INTERNATIONAL MARITIME ORGANIZATION

E

4 ALBERT EMBANKMENT
LONDON SE1 7SR
Telephone: +44 (0)20 7735 7611 Fax: +44 (0)20 7587 3210

MSC.1/Circ.1512
13 July 2015

GUIDELINE ON SOFTWARE QUALITY ASSURANCE AND HUMAN-CENTRED DESIGN FOR E-NAVIGATION

1 The Sub-Committee on Navigation, Communications and Search and Rescue (NCSR), at its second session (9 to 13 March 2015), agreed on the *Guideline on Software Quality Assurance and Human-Centred Design for e-navigation*.

2 The Maritime Safety Committee, at its ninety-fifth session (3 to 12 June 2015), having considered the proposal by NCSR 2, approved the *Guideline on Software Quality Assurance and Human-Centred Design for e-navigation*, as set out in the annex.

3 The guideline is intended to ensure that software trustworthiness and user needs are met through the application of Software Quality Assurance (SQA) and Human-Centred Design (HCD) in the development of e-navigation systems.

4 The guideline is also intended to support the principles identified in SOLAS regulation V/15 (Principles relating to bridge design, design and arrangement of navigational systems and equipment, and bridge procedures).

5 Member Governments are invited to bring this Guideline to the attention of all parties concerned.

***

ANNEX

GUIDELINE ON SOFTWARE QUALITY ASSURANCE AND HUMAN CENTRED-DESIGN FOR E-NAVIGATION

1 Introduction

1.1 Navigation systems increasingly provide a variety of information and services for enhancing navigation safety and efficiency. These systems require the connection and integration of onboard navigational systems as well as shore-side support systems and involve the collection, integration, exchange, presentation and analysis of marine data and information.

1.2 The merits of navigation systems can be found not only in their range of functions but also underpinned by their trustworthy software and overall usability. This guideline is intended to complement and support the principal requirements specified under SOLAS regulation V/15.

1.3 Achieving trustworthy software and usability in the development of complex systems requires a disciplined and structured approach. This guideline encourages such an approach in the development and management of e-navigation systems, with particular focus on Software Quality Assurance (SQA) and Human-Centred Design (HCD) that includes Usability Testing (UT). Systems so designed, developed and managed throughout their life cycle deliver improved user performance, being stable and resilient, and, most importantly, support users in low and high workload environments, such as during challenging navigation and environmental conditions when users are most vulnerable to making mistakes and when error management and recovery is essential. Other important benefits include limiting the amount of operator familiarization training that is needed and the time and resources required for system maintenance and support.

1.4 SQA focuses on defining and testing software quality and how that helps meet user requirements to ensure that high quality, robust, testable and stable software is used in e-navigation systems. E-navigation software quality needs to be evaluated to ensure relevant quality characteristics meet the requirements of the system.

1.5 The basic premise of HCD is that systems are designed to suit the characteristics of intended users and the tasks they perform, rather than requiring users to adapt to a system. UT is a key component of HCD and uses methods that rely on including users to test the ability of systems to support user needs. UT helps to identify potential problems and solutions during design and development stages by using an iterative approach to testing where the design evolves through rounds of prototyping, testing, analysing, refining and testing again.

1.6 The combination of SQA and HCD (including UT) provides opportunities to guide system design and development to improve data quality and information analysis, and to generally meet user needs and enhance safety.

1.7 This guideline is not intended to be the sole source of guidance for SQA and HCD and associated activities. Rather, it is intended to provide a general understanding of SQA and HCD for the effective design and development of e-navigation systems. It draws extensively on existing relevant international standards. Appendix 1 provides a list of recommended international standards used to support this guideline.

1.8 For any ISO/IEC standards referred to in this guideline, the current edition (including any amendments) applies, taking into account implementation periods, as applicable.

2 Scope

2.1 The scope of this guideline is to provide an overarching document to ensure that e-navigation quality design attributes are included in the development of e-navigation systems. Figure 1 provides an overview of the quality design attributes that should be considered and includes "product and data quality", "meet user needs", "security" and "functional safety". This guideline mainly addresses software quality, which incorporates "product and data quality" and "meet user needs". Consideration of all the design attributes will help ensure that software and human-based risks are addressed. Figure 1 also provides information on relevant standards that developers and designers of e-navigation systems should consider in ensuring all quality attributes are addressed ensuring overall system quality.

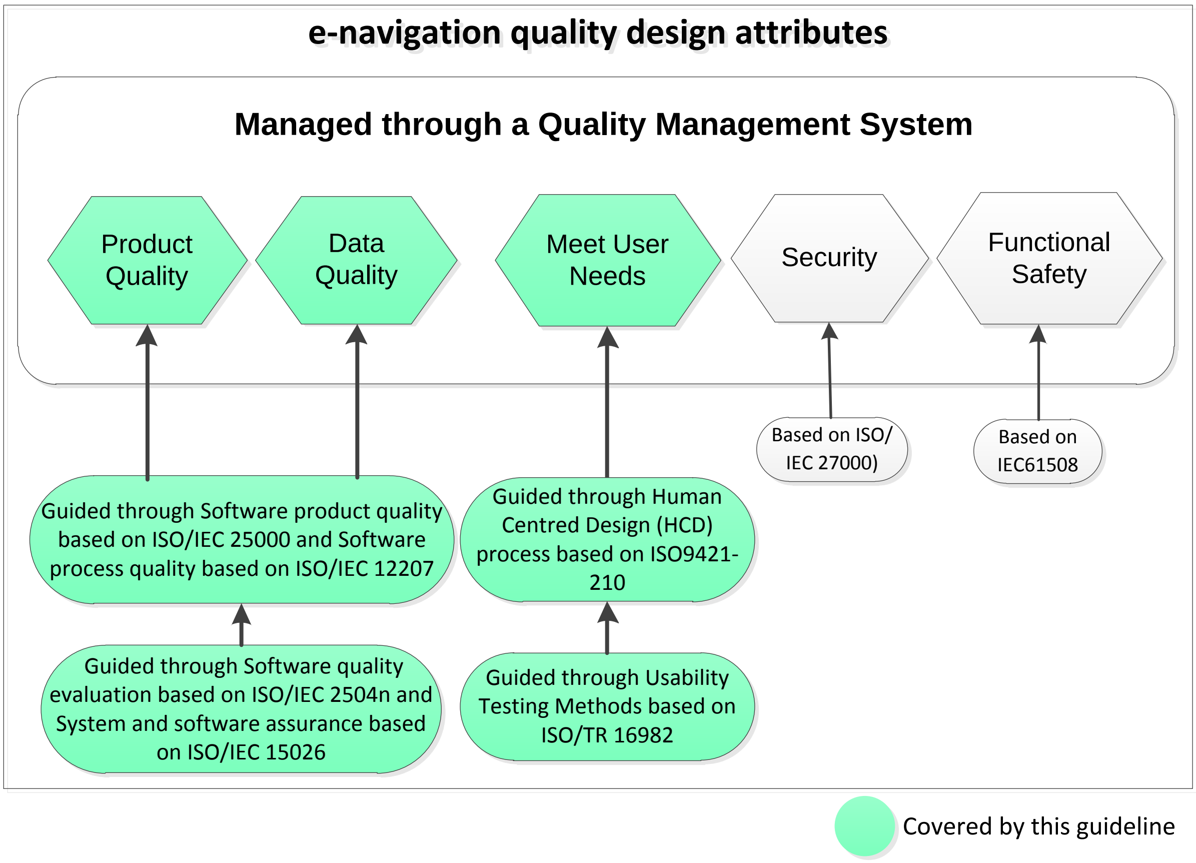

e-navigation quality design attributes
Managed through a Quality Management System
Product Quality
Data Quality
Meet User Needs
Security
Functional Safety
Based on ISO/ IEC 27000)
Based on IEC61508
Guided through Software product quality based on ISO/IEC 25000 and Software process quality based on ISO/IEC 12207
Guided through Human Centred Design (HCD) process based on ISO9421-210
Guided through Software quality evaluation based on ISO/IEC 2504n and System and software assurance based on ISO/IEC 15026
Guided through Usability Testing Methods based on ISO/TR 16982
Covered by this guideline

