

STEVE EASTERBROOK

ANNOUNCING

CMTrends

2017 S.W.A.T.

Seminars, Workshops, And Training

Orlando, Florida

28 - 30 August 2017

SAVE THE DATE

Rosen Centre Hotel

9840 International Drive

Orlando, FL 32819

REGISTER HERE

CMTrends

2017 S.W.A.T.

Seminars, Workshops, And Training

You are Invited!

August 28 - 30, 2017 in Orlando, Florida

You are invited to learn about the latest configuration management process improvement techniques at **CM Trends 2017: Seminars, Workshops, and Training**!

This event is intended to put you in touch with a dedicated group of professionals working to improve their CM Processes. 30-minute lectures and 1-hour long workshops will teach you new skills, perspecitives, techniques, and about the direction CM is headed. You will also have ample time to speak with PLM / CM tool exhibitors, participate in Q&A sessions with the speakers, and network with your peers.

Learn more about CM Trends 2017 at: www.CMPIC.com/configuration-management-seminar

Register now: **www.CMPIC.com/registration**, **1-434-525-8648**, or **info@CMPIC.com**

Discounted Post-Event Classes

August 30 - September 1, 2017 in Orlando, Florida

Every year CMPIC hosts discounted CM certification and training courses to celebrate a successful CM Trends event. These courses (typically $1295) will be offered at $995 to the public or only $800 in addition to your CM Trends 2017 registration. Register now to learn more about one of the following:

Course 6, "SAE EIA-649B Principles & Applications" certification class
Course 7, "Configuration Management Assessor" certification class
Course 9, "CM Standards & Practices Update" refresher class

Learn more at: www.CMPIC.com/2017_CMTrends_Classes
Download our Justification Letter to attend.

Register Now

CMTrends

2017 S.W.A.T.

Seminars, Workshops, And Training

Location: Venue & Hotel

All CM Trends 2017 Seminars, Workshops, and Training classes will be held at:

ROSEN CENTRE HOTEL
9840 International Dr.
Orlando, Florida 32819
www.RosenCentre.com

The Rosen Centre Hotel is located on the famous International Drive, a quick walk to shopping, dining, and entertainment.

Room Rate: $114.00 (single/double) per night, available August 23 - September 5, 2017.
Make your sleeping room reservations soon. Room block typically fills up early. Email kerri@cmpic.com if you encounter any problems making your sleeping room reservation.

To reserve a sleeping room at this discounted group rate, please:
Call: 1-800-204-7234 & mention "CMPIC: CM Trends", *or*
Reserve Online: https://bookings.ihotelier.com/Rosen-Centre/bookings.jsp?themeId=18677&groupID=1643718&hotelID=2018

*** This group discounted rate and room block will expire on Friday, July 28, 2017.** Reservations made after July 28th will be on a space-available basis at the hotel's current selling rate.

CHECK IN/CHECK OUT: Check in time is 3:00 p.m. and checkout time is 11:00 a.m.

SERVICES & AMENITIES:
Complimentary in-room Wi-Fi & hardwired internet access, in-room safe, 8 on-site dining options, 24-hour fitness center, spa, swimming pool, on-site parking (discounted self-parking rate of $7/day available during event dates; $20/valet).

Register Now

Where Are The Parts?

A Metal-Bender's Guide to Software Manufacturing

by Rick St. Germain
CMPIC Canada

Bor-ing. I'm sitting here on a cheap plastic chair watching our million-dollar 5-axis CNC machine crank out toy tops, freebies those marketing geniuses hand out at this trade show. Now don't get me wrong — as an unrepentant metal-bender, nothing makes me happier than seeing elegance emerge from a chunk of hard steel spinning on my lathe. It's what I do best. But after the first hundred or so, it gets a little old. The mind starts to wander.

As I watch the sequence of G-codes scrolling down the screen, I'm constantly amazed at how software can make tools dance like that. CAD models, tool paths and G-code? I totally get that. But how G-code drives my machine? Mmmmmm, not so much.

