G E N E R I C   C M

C O N F I G U R A T I O N   M A N A G E M E N T
E X P L A I N E D
Revision 1

By Richard L. Rumage

CONTENTS

FIGURES

TABLES

REVISIONS

Rev. 1, 6/2/06: Inserts Figure 6, previously omitted, and renumbers subsequent figures appropriately. Adds “Revisions” page and renumbers subsequent pages appropriately. Adjusts Tables of Contents accordingly. Adds “Revision” number to headers. Adds e mail address to pages 5 and 37.

INTRODUCTION

This book was written for industrial managers and others, military or commercial, who know very little about Configuration Management (CM) but still have to deal with it. CM is usually learned in a haze of hair-splitting jargon, confusing regulations, obscure specifications, and endless confrontations none of which are easy. Too often the result is frustration, misunderstanding, and anger.

In this book, small as it is, I have tried to provide a straightforward explanation of what CM is, what it does, how it works, and why. I have avoided the "Army Way" or the "Navy Way" in favor of the generic way. It's candid, unorthodox, based on solid experience[1], and most likely politically incorrect. It's written in common English however it does presume a reasonable understanding of engineering, manufacturing, procurement, deployment, and their relationships.

A number of side issues were left out of the basic text in favor of brevity. However, they are covered in the separately downloadable Appendix. This material is not essential to understanding generic CM. However, much of it is helpful in understanding people's behavior.

This book is neither a commercial venture nor the prelude to one. I am fully retired and intend to remain so. However, I will try to answer *genuine* questions about the content as time and energy allow. I can be reached at cmg@netlink.net

If you find that this effort helps your understanding of CM please tell others. No other advertising campaign is planned. If the book has merit word of mouth will do the job.

[1] Detailed in the Appendix for those who care about such things.

1. VIEW FROM THE TOP

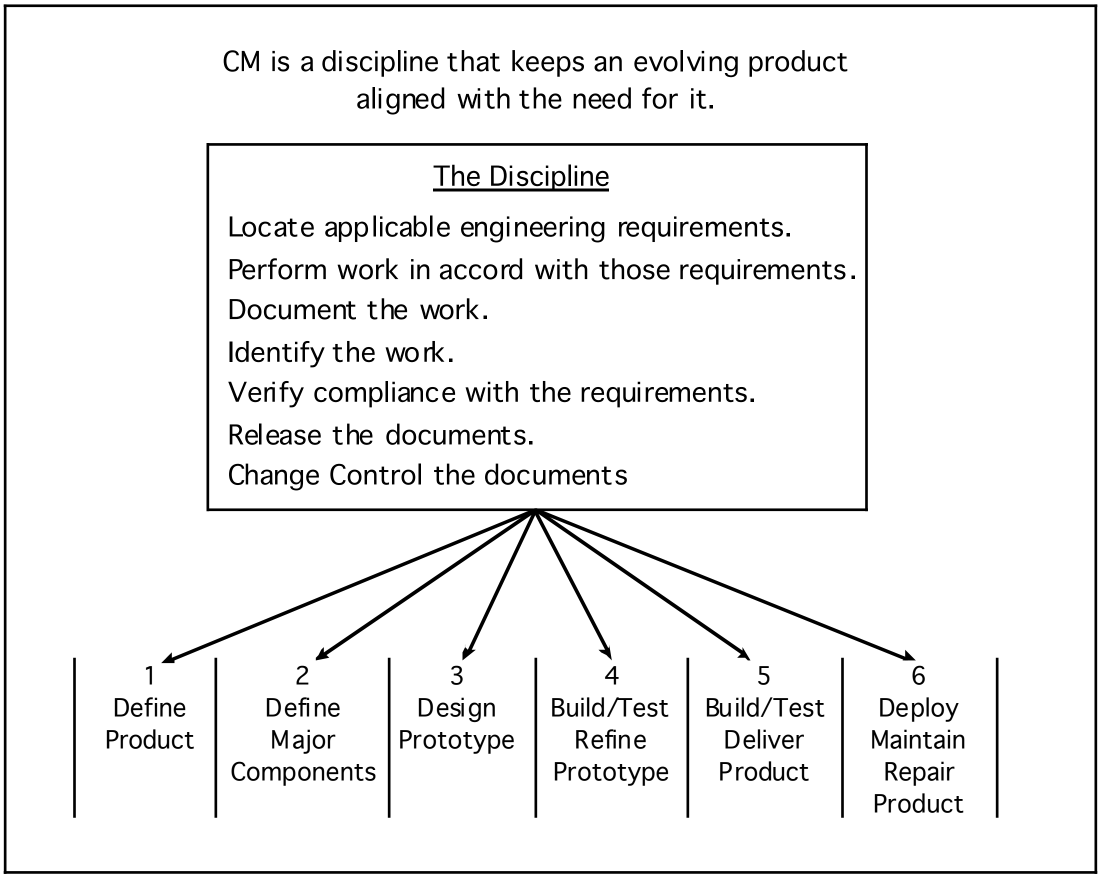

Fig. 1 Generic CM

If you can understand the elements and relationships in Figure 1, you will understand CM. It's not that hard. But before we look at its pieces let's consider why we have it at all.

CM exists to prevent design drift. It may seem odd, but during the life of a program (5 – 15 years) it was quite possible for say a fighter design to morph into an unintended fighter-bomber before responsible people realized that the design was drifting. There were other similar diversions. Ignorance or innocence, poor instruction, human error, misguided engineers, and deliberate misuse of government resources are among the causes. During the 50's and 60's the problem became intolerable and the Air Force took the lead in finding a solution. Configuration Management, System Engineering, and Weapon System Management were among the corrective actions. If all of this seems farfetched it's a matter of record and there are still a number of us able to attest to its reality.

Now, let's consider the pieces.

***

2. WHAT IS IT?

CM is a discipline that keeps an evolving product aligned with the need for it.

Fig. 2 Definition

A discipline is specialized know-how; in this case how to establish and maintain alignment of the product with the need for it.

An evolving product is one undergoing design, development, or modification.

The product is kept in alignment during its evolution because it's much less expensive than realignment afterward.

A product is aligned with need when it does (or will in the case of an evolving product) meet the need for it.

There are other far more official definitions than this one. However, the reader must know CM before they are understandable. Though sometimes useful in bureaucratic infighting, for us it's self-defeating.

Some may find the use of the term ”alignment” unusual. However, it replaces convoluted commentary on the reason for CM with a simple idea. CM exists to see that a product meets the specific need for it.

CM is always a means and never an end in itself!

***

3. THE DISCIPLINE

Locate applicable engineering requirements.
Perform work in accord with those requirements.
Document the work.
Identify the work.
Verify compliance with the requirements.
Release the documents.
Change Control the documents

Fig. 3 The Discipline

It's important to recognize that every one of the elements listed above was in use long before CM came along. The Air Force simply borrowed them when it created the Discipline. However, the view that CM is nothing more than a new name for old practices is incorrect. The point to the Discipline is not the elements it applies but the way that it applies them. However, before going there let's be sure we have a common understanding of these elements.

Requirements

Engineering Requirements are usually things that must be done to complete the design, turn it into working hardware, or maintain and repair the product. They also include the physical and functional interfaces that the item must meet along with test, safety, shipping, maintenance, and repair provisions.

