Lessons learnt from IEC61508 software assessments

R H Campbell MEng CEng FS Eng MIET InstMC, R M Phillips MEng FS Eng MIET, C Allsopp MSc, BSc

* *Frazer-Nash Consultancy, Bristol, UK*

* Corresponding Author. Email: rhcampbell@iee.org

Synopsis

With the advances in platform automation and the publication of NAN 06/2018 - Software integrity (previously NAN 02/2016), there has been a drive towards compliance to IEC61508 in the naval domain. Over the last few years Frazer-Nash have conducted a number of Original Equipment Manufacturer (OEM) audits and assessments to determine whether the requirements for safety critical software development in IEC61508 have been followed for a specific system or if a supplier's development processes are in line with the aspects of the standard which focus on software development, namely Part 3 - Software requirements. These audits have revealed some common problems across suppliers and highlighted that IEC61508 requires both organisational safety management processes as well as those specific for a system.

As the naval industry looks to adopt the processes outlined in IEC61508 or an equivalent standard, this paper will present some of the lessons learnt from our IEC61508 assessments and offer some advice for new and existing suppliers. The paper will highlight some of the issues going forward as the development of safety critical systems is not a new concept but the specific factors arising in the naval domain from operating in variable environments, changing safe state conditions and ever increasing function complexity, present a key challenge.

Included in the paper will be a view on how other industries are tackling IEC61508 compliance and where the strategies that have been adopted may be applicable in the naval domain as well as new tools which could assist with the development of safety critical systems.

Keywords: Functional Safety; IEC61508; Safety Critical Software

1. Introduction

Unlike a random hardware failure which can be predicted and failure rates quoted, software does not fail probabilistically; any error encountered is a systematic error. Due to the size and complexity of software it is not usually possible to exhaustively test its behaviour. Hence, should the operating environment change such that the software enters a state that has not previously been tested, there is a risk that previously un-executed code could cause a system failure. To minimise the potential for this, and to ensure the integrity of the software is in line with the reliability of the hardware it is operating on, IEC61508 provides guidance to be followed when producing high integrity Electronic, Electrical or Programmable Electronic Systems (E/E/PES).

The proliferation of automation and software within the naval domain has grown significantly within the lifetime of the current UK Navy fleet and the Naval Authority's guidance for the development of software for both vessel wide systems and smaller 'package' units has now been steered towards adopting IEC61508 [Ref. 3] principles. Naval Authority Notice (NAN) 06/2018 – Software Integrity [Ref. 4] gives the latest guidance from the Naval Authority Group. NAN 06/2018 states the guidance is applicable for all equipment whose failure could cause, impair the mitigation of or impair the recovery from a Key Hazard.[1]

Author's Biographies

Ross Campbell is a Group Leader within the Frazer-Nash Consultancy Electronic and Electrical Engineering Technical Area. Ross has been working in the field of Functional Safety for the last 8 years and delivered a range of projects from Independent Technical Assessments of Safety Systems to design and commissioning of SIL 2 systems; Ross has also lead the development of Safety Cases of Class 1 and 2 Nuclear Safety Systems and is the company representative on the 61508 Association.

Ross Phillips is a Consultant engineer who started out in the Power Generation sector as a Control and Instrumentation engineer. Various roles since then built up his experience and expertise within ATEX/DSEAR, Alarm Management, HMI design and Functional Safety. A TUV qualified FSEng, Ross has audited UK Destroyer suppliers, assessed plant design changes within the Nuclear Power Industry and written mechanical safety specifications within the Transport sector.

