PAS 754:2014

Software trustworthiness – Governance and management – Specification

MAKING SOFTWARE BETTER

HM Government

bsi.

Publishing and copyright information

ISBN 978 0 580 83242 0

ICS 35.040

Publication history

First published May 2014

Contents

Foreword

This PAS was sponsored by the Trustworthy Software Initiative (TSI), a public good activity supported by the UK Government National Cyber Security Programme (NCSP) on behalf of stakeholders from the public and private sectors and academia. Its development was facilitated by BSI Standards Limited and it was published under licence from The British Standards Institution. This PAS came in to effect on 30 May 2014.

Acknowledgement is given to the technical author Ian Bryant, from TSI, and to the following organizations that were involved in the development of this PAS as members of the steering group:

- Association of British Certification Bodies (ABCB)
- Centre for the Protection of National Infrastructure (CPNI)
- Department for Business, Innovation & Skills (BIS)
- Group 5 Training Limited
- The Institution of Engineering and Technology (IET)
- Microsoft
- The Motor Industry Software Reliability Association (MISRA)
- Nexor Limited
- Oxford Brookes University
- QinetiQ Group
- Trustworthy Software Initiative (TSI)

Acknowledgement is also given to the members of the wider review panel consulted in the development of this PAS.

This PAS is not to be regarded as a British Standard. It will be withdrawn upon publication of its content in, or as, a British Standard.

The PAS process enables a specification to be rapidly developed in order to fulfil an immediate need in industry. A PAS can be considered for further development as a British Standard, or constitute part of the UK input into the development of a European or International Standard.

Use of this document

It has been assumed in the preparation of this PAS that the execution of its provisions will be entrusted to appropriately qualified and experienced people, for whose use it has been produced.

Presentational conventions

The provisions of this standard are presented in roman (i.e. upright) type. Its requirements are expressed in sentences in which the principal auxiliary verb is "shall".

Commentary, explanation and general informative material is presented in smaller italic type, and does not constitute a normative element. The word "should" is used to express recommendations, the word "may" is used to express permissibility and the word "can" is used to express possibility, e.g. a consequence of an action or an event.

Spelling conforms to The Shorter Oxford English Dictionary. If a word has more than one spelling, the first spelling in the dictionary is used.

Contractual and legal considerations

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with this PAS cannot confer immunity from legal obligations.

Executive summary

From smart phones to power stations, airliners to e-commerce, our economy and society is increasingly dependent on software in many different guises. This makes software trustworthiness an underlying concern for all those who commission, write and use it.

This PAS, sponsored by the UK Trustworthy Software Initiative, is to provide a consensus specification for software trustworthiness, either as a stand-alone document or as a companion and complement to other relevant standards.

This specification identifies five aspects of software trustworthiness: safety, reliability, availability, resilience and security. The set of principles and techniques for any software implementation needs to be suited to the context and intended use.

This document describes a widely applicable approach to achieving software trustworthiness, which is based on the following concepts:

- Governance. Before producing or using any software which has a trustworthiness requirement, an appropriate set of governance and management measures shall be set up.
- Risk assessment. The risk assessment process involves considering the set of assets to be protected, the nature of the adversities that may be faced, and the way in which the software may be susceptible to such adversities.
- Control application. Risk shall be managed through the treatment of risk by the application of appropriate personnel, physical, procedural and technical controls.
- Compliance. A compliance regime shall be set up to ensure that creators and users of software ensure that governance, risk and control decisions have been implemented.

It also recommends the use of a trustworthy software management system, either as a standalone entity or by relevant extension to existing management system(s), including:

- creating a trustworthy software defect and deviation list;
- implementing control measures;
- creating a trustworthy software release authority;
- building a trustworthy software constraint and dependency model;
- using of trustworthy software release notices.

0 Introduction

0.1 Aim

The aim of this PAS, sponsored by the UK Trustworthy Software Initiative (TSI), is to provide a specification for software trustworthiness.

0.2 Objectives

This specification is intended to be widely applicable to software in its many guises from embedded equipment through consumer devices to industrial control systems. It aims to provide a consensus specification for software trustworthiness, either as a stand-alone document, or as a companion and complement to other relevant standards, by collating good practice from the five main facets of trustworthiness that currently typically operate in isolation (safety, reliability, availability, resilience and security).

In conjunction with methodologies such as *TickITplus*, a UK scheme that embraces quality management across IT in the form of a capability maturity method, and other similar frameworks PAS 754 could provide a foundation for software trustworthiness within organizations.

It supports the TSI's objectives as a public good initiative to improve software performance across organizations in all areas.

By helping to improve software quality, this specification could result in significant savings for the economy and reduce the risk of major disruptions to a range of industries across both the private and public sectors.

NOTE *See* Risk and Responsibility in a Hyperconnected World *[1].*

The requirements of PAS 754 can enable an organization to, for example:

- improve controls;
- improve operational effectiveness and efficiency;
- improve organizational learning.

These in turn can result in:

- improved stakeholder confidence and trust;
- increased likelihood of achieving objectives;
- reduced risk;
- enhanced business reputation.

0.3 Claims of conformance

0.3.1 General

An organization may claim conformance with PAS 754.

0.3.2 Form of claim

All claims are required to include a reference to PAS 754.

0.3.3 Basis of claim

A claim of conformance can be made on the basis of:

a) a first-party conformity assessment performed by the organization (self-assessment);

b) a second-party conformity assessment performed by, for example, a trade association; or

c) a third-party conformity assessment performed by an organization, such as a certification body, that is independent of both the organization and any linked trade association.

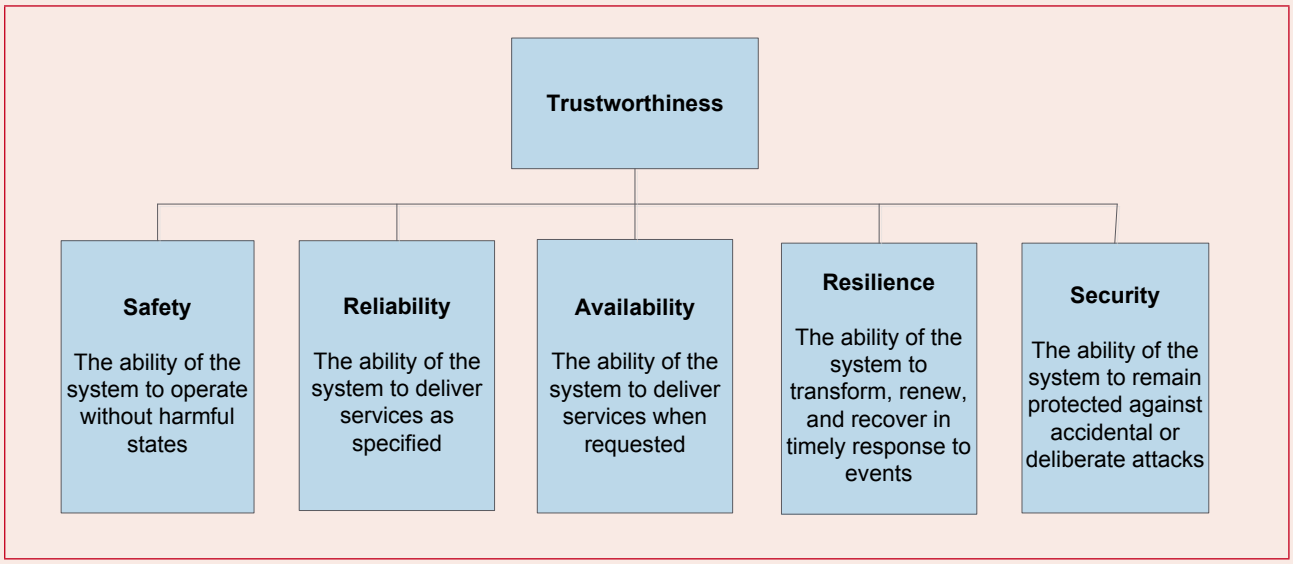
0.4 Context

0.4.1 Approach

The PAS is intended for any organization that seeks to establish or improve confidence in its software trustworthiness. It is applicable to all organizations regardless of their size, type and the nature of their business.

For this specification, software trustworthiness is identified as consisting of five facets, as described in Figure 1.

Figure 1 – Facets of trustworthiness



It is important that organizations review every software implementation to see which aspects apply and derive a set of principles and techniques to suit the context and intended use.

For each of these facets of trustworthiness there will be objectives, of varying complexity.

A common set of implied objectives apply to most software implementations, not least because of legal and regulatory requirements.

Safety aims to provide assurance that:

- the stated requirements are accurate and appropriate;
- safety issues are considered via safety requirements;
- the architecture and design are consistent with and reflect the stated requirements;
- technical defects are absent;
- application defects are absent;
- the stated requirements are identifiable in the low-level code and that all low-level code implements at least one stated requirement;
- the test data sets reflect the stated requirements;
- the test data sets cover the low-level code to a specified degree.

Reliability aims to provide assurance that:

- all the patterns of use are reflected in the stated requirements;
- there are no technical defects;
- the test data sets reflect patterns of use;
- there are no application defects.

Availability aims to provide assurance that:

- the stated requirements are accurate and appropriate;
- the architecture and design are consistent with and reflect the stated requirements;
- technical defects are absent;
- application defects are absent;
- the test data sets reflect the stated requirements.

Resilience aims to provide assurance that:

- the stated requirements are accurate and appropriate;
- the architecture and design are consistent with and reflect the stated requirements;
- technical defects are absent;
- application defects are absent;
- the test data sets reflect the stated requirements.