Work

Work produces the item(s) called for by the requirements. It may be hardware, further detailed design, or maintenance and repair instructions.

Documentation

Documentation communicates information generated by the work to people or to machines, near or far, now or in the future.

Identification

Identification distinguishes one thing from another *precisely* by using a *unique* identifier.

Verification

Verification proves that documentation or hardware complies with the requirements for it.

Release

Release makes *verified* documentation available to down-stream functions and maintains an accurate record of it.

Change Control

Change Control maintains the integrity of released (verified) items by approving or disapproving changes to them.

***

4. PHASES & BASELINES

A phase is a block of time and a package of work with a discrete beginning and end. Phasing is a program planning practice intended to make programs more manageable. CM depends upon and influences phases but defining them is a program-planning task.

The best phasing is compatible with the product, the program, the industry, and the buyer. Most industries have evolved patterns that work for their products. However, in every case, phasing should be carefully tailored to fit the specific product and program. There is no standard formula! It's a matter of informed judgment.

Figure 4 displays a common model of program phases before CM while Figure 5 shows common program phases after CM was created

Design Prototype	Build/Test Refine Prototype	Build/Test Deliver Product	Deploy Maintain Repair Product

Fig. 4 Phases Pre CM

1 Define Product	2 Define Major Components	3 Design Prototype	4 Build/Test Refine Prototype	5 Build/Test Deliver Product	6 Deploy Maintain Repair Product

Fig. 5 Phases Post CM

Phases 1 & 2 were forced into being by the CM requirement to have a full and formal functional definition of the product before detail design begins; a requirement discussed in the next section.

In a different situation some of these phases might be combined or eliminated while others might be divided. The names would no doubt differ but the concept would remain the same. Phase the program to fit the product and the need.

<u>Baselines</u>: A baseline is a line serving as a basis, e.g. for measurement, calculation, or location. Since the authors of CM expected all three, they required several baselines and permitted others. In fact it is both possible and desirable to end Phases 1 through 4 with a baseline as shown in Figure 6. (Phases 5 & 6 produce changes to the Product Baseline but do not generate a new one.)

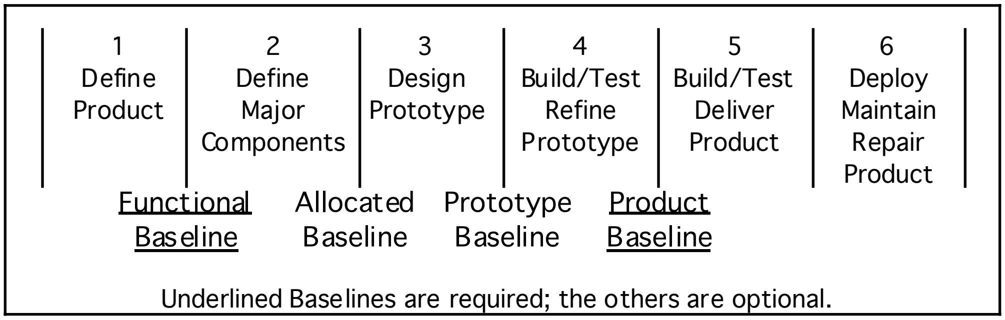

Fig. 6 Baselines

- Functional Baseline = documents containing the functions of the system
- Allocated Baseline = documents containing system functions allocated to major components.
- Prototype Baseline = documents containing the detailed design from which the Prototype is built.
- Product Baseline = documents containing the detailed design from which the product is built, maintained, and repaired.

We often make a buzzword out of baseline and attribute all kinds of powers to it. It's important to remember that there is no magic there. It is simply a symbol that stands for (1) a particular set of documents and (2) the point at which Change Control of those documents begins. There are other ways to do it but the baseline concept is a very useful convenience.

Baselines are established for each Phase by Release as shown in the boxed text below.

<u>The Discipline</u>

Locate applicable engineering requirements.
Perform work in accord with those requirements.
Document the work.
Identify the work.
Verify compliance with the requirements.
Release the documents. Establish Baseline
Change Control the documents

Fig. 7 Establishing Baselines

***

5. HOW IT WORKS

The Discipline is applied to each phase of the program.

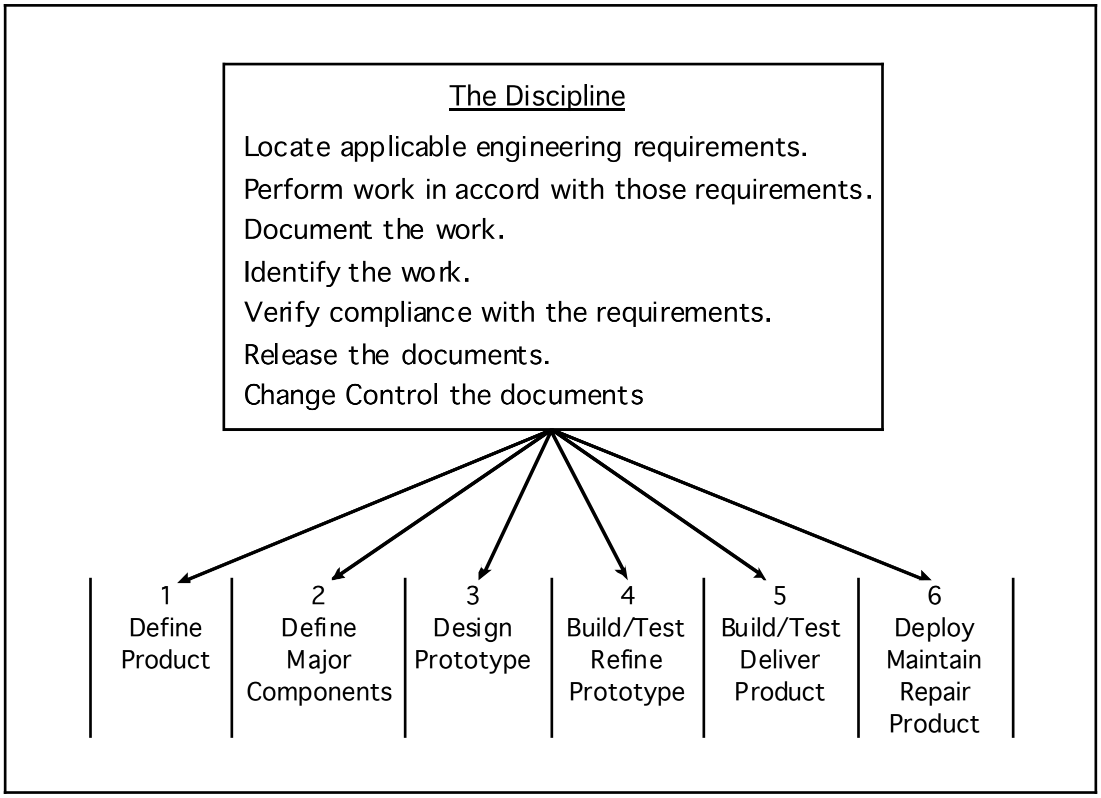

Fig. 8 Applying the Discipline

Phase 1. Define the Product

Locate Requirements: In the beginning the applicable engineering requirements for Phase 1 must be derived from the need for the product.

Need
Strategists and tacticians derive need from threat analysis. They may describe it in detail or as the concept for a product to be developed. CM does not define or evaluate need. It cannot distinguish a genuine need from a phony one. It will foster either one with equal vigor.

