
**Information technology — Software
asset management —**

**Part 5:
Overview and vocabulary**

*Technologies de l'information — Gestion de biens de logiciel —
Partie 5: Vue d'ensemble et vocabulaire*

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 19770-5:2013

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 19770-5:2013



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2013

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

	Page
Foreword	iv
0 Introduction	v
1 Scope	1
2 Normative references	1
2.1 Approved references	1
2.2 References under development	1
2.3 World Wide Web Consortium (W3C) references	1
2.4 Internet Engineering Task Force (IETF) references	1
3 Terms and definitions	2
3.1 General terms and definitions	2
3.2 Terms and definitions related to processes	5
3.3 Terms and definitions related to information structures	6
4 Software asset management (SAM)	9
4.1 Introduction	9
4.2 The need to manage software assets	9
4.3 Foundation principles	11
4.4 Relationships to principles defined in other standards	12
4.5 Principles of process definitions	12
4.6 Evaluation of process definition conformance	13
4.7 Principles of information structures	13
4.8 Evaluation of information structure definition conformance	14
4.9 Critical success factors	14
5 SAM family of standards	14
5.1 General information	14
5.2 Standards specifying processes	15
5.3 Technical reports providing guidance for process standards	16
5.4 Standards specifying information structures	16
5.5 Technical reports providing guidance for information structure standards	17
5.6 Overview standards	17

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 19770-5 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

ISO/IEC 19770 consists of the following parts, under the general title *Information technology — Software asset management*:

Part 1: Processes and tiered assessment of conformance

Part 2: Software identification tag

Part 5: Overview and vocabulary

The following parts are under preparation:

Part 3: Software entitlement tag

Part 7: Tag management

Guidelines for mapping of industry SAM practices with the ISO/IEC 19770 family of standards and Guidelines for the application of ISO/IEC 19770-1 for small organizations will form the subjects of future Parts 8 and 11, respectively.

0 Introduction

0.1 Overview

International Standards in the ISO/IEC 19770 family of standards for software asset management (SAM) address both the processes and technology for managing software assets and related IT assets. Because IT is an essential enabler for almost all activity in today's world, these standards must integrate tightly into all of IT. For example, from a process perspective, SAM standards must be able to be used with all Management System Standards, because software and software management are essential components of any modern Management System. From a technology perspective, SAM standards for information structures provide not only for data interoperability of software management data, but also provide the basis for many related benefits such as more effective security in the use of software. SAM standards for information structures also facilitate significant automation of IT functionality, such as improved authentication of software and linking to national vulnerability databases for more automated exposure identification and mitigation.

0.2 SAM family of standards

The ISO/IEC 19770 family of standards is intended to assist organizations of all types to implement and operate a software asset management system using both process and technology. The ISO/IEC 19770 family of standards consists of the parts listed in the Foreword.

NOTE ISO/IEC 19770-4, ISO/IEC 19770-6, ISO/IEC 19770-9 and ISO/IEC 19770-10 are either related to projects that have been withdrawn, or are reserved for future use.

0.3 Purpose of this part of ISO/IEC 19770

This part of ISO/IEC 19770 provides an overview of software asset management, which is the subject of the ISO/IEC 19770 family of standards, and defines related terms.

This part of ISO/IEC 19770 is divided into the following clauses:

- [Clause 1](#) is the scope;
- [Clause 2](#) describes the normative references;
- [Clause 3](#) describes the terms, definitions, symbols, and abbreviations used in this standard;
- [Clause 4](#) introduces software asset management, describes the alignment of SAM standards with other ISO and ISO/IEC standards, and defines principles of SAM processes and data structures;
- [Clause 5](#) gives an overview of the SAM standards family;

The terms and definitions provided in this part of ISO/IEC 19770:

- a) cover commonly used terms and definitions in the ISO/IEC 19770 family of standards;
- b) will not cover all terms and definitions applied within the ISO/IEC 19770 family of standards; and
- c) do not limit the ISO/IEC 19770 family of standards in defining terms for their own use.

To reflect the changing status of the SAM family of standards, this part of ISO/IEC 19770 is expected to be updated on a more frequent basis than would normally be the case for other ISO/IEC standards.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 19770-5:2013

Information technology — Software asset management —

Part 5: Overview and vocabulary

1 Scope

This part of ISO/IEC 19770 provides:

- a) an overview of the ISO/IEC 19770 family of standards;
- b) an introduction to software asset management (SAM);
- c) a brief description of the foundation principles and approaches on which SAM is based; and
- d) consistent terms and definitions for use throughout the ISO/IEC 19770 family of standards.

