Finding CM in CMMI

Anne Mette Jonassen Hass
DELTA

Configuration management (CM) acts as a central nervous system in system development, and is a prominent key process area and generic practice in Capability Maturity Model® Integration (CMMI®) and other maturity models. As such, implementing and improving CM in the company may seem like an overwhelming task. This article takes you through the requirements in CMMI step by step and offers practical and ready-to-use advice on how to get started and how to get better in this interesting and rewarding area.

Configuration management (CM) acts as a central nervous system in system and software development. If you do not have it, you definitely need it. When you implement it, do it once and very, very carefully. When done, you cannot imagine a life before it.

Every company developing products should consider CM. It is integral to process improvement and the concept of using maturity models to support process improvement, which is becoming more and more integrated in the industry. In the Software Engineering Institute's Capability Maturity Model® (CMM®) Integration℠ (CMMI®), CM plays a prominent part as a key process area at CMMI Level 2 in the staged representation. In both the staged and the continuous representation, CM is a generic goal for all key processes.

Staged representation is similar to CMM V.1.1 where each key process area is assigned to one specific maturity level out of the five defined (initial, managed, defined, quantitatively managed, and optimizing). In continuous representation, each key process area can reach the six defined capability levels (incomplete, performed, managed, defined, quantitatively managed, and optimizing).

CM is, however, not something that most companies are comfortable using. The author has participated in more than 50 assessments in Denmark over the last eight years. In each assessment, a list of the top five candidates for process improvement was produced. The three most frequently appearing processes follow:

1. Project Management (75 percent).
2. CM (55 percent).
3. Test (51 percent).

More than half the companies needed to improve their practices for CM. This made it the second most appearing process.

Introducing CM into a company is, however, not an easy task. Not many controlled experiments on process improvement with the subject of CM exist. For example, under the project Software Improvement Case Studies Initiative (SISSI)[1] – The Business Benefits of Software Best Practice," which has been performed for the European Software Institute, only five out of 50 experiments had CM as their subject [1, 2, 3, 4, 5]. Some of the trends in the conclusions of these reports follow:

> *"The success or failure of a company introducing CM may depend on the people allocated to the task. The person driving the initiative must be a fiery soul or a true believer."*

- Introduction of CM is a great advantage for the company.
- Management support is essential.
- It is important to perform pilot tests of the CM processes before they are rolled out at a greater scale.
- Introduction of CM is difficult.

When a company is facing the task of improving its CM, it may seem like an overwhelming task. Where do you start? Where do you go? How do you get there? Using CMMI for example, a company can get an appraisal of the capability of their CM process. But even without a formal assessment, the definition of the CM Key Process Area may be used as a guideline for implementation and improvement of CM in a company.

CM Benefits

CM brings facts and control to project management. Now, what is that worth? It is hard to say – not many have estimates of the unforeseen problems that come from lack of control and unknown facts.

The savings from a working CM system may stem from preventing one or more of the following problems (among others):

- Work is based on the wrong basis.
- Errors once corrected reappear again.
- Unwanted functionality is introduced, or functionality is forgotten.
- It is difficult or impossible to reestablish a running product.
- It is difficult or impossible to make changes in an existing system.
- It is difficult or impossible to revert to an earlier version that works.
- It is difficult to establish stable decisions because decisions are constantly reversed.

CMMI Definition

The CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing V.1.1 states the following:

> The purpose of CM is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits. [6]

In CMMI, a number of goals are defined for each key process area and a number of practices are defined for each goal (see Table 1 for CM goals and practices).

Getting Started

There are many ways to engage in introducing and improving CM in your organization. No way is definitely *the way* – each company must find its own way. Getting started on anything completely from scratch is not very easy; on the other hand, it is hardly ever necessary. Even if a company has the feeling that no CM is being performed at all, that is very rarely the case.

℠ CMM Integration is a service mark of Carnegie Mellon University.
® CMMI is registered in the U.S. Patent and Trademark Office.

Employees may have knowledge and practical experience of CM gained from previous jobs or during their education.

Getting the Right People

People drive changes and improvements. The success or failure of a company introducing CM may depend on the people allocated to the task. The person driving the initiative must be a fiery soul or a *true believer*. Without a burning desire to see CM at work in the company, the person in charge is prone to give up long before success is achieved. All companies have dedicated and enthusiastic people who want to do as good a job as possible.