Establishing need is a difficult job. Elements of Congress or the White House may be involved. Budgets will be impacted. Doubtful approaches may require feasibility studies. Contractors may or not participate but they don't make the decisions. All of this work and more is repeatedly reviewed, critiqued, and revised. The process is quite formal and proceeds through stages. It may take weeks, months, or years. When the time comes, the responsible authority makes the critical decision and “need” is established or the effort is shelved.

For the military, law, regulation, or directive defines the “responsible authority”. In the commercial world need is better known as opportunity and is identified by the upper management of the company. With need established, a program can be defined (phased) and CM begins.

The Work: A product can be defined in terms of what it is (physically), what it does (functionally), or both. At this stage of development, it’s necessary to define the product functionally; that is, by stating the functions it must perform to satisfy the established need. However, if a major component such as an aircraft engine already exists it will be described by its top-level identification and that includes its physical characteristics.

System Engineers define the product by translating the functions it must perform into engineering terms suitable as requirements to be met by detail design engineers. The job is difficult. It must do enough but not too much.

Over Specification
Over-specifying is the practice of defining unnecessarily stringent requirements in the hope that the extra effort needed to achieve them will assure that the real requirements are met. This practice can turn up at almost any point during a program but when it occurs at the beginning the consequences are more severe and longer lasting. Every engineering decision becomes more difficult to make and to implement until the truth surfaces. The consequence is usually greater cost, wasted time, and increased change volume. Over-specifying seldom achieves its goal. Those who define need or convert it into engineering requirements have a heavy obligation. Meeting it is both art and science.

Document & Identify: The requirements are documented, usually in a specification often called the System Specification, and identified with a specification number.

Verify: The finished specification is evaluated by another set of experts. Their job is to determine that it does or does not meet the established “need”. If it does we go forward. If it doesn’t, it is corrected.

Release & Change Control: The verified System Specification is released which establishes the System Baseline and places its content under Change Control.

As we leave Phase 1 please note that each element of the Discipline has been applied in the sequence specified in Figure 8. These elements in this sequence will be applied to each phase of the program.

Phase 2. Define Major Components

Originally, a separate specification was to be written for each major component of the product. Each specification was assigned to a group of design engineers who designed the physical component accordingly. Initially, this approach worked but not well.

> The System Engineers who write the component specifications were given to over specifying which was deeply resented by the Design Engineers who had to meet unrealistic requirements.
>
> In most cases Design Engineers were quite capable of working with the System Specification. Further definition was seldom necessary. So the whole Component Specification notion was seen as overkill.

Much hemming and hawing resulted in the phase being made optional and it was oftentimes abandoned. But this action ignores at least two situations. When a component is to be designed by a subcontractor a separate specification for the design of that component is necessary for clarity and reference in the subcontract. When a component is to be supplied by the government a separate specification of exactly what is to be supplied will prevent major misunderstanding.

In multi-division corporations it is not uncommon to have the basic program under the control of one division while another division designs a major component of the product. It is usually wise to have a component specification or equivalent to control the design work to be done by the other division. It’s easy to get into trouble without one.

Now, it’s altogether practical to combine Phases 1 & 2 by having necessary component specifications prepared as a second task in Phase 1. On the other hand there is a great advantage to having the product requirements thoroughly settled in Phase 1 before becoming involved with the snarls of major components. A Phase 2 also provides an additional checkpoint for the program.

The Program Manager designates the components that will be subject to separate specification. Generally they will be subcontracted or government furnished items but they can be other items that he feels require special treatment. It is not uncommon for the customer to require special treatment for one or more items for his own purposes.

Locate Requirements: The requirements to be allocated (assigned) to major components are located in the System Specification.

The Work: System Engineers allocate (assign) requirements from the System Specification to each major component.

Document & Identify: The requirements are usually documented in a specification for each major component and identified with a specification number. (These specifications are referred to in this book as Component Specifications.)

Verify: Each Component Specification is evaluated by another set of experts. Their job is to determine that it does or does not meet the requirements of the System Specification. If it does, things go forward. If it doesn't, it is corrected.

Release & Change Control: Verified Component Specifications are released which establishes the Allocated Baseline and places its content under Change Control.

Phase 3. Design Prototype.

Locate Requirements: The requirements for this phase are in the System Specification (from Phase 1) and the Component Specifications (from Phase 2).

Subcontract: Issue subcontracts as appropriate referencing the applicable Component Specification from Phase 2.

The Work (for both Sub and Prime Contractors): Detail Designers meet Phase 1 & 2 requirements by selecting and arranging parts and materials to perform the functions required. Test Engineers develop methods (usually demonstrations) to prove that the functional requirements have been met.

Document & Identify: The detail design requirements and tests are usually documented in drawings, specifications, and text procedures identified with document numbers.

Verify: Other experts, most often checkers and editors, examine each document. Their job is to determine that it does or does not meet the requirements of its next assembly. If it does, things go forward. If it doesn't, it is corrected.

Release & Change Control: Verified drawings, specifications, and test procedures are released which establishes the Prototype Baseline and places its content under Change Control.

Phase 4. Build, Test, & Refine Prototype.

Locate Requirements: The requirements for this phase are found in the drawings, specifications, and test procedures created during Phase 3.

The Work (Task 1 - Build): Technicians and mechanics fabricate and assemble parts and materials into one or more prototypes.

Verify: Their work is inspected to assure compliance with requirements.

The Work (Task 2 - Test): Technicians run the test (demonstrate) the prototype(s) and record the results.

Document & Identify: The results are documented in a test report identified with a document number.

Verify: The report is evaluated to assure compliance with the requirements. This is the acid test for a program. The test procedures are designed to show that the prototype does or does not meet the requirements defined in the Phase 1 & 2 specifications (which contain the need in engineering terms). Every reasonable effort is made to conduct such tests under conditions as close to those in actual deployment as possible. If the prototype fails these tests, corrective action is required. If it passes, a suitable celebration follows and the work goes forward.

The Work (Task 3 - Refine): Oftentimes the techniques used to produce the prototype are unsuitable for volume production. So production specialists review the prototype design for producibility and recommend refinements (changes) to enhance production. Acceptance test methods are defined to prove that production hardware meets its requirements.

Document and Identify: The proposed changes are documented in Change Proposals and identified with Change Proposal Numbers. Acceptance tests are documented in procedures and identified with document numbers.

Verify: Proposed changes are evaluated to assure that they will not degrade hardware performance. Acceptance tests are evaluated for adequacy.

Release & Change Control: The Prototype drawings, specifications, and test procedures along with verified changes to them are released which establishes the Product Baseline and places its content under Change Control.

Phase 5. Build/Test Deliver Product.

Locate Requirements: The requirements for this phase are found in the Product Baseline; the drawings, specifications, and test procedures released at the end of Phase 4.

The Work: Technicians and mechanics fabricate and assemble parts and materials into products. Maintenance Engineers prepare such instructions and manuals as may be required to maintain and repair the product in the field.

Document & Identify: Proposed changes if any to Product Baseline Documents are documented in Change Proposals and identified with Change Proposal Numbers. Maintenance and repair instructions are documented, usually as Manuals, and identified with manual numbers.

