

Information technology — Guide for ISO/IEC 12207 (Software Life Cycle Processes)

ICS 35.080

National foreword

This British Standard reproduces verbatim ISO/IEC TR 15271:1998 and implements it as the UK national standard.

The UK participation in its preparation was entrusted to Technical Committee IST/15, Software engineering, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the BSI Standards Catalogue under the section entitled “International Standards Correspondence Index”, or by using the “Find” facility of the BSI Standards Electronic Catalogue.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, the ISO/IEC TR title page, pages ii to v, a blank page, pages 1 to 49 and a back cover.

The BSI copyright notice displayed in this document indicates when the document was last issued.

This British Standard, having been prepared under the direction of the DISC Board, was published under the authority of the Standards Committee and comes into effect on 15 October 1999

© BSI 10-1999

ISBN 0 580 35111 4

Amendments issued since publication

Amd. No.	Date	Comments

TECHNICAL REPORT

ISO/IEC TR 15271

First edition
1998-12-01

Information technology — Guide for ISO/IEC 12207 (Software Life Cycle Processes)

*Technologies de l'information — Guide pour l'ISO/CEI 12207 (Processus
du cycle de vie du logiciel)*



Contents

1 Scope 1

1.1 Purpose..... 1

1.2 Audience..... 1

1.3 Prerequisites 1

2 References..... 1

3 Notation 2

4 Basic concepts behind ISO/IEC 12207 2

4.1 Engineering discipline..... 2

4.2 Software life cycle architecture 2

4.2.1 Modularity..... 2

4.2.2 Responsibility 3

4.3 The nature of the processes 3

4.3.1 Primary processes..... 3

4.3.2 Supporting processes 4

4.3.3 Organizational processes 4

4.3.4 Process refinement..... 4

4.4 Processes and projects..... 5

4.5 Processes and organizations 5

4.6 Software and systems 6

4.6.1 Interface with systems engineering..... 6

4.6.2 Relation between software and the system 6

4.6.3 Systems based on software 8

4.6.4 Classification of system and software activities 8

4.7 Management and planning	9
4.7.1 Project management plan	9
4.7.2 Subordinate plans	10
4.7.3 Document control	10
4.8 Implementation of quality management principles	10
4.8.1 Integration of quality into the life cycle	10
4.8.2 Quality Assurance process	10
4.8.3 Improvement process	10
4.9 Flexibility and responsiveness to evolving technology	10
4.10 Processes and documentation	11
4.11 Software metrics	11
4.12 Compliance	11
4.13 Summary	11
5 Implementing ISO/IEC 12207	12
5.1 Overview	12
5.2 Plan the implementation	12
5.3 Tailoring ISO/IEC 12207	13
5.3.1 Identify the project environment and characteristics	14
5.3.2 Solicit inputs	15
5.3.3 Select processes, activities and tasks	15
5.3.4 Document the tailoring decisions and rationale	15
5.4 Conduct pilot project(s)	15
5.5 Formalize the approach	16
5.6 Institutionalize the approach	16
6 Application on projects	16
6.1 Factors in applying ISO/IEC 12207	16
6.1.1 System life cycle model	16
6.1.2 Organizational policies and procedures	17
6.1.3 System characteristics	17
6.1.4 Software characteristics	18
6.1.5 Software maintenance strategy	18
6.1.6 Life cycle model of the project	18
6.1.7 Diversity of the parties involved	19

