Disclosure to Promote the Right To Information

Whereas the Parliament of India has set out to provide a practical regime of right to information for citizens to secure access to information under the control of public authorities, in order to promote transparency and accountability in the working of every public authority, and whereas the attached publication of the Bureau of Indian Standards is of particular interest to the public, particularly disadvantaged communities and those engaged in the pursuit of education and knowledge, the attached public safety standard is made available to promote the timely dissemination of this information in an accurate manner to the public.

Indian Standard
SOFTWARE DEPENDABILITY THROUGH THE SOFTWARE LIFE-CYCLE PROCESSES — APPLICATION GUIDE

ICS 03.120.01:31.020.30

© BIS 2005
BUREAU OF INDIAN STANDARDS
MANAK BHAVAN, 9 BAHADUR SHAH ZAFAR MARG
NEW DELHI 110002

October 2005

Price Group 11
NATIONAL FOREWORD

This Indian Standard which is identical with IEC 61713 (2000) 'Software dependability through the software life-cycle processes — Application guide' issued by the International Electrotechnical Commission (IEC) was adopted by the Bureau of Indian Standards on the recommendations of the Reliability of Electronic and Electrical Components and Equipments Sectional Committee and approval of the Electronics and Information Technology Division Council.

The text of the IEC Standard has been approved as suitable for publication as an Indian Standard without deviations. Certain conventions are, however, not identical to those used in the Indian Standards. Attention is particularly drawn to the following:

a) Wherever the words 'International Standard' appear referring to this standard, they should be read as 'Indian Standard'.

b) Comma (,) has been used as a decimal marker while in Indian Standards, the current practice is to use a point (.) as the decimal marker.

In this adopted standard, reference appears to certain International Standards for which Indian Standards also exist. The corresponding Indian Standards which are to be substituted in their places are given below along with their degree of equivalence for the editions indicated:

<table>
<thead>
<tr>
<th>International Standard</th>
<th>Corresponding Indian Standard</th>
<th>Degree of Equivalence</th>
</tr>
</thead>
</table>

Only the English language text in the International Standard has been retained while adopting it in this Indian Standard, and as such the page numbers given here are not the same as in IEC Publication.

The Technical Committee responsible for the preparation of this standard has reviewed the provisions of the following International Standards and has decided that they are acceptable for use in conjunction with this standard:

<table>
<thead>
<tr>
<th>International Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEC 61160</td>
<td>Formal design review</td>
</tr>
<tr>
<td>ISO/IEC-12207</td>
<td>Information technology — Software life cycle processes</td>
</tr>
</tbody>
</table>
1 Scope

This International Standard provides guidance on those aspects of software life-cycle activities that have a bearing on the achievement of dependable software. The software life-cycle activities are defined in the context of software life-cycle processes. This guide is intended to be used to support IEC 60300-3-6.

The software life-cycle activities identified can be part of a dependability programme for a system or in relation to a product containing software. The activities identified will help achieve software that is reliable and maintainable and help ensure that appropriate maintenance support will be provided. The activities can be applicable throughout the entire software life cycle or be limited to a subset of the software life-cycle processes depending upon the users of the software. The relationship between the users of the software and the software life-cycle processes is shown in annex B. They are grouped according to the various individual software life-cycle processes that represent the overall software life-cycle as defined in ISO/IEC 12207.

Emphasis is placed on the dependability requirements and activities applicable in the primary software life-cycle processes.

This guide can be used by acquirers, suppliers, developers, operators or maintainers of software. In addition to software and dependability specialists, this guide is intended for use by project managers, quality practitioners and other project participants who develop or use systems or products containing software.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

IEC 60050(191), International Electrotechnical Vocabulary (IEV) – Chapter 191: Dependability and quality of service


IEC 61160, Formal design review

ISO/IEC 12207, Information technology – Software life cycle processes

ISO 8402, Quality management and quality assurance – Vocabulary
3 Definitions

For the purpose of this International Standard, the terms and definitions of IEC 60050(191), ISO/IEC 12207 and ISO 8402 apply together with the following.

3.1 dependability
collective term used to describe the availability performance and its influencing factors: reliability performance, maintainability performance and maintenance support performance. [IEV 191-02-03]
NOTE Software dependability will be described in terms of software reliability, software maintainability and software maintenance support.

3.2 dependability function
term used to describe an individual aspect of the software dependability requirement specification. The dependability function can describe reliability, maintainability or maintenance support-related dependability requirements.

3.3 maintenance release
product release which is carried out for corrective maintenance rather than to provide enhanced functionality.

3.4 software reliability
probability that no fault in any software element of a system will be activated in a given time interval in the operation of the system under given conditions.

3.5 software maintainability
ease with which a detected fault in a software element of a system can be corrected.

3.6 software maintenance support
ability of an organization, under given conditions, to provide upon demand the resources required to maintain a software element of a system, under a given maintenance policy.
NOTE The given conditions are related to the software element itself and to the conditions under which it is used and maintained. [IEV 191-02-08, modified]

To assist understanding of this standard, the following definitions of terms used in ISO/IEC 12207 are repeated here.

3.7 acquirer
organization that acquires or procures a system, software product or software service from a supplier.
NOTE The acquirer could be one of the following: buyer, customer, owner, user, purchaser.

3.8 acquisition
process of obtaining a system, software product or software service.
3.9
agreement
definition of terms and conditions under which a working relationship will be conducted

3.10
audit
conducted by an authorized person for the purpose of providing an independent assessment of software products and processes in order to assess compliance with requirements

3.11
baseline
formally approved version of a configuration item, regardless of media, formally designated and fixed at a specific time during the configuration item's life cycle

3.12
configuration item
entity within a configuration that satisfies an end-use function and that can be uniquely identified at a given reference point

3.13
contract
binding agreement between two parties, especially enforceable by law, or a similar internal agreement wholly within an organization, for the supply of software service or for the supply, development, production, operation, or maintenance of a software product

3.14
developer
organization that performs development activities (including requirements analysis, design, testing through acceptance) during the software life-cycle process

3.15
evaluation
systematic determination of the extent to which an entity meets its specified criteria

3.16
firmware
combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device. The software cannot be readily modified under program control

3.17
life-cycle model
framework containing the processes, activities, and tasks involved in the development, operation, and maintenance of a software product, spanning the life of the system from the definition of its requirements to the termination of its use

3.18
maintainer
organization that performs maintenance activities

3.19
monitoring
examination of the status of the activities of a supplier and of their results by the acquirer or a third party
3.20 non-deliverable item
hardware or software product that is not required to be delivered under the contract but may be employed in the development of a software product

3.21 off-the-shelf product
product that is already developed and available, usable either "as is" or with modification

3.22 operator
organization that operates the system

3.23 process
set of timely ordered and interrelated activities, which transform inputs into outputs
NOTE The term "activities" covers use of resources.

3.24 qualification
process of demonstrating whether an entity is capable of fulfilling specified requirements

3.25 qualification requirement
set of criteria or conditions that have to be met in order to qualify a software product as complying with its specifications and being ready for use in its target environment

3.26 qualification testing
testing, conducted by the developer and witnessed by the acquirer (as appropriate), to demonstrate that a software product meets its specifications and is ready for use in its target environment

3.27 quality assurance
all the planned and systematic activities implemented within the quality system, and demonstrated as needed, to provide adequate confidence that an entity will fulfil requirements for quality
NOTE 1 There are both internal and external purposes for quality assurance:
   a) internal quality assurance: within an organization, quality assurance provides confidence to management;
   b) external quality assurance: in contractual situations, quality assurance provides confidence to the customer or others.
NOTE 2 Some quality control and quality assurance actions are interrelated.
NOTE 3 Unless requirements for quality fully reflect the needs of the user, quality assurance may not provide adequate confidence.