Verify: The hardware is subject to acceptance inspection (examination and test) to assure that it conforms to the Product Baseline. Proposed changes if any are evaluated to assure that they will not degrade hardware performance. Manuals are evaluated to assure that they comply with the Product Baseline.

Release & Change Control: Verified changes if any are released as changes to the Product Baseline and placed under Change Control. Verified manuals are released and placed under Change Control.

Delivery: The completed product is delivered.

Phase 6. Deploy, Maintain, Repair

Locate Requirements: The requirements for this phase are found in the Product Baseline; the drawings, specifications, and test procedures released at the end of Phase 4 and any maintenance and repair instructions or manuals released during Phase 5.

The Work: Technicians and mechanics perform routine maintenance and repair of the product as necessary

Document & Identify: Generally this work is recorded in logbooks. Changes to the Product Baseline are documented in Change Proposals and identified with Change Proposal Numbers.

Verify: The maintained or repaired product is subject to Acceptance Inspection (examination and test) to assure that it continues to conform to the product Baseline. Proposed changes if any are evaluated to assure that they will not degrade hardware performance.

Release & Change Control: Verified changes if any are released as changes to the Product Baseline and placed under Change Control.

Change Control

The product is built, maintained, repaired, or modified only in accordance with released documentation. Once something has been released it can't be changed without going through the procedure shown in Figure 9.

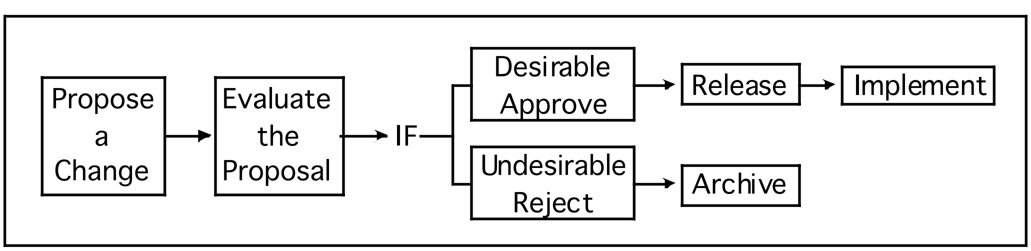

Fig. 9 Change Control

An approved Change Proposal becomes a Change Order that is implemented by releasing it. In due course, released changes are incorporated into the documents they change. Those changed documents are reviewed for continuing compliance and re-released to replace the old documents plus changes.

There is much more to be said about Change Control but this is not the place to say it. Those with a special interest in the subject (and it is a fascinating one) should download the Appendix where they will find it discussed at length.

***

6. CHAIN OF VERIFICATION

The Chain of Verification shown in Figure 10 is the crux of CM.

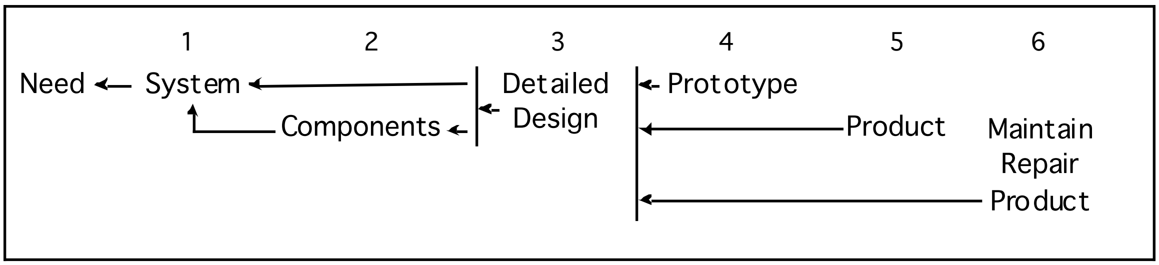

Fig. 10 Verification

The logic is simple.

If the Product complies with the Detailed Design,

The Product meets the need,

Because the Detailed Design complies with
the Component and System Specifications,

And they specify the need in engineering terms.

Such alignment is accomplished by applying the CM Discipline:

- Identify requirements to know what needs to be done.
- Do the work.
- Document the work result to communicate it.
- Identify the work (documents or hardware) to distinguish it.
- Verify compliance with the requirements.
- Release the documents to establish a record.
- Change Control the documents to preserve their integrity.

To each phase of the program:

- Define Product
- Define Major Components
- Design Prototype
- Build/Demonstrate Prototype
- Build Product
- Maintain Product

At this point it's tempting to write those fatal words, "That's all there is to it!" because in one sense they're true. Yet, there is more to be touched on and we will do so. However, if you understand the preceding page you understand the essence of CM.

A great deal has been said about the importance of staying aligned with need. What happens if the need is no longer valid or there was a mistake in the definition in the first place? The answer is move to change it! Now!

Propose a formal change to the System Specification that will correct the deficiency. If it's approved, implement the change. This can be a major undertaking if the program is well along because changes must be made at every level of detail. Still, it must be done.

***

7. SOFTWARE

Originally, enthusiasts attempted to force fit Software to CM or CM to Software. It didn't work. CM imposes structure and constraints. Typically Software, as an emerging technology, vigorously fought any structure or constraint from any quarter, including its own.

Perspective

Initially, no one really knew what software was. Anyone who could make it work was seen as a black art practitioner. Eventually, these practitioners evolved their own jargon, methods, and rituals. They could do no wrong until they became exuberantly optimistic, enormously expensive, and exceedingly unpredictable. This evolutionary pattern is relatively standard for any new technology.

Emerging technologies are properly developed in feasibility programs designed to accommodate their idiosyncrasies and develop their promise by pushing the state of the art. They do not belong in design and development programs destined for production until they are mature enough to withstand the rigors of need, predictability, cost, and schedule.

However, there is no formula that determines maturity and there are always those who push the state of their art into collision with the realities of industrial practice. The result is a clash of languages, methods, and rituals. Only time, patience, and experience produce the mutual understanding necessary for successful integration. All new technologies pass through this trial by fire. If they are viable, they eventually become commonplace.

Software is now commonplace. But remnants of its evolution linger and there is still much potential to be developed. It's wise to remember those facts.

As A Product Component

Software is developed by programmers to run in a hardware device developed by hardware engineers. It's easier to think of the software and hardware together as a single component that performs a function. This component is often called a computer system, Figure 11, even though the hardware may be a simple microprocessor and the software no more than a few lines of code.

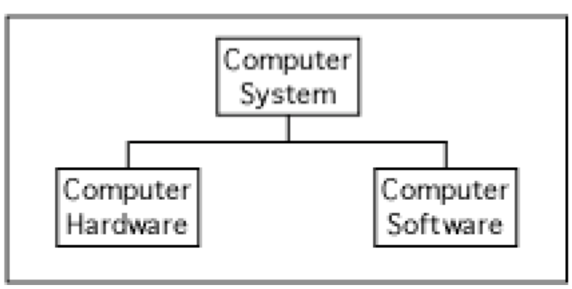

Fig. 11 Simple Computer System

When a computer system is a component of the product, CM applies in the same way that it applies to hardware regardless of the differences in jargon. However, adjustments similar to those required for any specific technology must be made. For instance, undimensioned drawings were developed to accommodate the photo-etching technology. The documentation of Software may look odd and may be recorded on a disk rather than paper but it's still Documentation. Generic CM does not change.