Security aims to provide assurance that:

- the security requirements consider all security issues;
- the architecture and design satisfy the security requirements;
- there are no security defects in the code;
- the test data reflects the security requirements.

0.4.2 Organizational controls

In order to deliver trustworthy software, an organization requires a set of underpinning controls that apply to all activities.

The software management system aims to provide assurance that:

- all personnel are appropriately qualified;
- adequate resources are allocated;
- all necessary communication takes place;
- activity proceeds in a series of measured steps;
- specific steps are performed independently;
- activity proceeds in a timely manner;
- all verification processes are completed within the specified criteria.

The software technical infrastructure aims to provide assurance that:

- all information, designs, algorithms and other such artefacts are retained for future use and analysis;
- the design and coding artefacts are adequately documented;
- all past and present versions of the software are available at any time and that future versions will similarly be available;
- all appropriate test data sets can be applied to the corresponding version and any future versions of the software;
- regression testing can be applied in order to ensure that the software changes only in the required manner.

0.4.3 Challenges

Software problems are generally characterized as one of three types:

- Weaknesses, which are generic classes of potential deficiency in software, such as buffer overflows.
- Vulnerabilities, which can be:
  - the existence of a generic weakness in a particular platform, such as a buffer overflow occurring in a specific operating system or application;
  - interactions between multiple software elements that bypass intended controls;
  - accidental actions of software developers that result in defects and errors;
  - deliberate actions of software developers that bypass intended controls, such as trap doors that permit unauthorized access to the system.
- Susceptibilities, which are the confirmed presence of one or more vulnerability within an implemented system, such as the presence of an operating system with a buffer overflow defect. Susceptibilities in systems stem from:
  - initial implementation;
  - changes to software, such as from adding new facilities or the correction of detected errors ('patching');
  - use of utility programs, which may be capable of circumventing security measures in the controlling or application software.

For the application of these terms specifically to software, see Clause **3**.

0.4.4 Tailoring

This PAS is scoped to include all aspects that contribute to trustworthiness of software, as illustrated in Figure 2.

This is achieved by using the appropriate elements of the consensus framework of measures – the trustworthy software framework (TSF) – decomposed as shown in Figure 3 and detailed in Clauses **5** and **6** and the Annexes.

This comprehensive trustworthy software framework (TSF) provides a domain- and implementation-agnostic way to reference the large existing body of knowledge, including functional safety, information security, and systems and software engineering and therefore acts as a collation of good practice for software trustworthiness.

When used as a stand-alone document for organizations with no current approach to software trustworthiness, this specification will facilitate the deployment of the TSF for software in its many guises from embedded equipment through consumer devices to industrial control systems.

For organizations that already address software trustworthiness though the lens of one or more of the five main facets of trustworthiness that typically operate in isolation (safety, reliability, availability, resilience and security), this specification provides a companion and complement to other relevant standards, and reviewing the concepts, principles and techniques in this specification alongside practices and management systems derived from individual facets allows the identification of gaps and enhancements.

Figure 2 – Aspects of trustworthiness

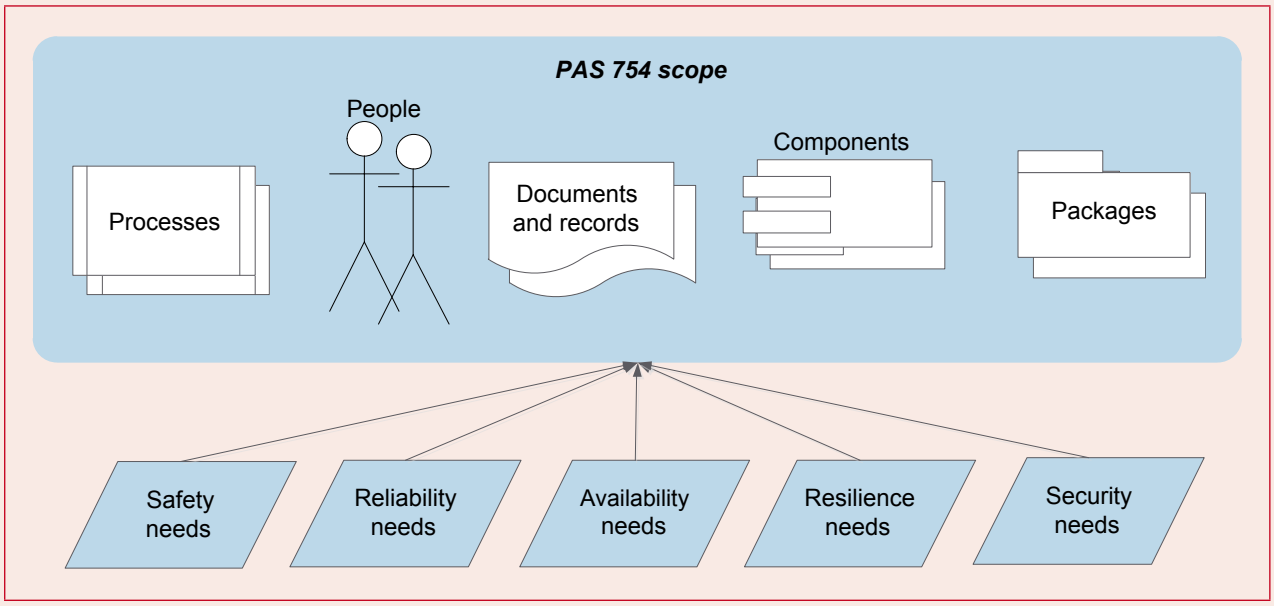
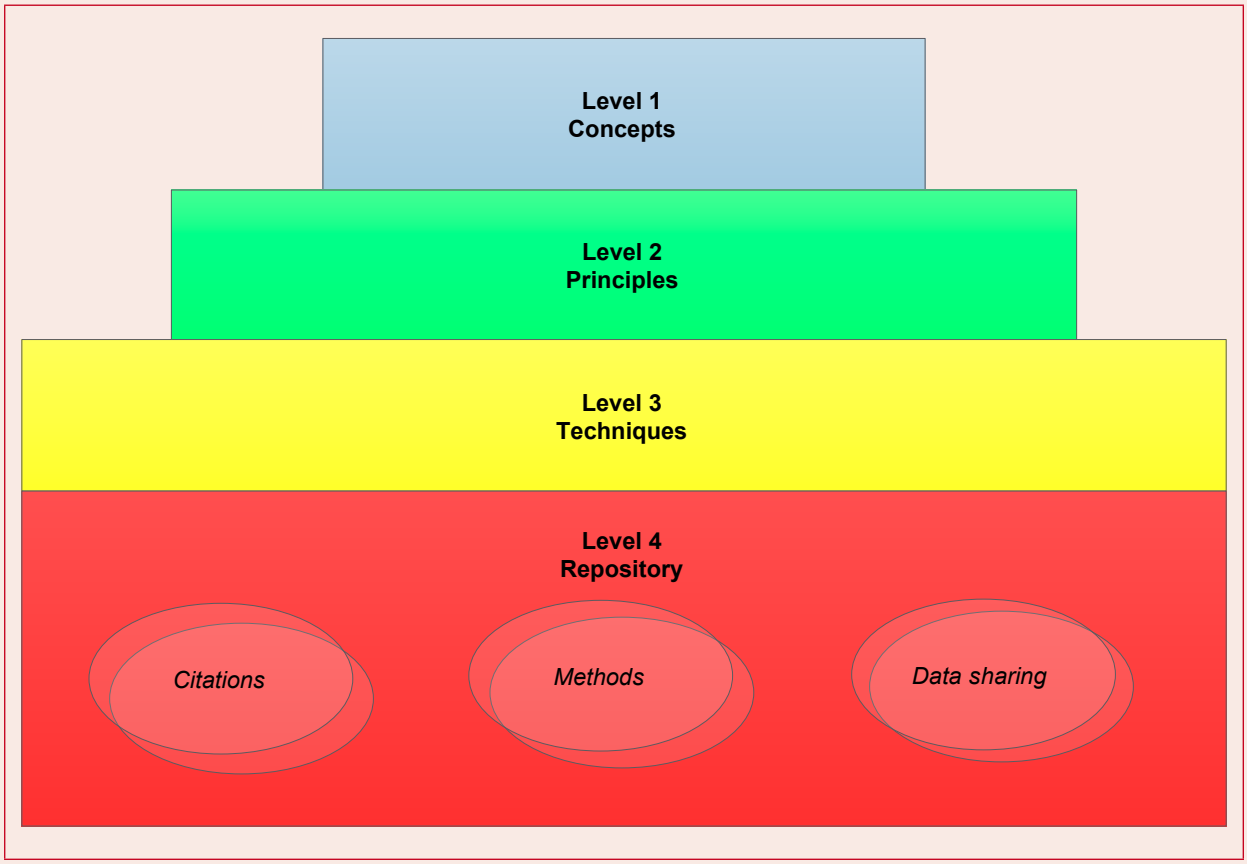


Figure 3 – Trustworthy software framework



The Trustworthy Software Initiative (TSI) is an independent UK organization supported by the public and private sectors and academia, which is charged with maintaining a repository of citations, methods and data sharing techniques about creating trustworthy software. More information is maintained on the website at: www.uk-tsi.org.

This PAS does not specify how any technique should be applied to a specific domain of application. This information is available in other standards, such as BS ISO/IEC 15408 and BS ISO/IEC 27001 for information security, and BS EN 61508 for functional safety.