Figure 1: Concepts and standards for e-navigation quality design attributes

2.2 This guideline is intended to be used by all stakeholders involved in the design and development of e-navigation systems, with its primary users being those who develop and test e-navigation systems. Stakeholders include equipment designers and manufacturers, system integrators, maritime authorities and regulators, shipbuilders, shipowners, ship operators, Vessel Traffic Service authorities and Rescue Coordination Centres, and other relevant international organizations such as the International Association of Marine Aids to Navigation and Lighthouse Authorities (IALA) and the International Hydrographic Organization (IHO).

2.3 Table 1 provides a summary of stakeholder involvement in the application of this guideline at each stage of the e-navigation system's life cycle.

Table 1: Stakeholder involvement

Life cycle Stage	**Stakeholder**
Analysis Operational System Feedback	Manufacturers/system designers, users, shipowners, ship operators, regulatory authority
Stage 1: Concept development	Manufacturers/system designers, users
Stage 2: Planning and Analysis	Manufacturers/system designers, users
Stage 3: Design	Manufacturers/system designers, users
Stage 4: Integration and Testing	Manufacturers/system designers, users, approval authority (regulator), shipowners, ship operators
Stage 5: Operational	Users, shipowners, ship operators and manufacturers/system designers
Disposal	Shipowners, ship operators and manufacturers/system designers

2.4 The provisions in this guideline are goal-based and are not intended to specify or discourage the use of any particular quality assurance, management process, or testing method. Hence, detailed and prescriptive design requirements, which specify design solutions, are not covered.

2.5 It is recommended that users of this guideline be generally familiar with contemporary quality management processes, software quality assurance and human factors.

2.6 This guideline does not address training requirements.

3 Definitions

3.1 **Data quality:** The degree to which quality characteristics of data have the intrinsic potential to satisfy stated and implied needs.

3.2 **Data Quality Assurance (DQA)**: A set of processes, that ensures that shore and shipboard based data used by e-navigation systems meets and complies with required quality specifications.

3.3 **Effectiveness:** Measure of accuracy and completeness with which users achieve specified goals.

3.4 **Efficiency:** Resources expended in relation to the accuracy and completeness with which users achieve goals.

3.5 **E-navigation:** The harmonized collection, integration, exchange, presentation and analysis of marine information on board and ashore by electronic means to enhance berth-to-berth navigation and related services for safety and security at sea and protection of the marine environment.

3.6 **Human factors:** The scientific discipline concerned with the application of validated scientific research about people, their abilities, characteristics and limitations to the design of systems they use, environments in which they function and interact, and jobs they perform to optimize human well-being and overall system performance.

3.7 **Human-Centred Design (HCD)[1]:** An approach to system design and development that aims to make interactive systems more usable by focussing on the use of the system; applying human factors, ergonomics and usability knowledge and techniques.

3.8 **Product quality:** The degree to which a product or system meets functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability and portability as defined by ISO/IEC 25010 or relevant standards. The overall product quality is a result of quality of hardware, software and data.

3.9 **Satisfaction:** Freedom from discomfort along with positive attitudes towards the use of the system.

3.10 **Socio-technical system:** A system that includes interaction between people, technology (i.e. equipment and systems) and their physical and organizational environments.

3.11 **Software quality:** The degree to which a software product (system, component or process) meets specified requirements with the aim of also meeting stakeholder expectations.

3.12 **Software Quality Assurance (SQA):** A set of processes that ensures software meets and complies with required quality specifications. Designated SQA processes align with a system design life cycle.

3.13 **Software quality evaluation:** A systematic examination of the extent to which a software product is capable of satisfying stated and implied needs.

3.14 **Software quality in use:** Capability of a software product to enable specific users to achieve specific goals with effectiveness, productivity, safety and satisfaction in specific contexts of use.

3.15 **Stakeholder:** An individual or organization having a right, share, claim or interest in a system.

3.16 **System:** Combination of interacting elements organized to achieve one or more stated purposes. A system can consist of products (tools used to achieve a specific task), equipment, services and/or people.

3.17 **System life cycle (Life cycle):** The stages containing the processes activities and tasks spanning the life of the system and/or product from the definition of its requirements to the termination of its use; life cycle covers its conception, design, operation, maintenance, support and disposal.

3.18 **Usability:** The extent to which systems can be used by users to achieve specified goals with effectiveness, efficiency and satisfaction, in a specified context of use.

3.19 **Usability Testing (UT):** Evaluation methods and techniques used to support Human-Centred Design (HCD) and used for the purpose of increasing the usability of a system.

3.20 **User:** Anyone interacting with the system, including its operators and maintainers.

[1] The term "human-centred design" is used rather than "user-centred design" in order to emphasize that this process also addresses impacts on a number of stakeholders, not just those typically considered as users. However, in practice, these terms are often used synonymously.

Usable systems can provide a number of benefits including improved productivity, reduction in training needs, enhanced user well-being, avoidance of stress, increased accessibility, and reduced risk of harm.

4 Quality management systems

4.1 It is recommended that SQA, HCD and associated activities are performed using a quality management system such as ISO/IEC 90003 or relevant standards to ensure that quality requirements are embedded in the development life cycle process in order to achieve software quality, meet user needs and enhance safety of e-navigation systems.

4.2 This guideline can be applied to the design of systems with varying levels of complexity, regardless of whether a new system is being developed or an existing system is being modified.