This part of ISO/IEC 19770 is applicable to all types of organization (e.g. commercial enterprises, government agencies, and non-profit organizations).

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

2.1 Approved references

ISO/IEC/IEEE 24765, *Systems and software engineering — Vocabulary*¹⁾

2.2 References under development

ISO 55000, *Asset management — Overview, principles and terminology*²⁾

2.3 World Wide Web Consortium (W3C) references

Extensible Markup Language (XML) 1.1 (Second Edition), W3C Recommendation, <http://www.w3.org/TR/2008/REC-xml-20081126/>

XML Schema Definition Language (XSD) 1.1 Part 1: Structures, W3C Recommendation, <http://www.w3.org/TR/xmlschema11-1/>

XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, W3C Recommendation, <http://www.w3.org/TR/xmlschema11-2/>

2.4 Internet Engineering Task Force (IETF) references

RFC 1034, *Domain Names – Concepts and Facilities*, November 1987, <http://tools.ietf.org/html/rfc1034>

1) ISO/IEC/IEEE 24765 is a “snapshot” of the SEVOCAB (systems and software engineering vocabulary) database, which is available at: <http://www.computer.org/sevocab>

2) To be published.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply. The definitions are subdivided into functional groupings for ease of reference.

3.1 General terms and definitions

3.1.1

asset

something that has potential or actual value to an organization

Note 1 to entry: Value can be tangible or intangible, financial or non-financial, and includes consideration of risks and liabilities. It can be positive or negative at different stages of the asset's life.

Note 2 to entry: For most organizations, physical assets usually refer to equipment, inventory and properties owned by the organization. Physical assets are the opposite of intangible assets, which are non-physical assets such as leases, brands, digital assets, use rights, licences, intellectual property rights, reputation or agreements.

Note 3 to entry: A grouping of assets referred to as an asset system could also be considered as an asset.

[SOURCE: ISO 55000:—, 3.2.1]

3.1.2

asset management

coordinated activities of an organization to realize value from *assets* (3.1.1)

[SOURCE: ISO 55000:—, 3.3.1, modified — The Note has been deleted.]

3.1.3

baseline

formally approved version of a *configuration item* (3.2.1), regardless of media, formally designated and fixed at a specific time during the configuration item's life cycle

[SOURCE: ISO/IEC/IEEE 24765:2010, 3.240, definition 2]

3.1.4

bundle

grouping of products which is the result of a marketing/licensing strategy to sell entitlements to multiple products as one purchased item

Note 1 to entry: A bundle can be referred to as a "suite", if the products are closely related and typically integrated (such as an office suite containing a spreadsheet, word processor, presentation and other related items).

Note 2 to entry: Bundles can also refer to software titles that are less closely related such as a game, a virus scanner and a utility "bundled" together with a new computer, or to groups of entitlements, such as multiple entitlements for a backup software product.

3.1.5

computing device

functional unit that can perform substantial computations, including numerous arithmetic operations and logic operations with or without human intervention

Note 1 to entry: A computing device can consist of a stand-alone unit, or several interconnected units. It can also be a device that provides a specific set of functions, such as a phone or a personal organizer, or more general functions such as a laptop or desktop computer.

[SOURCE: ISO/IEC/IEEE 24765:2010, 3.513 (computer), modified — "with or" has been added to the definition.]

3.1.6**corporate board or equivalent body**

person or group of people who assumes legal responsibility for conducting or controlling an organization at the highest level

3.1.7**customer**

organization or person that receives a product or service

[SOURCE: ISO/IEC/IEEE 24765:2010, 3.696, definition 1]

3.1.8**end-user**

person or persons who will ultimately be using the system for its intended purpose

Note 1 to entry: In the ISO/IEC 19770 family of standards, an end user will generally be defined in terms of a specific *software component* (3.1.15) of a system.

[SOURCE: ISO/IEC/IEEE 24765:2010, 3.696 (end user), definition 1, modified — Note 1 to entry has been added.]

3.1.9**license compliance audit**

audit that reconciles license-related information from multiple information sources, such as entitlement consumption against entitlement rights

Note 1 to entry: For a formal definition of audit, see Annex SL of the ISO/IEC Directives, Part 1 and Consolidated ISO Supplement.