3.28 release
particular version of a configuration item that is made available for a specific purpose (for example, test release)
3.29 request for proposal
tender
document used by the acquirer as the means to announce its intention to potential bidders to acquire a specified system, software product or software service

3.30 retirement
withdrawal of active support by the operation and maintenance organization, partial or total replacement by a new system, or installation of an upgraded system

3.31 security
protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them

3.32 software product
set of computer programs, procedures, and possibly associated documentation and data

3.33 software service
performance of activities, work, or duties connected with a software product, such as its development, maintenance and operation

3.34 software unit
separately compilable piece of code

3.35 statement of work
document used by the acquirer as the means to describe and specify the tasks to be performed under the contract

3.36 supplier
organization that enters into a contract with the acquirer for the supply of a system, software product or software service under the terms of the contract

NOTE 1 The term "supplier" is synonymous with contractor, producer, seller or vendor.
NOTE 2 The acquirer may designate a part of its organization as supplier.

3.37 system
integrated composite that consists of one or more of the processes, hardware, software, facilities and people, that provides a capability to satisfy a stated need or objective

3.38 test coverage
extent to which the test cases test the requirements for the system or software product

3.39 testability
extent to which an objective and feasible test can be designed to determine whether a requirement is met
3.40
user
individual or organization that uses the operational system to perform a specific function
NOTE The user may perform other roles, such as acquirer, developer, or maintainer.

3.41
validation
confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled
NOTE 1 In design and development, validation concerns the process of examining a product to determine conformity with user needs.
NOTE 2 Validation is normally performed on the final product under defined operating conditions. It may be necessary in earlier stages.
NOTE 3 "Validated" is used to designate the corresponding status.
NOTE 4 Multiple validations may be carried out if there are different intended uses.

3.42
verification
confirmation by examination and provision of objective evidence that specified requirements have been fulfilled
NOTE 1 In design and development, verification concerns the process of examining the result of a given activity to determine conformity with the stated requirements for that activity.
NOTE 2 "Verified" is used to designate the corresponding status.

3.43
version
identified instance of an item
NOTE Modification to a version of a software product, resulting in a new version, requires configuration management action.

4 Software life-cycle processes

A software life-cycle process is a set of planned activities or tasks necessary to achieve a stated objective of a software project. There are three groups of processes: primary, supporting and organizational, involving activities relating to a software product from its conception to its retirement.

The five primary processes provide guidance to acquirers, suppliers, developers, operators and maintainers of systems or products containing software.

The eight supporting life-cycle processes support another process and can be used with the five primary processes. The supporting processes are documentation, configuration management, quality assurance, verification, validation, joint review, audit and problem resolution.

The four organizational processes can be used by the major parties or primary processes depending upon the organizational or project needs. The organizational processes are management, infrastructure, improvement and training. Details of the software life-cycle processes and their use are given in ISO/IEC 12207.

The relationships of the software life-cycle processes with the dependability programme elements and tasks described in IEC 60300-2, as they relate to products containing software, is described in IEC 60300-2. Cross-references are made to specific subclauses of IEC 60300-2 in annex A.
5 Dependability activities in the primary software processes

In order to achieve dependable software, it is necessary to identify and implement those activities and tasks that particularly influence dependability. These activities should take account of the intended use, application, operation and environment of the system or product containing software. Consideration should be given, as appropriate, to activities necessary in each of the five primary life-cycle processes; these are described in 5.1 to 5.5.

5.1 Acquisition process

The acquisition process is the term used to describe the activities and tasks of an acquirer involved in acquiring a system or product containing software. From the dependability viewpoint, the software product reliability performance and maintainability performance and the supplier maintenance support performance are the main aspects to be considered. The acquisition process main elements are the specification of dependability requirements, the assessment and selection of a supplier, the preparation of contractual documents, supplier monitoring and acceptance and control. These elements are described in 5.1.1 to 5.1.5.

5.1.1 Specification of dependability requirements

The requirements specification prepared by the acquirer should be expressed in those factors that influence dependability of the software product, i.e. the dependability requirements can be specified in terms of reliability, maintainability and maintenance support requirements. The following requirements should therefore be specified by the acquirer or jointly with the supplier:

a) Software reliability requirements

Requirements for continuous system operation; these should be clearly specified in terms of elapsed time per given time interval, number of operations where no fault occurs before a failure occurs.

The required system operational availability performance i.e. the proportion of operational time when the system is available to perform its functions; this should be expressed in recognizable terms such as mean time to repair (MTTR) and mean time between failures (MTBF); the conditions that constitute failure of the system should also be defined.

Any environmental factors that impose special reliability, requirements for the intended system application.

The recovery conditions of the system, i.e. any time to restore or functional conditions that should be met when restarting a system following a system failure.

Defined fail-safe requirements such as data that should not be lost or safe function states that need to be attained or downgraded operation in the event of a system failure.

Any requirements for third-party evaluation, assessment or certification should be identified. The impact of such requirements on the software design requirements should be considered.

If there is a minimum level of service to be provided and the circumstances (length of time, frequency, etc.) under which reversion to that minimum would be acceptable, these should be identified and specified.

Equal dependability requirements rarely exist for all aspects of a system; acquirers should be encouraged to identify exactly what requirements apply to which parts of a system.
The acquirer should determine what additional costs, resources, etc., will arise either as a direct result of his requirements or as a result of any particular solution to it.

b) Software maintainability requirements

Any environmental factors that impose special maintainability requirements for the intended system application. In addition, the skill level and training level for maintainability should be stated. The maintainability requirements should be expressed in recognizable terms such as mean time to repair (MTTR).

c) Software maintenance support requirements

Any environmental factors that impose special maintenance support requirements for the intended system application. In addition, the skill level and training requirements for maintenance support should be stated. The maintenance support requirements should be expressed in recognizable terms such as mean time to repair (MTTR) or mean logistic delay time (MLDT).

5.1.2 Selection of supplier

The ability of the supplier to develop or supply dependable software and to provide an effective support service during the lifetime of the product is a prime requirement. If the supplier has already acquired or developed software products then the existing products or support services associated with them could be usefully considered when selecting a supplier. If the supplier is being considered as a potential software developer, his development process activities should be assessed according to the guidelines given in 5.3 for developing dependable software products. The selection of supplier should also include the supplier understanding of: (1) technical requirements – software requirements flow-down from system requirement, (2) contractual requirements – agreements, conditions, and terms which affect the software portion of the acquisition. Other supplier selection criteria should also include the past performances on (1) software development, (2) software project management (including software QA, configuration management CM, reliability testing, and maintenance, etc.), (3) software transition to (user) support, (4) software risk assessment and mitigation planning, and (5) software security. When assessing and selecting a supplier, consideration should be given to the following aspects:

a) Existing software product

If data is available, the dependability of products previously provided by the supplier should be assessed if he is a software developer, i.e. the reliability performance, maintainability performance of the products and maintenance support performance of the supplier. The measurement baseline for these performance characteristics should be recognizable availability measures such as MTTR for measuring maintenance support performance or MTBF for measuring reliability performance. If data on the dependability of previously supplied products is not available, the acquirer could consider an indirect method of assessing product dependability by assessing the development processes used by the supplier to produce software products. This approach is discussed in 5.1.2b). The dependability of the software product should be assessed whilst the product is in its defined system environment.