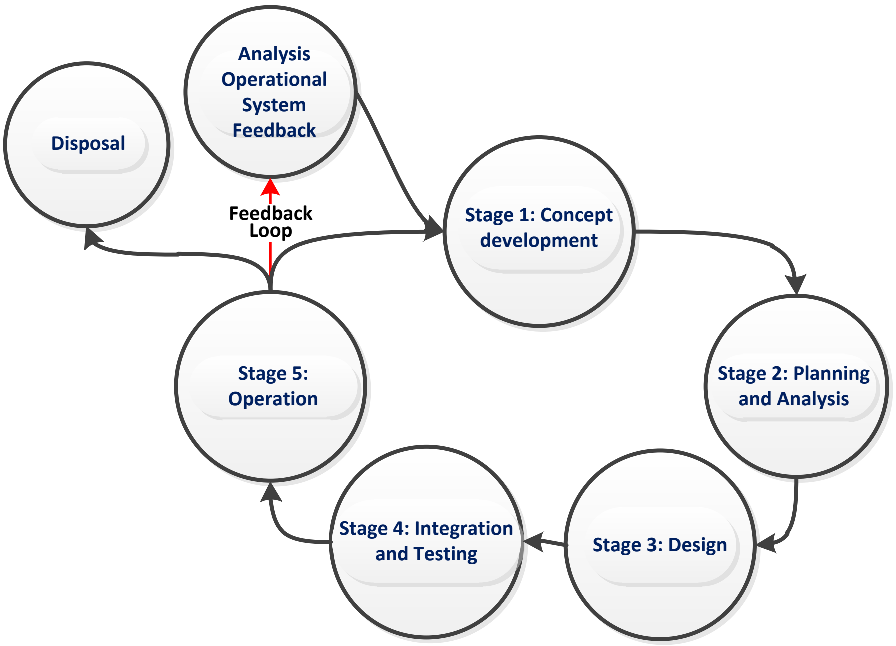

Disposal
Analysis Operational System Feedback
Feedback Loop
Stage 1: Concept development
Stage 2: Planning and Analysis
Stage 3: Design
Stage 4: Integration and Testing
Stage 5: Operation

Figure 2: Generic life cycle

4.3 Figure 2 shows a typical generic life cycle[2] with the stages recommended as a minimum for the application of this guideline to the development of e-navigation systems:

.1 Analysis of operational system feedback;

.2 Stage 1: Concept development;

.3 Stage 2: Planning and analysis;

.4 Stage 3: Design;

.5 Stage 4: Integration and testing;

.6 Stage 5: Operation; and

.7 Disposal.

[2] The system life cycle management approach in IEC61174 (for ECDIS) can be referred to as an example.

4.4 The aim of SQA, HCD and associated UT activities is to ensure that for each stakeholder, user and task requirements are considered in the development process. This takes into account interactions between people, technology and the physical and organizational environments within which they work. Outcomes can be maximized if SQA, HCD and associated activities are applied by teams with relevant multidisciplinary skills and experiences.

4.5 SQA and HCD are performance- and risk-based processes. Hazards are identified, associated risks assessed and, if necessary, risk reduction and control measures are implemented to ensure an acceptable level of quality, usability and safety. Because they are performance-based processes, validation is based on how the outcomes are achieved.

5 Software quality assurance (SQA)

5.1 Key to ensuring software quality in e-navigation is to address the quality attributes that need to be considered in the development and design of e-navigation systems as highlighted in figure 1.

5.2 Software in support of e-navigation can be a product on its own, or part of a larger system and includes data and information. A key function of e-navigation software is to harmonize, integrate, exchange, present and analyze maritime data and information to meet user needs.

5.3 Functional Safety: The performance of systems related to e-navigation software should be assured in terms of required functions and level of integrity. The reliability and availability of safety-related functions should be specified based on stakeholder requirements and traceable through documentation. Functional safety requirements should be defined, implemented and managed throughout the life cycle. The required level of functional safety can vary depending on the designed functionality and intended use, and should be determined by an appropriate risk-based process. Guidance for ensuring functional safety is provided in IEC 61508 or relevant standards.

5.4 Security: It is important to consider and properly address security to prevent cyber-attacks, hacking or other illegal intrusions. Any e-navigation implementation should provide a secure digital environment, in particular: addressing avoidance, prevention and detection of any cyber security threats, locally, regionally and internationally. Guidance on software and cyber security is provided in ISO/IEC 27000 or relevant standards.

5.5 Software Quality Models for e-navigation: This section introduces three types of quality models for e-navigation software systems that are defined by the ISO/IEC 25000 series:

.1 Product quality;

.2 Data quality; and

.3 Quality-in-use.

5.6 The Product quality model categories are: functional suitability, performance efficiency, compatibility, usability[3], reliability, security, maintainability and portability.

5.7 Software quality is also dependant on the quality of input data, which should conform to relevant international standards. As shown in figure 1, data quality is one of the key attributes of e-navigation systems. Data quality requirements and data quality characteristics should be based on ISO/IEC 25012 and related standards (i.e. International Hydrographic Organization

[3] It should be noted that ISO 25010 uses "usability" to describe the attributes that confer quality-in-use. The usage of usability in this guideline is different but very close to quality-in-use.

(IHO) standards for nautical information including Electronic Navigational Charts (ENC)). These standards propose a general data quality model to support organizations to acquire, manipulate and use data with the necessary quality characteristics. It is recommended that Data Quality Assurance (DQA) is performed using a quality management system such as ISO/IEC 90003 or relevant standards.

5.8 A systematic approach to ensure data quality is recommended and can include:

.1 defining and evaluating data quality requirements in data production, acquisition and integration processes;

.2 identifying data quality criteria, also useful for re-engineering, assessment and improvement of data; and

.3 evaluating the compliance of data with legislation and other relevant requirements.

5.9 Producers of input data should have life cycle management practices in place to handle possible data format changes during the life cycle. These life cycle management practices should include timely announcements to software producers and end users about such changes. As part of DQA, producers of input data should test all data in service for conformance with relevant international standards.

5.10 The quality-in-use of a system characterizes the impact that the product (system or software product) has on stakeholders, measuring effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use. It is determined by the quality of the software, hardware and operating environment and by the characteristics of the users, tasks and social environment. All these factors contribute to the quality-in-use of the system. Examples of quality-in-use measures are given in ISO/IEC 25024.

5.11 Appendix 2 provides details of recommended sub-activities to be undertaken during the software life cycle to ensure the development of better quality software.

5.12 Software quality evaluation: The required software quality depends on the intended use or objectives of the system of which the software is a part. Software products need to be evaluated during design, implementation and integration to determine whether the relevant quality characteristics are met.

5.13 Software quality evaluation processes are defined in relevant international standards, such as ISO/IEC 25040 which contains the following activities:

.1 define the purpose and scope of the evaluation and identify software quality requirements;

.2 specify and develop the quality measures and establish decision criteria;

.3 develop the evaluation plan;

.4 carry out the evaluation applying quality measures and the decision criteria; and

.5 review the evaluation results and prepare an evaluation report and provide feedback.

5.14 For each activity, applicable measurement tools, constraints, inputs and outputs are identified. Outputs of previous activities can be used as inputs to subsequent stages. The first activity may include output from previous evaluations as an input.

5.15 When an evaluation is performed concurrently with software product development, associated activities can be performed as part of software life cycle processes (ISO/IEC 12207 or relevant standards) and/or system life cycle processes (ISO/IEC 15288 or relevant standards).

5.16 Figure 3 outlines the main activities that should be undertaken in the software life cycle, as below:

.1 Pre-activity: Preliminary hazard analysis;

.2 Activity 1: Definition of stakeholders and system requirements;

.3 Activity 2: Analysis of system requirements;

.4 Activity 3: Software architecture design and implementation;

.5 Activity 4: Software testing, installation and acceptance;

.6 Activity 5: Software operation and maintenance; and

.7 Activity 6: System disposal.