3.1.10**license model**

class of licenses with common characteristics

Note 1 to entry: Examples of license models can be site license, OEM License, and per-computer.

3.1.11**platform**

type of computer or hardware device and/or associated operating system, or a virtual environment, on which software can be installed or run

Note 1 to entry: A platform is distinct from the unique instances of that platform, which are typically referred to as devices or instances.

3.1.12**SAM program scope**

clear statement listing of all parts of the organization and types of software, assets, platforms, etc. covered by a SAM program

3.1.13**software**

all or part of the programs, procedures, rules, and associated documentation of an information processing system

Note 1 to entry: There are multiple definitions of software in use. For the purpose of this part of ISO/IEC 19770, it is typically important to include both executable and non-executable software, such as fonts, graphics, audio and video recordings, templates, dictionaries, documents and information structures such as database records.

[SOURCE: ISO/IEC/IEEE 24765:2010, 3.2741, definition 1, modified – Note 1 to entry has been added]

3.1.14
software asset management
SAM

control and protection of software and related assets within an organization, and control and protection of information about related assets which are needed in order to control and protect software assets

Note 1 to entry: For reference, a corresponding industry definition is “all of the infrastructure and processes necessary for the effective management, control and protection of the software assets within an organization, throughout all stages of their lifecycle”.

3.1.15
software component

entity with discrete structure, such as an assembly or software module, within a system considered at a particular level of analysis

Note 1 to entry: In this part of ISO/IEC 19770, software component refers to a part of a whole, such as a component of a software product, a component of a software identification tag, etc.

3.1.16
software consumer

entity that uses an *entitlement* (3.3.5) of a *software package* (3.1.21)

3.1.17
software creator

person or organization that creates a *software product* (3.1.23) or *package* (3.1.21)

Note 1 to entry: This entity might or might not own the rights to sell or distribute the software

3.1.18
software entitlement

software license use rights as defined through agreements between a *software licensor* (3.1.20) and a *software consumer* (3.1.16)

Note 1 to entry: Effective use rights take into account any contracts and all applicable licenses, including full licenses, upgrade licenses and maintenance agreements.

3.1.19
software license

legal rights to use software in accordance with terms and conditions specified by the *software licensor* (3.1.20)

Note 1 to entry: “Using a software product” can include: accessing, copying, distributing, installing and executing the software product, depending on the license’s terms and conditions.

3.1.20
software licensor

person or organization who holds the rights to issue a software license for a specific software package

3.1.21
software package

complete and documented set of *software* (3.1.13) supplied for a specific application or function

Note 1 to entry: In the ISO/IEC 19770 family of standards, the term software package refers to the set of files associated with a specific set of business functionalities that can be installed on a computing device and has a set of specific licensing requirements. In the ISO/IEC 19770 family of standards, the terms “software product” and “software package” are used synonymously depending on the context of the item described.

3.1.22**software packager**

entity that packages or bundles software created by others

Note 1 to entry: This can be done for example by a value added reseller who bundles a software package to work with an embedded system, or by a software reseller who is licensed to combine a number of different software products into a single bundle.

3.1.23**software product**

complete set of *software* (3.1.13) designed for delivery to a *software consumer* (3.1.16) or *end-user* (3.1.8) that may contain computer programs, procedures and associated documentation and data

Note 1 to entry: In the ISO/IEC 19770 family of standards, the terms “software product” and “software package” are used interchangeably depending on the context of the item described.

3.1.24**software usage**

consumption against a *software entitlement* (3.1.18) measured as defined by the terms and conditions of that entitlement

Note 1 to entry: Depending on the specific terms and conditions, usage can include accessing, copying, distributing, installing and executing software.

3.1.25**stock keeping unit****sku**

identification, usually alphanumeric, of a particular product that allows it to be tracked for inventory and *software entitlement* (3.1.18) purposes

Note 1 to entry: The term “stock keeping unit” is typically associated with unique products for sales purposes, such as software entitlements. It may not correspond uniquely to specific software products, but may instead represent packages of software, and/or specific terms and conditions related to software products such as whether it relates to a full product, upgrade product, or maintenance on an existing product.

3.2 Terms and definitions related to processes**3.2.1****configuration item****CI**

component of an infrastructure or an item which is or will be, under control of configuration management

Note 1 to entry: Configuration items may vary widely in complexity, size and type, ranging from an entire system including all hardware, software and documentation, to a single module or a minor hardware component.