The dependability of the product under consideration should be compared with those of competing products. The ability of the supplier to provide mature software products often has a bearing on this aspect of the product. The comparison should be made using similar measures, if possible considered under similar conditions.
If the supplier is a software developer, then his maintenance support organization structure and product support procedures should be assessed in accordance with the maintenance process dependability guidelines in 5.5.

If the supplier is a software developer, then does he perform any kind of reliability growth testing for the purpose of enhancing product dependability through identification, analysis and correction of faults, and the verification of the effectiveness of the corrective action? Further information on methodologies for reliability growth testing such as the use of reliability growth models is given in 6.8.4 of IEC 60300-3-6. Any reliability growth programme implemented by the supplier should be reviewed to compare current achieved status with projected dependability targets.

b) New software product

If the supplier is a software developer, then the methods he will use to measure dependability as defined in the requirement specification should be assessed for consistency and completeness.

If software is to be developed, the supplier's software development and implementation process should be reviewed in accordance with the development process dependability guidelines (see 5.3). These guidelines advise on the approaches for assessing the developer's methods and procedures for analyzing, specifying, designing, coding, testing and installing dependable software.

c) Software process maturity

If the software developer obtains certain software process maturity ratings, such as the Software Engineering Institute's capability maturity model (CMM), it should be made available for review and taken into consideration as part of the evaluation criteria. This type of model provides some measurements of an individual software developer qualification to perform the software work or to monitor the state of the software process used on an existing software effort.

5.1.3 Preparation of contracts

Evaluation and selection of a supplier (see 5.2) will often involve negotiation on support and supply conditions, it is essential therefore that contractual documents should include comprehensive requirements for dependability, particularly where the contract includes the development of software. With some smaller supplier companies, the contractual documents might not be a formal contract but could be limited to a tender or an order document. In these cases, the term "contract" is extended to cover tender or order documents. The following aspects related to dependability should be specified in the contract after negotiation of mutually acceptable terms and where they can be explicitly defined:

a) the conditions for acceptance, reliability requirements, maintainability requirements and maintenance support requirements;

b) a statement of availability in terms of availability requirements. If the software is to be supplied as part of a system, then the dependability requirements, when operating in the system environment, should be stated. If there are particular dependability standards applicable, these should be specified;

c) an acquirer should state the availability requirements. This will ensure the original availability requirements are stated. If the supplier cannot or will not achieve the stated requirements, then these will be re-negotiated with contract modification or specification revision to reflect the changes;
d) the process for a supplier to make changes to a product needs to be agreed with the acquirer. The process for recording and controlling changes to a software product should be a formal change process such as the software configuration management procedure that will be used and a software change control board (SCCB) to be agreed with the supplier;

e) specified dependability requirements, though rarely absolute, are needed in the contract. For instance, decreased dependability in one area of a total system can cause increased costs in another, for example provision of a hot-standby terminal, more maintenance staff, increased spares holding, etc. There is often a trade-off point where the cost of providing the level of dependability asked for in a system can exceed the cost of compensating for its absence. A contract should allow trade-off to be proposed and costed, and should define the associated constraints (perhaps by specifying a minimum level of service to be provided by the part of the system being procured, or by stating that service personnel cannot be increased/decreased, etc.). Acquirers should be encouraged to specify as explicitly as possible the problem they need to have solved, and any constraints which apply to the solution.

In principle, the acquirer should specify availability but the supplier might negotiate contract terms due to the inability to control reliability, maintainability and or maintenance support. If availability is expressed as a function of reliability, maintainability and maintenance support,

\[
\text{Availability (A)} = f(\text{Reliability (R)}, \text{Maintainability (M)}, \text{Maintenance support (MS)})
\]

For some critical system applications, it is important that availability (A) and maintainability (M) are carefully specified. Availability, for instance, is one of the most critical informations needed for the military. In the medical community, availability of a medical device containing software is often critical to a patient's life;

f) the requirements and conditions for maintenance support: if there are specific maintenance or availability requirements such as time to respond or frequency of maintenance on the system or product containing software, these should be specified. If there are particular regulations or standards these should be specified.

5.1.4 Supplier monitoring

Monitoring of the supplier's review, audit, verification and validation activities (see clause 6) by the acquirer will contribute to the achievement of the dependability of the software being acquired through the verification and validation of the specified dependability functions. The acquirer also should co-operate fully with the supplier by providing any necessary dependability-related information in a timely manner and resolving any pending issues.

5.1.5 Acceptance and completion

The acquirer should ensure that the defined dependability aspects of the requirement specification are included in the preparation and conductance of the acceptance test. The acquirer will accept the deliverable software from the supplier when all the defined dependability acceptance conditions are satisfied.
5.2 Supply process

The supply process covers the activities of the supplier. From the dependability viewpoint, the following additional reliability (R), maintainability (M) and maintenance support (MS) aspects of the supply process activities should be considered. In addition, a reliability growth programme could be required if, for example, this was negotiated during supplier selection (see 5.1.2) or contract negotiation.

5.2.1 Initiation

The supplier should specifically review the dependability requirements of the acquirer and identify the reliability, maintainability and maintenance support requirements and whether these requirements can be met with a new or existing product.

Depending on the results of the review the supplier should make a decision on the level of support required to meet the reliability, maintainability and maintenance support requirements and the resulting costs before deciding whether to bid or accept the contract or propose appropriate alternative solutions.

5.2.2 Preparation of response

With regard to the preparation of a response by the supplier:

a) The supplier's proposal should meet the dependability requirements or standards specified by the acquirer. Where the proposal does not, proposed trade-offs should be identified along with differences between the trade-offs and the initial requirements (in terms of service level, for instance) in order that their value may be judged by the acquirer.

b) If the supplier's proposal requires software development, it should specify if a reliability growth programme or trial is to be implemented by the supplier.

c) The supplier's proposal should include a technical response, which should include a description of the understood reliability, maintainability and maintenance support problems, the proposed solution and whether it fully meets the dependability requirements.

d) The supplier's proposal should include a management response which should include a description of the approach, project milestones and schedule.

e) The supplier's proposal should include a financial response.

f) The supplier's proposal should include a training response which should describe the types of training being offered and their relationship to providing and maintaining trained personnel for the acquisition, supply, development, operation and maintenance processes.

5.2.3 Contract

The supplier should enter into a contract with the acquirer to supply the software product or service. The software product or service dependability requirements should be defined as described in 5.1.3.