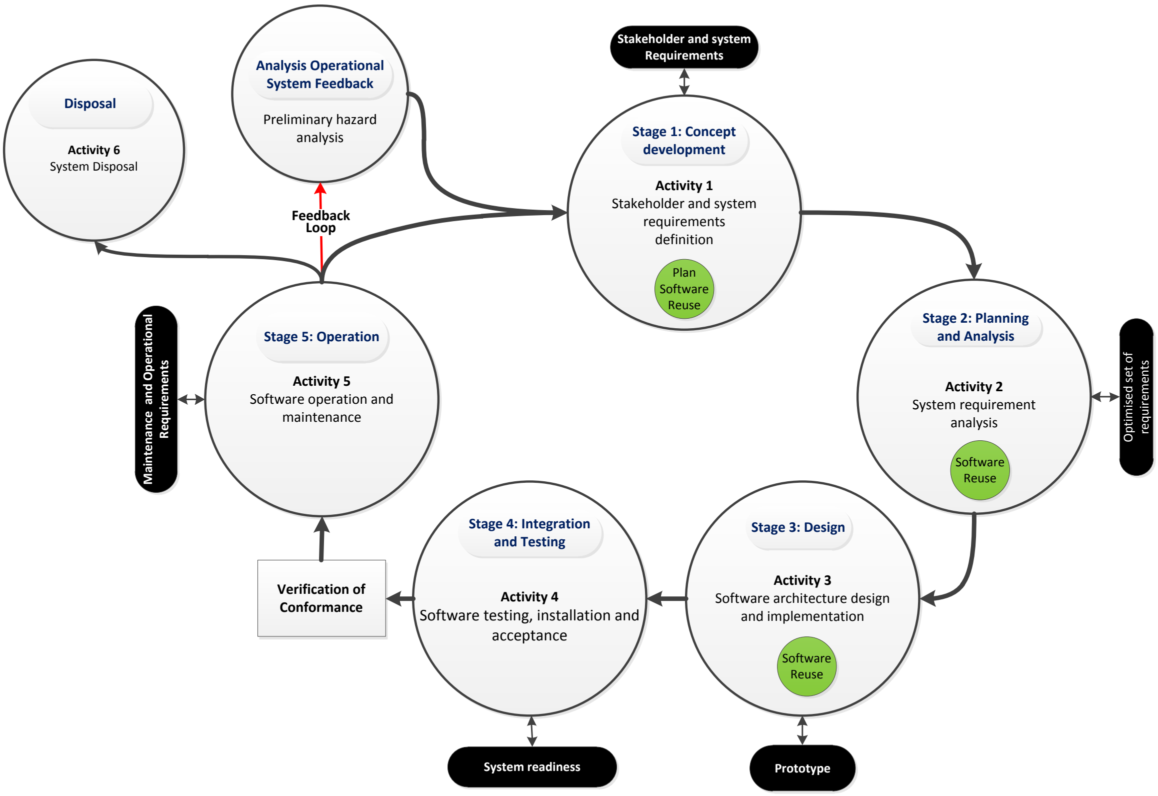

Disposal
Activity 6
System Disposal
Analysis Operational System Feedback
Preliminary hazard analysis
Feedback Loop
Stakeholder and system Requirements
Stage 1: Concept development
Activity 1
Stakeholder and system requirements definition
Plan Software Reuse
Stage 2: Planning and Analysis
Activity 2
System requirement analysis
Software Reuse
Optimised set of requirements
Maintenance and Operational Requirements
Stage 5: Operation
Activity 5
Software operation and maintenance
Verification of Conformance
Stage 4: Integration and Testing
Activity 4
Software testing, installation and acceptance
System readiness
Stage 3: Design
Activity 3
Software architecture design and implementation
Software Reuse
Prototype

Figure 3: Overview of Software Quality Assurance activities

Activity 1: Definition of stakeholders and system requirements

5.17 This activity involves specifying the required characteristics and identifying the context of use of the system being developed. During this activity validation and conformance requirements of the system will also be identified.

Activity 2: Analysis of system requirements

5.18 This activity involves defining a set of functional and non-functional system requirements with various configurations developed in order to ensure an optimized solution. This activity results in a prioritized, approved and updated set of system requirements including SQA requirements which are consistent and traceable.

Activity 3: Software architecture design and implementation

5.19 This activity involves defining and structuring the elements of the system, ensuring it meets defined software quality requirements. The verification between the system requirements and the system architecture should also be carried out during this stage. A strategy for software integration based on the priorities of the system requirements needs to be developed with criteria to verify compliance.

5.20 An important aspect to be considered during the early stages of software design is software reuse. This needs to be considered during stages 1 to 3 of the software life cycle. Software reuse is the use of existing software assets in some form within a software development process. Software assets include products from prior developments such as components, test suites, designs and documentation. Software assets may be modified as needed to meet new system requirements.

Activity 4: Software testing, installation and acceptance

5.21 This activity ensures that the integrated software is compliant with the system requirements. Appropriate methods and standards for testing software should be developed to ensure the reliability and validity of the software qualification test and, as much as possible, conformance to expected results. Software qualification testing should take place in its intended operational environment. As previously mentioned, appropriate test data sets provided by relevant international organizations such as IALA and IHO should be used to ensure conformance to shore-based data. An important pre-condition is to ensure that the use of shore- and ship-based data has been subject to a DQA process. This activity also involves evaluating and testing the integrated system using pre-defined criteria, with evidence produced that demonstrates quality assurance.

5.22 Verification of conformance: It is recommended that certificates of conformance to existing software and data quality should meet relevant standards to ensure the verification of software systems.

5.23 It is recommended that the verification process for e-navigation SQA be carried out by reviewing the related documents on the e-navigation software system or data, by inspecting the implementation of the e-navigation software system and testing the software functions. It is recommended that the testing environment covers berth-to-berth operation, ship-to-ship communication, ship-to-shore communication as well as shore-to-shore communication.

Activity 5: Software operation and maintenance

5.24 This activity involves the identification and evaluation of conditions for correct operation of the software in its intended environment. An operation and maintenance strategy needs to be developed in consultation between the software developers and users. This will ensure that any software and system modifications, upgrades, changes to the existing system interface and updating of system and software documentation are appropriately managed and do not compromise product requirements or safety.

Activity 6: System disposal

5.25 A system disposal strategy should be developed to facilitate knowledge retention and analysis of long-term impacts. A hardware disposal strategy should also be developed to promote the use of non-hazardous materials during manufacturing.

5.26 Note that some of the software quality activities described in this section will also overlap with the HCD process activities described in section 6.

6 Human-Centred Design (HCD)

6.1 HCD helps to ensure that human factors-related knowledge and techniques in system design and development processes are addressed, thus ensuring that user needs and safety are met. The primary goals of usability and safety through efficiency, effectiveness, risk reduction and satisfaction should always be maintained.

6.2 Key elements of HCD are the involvement of multi-disciplinary teams including users and an iterative approach to design. HCD is driven by knowledge about use, derived from evaluation and testing with users, the results of which drive a formal feedback loop in each of the design stages to ensure usability and safety. E-navigation systems should aim to ensure that navigational and associated tasks are effectively supported, with usability being the measure that is tested to ensure that this is achieved.

6.3 Figure 4 outlines the activities that should be undertaken in each of the life cycle stages, illustrating the interdependence of each activity. The following HCD activities are carried out to inform development throughout the life cycle:

.1 Pre-activity: Conduct Early Human Element Analysis (EHEA);

.2 Activity 1: Understand and specify the context of use;

.3 Activity 2: Identify the user requirements;

.4 Activity 3: Produce and/or develop design solutions to meet user requirements;

.5 Activity 4: Evaluate the design against usability criteria; and

.6 Activity 5: Maintain operational usability.

