

Configuration Management and ISO 9001

Robert Bamford, William J. Deibler II
Software Systems Quality Consulting

The software engineering practices associated with software configuration management (SCM or CM) offer a number of opportunities to address requirements found in the International Standard, ISO 9001. From a management perspective, the principles and practices of CM represent an accepted and understood foundation for implementing ISO-compliant processes in software engineering organizations. In addition, the growing number of tools for automating CM practices are avenues for improving the efficiency and effectiveness of these processes.

This article begins with brief, general definitions of configuration management and of ISO 9001.

Configuration Management

While there is no single definition of CM, there are three widely disseminated views from three different sources: the Institute of Electrical and Electronics Engineers (IEEE), The International Organization for Standardization (ISO), and the Software Engineering Institute (SEI) at Carnegie Mellon University.

The IEEE perspective on CM

A most widely understood description of the practices associated with configuration management is found in the IEEE Standard 828-1990, *Software Configuration Management Plans*¹:

[Numbers in brackets are added]

"SCM activities are traditionally grouped into four functions: [1] configuration identification, [2] configuration control, [3] status accounting, and [4] configuration audits and reviews."

IEEE Standard 828-1990 goes on to list specific activities associated with each of the four functions (the number of the paragraph containing the reference appears in parentheses):

- ◆ **Identification:** identify, name, and describe the documented physical and functional characteristics of the code, specifications, design,

and data elements to be controlled for the project. (Paragraph 2.3.1)

- ◆ **Control:** request, evaluate, approve or disapprove, and implement changes (Paragraph 2.3.2)
- ◆ **Status accounting:** record and report the status of project configuration items [initial approved version, status of requested changes, implementation status of approved changes] (Paragraph 2.3.3)
- ◆ **Audits and reviews:** determine to what extent the actual configuration item reflects the required physical and functional characteristics (Paragraph 2.3.4)

This list is similar to the set of activities noted by Pressman²:

"Software configuration management is an umbrella activity ... developed to (1) identify change, (2) control change, (3) ensure that change is being properly implemented, and (4) report change to others who may have an interest."

The ISO perspective on CM

In the guideline document, ISO 9000-3:1991 *Guidelines for the application of ISO 9001 to the development, supply and maintenance of software*, the International Organization for Standardization identifies a similar set of practices as CM³.

"Configuration management provides a mechanism for identifying, controlling and tracking the versions of each software item. In many cases earlier versions still in use must also be maintained and controlled.

- "The [CM] system should
- "a) identify uniquely the versions of each software item;
 - "b) identify the versions of each software item which together constitute a specific version of a complete product;
 - "c) identify the build status of software products in development or delivered and installed;
 - "d) control simultaneous updating of a given software item by more than one person;
 - "e) provide coordination for the updating of multiple products in one or more locations as required;
 - "f) identify and track all actions and changes resulting from a change request, from initiation ... to release."

The SEI perspective on CM

Based on a review of currently available tools and an evolving understanding of the organizational role of CM, the SEI advocates a broader definition of CM in SEI-92-TR-8⁴:

"The standard definition for CM taken from IEEE standard 729-1983 [updated as IEEE Std 610.12-1990] includes:

"Identification: identifying the structure of the product, its components and their type, and making them unique and accessible in some form

"Control: controlling the release of product and changes to it throughout the life cycle ...

"Status Accounting: recording and reporting the status of components and change requests, and gathering vital statistics about components in the product

"Audit and review: validating the completeness of a product and maintaining consistency among the components ...

"[The IEEE] definition of CM ... needs to be broadened to encompass ... :

"Manufacturing: managing the construction and building of the product

"Process management: ensuring the correct execution of the organization's procedures, policies, and life-cycle model

"Team work: controlling the work and interactions between multiple developers on a product."

ISO 9001

In 1987, the International Organization for Standardization in Geneva Switzerland published **ISO 9001, Quality Systems - Model for quality assurance in design / development, production, installation, and servicing**⁵. ISO 9001 is the most comprehensive model in the ISO 9000 series of standards. It describes a minimum set of activities found in companies and organizations that consistently produce products that satisfy customer requirements. The policies, procedures, standards, records, and associated business activities are *the quality system*. While ISO 9001 is written to describe any company providing any product or service, it tends to employ manufacturing terminology, which must be interpreted for

non-manufacturing environments, including service and software providers.