0.4.5 Segmentation

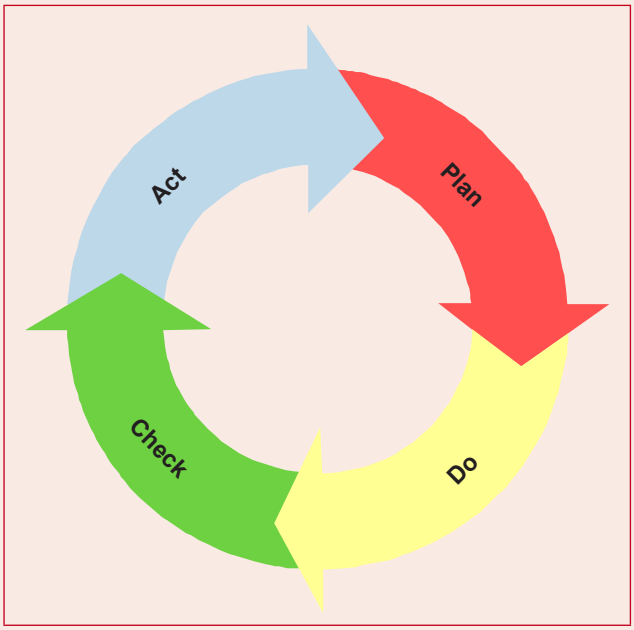
For the purposes of this PAS, the software audience can be divided into three groups:

- Mass Market with an Implicit Need[1)] (M/I) for software trustworthiness;
- Mass Market with an Explicit Need[2)] (M/E) for software trustworthiness;
- Niche Market with an Explicit Need[3)] (N/E) for software trustworthiness.

0.4.6 Continuous improvement

Continuous improvement of trustworthy software practices within an organization can be achieved using the PDCA (plan–do–check–act) cycle (see Figure 4).

Figure 4 – PDCA cycle



[1)] For the Mass Market with Implicit Needs, a majority of software trustworthiness requirements are perceived as non-functional e.g. a "non interference" property.
[2)] One or more functional requirements are for software trustworthiness.
[3)] For instance government and critical national infrastructure (CNI).

1 Scope

This PAS specifies requirements for software trustworthiness. It is intended to set out a widely applicable approach that can be customized for any organization and applied to software in its many guises from embedded equipment through consumer devices to industrial control systems.

This PAS defines the overall principles for effective software trustworthiness, and includes technical, physical, cultural and behavioural measures alongside effective leadership and governance. This PAS identifies the necessary tools, techniques and processes and addresses safety, reliability, availability, resilience and security issues.

This PAS does not specify the detailed processes or actions that an organization follows in order to achieve these outcomes.

NOTE 1 *These are defined in other standards, or can be defined by the organization.*

NOTE 2 *For organizations that already address software trustworthiness through the lens of one or more of the five main facets of trustworthiness that typically operate in isolation (safety, reliability, availability, resilience and security), this specification provides a companion and complement to other relevant standards, and reviewing the concepts, principles and techniques in this specification alongside practices and management systems derived from individual facets allows the identification of gaps and enhancements.*

This PAS is applicable to any organization aiming to adopt software trustworthiness practices.

2 Normative references

The following documents, in whole or 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 and amendments) applies.

BS ISO/IEC 27001:2013, *Information technology – Security techniques, Information security management systems – Requirements*

BS ISO/IEC/IEEE 42010, *Systems and software engineering – Architecture description*

ISO/IEC 15288, *Systems and software engineering – System life cycle processes*

3 Terms, definitions and acronyms

3.1 Terms and definitions

For the purposes of this PAS, the following terms and definitions apply.

3.1.1 adversity

superset of external factors likely to have undesirable effects on software, being the aggregate of the set of hazards (undirected events) and threats (directed, deliberate, hostile acts)

3.1.2 cyber security

collection of tools, policies, security concepts, security safeguards, guidelines, risk management approaches, actions, training, best practices, assurance and technologies that can be used to protect the cyber environment and organization and users' assets

NOTE *Organization and users' assets include connected computing devices, personnel, infrastructure, applications, services, telecommunications systems, and the totality of transmitted and/or stored information in the cyber environment. Cyber security strives to ensure the attainment and maintenance of the security properties of the organization and users' assets against relevant security risks in the cyber environment. The general security objectives include:*

- *availability;*
- *integrity, which may include authenticity and non-repudiation;*
- *confidentiality.*

[ITU-T Recommendation X.1205]

3.1.3 defect

non-fulfilment of an explicit or implicit requirement related to an intended or specified use

3.1.4 deferral

documented and risk managed decision to not resolve a defect or deviation

3.1.5 deviation

non-conformity with specification

3.1.6 organization

person or group of people that has its own functions with responsibilities, authorities and relationships to achieve its objectives

NOTE *The concept of organization includes, but is not limited to, sole trader, company, corporation, firm, enterprise, authority, partnership, institution, charity or association, or part or combination thereof, whether incorporated or not, public or private.*

[BS EN ISO 9000:2005, **3.3.1**]

3.1.7 risk management

coordinated activities to direct and control an organization with regard to risk

[BS ISO 22301:2012, **3.51**]

3.1.8 susceptibility

existence of a generic vulnerability in a particular implementation such that it could lead to a failure

3.1.9 system

combination of interacting elements organized to achieve one or more stated purposes

[ISO/IEC 15288]

3.1.10 tailoring

developing and applying trustworthy software concepts, principles and techniques to suit each specific environment

3.1.11 top management

person or group of people who directs and controls an organization at the highest level

[BS EN ISO 9000:2005, **3.2.7**]

3.1.12 trustworthy software constraint and dependency model (TSCDM)

document that explains all external constraints and dependencies involved in software deployment, configuration and operation

3.1.13 trustworthy software defect and deviation list (TSDDL)

document that encapsulates all defects and deviations that have been identified, their status, and any short term mitigations required to address deferrals until they can be resolved

3.1.14 trustworthy software framework (TSF)

domain- and implementation-agnostic reference to the existing body of knowledge, including functional safety, information security, and systems and software which provides a consensus collation of good practice for software trustworthiness

3.1.15 trustworthy software management system (TSMS)

document that explains the organization's approach to the implementation of trustworthy software concepts, principles and techniques as applicable to its participation in the specification, realization and/or use of software

3.1.16 trustworthy software release authority (TSRA)

competent person responsible for ensuring that consideration has been given to all relevant trustworthy software concepts, principles and techniques before release, including the TSCDM

3.1.17 trustworthy software release note (TSRN)

document that summarizes for those deploying, configuring and operating the software item the facets of trustworthiness that have been addressed during its specification and realization, the constraints and dependencies that apply, and any unmitigated defects and deviations, along with any short term mitigations required until they can be removed

3.1.18 trustworthy

appropriately addresses safety, reliability, availability, resilience and security issues

3.1.19 vulnerability

instantiation of a generic software weakness in a particular platform that could be exposed

3.1.20 weakness

generic classes of potential deficiency in software

3.2 Acronyms

For the purposes of this PAS, the following acronyms apply.

PACE	pragmatic, appropriate and cost effective
TL	trustworthiness level
TPM	trusted platform module
TRL	technology readiness level
TSI	Trustworthy Software Initiative

4 Approach

4.1 Applicability

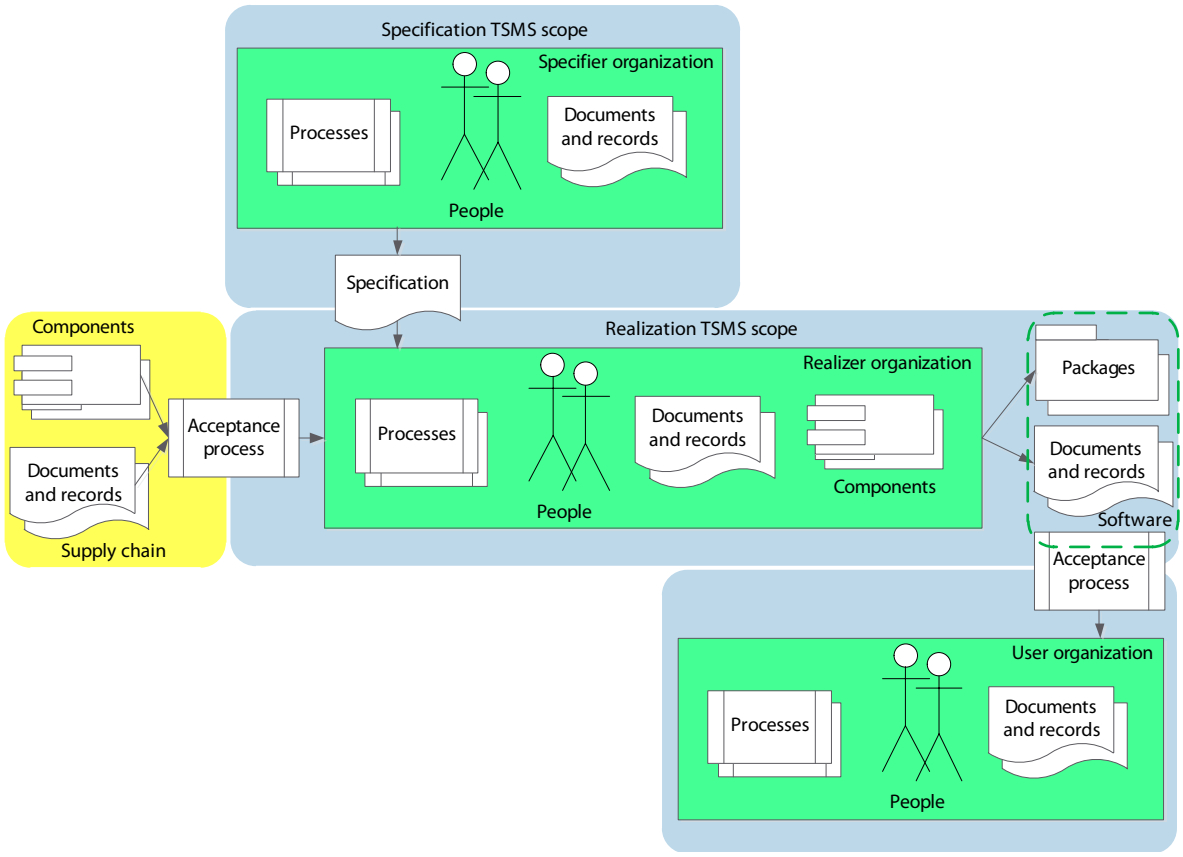
The trustworthy software framework (TSF) is designed to cover all aspects of the system and software life cycle, as defined by ISO/IEC 15288, but shall only be applied to the element(s) of the life cycle as relevant to the organization, which for the purposes of this PAS are grouped as illustrated in Figure 5:

- Specification – Collating the requirements for the explicit and implicit characteristics of software to be acquired;
- Realization – Producing the software (a multi-phase activity including design; implementation; integration; test);
- Use – Deploying, configuring, operating and/or using the software.

NOTE 1 *These groupings may be used iteratively where software is replaced or augmented.*

NOTE 2 *Any organization attempting trustworthy software practices should aim for continuous improvement of its application of concepts, principles and techniques. Its trustworthy software management system (TSMS) should include recognition of the need for continuous improvement using the PDCA (plan–do–check–act) cycle (see Figure 4) unless another approach is documented.*

Figure 5 – Use during life cycle



4.2 Categorization

Organizations shall review their drivers and requirements for trustworthiness, based on the following categories:

- Mass Market with Implicit Need (M/I) for software trustworthiness;
- Mass Market with Explicit Need (M/E) for software trustworthiness;
- Niche Markets with Explicit Need (N/E) for software trustworthiness.

4.3 Facets of trustworthiness

The TSF is designed on the basis that the same principles can be invoked as building blocks for trustworthy software across the following five facets of trustworthiness, and each item of software shall therefore be reviewed for both explicit, and credible implicit, requirements for delivery of these abilities:

- Safety – the ability of the system to operate without harmful states;
- Reliability – the ability of the system to deliver services as specified;
- Availability – the ability of the system to deliver services when requested;
- Resilience – the ability of the system to transform, renew, and recover in timely response to events;
- Security – the ability of the system to remain protected against accidental or deliberate attacks.

***NOTE** Explicit requirements only exist in the N/E and M/E use cases, which typically should also be supplemented by implicit, non-functional requirements (NFR) for trustworthiness. For the M/I use case, all trustworthiness requirement are inherently non-functional.*

4.4 Trustworthiness level assessment

For each item of software; having established which facet(s) of trustworthiness are required either explicitly or implicitly, a trustworthiness level (TL) shall be established, based on an assessment of the role software plays in the overall system or service to be delivered, and the maximum impact that a defect or deviation in such software would have on the system or service.

The software role segmentation shall be based upon the degree to which the source of trustworthiness in a component, composed sub-system or system is dependent on software:

- Paramount role – where software provides the sole source of trustworthiness in a component, composed sub-system or system;
- Explicit role – where software provides the main source of trustworthiness in a component, composed sub-system or system;
- Implicit role – where software provides a major source of trustworthiness in a component, composed sub-system or system;
- Ancillary role – where software only provides a minor source of trustworthiness in a component, composed sub-system or system.

Potential impact shall be assessed, for which a simple 4 level scale will typically suffice, which shall be based upon the organizational context:

- None;
- Routine;
- Significant;
- Critical.

These assessments shall be used as shown at Figure 6.

Figure 6 – Trustworthiness level matrix

Role	Impact			
	None	Routine	Significant	Critical
Paramount	N/A	TL3	TL4	TL4
Explicit	N/A	TL3	TL3	TL4
Implicit	N/A	TL2	TL3	TL3
Ancillary	TLO	TL1	TL2	TL3

The organization shall document the assessed TL, which shall be the basis for prioritizing efforts in software trustworthiness. This assessment shall be periodically reviewed throughout the software's life cycle.

NOTE 1 *An informative set of interpretations of the TL against BS ISO/IEC 15504 (SPICE) concepts would be:*

- *TL0 Software trustworthiness not required;*
- *TL1 Software trustworthiness delivered in a due diligence manner;*
- *TL2 Software trustworthiness delivered by managed processes;*
- *TL3 Software trustworthiness delivered by established processes;*
- *TL4 Software trustworthiness delivered by predictable or optimising processes.*

All assessments and treatments shall be applied in a pragmatic, appropriate and cost effective (PACE) manner, using trustworthy software concepts, principles and techniques to suit each specific environment.

NOTE 2 *For lower assessed TL requirements, rather than choosing and refining techniques from the descriptive set presented in the comprehensive TSF, the organization has the option of using a prescriptive baseline TSF subset which should provide pareto coverage of the most commonly encountered risks.*

4.5 Deployment

Any organization seeking to specify, realize or use software in a trustworthy manner shall deploy the relevant elements, as illustrated in Figure 7.

4.6 Fundamental control measures

Any organization seeking to specify, realize or use software in a trustworthy manner shall demonstrate the existence of the following fundamental control measures:

- Trustworthy software management system (TSMS);

  NOTE 1 *The TSMS should contain a level of detail appropriate to the scale of the organization and the TL requirements; for instance a small organization producing TL1 software may have a very simple summary document, whereas a large organization producing TL4 software necessitates very detailed and prescriptive TSMS. In either case the TSMS should draw out the principles laid down in this document and map them to the environment.*

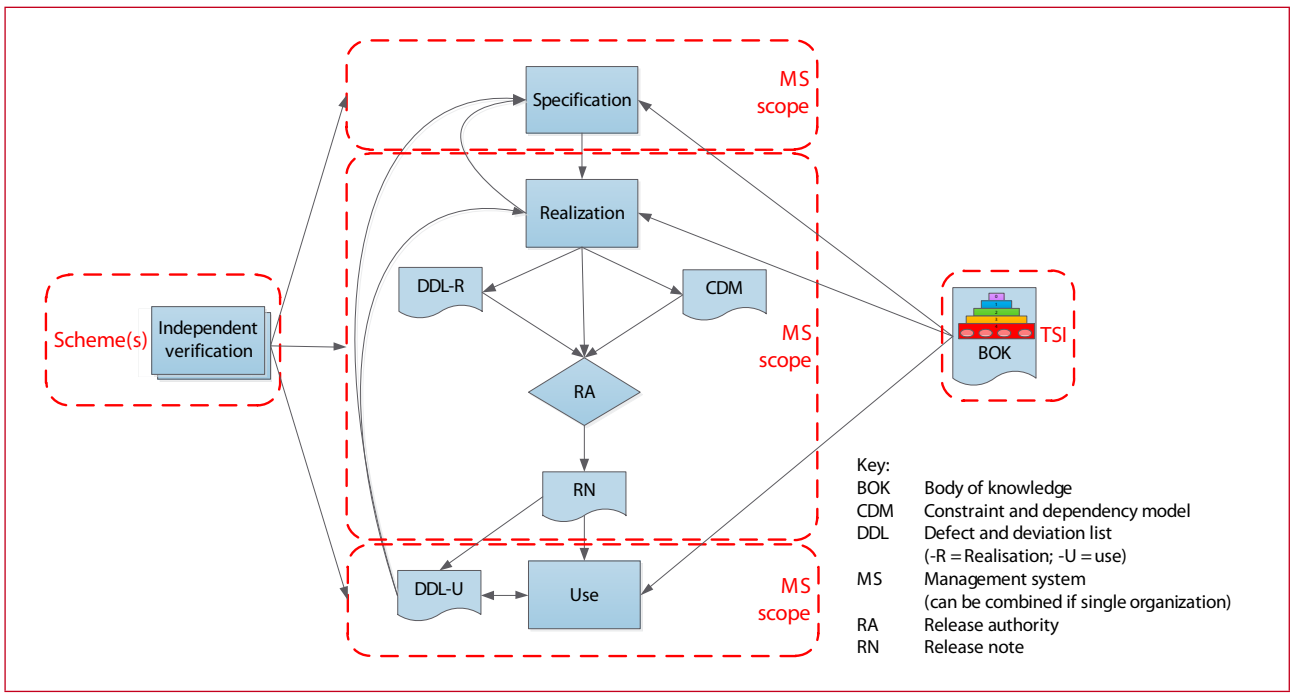
  NOTE 2 *The TSMS does not need to be a separate document, if it can be satisfactorily integrated with other management systems, in line with the approach taken, for instance, by TickITPlus.*

- Trustworthy software defect and deviation list (TSDDL).