6.4 Fundamental to HCD is the collection of user feedback through UT. UT is an effective means to discover and resolve potential usability and design issues early as well as throughout the life cycle of a system by using an iterative testing approach to ensure a safe, satisfactory, effective and efficient system. Evaluation through usability testing is carried out iteratively at all stages in the life cycle and provides input for future versions of systems.

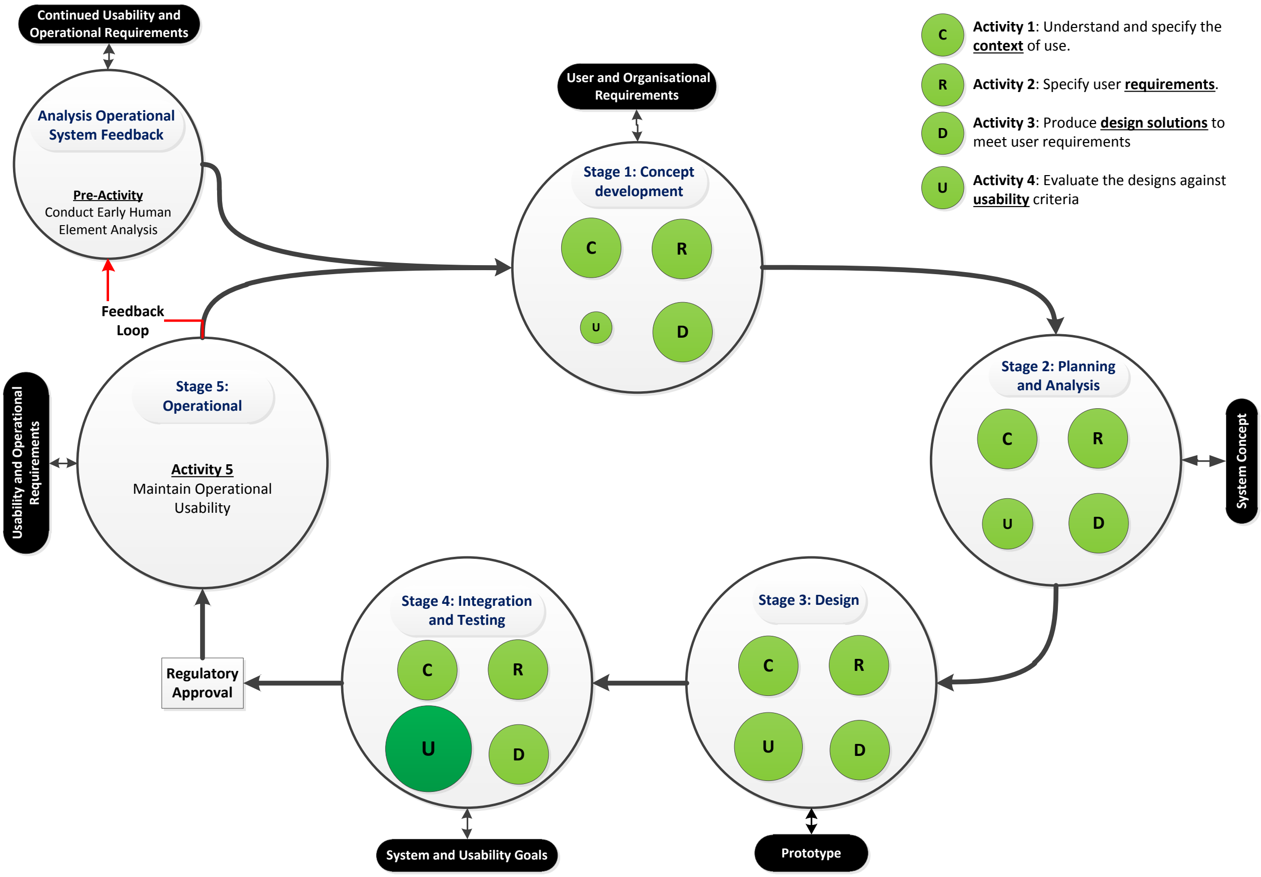

Continued Usability and Operational Requirements
Analysis Operational System Feedback
Pre-Activity
Conduct Early Human Element Analysis
Feedback Loop
User and Organisational Requirements
Stage 1: Concept development
C
R
U
D
Activity 1: Understand and specify the context of use.
Activity 2: Specify user requirements.
Activity 3: Produce design solutions to meet user requirements
Activity 4: Evaluate the designs against usability criteria
Usability and Operational Requirements
Stage 5: Operational
Activity 5
Maintain Operational Usability
Stage 2: Planning and Analysis
System Concept
Stage 4: Integration and Testing
Stage 3: Design
Regulatory Approval
System and Usability Goals
Prototype

Figure 4: Overview of HCD for e-navigation systems

Pre-activity – Conduct an Early Human Element Analysis (EHEA)

6.5 The pre-activity involves establishing an understanding of usability issues. This may include conducting an Early Human Element Analysis (EHEA), which involves capturing lessons learned from operating similar systems, prioritization of human element issues, all documented in a risk register. Advantages of an EHEA include identifying emerging or innovative requirements and needs.

Activity 1: Understand and specify the context of use

6.6 Context of use consists of the users' characteristics (and their associated individual cognitive and physical factors), their goals and interactions with tasks, stakeholders, physical operating environments (i.e. the work environment where the e-navigation system is being used), and social and management environments (i.e. training, the company and its management policies and procedures). The context of use appropriately takes into account relevant aspects of socio-technical systems. As shown in figure 4, this is conducted during stages 1 to 4 of the design and development life cycle of a system with varying levels of intensity, as necessary.

6.7 Information gathered when establishing context of use can help frame scenarios which provide realistic examples of future use. Scenarios encourage designers to consider the characteristics of the intended users, their tasks and environment, and they enable usability issues to be explored at an early stage in the design process. They are best developed in conjunction with users. Scenarios also need to include common, routine tasks and rarely performed but critical tasks (e.g. those that need to be performed in an emergency).

6.8 When new systems are to be used in combination with existing systems, the context of use needs to include the overlapping elements and the interaction of the new system with the elements of the other systems.

Activity 2: Identify user requirements

6.9 The user requirements include user needs and task-related needs identified in the context of use of a system and task-related activity. This involves progressing user and contextual needs into an explicit statement of user requirements in relation to the intended context of use and the business objectives of the system.

6.10 Activity 2 involves some or all of the following:

.1 clarification of system goals;

.2 analysis of stakeholders' needs and expectations;

.3 analysis of user needs and expectations;

.4 resolution of conflicts between different user and task requirements;

.5 identification of safety issues (risks and hazards);

.6 analysis of training needs;

.7 analysis of system/equipment familiarization requirements;

.8 generation of operational concept and top-level system requirements;

.9 ensuring the quality of user requirement specifications; and

.10 further development and refinement of task-based scenarios and test cases.

Activity 3: Produce and/or develop design solutions to meet user requirements

6.11 Activity 3 involves applying the knowledge gained earlier about the intended context of use, including user roles, responsibilities, tasks and their outputs to design solutions.