6.1.8 Software types	19
6.1.9 Project size	20
6.1.10 Project criticality	20
6.1.11 Technical risk	20
7 Application in organizations	20
7.1 Considerations and techniques	20
7.2 Application opportunities	20
7.3 Management commitment	21
8 Application using a system life cycle model	21
8.1 System life cycle model	21
8.2 Software life cycle model	22
8.3 Example of ISO/IEC 12207 in a generic system life cycle model	22
8.4 Needs determination activity	23
8.5 Concept exploration and definition activity	23
8.6 Demonstration and validation activity	23
8.7 Engineering/development activity	24
8.8 Production/manufacturing activity	24
8.9 Deployment/sales activity	24
8.10 Operations activity	24
8.11 Maintenance and support activity	24
8.12 Retirement activity	25
8.13 Software life cycle processes in a generic system life cycle model	25
Annexes	
A Quality processes and evaluation requirements	26
B Process output categorization	28
C Life cycle models	32
D Examples of tailoring	39

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The main task of technical committees is to prepare International Standards. In exceptional circumstances a technical committee may propose the publication of a Technical Report of one of the following types:

- type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;
- type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;
- type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard (“state of the art”, for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until data they provide are considered to be no longer valid or useful.

ISO/IEC TR 15271, which is a Technical Report of type 3, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software engineering*.

Information technology — Guide for ISO/IEC 12207 (Software Life Cycle Processes)

1 Scope

1.1 Purpose

The purpose of this Technical Report is to provide guidance on the application of ISO/IEC 12207.

This Technical Report elaborates on factors which should be considered when applying ISO/IEC 12207 and does this in the context of the various ways in which ISO/IEC 12207 can be applied. The guidance is not intended to provide the rationale for the requirements of ISO/IEC 12207.

The three fundamental life cycle models are discussed and examples of tailoring are provided.

1.2 Audience

This Technical Report is written for those who will use or apply ISO/IEC 12207 in contractual situations, on a project irrespective of size or complexity, in an organization as a self-evaluation or for software process improvement initiatives.

This Technical Report discusses how ISO/IEC 12207 may be used in relation to various types of software and indicates which processes may be relevant in each case.

This Technical Report supports ISO/IEC 12207 when it is used as a requirements document and also for use as a template for guidance. (An example of the latter is when ISO/IEC 12207 is self-imposed as part of a process improvement exercise.) The whole Technical Report should be understood but it may be used in relation to particular situations by referring to specific clauses.

1.3 Prerequisites

The prerequisites to using this Technical Report are:

- a) Availability of ISO/IEC 12207;
- b) Familiarity with ISO/IEC 12207;
- c) Familiarity with the relevant organizational policies;
- d) General knowledge of software management, software engineering and software life cycle models.

2 References

This Technical Report makes reference to the following standards:

ISO/IEC 12207:1995, *Information technology — Software life cycle processes*.

ISO/IEC 9126:1991, *Information technology — Software product evaluation — Quality characteristics and guidelines for their use*.

ISO/IEC TR 15504 (all parts), *Information technology — Software process assessment*.

3 Notation

Diagrams depicting the processes and activities of ISO/IEC 12207 follow the style used in ISO/IEC 12207 as shown in Figure 1.

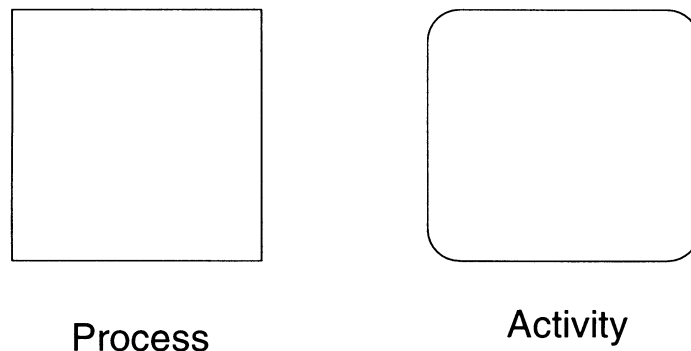


Figure 1 — Drawing notation

4 Basic concepts behind ISO/IEC 12207

4.1 Engineering discipline

The application and practice of software engineering is a relatively young discipline when compared to the traditional branches of engineering. As a result, the control which usually accompanies traditional engineering projects has not always been achieved for software.

