
DOC TYPE: Summary Of Voting
TITLE: Summary of Voting
PDTR 15271 Guide to ISO/IEC 12207 Software Life-Cycle
Processes
SOURCE: SC7 Secretariat
PROJECT: 07.26
STATUS: PDTR Ballot
REFERENCES: N1492
ACTION ID: FYI or ACT.
DUE DATE:
DISTRIBUTION: Electronic: SC7_AG; Disk: P, O and L
MEDIUM: E-Mail - uuencoded Word file; Diskette - Word file
NO. OF PAGES: 25
DISK: 7

Jean-Normand Drouin
Francois Coallier

ISO/IEC JTC1/SC7 Secretariat

ISO/IEC JTC1/SC7 N1528
SUMMARY OF VOTING

Project: 07.26
Subject: PDTR 15271 Guide to ISO/IEC 12207 Software Life-Cycle
Processes
Reference: N 1492 Ballot: PDTR Ballot
Circulation date: 1996 03 01 Closing date: 1996 06 10
Circulated to: P, O, L members Circulated by: SC7 Secretariat

P MEMBERS APPROVING (* = WITH COMMENTS)

CANADA*, CZECK REPUBLIC*, FINLAND, GERMANY, ITALY, KOREA, MEXICO*,
NETHERLANDS, NORWAY, RUSSIAN FEDERATION, SOUTH AFRICA*, SWEDEN

P MEMBERS DISAPPROVING (* = WITH COMMENTS)

AUSTRALIA*, JAPAN*, UK*, USA*

P MEMBERS ABSTAINING

P MEMBERS NOT VOTING

BELGIUM, DENMARK, HUNGARY, IRELAND, ISRAEL, ROMANIA, SINGAPORE, SPAIN,
THAILAND, UKRAINE

O MEMBERS AND LAISIONS (* = WITH COMMENTS)

LATE VOTES
=====

BRAZIL* (approves), FRANCE (abstains)

AUSTRALIA

Australia votes with disapproval and provides the following comment:

AUS-1

For completeness the following Section should be added after the existing Section 2.5 and the subsequent Sections renumbered to conform.

This will clear up a lot of the confusion surrounding the relationship between supporting processes and the way in which these processes are managed by a project.

2.6. Management of Supporting Processes

2.6.1 General.

When supporting processes (such as CM, Documentation or QA) are involved in a project and are required to be managed at the project level, management of the supporting process should develop plans and, if required by either project management or organisational management, these plans should be documented

2.6.2 Plans.

The supporting process plans should be subordinate to the project management plan and may in that context stand alone or preferably be appended to the project management plan. Documented plans should be approved by project management and placed under change control. They should be updated as the project plan is updated, with particular emphasis on schedule and budget implications. Supporting process plans should contain, as a minimum:

- the activities and tasks to be performed
- the means of initiation - by direction or time-based resourcing
- cost reporting
- for extended tasks, the means of reporting progress
- problem or exception reporting
- completion criteria
- completion reporting.

2.6.3 Reporting.

All reporting on activities related to supporting processes should be directed to project management either directly or through the organisational management depending on existing or contract specific policies and procedures. Problem or exception reports should be brought to the attention of project management where they may be analysed for impact on the project in terms of: cost, schedule, functionality included, and quality. There should exist a mechanism for conflict resolution or escalation so that any disputes between supporting process management and project management may be resolved by an appropriately authorised level of organisational management.

2.6.4 Progress.

Where -milestones- are in place and achievement of the -milestone- is dependent upon one or more reports from supporting processes, it is important that these achievements are reported in an accurate and timely manner in accordance with the approved plans. As it is not uncommon for -milestones- to be contractually linked to performance of supporting processes (for example -achievement of a particular baseline-), it is essential that plans be synchronised, and project management should be made aware as soon as possible of any difficulties being experienced by the supporting process in completing tasks related to the -milestones-.

2.6.5 Sub-contracted Supporting Processes.

Whenever supporting processes are performed by organisations outside the organisational control of the project management, it is important to realise that there are two sets of relationships in place: those between the supported project management and the supporting process management, those between the supported and supporting organisational managements.

Agreements or contracts should recognize this when considering aspects of planning, implementation, control and reporting and so clearly define technical and management reporting, information flow, and dispute resolution. Synchronisation of plans may be more difficult under sub-contract agreements and tasking.

BRAZIL

Brazil would like to have the following comments registered.

Comments :

Item 2.2.1 e) We didn't understand this item. It is not so clear.

Item 2.9 Processes and documentation

... plans, specifications, or test requirements. However this guide addresses some of the contents to compose certain documents. See Annex B for details of documentation requirements.

Item 3.1 - to include in the figure 1 c) Determine the project environment and characteristics.

Item 6 figure 4 - The business processes in the organization many times are integrated, so the figure needs to be change.

I suggest the inclusion of two new documents, in Annex B, page 25, as follow:

a) Section 5.3.4.1

Process: Development

Outputs: User Interface Specification (contents: screens, forms, reports)

Output Type: Specification

b) Section 5.3.8.2

Process: Development

Outputs: Product Version Generation (contents: how to generate a software version through makefiles, compilation, etc.)

Output Type: Specification

Ps: This guide doesn't mention other ISO standards or item works. The project SPICE (WG10) has a mapping to ISO 12207 in document Part 2: A model for process management, Annex E.

Andre Valle

abvalle@actech.com.br, abvalle@omega.lncc.br

ABNT - Brazilian National Standardisation Body

CANADA

Canadian Comments on PDTR 15271, Guidebook for Life cycle process. (ref. 7/1492)

Canada approves the ballot with the following comments..

Comment set # A

Comments on Proposed Draft Technical Report - Guide for ISO 12207				
Page	Section	Heading	Para	Comments
1	1.2	Audience	1	This statement needs to be clarified and tied in with the Field of application stated in the standard. I suggest: "This guide can be used by software practitioners using the standard. The standard may be used when defining processes to be used in a project, where a contract or informal agreement exists between two Parties. The standard can also be used as a template or model for software process improvement initiatives. The standard describes typical processes and activities which can be tailored to fit the situation."
2	2.1	Engineering Discipline	1	This para refers only to Development and maintenance when it intends to describe the whole of the software lifecycle. I suggest it be reworded: "Software engineering is defined as ... (insert ISO defn of software engineering). The application and practice of software engineering is a relatively young discipline. Only in the last thirty to forty years, have large systems been developed that rely upon software. Often the practice of software engineering has been approached from the perspective of a craftsman, each project providing unique challenges, and dealt with uniquely. This approach frequently results in projects or products which are unreliable, and costly."
2	2.1	Engineering Discipline	2	Replace "software development and maintenance" with software engineering.
2	2.2.1	Modularity	1a	Referenece to "parts" of a process may serve to further muddy definitions . I suggest the wording of the standard be used "activities"
2	2.2.2	Responsibility	2	Suggest change to "...the execution of individual activities or tasks ... Again to maintain consistency.
4	2.3.4	Process Refinement	1	Suggest change to "...An activity within a process"
4	2.4	Processes and Org	1	Perhaps add to the end of the para: "with one of the organizations being identified as the responsible party.
4	2.4	Processes and Org	3	Suggest change to: "...intended not to conflict" , removing be in .
4	2.5	Processes and Projects	1	Suggest change to: "The standard describes the complete array of processes occurring in a large and/or complex project. Yet it is designed to be tailorable for any type, size or complexity of software project .
6	2.11	Compliance	1a	Suggest change to : " ...organisation declares publicly, ..."
7	3	Implementing the standard	1	This needs to be beefed up, to clarify for the reader what they are about to do. Recommend that Sections 3 and