5.2.4 Planning

The supplier should develop and document plans based upon the planning requirements. Items and example of plans should be considered under the umbrella of the project management plans. These project management planning items are critical and have an effect on the system reliability, maintainability, and maintenance support.
Planning items to be considered under Project Management Plans are.

a) Project management planning – project organizational structure, responsibility and authority – (project management plan)

b) Engineering environment for development, operation or maintenance – (system engineering management plan-SEMP, software development plan-SDP)

c) Work breakdown structure including software products, services, resources, software size, etc. – (WBS)

d) Management of the quality characteristics of the software product – (software QA plan)

e) Reliability program including reliability growth test, software stress testing, etc – (reliability program plan)

f) Failure mode and criticality analysis, single-point failure and isolation fault – software FMECA

g) Management of the safety, security, and other critical requirements – (software safety plan, computer security plan)

h) Subcontractor management plan

i) Verification and validation approach and agent – (IV&V plan)

j) Risk management, which involves potential technical, cost, and schedule risks – (risk management plan)

k) Training of personnel – (training plan)

5.2.5 Execution and control

The supplier should consider the execution and control of the following dependability activities.

a) The supplier should implement and execute the project management plans referred to in 5.2.4.

b) If the supplier is developing a software product, then the dependability activities defined in 5.3 should be carried out.

c) If the supplier is using software contractors, he will act as an acquirer when acquiring the software from the contractors. It is important therefore that the supplier, with respect to his software contractors, should carry out the dependability activities defined in 5.1.

d) If the supplier is operating the software product or service, then the dependability activities defined in 5.4 should be carried out.

e) If the supplier is maintaining the software product, then the dependability activities defined in 5.5 should be carried out.

f) If the supplier is supporting the software product, then the support services required to maintain the specified availability level, or which are the result of a specified requirement or solution to a requirement, should be identified. The supplier should advise the acquirer what additional costs will arise as a result of such support services being provided.

g) The areas of control should include monitoring and controlling the progress of these dependability requirements, the progress of technical performance, their associate costs, and schedules and reporting of project status. An example will be to track the technical performance measurements (TPM) of reliability and maintainability profiles. This section should also include problem identification relating to the dependability function, recording, analysis and resolution.
5.2.6 Review and evaluation

Review and evaluation of the reliability, maintainability and maintenance support aspects of the software product are important supporting life-cycle process activities. The supporting life-cycle processes involved are the documentation process, configuration management process, quality assurance process, verification process, validation process, joint review process, audit process and problem resolution process. The supplier organization is responsible for ensuring that the processes are in existence and functional. The review and evaluation activities can be internal or external depending upon whether they are in conjunction with the management of the supplier or acquirer respectively. The supporting life-cycle processes do, of course, support all aspects of the acquisition, supply, development, operation and maintenance processes and as a result provide a major indirect contribution to the level of reliability, maintainability and maintenance support achieved. The acquirer should therefore take an interest in all review and evaluation activities as well as those directly associated with reliability, maintainability and maintenance support.

Review and evaluation activities to be considered are the following.

a) The supplier should carry out the review and evaluation activities defined in the project plan and communicate the documented results to the acquirer if it is specified in the contract or is required by the acquirer. This should include both internal and external review and evaluation activities. If it is not specified that the review and evaluation results be communicated to the acquirer, they should always be documented and be available to the acquirer on demand.

b) The review activities should include specific reference to dependability requirements.

c) Satisfaction of dependability requests or specific standards or regulations should be demonstrated to the acquirer. In most cases, for large systems, the supplier will need time to demonstrate that dependability requirements have been met. A reliability growth programme will provide a means of demonstrating or evaluating this over a period of time.

5.2.7 Delivery and completion

The acquirer should ensure that all specified or contracted software-related activities have been completed or have achieved a mutually acceptable state before the software product or service is delivered. This could, for example, involve the completion of a reliability growth programme or the continuation of a programme to an agreed plan.

Software related activities to be considered at completion and delivery of the product by the supplier are the following.

a) The supplier should deliver the software product or service as specified in the contract and which has been the subject of the supply process dependability activities described in 5.2.1 to 5.2.6.

b) The supplier should support the delivered software product according to the dependability activities defined in 5.5.

5.3 Development process

The development process describes the activities of the developer, the organization that defines and develops the software product. If the acquirer requires the supplier to develop a software product, it is important that all the relevant activities of the development process (see ISO/IEC 12207) and the supporting life-cycle processes are carried out. In particular, the review and evaluation activities of the supporting life-cycle processes (see 5.2.6)
should be carried out by the supplier, as these can have a direct influence on the dependability performance of the software product under development. The type of technology and software tools used by the supplier will have an influence on the dependability performance by helping the supplier to achieve a repeatable and controlled development process.

The software design methodology should be a recognized industry standard (for example object-oriented design) which is appropriate to the software product under development. This will enhance the prospect of a well-structured design. The type of technology should be modern and suited to the design methodologies being used. An example of this would be networked workstations that allow efficient and effective group working. The software tools should be to recognized industry standards and supported by a recognized industry supplier of tools. Examples of this would be high-level language compilers compatible with the design methodology or configuration management and database tools that will allow efficient management of software revisions.

The use of modern and sophisticated technologies does carry an element of risk. Therefore, methods for continuously identifying, prioritizing, monitoring, managing and tracking risks is recommended.

The software developer’s activities should include the use of prototyping, modelling and simulation. These practices are sometimes necessary for defining, clarifying user requirements and validating implementation practicality of system and user interface concepts.

Many aspects of the development process can influence the dependability performance of a software product; the following activities should be implemented.

5.3.1 Process implementation

The process implementation activity defines or selects a software life-cycle model appropriate to the scope, magnitude and complexity of the project. The activities and tasks of the development process are mapped onto the selected life-cycle model and appropriate tools, methods, software languages and standards selected by the developer to enable the activities of the development process to be carried out. Selection of the appropriate life-cycle model is therefore important for the most effective implementation of the life-cycle processes and associated dependability activities described in the following clauses.

The following references will assist in the selection of appropriate process implementation activities.

a) Information on mapping of software life-cycle processes onto software life-cycle phases can be found by referring to IEC 60300-3-6.

b) If the developer has a dependability programme (see IEC 60300-2), the development process for software products should be documented and referenced in the dependability programme. Further information can be obtained by referring to IEC 60300-3-6.

5.3.2 System requirement analysis

The objective of the system requirements analysis should be to produce a precise and complete self-consistent specification of functions. The dependability requirements should be specifically included in this analysis.
The following system requirement analysis activities should be considered.

a) The overall dependability requirement of the system to be developed should be analyzed in terms of reliability, maintainability and maintenance support and each dependability function, related in any way to software, identified. The following dependability functions should be identified as appropriate:
   - Functions related to the detection and management of faults in the software product and associated hardware.
   - Functions related to the periodic testing of backup or supporting functions on-line and off-line.
   - Functions that allow the software product to be maintained.
   - Functions related to maintenance support.
   - Functions that specify availability of the software product.
   - Functions that specify the software product testability.
   - Functions related to the preparation of documentation.
   - Functions related to dependability regulations or standards.
   - Functions related to qualification testing.
   - Functions related to the operation of the software product.
   - Functions related to the start-up and shutdown of the software product.
   - Functions related to user support.

b) Each identified dependability function should be specified in precise and consistent terms. The degree of precision with which each function can be specified will depend upon the type of software product and its application.

c) A check should be made that the specification of dependability functions is complete by ensuring that each identified function has an associated specification.

d) Every relevant regulation or dependability standard should be taken into account in the analysis of systems analysis requirements and dependability function specification.

5.3.3 System architectural design

When considering the overall system architecture the dependability performance requirements identified as dependability functions (see 5.3.2) are to be taken into account. Design criteria can have a bearing on the qualitative aspects of availability and maintainability performance but can also be complementary to any quantitative dependability requirements such as whether the product has to be such that no single fault can lead to a critical state of the product. Top-level architectural design of the system therefore should take account of known established or proven system designs which preferably have proven records for operation and maintenance and are appropriate for the identified dependability functions. If available, the results of reliability growth programmes might help identify the suitability of particular system architectures. For existing products, the top-level architectural design should take into account risks such as instability in the design following a vendor update or availability of the product for the projected life cycle of the system.