The derivation of technical requirements for Software is sometimes debated as a chicken or egg problem. Are requirements derived from a computer that needs Software or from Software that needs a computer? The answer is neither. They are properly derived from a function of the product that is best performed by a computer system. Both the software and the computer must be designed (or selected) to operate as compatible components of that computer system to perform that function. Of course, that function has its origin in "need" passed down through the phases. It may be identifiable in Phase 2 as a major component or not until Phase 3 as an element of the design.

High-risk takers, hoping to achieve phenomenal breakthroughs, often use immature technology. The consequence is really a feasibility effort embedded in a development program. It's very high-risk but sometimes it works. However, be mindful that the objective of a development program is to "Do this". For feasibility the objective is, "Can this be done"? These are not compatible goals! The usual development program techniques will not work in the feasibility effort. It should be encapsulated as a side-program until it is successful. While some elements of CM can be helpful within the capsule, the full spectrum is seldom advisable. Participants won't be tied down and risk-takers exhibit effervescent certainty instead of acknowledging risk. Usually, encapsulation is not achieved and the development effort suffers accordingly.

The fact remains that CM is applicable to Software, as a product component, with little more accommodation than that required for other evolving technologies.

As The End Product

Software is the end product for many companies. They do nothing else. However, they program it to work in a computer or it's useless.

The "need" in this case is derived from the computer system being developed or used by some other organization. That system is expected to perform a specific function(s). That function must be specified in enough detail to allow the development of software. These activities must be disciplined enough to produce a computer system that performs the specific function(s) required. CM can and should be applied.

Commercial Market

A wide variety of computers already exist in the commercial market. Now comes a company dreaming of a killer application. What must they do to achieve their dream?

They must know or gather the significant attributes of every existing computer model that is expected to host their application. Those attributes are one part of the "need". The other part is the attributes of the application they expect to develop. However, this kind of “need” is soft - and fickle.

The difficulties of development and the limitations of finance will cause computer models or software attributes to be added or deleted. These changes can be made with relative impunity. The only limitations are future market share, competition, and cost. These circumstances usually turn “need” into a mush that defeats the effective application of CM. However, if the “need” is stable, CM can be successful.

Some of these companies use one or more of the CM elements and call it CM. They are incorrect but no great harm is done unless someone else is depending on them to use the real thing. In that case, things come unglued.

There are many companies now offering to help others implement CM. The offer is usually based on software products that they sell. They are likely dealing in good but mistaken faith. Generally, these folks are offering various combinations of bundled computerized practices that they believe constitute CM. These programs may or not integrate with various existing management systems. The day may come when the entirety of CM is fully automated but it’s not here yet! If you decide to consider such a product:

- Examine it against your *well-defined* need.
- Try before you buy!
- If it meets your need, buy it, however --
- If it’s a first generation program that will operate or impact critical activities require indemnification against consequential damages or take other protective steps. It’s almost certain that you will need them.

***

8. GOVERNMENT CONTROL

Background

Commercial firms may use or not use CM in any way they like unless they choose to do business with the government. Generally firms doing business with the military and some other agencies are required to use CM but only as specified in their contracts. Some of what's specified will relate to government control – that is, approval of original documents and subsequent changes to them by the government customer.

Generally, controllers want more and the controlled want less. This is more than a contest of ego and power. No control is chaotic. Complete control is paralyzing. The goal is to eliminate chaos without paralyzing anything. Even so, there is always someone demanding more so that achieving a reasonable balance is exceptionally difficult.

The military customer will insist on control of performance, outside interfaces, supportability, safety, and security requirements. These elements are usually stated in the System Specification. In addition, he will want control of Demonstration Test Procedures and Acceptance Test Procedures. Figure 12 displays this approach.

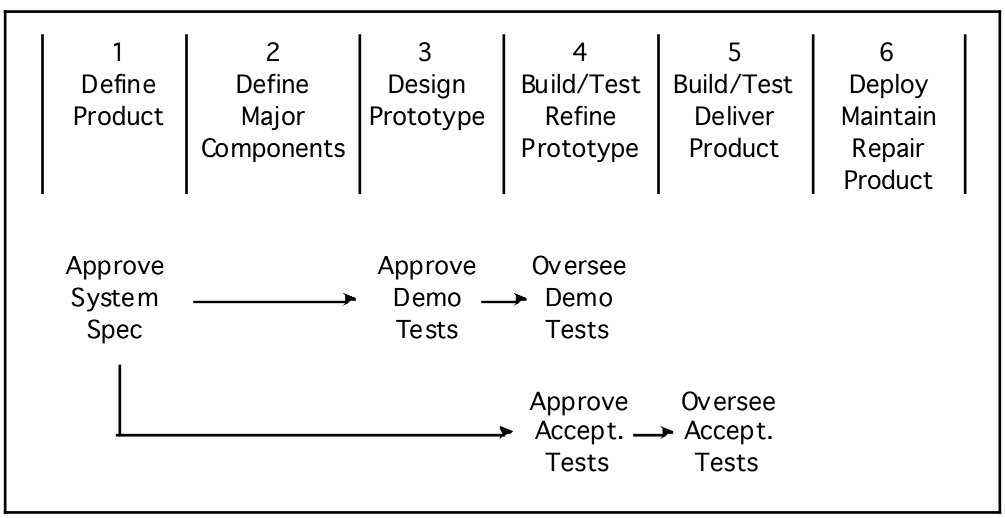

Fig. 12 Government Control

The System Specification defines the functions that must be performed. The Demonstration Tests prove that the design complies with the specification. The Acceptance Tests prove that the manufactured product complies with the design. These elements may be difficult but they are manageable and certainly reasonable for any prudent customer. However, they are not always enough to warn that a program is troubled in time to prevent failure. So more controls are applied.

The military buyer has legitimate reason for concern. The stakes are high. The obligations are great. They extend from the need to adequately equip fighting forces to the control of a large part of the pubic purse. Regrettably, unscrupulous or incompetent contractors do exist. There is a serious need to find them out before they ruin a program. Even so, excessive control does more to punish the ethical, raise the costs, and obscure the unscrupulous than it does to catch anyone. Periodic technical and management progress reviews by knowledgeable reviewers are far more effective.

Nonetheless, the government has extended its control over contractors by citing military standards in contracts. These standards take various forms and cover most conceivable conditions. The contractors' plea to "tell us what you want, not how to do it" fell on deaf ears. At the same time, contractors caused much of this excess with "can't-be-done" challenges that provoked "then-we'll-tell-you-how" responses.

Acquisition Reform[2]

A few years ago the military became earnest about Acquisition Reform. ("Acquisition" means procure, purchase, or buy.) In brief, the reform would:

- Stop imposing CM requirements by citing military standards. Instead, ask the contractor how he applies CM. Evaluate his method against industry standards and accept it if appropriate.

  The military has been moving to replace the old standards with more generalized provisions and greater reliance on CM Plans. This method will promote communication and likely result in more useful CM Plans written by the contractor, evaluated (approved) by the government, and cited in the contract.

- Require government configuration control of performance requirements rather than the detail design *in most cases*.

  In simplest terms, Figure 12 presents one form of this approach. Overall, it should provide adequate government control while removing many lower level complications.

- Base government control on adequacy of process rather than inspection of product.