Software ACTS like a part with form, fit, and function. Well, sort of. Function certainly. It merits it's own part number on the memory chip's label. Not so sure about form. But if G-code is a software end-item, then where are the parts that go into it? How is it manufactured? When I ask the experts, I get the "Trust-Me" thing: "Add a quart of scotch and magic happens." I don't buy that. I'd be much happier if I could just see the parts. All of them. Including software.

Glancing over at the PLM vendor's booth across the aisle, I spot a guy half-heartedly poking at his laptop. Pale complexion, clean fingernails — must be a software geek. He seems to be enjoying conscripted booth-bunny duty about as much as I do. As closing time approaches, things go quiet so I grab a top and head over for a chat.

Misery loves company. Peter and I hit it off right away and we end up in the hotel bar talking shop over a few beers. When the subject of software came up, I leapt at the opportunity to have him settle my conundrum. It cost me a few rounds but, hey, it was worth it.

What's in the Box?

On a cocktail napkin, he sketches out the three core components of a computer. **Memory** stores patterns of binary digits or "bits" (ones and zeros), organized into "bytes" (8 bits), or "words" (16, 32 or 64 bits), each with its own unique memory address. The G-code instructions I see on my screen are actually bit patterns on a memory chip inside my machine controller. Each instruction has its own address. Okay.

Next he draws the **Processor**, an electronic component that uses an internal "instruction set" to decode bit patterns from memory to perform certain pre-programmed logical operations like additions, bit shifting, or setting status pins. The results are

continue reading

then presented to its output pins.

Um, whatever. More beer — this better be good, Peter.

Finally, he sketches a block labelled Interface that connects the processor output pins to device drivers for things like the stepper motors in my CNC machine.

Hey, I totally get that.

That means that the hardware card in the controller box is actually a computer running software that controls my CNC machine. Very cool.

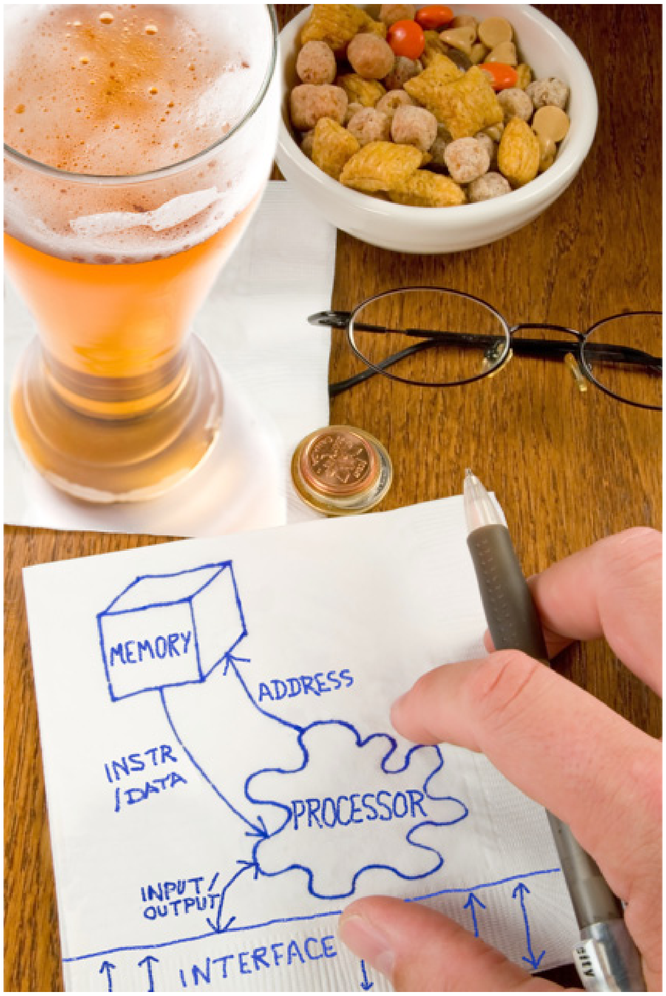
Then he drops a bombshell.