Collecting Best Practices

It is a good idea for a company introducing CM to try and create a map of the existing knowledge on the subject within the company itself. Authorized assessments or more informal interviews with employees working on ongoing projects may be used. It is well worth the effort to analyze the information gained from these and use it to collect existing detailed information about practices and knowledge. Starting the CM initiative based on existing practices will enhance the chances of success significantly. Best practices may also be collected from external sources. A good place to start the search is in the "Configuration Management Yellow Pages" on the *CM Crossroads* Web site at <www.cmcrossroads.com/yp>.

Focus

To get CM off the ground, it is important to focus on the employees' understanding of CM and their capabilities to perform CM as on the company's need for control. This should not in any way contradict the organization's interests; at the end of the day, the company depends on the employees' ability to carry out the CM plans.

Look Ahead

At the initial introduction of CM into a company, it is important to use a stepwise approach. Begin with a few steps, and then introduce more advanced aspects in due course. A stepwise approach includes the picking of low-hanging fruit. Set up goals obtainable within a short period of time – two to three months at the most – and celebrate when the goals are reached. However, even in a stepwise approach, it is a good idea to have at least some idea of where the long-term goal is.

The First Steps

The following is a suggestion on how to start CM from virtually nothing.

CM Goals and Practices

SG 1		Establish Baselines
	SP 1.1-1	Identify Configuration Items
	SP 1.2-1	Establish a CM System
	SP 1.3-1	Create or Release Baselines
SG 2		Track and Control Changes
	SP 2.1-1	Track Changes
	SP 2.2-1	Control Changes
SG 3		Establish Integrity
	SP 3.1-1	Establish CM Records
	SP 3.2-1	Perform Configuration Audits
GG 2		Institutionalize a Managed Process
	GP 2.1	Establish an Organizational Policy
	GP 2.2	Plan the Process
	GP 2.3	Provide Resources
	GP 2.4	Assign Responsibility
	GP 2.5	Train People
	GP 2.6	Manage Configurations
	GP 2.7	Identify and Involve Relevant Stakeholders
	GP 2.8	Monitor and Control the Process
	GP 2.9	Objectively Evaluate Adherence
	GP 2.10	Review Status with Higher-Level Management

Note: SG is specific goal for the CM. GG is general goal for all key process areas. SP is specific practice. GP is general practice.

Table 1: *CM Goals and Practices*

Establish Baselines

The primary goal is to get control over the source code and corresponding objects. This may be done by first defining and assigning CM responsibilities in the organization. Appoint at least one CM person responsible as a frontrunner.

Next, define what types of objects to place under CM. These may be individual objects such as specifications, source code modules, other files, etc., or composite objects like subsystems, partial deliveries to the customer, or entire systems. For all these types of objects, a convention for unique identification must be defined, and who has the authority over the objects (e.g., who can approve them) must be defined.

Following this, identify what has already been produced, if anything. The found objects must be stored in a controlled library with relevant information; they become configuration items. A configuration item is any intermediate working product, product component, or product that is placed under CM. Composite items of what have already been produced must also be defined, along with specifying how they are composed.

Lastly, define the rules for making releases of configuration items and to register who has which releases.

Track and Control Changes

When developing and maintaining a product, changes are inevitable. The purpose of change control is to be fully in control of all change requests and all implemented changes; a formal change control must be established for the items now under CM.

Life cycles for incidents and changes must be defined. An incident is every (significant) unplanned event observed (during test), and requiring further investigation according to British Standard 7925-1. The initiation of change control is the occurrence of an incident. It may, for instance, be a wrong formulation in a document, a coding mistake found during a walk-through, an enhancement request arising from a new idea from the customer, or a change required in the code because of the upgrade to a new version of the middleware supporting the system. All incidents must be registered, and for this a template must be produced.

A Change Control Board (CCB) is responsible for the change control of each individual configuration item, so at least one CCB must be established. The CCB is a group of people responsible for assessing, and subsequently approving or rejecting proposed changes to configuration items. *A CCB must be formed according to the* *wanted level of formality. It may consist of the author or producer of an item, a peer group, or a number of managers. A project may have several CCBs with different areas of responsibility.*

When analyzing an incident, the CCB must consider the cost of implementing any changes compared to the cost of not implementing these changes. The CCB decides whether or not the incident should cause one or more changes to configuration item(s). Each change to be made should be documented in a change request issued by the CCB. Make sure all incident registrations are handled by the CCB. Most importantly *do not accept that changes are implemented based on any other input than a change request from the CCB!*

A change process is a miniature development project in itself. Each phase should be described in details stating the responsibility and specific actions. The phases are outlined in Table 2.