The underlying philosophy of ISO/IEC 12207 is that aspects such as software development and maintenance should be conducted in a manner which exhibits engineering discipline. Following this approach allows the establishment of a framework which has clear linkages to the system engineering environment i.e. one which includes software, hardware, people and business practices.

4.2 Software life cycle architecture

ISO/IEC 12207 establishes a top-level architecture of the life cycle of software from conception through retirement. The architecture is constructed with a set of processes and interrelationships among these processes. The processes are based upon two primary principles: modularity and responsibility.

4.2.1 Modularity

The processes in ISO/IEC 12207 are modular, in that they are:

- a) Strongly cohesive. All the parts of a process are strongly related;
- b) Loosely coupled. The number of interfaces among the processes is kept to a minimum.

In principle, each process is dedicated to a unique function within the life cycle and may employ another process for a specialised function. The following presents the rules for identifying, scoping, and structuring the processes:

- a) A process must be modular i.e. one process should perform one and only one function within the life cycle and the interfaces between any two processes should be minimal;
- b) Each process is invoked in the architecture;
- c) If a process A is invoked by a process B and only process B, then A belongs to B;
- d) If a function is invoked by more than one process, then the function becomes a process in itself;
- e) It must be possible to verify any function within the life cycle model;
- f) Each process should have an internal structure defined sufficiently so as to be executable.

4.2.2 Responsibility

In ISO/IEC 12207, the terms “organization” and “party” are nearly synonymous. An organization is a body of persons organized for some specific purpose, and may be as diverse as a corporation, agency, society, union or club. The size of an organization may vary from one person to many persons. When an organization, as a whole or a part, enters into a contract, it is a party. Organizations are separate bodies, but the parties may be from the same organization or from separate organizations.

Each process in ISO/IEC 12207 is considered to be the responsibility of a party. An organization may perform one or more processes. A process may be performed by one organization or more than one organization, with one of the organizations being identified as the responsible party. A party executing a process has the responsibility for that entire process even though the execution of individual tasks may be by different people.

The responsibility feature of the life cycle architecture facilitates tailoring and application of ISO/IEC 12207 on a project, in which many persons may be legitimately involved.

4.3 The nature of the processes

The processes are grouped into three broad classes:

- Primary;
- Supporting;
- Organizational.

4.3.1 Primary processes

The primary processes are:

- Acquisition;
- Supply;
- Development;
- Operation;
- Maintenance.

In practice, the Acquisition process causes the initiation of the software life cycle. The Supply process responds by performing the Development, Operation, and/or Maintenance processes.

4.3.2 Supporting processes

The supporting processes are:

- Documentation;
- Configuration management;
- Quality assurance;
- Verification;
- Validation;
- Joint review;
- Audit;
- Problem resolution.

A supporting process may be employed by another process which thus supports the former with a specific purpose.

4.3.3 Organizational processes

The organizational processes are:

- Management;
- Infrastructure;
- Improvement;
- Training.

An organization may employ these processes organization-wide to establish, implement, and improve a life cycle process.

4.3.4 Process refinement

Each process is further defined in terms of its own constituent activities, each of which is further defined in terms of its constituent tasks. An activity within a process is a set of cohesive tasks. Within ISO/IEC 12207 there are:

Table 1 — Process breakdown

Class	Processes	Activities	Tasks
Primary	5	35	135
Supporting	8	25	70
Organizational	4	14	27
Total	17	74	232

A task is expressed in the form of a requirement, self-declaration, recommendation, or permissible action. For this purpose, ISO/IEC 12207 carefully employs certain auxiliary verbs to differentiate between the forms of tasks:

- “Shall” is used to express a binding provision between two or more parties;
- “Will” to express a declaration of purpose or intent by one party;
- “Should” to express a recommendation among other possibilities;
- “May” to indicate a course of action permissible within the limits of ISO/IEC 12207.