The Executability Test

If the sequence of bit patterns sitting in memory doesn't correspond to the instruction set of the target processor, it can't "execute". If it doesn't execute, it's not software, it's just data. That's the Executability Test.

To create executable software, we go through a manufacturing process that takes the design intent of **Source Code**, specified by developers in a human-readable text-based language, and translate it into **Machine Code** expressed in a machine-readable sequence of bit patterns corresponding to the instruction set of a specific processor.

This manufacturing process is called a **Build**. Technically, it's an "isomorphic translation" — simply changing one format to another without affecting the design intent. Manufacturing tools called **Compilers** perform this translation. They're executable software programs in their own right, running on a particular processor inside a development computer. Each compiler is designed to translate from one specific source code language to one specific processor instruction set, so we need to make sure we pick the one that matches our source and target formats. What is executable software for



one processor is just meaningless data to another with a different instruction set.

"So how does that software manufacturing process work, Peter? Do we have parts yet?"

Step 1: Compile

"I'm getting to that."

Source code expresses design intent, usually in text form, so it acts like a specification for manufacturing software — very document-like. We treat source code files like any other document — identifying them with a name, type and a version number that reflects the changes made.

Source code's specification statements direct the compiler to manufacture the equivalent machine

continue reading

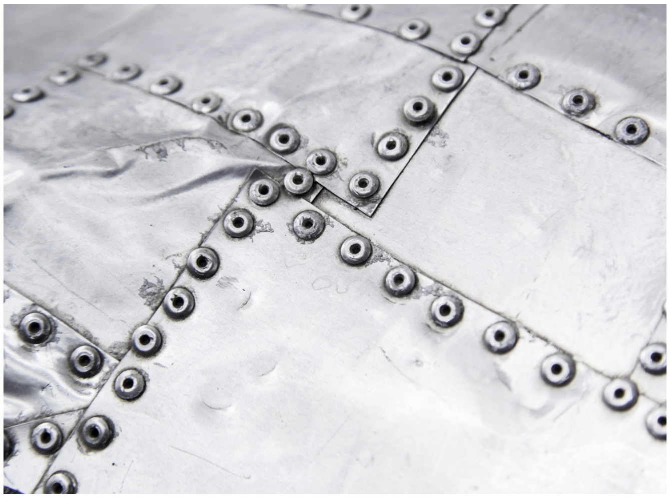
code using the instruction set of the target processor. Each source code file is compiled into one machine code file called a binary (.bin) or object (.obj) file. Since they're expressed in machine language, binary files are executable software.

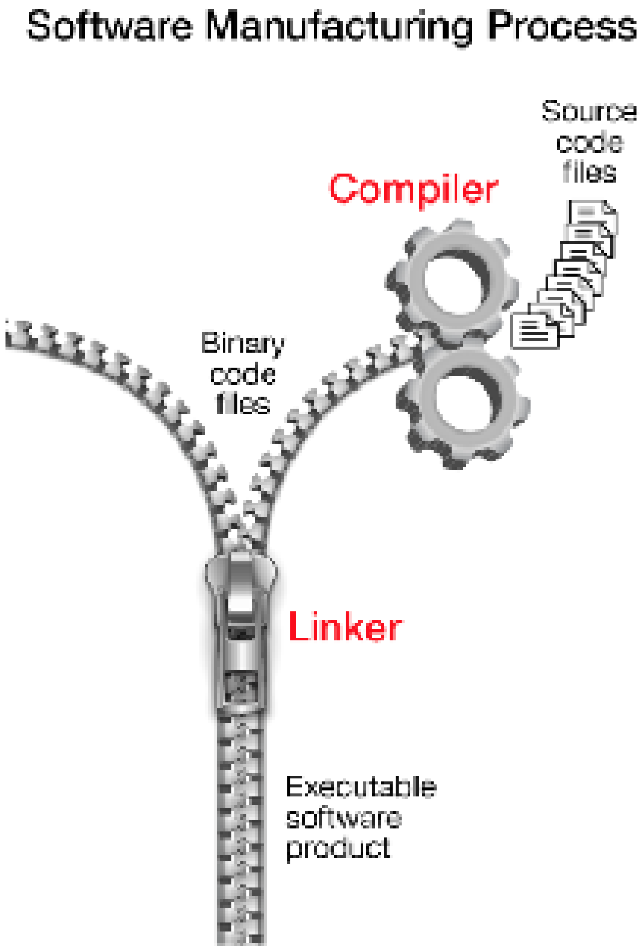
If there's only one source file to be translated, then we're done. The G-code for your CNC machine falls into this category. The processor on the the machine's motion controller card runs a piece of software called an **Interpreter**, a simplified kind of compiler that translates and executes G-code statements line-by-line rather than compiling the whole file at once.