5.3.4 Software requirements analysis

The objective of the software requirements analysis should be to produce a precise and complete self-consistent specification of functions for each module or item of software. The dependability requirements should be specifically included in this analysis.
The software requirements analysis should include the following.

a) The dependability requirement of the software to be developed should be analyzed in terms of reliability, maintainability and maintenance support and each dependability function identified.

b) Each identified dependability function should be specified in precise and consistent terms. The degree of precision with which each function can be specified will depend upon the type of software product and its application.

c) A check should be made that the specification of dependability functions is complete by ensuring that each identified function has an associated specification.

d) Specifications for engineering design requirements should cover qualification requirements and relevant standards, availability requirements and maintenance requirements.

5.3.5 **Software architectural design**

When considering the software architecture, the dependability performance requirements identified as dependability functions (see 5.3.2) are to be taken into account by considering those aspects of design which impinge upon the defined dependability functions. As for system architectural design, the software architectural design should take account of known or proven designs which are appropriate for the identified dependability functions.

5.3.6 **Software detailed design**

The detailed design of the software should be based on the above described system and software architecture. The design method should facilitate the inclusion of the following features in the software designs:

a) The developer should have well-documented and established software detailed design activities that facilitate the design of software that unambiguously reflects the requirements of the specified dependability functions.

b) The design should facilitate the addressing of such features as modularity, maintainability, testability, verification and validation.

c) The design input and output data at each stage of the software development process should be recorded, analyzed and documented for use in project review and to assist product design improvements.

d) The procedures for design change control and reviews should be specified and implemented (see IEC 61160).

e) The design verification process should be documented and implemented.

f) Established design methodologies and development tools should be used especially those that facilitate the expression of information flow between modules, data structures and dependencies specified in the identified dependability functions.

g) User documentation should be developed and updated as required in conjunction with the software. It should be consistent with the revision level of the software.

5.3.7 **Software coding and testing**

A suitable set of integrated tools such as language compilers, configuration management tools and automated testing tools should be used during the development of the software. Selection of the set of tools should be based on proven products that are well supported by established software manufacturers.
The following coding and testing activities should be considered.

a) The developer should have well-documented and established software coding and testing procedures which specify good programming practice and documentation.

b) The procedure for software product test and validation should be documented and should include a corrective action procedure. If there are specified dependability requirements, tests should be included that relate directly to these requirements.

c) There should be a fully documented, coordinated procedure for reporting software defects and tracking their subsequent correction. It is important for software dependability that this procedure is well established and efficiently implemented in order to ensure that any defects that impact the availability of the required software functionality are corrected quickly and efficiently. The developer, to provide an analysis of fault types, can also use the data collected by the defect reporter/tracker, frequency of occurrence and fault patterns per configuration item or be used to demonstrate reliability growth.

5.3.8 Software integration

Software integration should specifically test for correct implementation of all the identified dependability functions. The test should demonstrate that associated components interact correctly to perform the specified dependability functions. The integration test results should be presented in an auditable form.

The procedure for software integration, system test and installation should be fully documented.

5.3.9 Software qualification testing

If there are specific quantitative or qualitative dependability qualification targets defined in the requirement specification, these should have been included in the specified dependability functions and the associated integration tests. Because of the nature of dependability, it might be mutually agreed between supplier and acquirer that software qualification testing be carried out over a period of time when the system is in use (or in extended trials, whichever is most appropriate to the context). The results of these qualification tests should be audited for compliance with the specified qualification targets.

Aspects of software qualification testing to be considered are the following.

a) The developer should conduct qualification testing in accordance with any specific dependability qualification requirements.

b) The developer should evaluate test coverage and conformance with any dependability requirements.

c) There should be a full programme of technical reviews, internal audits and change control reviews for any specific qualification tests to ensure that the qualification tests relate accurately to the specified dependability qualification requirements and that any proposed changes to the qualification tests do not compromise the accuracy of their relationship with the required dependability qualification targets.
5.3.10 System integration

The system is the term used to describe the final operating combination of the software with the product hardware. System integration will involve the integration of the software with the product hardware components. It is the dependability performance of the integrated system that is tested and checked against overall qualitative and quantitative dependability requirements. The system reliability and maintainability performance should therefore be tested and confirmed to meet the identified dependability functions.

System integration should include the following activities.

a) The procedure for system integration and test should be fully documented.
b) The system integration test results should be subject to a full programme of technical reviews, internal and external audits for compliance with specified system quantitative and qualitative dependability requirements.

5.3.11 System qualification testing

The objective of system qualification testing is to ensure that, after system integration, each system requirement is tested for compliance and that the system is ready for delivery. System qualification testing of dependability performance can be achieved by testing each identified dependability function requirement for compliance. The system dependability function requirements should be evaluated for compliance with respect to reliability and maintainability requirements through the identified dependability functions. Because of the nature of dependability, it might be mutually agreed between supplier and acquirer that system qualification testing be carried out over a period of time when the system is in use (or in extended trials, whichever is most appropriate to the context). Another method for determining whether the overall system reliability complies with requirements is to assess the software reliability of the integrated system by means of a technique which models the development process and/or source code properties. A management framework and data collection procedure is set up and the resulting data is combined with the model to produce an assessment of the overall system reliability. The qualification test results should be audited for compliance with dependability qualification requirements and a qualification report prepared. In some cases, the supplier and customer might reach a mutual agreement to initiate a reliability growth programme if qualification test results indicate that dependability performance improvements are required before the system is compliant with requirements and can be achieved through a cooperative programme of work.

System qualification testing activities should include the following.

a) The developer should conduct overall system qualification testing in accordance with any specific dependability qualification requirements.
b) The developer should evaluate the test coverage and conformance with any overall system dependability requirements.

5.3.12 Software installation

Software installation testing is closely coupled with system qualification testing and is part of the checks carried out to check that the software product is ready for delivery. The conditions under which software is installed and operated will have a bearing on the reliability and maintainability performance achieved. Installation testing should therefore be carried out under the installation conditions specified.
Software installation activities should include the following.

a) The developer should install the software product or system according to the installation documentation and verify that it is installed and operating as required. Because of the nature of dependability, it might be mutually agreed between supplier and acquirer that software installation testing be carried out over a period of time when the system is in use (or in extended trials, whichever is most appropriate to the context).

b) Conformance with any specified installation related dependability requirements should be verified and documented.

5.3.13 Software acceptance support

Software acceptance support is an important developer activity which allows the acquirer to carry out effective joint review and qualification testing of the software product. The commitment to support acceptance testing should be given by the developer especially if prolonged or difficult dependability requirements compliance testing is expected.

The developer should provide appropriate initial and continuing support to the acquirer until the specified dependability requirements have been demonstrated.

5.4 Operation process

The operation process defines the activities of the operator and the organization that operates the system or product in its live environment. This process refers to the system or product and not the software alone because the operation of the software product is an integral part of the system or product. The dependability performance of the software during its operation will depend on the software operation and maintenance procedures implemented within the system environment. The operation and maintenance procedures required to meet the identified dependability functions should be identified and actions taken to check that they are being carried out during the operation of the system. The operation process constituent activities are process implementation, operational testing, system operation and user support. The software operation and maintenance procedures will be considered under these headings.