Comments on Proposed Draft Technical Report - Guide for ISO 12207				
Page	Section	Heading	Para	Comments
9	3.2	Tailoring	2	3.1 be revised . This guide is supposed to be expanding upon the standard. This para is adding relatively little value.
10	3.3	Implement'n Strategy		This section seems to be a series of unconnected bits of advice. Recommend rework to make this section stronger
11	4.1	Factors in applying the standard	1	Suggest that the factors to be considered could be grouped as follows: Organizational Issues; Project Risk; and Capability/Maturity of Resources. Looking at applying this standard from a different perspective, a business process reengineering view, one needs to assess: where the organization/project is currently; what constraints are acting upon the organization; what risks must be dealt with; what are the resources available; and what is the desired outcome.
16	5	Application in organizations	All	The introduction to this guide implied that the standard could be used by organizations wishing to improve their software processes. I think this section could emphasize this application more. This standard can be used alone to lead towards process improvement, or in conjunction with assessment methods such as SPICE or SEI's Capability Maturity Model.

Comment #B.1: Suggestions for Revisions to Section 3

3. Implementing the Standard - Overview

The standard may be implemented for a variety of reasons:

- For use in a specific project, to define the software processes, activities and tasks required
- By an organization in an initiative to improve its software processes
- As a component within a larger Systems Life Cycle process.

Whatever the reason for implementation of the standard, the typical implementation process consists of the following activities:

1. Plan Implementation
2. Tailor Standard
3. Conduct Pilot Project(s)
4. Institute
5. Institutionalize

These activities are typical of the process that must be followed when introducing changes into an organization or project.

3.1 Plan Implementation

3.1.1 Identify the reason for implementing the standard and the results or benefits to be achieved by the implementation

3.1.2 Identify roles and responsibilities of the team/organization involved in implementing the standard, assigning a single point of responsibility for each process. In many cases, one individual or organization may be responsible for more than one process, particularly in small projects or organizations.

3.1.3 Identify the resources available for the implementation of the standard, such as time, money, people, and equipment.

3.1.4 Create and document the project plan for implementing the standard.

3.2 Tailor the standard

3.2.1 Identify the existing project or organizational environment. What processes, policies and procedures are already in place? Where are the problem areas? What is the organizational culture (Early adaptors, adverse to change)? What System or Project life cycle will be used. What are the support requirements?

3.2.2 Analyse the project/organizational risks. Is there a high business risk involved? What are the technical risks - is this a safety critical system? ; is new technology being used ?

3.2.3 What level of maturity currently exists in the organization?

3.2.4 Obtain input from affected organizations

3.2.5 Identify the processes to be implemented. Identify the gaps that exist between the existing processes and the standard to be implemented. Prioritize the work effort. Select the processes, activities and tasks to be implemented

3.2.6 Document the tailoring decisions and rationale.

3.3 Conduct Pilot Project

3.3.1 Select the highest priority work , that will result in significant improvements, with a high probability of success, and that can be expected to provide quick, visible results.

3.3.2 Select a team of volunteers to conduct this pilot project, publicize and reward their efforts.

3.3.3 Plan the pilot project

3.3.4 Document detailed process, activities and tasks. Document lessons learned throughout the life of the pilot project. Incorporate the lessons learned into new process.

3.4 Institute the new process

This activity involves the tasks to introduce a new process into the project or organization. Issues such as training, documentation, provision of support tools for this process, and the tracking and oversight of the new process' use and acceptance.

3.5 Institutionalize the process

This activity focuses upon the tasks involved in ensuring that the process is used throughout the project or organization, consistently and automatically.

Comment Set #C:

C.1. ref page 4, section 2.5 Processes and projects, 1st para.

The term 'large' projects as used difficult to deal with as it is undefined. While there are some generally accepted sizing factors (dollars, function points, or LOC) none are used as reference. The same for complexity. As well size and complexity are also relevant to the organisation developing/acquiring the software. What is small or medium to some is large to many others. Is there some way to provide a sample or suggested method of scoping size and/or complexity?

C.2. Page 39 section D.3.2 Tailoring decisions

Although a sample Table 8 maps the standard to some reference which outlines the approach to RAD. Is this a mythical example or is there an actual referent? A reference to a specific document or documents would be helpful if possible?

C.3. Page 44 section D.4.4 Software engineering and quality assessment, subsection a)

The guide refers only to the use of lines of code. This is too restrictive and does not reflect the use of Function Points or other accepted methods of estimating and tracking size, productivity, etc. The guide does not reference the maintenance of documentation which is also a critical component of the software product.

C.4. Ref. Section 2.4 Processes and Organizations - Paragraph 3

Can or should this Guide also offer any suggestions, techniques or guidelines to assist with this resolution?

C.5. Ref. Section 2.10 Software Metrics

Can or should this Guide provide references to other standards which deal with metric; i.e. IEEE-Std-1061.

C.6. Ref. Section 4.1

Factors in applying the standard What about organizational and individual capabilities? For example, levels of familiarization, degree of buy-in and commitment, training and education, experience.

P. Voldner

CZECH REPUBLIC

Czech Republic comments to PDTR 15271 Guide to ISO/IEC 12207 Software life cycle processes

Czech Republic vote is:

Approval of the draft with the following general comment:

On the figure 9 there are described some new activities, which are not contained on ISO/IEC 12207. These activities may be sometimes useful. We hope, that the presentation of such activities on the guide, being on the TR level is possible from the point of view of standardisation rules.

JAPAN

Japan's comments on ISO/IEC N1492 (PDTR 15271 - Guide for ISO/IEC 12207)

Note:

G: General comment

TM: Technical Major comment

Tm: Technical Minor comment

E: Editorial comment

JPN-TM1(2.6 "Interface with systems engineering")

The section (J2.6 Interface with systems engineering (needs to be modified with additional sections like the following.