Usually, though, we have hundreds or even thousands of source files that contain functions that call each other. A function call is simply a jump to execute a function in another memory location.

Think of it this way: an airplane is built with, say, ten thousand identical rivets. In hardware, one rivet design specification is used to manufacture ten thousand rivets. Each rivet is used only once. The specification is not the part — the rivets are.

In software, you still have one rivet-function specification, but you manufacture only one executable rivet-function then call it ten thousand times. The single rivet-function is re-used as many times as needed simply by jumping to it's binary code, executing it, then jumping back to where you left off.





A source code file is a specification, not a part. It's resulting binary file passes the executability test, so it's equivalent to a software part.

Woohoo! I knew it. But how come I can't find them?

That's because, in most instances, they're consumed immediately into the final software product and not retained separately.

An issue arises when a function in one binary file calls a function in different binary file. Since the source code files are compiled individually, the compiler can't "see" a function located outside the local file — it doesn't know where to jump. To fix that we need to assemble all the binaries into a single executable software product. That's the next step in the software manufacturing process.

continue reading

Step 2: Link

The Linker is the manufacturing tool that does this assembly job. It works liker a zipper. The Compiler cranks out individual parts (zipper teeth) for every specification (source code). The Linker then assembles those parts into a single executable software product.

The Linker does this in two passes. In the first pass, it loads the individual blocks of machine code into successive blocks of memory. While doing that, it populates a table with the starting memory address of every function it finds. In the second pass, it uses those addresses to populate the pointers for every function call.

The result is a single executable file (.exe) containing all of the individually-compiled binaries zipped up together, with cross reference pointers included. That's your software product end-item.

Lock and Lead

When it's released, the executable software product is typically packaged with a set of helper applications such as loaders, installers, and de-installers, along with non-executable components such as configuration and license files, release documentation and supporting data. A **Software Load** is equivalent to hardware's Packaged End Item and is managed in the same way as any other product release. It's assigned its own part number that appears on the label attached to the medium that holds it.

So there you have it — software manufacturing demystified. I finally get it.

A Box of Parts

As we were packing up at the end of the show, Peter walked over and handed me a small box the size of a softball.

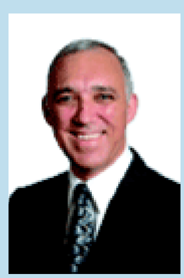
"We did a software build last night and I had them save some of the software parts for you as a souvenir."

"Very cool, Peter. Thanks for teaching me how software makes my machine dance."

He laughed. "No problem. Next time you can teach me how to make a top on your machine."

After he left, I got to wondering. I thought he said that software parts were… Hey, wait a minute!
I ripped open the box.

Hah! Gummi Bears — I've been had.



Rick St. Germain is a CM researcher, consultant, trainer, and coach with over 30 years experience in implementing military and commercial CM processes for both hardware and software. He is President and Managing Director of Nouvella Consulting Services based in Ottawa, Canada, and is Chief of Canadian Operations for CMPIC Canada. He can be reached at rstgermain@rogers.com

RICK ST. GERMAIN

continue reading

Who Really are the Users of Configuration Management Software?

by Tom Tesmer
CMstat

While recently speaking with the CM manager at an A&D site I asked what type of users did they have of their CM software. The response was that there were some doing a lot of work, but most didn't do much at all!

After my shock from this answer, it took me a moment to realize this was not about the work habits of his team, but how users worked – or not – with their CM software.