  NOTE 3 *The TSDDL does not need to be a separate document, provided that its functions can be satisfactorily provided by, and traced through, other documents or records systems.*

  NOTE 4 *The names used for these measures can be tailored for the specific organizational context.*

Figure 7 – Deployment model



4.7 Realization control measures

Any organization engaged in the realization stage of software shall demonstrate the existence of the following additional realization control measures:

- Trustworthy software constraint and dependency model (TSCDM);

  NOTE 1 *The TSCDM does not need to be a separate document, provided that its functions can be satisfactorily provided by, and traced through, other documents or records systems.*

- Trustworthy software release authority (TSRA);

  NOTE 2 *The TSRA does not need to be a dedicated role, provided that its functions can be satisfactorily discharged by a suitably responsible and empowered member of the organization.*

- Trustworthy software release notice (TSRN).

  NOTE 3 *The TSRN does not need to be a separate document, provided that its functions can be satisfactorily provided by, and traced to, other documents.*

  NOTE 4 *The names used for these measures can be tailored for the specific organizational context.*

5 Concepts

5.1 Governance

To establish confidence in its software trustworthiness, the organization shall implement governance and management arrangements, where management is identified as consisting of risk, control and compliance.

NOTE *Arrangements should be appropriate for the size, type and the nature of the business.*

These governance and management arrangements shall include coverage of the control measures defined in **4.5**, and **4.6** for organizations realizing software.

These arrangements shall be aligned with the needs of stakeholders, such as customers, and the way they rely on the organization as part of an extended supply chain.

NOTE *The detailed processes or actions that an organization should follow in order to achieve software trustworthiness can only be defined by the organization once this environment is understood.*

5.2 Risk

The way in which software deliverables are likely to be used shall be reviewed to assess the needs for software trustworthiness, including scoping risks driven by external dependencies, such as customers, and the way in which these rely on the organization as part of an extended supply chain and risks from the extended supply chain.

Once this understanding has been obtained, a risk assessment process shall be performed for the software, relating to the set of assets to be protected, the nature of the adversities that might be faced and the way in which the software might be susceptible, in accordance with **4.4**.

5.3 Controls

To achieve software trustworthiness, risk management shall be applied, primarily by the treatment of risk through the application of controls. If elimination is not possible then toleration, transfer or termination shall be considered, with any decision taken explicitly recorded.

NOTE *Although the focus is on software, controls can be further subdivided as:*

- *Personnel – considering the people involved with making trustworthy software;*
- *Physical – protecting the trustworthy software artefacts and environments;*
- *Procedural –the processes used to specify, implement and realize trustworthy software;*
- *Technical – those achieved by software environment itself.*

5.4 Compliance

Having established a set of governance measures, understood the risk, and decided on an appropriate set of controls, a compliance regime shall be instituted by the developers and users of the software, to ensure that these decisions have been implemented and are maintained.

6 Principles

6.1 Applicability

The TSF shall only be applied to the element(s) of the system and software life cycle as relevant to the organization and intended use of software, as illustrated by Annex A.

***NOTE** For lower assessed TL requirements, rather than choosing and refining techniques from the descriptive set within this comprehensive TSF, the organization may wish to adopt a prescriptive baseline TSF subset which should provide pareto coverage of the most commonly encountered risks.*

6.2 Governance (GV)

6.2.1 General

Before producing or using any software which has a trustworthiness requirement, an appropriate set of governance and management measures shall be put in place, as defined in the TSMS.

6.2.2 GV.01 – Understand general environment

In order to establish the set of governance and management processes, the contextual background shall be understood, including the legal and regulatory environment, the technology to be employed, the culture within the organization and its customers and the nature of the extended supply chain.

6.2.3 GV.02 – Understand trust environment

To understand the need for software trustworthiness, the way in which the deliverables are likely to be used shall be reviewed and recorded in the TSCDM before implementation. This shall include external dependencies, such as customers, and the way in which these rely on the organization as part of an extended supply chain, in particular.

From this understanding, any special requirements (e.g. for assurance, privacy and cryptography) shall be derived.

6.2.4 GV.03 – Implement formal management regime

A formal TSMS shall be instituted, which shall include specific accountability and responsibility in top management for trustworthy software, along with the separation of functions, an explicit role of TSRA, and formal acceptance for third-party products and services.

6.3 Risk (RI)

***NOTE** The risk assessment process involves considering the set of assets to be protected, the nature of the adversities that may be faced, and the way in which the software may be susceptible to such adversities.*

6.3.1 RI.01 – Understand general risks

In order to assess the risk to trustworthy software, the utility[4)] of assets to be stored, processed or forwarded:

- shall be captured;
- the superset of external factors likely to have deleterious effects on software – being the aggregate of the set of hazards (undirected events) and threats (directed, deliberate, hostile acts) – shall be listed;
- the vulnerabilities of both bespoke and off-the-shelf elements shall be understood such that an analysis can be completed, factoring in proportionality of protection.

6.3.2 RI.02 – Understand trustworthiness risks

In addition to the general risk assessment (RI.01), an understanding of factors particular to the specific trustworthiness shall be obtained, to include factors such as maturity of technology, and current and emergent weaknesses, attack patterns, malware and vulnerabilities.

6.4 Controls (CO)

6.4.1 General

In the context of software trustworthiness, risk shall be managed through the treatment of risk by the application of controls, although if this is not possible then risk shall be managed through toleration, transfer or termination. Any such decision shall be explicitly taken and recorded.

[4)] Utility is the subjective importance of an asset.

6.4.2 Personnel (CO-PE)

NOTE This clause addresses those measures achieved by considering the people involved with making trustworthy software.

6.4.2.1 PE.01 – Maintain practitioner competence

The various practitioners involved in the specification, realization and operation of software shall be appropriately educated, trained and verified.

NOTE This should include continuing professional development (CPD), and the gaining of relevant experience, such that competence is achieved and maintained.

6.4.2.2 PE.02 – Maintain organizational competence

The organization, and its TSMS, shall be verified for appropriate competence being achieved and maintained.

6.4.2.3 PE.03 – Management of people risk

The organization shall have a single accountable owner of people risk, and ensure that people risks are comprehensively monitored.

6.4.3 Physical (CO-PH)

NOTE Those (controls) achieved by protecting the trustworthy software artefacts and environments.

6.4.3.1 PH.01 – Protect physical environment

The environments in which software is specified, realized and operated shall be appropriately protected, including the separation of development, test and operational facilities in accordance with ISO/IEC 27001.

6.4.3.2 PH.02 – Provide artefact protection

Artefacts (including source code and data) shall be protected from unauthorized access.

NOTE For TL3+ review need for electromagnetic protection of platforms and cryptographic protection of sensitive data.

6.4.4 Procedural (CO-PR)

NOTE This clause addresses those controls achieved by the processes by the way that trustworthy software is specified, implemented and realized.

6.4.4.1 PR.01 – Perform project management

The project by which software is specified, implemented and realized shall be planned, including capture of requirements, explicit product descriptions being generated and a process of peer review and validation implemented.

6.4.4.2 PR.02 – Perform supplier management

The supply chain for software shall be understood, so that trustworthiness can be specified and verified.

6.4.4.3 PR.03 – Understand requirements

The requirements for software shall be understood, including explicit (functional) requirements (FR); implicit (non-functional) requirements (NFR); implicit, non-objective requirements (NOR).

The use cases for software shall be understood, and any derived requirements (DR) arising during realization shall be recorded.

6.4.4.4 PR.04 – Maintain configuration management

All elements of software shall be subject to configuration management (CM), including specification, realization and release. This shall include producing and maintaining a TSCDM, and a product release and acceptance / commissioning process with a TSRN which shall be issued under the authority of the TSRA.

6.4.4.5 PR.05 – Confirmation of assurance

In order to achieve confirmation of trustworthy software characteristics, an assurance case shall be developed and maintained, which will form the basis for assurance and acceptance review by the TSRA and user(s).

6.4.4.6 PR.06 – Perform trusted software asset management

Processes shall be implemented to manage the software asset throughout its life.

NOTE This should include delivery, acceptance, asset recognition and review, and decommissioning.

6.4.4.7 PR.07 – Maintain defect management

All defects identified both during realization and in service shall be recorded in a TSDDL, reported and assessed, with rectification at earliest opportunity using formal process for monitoring of deferrals.

6.4.5 Technical (CO-TE)

NOTE This clause addresses those controls achieved by the software environment itself.

6.4.5.1 TE.01 – Follow architecture-driven implementation

All software design shall be based on an understanding of the architectural context, encompassing the properties of the system in its environment as embodied in its elements, and their inter-relationships, in accordance with ISO/IEC 42010.

NOTE For TL3+ consider architectural reference model (ARM), from which architectural reference case(s) are produced, allowing appropriate generic design and/or effect classes to be selected, and an architectural specification case(s) developed.

6.4.5.2 TE.02 – Make appropriate tool choices

Tools used throughout the realization cycle shall be appropriately selected, to include development environment(s), programming language(s) and associated coding standards and testing tools. These tools shall be configured such that their facilities that help enforce software trustworthiness are exploited.

NOTE Most programming tools, such as compilers, have options to check for weaknesses during code production, but these are frequently either ignored or switched off due to a perception that treatment of errors and warnings will slow down the realization process and require extra resources. Although this assumption may be true for the particular activity, many studies have indicated that whole-life time and resource expenditure is actually reduced by dealing with such errors and warnings at the first time they are encountered rather than retrospectively.

6.4.5.3 TE.03 – Follow systematic design

A system of formally developing and recording high-level design and low-level design information shall be followed, wherever possible using proven components (e.g. libraries), and with specific procedures for handling of third-party components (including open source software), and the realization shall be documented.

6.4.5.4 TE.04 – Follow structured implementation

Bespoke components shall be produced in accordance with good practice coding standards, and the realization shall be documented.