Chris Allsopp is the Group Leader for the Embedded Software team within Frazer-Nash Consultancy. Chris has 10 years' experience of developing and assuring safety critical software across a range of sectors including defence, nuclear and aerospace. Chris has worked on various naval platform and combat system projects and has led Frazer-Nash's support to the development of Class 1 Nuclear Safety Systems across the full lifecycle

http://doi.org/10.24868/issn.2631-8741.2018.015

NAN 06/2018 comprises of two parts: Part A – Key Hazard Assessment and Part B – Key Hazard Mitigation. Part A is applicable to all projects and involves failure analysis to identify whether the system can cause, impair the mitigation of or the recovery from a key hazard. If the failure analysis identifies the system within this category then Part B is applicable. In Section B3 the guidance gives two routes for compliance of new software: compliance to Def Stan 00-55 [Ref. 7] or a combination of compliance to a recognised standard such as IEC61508 and a software safety case. Although Def Stan 00-55 is given as an alternative route to compliance many of the principles of the standard align with IEC61508 and so the topics covered in this paper are applicable. Def Stan 00-55 also permits the use of IEC61508 as a means to build towards a compliant system but recognises there are military standard deltas which need to be addressed.

This paper firstly identifies the common areas of concern and non-compliances we have witnessed when assessing software based systems for naval use against IEC61508 compliance. This paper then goes on to discuss five strategies for consideration when developing software and systems to be compliant to either SIL 1 or SIL 2. These include strategies which have been adopted in other sectors and we believe could be suitable for the naval domain.

The authors note previous papers have identified 'lessons learnt' from the functional safety activities, including from Reeve [Ref. 1] and Generowicz [Ref. 2], which focus on the development of individual devices looking for certification of SIL 'capability' and the assessment of process / manufacturing plants to determine safety system requirements respectively. Whilst the key findings and commentary in these papers remains valuable in the development of safety critical systems today, the authors perceived the value of a paper to discuss the challenges associated specifically with the naval domain and the complexity of integrating multiple devices or systems to provide the required safety functions at a whole vessel level.

2. Common Areas of Concern Identified

This section identifies some of the common areas, highlighted by previous assessments, which did not meet the requirements of IEC61508. These areas of concern were at varying stages across the development lifecycle of a system and often as a result of the requirements of NAN 06/2018 Part B being addressed too late on in the development of a vessel. Carrying out assessments whilst detailed software design is either underway or already completed makes demonstrating compliance to a safety critical software standard very challenging. A disconnect can arise between the vessel system requirements and the requirements a sub-system or component supplier actually works towards, which ultimately may result in significant delays and added costs.

2.1. Project Planning

As stated in IEC61508-1 Section 6.1, the objectives for the management of functional safety is to '*specify the responsibilities....of those who have responsibility for an E/E/PE safety related system*' and '*to specify the activities to be carried out.*' During assessments of suppliers we rarely found a transparent and detailed functional safety plan with defined roles and responsibilities. Project plans were often produced to show the development of a system but this rarely included specific functional safety detail.

A functional safety plan should describe the activities at each stage of the project which are required to develop and substantiate high integrity software. Given the complexities of safety critical software and the number of required activities to demonstrate compliance, a functional safety plan is an essential tool to structure the key development stages from producing the system requirements (developed from the hazard analysis), to implementing a system which meets those requirements and finally, demonstrating by suitable means the requirements have been met.

IEC61508-1 gives the necessary attributes of a complete project functional safety plan; for naval applications this would likely involve multiple plans. A project/vessel wide plan would define the framework and reference further plans for individual systems as well as signposting to plans provided by each supplier of a sub-system. These plans should be lifecycle documents, updated to show any modifications in the development process. From the project wide plan an auditor would then be able to trace through to system and sub-system plans as well as observe the document architecture which describe the development and substantiation of each system. We have not seen such an integrated approach in any project we have assessed within the naval domain. Such transparency builds confidence that interfaces within a project are being managed and aids any independent assessment.