  This is an established verification method often used for manufacturing processes. In essence, if the process yields a good product use the process inspection for acceptance of the hardware produced by the process. Wider use would greatly simplify many operations. It would not reduce government control but it would reduce the idiosyncrasies of individual reviewers and inspectors.

---

[2] MIL-HDBK-61A

- Replacing military standards with industry standards whenever possible.

  This approach is very helpful to firms with both commercial and defense products because they can base their operations on one set of standards instead of two. It also encourages the use of existing off-the-shelf commercial items that can greatly reduce cost to the government.

There is an additional item that is related to this strategy.

- Development of a government-industry database that includes all CM information necessary to support and maintain products (including software) during the life cycle of the product. It may be extended to re-procurement of entire systems.

  This is a laudable but formidable goal fraught with difficulty. It should have no effect on Generic CM. However, it will require large amounts of detail managed with exquisite precision. If implemented wisely and slowly it has the potential of becoming a very effective tool. However, it also has the potential of smothering CM with the effort to improve the process.

The brevity of the foregoing descriptions and comments does not demean the importance of the effort. It is significant. Many believe that the contactors' plea has finally been heard. However, implementation will not alter Generic CM.

Ultimately, successful implementation depends upon competent professionals of good faith and good sense. But, it won't be fast. Existing programs operating under the old rules will likely continue in that fashion. The difficulty of changing in midstream is greater than the benefit to be realized. Changes that appear to be simple as policies are much more complicated in practice. Both the old and the new will co-exist for some time. Therefore, both are covered in the following sections as appropriate.

***

9. CONTRACTS

If it isn't in the contract, it doesn't have to be done.
If it is in the contract, do it or change the contract!

For most people, contracts are part of the world's dullest reading so they don't read them! Most organizations maintain a Contracts Department responsible for getting the contract right in the first place and then telling others what it means. Unfortunately, these departments seldom include anyone who is CM sensitive let alone knowledgeable. The result is trouble for both government and company. The solution is to have a CM Specialist review the contract and explain the CM Requirements.

CM Requirements

Government contracts call for CM in a wide variety of ways. It would be nice to find the requirements in one place but that seldom happens. They pop up in various sections of the contract in specifications and in other documents referenced in the contract including requirements for technical data.

Although Acquisition Reform, discussed above, promises considerable improvement it will take a while. Until that time comes, and most likely afterwards, it is wise to search the entire contract for CM requirements

CM Plans - Old

CM Plans came into being in an effort to consolidate and clarify CM requirements. As with so many things, they became additive rather than consolidating. However, they never gained high standing as requirements. Various sections of the contract document continued to overshadow them.

Generally a CM Plan, prepared in accordance with specified format and content, was called for in the Request for Proposal. The Plan was expected to detail the way in which CM would be applied to the program and to some extent it did. However, for the most part, the Plans became sales documents hyping the contractor's CM capability.

The Plan was submitted as part of the contractor's proposal and, after being picked at, became a part of the contract. Thereafter, it was put on a shelf and never seen again unless a dispute developed. In that case, the parties miraculously rediscovered the Plan and began finger pointing. It was an ignoble end for a noble effort.

Even so, it was unwise to treat the CM Plan as nothing more than a sales document. When it was part of the contract, it could be enforced! So both parties were well advised to prune the hype. Unfortunately, this happy outcome seldom occurred because most of the innocents assigned couldn't distinguish hype from reality. They were as likely to cut a necessity and expand the fluff as not.

CM Plans - New

Acquisition Reform promises a major change in this situation. First the military intends to detail its requirements and expectations in its own CM plan made available to contractors before the first Request for Proposal is written. It expects the contractor to participate in such planning. Second, the Contractor will produce his own CM plan that details his response to the military plan. Both plans would likely be referenced in the contract.

This approach has an enormous potential provided that: (1) the innocents on both sides are replaced by competent knowledgeable people and (2) the powers-that-be pay attention to the results. Otherwise, the probable course is degeneration into a greater display of the same old sales hype.

Requirement Development

Specific CM Requirements usually begin in a Request for Proposal. It's hard dull work but it‘s necessary to read the whole thing at least once, highlighting CM Requirements as you go, in order to find all of them. Yes, that reading includes the boilerplate, most of the referenced documents, and most certainly any CM Plan.

Once the requirements are identified, they should be placed in a framework that displays the relationship of one to another and to the basic work of the program. It's important to understand the consequences and then decide how the requirements will or will not be met.

Compliance considerations usually get short shrift because the emphasis at this point is on winning the contract. A good CM proposal will not win and a poor one will seldom lose a contract. Few organizations will admit to themselves that they cannot or should not meet a given CM requirement. So they accept it and sometimes add embroidery.

The contractor's proposal is submitted, reviewed, and negotiated. This is the time and place to adjust CM Requirements if it couldn't be done before. However, it's awkward because the negotiators really have other matters on their minds. Unfortunately, both the customer and the contractor seldom have people at the table who really understand the fundamentals. So the debate turns upon individual elements such as Documentation. They tend to overwhelm.

Again, hard dull work is necessary when the Contract is issued. It must be read again at least once to find all of the requirements. Yes, requirements can and do change between submittal of the proposal and issuance of the contract.

Contracting Patterns

Figure 13 displays one relatively common pattern of contracting. It is important to see that the CM approach *for the program* is consulted and applied correctly each time a new contract is created. Otherwise, the requirements in the next contract can easily become incompatible with those in the last one. This result is particularly likely if either contractor or government personnel are changed along with the contract. Each new person brings his perception with him. Unless it is conformed to the approach for the program it will result in new interpretations which can easily be incompatible with all that has gone before.

This concern is even more important if the Contractor (the company) is also changed or a second Contractor is awarded a contract pursuant to competition.

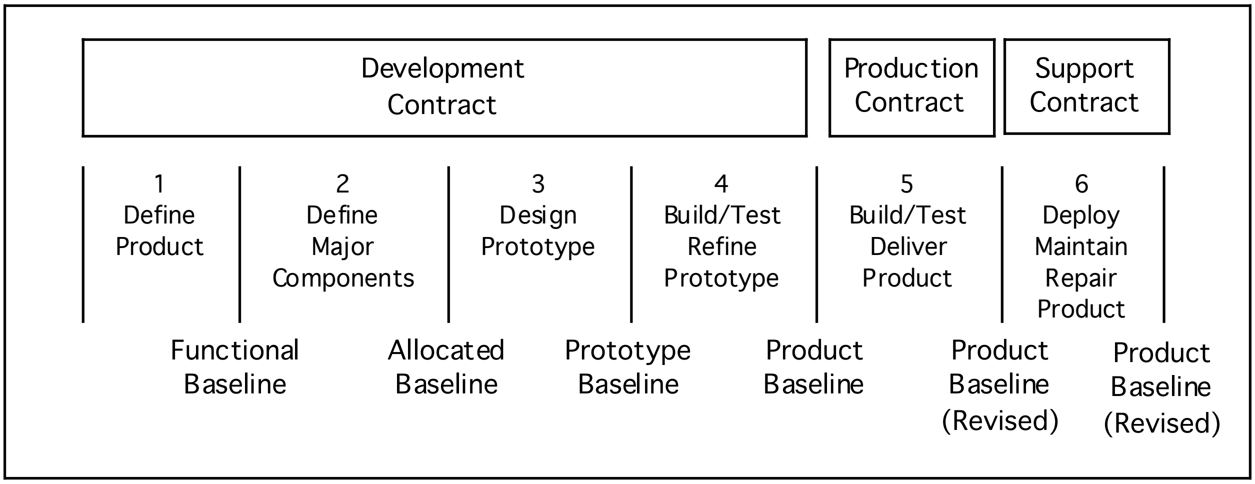

Fig. 13 Typical Contracting Pattern

Subcontracting

Most major contractors engage one or more subcontractors to design certain components of the system. In such cases, CM applies to the subcontracted components. Figure 14 illustrates such a situation.