NOTE Including using relevant recognized data formats; algorithms; timing and synchronization approaches.

6.4.5.5 TE.05 – Seek trustworthy realization

When software is being designed, known failure and attack pattern modes shall be reviewed, with components implemented in accordance with desired design/effect pattern(s), factoring in layered mechanisms to provide defence-in-depth, and where necessary appropriate cryptographic key management process and anti-tamper measures.

NOTE Mitigations for all identified failure modes should be implemented, including, as appropriate:

- *isolation for untrusted components (e.g. sandboxing);*
- *isolation for high consequence code and data;*
- *malicious and mobile code control;*
- *control of network services and users, with appropriate access control mechanisms (authentication, authorization and mediation);*
- *control, of and access to, log / audit / accounting / trace facilities;*
- *provision of special controls (e.g. passwords, cryptography) at appropriate strength.*

6.4.5.6 TE.06 – Minimize risk exposure

To reduce the occasions when defects can arise or be exploited, only minimum privileges shall be used, with all other actions defaulting to "not permitted".

NOTE All program data, executables, and configuration data should be separated; entry / exit points and use of interfaces to environment resources should be minimized.

6.4.5.7 TE.07 – Practice hygienic coding

To reduce the degree to which defects can arise or be exploited, coding approaches shall be structured and aligned with coding standards.

NOTE 1 For example, including such items as:

- *all variables, pointers and references being properly initialized at first and subsequent uses;*
- *all input data, messages and output data being validated;*
- *implementations of all algorithms being validated;*
- *error handling being comprehensive and "fail safe and secure";*
- *a consistent naming convention being applied;*
- *resource access (e.g. buffers, stacks, variables, macros, memory, cache and files) being explicitly managed;*
- *detritus (e.g. temporary files / logs) being removed.*

NOTE 2 For TL3+ consider log/trace facilities, with ability to audit the data.

6.4.5.8 TE.08 – Use methodological production

In order to understand the delivery approach for a particular implementation, before commencing build definition; customized checklists; integration standards; dependencies and assumptions shall be produced and maintained.

During realization the organization shall enable and use compiler checking features; remove unused functions; configure components; perform unit testing before submitting components to integration.

6.4.5.9 TE.09 – Perform internal pre-release review

The internal integration and release function shall perform:

- QA testing;
- load / performance testing;
- regression testing;
- acceptance testing, culminating in the production of a TSRN covering dependencies, assumptions and deferrals.

NOTE *Other techniques including exploratory and fuzz testing might be considered.*

6.4.5.10 TE.10 – Perform internal verification

Verification of software shall include:

- code analysis, including malware detection;
- usability analysis, considering the possibility of human error or misuse.

NOTE *Other techniques might be considered, including:*

- *composition analysis;*
- *traceability analysis;*
- *fuzz testing.*

6.4.5.11 TE.11 – Enable dependable deployment

When being deployed, a chain of custody for components shall be maintained; the configuration shall be made consistent with requirements, including only minimum necessary privileges. Once in use, the software shall be monitored for anomalous behaviour, and a patching regime developed to allow for the application of routine, critical and emergency repairs.

NOTE *For TL3+ consider executing code analysis and heuristic/behavioural monitoring of implemented software.*

6.5 Compliance (CM)

6.5.1 General

A compliance regime shall be in place within the organization to ensure that governance, risk and control decisions have been implemented and is maintained.

NOTE *The compliance regime is necessary both within the organization producing the software, and those using the software.*

6.5.2 CM.01 – Perform acceptance verification

For acceptance from the supplier, verification of product/component shall take the form of weakness testing. A subsequent recurring compliance testing regime shall be implemented.

NOTE 1 *Weakness testing might include fault injection, and for acceptance into service might take the form of penetration testing, fault injection and robustness testing.*

NOTE 2 *For TL3+ consider independent specialist activity.*

6.5.3 CM.02 – Maintain ongoing review

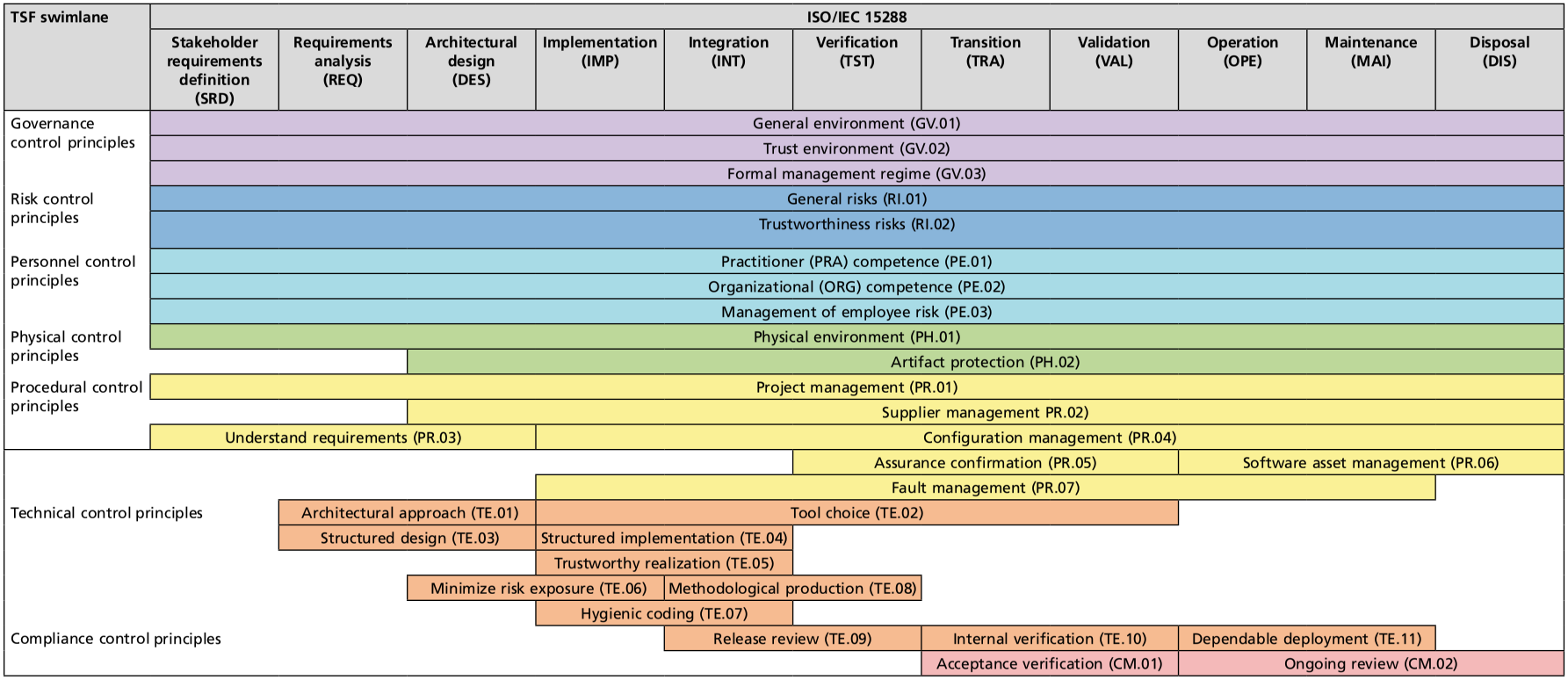
Once the software is deployed, management processes shall be regularly reviewed to ensure they are still relevant. Operational risk reviews shall include checking progress against TSDDL deferrals, with internal audit processes including reviews of any software issues encountered and metrics including efficacy of remediation of TSDDL deferrals.

NOTE *Indicative frequency of reviews:*

- *TL1 Yearly;*
- *TL2 Six monthly;*
- *TL3 Quarterly;*
- *TL4 Monthly.*

Annex A (informative)
PAS 754 in the system life cycle

Figure A.1 – PAS 754 in the system life cycle



Totals	Level 1	Concepts	4
	Level 1.1	Swimlanes	7
	Level 2	Principles	30
	Level 3	Techniques	150
	Level 4	(Repository)	Unlimited

Annex B (informative)
Techniques for delivery of PAS 754 requirements

Table B.1 gives an illustration of techniques that can be used to deliver the requirements described in this PAS.

The table is provided to facilitate an integrated approach to implementation of software trustworthiness. The table can also aid the inclusion of software trustworthiness into an existing management system.