Each functional safety plan is required to include the responsibilities of the key project roles at different stages in development of the project or associated sub-system. As noted in IEC61508-1 Sections 6.2.13 and 6.2.14 the competence of all persons involved in a safety critical system shall be assessed to be appropriate. The plan is a useful document either to include the roles and responsibilities and demonstrate the individuals performing these

http://doi.org/10.24868/issn.2631-8741.2018.015

roles are suitably qualified and experienced, or to reference separate competency assessments. During our assessments we found large variability in the management of competence, ranging from no competency management framework at all, to organisations with structured levels of knowledge and experience which were signed-off by the company technical expert / design authority of each technical discipline. Discussion of specific competency management frameworks is outside the scope of this paper but guidance for applying competence management to work with Safety Systems is given by the IET [Ref. 8].

Separately from the overall functional safety plan, IEC61508-3 gives the requirements for a specific software safety lifecycle which '*shall be divided into elementary activities with the scope, inputs and outputs specified for each phase*.' With IEC61508 having conditions for compliance with each stage of a lifecycle, developing a software safety plan is essential to ensuring these conditions are appropriately considered. Section 7.1.2.7 of Part 3 states '*Success in achieving systematic safety integrity depends on selecting techniques with attention to the following factors:*

> *– the consistency and the complementary nature of the chosen methods, languages and tools for the whole development cycle;*
>
> *– whether the developers use methods, languages and tools they fully understand;*
>
> *– whether the methods, languages and tools are well-adapted to the specific problems encountered during development.*'

The software safety plan should demonstrate compliance to these requirements. Previous assessments have again reviewed a varying degree of compliance, from suitably detailed software safety plans which describe the framework for developing the software and competence of those leading the defined activities, to other projects for which a software safety plan did not exist. Experience has shown that the confidence level in the final software has had a strong relationship to whether a software safety plan had been produced and the content within it.

2.2. Requirements Traceability

On a naval vessel the requirements for a sub-system should flow down from the whole vessel and system requirements. Requirements will originate from both safety and functional domains and describe the characteristics and performance of systems and eventually sub-systems.

IEC61508 states the need for a software safety requirements specification. Our preference would be for this requirement to be met by a software requirements specification with a safety section, as opposed to two separate documents. This means the complete system software requirements can be captured in one place. These requirements should include any design or development standards as well as good practice the system should conform to.

Our assessments often found the safety requirements to be focused on a sub-system instead of the function(s) being performed by the system. For example, a requirement might simply state that the sub-system shall meet SIL 2, with no additional detail to understand what the functional requirements were, nor traceability to understand which whole vessel requirement this function related to. This could mean parts of the sub-system which did not relate to the required function were being assessed for compliance where this may not have been necessary.

Traceability back to higher level system and vessel requirements helps to give confidence these principal requirements have been met. A well-structured process also helps in the management of interfaces so each system or sub-system developer is fully aware how their system will interact with those around it. Assessments often highlighted a lack of evidence of these interfaces being actively managed, indicating potential programme risks.

In Section 3.2 we discuss an approach which could help ensure requirements are managed and 'flowed down' to all applicable sub-systems.

2.3. Failure to Perform integration testing

During the course of our naval domain assessments, we frequently observed that integration testing in the final system configuration is only scheduled to be carried out on-board the vessel after sub-system commissioning. This is a surprising approach as it carries significant technical and programme risk.

Given the increasing complexity and interconnectedness of marine systems, more robust techniques should be adopted to build confidence in the system under development and its ability to integrate with other platform systems. Good practice has been applied in other technical streams within the naval domain, for example combat systems, as well as other highly regulated industries we are familiar with. This includes:

- Incremental integration of sub-systems and systems as development maturity increases;
- Use of simulators and interface emulators to formally and informally support the de-risking of integration during development;
- Following a model-based design approach as this drives consideration of external systems and physical interactions to the heart of the development process.

In Section 3.3 we discuss an approach which could help address this risk for safety system development within the naval domain.