Note 2 to entry: Configuration items are commonly defined as part of service management practice and may vary widely in complexity, size and type, ranging from an entire system including all hardware, software and documentation, to a single module or a minor hardware component.

[SOURCE: ISO/IEC/IEEE 24765:2010, 3.563, definition 3, modified — Note 2 to entry has been added]

3.2.2**definitive software library****DSL**

secure storage environment, formed of physical media or of one or more electronic software repositories, capable of control and protection of definitive authorized versions of all software *configuration items* (3.2.1) and masters of all software controlled by *SAM* (3.1.14)

3.2.3**local SAM owner**

individual at a level of the organization below that of the *SAM owner* (3.2.7) who is identified as being responsible for SAM for a defined part of the organization

**3.2.4
procedure**

specified way to carry out an activity or process

Note 1 to entry: When a procedure is specified as an outcome, the resulting deliverable will typically specify what must be done, by whom, and in what sequence. This is a more detailed level of specification than for a *process* (3.2.5).

**3.2.5
process**

set of interrelated or interacting activities, which transforms inputs into outputs

Note 1 to entry: When a process definition is specified as an outcome, the resulting deliverable will typically specify inputs and outputs, and give a general description of expected activities. However, it does not require the same level of detail as for a *procedure* (3.2.4).

[SOURCE: ISO/IEC/IEEE 24765:2010, 3.2217, definition 1, modified — Note 1 to entry has been added.]

**3.2.6
release**

collection of one or more new or changed configuration items deployed into the live environment as a result of one or more changes

[SOURCE: ISO/IEC 20000-1:2011, 3.2.3]

**3.2.7
SAM owner**

individual at a senior organization-wide level who is identified as being responsible for SAM (3.1.14)

**3.2.8
SAM practitioner**

individual involved in the practice or role of managing software assets

Note 1 to entry: A SAM practitioner is often involved in the collection or reconciliation of software inventory and/or software entitlements.

**3.2.9
tier**

grouping of process definitions

**3.2.10
value baseline**

measure of a set of assets before an optimization, assigning relevant values to each group of assets being tracked

**3.2.11
reseller**

organization that purchases goods or services with an intention of selling them to another customer and possibly supporting them

3.3 Terms and definitions related to information structures

**3.3.1
child tag**

tag (3.3.15) that has a subsidiary relationship to another tag

Note 1 to entry: For example, child entitlement tags could be created for allocation purposes.

3.3.2 configuration management database CMDB

database containing all the relevant details of each *configuration item* (3.2.1) and details of the important relationships between them

Note 1 to entry: When aligning service management with SAM, it may be convenient for the organization to ensure that CIs cover all software within the *scope* (3.1.12) of SAM, i.e. it may be an advantage for anticipated manifestations of controlled/licensed software usage to be fully mapped to CIs and so accountable through all the service management processes using CIs.

[SOURCE: ISO/IEC/IEEE 24765:2010, 3.566, modified — Note 1 to entry has been added.]

3.3.3 consolidation tag

type of *tag* (3.3.15) used to represent a grouping of multiple other tags

Note 1 to entry: For example, in entitlement management a consolidation tag may be used to facilitate subsequent creation of children tags with entitlement quantities which do not match the original granted entitlements.

3.3.4 element

component of a *tag* (3.3.15) that provides information related to the entity represented by the tag

3.3.5 entitlement

see *software entitlement* (3.1.18)

3.3.6 extended element

element (3.3.4) within a tag that provides additional information beyond that documented explicitly in the standard

3.3.7 extensible markup language XML

license-free and platform-independent markup language that carries rules for generating text formats that contain structured data

[SOURCE: W3C Recommendation *Extensible Markup Language (XML) 1.1 (Second Edition)*, 1.2]

3.3.8 globally unique identifier GUID

16-byte string of characters that is generated in a manner that gives a high probability that the string is unique in any context

Note 1 to entry: Other globally unique identifier algorithms can be used in some situations. In general, alternative algorithms use Uniform Resource Identifier (URI) based structures, so the id owner's registration identifier (regid) is included in the identifier.

Note 2 to entry: In this part of ISO/IEC 19770, GUID as an all capitalized term refers specifically to the 16 byte version. If the term is in lowercase (guid), it refers to a general algorithm that can use either a URI, or a 16-byte-based identifier.