Quite often no independent change requests are created. However, this is not a very good idea, especially in the cases where an incident causes changes in several configuration items.

It must be ensured that the CCB approves all implemented changes. All new changed configuration items must be identified and stored in the controlled library with the corresponding information. The last task in the incident life cycle is to provide feedback on decisions to every stakeholder.

Establish Integrity

It is necessary to register data for configuration items such as names, versions, and status, and to store all incident registrations and change requests. Information kept in the CM system is a gold mine of metrics for the project. Make sure to keep relevant information available. Reports such as release notes, item status lists, item history lists, item composition lists, and trace reports should be defined. Audits are not discussed here as the responsibility for these often lies in quality assurance in modern companies.

Good Enough Is Not Always Enough

A CM system as described can be a very good beginning. It will, however, in most cases not be sufficient for a project or a company to get full profit from performing CM. A project well begun is half done, and even a minimal system provides a good starting point for the expansion of CM to a more comprehensive system in order for the benefits to grow. It is, however, important to keep in mind that the scope and the degree of formality must never exceed what is profitable for the company.

Institutionalize a Managed Process

For a key process area to reach a specific level in CMMI terms, the generic practices need to be fulfilled as well as the specific practices. The generic practices are discussed briefly below in view of CM.

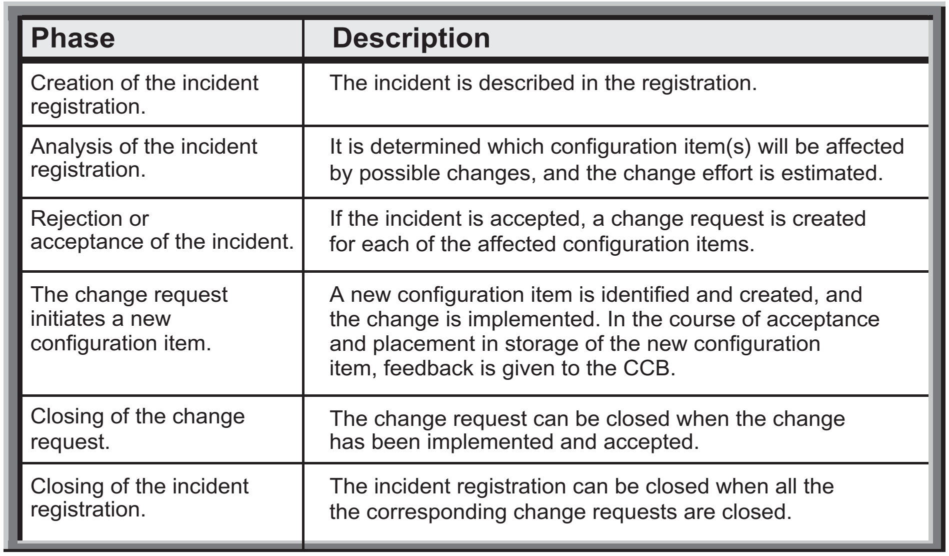
Establish an Organizational Policy

An organizational policy is the responsibility of top management and defines the organization's philosophy towards CM. It must be short and high-level and include a definition of CM to be used as the basis for the CM work in the organization, a description of the CM process to be used, ways of evaluating the CM performance in the organization, and the approach to CM process improvement.

Plan the Process

Even if the CMMI did not promote planning, the starting point for good CM is in planning. If you do not plan, you plan to fail! The work with the plan deepens the understanding of the task and provides the basis for the actual performance of the work to be undertaken.

Table 2: *Change Control Phases*



Phase	Description
Creation of the incident registration.	The incident is described in the registration.
Analysis of the incident registration.	It is determined which configuration item(s) will be affected by possible changes, and the change effort is estimated.
Rejection or acceptance of the incident.	If the incident is accepted, a change request is created for each of the affected configuration items.
The change request initiates a new configuration item.	A new configuration item is identified and created, and the change is implemented. In the course of acceptance and placement in storage of the new configuration item, feedback is given to the CCB.
Closing of the change request.	The change request can be closed when the change has been implemented and accepted.
Closing of the incident registration.	The incident registration can be closed when all the the corresponding change requests are closed.