2.4. Tool Validation

Across industry, the development of safety related software is increasingly reliant on the use of supporting tools throughout the lifecycle. These tools offer increased automation and aid in managing design complexity. IEC 61508-3 recommends such approaches, but requires arguments to be made regarding the suitability of tools in relation to their impact on the executable code. This is specifically discussed in Section 7.4.4 of IEC61508-3 and in Section 3.2.11 of IEC61508-4; the latter defining the three classes of tool. It is important to note that this validation should be performed in the context of the use cases of the tools and the wider development processes that mitigate the potential insertion of errors and a failure to detect them.

Our experience is that tool validation is either not performed or not documented in a satisfactory manner, potentially leading to errors being introduced into the software or errors not being detected. A lack of tool validation could be identified during audit or certification activities, requiring significant generation of evidence or development rework. This is likely to lead to substantial cost and delivery implications.

2.5. Use of COTS Items and Proven In Use Substantiations

IEC 61508 allows for the use of a component or system which has not been developed in accordance with the standard, provided instead, a ‘Proven In Use’ (PIU) argument can be made for why it is suitable for the required SIL of the function it performs or contributes to. At first glance, this seems to be a suitable approach to incorporate systems with software that was perhaps developed ‘pre-61508’ but have been “tried and tested” over many years. In reality however, the standards place considerable constraints on the information required to make such an argument, which in practice can be almost impossible to collect. For example, in a recent assessment, the software elements of a proposed sub-system which had historically been used on vessel throughout the world, fell short of the PIU constraints in multiple ways:

- The proposed solution had a different version of software than previously installed systems;
- The supplier was unable to provide data for all known faults of the systems currently in service;
- There were no records on the competence of the designers and developers who created the original product which has been iteratively built upon over the years;
- No information was given to demonstrate the proposed operating profile and physical environments would be similar to the previous installations;
- The system itself used COTS components (with software) which would themselves need assessment to a similar degree; and,
- Protection functions within the system utilised software which had ‘always been used’ but there was no information on how frequently these protection functions had been called upon.

Making a PIU argument has the added drawback that it is often seen as self-certification of a component or system, with the burden of responsibility passing from the supplier to the end user who then needs to have confidence that under regulatory scrutiny, the body of evidence amassed would meet the constraints outlined in the standard; a decision which is always going to be open to subjective interpretation.

Identifying weaknesses in potential PIU substantiations early in a project and suitable mitigations, which could include re-engineering the function/system to current high integrity standards, would help to reduce this risk.

In Section 3.4 we discuss an approach to assessing and managing COTS components which could assist with structured identification of weaknesses.

2.6. Varying Safe State

Often in the industry sectors in which we have performed assessments, there has been a clearly defined desirable “safe state” that should be reached if a fault or mal-operation occurs. This is predominantly a ‘powered down, energy dissipated’ state. Occasionally, such as with reactor cooling system in the nuclear power industry, the safe state is energised or operational. In the naval domain however, the desired safe state may well change depending on what function the ship is currently performing (i.e. the mission state). A designer needs to therefore consider both the functional safety requirements and the potential mission criticality of systems when determining the integrity level required.

For package systems this can mean additional bespoke functionality and care is required to ensure this project specific modification provides the additional operational performance when required whilst also preventing system failure. We have seen varying examples of this functionality being clearly defined. An approach to address this and the wider topic of requirements management is considered in Section 3.2.

3. Strategies for Consideration

Through our collective IEC61508 assessment work and other systems engineering projects, we have reviewed or applied strategies and techniques which may be applicable to the naval domain to help meet the IEC61508 principles as required in NAN 06/2018 – Software Integrity.

3.1. Apportioning Reliability Across Systems Early in the Design

High level functional and performance requirements critical to mitigating key hazards are often known at an early stage of vessel design (e.g. concept design). To address the risk of not knowing the reliability requirements of a sub-system prior to its design beginning, high level requirements could be apportioned across all the sub-systems at the early concept stage. As the design matures the reliability requirements could be fine-tuned to optimise the overall functional requirement but often this is hardware driven and the software remains within the same SIL resulting in the same development and verification and validation requirements.