To ensure a uniform interpretation of ISO 9001 for software engineering organizations, ISO published **ISO 9000-3, Guidelines for the Application of ISO 9001 to the development, supply and maintenance of software.**

The key issues ISO 9000-3 addresses are that:

- ◆ Product exists earlier in software (during design and development)
- ◆ Software product can be proliferated easily

Focusing on these issues mirrors the guidance in Clause 7.4 of ISO 9000-1:1994⁶ :

The process of development, supply, and maintenance of software is different from that of most other types of industrial products in that there is no distinct manufacturing phase. Software does not “wear out” and, consequently, quality activities during the design phase are of paramount importance to the final quality of the product.

Note that ISO 9000-1 and ISO 9000-3 provides guidance. ISO 9001 is the only source of the requirements against which compliance in software engineering practices is assessed.

ISO 9001 and Configuration Management

Tracing the relationship between ISO 9001's requirements and CM practices begins with an examination of the guidance in ISO 9000-3.

ISO 9000-3 and configuration management

ISO 9000-3 contains two appendices, Annex A and Annex B, that provide cross references between ISO 9001 and ISO 9000-3. According to Annex A, five sections of ISO 9001 correlate to ISO 9000-3, Paragraph 6.1, Configuration Management:

- ◆ 4.4 Design control
- ◆ 4.5 Document data control
- ◆ 4.8 Product identification and traceability
- ◆ 4.12 Inspection and test status
- ◆ 4.13 Control of nonconforming product

Each of these sections of ISO 9001 contains a portion of the traditional CM process.

4.4 *Design control* addresses all of the steps in the software development life cycle: planning, specification, design, coding, testing. Section 4.4 requires that design inputs and outputs be documented, reviewed, verified, controlled, approved, and modified according to documented procedures. *Design inputs* and *outputs* include plans (project life cycle definition), specifications, prototypes, requirements documents, progress reports, review results, test plans, test cases/scripts, development tools, code, and test reports.

ISO 9001 4.4.9 *Design changes*, in conjunction with ISO 9001 4.14.2 *Corrective action*, and 4.13 *Control of nonconforming product*, requires that each change be traceable to an appropriate source and approval. For software product there should be a clear path between a change request spawned by a fault report or enhancement request and a change in a specific product component to correct the fault or to implement the enhancement. An interested party should be able to pick up the path at any point and follow it forward to the released change and backward to the change request or to the fault report. 4.5 *Document and data control* addresses the identification, protection, approval, and availability of current issues of all pertinent product- and project-related documents, including designs, specifications, plans, and schedules.

Because a fundamental function of CM is making current configuration items available, the CM practices and tools can be applied to the control of product- and process-related documentation and data.

4.8 *Product identification and traceability* requires that each version of a configuration item be identified by some appropriate means.

4.12 *Inspection and Test Status* requires procedures to identify what verification steps and tests have been completed and what results have been achieved by the product or product

components at each phase in the defined development life cycle.

4.13 *Control of Nonconforming Product* requires procedures to ensure that untested, defective, or incorrect versions (e.g., down level) of the product are not *inadvertently* used. This paragraph of ISO 9001 also requires a procedure to determine the disposition of nonconforming product at all stages.

For software, the bulk of the activity related to non-conforming product is in the correction of faults identified during all phases of development (e.g., during requirements definition, prototyping, integration testing, and beta testing) and after the product has been released (e.g., customer reported faults).

Beyond ISO 9000-3

There are a significant number of additional areas of ISO 9001 that can be addressed through CM-related activities.

- ◆ 4.1 Management responsibility
- ◆ 4.2 Quality system
- ◆ 4.4.2.2 Organizational and technical interfaces
- ◆ 4.6 Purchasing
- ◆ 4.7 Control of customer-supplied product
- ◆ 4.9 Process control
- ◆ 4.14 Corrective and preventive action
- ◆ 4.15 Handling, storage, packaging, preservation, and delivery
- ◆ 4.16 Control of quality records
- ◆ 4.19 Servicing
- ◆ 4.20 Statistical techniques

4.1 Management responsibility

Reports produced by the CM system on progress and exceptions support management review of the suitability and effectiveness of the development practices, as part of the quality system (ISO 9001 4.1.3).