2.6 Software and systems

2.6.1 Interface with systems engineering

2.6.2 Relation between software and systems

2.6.3 The system based on software

2.6.4 Classification for software and systems with activity level

The contents of current 2.6 will be applied to 2.6.1 as the new contents.

JPN-TM2(new clause 2.6.2 "Relation between software and systems")

The first paragraph of current section 6. and the two diagrams, Figure 3 and 4, needs to be applied to 2.6.2 as the new contents.

JPN-TM3(new clause 2.6.3 "The system based on software")

The content of chapter 2.6.3 is as follows: Although the standard defines the system, it only covers the life cycle process such as development, operation and maintenance of the system focused on the software. Therefore there is no definition for the hardware life cycle process in the standard.

JPN-TM4

(new clause 2.6.4 "Classification for software and systems with activity level")

The contents of chapter 2.6.4 is as follows:

In the list of activities of the development process in the standard, two (kinds of activities will be recognized. One is for system, and the other for software.

The first word of the activity name will tell which an activity is for, system or software. For example, System requirements analysis and System integration are for system, and Software requirements analysis and Software integration are for software. The following figure is a conceptual chart to show all the activities divided in to two groups based on which the activities are for.

(Insert figure 1 here)

As seen in the figure above, the system process begins with "5.3.2 System requirements analysis" and ends up with "5.3.11 System qualification testing." As described in CHAPTER 6 APPLICATION IN A SYSTEM LIFE CYCLE, a system is a combination of hardware, software, and manual operation, the division of a system into these three being performed through "5.3.3 System architectural design", and the software process, which is emerged through the architectural design, will in turn begins with "5.3.4 Software requirement analysis" and ends up with "5.3.9 Software qualification testing". With the completion of software development, hardware and manual operation will be integrated through "5.3.10 Software integration", and then "5.3.11 System qualification testing" will be performed. Based on the processes described above, it is clear that the system process is a superset of software process. In addition, it will be recognised from the diagram above that the activities are well arranged in a V-figure with "5.3.7 Software coding and testing" at its centre. That is, "5.3.2 System requirement analysis" and "5.3.11 System qualification testing", and "5.3.4 Software requirement analysis" and "5.3.9 Software qualification testing" are placed in symmetrical places each other ,respectively. Also recognised in the diagram is a fact that the processes dividing a system into software and integrating

software into system are placed symmetrical positions. That is to say that this standard is organised through the consideration of the V-figure concept.

Figure1 Classification for software and systems with activity level

JPN-TM5(2.11 "Compliance")

We want to change the note in 2.11 "Compliance" as follows to make it clear. NOTE-The standard contains a set of individual requirements i.e. "shall" statements. Use of the standard are not forced to comply with document as a whole i.e. there is no "shall" statement in the standard which says that you have to comply with the whole standard. This would be on agreement between two parties. The parties can agree that the standard as it is will be the basis of an agreement. The parties can also agree to tailor the standard to suit their particular requirements.

JPN-TM6 (new clause 2.12 "The position of this standard, standards in organisation and techniques") It is recommended that a new clause be inserted next to 2.11, and numbered as 2.12. The title of new clause is "The position of this standard, standards in organisation and techniques". Then the contents of the current 2.12. Summary should be covered in 2.13. The contents of the new clause 2.12 is as follows:

This standard is a collection of the highest level activities, such as software acquisition, supply, development, operation, and maintenance. It is usually the case that an organisation has been utilising its own existing standards and specific techniques for live software development. When applying this standard to an organisation, therefore, it is better to make clear the relationship between this standard, the organisation's own standards, and the various techniques that have been employed. Figure 2 shows one of the examples of those relationship. Although the one described in the diagram should not be taken to be the only one which is correct, it is useful when applying this standard to an organisation.

(Insert figure 2 here)

This standard is located at the first level. Standards in organisation are located at the second level. The third level is for detailed development activities, techniques, and tools that are specific to project characteristics. The terms defined and used in the second and the third levels are required to conform this standard. However, since the terms employed in organisational standards have their origins in historical reasons, and have been established in a long history of organisations, it is quite difficult to change those terms in a short period of time. For those cases, it is recommendable to prepare tables that show correspondence between the terms in this standard and the ones in the organisational standard. Preparing two kinds of table will be helpful; one in which the terms in this standard are listed against the ones in the organisational standard, and the other vice versa. When non-traditional development methods, such as RAD, are employed in software development processes, the activities involved look different in part from those in this standard, and this standard might be often considered to be useless for non-traditional methods. Referring to the figure 2, however, it is clear that this standard is defined at the higher level without respect to specific techniques such as RAD. RAD is simply one of the techniques and hence be placed at the lower levels. When employing specific techniques such as RAD, therefore, it is recommended to tailor this standard to the process and the work items of the techniques such as RAD.

Figure 2. The position of this standard, standards in organisation, and techniques

JPN-TM7(4.1.8 "Software types")

The structure from a) to g) in section (J4.1.8 Software types (J needs to be restructured like the following . Because, from b) to d) intend to mean the subdivision of Off-the-shelf, however the current structure is not good enough to understand .

current restructured

- a) 1) New development
- 2) Off-the-shelf
- b) a) Use of off-the-shelf software exactly `` as is ``. This type....
- c) b) Incorporation of off-the-shelf software without modification....
- d) c) Modification of off-the-shelf software.....

-
- e) 3) Software or firmware embedded in or integral to a system
 - f) 4) Software that is stand-alone
 - g) 5) Non-deliverable software

Because of this change, the last sentence of paragraph g), in b) through f), needs to be changed with `` in 2) through 4) ``.

JPN-TM8(4.1.8 "Software types")

We can not understand clearly the following sentence in 4.1.8 c), so please change the sentence to understand easily and clearly..... without modification but where for example application table configuration is required .

JPN-TM9(Annex. B "Documentation requirements)

The title of Annex. B needs to be changed with `` Output and type for each process, because the new title is more suitable for the contents .

JPN-TM10(Annex. C "Life cycle models")

Process mapping table should be added to help readers to understand more easily that SLCP can be mapped into life cycle model. Below is example for Evolutionary model.

Fig. Example of process mapping table of SLCP in Evolutional model.

Marked 'X' means that some activities or tasks will be done with process. It is helpful to make process mapping table like above. User can enlarge process mapping table and can unfold any process to detail activities and tasks.

JPN-TM11(Annex.D.3 "RAD example")

We want to delete Table 8"RAD mapping to the standard" in Annex.D.3. RAD description from RAD- Ref 1 to RAD- Ref 20 in the table are not activities of the standard but contents of DSDM document. Therefore ,it is inadequate to use the table for mapping RAD description to the standard.