This would at least give an indicative integrity requirement for the subsystem and subsequently an initial constraint for a sub-system developer to aim for. The overall vessel designer would also have confidence that the significant project risk of the high level requirements not being met by the summation of all the sub-systems is being well managed. From a safety software perspective this gives software engineers early clarity on the likely integrity requirements and they can therefore put in place appropriate development processes before the system design begins.

3.2. Requirements Modelling

Frazer-Nash have been using a Model Based System Engineering (MBSE) approach to model the whole system requirements for a large and complex marine design project. It has proven useful in specifying complex relationships between seemingly contradictory requirements and could be applied successfully to understand the relationship between different mission states on the functional safety requirements for naval systems. Our experience is that maximum value is obtained from MBSE when it is applied early in the design process and then iteratively refined as the design matures. This could structure the process of allocating safety functions to a concept design which can then be apportioned to subsystems as they are developed; reducing the likelihood of over or under specifying the required SIL for any one component or subsystem.

Software tools originally developed for the automotive industry and now being adapted for general use, show promise in being able to automatically derive a quantitative analysis from a SysML model of functional safety requirements, both for hardware and software functionality.

Specifically from a IEC61508 point of view, the other major benefit from an MBSE approach is the ease of demonstrating compliance traceability throughout the software lifecycle process; high level functions are linked to the detailed design requirements, with unique identifiers being produced that can then be referenced to in the verification and validation stages of the lifecycle.

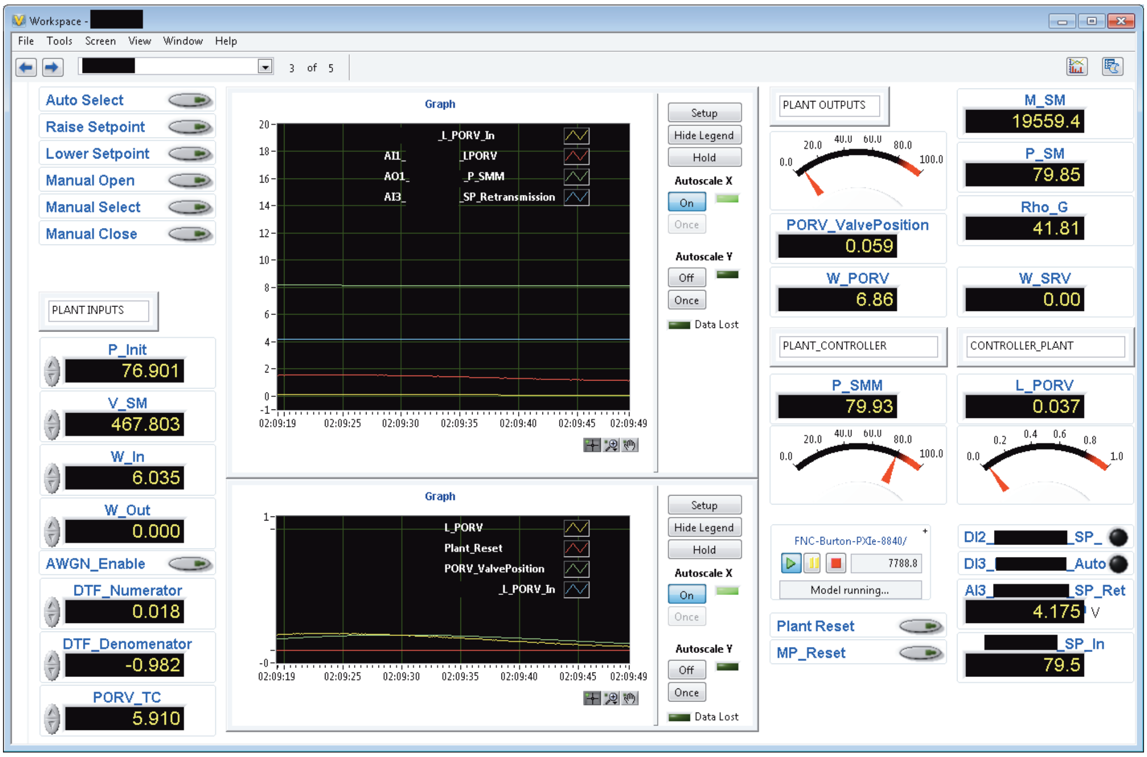
3.3. Simulation and X-in-the-Loop (XIL) Testing