3.3.9 legacy software

software (3.1.13) originally created without tags

3.3.10 mandatory element

element (3.3.4) that is required to be present in a tag in order to claim conformance with a standard

3.3.11

optional element

element (3.3.4) that may or may not be present in a tag

3.3.12

recommended element

element (3.3.4) that is not required to be present in a tag but is strongly encouraged to be included by a *tag creator* (3.3.16)

3.3.13

registration identifier

regid

identifier created from a domain name (see RFC 1034) and the date when the domain was owned by a specific individual or company, allowing an individual or company to have their own unique namespace and be their own registration authority for all software configuration items they publish without requiring a separate industry based registration authority

3.3.14

software identification tag

SWID tag

file comprised of *mandatory elements* (3.3.10), *optional elements* (3.3.11) and *extended elements* (3.3.6) containing authoritative identification information about a *software configuration item* (3.2.1)

3.3.15

tag

information structure that provides authoritative information about a software asset in order to facilitate its management

3.3.16

tag creator

entity that initially creates a tag

Note 1 to entry: This entity can be part of the organization that created the software, in which case the tag creator and software creator will be the same. The tag creator can also be a third party organization unrelated to the software creator (such as in the case where tags are created for legacy software by third party organizations).

3.3.17

tag modifier

software packager (3.1.22) or *software consumer* (3.1.16) that modifies a tag after it has been created

Note 1 to entry: Modification of any tag is limited to the elements that the software licensor has authorized and is done based on license or contractual agreements with the tag creator and/or software creator. The tag modifier can be allowed to add values to a software identification tag (such as the case of a reseller adding details about where the product was purchased), or can be allowed to modify existing portions of the tag (such as the case of a VAR making a set of software look like it comes from a single entity).

3.3.18

tag provider

entity that creates [*tag creator* (3.3.16)] or modifies [*tag modifier* (3.3.17)] software identification tags for a *software package* (3.1.21)

Note 1 to entry: A tag provider can be part of the software provider organization, or can be a third party organization or the software consumer.

3.3.19

Uniform Resource Identifier

URI

compact sequence of characters that identifies an abstract or physical resource available on the Internet

Note 1 to entry: The syntax used for URIs is defined in IETF RFC 3986.

[SOURCE: IETF RFC 3986, 1]

3.3.20**valid**

tag that follows the specified *XML Schema document* (3.3.22) and is valid from an XML perspective

3.3.21**version**

unique string of number and letter values indicating a unique revision of an item

Note 1 to entry: Versions are often referred to in software to identify revisions of software that provide unique functionality or fixes. A version typically has multiple parts with at least a major version indicating large changes in functionality or user interface changes and a minor version indicating smaller changes in functionality or user interface changes.

3.3.22**XML schema document****XSD**

document that describes the structure of XML information

[SOURCE: W3C *XML Schema Definition Language (XSD) 1.1 Part 1: Structures*, 1]

4 Software asset management (SAM)**4.1 Introduction**

Asset management (see 3.1.2) is a well-established discipline that defines a system consisting of interrelated and interacting parts to establish policies, objectives, strategies, plans and activities to maximize performance and value from a portfolio of assets in the delivery of organizational objectives over a specified period of responsibility. In this context “parts” includes business processes and governance activities, “performance” includes operational, financial and legal performance, and “value” includes minimization of costs and risks.

Asset management is applied at every stage of an asset’s lifecycle. In many industries asset management plays a key role in determining the operational performance and efficiency of an organization.

Information technology (IT) asset management (ITAM) is a sub-discipline of asset management that is specifically aimed at managing the life cycles and total costs of IT equipment and the infrastructures that they comprise. ITAM is vital to support life cycle management and strategic decision making for the IT environment, and incorporates specific approaches to handle the portability of some types of IT asset (e.g. laptops and smart phones).

Software asset management (SAM, see 3.1.14) is a further sub-discipline that is specifically aimed at managing the acquisition, release, deployment, maintenance and eventual retirement of software assets. SAM processes provide effective management, control and protection of software assets within an organization. SAM incorporates specific approaches to handle challenges that are unique to SAM, such as the real-time mobility of software assets in a distributed and virtualized environment, where such mobility leads to unmanaged growth in number and diversity of software assets.

Although SAM is a sub-discipline of ITAM, for most practical purposes the scope of both is the same, because SAM requires the inclusion in its scope of all other related assets that are necessary to use or manage software in scope. In practical terms this means that all IT hardware assets used for deploying, executing, and managing software must be included in the scope of SAM.