4.2 Quality system

For a software engineering organization, the CM policies, procedures, and standards represent a significant portion of the quality system. Tools to support and automate the CM process support and enforce adherence to policies, procedures, and standards.

These procedures can also be automated through integrated workflow and groupware tools that increase the effectiveness and efficiency of the information exchange.

4.4 Design control

CM practices can go beyond control of configuration items to ensure that necessary information regarding status and change is communicated to appropriate individuals and organizations (ISO 9001 4.4.2.2).

Standard distribution lists and notification procedures linked to specific activities prevent significant missed communication.

4.6 Purchasing

The primary application of this paragraph of ISO 9001 in software engineering environments is to third-party development. When an organization subcontracts software development to a third party, evaluation of the subcontractor's CM practices is a critical component of the vendor selection and approval process.

If appropriate, the subcontractor can be required to follow or implement specific CM practices.

Intermediate or final product and all related documentation (e.g., specifications, plans, progress reports, test reports) received from the subcontractor can all be treated as if they were in-house developed configuration items. Applying CM practices to third-party development is of particular benefit for coordinating in-house integration and verification activities and for fault resolution.

4.7 Control of customer-supplied product

In software development, customer-supplied product includes software that is used in the development process or that is included in the product to be delivered to the customer. This software is specified by the customer and supplied by the customer or by a third party; the software can be a standard, off-the-shelf (shrink-wrapped) product or one that is custom developed.

Depending on how the included software is packaged and distributed, ISO requirements to verify, store, and

maintain this software appropriately may met by considering the included software as a configuration item. Requirements for the verification of customer-supplied product apply only to those portions of the customer-supplied product that are used in conjunction with the supplier's product. For example, if the customer specifies that the supplier's product is to run under UNIX System 5, the supplier's responsibility is to verify that the developed product works as specified under UNIX System 5. The supplier must identify and report any errors in UNIX System 5 that impact the operation of the supplier's product.

4.9 Process control

Process control, in conjunction with 4.4.2 *Design and development planning*, 4.2.2 *Quality system procedures*, and 4.2.3 *Quality planning* clarifies ISO 9001's implicit requirements for documented procedures, suitable production equipment, monitoring and control of process and product characteristics, and approval of processes and equipment.

In software engineering environments, the project management and CM processes combine to address the majority of these requirements.

By automating the product build process, a CM system contributes significantly to the effectiveness and efficiency of the software production process, both for intermediate versions of the product and for a released version. This becomes particularly significant, when multiple versions of the product are being developed or maintained in parallel.

4.14 Corrective and preventive action

A significant portion of corrective action is creating the mechanism to ensure that customer-reported problems are resolved in an appropriate manner. The same requirements pertain to problems identified in the development process, starting from the point at which the software product or item comes under CM control.

Incidents must be tracked from report, through classification, and, if appropriate, to resulting changes in the product.

CM practices, particularly those related to change management, product maintenance, and status accounting, ensure that incidents that result in product change are always handled properly.

There is significant opportunity for improving the efficiency of product support and software engineering organizations by minimizing the amount of manual intervention and effort in moving information between the problem tracking and the CM systems.

4.15 Handling, storage, packaging, preservation, and delivery

For software product, the CM practices address all of the handling, storage, packaging, preservation, and delivery requirements at least to the point where responsibility for the product is turned over to software production.

ISO 9001 4.15.2 *Handling* specifies "methods of handling product that prevent damage or deterioration".

For software product, this requirement is interpreted to include activities like virus checking if an outside replication vendor is used and off-site storage of product masters as a minimum level of disaster recovery.

Automated support for the build process, included in most CM tools, reduces opportunities for error and can increase confidence in intermediate test results and in final product integrity.

4.16 Control of quality records

In ISO 9000, quality records are the records that establish that processes were followed and that quality requirements were met. By definition, *quality records* includes records of:

- ◆ Product identification
- ◆ Non-conformity review and disposition
- ◆ All verification and validation activities, including: design review minutes, test logs and records, and fault reports

While these records are not documents (and are not subject to the requirements of ISO 9001 Paragraph 4.5), requirements for *identification*,

collection, indexing, filing, storage, maintenance, and disposition of quality records can be addressed through procedures implemented as part of the CM system.

4.19 Servicing

ISO 9000-3 ties *servicing* to all aspects of software maintenance, including problem resolution, interface modification (e.g., support for additional or modified hardware components), functional expansion or performance improvement.