All the CM activities to be performed must be described in the CM plan for the project. Note that the CM plan may be included as a chapter in the overall project plan if that is preferred. The CM plan does not need to be an independent document, but always make sure that it is easy to recognize, and understand the connections between the CM activities and the other activities in the project. And independent of the size, do not forget to allow for the time to plan the CM.

When planning CM, the purpose, success criteria, and level of ambition must be defined. The plan must be a living document and be used – beware of write-only information. Also, record what has consciously been left out at the point of time when writing the plan. Make sure that all stakeholders, including employees on the project team, are familiar with the plan and willing to adhere to it.

A template is a great help. In many companies, templates for plans in general and even specifically for CM plans exist. Use them! A CM plan may be structured in the following way:

1. Introduction
   1.1 Purpose of the Plan
   1.2 Scope of the CM Task
   1.3 Vocabulary
   1.4 References
2. Management and Relations to the Environment
   2.1 Organization
      Specify how the overall organization of CM is formed, and how this fits into the rest of the project organization. Describe which CM roles are to be filled.
   2.2 Responsibilities
      Define the following clearly and unambiguously:
      - Who is responsible for performing which CM activity?
      - Who is responsible for approval of objects prior to placement under CM?

   2.3 Interface Control
      Define how the interfaces to external objects (software, hardware, etc.) are handled with regard to CM.
   2.4 Subcontractor Management
      Define how new versions are to be tested for approval, who is doing this, and how the deliveries from the subcontractor(s) are introduced into the project in a controlled way.

2.5 Relevant Standards

Describe which guidelines and policies the concrete project adheres to.

3. Activities

3.1 Identification

Describe and plan the activities concerning naming conventions for configuration items and the rest of the CM data (meta data). At the time of the planning, some of this information is typically yet unknown, e.g., module names. In this case, it must be described how and when this missing information will be provided. This section may describe how the information is stored. Detailed techniques may be placed in an appendix.

3.2 Storage and Release

Describe the handling of the controlled library, i.e., how and when the library is built and deployed, and how information is collected. Consider how long the various types of CM information should be kept. Finally, it may be described how necessary backup of the libraries and other information is performed. This is, strictly speaking, not a CM activity, but it is certainly in the interest of CM to ensure that backups are taken, kept in a safe place, and can restore correctly.

3.3 Change Control

This is possibly the section most important to get in place so that an appropriate balance between formalism and flexibility is reached. Define who has the authority to request changes to a configuration item, i.e., who are members of the CCBs for the various object types. A CCB may delegate the authority to individuals or other roles; this should be planned and documented. Define how incident registrations and change requests for various types of objects, e.g., documentation, code, support software and tools, are to be handled, that is, defining the life cycles for incidents and changes. When these procedures are established, it must be ensured that events and changes can be handled fast enough so that CM is not perceived as an unnecessary bottleneck in stressed situations such as important deliveries to customers.

3.4 Status Reporting

Great benefits are to be gained from performing CM; valuable information is easier to find and use. Considerations regarding status reporting provide significant input about which data should be registered and for how long the information should be kept. Use this section to define how status information about the configuration items is collected, treated, and reported on. Also define periodic reports and how ad-hoc or dynamic queries should be handled.

4. Schedule

4.1 Tasks

List the detailed tasks here. Make a special effort not to forget anything – avoid invisible work, i.e., work that the staff has to perform but which is not covered in the plan and the schedule. Also determine and plan required training. Document as far as possible the actual people who will be performing the related tasks. The interfaces or connections to the overall project plan should be stated.

4.2 Milestones

CM has a number of milestones, e.g., the CM system is ready for use for the next activity in the project, the CCB(s) is (are) established, deliveries of subsystem and systems, and product release. List the milestones and state in which phases there is a need for which roles and resources, preferably with a reference to the overall project plan.

4.3 Diagrams and Charts

Support the planning descriptions with diagrams and charts. Follow up on the schedules, i.e., clearly mark on diagrams and charts where the project is, and frequently re-plan so the plan reflects reality.

5. Tools, Techniques, and Methods

Depending on the capability level of the organization, this chapter may be the easiest or the hardest part of the plan to write. If there is no central place to get help, like an organization-wide quality system, and the descriptions in this chapter are to be useful at all, the chapter will be hard to write, possibly fairly big, and it may take a long time. If, on the other hand, general processes exist, simple references and a few descriptions of tailoring may be sufficient. In any case, do not underestimate the importance of this information, and the work involved in providing it.