4.2 The need to manage software assets**4.2.1 General**

The well-known and inexorable reduction in the cost of computing hardware means that a larger and larger portion of the cost of creating and maintaining an IT infrastructure is related to software assets rather than physical assets. The functionality, complexity and importance of software assets have also

markedly increased. Thus it is increasingly important that an organization gains the best value possible for the lifecycle costs of those assets.

Good practice in SAM should result in the types of benefits described in the following subclauses, namely:

- a) direct benefits (see [4.2.2](#));
- b) cost control (see [4.2.3](#)); and
- c) risk management and mitigation (see [4.2.4](#)).

In addition, certifiable good practice should allow management and other organizations to place reliance on the adequacy of these processes, and the benefits described below should be achieved with a high degree of assurance.

4.2.2 Direct benefits

SAM should provide the following direct benefits:

- a) effective deployment of software to the organization supporting the achievement of business objectives;
- b) better quality decision making because of more complete and more transparent information availability (for example, IT procurement and system development decisions may be made more quickly and more reliably with better quality data);
- c) being able to deploy new systems and functionality more quickly and reliably in response to market opportunities or demands;
- d) providing IT which is more closely aligned to business needs, thus ensuring that all users have access to appropriate software and applications;
- e) being able to handle the IT aspects of business acquisitions, mergers or demergers more quickly;
- f) higher quality IT strategy, enabling the creation of a flexible infrastructure based on modern architectures e.g. cloud-based services and private cloud resiliency; and
- g) better personnel motivation and client satisfaction through having less IT problems.

4.2.3 Cost control

SAM should facilitate cost control including in the following areas:

- a) reduced direct costs of software and related assets, such as by negotiating better pricing through improved use of volume contracting arrangements, and by avoiding purchasing new licenses when old ones can be redeployed;
- b) reduced time and cost for negotiating with suppliers because of better information availability;
- c) reduced costs through improved financial control, such as through better invoice reconciliation and more accurate forecasting and budgeting;
- d) reduced infrastructure costs for managing software and related assets, by ensuring that required processes are efficient and effective;
- e) reduced support costs which are significantly affected by the quality of SAM processes, both directly within IT and indirectly within end-user areas; and
- f) better identification of components with the IT infrastructure that incur high costs.

4.2.4 Risk management and mitigation

4.2.4.1 Introduction

SAM should assist in the management and mitigation of risk in a number of areas, as follows:

- a) operational (see [4.2.4.2](#));
- b) security (see [4.2.4.3](#)); and
- c) compliance (see [4.2.4.4](#)).

4.2.4.2 Operational risk management and mitigation

SAM should facilitate the management of business risks including:

- a) risk of interruption to IT services; and
- b) risk of deterioration in the quality of IT services

4.2.4.3 Security risk management and mitigation

SAM should help manage and strengthen security through the following:

- a) higher assurance about the authorization of installed and/or used software;
- b) better identification of non-authorized software; and
- c) tighter control of the patch process for installed software.

4.2.4.4 Compliance risk management and mitigation

SAM should promote, and simplify the management, of compliance through minimization of the following:

- a) legal and regulatory exposure, especially in regard to personally identifiable information and privacy;
- b) license non-compliance;
- c) policy non-compliance; and
- d) risk of damage to public image arising from any of the above.

4.3 Foundation principles

Software asset management (SAM) as defined in the ISO/IEC 19770 standards is based upon the following principles:

- a) That the scope (see [3.1.12](#)) of the SAM program ultimately includes all types of software (see [3.1.13](#)) and related assets, regardless of the nature of the software. For example, it can be applied to executable software (such as application programs, operating systems and utility programs), non-executable software (such as fonts, graphics, audio and video recordings, templates, dictionaries, documents and data) and software used other than by installation (such as software as a service and connection-based usage);
- b) That the definitions should be applicable to a wide variety of organizations from small to international, to situations where SAM is performed in-house as well as outsourced, and to implementation approaches that range from the highly centralized to completely distributed;
- c) That SAM should support a variety of delivery mechanisms (e.g. mobile, premise-based, cloud-based, hosted etc.); and

d) That SAM should support a variety of license models (see [3.1.10](#)).

The following forms of assets are within the scope of the ISO/IEC 19770 standards:

- **software for use:** all types of software as in a) above;
- **entitlements:** software use rights, reflected by full ownership (as for in-house developed software) and licenses (as for most externally sourced software, whether commercial or open-source); and
- **media:** holding copies of software for use.