Activity 3 may involve some or all of the following:

.1 development of prototypes and/or specific test beds;

.2 development of design solutions and altering them based on UT and other feedback;

.3 designing user-system interaction and user interface to meet context of use and usability requirements; and

.4 development of a maintenance/support regime.

Activity 4: Evaluate the design against usability criteria

6.12 Activity 4 is the basis on which UT is carried out as appropriate to the particular stage in the life cycle. The evaluation of the design against usability criteria should be conducted before a system is deployed operationally and should, as a minimum, employ test participants who are representative of user groups.

6.13 Planning the UT involves:

.1 selecting scenarios and test cases;

.2 identifying and recruiting testing participants;

.3 choosing methods, techniques and documentation for collecting and analysing data; and

.4 determining acceptance criteria.

6.14 Measurements of usability should include effectiveness, efficiency and satisfaction. Appropriate methods include expert evaluation (such as observation of scenario/task performance), questionnaires, interviews, walk-throughs, task-based user testing and observations. Typical measures for these are included in ISO/TR 16982:2002. Appendix 3 includes an example of a usability method referred to as the "usability rating method" applied to ECDIS.

Activity 5: Maintain operational usability

6.15 Activity 5 addresses HCD in a system's operation. Throughout a system's operational life users are trained and will use the system. They are therefore able to provide accurate feedback on use and usability. This feedback may lead to refinements to the system and subsequently improved performance in newer versions, and hence activity 5 is linked to the pre-activity through a feedback loop.

APPENDIX 1

International standards on SQA, HCD and associated activities

Topic	Relevant standard	Subject
Human-Centred Design	ISO 9241-210	Ergonomics of human-system interaction – Human-centred design for interactive systems.
	ISO 9241-110	Ergonomics of human-system interaction – Dialogue Principles.
	ISO TR 18529	Ergonomics of human-system interaction – Human-centred life cycle process definitions
Usability Testing	ISO/TR 16982	Ergonomics of human-system interaction – Usability methods for supporting human-centred design.
System and software Quality Requirements and Evaluation (Square)	ISO/IEC 25010	Systems and software quality models
	ISO/IEC 25012	Data quality models
	ISO/IEC CD 25024	Measurement of data quality (under development and replacing ISO/IEC TR 9126-4:2004)
	ISO/IEC 25040 ISO/IEC 25041 ISO/IEC 25042 ISO/IEC 25045	Quality Evaluation Division (Evaluation process, guides and modules)
	ISO/IEC 25060	Common Industry Format (CIF) for usability: General framework for usability-related Information
	ISO/IEC 25062	Common Industry Format (CIF) for usability test reports
System and Software Assurance	ISO/IEC 15026-1	Part 1: Concepts and vocabulary
	ISO/IEC 15026-2	Part 2: Assurance case
	ISO/IEC 15026-3	Part 3: System integrity levels
	ISO/IEC 15026-4	Part 4: Assurance in the life cycle
System and software life cycle processes	ISO/IEC 15288	System life cycle processes
	ISO/IEC 12207	Software life cycle processes
Ships and marine technology – Computer applications	ISO 17894	General principles for the development and use of programmable electronic systems in marine applications
Software Quality Management	ISO/IEC 90003	Guidelines for the application of ISO 9001 to computer software

APPENDIX 2

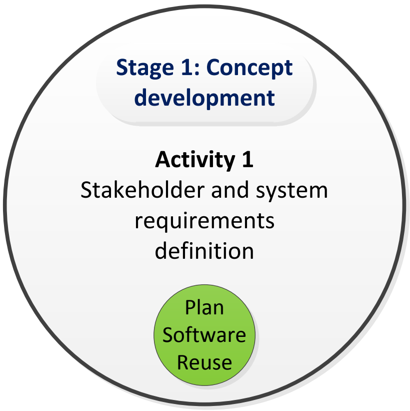
Recommended Software Quality Assurance activities and sub-activities

1 This appendix details actions and associated expected outcomes that can be used to assist with software development and Software Quality Assurance (SQA) activities.

2 Activities, and where appropriate sub-activities, can be specific or holistic in nature. The expected outcomes may result in documentation which should in general align with the requirements of the quality management system being used. This will in many cases result in evidence showing that the results of activities undertaken comply with top-level requirements for the e-navigation systems being developed.

3 Depending on the required characteristics of the software system, boundaries between activities may be flexibly arranged to help assist with effective SQA across the software life cycle.

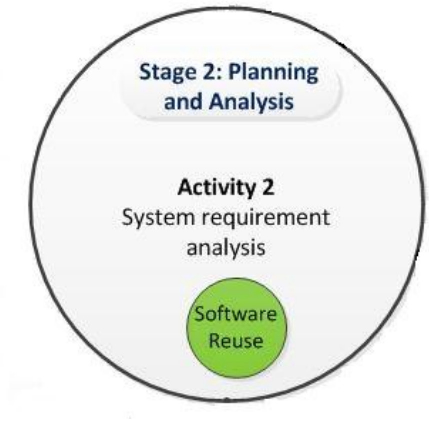
4 For Activity 1, it is recommended to define stakeholder requirements which can include the following actions and expected outcomes:


Stage 1: Concept development
Activity 1
Stakeholder and system requirements definition
Plan Software Reuse

- Define stakeholder requirements
- Plan software reuse

Activity or sub-activity	Actions/Outcomes
Stakeholder requirements definition	✓ Elicit needs of stakeholders and identify the context of use; ✓ Develop specification of the required characteristics and context of use; ✓ Definition of the constraints on the system to be developed; ✓ Traceability of stakeholder requirements to stakeholders and their needs; ✓ The basis for defining the system requirements; ✓ The basis for validating the conformance of the services; and ✓ A basis for negotiating and agreeing to supply the system to be developed.

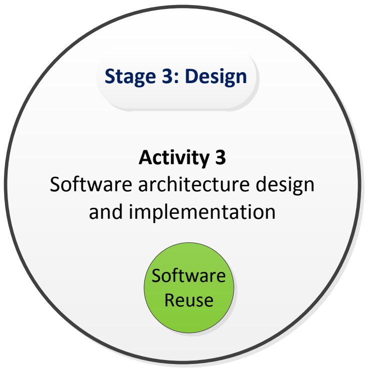
5 For Activity 2, it is recommended to conduct a system requirement analysis which can include the following actions and expected outcomes:



- System requirement analysis
- Software reuse

Activity or sub-activity	Actions/Outcomes
System requirement analysis	✓ A defined set of functional and non-functional requirements; ✓ Systems configuration for the optimized solution; ✓ Correctness and testability analysis of the system requirements; ✓ Impact analysis of the system requirements on the operating environment; ✓ Prioritized, approved and updated set of the requirements when needed; ✓ Consistency and traceability between the system requirements and the stakeholder's requirements baseline; and ✓ Impact analysis of changes to the baseline for cost, schedule and technology.