JPN-TM12(Annex.D.3 "RAD example")

We want to add adequate section-ref to each work item in Table11"Rapid Application Development example" in the Annex.D.3. (Section-ref shows the number of activity or task of the standard)

MEXICO

From the member body of: **MEXICO**
Guide to ISO/IEC 12207 Software life cycle processes

- We support the draft proposal with the following comments:

Editorial Comment 1:

2.12 Summary (page 6 line 33)

“The standard strongly recommends that its architecture.....”

It was not find that explicit recommendation in the standard, the phrasing should be change to: “it is strongly recommended that standards architecture.....”

Editorial Comment 2:

Life Cycle Function it is not defined in this guide or any of the references. Process and activities are defined but function is not. The relationship between process and function.

Editorial Comment 3 :

2.2.2 Responsibility: page 2, line 30:

Delete “club”, “union” and “society”, or substitute for other examples that would fit better in the context of the users of this guide.

Editorial Comment 4:

3.1 Getting started (page 8, paragraph c):

Section 3.4 of annex B in the standard is relevant for this paragraph and should be referenced. However, this annex B is also referenced in section 3.2 Tailoring on Page 9. In which way section of annex B should be used in each step of this guide ? . It is not clear.

SOUTH AFRICA

Register: Log and Disposition of Review Comment (Guidebook for ISO 12207 CD 2/96)

Submitted by South African Bureau of Standards

Contact: Professor AJ Walker (walker@odie.ee.wits.ac.za)

Comment Record				Comment Status\ (Maj.\ Min.)	Disposition of Comment
Comment Ref.	Comment Detail	Product Revision	Product Section Ref.		
1.	2nd paragraph e) ‘...must be verifiable as a minimum...’ does this mean ‘must at least be verifiable’ OR does this mean ‘must be shown to be a minimum life cycle function’?	CD	p2,2.2.1	Min	Intent must be clarified ‘must at least be verifiable’
2.	They provide for conducting major functions...’	CD	p3,2.3.1	Maj.	Should not ‘primary’ be used in place of ‘major’? Major implies subservient processes - which they are not.
3.	.as far as possible...’	CD	p11,4, Line 7 from top.	Min	Replace ‘much’ with ‘as far as possible’.
4.	last paragraph ‘...significant management oversight...’ ‘oversight’ normally interpreted as neglect?	CD	p13, 4.17	Min.	Replace ‘oversight’ with ‘supervision’
5.	oversight’ see previous comment	CD	p14, 4.1.9	Min.	Replace ‘oversight’ with ‘supervision’
6.	section 6.4.1.1 does not exist	CD	p15, 4.1.10	Min.	Suggest ‘the standard’ be replaced with ‘ISO 12207’
7.	Life cycle models Spiral model needs to be discussed (see p17 of Mock up and Prototype standard section 9 Example A ‘...these cycles can be represented in the form of spirals which advance the project with each successive pass...’	CD	p28 Annex C	Maj.	A reference to, or guidance on the spiral model used in Mockup and Prototype would be beneficial.
8.	b) ‘...are never executed...’	CD	p44, Annex D.4.4	Min.	Noted that ‘d’ is missing in ‘executed’

UNITED KINGDOM

UK comment accompanying a vote of disapproval on PDTR 15271 Guide to ISO/IEC 12207 (Software life cycle processes) 7N1492

Comments formatted for emailing - Identifier is clause/page/line, then review comment.

- 1.1 1/5 It should not be the purpose of a guide to define the necessary concepts, nor to be the glossary for the standard.
- 1.1 1/6 Pity that this guide does not explain the rationale for the requirements, because without such statements of benefit 12207 is difficult to understand.
- 1.2 1/7 It is not clear whether this guide is to read as a novel - from beginning to end; or as a text book - whole section according to need; or as a reference book - look up to topic in index.
- 1.2 1/7 It is not clear how this guide will help the three types of users of 12207 the acquirer & supplier under contract the in-house customer and developer/maintainer under informal agreement the project team for self-assessment and self-improvement
- 1.2 1/7 It is not clear whether 12207 is really applicable to vendors of commercial packages where the final acquirer is represented during development by the vendor's marketing department. This guide should clarify.
- 1.3 1/15 and
- 2.1 2/3 Insert "Computer" before "software", because wide world understanding of "software" is intellectual property on a retaining medium, such as newsprint, video film, computer magnetic media.
- 2.1 2/2 Software crafting does not compare at all with traditional engineering. Define "engineering". My definition is "Application of science to create a cost effective solution". Computer software creation fails on both counts: it has no science, it is seldom seen as cost effective at the outset, though once installed and in use, often demonstrates that work could not be done without it.
- 2.1 2/6 The underlying philosophy of 12207 is not self evidently linked to this collection of "basic concepts". Clarify
- 2.1 2/6 The underlying philosophy of 12207 and "engineering discipline" are not self-evidently linked. Define "engineering discipline" as this guide wishes to explain it for 12207. My definition of "engineering discipline" is "to achieve more with less resources and energy". My interpretation of the underlying philosophy of 12207 is rigid architecture and rigid structuring, where processes are only composed of activities, which are only composed of tasks, and there are not other classes of work; and the deliverables are configurations composed only of components, composed only of units and there are no other classes of product. Such rigidity does not equal rigor, nor is such rigidity expressed in many other branches of engineering.
- 2.2 2/12 The engineering term for clearing a machine off the premises when it has ceased work is "decommissioning". If 12207 is claiming to move into the realm of engineering, shouldn't it use or relate to engineering terminology.
- 2.2 2/12 12207 does not refer to "key" processes. Use terminology of 12207 or relate 12207 terms to the engineering terms in general use. When "key" has been replaced by 12207 term, then also relate "the-replacement-for-key" to the "framework which has clear linkage to system, including...people and business practices".