5.4.1 Process implementation

To implement the operation process the operator should plan and define the tasks and activities he will carry out to implement the constituent activities. When considering dependability performance, the plan should include the following.

a) The operation procedures should be documented and available to the users. The manuals should be kept up to date via an active user document update service.

b) Training for system operation, where necessary, should be provided to the operators.

c) Customers' complaints during system operation and servicing should be retained via a documented reporting procedure and analyzed for prompt corrective action where appropriate.

d) Procedures should be defined which specify any routine actions (for example backing up or initializing data, start-up or shutdown actions) which are necessary in order to meet the identified dependability functions.
e) The scope of the software activities should be specified.

f) Procedures should be defined which specify how the software should be tested in its operational environment and how the results of the testing are to be linked to maintenance actions or to any reliability growth programme that is being implemented.

If a developer is specifying the operational testing, the results should be linked to assessing whether the software product is ready for operational use.

5.4.2 Operational testing

Operational testing is carried out to check whether the system or software product has met the specified criteria for releasing it for operational use. When considering operational dependability, the required reliability and maintainability performance of the system or software product is considered. The reliability and maintainability requirements have been specified in terms of dependability functions (see 5.3.2). Operational testing should be carried out to check whether the specified system dependability function requirements have been met.

Operational testing activities should include the following.

a) Assessment of the system dependability will require the establishment of a framework for collection of data, selection of the appropriate software reliability assessment technique and comparing the assessed dependability with the specified system dependability function requirements. If there are requirements for conformance with specific dependability standards or regulations, these should be covered by the system dependability function requirements.

NOTE Details on selection of the appropriate data collection method and reliability assessment technique, for example, selection of an appropriate product or reliability model to process the collected data, are given in BS 5760: Part 8.

b) Depending upon the system operational environment and usage it might be necessary to collect data over an extended period of time before system software reliability can be assessed. Supplier and acquirer should mutually agree the required period of data collection. Where it is not possible to specify and test the system against quantitative dependability requirements, the acquirer and supplier might mutually agree to the delivery of the software product following initial qualitative reliability assessment subject to the longer term implementation of a data collection and reliability growth programme to achieve dependability goals.

c) The dependability performance of the system after delivery to the user can also be affected by errors or omissions in the operational procedures used by the operator. Records should be kept of any changes in operational procedure so that any related changes in system reliability can be identified and operational procedures improved.

5.4.3 System operation

The system operation activity is defined as the operation of the system in its intended environment according to the user documentation. Consideration should therefore be given to checking for compliance with those dependability function requirements which define the environment that the system is to operate in and also that the user documentation accurately specifies how the system should be operated.
Recommended system operational activity checks are the following.

a) The check for compliance with those dependability functions that specify the type and frequency of the maintenance support activities should therefore be carried out. Typical maintenance support activities, which can have an important impact on dependability or availability, are periodic backup of system data, systematic update of the software to the latest revision levels or regular maintenance of the product hardware. The objective of these checks is to ensure that all defined maintenance functions are carried out in such a way that the software is operated with the latest error-corrected software revisions on the well maintained hardware with regular system backups so that the system can be restored with minimum disruption in the event of a system failure. Some maintenance activities such as backups should be carried out when there is least risk to the system operation, for example when the system is off-line or when there is minimum operator or system activity.

b) The user documentation should be checked for accuracy, completeness and ease of use by the operator, as incorrect, incomplete or difficult to use documentation can lead to operator error or system operational failure. If possible an audit of the identified operator functions against each section of the documentation should be conducted by the supplier and any errors and omissions noted. The results with a report of any corrective actions should be reported to the acquirer.

c) It might not be possible to identify errors or omissions in the user documentation prior to delivery of the system to the user. Records should therefore be kept of any changes in operational procedure after delivery, so that any related changes in system reliability can be identified and operational procedures improved.

5.4.4 User support

The user support activity is defined as those supplier tasks which provide assistance and consultation to the user and the feedback of user requests or problem reports to the maintenance process (see 5.5). The user support activity can have a significant effect on the availability and dependability performance of the software product by, for example, providing rapid and expert advice if the user requests support or providing an efficient process for feedback of user problem reports to the software development and update processes. The user support activity should include the following tasks.

a) The supplier should provide a well-organized and expert support service that is capable of responding quickly and efficiently to requests for support from a user. The support service should be able to respond over the phone to calls for support or provide timely on-site assistance if this service has been negotiated between user and supplier.

b) The supplier should be prepared to provide software support services outside normal working hours if there is a specific requirement for this or if analysis of the dependability functions results in a mutual agreement to provide this.

c) The supplier should have a well-specified and proven procedure for receiving problem reports or enhancement requests from the user, generating solutions and implementing the resulting software update on the user’s system.
d) Where possible, the supplier should provide facilities to the user that will enable him to improve dependability performance by accessing product functional information more efficiently for himself. User access via Internet or Fax back to a knowledge database that is maintained by the supplier is an example of this.

e) The supplier should record all support incidents for analysis and feedback to the development and maintenance processes. Where there is a reliability growth programme, the results of support incident analysis should be included in that programme.

f) The supplier should provide a full programme of user training on system operation either according to specific request from the user or according to the supplier's assessment of the requirements. The supplier should provide further follow-up user training on site if the identified dependability functions and performance requirements involve critical operator actions.

5.5 Maintenance process

The maintenance process defines the activities and tasks of the software maintainer. The influencing factors of dependability have been defined as reliability, maintainability and maintenance support, and, hence, correct implementation of the maintenance process activities and tasks have a critical influence on the achievement of software dependability. The process activities are process implementation, problem and modification analysis, modification implementation, maintenance review/acceptance, migration and software retirement.

Each of these activities will be considered from a dependability point of view in the following subclauses, and emphasis placed on the importance of implementing what are likely to be costly activities and checking for compliance with all identified maintenance requirements. In view of their strong influence on software dependability, it is important that the costly nature of the maintenance process activities does not reduce the likelihood of them being carried out.

5.5.1 Process implementation

The process implementation activity consists of the tasks that enable the maintainer to develop, document and execute procedures for carrying out the activities of the maintenance process described in 5.5.2 to 5.5.6. The supplier should ensure that there is a set of documented procedures in place for receiving, recording and tracking problem reports and modification requests from users and providing feedback to users. The supplier should ensure that the documented procedures are being implemented by both user and supplier as there is a direct connection between system availability and dependability and the efficient reporting and correction of software problems. When considering the dependability aspect of this activity, dependability functions covering these requirements should have been identified and checked for compliance as part of the operational testing (see 5.4.2). Included in the process implementation tasks is training of the system maintainers, where necessary, to ensure that they are able to implement the documented procedures competently.
5.5.2 Problem and modification analysis

The problem and modification analysis activity consists of the analysis of the problem report or modification request for its impact on the system in relation to its size, its cost, the time for modification and the effect on performance. When considering software dependability, the impact of the problem report or modification request on dependability performance should be considered by viewing the impact on each of the identified dependability functions. The supplier should check for compliance with the original dependability function requirements. If there is a conflict between the results of the analysis and achieving compliance with dependability function requirements, there should be agreement and approval before proceeding with the implementation of a modification.

The maintainer should document each problem report or modification request, the results of the analysis and the implementation options developed from the analysis results. If the developer has a reliability growth programme, the results of the analysis should be included in that programme.