At this site, typical to many others I have visited, there were a relatively small number of CM specialists – the masters and Super Users – using CM software to create, update, and check configuration data all day long. They did the "heavy lifting" of the CM profession. Not surprising, most of these practitioners were quite experienced and well trained, often with certifications from CMII or CMPIC.

Yet, most CM software users are not CM experts in product engineering but casual users from engineering services, quality assurance, procurement, field service, logistics, and sub-contractors. Their job function requires them to use CM software tools, but perform very little CM work per se. As example, they may want to find configuration-related data quickly, trace the history if needed, navigate relationships of the as-is configuration, and assess impact of changes to future states. They are, however, not doing the type of CM work that someone in engineering

or manufacturing using PDM software typically performs.

This exchange reminded me of what CMstat learned long ago as we developed our CM solution PDMPlus; most CM users are not CM experts at all, nor should we force them to be. Most are "read only" types; searching for, viewing, and consuming configuration-related data to do their job, which for the most part is not doing configuration management.

As we in the software industry all know, but few are willing to admit, any software can be declared "easy to use" if there are enough expert users constantly working in the product. No matter how unintuitive, antiquated, or abusive a software user experience may be, over time it can become comfortable and

continue reading

trustworthy, sadly in the same way hostages often relate to their captors.

These experts who eventually can make any software easy to use for themselves are only a small percentage of the larger user community who may not have the same patience. The real test of usability comes from the population of poorly trained, or even untrained, casual users who want to log in, search for, then retrieve their desired information, and get out. The best user experience for them is a short, almost nonexistent one, where they did very little work. Any time spent slogging through navigation screens, help tutorials, or seeking other users out is time wasted.

The challenge with developing specialized industry-specific CM tools – like those required by the A&D supply chain or after-market service providers in MRO – is to satisfy the deep functionality required by the CM specialist without sacrificing the intuitive ease of use for the occasional, non-expert user. Over the years we at CMstat have struggled like most software providers with adding new functionality that our more expert customers wanted, but without adding complexity for the occasional user. Our focus on serving customers in just a few key industries has made resolving these conflicting requirements easier to balance, but it remains a diligent work in progress.

This delicate balancing act is where homegrown and customized CM software, often developed by user organizations and then bolted on to enterprise solutions by their IT support groups, struggle the most. The code developers dutifully listen to the precise yet frequently changing requirements of their primary users, as they should. It is no surprise they produce a solution expertly designed for expert users, and then run out of time and budget to make it usable by anyone else. This also explains why enterprise PLM solutions and industry-agnostic PDM software struggle with CM for industries that have their own set of standards and best practices. These solutions are so obese with general PLM functionality that occasional users rarely feel

comfortable or confident.

The best commercial-off-the-self (COTS) industry-focused solutions for CM should address all the functionality required by configuration management specialists while still providing an instantly instinctive experience for the casual user. After all, while there are many benefits of CM for the user, the ultimate value accrues to the enterprise because the work produced is accessible to many others throughout the organization.

We'd like to hear an estimate of the ratio of routine expert to occasional non-expert CM software users at your site. Send an e-mail to information@cmstat.com. If you would like to see a demonstration of how CMstat's PDMPlus balances functionality with usability, give us a call at +1.877.537.1959.

Tom Tesmer is President of TPT Technologies, Inc. doing business as CMstat. He drives the functions of Strategic Planning, Business Development, and Operations & Financial Planning within CMstat. Prior to joining CMstat, Tom was President of Evergreen International Helicopters, Inc. serving government and industrial clients world-wide. Tom's experience working with many A&D customers over the years has led him to become a well-respected champion and advocate for Configuration & Data Management. Mr. Tesmer received an Honorable Discharge from the U.S. Army after service as a Warrant Officer Helicopter Pilot participating in combat operations during the Viet Nam war. He holds a Bachelor of Arts degree from Western Michigan University.