-
- 2.2 2/13 Are key processes more than the “primary” of 12207? If so, then suggest delete “key” in this paragraph.
- 2.2.1 2/15 Improve sentence by deleting “that is” and: insert “in that” after “modular”
- 2.2.1 2/15 When I started programming I did “modular design”, which was collecting the functions together into manageable lumps such that they could be invoked. This was replaced by structured programming, after Yourdon & Constantine (who borrowed the principles from other branches of engineering), which had the rules of Strongly cohesive: Loosely coupled. I am confused that 12207 claims to be modular by following the rules of “structure”. I agree 12207 is modular, in that it collects functions together so that they can be invoked and does not follow the rules of “structure”. I agree 12207 is modular, in that it collects functions together so that they can be invoked and does not follow the rules of “structure”. Delete a) & b). Retain “modular”
- 2.2.1 2/21 These definitions of process accurately reflect 12207, and show that 12207 is concerned with activity and not with results. Process analysis sprang up to refocus working practices on achieving value for the customer, not just spinning wheels for the amusement of the workers. 12207 fails to use process analysis, so to prevent degradation of the term, use of the word “process” should be minimized in this guide. Use “process” only to refer to 1207 named process which are written with a capital letter.
- 2.2.2 2/29 Rationalize the use of italics. Other use is in 2.8.b) where indicates colloquial speech.
- 2.2.2 2/29-38 Responsibility is irrelevant in process analysis, [except in simplistic blame oriented work partitioning]. A process owner with the customer at both end of the operation (giving input and taking products) is the only one with “responsibility”. 12207 is not organized for nested customer-supplier-customer relationships. 12207 takes the old fashioned approach that the acquirer calls the developer and thus making development the only important part of the software life cycle. This project “life cycle” is usually less than 20% of the product life cycle is not recognized. Clarify in the guide that 20% of the life cycle has over 80% of the tasks.
- 2.2.2 2/36 “different people” implies dedicated QA, CM, documentation personnel working in only one 12207-process. State here that several process may be performed by a single individual.
- 2.2.2 2/37 It is not self-evident how responsibility and tailoring are linked. Clarify.
- 2.2.2 2/38 In what way can personnel not be “legitimately” involved. Is this implying that organizations that do not employ 12207 are doing illegitimate activities?
- 2.3 3/ all too many synonyms (or apparent synonyms): “framework” “architecture” “engineering environment” “discipline” “Key processes” “primary principles” “function” “classes” “Follow ISO Directives and use a single consistent term.
- 2.3.1 3/13 Is this “function” the same as the “function” on page 2 line 18?
- 2.3.2 3/26 Is this “specific purpose” the same as “specialized life cycle function” on page 2 line 19?
- 2.3.3 3/34 In what way are organization and corporate related. Suggest explain in section 2.2
- 2.3.4 4/5 This table does not add to the usability of 12207. Delete. Replace with the process relationship block diagram that is used in the Annexes.
- 2.4 4/15 “Supplier” is generic name for any performer of a process that has any sort of output.
- 2.4 4/16-17 is clarification of 2.2.2 paragraph 2. Delete here and explain in 2.2.2.

-
- 2.4 4/23 “self-imposed tasks” is not self evident. Explain in section with own heading.
- 2.4 4/24 Convert “is intended not to be in conflict with” into the positive phrase “is to be harmonized with”.
- 2.5 4/26 Section is miss-titled. The following text is about tailoring. Change the title to “Tailoring”.
- 2.5 4/30 “Integral” is a contradictory word in British and American English. In British it can mean “complete and standalone”. In American it means “integrated into something which is incomplete without it”. “or integral” does not add clarification to “embedded”. Delete “or integral”.
- 2.5 4/30-31 “natural positional sequence” is contrary to “not...time sequence”. It is not necessary to continue to explain the editing of 12207 and the placement of the text in a 2 dimensional document. Simply clarify that the placement of the processes does not dictate the life cycle sequence and that iterations or omissions are available through tailoring.
- 2.5 4/35 Avoid using “task” as a verb.
- 2.5 4/35 missing Here is the opportunity for saying 12207 has but a single mandatory requirement: it must be tailored.
- 2.5 5/2 “supplier (now as an acquirer)” contracts 2.4 paragraph 1 where organizations derive its [fixed] name from the process it performs. Suggest delete 1st sentence of paragraph 2.4
- 2.6 5/6 “principles of general systems engineering” are not self-evident. Explain and give reference to authoritative source.
- 2.6 5/7 “To a certain extent” is friendly but meaningless. 12207 is designed specifically for software as part of a system of hardware. 12207 can be molded/bent to accommodate application/business software, and with elastic imagination cover commercial packaged software.
- 2.6 5/8 “culled” means slaughtering the oldest and weakest from the herd to keep the numbers down and the breeding stock high. Replace with “extracted”.
- 2.6 5/9 I do not agree that the “system” perspective of the standard is useful when there are no system-level standards available. Useful to whom? Who says there are no system level standards? (Just because SC7/WG7 task force advocated starting a system level project does not mean there are no national standards.)
- 2.6 5/11 I do not agree that the “system” perspective “may be treated as useful guidance” when the software is a system in itself. In what way useful? In what way guidance. This guide must clarify or remove these assertions.
- 2.6 5/12 I do not recognize in what way the system perspective of 12207 “pertains to operation and maintenance”. Clarify or delete.
- 2.7 5/16 “quality management principles” are not clarified by any normative reference. ISO 9004-8 is being drafted to be this title, and 12207 has almost nothing in common with 9004-8. 12207 is all about quality as testing (the “check” of the PDCA cycle): 9004-8 sets out quality as a planning and enabling management activity to enhance productivity to minimize the “checking”.
- 2.7.1 5/16 The title is not a good synopsis of the text. Replace with “PDCA cycle”
- 2.7.1 5/17 “total software life cycle” is not explained earlier. Refer to 2.2, or better move this 1st sentence to 2.2
-

-
- 2.7.1 5/19 PCDA cycle is not in “each process of the life cycle”. It is only in the development activities which is but one of the primary processes. PDCA is not in any of the supporting or organizational processes, despite them all starting with “planning”.
- 2.7.1 5/20 I do not recognize where in 12207 “process and personnel...are assigned...quality related activities”. For me quality is in the “Do” part of PDCA because quality is the satisfaction of the requirements.
- 2.7.1 5/23 I understood the 12207 QA process was compliance of processes with their plans (i.e. 12207 QA overlaps with 12207 auditing).
- 2.7.2 5/24 Compliance of products and services with contractual requirements is not applicable for two-thirds of usage of 12207 (in-house and self-imposition). Does that mean in these 66% QA process should be tailored out in favor of Verification? If so, then state clearly.
- 2.7.2 5/24 Contractual requirements contain the SOR (schedule of requirements, or statement of work), the compliance with which is the province of the Validation process. Does that mean in the instance of commercial acquirer with contract with commercial supplier, then the QA process is tailored out in favor of the Validation process?
- 2.7.2 5/26 Replace “permit” with “conduct”
- 2.7.3 5/29 Resolve whether “organizational” is synonymous with “corporate”. Ref 2.2.2 and 2.3.3
- 2.7.3 5/30 It is a strange view of incorporate bodies that the management of quality is beyond contractual obligations. Incorporated bodies exist to sell products (via formal or informal contracts) to customers to make money for owners (shareholders). Customer will pay and repeat paying only for quality products. This managing quality is the reason for managing the company. Delete “i.e. beyond contractual obligations”.
- 2.8 5/38 Add examples of programming languages with ISO standards. Add 4GL (database oriented, e.g. SQL); client/server (e.g. POSIX); 4.5GL (GUI oriented, e.g. COBRA) Arrange in generational order “assembler, COBOL, ADA, SQL, POSIX, COBRA”. Add reference to their ISO standards number in a new subsection of 1.4 “1.4.2 Information references”
- 2.8 5/39 “These are very much dependent” is chatty but not informative. Give guidance on the dependency criteria, or delete the phrase.
- 2.8 6/2 “what-to-do” and “how-to-do” are a matter of management perspective. The “What-to-do” of the chief executive = make profit, which translates into “How-to-do” of “make products”; “control expenses”; “sell product”; etc. Each of these CEO’s “how” becomes the “what-to-do” of the management level assigned with that level of responsibility. At the technician level the “What-to-do” is embodied in the statement of requirements, and the “how-to-do” becomes the [sub] system architecture which is parceled out to subcontractors to become their SOR (and their “what”). Either fix the management perspective of 12207 and this guide or delete the concept “What” and “How”.
- 2.8 6/7 I cannot detect the difference between a) 2nd bullet and c). Clarify or merge or delete c)
- 2.8 5&6 It is really necessary to sell the usability of 12207? Delete section 2.8
- 2.9 6/11 I do not understand what to expect of a documentation standard, so do not recognize 12207 as “not a documentation standard”. Certainly 12207 does require and name some documents (e.g. specifications, plans). 12207 does imply need to “combine outputs of a similar nature “by having a single process for writing, reviewing and distributing the documentation.