6 For Activity 3, it is recommended to conduct system architectural design and implementation which can include the following actions and expected outcomes:



- Software architecture design
- Implementation
- Software reuse

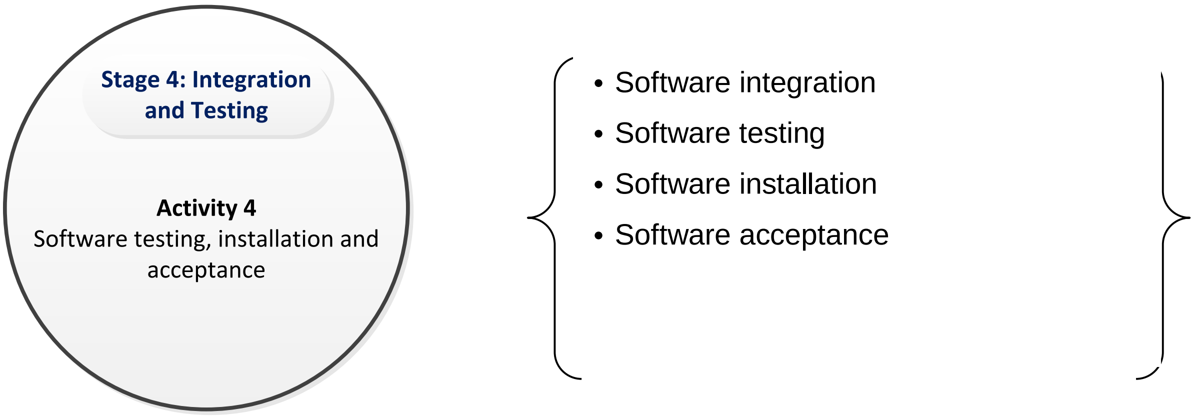
Activity or sub-activity	Actions/Outcomes
Software architectural design	✓ A software architecture design defining the elements of a system that meets the defined requirements; ✓ Functional and non-functional requirements of the system; ✓ Allocation of some of requirements to the elements of the system; ✓ Internal and external interfaces of each system element;

Activity or sub-activity	Actions/Outcomes
	✓ Verification between the system requirements and the software architecture; ✓ Traceability to the stakeholder's requirements base line; ✓ Maintaining the consistency and traceability between the system requirements and software architecture design; ✓ Base lining the relationships between the system requirements and the architecture design and informing all affected stakeholders; and ✓ Incorporating human factors principles and knowledge in system design.
Implementation	✓ A strategy for software integration based on the priorities of the system requirements; ✓ Criteria to verify compliance with the system requirements; ✓ Verification of system integration by using the defined criteria; ✓ A regression strategy for re-testing the system when changes are made; ✓ Establishment of consistency and traceability between the system design and the integrated system elements; ✓ An integrated system with compliance with the system design; and ✓ An integrated system with a complete set of usable deliverable system elements.

7 The software reuse activity falls within Activities 1, 2 and 3, which can include the following actions and expected outcomes:

Activity or sub-activity	Actions/Outcomes
Software reuse	✓ Establishing the policy, plan and processes for software reuse; ✓ Selecting representation forms for the domain models and the domain architectures; ✓ The boundaries of the domain and its relationships to other domains; ✓ A domain model that captures the essential common and different features, capabilities, concepts, and functions in the domain; ✓ A domain architecture describing the family of systems within the domain, including their commonalities and differences; ✓ Specification of assets belonging to the domain; ✓ Acquisition, development and maintenance of assets belonging to the domain throughout their life cycles; and ✓ Maintaining the domain models and architectures throughout their life cycles.

8 For Activity 4, it is recommended to conduct software integration, qualification and testing, installation and acceptance, which can include the following actions and expected outcomes:



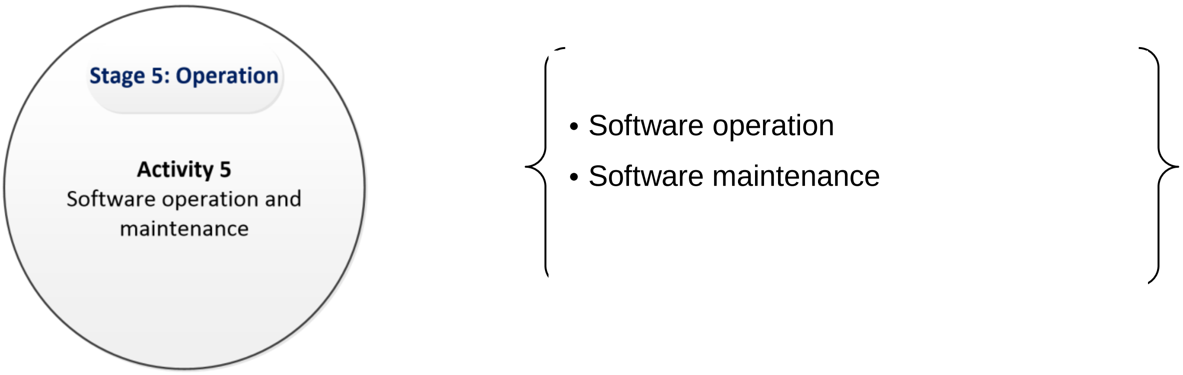
Stage 4: Integration and Testing

Activity 4
Software testing, installation and acceptance

- Software integration
- Software testing
- Software installation
- Software acceptance

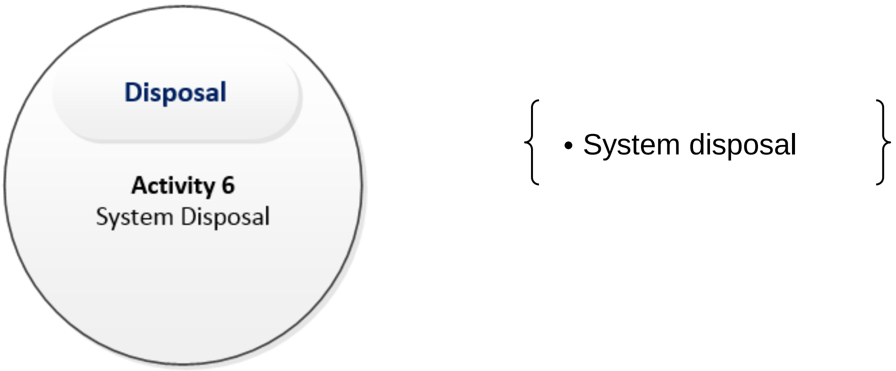
Activity or sub-activity	Actions/Outcomes
Software integration	✓ Test coverage of system requirements; ✓ Appropriateness of test methods and standards used; ✓ Conformance to expected results; ✓ Feasibility of software qualification testing; and ✓ Feasibility of operation and maintenance.
Software qualification testing	✓ A criteria for evaluating compliance with system requirements; ✓ Testing the integrated system using the defined criteria; ✓ Recording the test results; and ✓ Assuring readiness of the system for delivery.
Software installation	✓ A software installation strategy; ✓ Criteria for software installation showing compliance with the software installation requirements; ✓ Installing the software in the target environment; and ✓ Assuring readiness of the software product for use in its intended environment.
Software acceptance support	✓ The completed software system; ✓ Acceptance tests and reviews by acquirer; ✓ Putting the completed software system into operation in the intended environment; ✓ Identification of problems detected during acceptance; and ✓ Notification of the identified problems to the responsible party.