As stated in section 2.3 the lack of integration testing before final 'on ship' installation exposes naval programmes to significant technical, schedule and cost risk. This can be mitigated by the use of simulation and/or a model-based design approach to ensure the designers of a system are cognisant of interfaces (i.e. connected peer systems) and constraints (i.e. its physical environment) throughout the development process. By developing or acquiring representative emulators and simulators, the system under development can be regularly tested as its design matures to identify potential issues. Commonly, requirement or interface definition problems are found during XIL testing that would otherwise only be found during integration tests on board the platform.

The most well-known variant of XIL testing is Hardware-in-the-Loop (HIL) testing. This involves deploying the actual control software on the hardware under test but then connecting it to a real time simulation environment which provides representative input/output signals, based on a simulation of the plant and other systems. A HIL approach has proven particularly powerful in testing fault cases and scenarios that are difficult

http://doi.org/10.24868/issn.2631-8741.2018.015

or dangerous to generate in the real world. An example user interface for a HIL rig designed to test a valve controller in a complex safety related plant is shown in Figure 1. This approach permitted multiple phases of automated testing, the logging of large amounts of data to compare the simulated environment and the controller response, and ultimately enabled a high confidence factor in the suitability of the control software for its intended use.



Workspace - [redacted]

[redacted] 3 of 5

PLANT INPUTS

P_Init 76.901

V_SM 467.803

W_In 6.035

W_Out 0.000

DTF_Numerator 0.018

DTF_Denomenator -0.982

PORV_TC 5.910

PLANT OUTPUTS

PORV_ValvePosition 0.059

W_PORV 6.86

M_SM 19559.4

P_SM 79.85

Rho_G 41.81

W_SRV 0.00

PLANT_CONTROLLER

P_SMM 79.93

CONTROLLER_PLANT

L_PORV 0.037

FNC-Burton-PXIe-8840/ 7788.8

Model running...

DI2_[redacted]_SP_

DI3_[redacted]_Auto

AI3_[redacted]_SP_Ret 4.175 V

[redacted]_SP_In 79.5

Figure 1. User interface for a HIL test of a valve controller

3.4. Device Assessment Process

The Office for Nuclear Regulation (ONR) have their core safety requirements outlined in the Safety Assessment Principles (SAPs) [Ref. 9]. Where a system is ‘significantly dependent’ on the development of software the SAPs require the assessment of ‘production excellence’ and ‘independent confidence building measures.’ To assess production excellence the industry developed an assessment tool called Emphasis which has roots in the tables within IEC61508-3 and the techniques outlined in IEC61508-7. The tool prompts an assessor, through a series of questions, to understand if the development of a software based device meets with the systematic capability aligned with the integrity claim being placed upon the function it is performing. For an independent confidence building measure the ONR give examples such as an independent review of the software and an independent review of the test programme. This can often lead to requiring additional device testing. Where gaps are identified in the production excellence assessments, ‘compensating measures’ can be performed to address any partial or full non-compliances.

This approach goes further than other industries which often rely on the manufacturer certification. An efficient solution in the naval domain could be a Ministry of Defence (MoD) managed register of assessed and approved devices to select from. This approach would enable the MoD to recognise the assessments performed across multiple projects in a centrally managed register and give efficiencies when these components are re-used on future projects. For example the assessments from projects led by one whole ship supplier could be used on another which may be led by a different supplier preventing these assessment from being repeated.