-
- 2.11 6/25 NOTE on compliance is not comprehensible. I do not understand “enforced” in the context of tailoring. I do not understand “complete document”. 12207 did/does contain statement of compliance which 12220-2 had to copy. Was that removed at DIS/IS review? Whoopee.
- 2.12 6/29 I agree 12207 is “merely”, however the guide should not be denigrating its standard.
- 2.12 6/30 I do not wish to have “One key point to remember” I wish to have a guide that helps interpret 12207 into a usable set of requirements.
- 2.12 6/31 It is disputable that 12207 contains a set of “well-defined” building blocks. I contend that in the area outside the Development process, 12207 is neither well-defined nor a set.
- 2.12 6/33 & 34 12207 does not “strongly recommend” in tailoring that “not applicable” is annotated, because even the requirement to have a document in which to annotate the life cycle as related to 12207 is tailorable and can be omitted.
- 3/7/2 I do not understand “type of application scenario”. Define this concept in relationship to section 2.
- 3/7/3 I agree there are “specific issues with the application of the standard”. However the guide is to overcome these and not to emphasize them.
- 3/7/2 to 4 I am confused whether both “this section” and “following section” includes a) to c) Reword introduction to incorporate introductions from sections 4,5,6 explaining what each will clarify, and why there needs to be three separate sections.
- 3.1 7/11 Delete “(corporate)”
- 3.1 7/17 cross reference to section 5 is not satisfactory. Section 5 does not elaborate on “individual’s primary role”.
- 3.1 8/5 Move NOTE to head of list on page 7 rather than afterthought.
- 3.1 8/1 to 4 These are only the 12207 primary processes. Is a “primary role” linked exclusively to the 12207 primary process? If so, give guidance to players in the Supporting and Organizational processes.
- 3.1 8/6 to 20 This list is too many unstructured items to be useful. Separate into “(a) environments to the project” “(b) organizational characteristics” and “(c) project internal characteristics”
- 3.1 8/12 & 13 In the context of 12207, I do not recognize the difference between “life cycle model of project” and “development model”. I would recognize “product cycle” as perhaps started with a project (with its micro-life-cycle) implementing a development model. However in 12207 Development model and “project life cycle model” are synonymous. Clarify 12207 is applicable to product life cycle with project life cycle as a tailored subset.
- 3.1 8/29 Configuration Management, even in 12207, is more than management of change. CM delivers the management of the future structure of the product (the configuration); the state of readiness for use of components and their interdependencies. CM has the potential to deliver requirements traceability into implemented components, which in the software engineering world is claimed by V&V, but omitted by 12207.
- 3.1 8/33 Identify here the parties mentioned as an afterthought in “g”) on following page.
- 3.1 9/4 12207 already implies that V&V and CM is required to be performed by independent agents. Cancel that inference by avoiding examples which reference the support processes as performed by

separate parties. Break the mold and mention as interface agents “designer” or “integrator” or “establishing the project environment (Building Facilities Management company for example)”.

- 3.1 9/7 It has never been clear how 12207 can be tailored for a project and later “improved” for the organization. This guide needs to expand on the mechanisms for feedback and improvement into processes that 12207 rigidly describes as composed of a finite set of activities composed of a defined set of tasks.
- 3.2 9/11 & 12 Duplicates introduction to section 3. Is such reminder necessary? Better to clarify in one place, in introduction to section 3.
- 3.2 8/14 I do not understand how to discover “whichever application sections are relevant”. Explain
- 3.2 10/1 & 2 This “major driver” “according to the contract” is applicable in 33% of usage of the standard. Expand to cater for the other usages, particularly self-imposition by a software vendor of mass market package.
- 3.2 10/4 I am confused whether “entire software system” is software within a “total system” or software as a “system in itself” (ref 2.6)? If the “system in itself”, then suggest that the 12207 activities that follow design are mentioned stopping at the integration into the [hardware] system architecture.
- 3.2 10/5 Refer to the example in Annex D where it is shown that the number of tasks for safety critical is different from those for consumer software. Where such examples do not exist, then add to Annex D.
- 3.3 10/all I do not understand why the Guide to 12207 needs to describe the implementation of 12207 into a non-12207 organization. The Guides to ISO 9001, for example, do not describe the introduction of the quality system, only its design, operation and improvement. Delete 3.3.1 & 3.3.2 Merge 3.3.3 into 3.2
- 3.3.1 10/9 Despite the heading to 3.3, I cannot relate “implementation strategy” to contents of 12207. Perhaps I am misled by being in the internals of the standard having just read about tailoring. Engineering practice (not to say 12207 version of PDCA) suggests that planning for implementation should be the driving force of design and therefore section 3.3 should precede 3.2
- 3.3.1 10/9 & 10 “but maybe less obvious” is patronizing. Replace with separate explanation for each of the 3 usages of 12207, particularly where “in-house” does not establish a project with full responsibility.
- 3.3.1 10/14 Past tense: because the before tasks have ended successfully. Replace “are” with “were”.
- 3.3.1 10/15 12207 can already be accused of adding unnecessary work, so this guide must not infer that all project practitioners always need training. Always plan training, which starts with needs analysis and skill matching, both not mentioned by 12207. It is not self evident that training in 12207 will be necessary or desirable. Training is not equivalent to awareness of progress. Separate this concept to another bullet. It is not clear whether progress relates to introduction of use of 12207 or the progress of the project to deliver IT.
- 3.3.1 10/18 remove parentheses to clarify that “implementation” is the process of project management, not the results of the project (which is what I infer by “implementation).
- 3.3.1 10/7 Is this the “Considerations” for “need to be considered” in section 4?
- 3.3.2 10/31 In which sense is “limit exposure”? Who sees what (as a photographer) or who takes what risk (as a speculator). If the latter, then explain what risks should be managed.