CM practices ensure that the product is maintained in an orderly manner and that each approved change can be prioritized, tracked, and managed to completion.

Analysis of the data in the CM system related to all aspects of product maintenance can support systematic prioritization and planning for product and process enhancement (e.g., *What modules change most often? What modules cause the most problems? Is the effectiveness of testing continuing to improve?*)

4.20 Statistical techniques

While ISO 9001 contains no specific requirements for statistical process control, as noted above, CM-related activities generate a wealth of process and product data for analysis and comparison to plan: delivery dates, resources, benchmarking (e.g., lines of source code, executable size, performance), time to correct defects, etc.

Even if no *modern statistical methods* are implemented (e.g., Statistical Process Control, Design of Experiments, as suggested in Clause 20 in ISO 9004-1:1994⁷), this data is considered input for ISO 9001 4.9d, which requires “monitoring and control of suitable process parameters and product characteristics”.

The data in the CM system is a primary input for problem analysis and the identification of root causes in products and processes.

Summary - ISO 9001 and configuration management

As described in the preceding paragraphs, of the 20 sections of ISO 9001 that define a supplier's capabil-

ity to meet customer requirements, CM practices directly impact the following: Check marks in the following table indicate clauses of ISO 9001 that are addressed by CM practices.

Section of ISO 9001		
4.1	Management responsibility	✓
4.2	Quality system	✓
4.3	Contract review	
4.4	Design control	✓
4.5	Document and data control	✓
4.6	Purchasing	✓
4.7	Control of customer-supplied product	✓
4.8	Product identification and traceability	✓
4.9	Process control	✓
4.10	Inspection and testing	
4.11	Control of inspection, measuring and test equipment	
4.12	Inspection and test status	✓
4.13	Control of nonconforming product	✓
4.14	Corrective and preventive action	✓
4.15	Handling, storage, packaging, preservation and delivery	✓
4.16	Control of quality records	✓
4.17	Internal quality audits	
4.18	Training	
4.19	Servicing	✓
4.20	Statistical techniques	✓

In terms of improved efficiency, major opportunities exist in ensuring that the CM, project management, customer technical support, build management, and problem reporting and tracking systems are as tightly coupled as possible.

Beyond CM - Product Attributes and Tool Selection

Nothing in ISO 9001 requires that a specific tool or technology be employed. ISO's sole concern is that the implemented systems be *effective* in delivering the promised product or service. Tools that automate CM practices may improve the effectiveness and will improve the *efficiency* of systems that require significant manual intervention.

From an ISO implementation perspective, automation represents an opportunity to:

- ◆ Reduce process and project documentation
- ◆ Reduce requirements for training

- ◆ Ensure that required steps are completed
- ◆ Record progress and activity

The tool selection process should ensure that the selected tools support the current or planned software engineering practices. While some minor changes to engineering practices may be required (especially if standard tools with minimum customization are selected), the tools cannot be the basis for formulating engineering practices.

Considerations in tool selection

While a number of factors determine the appropriateness of a particular tool, the following is an initial list of information that is required to evaluate a tool for suitability.

- ◆ Product
 - Number of products
 - Shared/common components
 - Application complexity
 - Number of platforms supported; number of platforms on which development is performed
 - Maintenance of multiple versions (multiple platforms, application variants)
- ◆ Project
 - Size of project
 - Need to maintain, control, adapt, extend, etc. - project or product
 - Risks associated with the product, project, and related commitments
- ◆ Process
 - Code structure and techniques
 - Concurrent development
 - Paradigm (configuration items, developers' requirements for access, managers' requirements for control and information)
 - Stability and flexibility
- ◆ People
 - Organization size and experience
 - Capacity of the organization to adapt
- ◆ Existing tools that will be retained
 - In the engineering organization
 - In organizations that interface with engineering
- ◆ Integration with other tools under consideration

- Project management
- Build management
- Customer technical support
- Problem reporting and tracking

Based on Feiler's characterization of current CM tools⁸, certain aspects of tool functionality and process automation emerge as key differentiators among the competing models and tools.

- ◆ Merging and propagation of changes (to support multiple versions, concurrent development, shared/common components)
 - Among parallel versions
 - Between branches and main-stream or shared core
- ◆ Restoration of previous versions
- ◆ Identification and control of changes
- ◆ Identification and control of product and product components
- ◆ Automated build management

With this information, the ability of a tool to support a particular organization and its development practices can be evaluated objectively and any requirements for immediate or future customization can be defined.