5.5.3 Modification implementation

The modification implementation activity consists of an analysis to determine which documentation and associated software items need to be modified, followed by implementation of the identified modifications, test and evaluation of the modified software and documentation items. The development process and all its associated activities (see 5.3) should be used to produce the modified software items. The software dependability considerations are similar to those described for the development process (see 5.3) in that the dependability functions associated with the software items to be modified should be considered in a similar manner to that described for each activity of the development process.

The objective of the review should be to determine that the compliance with the original and any revised dependability function requirements has been achieved. The developer should therefore:

a) determine whether the requested enhancement or modification requires an associated modification to the dependability function specifications;
b) review the system and software design from the point of view of any modifications made to the dependability function specifications;
c) review the dependability requirement specifications in the light of any requested enhancements or modifications;
d) implement and test any software coding changes using the developer's established procedures and tools (see 5.3.7);
e) carry out software integration, qualification and installation testing to check that any modified dependability functions or qualification targets are correctly implemented and that any unmodified dependability functions have not been affected.

5.5.4 Maintenance review/acceptance

The maintenance review/acceptance activity is carried out by the maintainer to determine the integrity of the modified system and to obtain approval for completion of the modification. When considering software dependability, the assessment of system integrity will be in terms of compliance with both modified and unmodified dependability function requirements during system integration, qualification and installation testing. A review of the integration, qualification and installation test results for dependability function compliance should
therefore be carried out following completion of the modification implementation activity tasks (see 5.5.3). If an acceptable level of compliance is achieved, approval certifying that the modification has been completed according to contract specification can be given to the maintainer.

5.5.5 Migration

The migration activity defines the tasks that should be carried out if a system or software product (including data) is to be migrated from a previous to a new operational environment.

When considering software dependability the maintainer should consider the implications of the migration activity tasks on the dependability functions.

Examples of the migration task activities that should be considered are the following.

a) If any software product or data is produced or modified during a system or software product migration, its production and the consideration of the dependability functions should be as described for the activities and tasks defined for problem and modification analysis (5.5.2), modification implementation (5.5.3) and maintenance review/acceptance (5.5.4). The overall objective should be to check the integration, qualification and installation test results for dependability function compliance.

b) A migration plan should be developed, documented and executed. When carrying out a migration requirements analysis, the dependability function requirements should be included in this analysis and dependability function requirements specifications produced. When planning the migration execution, this should be done in cooperation with the user if there are specific system availability requirements to be met during the migration.

c) It is important that the user is given full information on when and why the migration is taking place and the level of support for the old environment after the migration to the new environment. The maintainer should ensure that the user is fully aware of any change in the level of support for the old environment so that the user can assess the potential implications on system availability if he does not migrate to the new environment and there is a change in the maintenance support function offered by the maintainer.

5.5.6 Software retirement

The software retirement activity defines the tasks that should be carried out if a system or software product is to be removed from active support by the operation and maintenance organizations at the request of the software product owner. A major contributing factor of dependability is maintenance support. Hence, the operation and maintenance organizations should consider the implications of the removal of active support on the software dependability function requirements.

A retirement plan for the removal of active support by the operation and maintenance organization should be developed and documented. The timescale for the removal of active support should be mutually agreed between user and maintainer such that the specified maintenance support functions are retained until the software product is retired or the user has migrated to a replacement software product, whichever is appropriate. The user should include archiving of the retiring software product, documentation and data in the retirement plan.
6 Dependability activities in the supporting software processes

The supporting life-cycle processes comprise eight processes. A supporting process supports another process as an integral part with a distinct purpose and contributes to the success and quality of the software project. A supporting process is employed and executed as needed, by another process. The supporting processes are discussed collectively because the main emphasis of this standard is the achievement of dependability through the primary life-cycle processes. The supporting processes contribute to dependability by contributing to the efficiency of the primary processes through their supporting functions. The supporting software life-cycle processes include documentation, configuration management, quality assurance, verification, validation, joint review, audit, and problem resolution. Each process serves a distinct purpose and contributes to the success and quality of the project, including its dependability. To help achieve dependable software, the following supporting process activities should be implemented.

a) The information produced by the life-cycle processes should be documented and maintained to the standards defined in the documentation process. Properly planned, designed, developed, produced, distributed and maintained documents, which will include dependability function specifications, are essential for efficient implementation of the primary processes.

b) Configuration management should be implemented for control of baseline software configuration for each product release. This is a particularly important process for maintaining control of modifications and releases of software and so will contribute to the maintainability and maintenance support aspects of dependability.

c) The joint review, audit and problem resolution processes activities should be consistent with the quality assurance process activities of the software project and form an integral part of all the software life-cycle processes. These supporting processes will contribute to dependability through the specified review and auditing of the dependability function activities.

d) The verification process determines whether the software has been produced according to the requirements and conditions defined in the software life-cycle processes and the validation process determines whether the software has been made according to the specified requirements and should be carried out as the software progresses through its life cycle via the primary and supporting processes. A final verification and validation should form part of the product acceptance plan. These processes contribute to dependability through the verification and validation of dependability function specifications defined in the primary processes.

7 Dependability activities in the organizational software life-cycle processes

Organizational software life-cycle processes are used by an organization to establish and implement an underlying structure made up of associated life-cycle processes and personnel. The organizational processes are discussed collectively because the main emphasis of this standard is the achievement of dependability through the primary life-cycle processes. The organizational processes contribute to dependability by establishing the necessary organization for the efficient implementation of the primary processes and their supporting processes.
The organizational life-cycle processes include management, infrastructure, improvement and training processes. To help achieve dependable software, the following organizational process activities should be implemented.

a) Management process

The management process contains the activities and tasks used to carry out product management, project management and task management of the primary and supporting processes. The management process activities cover the initiation, scope definition, planning, execution, control, review and closure of the project or process. The management process contributes to software dependability by establishing the organization and implementation of the primary process activities that specify, implement, review, evaluate and support the identified dependability functions.

b) Infrastructure process

The infrastructure process establishes and maintains the infrastructure needed for any of the other processes. The infrastructure includes hardware, software, tools, techniques, and standards for development, operation or maintenance of the software. The infrastructure process contributes to software dependability by establishing and maintaining the software tools and techniques that enable the software developer or supplier to achieve repeatable and controlled development and maintenance processes.

c) Improvement process

The improvement process is a process for establishing, assessing, measuring, controlling and improving a software life-cycle process. The process activities establish a suite of organizational processes required to implement the primary and supporting processes, set up an assessment mechanism to ensure their continuing effectiveness and implement improvements considered necessary as a result of the assessment. The improvement process can contribute to software dependability through feedback of the assessment results to a reliability growth programme. On a more general level, improvements in the effectiveness of the development and maintenance processes will indirectly contribute to a more effective specification, implementation, review, evaluation and support of the identified dependability functions. The improvement process or process qualification can be used to assess the effectiveness of processes throughout the life cycle.

d) Training process

The training process provides and maintains trained personnel for the acquisition, supply, development, operation and maintenance processes. Effective and efficient implementation of the above-mentioned process activities and tasks is dependent upon knowledgeable and skilled personnel and training material being available throughout the software product life cycle and for all of the life-cycle processes. The acquirer should therefore confirm that the supplier of the software has planned and implemented a training programme based on a review of project requirements of resources, personnel skills, types and levels of training, training documentation and implementation schedules. The training plan should therefore cover all the primary life-cycle process activities and tasks that contribute to the supply, operation and maintenance of the software product, and the supply of trained personnel for the identified process activities should be coordinated with the project implementation in such a way that they are available in time for each of the planned life-cycle activities. The training plan should be reviewed for compliance with these requirements.
The reliability and maintainability of the software will be enhanced if the design, coding and testing activities of the development processes are efficiently implemented by properly trained personnel. Efficient implementation by trained personnel of those activities and tasks described in 5.3, which are connected with the implementation of the design, code and tests of dependability functions will be an important contribution to achieving dependable software.