-
- 3.3.3 10/all I cannot relate this subsection to implementation of 12207 into a non-12207 organization. Should this mapping become the predominant life cycle of the organization from which later projects chose a further subset? What happens to 12207 organizations subsequently? This guide infers that tailoring is once off for the organization. Clarify that implementing 12207 is multilayered and repetitive. Implemented at organization (acquirer ad supplier); at process owner (operations or supplier); at technical level (any 12207 named process). For process owner and technical the implementation is repeated for each project or program of projects.
- 3.3.3 10/37 Defining the sequence and prescribing a life cycle are not synonymous. Delete “i.e.” and create two sentences.
- 3.3.3 10/38 If the aspect is “key” then it should be a step in 3.1, not hidden in subsection of implementing the standard into an organization.
- 3.3.3 10/39 I do not recall 12207 requiring the mapping to be documented. [Even if it is required, than requirement is tailorable].
- 3.3.3 10/all Insufficient guidance is given about mapping to life cycle leaving impression that mapping is too difficult to explain and therefore too difficult to achieve. Suggest also make reference to Annex D.
- 4/11 to 15/all It is not clear that all the following activities for “considering” add any value to a software development life cycle. 12207 is already accused o having lots of tasks with no apparent benefit, and this guide is reinforcing that impression.
- 4/11/4 to 6 sentence is too complex with list of “things”. Layout as bullets.
- 4/11/7 12207 should be better than “as much as possible”. State as success oriented.
- 4/11/7 There is no evidence in 12207 of interest in cost reduction.
- 4/11/8 Parties (ref 2.2.2) are the contractual partners. Is that who is meant by “all”. Clarify what tailoring is possible when 12207 usage is self-imposition and in-house.
- 4/11/all I am confused trying to relate this introduction to the flow in section 3.1. At which point (or orthogonal projection) has section 4 started.
- 4.1 11/14 Relisting the long list of 3.1.c) does not help interpretation of 12207. Since the listed items are intended to be the subsection headings of 4.1, then delete list from 3.1.c) and refer there to section 4.1.
- 4.1 11/18 It is not clear why “maintenance” is a “strategy”. Not is it clarified in section 4.1.5
- 4.1.1 11/25 Despite 2.6 and the 2nd sentence of 4.1.1 in parentheses, I cannot relate to the 66% usage of 12207 which is not derived from the “total system [where] the software is integrated back into the system”.
- 4.1.1 11/28 This Guide to 12207 has not so far explained the concept of 12207 itself being used as a “guide”. Explain in new subsection of 1.2 “1.2.2 Audience”.
- 4.1.1 11/all Section 6 is dedicated to the system life cycle. What is the relevance in repeating section 4? This subsection would appear to read with the same guidance by replacing “system” by “customer’s”.
- 4.1.1 11/29 It is not apparent from reading Annex C that it gives the “role of 12207 in these phases”.
- 4.1.1 11/39 Throw illumination on 12207 obscure invocation paths by clarifying that each instance of Maintenance is a micro-Development.

-
- 4.1.2 12/2 I do not understand how the supplier will obtain the policies of the acquirer, and vice versa. It is conceivable that there will be policies of other organizations (Integrators, Documentors, Building Facilities Managers, etc.) with which the project needs to comply?
- 4.1.2 12/5 & 7 To the non-military “Security” and “Privacy” are the synonymous. Distinguish why they should be separate in this guide.
- 4.1.2 12/12 Why is “public safety and privacy” listed here and in the bullets above.
- 4.1.2 12/14 “considered” is used hereafter with no context for the reason or the use to which the results will be put. Should “considerations” be periodic, single instant, or continuous?
- 4/1/2 12/15 Again “safety and security” gives the impression that this guide is only for the military (and thus adversarial 2 party contract). As alternative try “hardware resourcing” or “risk management”.
- 4.1.3 12/19 Yes, 12207 has rigidly defined the hierarchy for configuration items composed of components composed of units, fullstop. However that rigidity is not helpful in real life. In reality Configuration Items are recursive to level necessary for control. Avoid relating CI only to sub-systems, the high level pieces seen by the customer. Enable “CI” to be the subset of deliverable outsourced by the prime “Supplier” to a subcontractor, and “CI” to be those parts passed to further specialized subcontractor.
- 4.1.3 12/19 Why “especially”? What else is of interest to software? Surely “only”.
- 4.1.3 12/20 Which super-process performed the “allocation”. Usually this allocation should be done by the Acquirer before contact with the potential supplier. Thus the whole palaver of separation of hardware and software is irrelevant to the software life cycle standard.
- 4.1.2 12/20 By what criteria will “identification... of critical” be done?
- 4.1.3 12/22 Not coming from the embedded software world I do not view this list as “system-level” characteristics. To me coming from standalone software, these are performance, functionality, portability, reliability, ...and ...requirements collectively called quality characteristics as described in ISO 9126.
- 4.1.3 12/28 By what criteria will “desirable” be recognized? Isn’t it just another performance or portability requirement?
- 4.1.3 12/27 The only system characteristic that I can think of that might affect the software life cycle is participating in concurrent development with the hardware and firmware. Then the boundary would be in constant movement requiring the life cycle to be tailored for prototyping and other exploration of requirements with considerable effort on interface control.
- 4.1.3 12/30 The converse is “if few, then sloppily performed for a subset of the few”. Is that interpretation intended?
- 4.1.4 13/3 The number of CI which map to deliverables specified by the customer may be available, but the total number of configuration items will not be known when 12207 is being tailored by the Acquirer and Supplier during contract negotiations. Avoid using CI to mean only the products of the project. Use “project products”.
- 4.1.4 13/7 to 7 Most of these characteristics cannot be known when the tailoring is being done.
- 4.1.4 13/8 This rigid hierarchy of CI cannot be known before design, which is long after 12207 has been tailored specifically for the project. It is likely that a software project will have “many” [recursive] Cis.

If the software has “few” then a software life cycle is probably unnecessary. Clarify that the complexity of interdependency that causes a simple job to become a project should be the criteria for implementing the basic management of the configuration. Give some value to “many”.