Bibliography and Recommended Reading

- [Ba1] Wayne A. Babich, **Software Configuration Management**, Addison Wesley Publishing Company, Reading MA, 1986
- [Bb1] R.C. Bamford and W.J. Deibler, *Comparing, contrasting ISO 9001 and the SEI capability maturity model*, **IEEE Computer**, October 1993, Vol 26, No. 10, IEEE Computer Society, page 68
- [Bb2] R.C. Bamford and W.J. Deibler, **A Detailed Comparison of the SEI Software Maturity Levels and Technology Stages to the Requirements for ISO 9001 Registration**, Software Systems Quality Consulting, San Jose, Calif., 1993.
- [Bb3] R.C. Bamford and W.J. Deibler, **MKS RCS Version 6.2 and ISO 9001**, Software Systems Quality Consulting, San Jose, Calif., 1993.
- [Bb4] R.C. Bamford and W.J. Deibler, **ISO Implementation as a Managed Process - A Software Perspective**, Software Systems Quality Consulting, San Jose, Calif., 1993.
- [Da1] Susan A. Dart, **The Past, Present, and Future of Configuration Management**, CMU/SEI-92-TR-8, Software Engineering Institute, Carnegie Mellon University, Pittsburgh

- PA, July 1992 (available by anonymous ftp from ftp.sei.cmu.edu)
- [Fe1] Peter H. Feiler, **Configuration Management Models in Commercial Environments**, CMU/SEI-91-TR-7, Software Engineering Institute, Carnegie Mellon University, Pittsburgh PA, March 1991 (available by anonymous ftp from ftp.sei.cmu.edu)
- [Hu1] Watts S. Humphrey, **Managing the Software Process**, Addison Wesley Publishing Company, New York, August 1990 (Reprinted with corrections)
- [Ie1] IEEE Std 828-1990, **IEEE Standard for Software Configuration Management Plans**, IEEE, 345 E. 47th St., New York, NY (from the Spring 1991 Software Engineering Standards Collection, April 5, 1991)
- [Ie2] IEEE Std 610.12-1990, **IEEE Standard Glossary of Software Engineering Terminology**, IEEE, 345 E. 47th St., New York, NY (from the Spring 1991 Software Engineering Standards Collection, April 5, 1991)
- [In1] ISO 9001:1987 **Model for design/development, production, installation and servicing**, International Organization for Standardization, Geneva, Switzerland, 1987
- [In2] ISO 9000-3:1991 **Guidelines for the application of ISO 9001 to the development, supply and maintenance of software**, International Organization for Standardization, Geneva, Switzerland, 1991
- [In3] ISO 9000-1:1994 **Quality management and quality assurance standards - Guidelines for selection and use**, International Organization for Standardization, Geneva, Switzerland, 1994
- [In4] ISO 9004-1:1994 **Quality management and quality system elements - Guidelines**, International Organization for Standardization, Geneva, Switzerland, 1991
- [Pa1] Mark C. Paulk et al., **Capability Maturity Model for Software**, Version 1.1, CMU/SEI-93-TR-24, Software Engineering Institute, Carnegie Mellon University, Pittsburgh PA 15213, February 1993
- [Pa2] Mark C. Paulk et al., **Key Practices of the Capability Maturity Model**, Version 1.1, CMU/SEI-93-TR-25, Software Engineering Institute, Carnegie Mellon University, Pittsburgh PA 15213, March 1993
- [Pr1] Roger S. Pressman, **A Manager's Guide to Software Engineering**, McGraw-Hill, Inc., New York, 1993
- [Wh1] David Whitgift, **Methods and Tools for Software Configuration Management**, John Wiley & Sons, New York, 1992

Footnotes

- ¹ [Ie1], paragraph 2.3
- ² [Pr1], page 350
- ³ [In2], paragraph 6.1.1
- ⁴ [Da1], page 1

- ⁵ [In1]
- ⁶ [In3]
- ⁷ [In4]
- ⁸ [Fe1]

About the Authors

Robert Bamford and William J. Deibler II are the principals of SSQC. Founded in 1990 in San Jose, CA, SSQC is uniquely positioned to help software and hardware developers in six related areas:

- ◆ CMM and ISO 9001 Hybrid Multi-Model (HM2) Appraisal and Implementation
- ◆ Engineering Life Cycle Definition
- ◆ Education and Training
- ◆ Software Quality Assurance and Testing
- ◆ Business Process Reengineering and Benchmarking
- ◆ ISO 9000 Registration

SSQC offers HM2, a unique, hybrid appraisal method that defines and correlates the position of an organization with respect to both ISO 9001 and the CMM. The results of an HM2 assessment are a plan and framework for improving software engineering processes and for implementing the requirements of the two models.