9 For Activity 5, it is recommended to conduct the software operation process and the software maintenance process which can include the following actions and expected outcomes:



Activity or sub-activity	**Actions/Outcomes**
Software operation	✓ An operation strategy; ✓ Identification and evaluation of conditions for correct operation of the software in its intended environment; ✓ Testing the software to determine the operation in its intended environment; ✓ Operating the software in its intended environment; and ✓ Assistance and consultation for the stakeholders of the software product in accordance with the agreement.
Software maintenance	✓ A maintenance strategy to manage modification and migration of products according to the release strategy; ✓ Identification of the impact of changes to the existing system on organization, operations or interfaces; ✓ Updating system and software documentation as needed; ✓ Modification of products without compromising requirements; ✓ Migration of product upgrades including data upgrade to the customer's environment; and ✓ Informing all affected parties of the system software modifications.

10 For Activity 6, it is recommended to conduct the system disposal process which can include the following actions and expected outcomes:



- System disposal

Activity or sub-activity	Actions/Outcomes
System disposal	✓ A software/hardware disposal strategy; ✓ Disposal constraints; ✓ Destruction of software/hardware elements as needed; ✓ Storage of software/hardware elements as needed; ✓ The software environment left in an agreed-upon state; ✓ Records allowing knowledge retention of disposal actions and any analysis of long-term impacts; ✓ Evidence showing that the results above comply with top-level requirements of the e-navigation systems to be developed; and ✓ Confirmation that disposal is not detrimental to health, safety, security and the environment.

APPENDIX 3

Example of Usability Testing

1 This appendix provides information on Usability Testing (UT) and uses ECDIS as a closely aligned example relevant to future e-navigation systems. This UT example aligns with Stage 4 of the HCD process for evaluating the performance of essential tasks by competent users. The selection of test participants is important and has a bearing on the quality of test results.

2 If tasks require operations based on navigational experience or knowledge, then appropriate participants should be selected. Tasks that are generally performed by less experienced or knowledgeable personnel should be similarly tested.

3 The UT activity involves the following steps:

.1 Planning;

.2 Preparation;

.3 Undertaking and controlling tests;

.4 Evaluation of results; and

.5 Use of feedback.

4 Only the steps related to planning and evaluation of results are explained in this appendix since these steps are the most important.

5 A UT plan should be developed by defining scenarios and identifying the most important or critical tasks that users must perform. Users and the test environment should also be identified.

6 A goal-based approach should be used when setting the tasks with the aim of facilitating flexible yet practical assessment of the target system.

7 The following steps can be part of the goal-based approach:

.1 definition of goals based on the context of use of the system, which may come from functions stipulated in internationally agreed performance standards;

.2 specify functional requirements or the criteria to be satisfied in order to conform to the goals, taking into account the relevant performance standards and user requirements;

.3 specify "usability" requirements that must be achieved during testing, based on the aspects of effectiveness, efficiency and satisfaction; and

.4 prepare tests that will assist in verifying the extent to which the system conforms with the identified goals.

8 In the case of ECDIS goals could include "to plan and display the ship's route for the intended voyage and to plot and monitor positions throughout the voyage", based on SOLAS regulation V/19.2.1.4.

9 Similarly, functional requirements for ECDIS could be defined based on the IMO's ECDIS performance standard (resolution MSC.232(82)). The following example of ECDIS functional requirements relates to nautical data handling necessary for safe navigation, with the following sub-requirements:

.1 chart data handling (example: change display orientation, mode, etc.);

.2 own ship data handling (example: read position, speed, etc.); and

.3 tracked target (TT) and radar data handling (example: show TT symbols overlaid on ECDIS chart screen, etc.).

10 In the case of ECDIS, "usability" can be evaluated in terms of user effectiveness and efficiency for each of the tasks and overall satisfaction of the system (for example through subjective evaluation). As highlighted in table 1, measures of effectiveness relate the selected user goals to the accuracy and completeness with which these goals can be achieved. In this example, the achievement rate is used as a measure of "effectiveness". The four levels and their criteria are listed in table 1. Usability outcomes can be based on the "dialogue principles", as identified under ISO 9421-110, using UT methods based on ISO/TR16982. It is important that methods for evaluating usability are selected when devising the UT plan.

11 Scenarios and test tasks can also be created to satisfy the functional requirements. The following are examples of tasks for a basic display handling scenario:

Task 1: Adjust display modes and scale to meet operator's needs

Task 2: Obtain information about a lighthouse

Task 3: Measure the bearing and distance to a landmark

Task 4: Overlay a tracked target symbol and obtain information about the target

12 Criteria should be set to establish the degree to which tasks are achieved and also to capture user feedback on satisfaction with the operation of the system. Table 1 provides simple examples of achievement criteria for each task. Quantitative performance criteria such as time taken to complete tasks can also be included.

13 For the evaluation of system performance the level of task achievement can be useful (e.g. the time required to complete tasks). Questionnaires can assist with overall subjective system evaluation.

Table 1: Examples of achievement criteria for measures of effectiveness

Achievement level		Criteria
Achieved	1	✓ Participants understood the information correctly and operated properly with confidence. ✓ In case participants made some mistakes but noticed the mistakes immediately and achieved the goal smoothly, this should be considered "achieved smoothly".
	2	✓ Participants completed the task properly by themselves, even with some hesitation or confusion. ✓ In case participants took time to find the first action or to recover from errors but completed the task, this should be considered "achieved not smoothly".
Not achieved	3	✓ Even if participants completed the task properly, it should be considered "not achieved with errors" if the participants could not understand the information correctly or if achievement took a large number of interactions.
	4	✓ Participants could not complete the task by themselves and needed suggestions from the moderator.

14 To satisfy quality management system requirements a UT report should be developed. ISO/IEC 25062 provides an example for a template that can be used for a UT report.

UT methods that can be applied at various stages in the life cycle (based on ISO/TR 16982)

Name of the method	**Direct involvement of users**	**Short description of method**	**Life cycle stage**
Observation of users	Y	Collection of information in a precise and systematic way about the behaviour and the performance of users, in the context of specific tasks during user activity.	4
Performance-related measurements	Y	Collection of quantifiable performance measurements in order to understand the impacts of usability issues.	4
Critical incident analysis	Y	Systematic collection of specific events (positive or negative).	1
Questionnaires	Y	Indirect evaluation methods which gather users' opinions about the user interface in predefined questionnaires.	1 and 2
Interviews	Y	Similar to questionnaires but with greater flexibility involving face-to-face interaction with the interviewee.	2
Thinking aloud	Y	Involves having users continuously verbalize their ideas, beliefs, expectations, doubts, discoveries, etc. during their use of the system being tested.	3 and 4

Name of the method	**Direct involvement of users**	**Short description of method**	**Life cycle stage**
Collaborative design and evaluation	Y	Methods which allow different types of participants (users, product developers and human factors specialists, etc.) to collaborate in the evaluation or design of systems.	Any
Creativity methods	Y/N	Methods which involve the elicitation of new products and system features, usually extracted from group interactions. In the context of human-centred approaches, members of such groups are often users.	1 and 2
Document-based methods	N	Examination of existing documents by the usability specialist to form a professional judgement of the system.	1 and 2
Model-based approaches	N	Use of abstract representations of the evaluated product to allow the prediction of users' performance.	2 and 3
Expert evaluation	N	Evaluation based on the knowledge, expertise and practical experience in ergonomics of the usability specialist.	Any
Automated evaluation	N	Algorithms focused on usability criteria or using ergonomic knowledge-based systems which diagnose the deficiencies of a product compared to pre-defined rules.	4
Simulation	N	Use of computer simulation modelling tools used for initial evaluations.	2 and 3

___