- 4.1.4 13/11 Explain “evaluation-related activities” in context of 12207 which has the evaluation as supporting processes.
- 4.1.4 13/12 Are critical characteristics only at the system level? WG9, working on “integrity levels” agrees that the consequences of software are only seen through the human interface, which could be economic as well as life threatening.
- 4.1.5 13/13 It is not self-evident why maintenance requires a “strategy”.
- 4.1.5 13/16 In commercial software, future change is unknowable.
- 4.1.5 13/18 With today’s mobile staff (even those in the military) it is not possible to know the extent of training in future. Are these unknowables the reason for “strategy”? If so, make explicit and separate the knowable (who will maintain) from the unknowable.
- 4.1.4 13/19 It is interesting to see maintenance considering the maintenance environment, which may be passed from maintainer to maintainer during the 80% of the product life cycle that is maintenance. Make more explicit.
- 4.1.5 13/22 On what criteria/authority is it “advisable to have documentation available electronically”. This Guide should advise about preserving the document control environment so that electronic documents can be read during the extended life cycle because not all word processors that have existed are emulated (by backward compatibility) by modern word processors, and the same will be true in the future. Similarly the media on which electronic documents are stored deteriorates over time, and may have to devices for retrieval in the near future. (Do you know anyone with an inch disk drive?).
- 4.1.6 13/25 It is not necessary to keep repeating the same three types of life cycle. If an example is needed then use some not in Annex C (which will cause them to be demanded for Annex C). Try for example Isolated Package Procurement; Support of Legacy; Replacement of Islands of Information with Integrated Enterprise-wide Packages.
- 4.1.6 13/27 Replace “system” with “software” twice. 4.1.6 13/13 Replace “system” with “software”.
- 4.1.6 13/28 Are life cycle models standardized to sufficient extent that certain processes can be “prescribed”? Identify the authority that has standardized “each life cycle model”?
- 4.1.6 13/32 In the life cycles that feed output of one phase into the following, the relevant documentation is seldom complete before the next phase commences. The sole exception are Package Procurement by open competition directives, where the statement of requirements must be sent to all prospective suppliers simultaneously. Later in this life cycle the phases overlap. Delete “i.e.”.
- 4.1.6 13/32 By what criteria is “relevant documentation” decided?
- 4.1.7 13/35 Even in a process of collecting considerations for no apparent reason, the purpose served by identifying the “number of personnel in each party” is completely mystifying. Delete as irrelevant.
- 4.1.7 13/37 “large” and “significant” needs to be quantified. The management effort varies with perspective: a sup

Additional comments:

- a. Is it intended that the published document will be identified by the number “15721” - if so it bears little relation to the number of the parent standard of which it purports to be guide!
- b. Page 1 Para 1.1 states that “The guide does not provide interpretation for the requirements in the standard”. However, surely that is exactly what a guide should do!
- c. Page 1 Para 1.2. It is not clear from this paragraph whether the guide is written for those involved in a “contractual” situation.
- d. Page 1 Para 1.3 states that one of the prerequisites for using the guide is “to be familiar with the standard”. If this is the case, why repeat the information in the opening paragraphs 2.1 to 2.3.4 - much of which can already be found in the standard!
- e. Page 2 Para 2.1 Suggest that the word “rigour” be replaced by “control”.
- f. Page 2 Para 2.2.1 The word “function” is introduced in this paragraph, without any clear definition of its meaning!
- g. Page 5 Para 2.6 This paragraph describes software as being “a system in itself”. It is difficult to imagine how this could occur!
- h. Page 10 Para 3.3.1 Suggest that the word “mechanical” be replaced by “mechanistic”
- i. Page 10 Para 3.3.1c the use of the term “cutover” is not understood.
- j. Page 11 para 4.1.1 (line 30) This states that “the system life cycle phase would help ascertain whether the project would be under feasibility, prototyping...etc”. This is most confusing - the system life cycle phase is feasibility, or prototyping etc!!
- k. Page 13 Para 4.1.4 It is not understood (when deciding how to apply the standard), how a party could determine the “number of software configuration items”. The selection of software configuration items will be carried out under the supporting process of Configuration Management - and until that process is invoked the details will not be known!
- l. Page 16 Para 5.3 This para refers to “management commitment” and states that “in a two party situation this would be achieved by a contract”. It must be considered somewhat naïve to believe that the placing of a contract will ensure management commitment!!
- m. Page 34 Annex D This states that “the standard may be tailored by adding new activities”. However, the standard states quite clearly that “the tailoring process is by deletion of non applicable processes” and “addition of unique or special processes may be provided in the contract”. The information contained in the standard and the guide appears to be contradictory!
- n. Page 36 Whilst it is appreciated that examples are being highlighted here, we now have “shalls” appearing in a guide!

USA

USNB position on PDTR 15271, Guide to 12207 (8 Feb 1996)

GENERAL Comments:

USA - 1

The USNB thanks JTC1/SC7/WG7 for producing this guide on the use of ISO/IEC 12207, Software life cycle processes.

USA - 2

Terms used in the guidebook that are not defined or that are used differently in the standard than in the guide, i.e., 4.1.8.c "application table configuration" ; "specification" (used in the standard only to refer to the documentation of requirements for a system or software item, but used in the guidebook to refer to other outputs such as "detailed database design," "acquisition documentation" (Table 4, Annex B), and "infrastructure configuration" (Table 6, Annex B)

USA - 3

The guidebook refers to "levels of compliance." The concept of "levels of compliance" is not mentioned in clause 1.4 of ISO/IEC 12207. Corrective Action: Rationale needs to be provided for this interpretation.

TECHNICAL COMMENTS

USA - 4 4.1.8.b

states that off-the-shelf software used "as is" enters the Development process at software qualification testing. This needs clarification. Corrective Action: Explain that off-the-shelf software may enter the Development process of the system under development at any time. Clarify that the off-the-shelf software itself has already been designed and coded, but that testing needed depends upon factors such as criticality of the system, whether the software has a history of reliable use, and where it is used in the overall structure of the system.

USA - 5 6.3

of the guidebook defines a system life cycle process. Corrective Action: If the system life cycle process has been defined at the ISO level, the source of that definition should be cited. If not, the section defining the system life cycle should be deleted from the guide until the definition is agreed upon by the group responsible for the system life cycle standard.

USA - 6

Life cycle model , life cycle, and life cycle phases are used as if they were synonymous in section 4.1.1, 6.3, and Annex C, but refer to different things. The life cycle models described in Annex C appear to be incomplete using the definition of life cycle model in 12207.

Corrective Action: Ensure that life cycle references are used consistently and correctly .

USA - 7

Sections 3 and 4 refer to "implementing the standard", application on projects." However, much of the information relates to considerations for "tailoring" for a project rather than either "implementing "or "applying" the standard on a project.

Corrective Action: Revise to separate those tasks that refer to "tailoring" considerations from the "application" and "implementing "considerations.

EDITORIAL COMMENT

USA 8

Figures and tables are not referenced by name from within the body of the text

Corrective action: Reference table numbers and figures instead of using phrases such as "the following table".