When verification to establish a baseline is conducted the subcontractor's work is part of it. When demonstration of the system is required, the subcontractor's component is installed and demonstrated as part of the system. All of the comments about contracting apply to subcontracting as well.

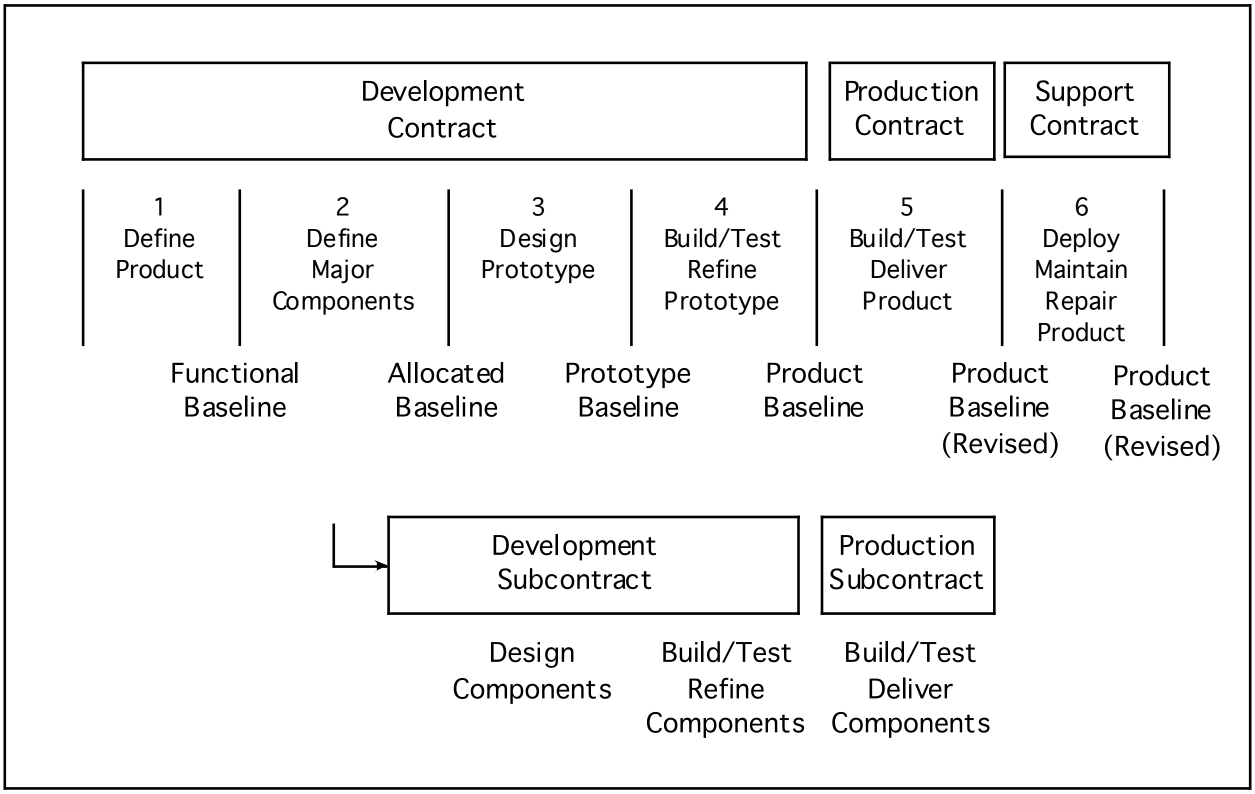

Fig. 14 Subcontracting

Competition

From time to time, the powers-that-be rediscover competition and demand greater use of it in government procurement as a cure-all for most ills. Sometimes it lowers costs and sometimes it doesn't but it always complicates the management of a program. In theory, it requires a competitive documentation package suitable for manufacture by any competent manufacturer. In practice, it requires enormous care and immense patience.

When the competition is for first production, the design contractor must produce the documentation package without the benefit of production proofing. This practice almost guaranties vociferous complaints from the competitor about the inadequacies of the package. Endless wrangling and high change rates follow.

No documentation package can be considered suitable for competition until it has been through Pilot Line and Initial Production proofing regardless of what they are called or how they are merged. Identical performance by two contractors is impossible without, and difficult with, enormous and expensive coordination between them. That coordination cannot take place during a competition.

When competition is required, *a high change rate for one to two years should be expected and planned for*. Unfortunately, high rates usually occur but the planning does not.

Competition does appear to lower unit cost in follow-on production. Whether this happy state is an overall cost saving depends on the cost of getting there and the duration of production. The answer is a good subject for an MBA thesis but not for this book.

Competition does not alter Generic CM but it complicates its application and coordination because two contractors must be synchronized. The same condition applies when another contractor is introduced, without competition, to achieve the required production volumes.

***

10. CM ORGANIZATION

There is no optimum organizational pattern or location for CM. It's not naturally an organization. It's a discipline. In the beginning, everyone was expected to learn enough CM to perform his part of the job just like everyone must learn enough mathematics to do his job. Of course, that didn't happen. Conventional wisdom took hold, "If everyone is in charge, no one is in charge, so put someone in charge." They did but it wasn't as effective as it sounds.

Tasks

The "Primary Tasks" in Table 1 are required to accomplish CM. The "Personnel Required" is based upon the skills needed to perform the tasks.

Primary Tasks	Personnel Required
Establish Need	Strategists & Tacticians
Establish Phases	Program Planners
Establish Contract	Contract Analysts & Lawyers
Define CM Requirements	CM Specialist
Define Product	System Engineers
Define Major Components	System Engineers
Subcontract, if necessary	Buyers (Purchasing Agents)
Design Prototype	Design Engineers & Programmers
Build Prototype	Manufacturing Engineers, Mechanics, & Technicians
Test Prototype	Test Engineers & Technicians
Refine Prototype	Producibility & Design Engineers
Build Product	Manufacturing Engineers, Mechanics, & Technicians
Test Product	Inspectors & Technicians
Deliver Product	Production Personnel
Maintain Product	Maintenance Engineers & Technicians
Repair Product	Maintenance Technicians
Document & Identify	Draftsmen, Checkers, Analysts, Writers, & Editors
Verify	Selected Technical Specialists, Engineers, &Technicians
Release	Release Clerks
Change Control	Selected agents of all elements of the organization.

Table 1 Who Does What

Generally, the personnel required by Table 1 existed and had an organizational home before CM was created. The only exception was the CM Specialist regardless of what we call him. Of course, it was different for each organization but generally here's what happened.

Someone was picked to figure it out, be in charge, take responsibility, do something, etc. Of course, he needed someone to be in charge of. So bits and pieces were taken from what once was called Documentation and they were re-named CM. And, there had to be a place to report so he was assigned to Administrative, Design or System Engineering, Quality Assurance, Logistics, Contracts, etc. The ingenuity displayed was not only ingenious but also disingenuous and almost entirely useless. The new CM organizations were without power while the organizations that had power remained uninhibited.