Provide Resources, Assign Responsibility, Train People

The provision of resources, assignment of responsibility, and training of people are parts of the planning activity. The plan must reflect the actual recourses available for the tasks.

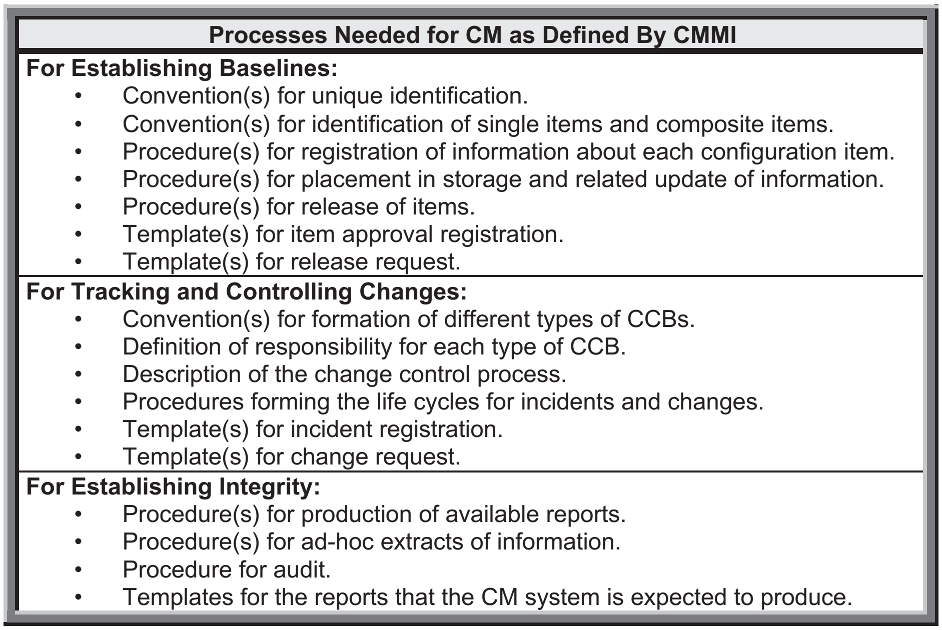
Manage Configurations

This means that products from the CM process must be placed under CM. It is appropriate to place the CM plan under CM, so it should adhere to the general rules of unique identification of objects and be stored in a controlled library, as well as having all changes be controlled in the specified way.

Identify and Involve Relevant Stakeholders

Almost everybody involved in the specification, development, usage, and maintenance of a system is a stakeholder in CM.

Table 3: *Processes Needed for CM as Defined By CMMI*



Processes Needed for CM as Defined By CMMI
For Establishing Baselines: • Convention(s) for unique identification. • Convention(s) for identification of single items and composite items. • Procedure(s) for registration of information about each configuration item. • Procedure(s) for placement in storage and related update of information. • Procedure(s) for release of items. • Template(s) for item approval registration. • Template(s) for release request.
For Tracking and Controlling Changes: • Convention(s) for formation of different types of CCBs. • Definition of responsibility for each type of CCB. • Description of the change control process. • Procedures forming the life cycles for incidents and changes. • Template(s) for incident registration. • Template(s) for change request.
For Establishing Integrity: • Procedure(s) for production of available reports. • Procedure(s) for ad-hoc extracts of information. • Procedure for audit. • Templates for the reports that the CM system is expected to produce.

Make a list and make it comprehensive – remember everybody.

Monitor and Control the Process

To be able to monitor and control processes, they must be understood and defined. The processes needed for CM as defined by CMMI in terms of procedures, conventions, and templates are listed in Table 3.

The fact that CM is performed during the entire lifetime of a product, for a number of different types of objects, and under various circumstances, poses specific requirements on the process descriptions for CM. It may be necessary to produce more variants of some of the processes, depending on the types of configuration items to handle, and/or the degree of formalism required. Types of configuration items, to mention a few, may be requirement specifications, source code, and plans.

Metrics for Controlling CM Performance

Tom Gilb said in "Principles of Software Engineering Management," that everything can be made measurable in a way that is better than not measuring at all [7]. Measurements may show new aspects of things you thought you knew everything about.