4.4 Processes and projects

ISO/IEC 12207 describes the set of processes used on large and/or complex software projects. However, ISO/IEC 12207 is designed to be tailorable for a software project of any type and of lesser sizes and complexities. It is also designed to be used whether the software is a stand-alone entity, or a part of the total system.

The processes, activities, and tasks in ISO/IEC 12207 are arranged in their most general, natural positional sequence. This positional sequence does not dictate the life cycle model sequence. It is intended that the software project select, order, tailor and iterate the processes, activities, and tasks as applicable or appropriate.

On the same project, ISO/IEC 12207 may be separately applied more than once. For example, in a given software development project, an acquirer may request a supplier to perform software development with the acquirer and the supplier executing one application of ISO/IEC 12207. The supplier may then request its sub-contractor to perform all or part of the software development. The supplier (now in an acquisition mode) and its sub-contractor (in supplier mode) execute a separate application of ISO/IEC 12207. In both situations, it is necessary to tailor ISO/IEC 12207 to reflect the arrangements.

See clause 6 Application on projects for further details.

4.5 Processes and organizations

An organization (or a party) derives its name from the process it is currently performing, for example it is called an acquirer when it performs the Acquisition process.

The processes in ISO/IEC 12207 form a comprehensive set to cater for a wide variety of organizations. An organization, small or large, depending on its business purpose, can select an appropriate subset of the processes (and associated activities and tasks) to fulfil that purpose. ISO/IEC 12207 is intended to be applied internally by an organization or contractually by two or more organizations. In order to facilitate application of ISO/IEC 12207 either internally or contractually, the tasks are expressed in contractual language. When applied internally, the contractual language is interpreted as self-imposed tasks as described in clause 7 Application in organizations.

ISO/IEC 12207 is to be harmonized with an organization's policies and standards that are already in place. It is usually the case that an organisation has been utilising its own existing standards and specific techniques for software development. When applying ISO/IEC 12207 within an organisation, it is therefore important to clarify the relationship between ISO/IEC 12207, the organisation's own standards, and the various techniques that have been employed.

Figure 2 shows one possible example of such relationships which may be useful when applying ISO/IEC 12207 within an organisation. ISO/IEC 12207 is located at the first level, standards in the organisation are located at the second level and the third level is for detailed development activities, techniques, and tools that are specific to a project. The terms defined and used in the second and the third levels are required to conform to ISO/IEC 12207.

Resolution of any conflicts is left to the organization applying ISO/IEC 12207 and may involve developing a mapping and if necessary, filling any gaps.

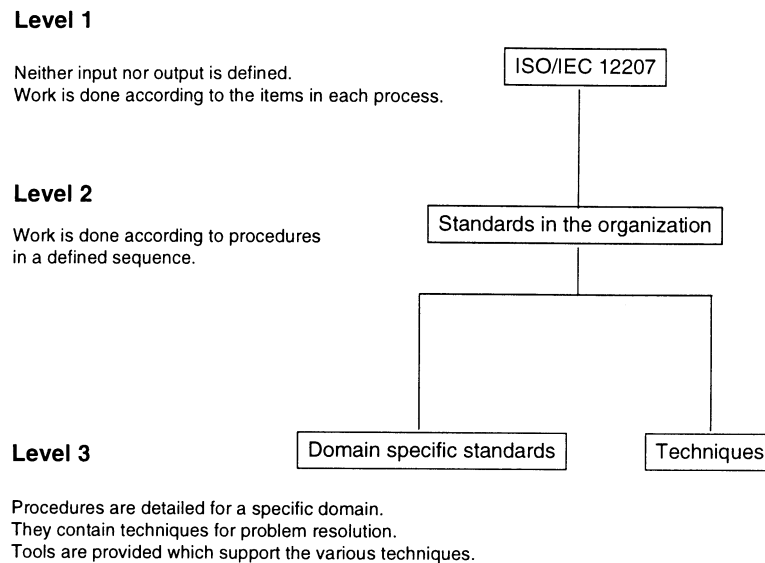


Figure 2 — Relationship with existing documents

4.6 Software and systems

4.6.1 Interface with systems engineering

ISO/IEC 12207 establishes a strong link between the system as a whole and the software. This is possible because ISO/IEC 12207 is based upon general systems engineering.