TOM TESMER

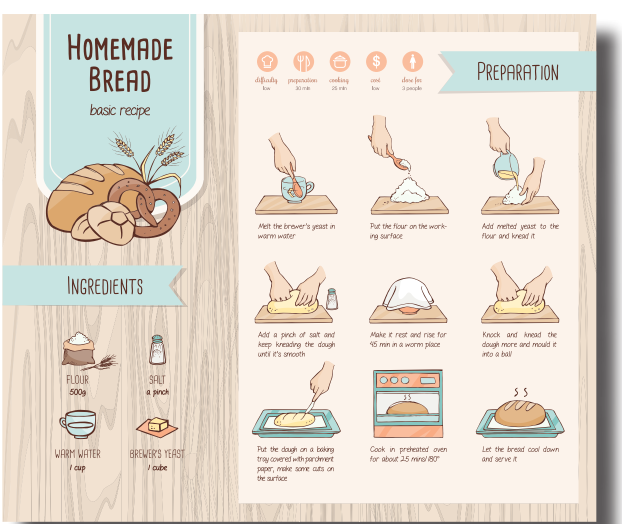
CM Origins Part 1

Configuration Management and Data Management

by Kim Robertson
Value Transformation LLC

I've spent most of my life wondering about the world around us and how the intellectual tools we have at our disposal came into being. Jack Wasson's Configuration Management for the 21st Century (www.CMPIC.com/whitepapers/whitepapercm21.pdf) is directly applicable. I particularly like the following two statements:

> "Those of (you) who know CM, know that it is everywhere. If you understand it, you see it daily, and even hourly."
>
> "In layman's terms, it is the plan (or blueprint) for a product, a process or a document before, during its lifecycle and beyond. It seems obvious, yet most have no idea what Configuration Management is, how it is performed or even how to begin doing it."



Control of processes is one of the necessary elements of Enterprise CM. In the broadest sense a process is a recipe for doing or observing something. It could be a recipe for applying vacuum coating lubricants to space hardware. It definitely includes a recipe for assuring consistency in making planetary orbit predictions. It could perhaps include recipes used in food preparation. Let's start with planetary orbits. Many of us have been taught that the foundations of integral calculus were laid down in 1635 by Bonaventura Cavalieri and refined by Sir John Wallace, Pierre De Fermat, Sir Isaac Newton and Gottfried Wilhelm Leibniz. The recently translated Cuneiform tablet below makes us rethink that a bit and push the date back 2,000 years. The Babylonians also successfully approximated the square root of 2 and applied the Pythagorean theorem nearly 4,000 years ago. This is more than a millennium before Pythagoras was born.

Late Babylon ca. 350-400 BC Akkadian

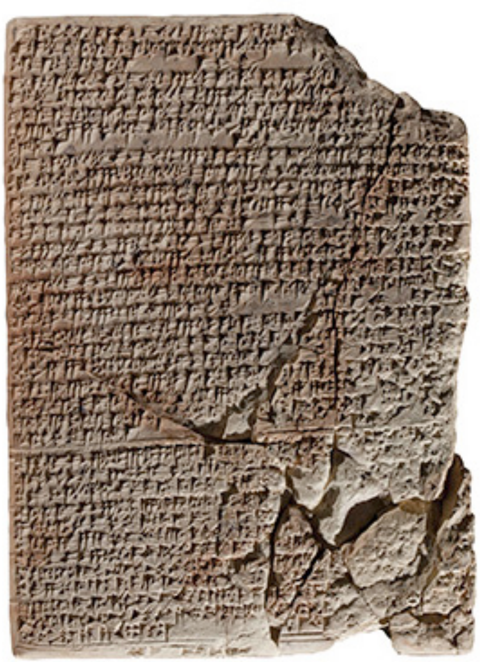
continue reading

The tablet is believed to contain a complex geometrical model that appears to be a rudimentary form of integral calculus to predict the path of Jupiter. Reaching a little bit let's look at published recipes as a form of process. They are certainly under document management and if by inference the publication or creation date can be deemed an early form of version control they are under a simplified form of CM and DM.