Table B.1 – Techniques for delivery of PAS 754 requirements

Control area (concepts)	Control serial (UDEF)	Control group (principles)	Control summary (techniques)	Control detail (techniques)
Governance (Gv)	GV.01.10	Understand general environment	Legal environment	Understand legal environment(s)
	GV.01.20		Regulatory environment	Understand regulatory environment(s)
	GV.01.30		Technical environment	Understand technical environment(s), such as cloud, including consideration of technology readiness levels (TRL)
	GV.01.40		Organizational culture	Understand organizational culture(s)
	GV.01.50		Supply chain	Understand whole supply chain
	GV.02.10	Understand trust environment	Special considerations for assurance	Understand special considerations for assurance
	GV.02.20		Special considerations for privacy	Understand special considerations for privacy
	GV.02.20		Special considerations for cryptography	Understand special considerations for cryptography

Table B.1 – Techniques for delivery of PAS 754 requirements *(continued)*

Control area (concepts)	Control serial (UDEF)	Control group (principles)	Control summary (techniques)	Control detail (techniques)
Governance (Gv)	GV.03.10	Implement formal management regime	Trustworthy software management system (TSMS)	Implement trustworthy software management system (TSMS)
	GV.03.20		Top management responsibility	Ensure top management accountability and responsibility exists for trustworthy software
	GV.03.30		Separation of functions	Implement separation of functions, at minimum between design/development, integration/test and operational functions
	GV.03.40		Trustworthy software release authority (TSRA)	Implement the role of trustworthy software release authority (TSRA)
	GV.03.50		Input formal acceptance	Implement a formal acceptance function for 3rd party products and services
Risk (Ri)	RI.01.10	Understand general risks	Information asset utility	Understand utility of information asset to be stored, processed or forwarded
	RI.01.20		Threat profiles	Understand threat profile of probable user
	RI.01.30		Adversity analysis	Perform adversity (hazard + threat) analysis
	RI.01.40		Vulnerability analysis	Perform vulnerability analysis of both bespoke and off the shelf elements
	RI.01.50		Risk analysis	Perform overall risk analysis, factoring in proportionality of protection
	RI.01.60		Define controls	Define controls required to mitigate identified risks
	RI.02.10	Understand trustworthiness risks	Technology maturity	Maintaining understanding of maturity of technology
	RI.02.20		Weaknesses	Maintaining understanding of current and emergent weaknesses
	RI.02.30		Attack patterns and malware	Maintaining understanding of current and emergent attack patterns and malware
	RI.02.40		Vulnerabilities	Maintaining understanding of current and emergent vulnerabilities

Table B.1 – Techniques for delivery of PAS 754 requirements *(continued)*

Control area (concepts)	Control serial (UDEF)	Control group (principles)	Control summary (techniques)	Control detail (techniques)
Controls – Personnel (Cn-Pe)	PE.01.10	Maintain practitioner (PRA) competence	Education	Practitioner education
	PE.01.20		Training	Practitioner training
	PE.01.30		Verification	Practitioner verification (VER)
	PE.01.40		Mentoring	Practitioner mentoring
	PE.01.50		Continuing professional development (CPD)	Practitioner continuing professional development (CPD)
	PE.02.10	Maintain organizational competence	Organizational awareness	Introduce and maintain organizational awareness
	PE.02.20		Organizational verification	Verification (VER) of organizational competence
	PE.03.10	Management of people risk	People risk owner	Single accountable owner of people risk
	PE.03.20		Personnel monitoring	Holistic personnel risk monitoring
Controls – Physical (Cn-Ph)	PH.01.10	Protect physical environment	Separation of facilities	Implement separation of development, test and operational facilities
	PH.02.10	Provide artefact protection	Source code protection	Implement protection of source code, covering confidentiality, integrity and availability (CIA)
	PH.02.20		Data protection	Implement protection of data
	PH.02.30		Electromagnetic protection	Implement appropriate electromagnetic protection of platforms
	PH.02.40		Cryptographic protection	Implement appropriate cryptographic protection of sensitive data
Controls – Procedural (Cn-Pr)	PR.01.10	Perform project management	Project plan	Produce and maintain a project plan
	PR.01.20		Product descriptions	Produce and maintain product descriptions
	PR.01.30		Specification validation	Peer review and validate specification as meeting user requirement
	PR.01.40		Realization validation	Peer review and validate realization as meeting system requirement

Table B.1 – Techniques for delivery of PAS 754 requirements *(continued)*

Control area (concepts)	Control serial (UDEF)	Control group (principles)	Control summary (techniques)	Control detail (techniques)
Controls – Procedural (Cn-Pr)	PR.02.10	Perform supplier management	Supply chain identification	Identify supply chain
	PR.02.20		Supply chain requirements	Establish supply chain quality, security and integrity requirements
	PR.02.30		Supply chain assurance	Establish supply chain quality, security and integrity assurance
	PR.02.40		Supplier verification	Supplier independent verification
	PR.03.10	Understand requirements	Functional requirements	Specify explicit / functional requirements (FR)
	PR.03.20		Non-functional requirements	Specify implicit / non-functional requirements (NFR)
	PR.03.30		Non-objective requirements	Understand implicit / non-objective requirements (NOR)
	PR.03.40		Use cases	Understand use cases
	PR.03.50		Derived requirements	Monitor and record derived requirements (DR)
	PR.04.10	Maintain configuration management	Specification configuration management (CM)	Implement and maintain configuration management (CM) of specification
	PR.04.20		Realization configuration management (CM)	Implement and maintain configuration management (CM) of realization, including artefact integrity and version control
	PR.04.30		Separate integration testing	Implement integration testing separately from development personnel
	PR.04.40		Product release	Formal process for product release
	PR.04.50		Acceptance review	Establish acceptance / commissioning review process
	PR.04.60		Trustworthy software constraint and dependency model (TSCDM)	Implement and maintain a trustworthy software constraint and dependency model (TSCDM)
	PR.04.70		Installation configuration management (CM)	Implement and maintain configuration management (CM) for installed software

Table B.1 – Techniques for delivery of PAS 754 requirements *(continued)*

Control area (concepts)	Control serial (UDEF)	Control group (principles)	Control summary (techniques)	Control detail (techniques)
Controls – Procedural (Cn-Pr)	PR.05.10	Confirmation of assurance	Assurance case	Produce and maintain an assurance case
	PR.05.20		Assurance review	Establish a process for assurance and acceptance review
	PR.06.10	Perform trusted software asset management	Trusted delivery	Implement trusted means of software delivery
	PR.06.20		Output formal acceptance	Implement an acceptance process for products and services
	PR.06.30		Software asset recognition	Implement trusted means of software asset recognition
	PR.06.40		Software assets review	Software assets should be reviewed regularly
	PR.06.50		Decommissioning definition	Define any special requirements for Decommissioning
	PR.07.10	Maintain defect management	Realization defect management	Ensure all defects identified during realization are recorded, reported and assessed, with rectification at earliest opportunity using process for monitoring through a realization trustworthy software defect and deviation list (R-TSDDL)
	PR.07.20		In-service trustworthy software defect and deviation list (I-TSDDL)	Ensure all in service defects and deviations are recorded in an in-service trustworthy software defect and deviation list (I-TSDDL), reported and assessed, with rectification at earliest opportunity using process for monitoring of deferrals
Controls – Technical (Cn-Te)	TE.01.10	Follow architecture-driven implementation	Architectural reference model (ARM)	Develop architectural reference model (ARM)
	TE.01.20		Architectural reference case(s)	Use architectural reference model to produce architectural reference case(s)
	TE.01.30		Design / effect class selection	Select appropriate generic design / effect classes for reference case
	TE.01.40		Architectural specification case(s)	Use architectural reference case(s) to produce architectural specification case(s)

Table B.1 – Techniques for delivery of PAS 754 requirements *(continued)*

Control area (concepts)	Control serial (UDEF)	Control group (principles)	Control summary (techniques)	Control detail (techniques)
Controls – Technical (Cn-Te)	TE.02.10	Make appropriate tool choices	Appropriate programming language(s)	Selection of appropriate programming language(s), considering known vulnerabilities and needs for typing
	TE.02.20		Appropriate coding standards	Produce and maintain coding standards
	TE.02.30		Appropriate proving techniques	Selection of appropriate tools for composing, tracing and proving specification, design and implementation
	TE.02.40		Appropriate testing tools	Selection of appropriate testing tools
	TE.03.10	Follow structured design	High level design	Map specification to high level design, with appropriate support for traceability
	TE.03.20		Low level design	Map high level design to low level design, with appropriate support for traceability
	TE.03.30		Proven components	When re-using components and libraries, these should have a trustworthy software provenance wherever possible
	TE.03.40		Open source handling	Procedures for handling of open source components and libraries
	TE.04.10	Follow structured implementation	Use coding standards	Produce bespoke components in accordance with coding standards
	TE.04.20		Use data formats	Use appropriate and recognized data formats
	TE.04.30		Select algorithms	Select appropriate algorithms
	TE.04.40		Appropriate timing and synchronization	Select appropriate timing and synchronization approach
	TE.04.50		Telemetry measures	Consider instrumentation and telemetry measures

Table B.1 – Techniques for delivery of PAS 754 requirements *(continued)*

Control area (concepts)	Control serial (UDEF)	Control group (principles)	Control summary (techniques)	Control detail (techniques)
Controls – Technical (Cn-Te)	TE.05.05	Seek trustworthy realization	Design / effect patterns selection	Select appropriate generic design / effect patterns for specification case
	TE.05.10		Review failure modes	Review design for failure modes
	TE.05.15		Review attack patterns modes	Review design for attack patterns modes
	TE.05.20		Configure off the shelf components	Source and configure off the shelf components in accordance with design/ effect pattern(s)
	TE.05.25		Implement design / effect class / patterns	Implement relevant design / effect class / patterns
	TE.05.30		Apply layered mechanisms	Apply layered mechanisms to provide defence-in-depth
	TE.05.35		Select key management	Select appropriate cryptographic key management process
	TE.05.40		Tamper resistance	Consider tamper resistance and detection measures, including TPM
	TE.05.45		Mitigate identified failure modes	Ensure mitigations are used for all identified failure modes
	TE.05.50		Sandboxing	Implement sandboxing where possible
	TE.05.55		Isolation	Implement isolation for high consequence code and data
	TE.05.60		Control of derogation	Continuity of protection and control of derogation
	TE.05.65		Control malicious code	Implement measures to control malicious code
	TE.05.70		Control mobile code	Implement measures to control mobile code
	TE.05.75		Control network services	Implement measures to control network services
	TE.05.80		Control users	Implement only the minimum set of users to meet requirements
	TE.05.85		Access control mechanisms	Implement appropriate access control mechanisms (authentication, authorization and mediation)
	TE.05.90		Special controls	Implement special controls (e.g. passwords, cryptography) at appropriate strength
	TE.05.95		Log / trace access controls	Control access to log / trace facilities