4.4 Relationships to principles defined in other standards

4.4.1 Introduction

SAM as defined in the ISO/IEC 19770 family of standards has been defined for consistency with the principles of other ISO standards families, as follows:

- a) ISO 9001 (see [4.4.2](#));
- b) ISO/IEC 20000 (see [4.4.3](#));
- c) ISO/IEC 27000 (see [4.4.4](#)); and
- d) ISO 55000 (see [4.4.5](#)).

4.4.2 Relationship to ISO 9001 principles

SAM planning and implementation processes as defined in the ISO/IEC 19770 family of standards in principle map to the 'Plan-Do-Check-Act' processes of ISO 9001.

4.4.3 Relationship to ISO/IEC 20000 principles

SAM processes as defined in the ISO/IEC 19770 family of standards are closely aligned to and intended to closely support the principles of IT service management as defined in ISO/IEC 20000.

4.4.4 Relationship to ISO/IEC 27000 principles

SAM processes as defined in the ISO/IEC 19770 family of standards are intended to support the security or Integrated Security Management System (ISMS) requirements that are defined or described in ISO/IEC 27000 family.

4.4.5 Relationship to ISO 55000 principles

SAM processes as defined in the ISO/IEC 19770 family of standards should be usable with, and dovetail into, the generic asset management system defined in the ISO 55000 family.

4.5 Principles of process definitions

The ISO/IEC 19770 family of standards contain process definitions in order to be able to codify best practices that form the current state of the art in SAM, and also to facilitate the benchmarking of SAM in different organizations.

However the process definitions (see [3.2.5](#)) used in the SAM family of standards follow a specific structure in order to be applicable to a wide variety of sizes and types of organization, as follows:

- a) The processes are defined in terms of the elements of title, objective, and outcomes. The definitions do not include activities, which are actions that may be used to achieve the outcomes. Those outcomes specified are designed to be readily assessable, but will not necessarily indicate the breadth of activities that may be needed to produce them;

- b) The processes are not detailed in terms of methods or procedures required to meet the requirements for outcomes of a process;
- c) The sequence of steps an organization should follow to implement SAM is not specified, nor is any sequence implied by the sequence in which processes are described. The only sequencing which is relevant is that which is required by content and context. For example, planning should precede implementation; and
- d) Documentation is not detailed in terms of name, format, explicit content and recording media.

NOTE These principles are expected to evolve in future process standards as they are required to meet additional requirements e.g. the requirements for a Management System Standard as defined in Annex SL of the supplement to the ISO/IEC Directives.

4.6 Evaluation of process definition conformance

Conformance to process definitions within the ISO/IEC 19770 family of standards may be performed in one of two ways:

- a) By demonstrating that all of the requirements of the process definition have been satisfied using the outcomes as evidence; or
- b) By demonstrating that all of the objectives of the process definition have been achieved.

When full conformance is achieved by demonstrating that all of the objectives for a defined tier (see [3.2.9](#)) have been met, two further requirements exist:

- c) Where a process area includes outcomes in different tiers, the objective for that process area shall be interpreted correspondingly for assessments of each tier; and
- d) An assessor shall, in addition to reviewing evidence demonstrating that all objectives are achieved, still take into account the specified outcomes for the respective tier. Where there is any failure to meet all specified outcomes, for each such outcome the assessor shall explain in writing their reason(s) for accepting the objectives of a tier are nevertheless still fully satisfied without need for that outcome.

4.7 Principles of information structures

The information structures defined in the ISO/IEC 19770 family of standards adhere to the following principles:

- a) The structures are designed to provide interoperability for software management data independent of vendor, platform or technology (such as virtualization);
- b) the structures are designed to be usable throughout the software product lifecycle i.e. from the creation stage, through the packaging and installation stages, to the installation, usage, and eventual de-installation stage;
- c) the structures are designed to incorporate a unique software_id that corresponds to a unique product at the binary level for distribution/update purposes. Uniqueness is guaranteed by a combination of a unique tag creator name and a tag creator maintained unique_id. A number of different information structures are interlinked by the unique software_id;
- d) the structures are designed to minimize the need for a registration authority;
- e) the structures are designed to both be readable by humans and interpretable by programs;
- f) the structures are designed to be neutral with respect to the platform with which the software assets are associated; and
- g) standard locations are defined for each type of platform where the structures are to be located.