Organization

A quick review of Table 1 will convince most people that there is no way to gather all of those tasks into one organization for direction by one individual. So what's the alternative?

> When things function adequately, leave them alone. If they are not functioning, take the corrective action that you should take without any special regard for CM.
>
> Acquire a CM Specialist and task him with (1) defining CM Requirements for inclusion in the contract and (2) for interpreting those requirements, as needed, for members of the organization.

Will it work? That depends upon the contract compliance attitude and function of the organization. If it's weak, CM will fail because people will ignore the contract. If it's strong, CM will likely succeed.

Of course this approach is too simple and direct to be acceptable in most places and many twists and turns have been and will be tried. Even so, please consider the following before you abandon the notion altogether.

The unique element that CM brings to any program is the assurance that the evolving product remains aligned with the need for it. It accomplishes that job through requirements specified in the contract. The result sometimes produces new uses for old methods but it does not produce new methods. For instance:

- Phases 1 & 2 were new but the use of Phases was not.
- Specifications to document the results of Phases 1 & 2 were new but specifications as such were not.

Consequently, people familiar with phasing or specifications, etc are called upon to modify past practice to fit new conditions. While adjustment is sometimes difficult, the people who can do it already exist in the established organization. There is no need to reorganize and every reason not to!

CM Specialist

On the other hand, someone who understands most of the complications and consequences of CM requirements is generally rare. Whether he's called specialist, guru, expert, or manager he must be found or created. The desirable qualifications are:

- A thorough understanding of the operation of an engineering-manufacturing organization.
- A thorough understanding of CM and its impact upon such an organization.
- The ability to negotiate with customer or contractor and define clear CM requirements for inclusion in contracts.
- The ability to interpret contract CM provisions.

The foregoing is a tall order and there are few ready-made candidates. If you grow your own, the person selected must be able to acknowledge the deficiencies in his own knowledge and be willing to correct them as rapidly as he can even though this task is never-ending.

Where should he report? *Wherever he can function effectively, reasonably free of backstabbing*. The Contracts Dept. is one possibility because he deals with contracts. Engineering is another because he deals with engineering requirements. Program Management is a third because his work influences the program. However, Contracts Departments are often weak. Engineering is often disinterested. And, Program Managers are sometimes given to stifling CM when it threatens their ability to ignore inconvenient requirements. The final choice depends entirely on the makeup of the organization in which the Specialist is to function.

CM Organization

The organization shown in Figure 15 is for those who can't accept the CM Specialist as a stand-alone position. The functions appear to be compatible. The Manager's time splits into 80% as specialist for Requirements and 20% as a supervisor for Release and Change Control.

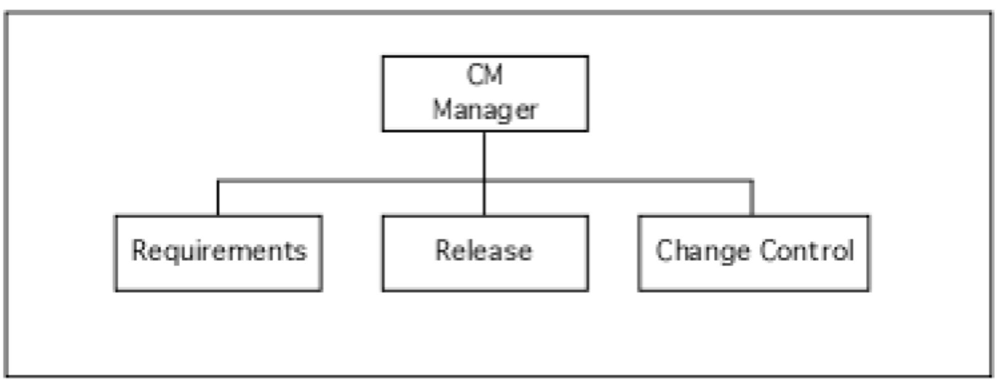

Fig. 15 CM Organization

However, the Achilles Heel is Change Control. It's almost certain that Change Control will swell to dwarf all else. The organization will morph into one where Change Control demands 100% of the time and the rest goes begging.

Summary

The organizational necessities for CM are:

- A reasonably competent engineering-manufacturing organization
- A reasonably effective means of assuring contract compliance.
- A competent CM Specialist

The Specialist will negotiate effective requirements for inclusion in the contract. The contract compliance method will see that the requirements are not ignored. The organization will perform in accord with the requirements.

The question of a reporting home for the CM Specialist will continue to bedevil because there is no really good answer. Therefore, he should report *wherever he can function effectively, reasonably free of backstabbing*. That location will differ from one organization to another and is best left to those who know something about the specifics.

***

11. IS IT REALLY NECESSARY?

Design Drift

Unless you can tolerate design drift, CM is a necessity. If you haven't experienced design drift it's hard to believe that all of this fuss is really necessary. So let's understand it. A cork will float aimlessly pushed hither and yon by currents of wind and water or a good shove. Odd as it may sound designs behave in the same way.

Programs take 5-15 years or more to complete and they take place inside of very large complicated organizations. Control of the design passes through many hands as people quit, retire, die, transfer etc. in both the contractor and customer organizations. This churning of personnel is amplified by the military practice of rotating assignments every three years. Most people are not fond of "writing it down" so they rely on memory which can be tricky. All of this flux makes it easier for zealots to substitute their own judgment to "make it better". All it takes is a few new people plus a foggy collective memory to cause or permit a change in direction. Let that happen several times over the years and a fighter aircraft easily morphs into a fighter-bomber.

Unfortunately, people of uncertain scruples have been known to collude in using an existing program as a test bed for a wholly new product without any authorization but their own. An excess of patriotism has motivated some, others are victims of raw ambition or plain greed, and believe it or not some are real innocents.

None of this is theory. The procurement literature of the 50's and 60's is full of examples and it's still available for review. It convinced the Congress and the military that corrective action was necessary. CM was one of the results.

Decision Criteria

If a specific device must be designed and manufactured to satisfy a particular need, CM or its equivalent is necessary! Many have tried but all have failed to find an equivalent. At best they have reinvented the wheel and given it a different name.

If the device to be designed and manufactured is not dedicated to satisfying a particular need, CM is really not necessary. Precise alignment has little to no significance but some of the elements shown in Figure 1 will still be required to a greater or lesser degree.

***

AFTERWORD

If you understand the elements and relationships in Figure 1, you understand Generic CM. Add the fact that existing personnel, with one exception, can perform all of the tasks required and CM is not so hard after all. The one exception is a CM Specialist needed to define and interpret CM Requirements and he isn't necessary if enough of the existing personnel really understand CM Requirements.

If you don't understand it, review the previous text with some care and thought. Take a look at the separately downloadable Appendix. A number of side issues are covered. This material is not essential to understanding generic CM. However, it is useful background and helps explain people's behavior.

If you still have a problem, I will try to answer *genuine* questions about the content as time and energy allow but I intend to remain fully retired. I can be reached at cmg@citlink.net

On the other hand, if this effort has helped your understanding of CM please tell others. No advertising campaign is planned. If the book has merit word of mouth will do the job.

***