Maintenance support will also be enhanced if the developer and supplier personnel have been trained in the necessary skills to implement the activities and tasks of the maintenance process. An efficiently run maintenance support activity will enhance the availability and dependability of the software product.

Efficient implementation of the operation process activities by trained personnel will help improve dependability performance by helping to minimize operational errors.
# Annex A

(informative)

**Association of software life-cycle processes with dependability programme elements and tasks**

<table>
<thead>
<tr>
<th>Software life-cycle processes</th>
<th>Dependability programme elements and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary life-cycle processes (ISO/IEC 12207)</strong></td>
<td><strong>(IEC 60300-2)</strong></td>
</tr>
<tr>
<td>5.1 Acquisition process</td>
<td>6.2 Contract review and liaison</td>
</tr>
<tr>
<td>5.2 Supply process</td>
<td>6.5 Externally provided products</td>
</tr>
<tr>
<td>5.3 Development process</td>
<td>6.2 Contract review and liaison</td>
</tr>
<tr>
<td>5.4 Operation process</td>
<td>6.3 Dependability requirements</td>
</tr>
<tr>
<td>5.5 Maintenance process</td>
<td>6.4 Engineering</td>
</tr>
<tr>
<td></td>
<td>6.6 Analysis, prediction and design review</td>
</tr>
<tr>
<td></td>
<td>6.7 Verification, validation and test</td>
</tr>
<tr>
<td></td>
<td>6.8 Life-cycle cost programme</td>
</tr>
<tr>
<td></td>
<td>6.9 Operation and maintenance support planning</td>
</tr>
<tr>
<td></td>
<td>6.10 Improvements and modifications</td>
</tr>
<tr>
<td></td>
<td>6.11 Experience feedback</td>
</tr>
</tbody>
</table>

| Supporting life-cycle processes | | |
|---------------------------------|---------------------------------|
| 6.1 Documentation process | 6.11 Experience feedback |
| 6.2 Configuration management process | Dependability records 1) |
| 6.3 Quality assurance process | 6.1.4 Configuration management |
| 6.4 Verification process | Quality systems 2) |
| 6.5 Validation process | 6.7 Verification, validation and test |
| 6.6 Joint review process | 6.7 Verification, validation and test |
| 6.7 Audit process | 6.6.8 Formal design review |
| 6.8 Problem resolution process | Quality systems 2) |
| | Management review 2) |
| | Dependability programme review 2) |

| Organizational life-cycle processes | | |
|------------------------------------|---------------------------------|
| 7.1 Management process | Dependability policy 2) |
| 7.2 Infrastructure process | Dependability management implement 1) |
| 7.3 Improvement process | Organizational 2) |
| 7.4 Training process | Quality systems 2) |
| | Dependability management implement 1) |

---

1) Project generic elements according to IEC 60300-1/ISO 9000-4.  
2) Dependability management elements according to IEC 60300-1/ISO 9000-4.
## Annex B
(informative)

### Interaction of users with primary software life-cycle processes

<table>
<thead>
<tr>
<th></th>
<th>Acquirer</th>
<th>Supplier</th>
<th>Developer</th>
<th>Operator</th>
<th>Maintainer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Acquisition process</strong></td>
<td>Performs acquisition activities</td>
<td>Assesses statement of work</td>
<td>Negotiates and signs contract</td>
<td>Ensures requirement specification meets operator needs</td>
<td></td>
</tr>
<tr>
<td><strong>Supply process</strong></td>
<td>Signs contract with supplier</td>
<td>Performs supply activities</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monitors supplier's activities</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Development process</strong></td>
<td>Witnesses qualification testing</td>
<td>Reviews execution of development procedures</td>
<td>Performs development activities</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Operation process</strong></td>
<td>Reviews performance of operational system</td>
<td>Defines operating procedures</td>
<td></td>
<td>Performs operation activities</td>
<td></td>
</tr>
<tr>
<td><strong>Maintenance process</strong></td>
<td>Provides maintenance support for acquirer</td>
<td>Makes software maintainable (e.g. good code structure and documentation)</td>
<td>Performs version control and updates training for operators</td>
<td>Performs maintenance activities</td>
<td></td>
</tr>
</tbody>
</table>
Bibliography


BS 5760: Part 8:1998, Guide to the assessment of reliability of systems containing software


ISO 9000-3:1997, Quality management and quality assurance standards – Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply installation and maintenance of computer software

Bureau of Indian Standards

BIS is a statutory institution established under the Bureau of Indian Standards Act, 1986 to promote harmonious development of the activities of standardization, marking and quality certification of goods and attending to connected matters in the country.

Copyright

BIS has the copyright of all its publications. No part of these publications may be reproduced in any form without the prior permission in writing of BIS. This does not preclude the free use, in the course of implementing the standard, of necessary details, such as symbols and sizes, type or grade designations. Enquiries relating to copyright be addressed to the Director (Publications), BIS.

Review of Indian Standards

Amendments are issued to standards as the need arises on the basis of comments. Standards are also reviewed periodically; a standard along with amendments is reaffirmed when such review indicates that no changes are needed; if the review indicates that changes are needed, it is taken up for revision. Users of Indian Standards should ascertain that they are in possession of the latest amendments or edition by referring to the latest issue of 'BIS Catalogue' and 'Standards : Monthly Additions'.

This Indian Standard has been developed from Doc : No. LTD 03 (2028).

Amendments Issued Since Publication

<table>
<thead>
<tr>
<th>Amend No.</th>
<th>Date of Issue</th>
<th>Text Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BUREAU OF INDIAN STANDARDS

Headquarters:

Manak Bhavan, 9 Bahadur Shah Zafar Marg, New Delhi 110 002
Telephone : 2323 0131, 2323 3375, 2323 9402 Website : www.bis.org.in

Regional Offices :

Central : Manak Bhavan, 9 Bahadur Shah Zafar Marg
NEW DELHI 110 002
Telephone : 2323 7617, 2323 3841

Eastern : 1/14 C. I. T. Scheme VII M, V. I. P. Road, Kankurgachi
KOLKATA 700 054
Telephone : 2337 8499, 2337 8561, 2337 8626, 2337 9120

Northern : SCO 335-336, Sector 34-A, CHANDIGARH 160 022
Telephone : 260 3843, 260 9285

Southern : C. I. T. Campus, IV Cross Road, CHENNAI 600 113
Telephone : 2254 1216, 2254 1442, 2254 2519, 2254 2315

Western : Manakalaya, E9 MIDC, Marol, Andheri (East)
MUMBAI 400 093
Telephone : 2832 9295, 2832 7858, 2832 7891, 2832 7892

Branches : AHMEDABAD. BANGALORE. BHOPAL. BHUBANESHWAR. COIMBATORE. FARIDABAD. GHAZIABAD. GUWAHATI. HYDERABAD. JAIPUR. KANPUR. LUCKNOW. NAGPUR. NALAGARH. PATNA. PUNE. RAJKOT. THIRUVANANTHAPURAM. VISAKHAPATNAM.

Printed at New India Printing Press, Khurja, India