To a certain extent, ISO/IEC 12207 is designed to function within a systems engineering process. When the software is part of the total system, the software is isolated from the system, produced, and integrated back into the system. This feature of ISO/IEC 12207 is useful when there are no system-level standards available. When the software constitutes the entire scope of interest, the system-level tasks may be treated as useful guidance. In either case, ISO/IEC 12207 provides for the meaningful participation of software engineering in systems engineering.

4.6.2 Relation between software and the system

A system is a specific combination of hardware, computers, software, material, people, and facilities as illustrated in Figure 3. In reality, it is the system that must perform. In the parent system, processes such as business processes exist. Software serves by providing for the execution of certain functions of these processes in computers. The software could be resident in a computer, embedded in a piece of firmware, or integral to a hardware item. In any case, the acquisition, supply, development, operation, or maintenance of the software needs to be in coordination and harmony with those of the parent system.

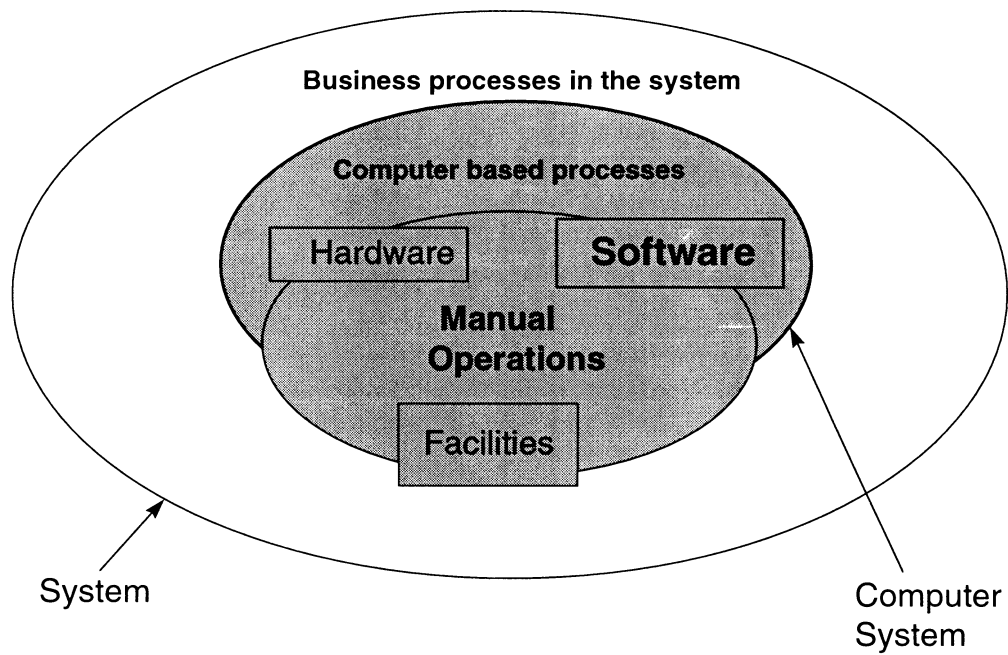


Figure 3 — Software in a system

Within an organization there may be a number of computer systems supporting the business processes as in Figure 4.