The following are suggestions for metrics that may be used for analyzing how CM is performed in a company. The CM processes are in focus here, not other processes and not the product. The list is by no means exhaustive.

- Number of registrations of configuration items in the CM system.
- Time interval in which the registrations have taken place.
- Time used for each individual registration.
- Incidents in connection with registrations.
- Incident rate for registrations, e.g., number of erroneous registrations per 100.
- Average time for registration.

Some of the metrics may be defined by configuration item type. Identical metrics may be defined for the following:

- Placement in storage.
- Releases.
- Handling of incident registrations.
- Handling of change requests.
- Completions of milestones defined in the CM plan.

Metrics may also be defined to include costs such as the cost of the activities.

There is no reason for collecting measurements if no one is going to analyze them or act according to the analysis results. Measurements must be collected over time for the balance point and the variation to be calculated, for example.

Process control is to find reasons for variations from the normal. Control over processes is gained by constantly analyzing new measurements in relation to the expected values, so processes that start to behave in an unexpected way can be identified. The reason for the unexpected behavior must be investigated and possible irregularities identified.

An unexpected behavior may be seen by a measurement that suddenly lies far outside the normal variation. This could be to either the positive or negative side. An example might be the handling time for an incident that is suddenly far faster than normal – is this because the CCB did not do their work properly, is it an incident that they have handled a lot of times before, or is there a totally different reason?

Process improvement is to change the reasons why something is considered normal. In conducting process improvement, you may ask yourself questions like the following:

- Why is the balance point where it is?
- How may the balance point be moved?
- Why does the variation have the size it has?
- How may the variation be decreased?

A balance point may be the average time it takes for an incident registration of a certain type to be handled. It may be worthwhile finding out if there are bottlenecks in the handling process that might be eliminated to decrease the average handling time.

Objectively Evaluate Adherence and Review Status With Upper Management

These practices are the responsibility of quality assurance and management and will not be discussed here.

Good Luck With CM

CM is not easy! If you think it is, you will be unable to solve the CM task in a professional way. CM is not difficult! All you have to do is to do it; if you understand the discipline, it is much easier to specify and plan the task so that it fulfills its purpose and becomes manageable.◆

References

1. Del Duca, Gianfranco. "Introduction of CM: Gaining a Competitive Edge." <www.esi.es/VASIE/Reports/All/24151/14.pdf>.
2. Soro, Roberto. "Applying GQM to Assess CM Practice for Better Interbank Services." <www.esi.es/VASIE/Reports/All/24151/sia.pdf>.
3. Garella, Alberto, and Giuseppina Tallo. "Configuration and Change Management to Rising Quality of Service." <www.esi.es/VASIE/Reports/All/24151/istiserv.pdf>.
4. Vidvei, Tor. "Introduction of a CM in Very Small Organizations." <www.esi.es/VASIE/Reports/All/24151/40.pdf>.
5. Roald, Helge M. "Introduction of a Common CM Framework." <www.esi.es/VASIE/Reports/All/24151/52.pdf>.
6. CMMI Product Team. Capability Maturity Model® Integration (CMMI SM), V.1.1. Pittsburgh, PA: Software Engineering Institute, Mar. 2002 <www.sei.cmu.edu/cmmi/models/model-components-word.html>.
7. Gilb, Tom. Principles of Software Engineering Management. Addison-Wesley, 1988.

Note

1. A description of the SISSI project, "Process Improvement Experiment Final Report," is at <www.esi.es/en/Projects/VASIE/Reports/All/21379/Report/21379.pdf>.

About the Author

Anne Mette Jonassen Hass, Mc.Sc.C.E., has been a consultant for DELTA since 1995, where she is now partner. She has 25 years experience in information technology (IT) and has performed more than 40 assessments. Hass has an Information Systems Examinations Board certificate in Software Testing Foundation and Practitioner, and is a certified BOOTSTRAP V.3.0 and Capability Maturity Model® Integration assessor. She is author of "CM Principles and Practice" and "Requirements Development and Management," and developed "Process Contest," a game that teaches IT concepts in a relaxed and entertaining way.

DELTA
Venlighedsvej 4
2970 Hoersholm
Denmark
Phone: +45 72 19 40 00
Direct: +45 72 19 44 23
Fax: +45 72 19 40 01
E-mail: amj@delta.dk