Drawing from experience in obtaining and maintaining one of the first ISO registrations for software in the US, SSQC offers an extensive array of on-site and public training, on-site assessment and implementation, and CMM- and ISO-related publications and tools. Active in the Quality community through participation in ANSI/RAB and ASQC and ISO/IEC committees, the principals of SSQC have published and presented numerous articles and papers on ISO 9000, the SEI Capability Maturity Model (CMM) and other quality-related topics.

Bill has an M.Sc. in Computer Science and 20 years experience in the computer industry, primarily in the areas of software and systems quality assurance, validation, and development. Since 1986, Bill has held Engineering management positions at National Semiconductor Corporation (Systems Division) and International Computers Limited (ICL). As Sys-

tems Quality Manager, he architected and implemented a quality system for ICL's Santa Clara Hardware and Software Development Center. The quality system was registered to ISO 9001 in 1990, one of the earliest registrations in the US.. Until he left ICL in 1993 to become a full-time partner in SSQC, Bill was responsible for maintaining ICL's registration through three years of surveillance assessments.

With a BS and MAT in Mathematics, Bob Bamford spent 8 years teaching at the secondary and university levels. Since 1975, he has managed training development, technical publications, professional services, and third-party software development for National Semiconductor Corporation (Systems Division) and International Computers Limited (ICL). In 1987, he became involved in the implementation of a Crosby-based Total Quality Management System, facilitating quality courses, managing the education team, and serving on the corporate quality council.

In 1990, he participated in ICL's successful initiative to achieve ISO 9001 registration for its Santa Clara hardware and software development center. Bob facilitated the teams defining and documenting the company's complete processes. From 1990 until he left ICL, he managed teams of auditors conducting semiannual value-added internal audits of departments and projects.

During their affiliation with SSQC, Bob and Bill have developed and published numerous courses, auditing tools, and articles on interpreting and applying the ISO 9000 standards and guidelines. Their articles have appeared in CEEM's Quality Systems Update, IEEE Computer, and Software Marketing Journal. They have presented research papers at numerous national and international conferences, including those sponsored by the American Society for Quality Control (ASQC), the Software Publishers Association (SPA), the Education Foundation of the Data Processing Management Association (EFDPM), the Software Engineer-

ing Institute (SEI) and Software Research Inc. (SRI).

Their courses have been attended by hardware and software engineering and manufacturing and service professionals from many of the country's leading technology companies. Their courses have been sponsored for their members by professional associations, including the ASQC, CSU Long Beach's Software Engineering Forum for Training, Semiconductor Equipment and Materials International (SEMI), Software Engineering Institute (SEI), UC Berkeley and UC Santa Cruz.

Both Bob and Bill regularly perform independent assessments of organizations and work with companies to assist them in their engineering practices and quality process definition. They also perform ISO 9000 registration audits as external resources under contract to the British Standards Institution (BSi).

They were active members of the Software Quality System Registration (SQSR) committee of the ANSI/ASQC Registrar Accreditation Board (RAB). This committee developed the US equivalent of the UK TickIT program and guideline documentation. They were principal authors and editors of "A Guide to Software Quality System Registration under ISO 9001".

They are active United States TAG members in the ISO/IEC JTC1 SC7 - Software Engineering Standards subcommittee.

You can contact SSQC at:

2269 Sunny Vista Drive
San Jose CA 95128
Voice 408-985-4476
FAX 408-248-7772
Internet: ssqc@concentric.net
<http://www.ssqc.com>

-
- 1 [Ie1], paragraph 2.3
 - 2 [Pr1], page 350
 - 3 [In2], paragraph 6.1.1
 - 4 [Da1], page 1
 - 5 [In1]
 - 6 [In3]
 - 7 [In4]
 - 8 [Fe1]