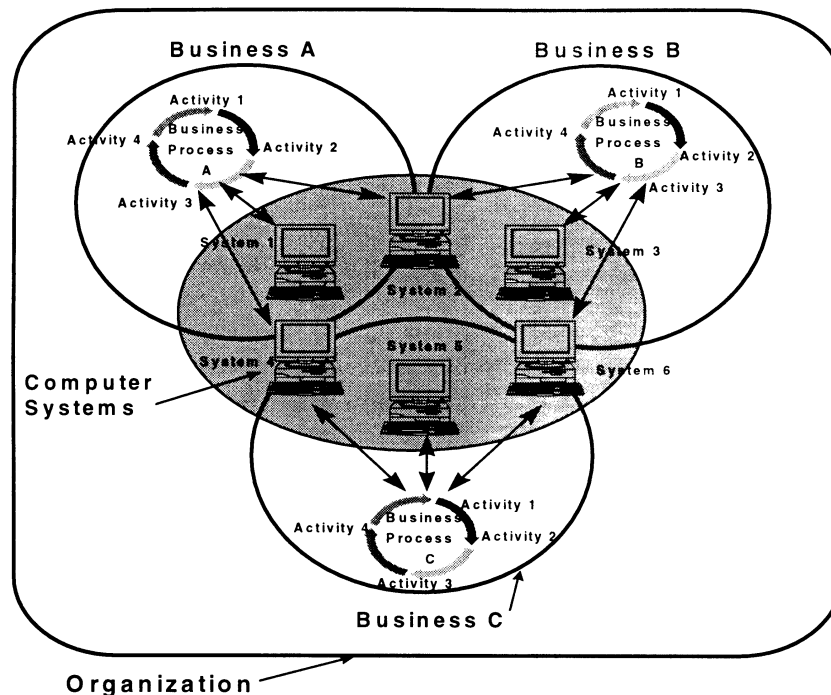


Figure 4 — Computer systems in an organization

4.6.3 Systems based on software

Although ISO/IEC 12207 defines the system, it only covers the life cycle process such as Development, Operation and Maintenance of the system focussed on the software. Therefore, there is no definition for the hardware life cycle process in ISO/IEC 12207.

4.6.4 Classification of system and software activities

Two types of activities are recognized in the Development process in ISO/IEC 12207, i.e. system and software. The scope for these activities is reflected in the name.

Figure 5 shows these activities divided into two groups based on the type and uses a V-presentation to illustrate the symmetry and correlation between the system and software activities.

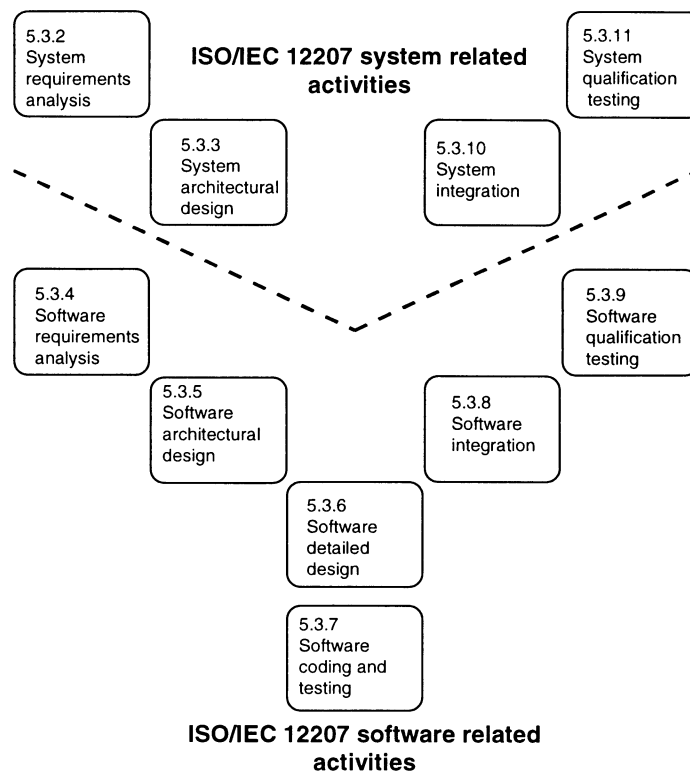


Figure 5 — ISO/IEC 12207 activity classification

As shown in Figure 5, the system activities in the ISO/IEC 12207 Development process begin with 5.3.2 System requirements analysis and terminate with 5.3.11 System qualification testing.