Table B.1 – Techniques for delivery of PAS 754 requirements *(continued)*

Control area (concepts)	Control serial (UDEF)	Control group (principles)	Control summary (techniques)	Control detail (techniques)
Controls – Technical (Cn-Te)	TE.06.10	Minimize risk exposure	Minimum privilege	Only grant minimum privileges required, with all other actions defaulting to not permitted
	TE.06.20		Separate program elements	Separate program data, executables, and configuration data
	TE.06.30		Minimize flow change	Minimize entry and exit points, and changes of control flow
	TE.06.40		Minimize external dependencies	Control use of interfaces to environmental and external resources, making least prescriptive assumptions (e.g browser agnostic)
	TE.07.05	Practice hygienic coding	Initialize data structures	Ensure all variables, pointers and references are properly initialized at first and subsequent uses
	TE.07.10		Validate input data	Ensure all input data is validated
	TE.07.15		Validate messages	Ensure all messages are validated
	TE.07.20		Validate algorithms	Ensure implementations of all algorithms are validated
	TE.07.25		Validate output data	Ensure all output data is validated
	TE.07.30		Error handling	Ensure error handling is implemented comprehensively, and "fails safe and secure"
	TE.07.35		Naming convention	Apply consistent naming convention
	TE.07.40		Manage resources	Manage resource access explicitly (buffers, stacks, memory, cache and files)
	TE.07.45		Sequencing of events	Allow for variable sequencing of events
	TE.07.50		Remove detritus	Explicitly remove detritus (temporary files / logs)
	TE.07.55		Auditable log / trace	Implement log / trace facilities, with appropriate ability to audit the data

Table B.1 – Techniques for delivery of PAS 754 requirements *(continued)*

Control area (concepts)	Control serial (UDEF)	Control group (principles)	Control summary (techniques)	Control detail (techniques)
Controls – Technical (Cn-Te)	TE.08.10	Use methodological production	Build definition	Produce and maintain build definition
	TE.08.20		Checklists	Produce and maintain customized checklists of trustworthy software techniques required to be followed for particular implementation
	TE.08.30		Compiler checking features	Enable and use compiler checking features
	TE.08.40		Integration standards	Produce and maintain Integration standards
	TE.08.50		Configure components	Integrate and configure components
	TE.08.60		Dependencies and assumptions	Document dependencies and assumptions
	TE.08.70		Unused functions	Removed unused functions
	TE.08.80		Unit test	Perform unit test before submitting components to integration
	TE.09.10	Perform internal pre-release review (VER)	QA test	Perform QA test
	TE.09.20		Load / performance testing	Perform load / performance testing
	TE.09.30		Perform regression testing	Perform regression testing
	TE.09.40		Acceptance testing	Perform system acceptance testing
	TE.09.50		Trustworthy software release notice (TSRN)	Issue formal trustworthy software release notice (TSRN)
	TE.10.10	Perform internal verification (VER)	Verify build integrity	Verify code only includes elements planned, scanning for malware and other sources of taint
	TE.10.20		Composition analysis	Perform and maintain composition analysis, with appropriate use of refinement tools
	TE.10.30		Traceability analysis	Perform and maintain traceability analysis, with appropriate use of theorem proving tools
	TE.10.40		Code analysis	Perform code analysis
	TE.10.50		Usability analysis	Perform usability analysis, factoring in human fallibility
	TE.10.60		Fuzz testing	Perform fuzz testing

Table B.1 – Techniques for delivery of PAS 754 requirements *(continued)*

Control area (concepts)	Control serial (UDEF)	Control group (principles)	Control summary (techniques)	Control detail (techniques)
Controls – Technical (Cn-Te)	TE.11.10	Enable dependable deployment	Source code persistence	Ensure source code can be obtained throughout life cycle, such as by escrow service
	TE.11.20		Chain of custody	Ensure chain of custody for components being deployed
	TE.11.30		Configure privileges	Configure implemented software to meet requirements, including minimum necessary privileges
	TE.11.40		Executing code analysis	Consider executing code analysis
	TE.11.50		Behavioural monitoring	Consider heuristic / behavioural monitoring of implemented software
	TE.11.60		Continual remediation	Updating and patching of implemented software, with routine, critical and emergency options, taking due cognisance of R-TSDL / I-TSDDL
	TE.11.70		Continual vigilance	Monitor for anomalies and trends throughout life cycle, including intrusions
Compliance (Cm)	CM.01.10	Perform acceptance verification (VER)	Product / component acceptance verification	Product / component independent verification – "weakness testing" including fault injection
	CM.01.20		System acceptance verification	System independent verification – "penetration testing" including fault injection
	CM.02.10	Maintain ongoing review	Review management processes	Ensure management processes are regularly reviewed
	CM.02.20		Operational risk reviews	Perform operational risk reviews, including progress against TSDDL deferrals
	CM.02.30		Internal audit	Ensure internal audit processes include consideration of software issues
	CM.02.40		Compliance testing	Maintain recurring compliance testing
	CM.02.50		Maintain metrics	Maintain metrics, including progress against TSDDL deferrals

Bibliography

Standards publications

For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

BS EN ISO 9000:2005, *Quality management systems – Fundamentals and vocabulary*

BS ISO 22301:2012, *Societal security – Business continuity management systems – Requirements*

BS ISO/IEC 15504, *Information technology – Process assessment*

BS ISO/IEC 15408, *Information technology – Security techniques – Evaluation criteria for IT security*

BS EN 61508, *Functional safety of electrical/electronic/programmable electronic safety-related systems*

Other publications

[1] WORLD ECONOMIC FORUM. *Risk and Responsibility in a Hyperconnected World.* Geneva: WEF, 2014.

Further reading

BS EN ISO 9001, *Quality management systems – Requirements*

BS EN ISO/IEC 17024, *Conformity assessment – General requirements for bodies operating certification of persons*

BS EN ISO/IEC 17025, *General requirements for the competence of testing and calibration laboratories*

BS EN ISO/IEC 17043, *Conformity assessment – General requirements for proficiency testing*

BS ISO 31000, *Risk management – Principles and guidelines*

BS ISO/IEC 20000-1, *Information technology – Service management – Part 1: Service management system requirements*

BS ISO/IEC 27000, *Information technology – Security techniques – Information security management systems – Overview and vocabulary*

BS ISO/IEC 27034-1, *Information technology – Security techniques – Application security – Part 1: Overview and concepts*

CENTRE FOR THE PROTECTION OF NATIONAL INFRASTRUCTURE (CPNI). *Holistic Management of Employee Risk (HoMER).* CPNI Guide 2012-021. London:CPNI, 2012.

NORTH ATLANTIC TREATY ORGANIZATION (NATO). *Software engineering – Report on a conference sponsored by the NATO Science Committee.* Garmisch, Germany: NATO, 7th to 11th October 1968.

SHEWART, W.A. *Statistical Method from the Viewpoint of Quality Control,* 1939

SOMMERVILLE, I. *Software Engineering: International Version*. London: Pearson, 2010.

THE OPEN GROUP. *Open Trusted Technology Provider Standard (O-TTPS) Version 1.0 – Mitigating Maliciously Tainted and and Counterfeit Products,* 2013.

British Standards Institution (BSI)

BSI is the independent national body responsible for preparing British Standards and other standards-related publications, information and services. It presents the UK view on standards in Europe and at the international level.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

Revisions

British Standards and PASs are periodically updated by amendment or revision. Users of British Standards and PASs should make sure that they possess the latest amendments or editions.

It is the constant aim of BSI to improve the quality of our products and services. We would be grateful if anyone finding an inaccuracy or ambiguity while using British Standards would inform the Secretary of the technical committee responsible, the identity of which can be found on the inside front cover. Similarly for PASs, please notify BSI Customer Services.

Tel: +44 (0)845 086 9001

BSI offers BSI Subscribing Members an individual updating service called PLUS which ensures that subscribers automatically receive the latest editions of British Standards and PASs.

Tel: +44 (0)845 086 9001
Email: plus@bsigroup.com

Buying standards

You may buy PDF and hard copy versions of standards directly using a credit card from the BSI Shop on the website **www.bsigroup.com/shop**. In addition all orders for BSI, international and foreign standards publications can be addressed to BSI Customer Services.

Tel: +44 (0)845 086 9001
Email: orders@bsigroup.com

In response to orders for international standards, BSI will supply the British Standard implementation of the relevant international standard, unless otherwise requested.

Information on standards

BSI provides a wide range of information on national, European and international standards through its Knowledge Centre.

Tel: +44 (0)20 8996 7004
Email: knowledgecentre@bsigroup.com

BSI Subscribing Members are kept up to date with standards developments and receive substantial discounts on the purchase price of standards. For details of these and other benefits contact Membership Administration.

Tel: +44 (0)845 086 9001
Email: membership@bsigroup.com

Information regarding online access to British Standards and PASs via British Standards Online can be found at **http://shop.bsigroup.com/bsol**

Further information about British Standards is available on the BSI website at **www.bsigroup.com/standards**

Copyright

BSI, 389 Chiswick High Road
London W4 4AL
United Kingdom
www.bsigroup.com

ISBN 978-0-580-83242-0