3.5. Field Data

Given experience of previous attempts to make PIU arguments (see section 2.5) and the difficulties associated in doing so, the naval industry and specifically the supply chain should consider steps that could be made to enable PIU claims to be more easily made in the future. For commercial reasons many suppliers to naval projects take a product line approach to system development, supplying subtle variants to individual programmes. Careful documentation could potentially allow data from all variants to be used to generate sufficient operational data to support future assessments, provided that it can be shown the dataset generated is accurate, all faults are recorded and the relevance to a specific product variant and its intended use can be shown.

As a side effect, more accurate operational information will allow naval operators to better evaluate OEM claims, inform future procurement decisions, supplier choice and through life management options.

4. Conclusions

There is a common understanding that the amount of automation, and associated software, in naval vessels will continue to increase and presents unique challenges such as varying safe state and space constraints; additionally with the rise of autonomous vessels the integrity requirements of the associated software systems will also rise. There is therefore a need to learn from previous projects and embed the principles and processes outlined in IEC61508 for future projects.

Six common areas of concern from our experience of assessing safety related systems for IEC61508 compliance have been raised within this paper and five possible strategies for consideration presented, to help reduce the project risk caused by these concerns.

Even for systems which do not fall within the key hazard category, following the software guidance within IEC61508-3 could help build confidence in the quality of the systems which, for a valuable asset such as a defence maritime vessel, is critical to successful operation. The structured processes in IEC61508-3 can provide a benchmark to help reduce the potential for systematic error and give a framework for producing well documented software aiding any future upgrade or modification requirements as well as ensuring traceability of requirements to demonstrate system performance.

The author's acknowledge a perception that complying with functional safety is an onerous process which carries significant cost and we regularly hear a request for 'IEC61508-lite' across many domains. The counter to this view, held to by the authors, is that following robust development processes on large and complex projects such as a naval vessel has significant benefits which will often more than compensate for the increased upfront work. Significant inefficiencies, and their associated costs, can be reduced over the project lifecycle as a more robust approach reduces the risk of interface issues and enables the complete system to show compliance against the requirements. Furthermore, IEC61508 recognises the need for the rigour of the process followed to be proportionate to the integrity required of the system. Ensuring those applying the IEC61508 principles are SQEP therefore is paramount to ensuring the process being applied does not become 'too onerous.'

5. Acknowledgements

The authors would like to thank Frazer-Nash Consultancy for the support to produce this paper.

6. References

1. IEC 61508 and IEC 61511 Assessments – some Lessons Learned, M H Lloyd, P Reeve
2. Lessons Learnt from Functional Safety Assessments, Mirek Generowicz, Rev. 0 10 May 2014
3. BS EN 61508: Functional safety of electrical/electronic/ programmable electronic safety-related systems
4. Naval Authority Notice 06/2018 – Software Integrity. Implementation 26/02/18.
5. JSP430 – Management of Ship Safety and Environmental Protection. Issue 5
6. DSA02-DMR-MOD Shipping Regulations
7. Defence Standard 00-55. Requirements for Safety of Programmable Elements in Defence Systems. Issue 4. 29 April 2016.
8. IET Code of Practice: Competence for Safety Related System Practitioners. First Published 2016.
9. Office for Nuclear Regulation. Safety Assessment Principles for Nuclear Facilities (SAPs). 2014 Edition 0.

7. Glossary of Terms

[1] The key hazard areas are identified in JSP430 Issue 5 [Ref. 5] as stability, structure, escape, evacuation and rescue, explosives, propulsion and manoeuvring systems, fire safety and aviation. This has now been superseded by the DSA02-DMR-MOD Shipping Regulations [Ref. 6] but the same hazards are listed as requiring certification.