Clause 8 of this Technical Report describes how a system is a combination of hardware, software and manual operations. The division of a system into these elements being performed through the 5.3.3 System architectural design activity of ISO/IEC 12207. The software activities which evolve from this architectural design in turn begin with 5.3.4 Software requirements analysis and terminate with 5.3.9 Software qualification testing.

Once software development has been completed, hardware and manual operations are integrated through the ISO/IEC 12207 subclause 5.3.10 Software integration, and 5.3.11 System qualification testing is then performed. Based on the activities described above, it may be deduced that the system activities are a superset of the software activities.

4.7 Management and planning

For each of the Primary and Supporting processes, management of the process at the project level is done following instantiation of the Management process. It is through this Management process that planning, execution and control of all other planned events is achieved. The items which should be included in a plan are defined in subclause 7.1.2.1 of ISO/IEC 12207, while subclause 7.1.3.2 provides for progress reporting and subclause 7.1.3.3 caters for problem reporting.

4.7.1 Project management plan

In the Supply process, subclause 5.2.4.5 of ISO/IEC 12207 requires that a project management plan is produced and in subclause 5.2.5.1 this plan is executed and controlled. The Supply process in subclause 5.2.5.3 further provides for monitoring and reporting against technical performance, cost and schedules.

4.7.2 Subordinate plans

The items which are listed for consideration in subclause 5.2.4.5 of ISO/IEC 12207 include those relating to processes from the Supporting and Organizational process categories. A number of these processes require that plans be produced e.g. quality assurance, verification and training. Depending on the size and complexity of the project, as well as consideration of the need for subcontracting out some or all of this work, these plans may either be incorporated into the project management plan or developed as separate subordinate documents.

Where subcontractors are used, these should be managed via subclause 5.2.5.4 of ISO/IEC 12207 taking into account the need for increased emphasis on ensuring that appropriate interfaces are established so that the plans can be synchronised.

A summary of the set of subordinate plans may be taken from Table B.2 and Table B.3.

4.7.3 Document control

The requirements for managing documents including plans are described in the Documentation process.

4.8 Implementation of quality management principles

ISO/IEC 12207 implements quality management principles and it does so in three basic ways:

4.8.1 Integration of quality into the life cycle

ISO/IEC 12207 provides the requirements for a comprehensive, integrated set of processes covering the software life cycle. It provides each process with access to a “plan-do-check-act” cycle through the Improvement process. It treats all quality related activities as an integral part of the software life cycle and appropriates those activities to relevant processes of the life cycle. That is, each process and the personnel responsible for performing the process are assigned relevant process-internal quality related activities.

4.8.2 Quality Assurance process

The Quality Assurance process is dedicated to assuring compliance of products and services with their specified requirements and established plans. Persons responsible for this process are provided with the necessary organizational freedom and authority. Organizational freedom means the freedom from those who are directly responsible for producing the product, while authority means the authority to conduct evaluations and initiate corrective actions.

4.8.3 Improvement process

ISO/IEC 12207 contains an Improvement process for further improving quality organization-wide i.e. independently of contractual obligations.

4.9 Flexibility and responsiveness to evolving technology

ISO/IEC 12207 is flexible and responsive to the evolving software engineering discipline. It accomplishes this by providing a top-level, open architecture i.e. ISO/IEC 12207 is:

- a) Useable with any:
 - Life cycle model (e.g. waterfall, incremental or evolutionary);
 - Software engineering method or technique (e.g. object oriented design, structured coding, top-down testing or prototyping);
 - Programming language (e.g. COBOL, Ada or assembly).

These are dependent upon the project and state of the art of technology, and their selection is left to the user of ISO/IEC 12207.