With that in mind after the 2016 holiday feasting let's start our culinary journey back in time with this partial list:

1896	Boston Cooking-School Cook Book	Frannie Farmer
1736	Le Cuisinier Moderne	Vincet La Chapelle
1570	Opera dell'arte del cucinare	Bartolemeo Scappi
1330	Yinshan Zhengyao	Hu Sihui
1226	Kitab al-Tabīh	al-Baghdadi
1126	Manasollasa	King Someshwara III
300	De Re Coquinaria	Apicius
2 BC	Banquet of the Sophists	Athenaeus
350 BC	Hdypatheia	Archestratus
1700 BC	Yale Culinary Tablets	Anonymous

The Yale tablets are a recent addition and contain 25 recipes, 21 for meat and 4 for vegetables. One of the Yale tablets is shown below. It is a recipe for stew:



Old Babylonian ca. 1,800 BC Akkadian

I know that some may disagree that these tablets have no relevance to CM and DM. You find what you look for … based on your search parameters. The more narrowly you define what CM and DM are the less you see of it in the world around you. We will explore this more deeply in future articles.

Kim Robertson is a NDIA Certified CM practitioner, consultant and trainer with over 30 years of experience in contracts, subcontracts, finance, systems engineering and configuration management. He has an advanced degree in operational management with a government contracts specialty and is the co-author of Configuration Management: Theory Practice and Application. He can be reached at Kim.Robertson@ValueTransform.com

KIM ROBERTSON

CMPIC's CM Training & Certification Courses

To register, please visit: www.cmpic.com/registration.htm
or contact the CMPIC office at: info@cmpic.com, 1-434-525-8648

- *CM Principles & Implementation* Certification Series, Courses 1 - 4
  - St. Augustine, FL March 21 - 30, 2017 - 2 Consecutive Weeks!
  - Bloomington, MN May 8 - 11 & June 5 - 8, 2017
  - Charleston, SC June 19 - 22 & July 17 - 20, 2017
  - Washington, DC area August 14 - 17 & September 11 - 14, 2017
  - Orlando, FL October 17 - 26, 2017 - 2 Consecutive Weeks!
- *EIA-649B Principles & Applications* Certification, Course 6
  - Charleston, SC February 6 - 8, 2017
  - Panama City, FL March 8 - 10, 2017 - Take back-to-back with Course 7!
  - Melbourne, FL March 29 - 31, 2017
  - Orlando, FL Aug. 30 - Sept. 1, 2017 - CM Trends Discount!
- *CM Assessor* Certification, Course 7
  - Panama City, FL March 13 - 15, 2017 - Take back-to-back with Course 6!
  - Orlando, FL Aug. 30 - Sept. 1, 2017 - CM Trends Discount!
- *SCM: Strategies, Techniques and Tools* Certification, Course 8
  - Panama City, FL April 3 - 6, 2017
  - San Diego, CA December 4 - 7, 2017
- *CM Standards & Practices Update*, Course 9
  - Orlando, FL Aug. 30 - Sept. 1, 2017 - CM Trends Discount!
  - Lexington Park, MD October 3 - 5, 2017
- *649-1 CM Requirements for Defense Contracts* Certification, Course 10
  - Charleston, SC April 24 - 26, 2017
  - Lexington Park, MD September 26 - 28, 2017
- ANNUAL EVENT - *CM Trends 2017: Seminars, Workshops, and Training*
  - Orlando, FL Aug. 28 - 30, 2017 - Experience the Full Spectrum of CM!

Click here for CMPIC's full public course schedule.

On-Site Certification

Did you know that CMPIC offers on-site certification and training for as few as five attendees? This is a great way to train your staff and eliminate the need for a large travel expenditure. Call us to find out more, or visit www.cmpic.com.

Submit an Article For This Newsletter

Do you have a CM story to tell? Would you like your CM article published in this newsletter? Send us an email and we'll provide details on how to get your article published. Please email: kerri@cmpic.com.

Contact Info

CMPIC LLC
P.O. Box 2131
Forest, VA 24551
ph: 1-434-525-8648
fax: 1-434-382-0677
email: info@cmpic.com
web: www.CMPIC.com

CMPIC Courses 1-10 are in partnership with:
UNIVERSITY of HOUSTON