Empirical Evaluation of the Impediments to an "As Desired" Model of Software Safety Assurance

Matt Osborne[1], Mark Nicholson[2], Richard Hawkins[3]

Department of Computer Science

University of York

Abstract *Recognised good practice for software safety assurance in safety-critical domains has been established in standards, and other guidance and literature. Anecdotally, this knowledge is not being transferred into the state of practice. Potentially, there are many reasons for this disparity, and it is likely that socio-technical impediments will have a large impact. To investigate the mismatch between theory and practice for software safety assurance, we have embarked on an empirical study. This study requires that we model safety assurance work as desired (the state of the art), the work as described (Open Standards and organisational processes) and the work as done (what engineers actually do in practice). Based on the results of this study, we will make recommendations to overcome the identified impediments to the adoption of good practice for software safety assurance. In this paper we discuss what constitutes work as desired, and address in detail the second aspect of this empirical study by describing an as-described model that is based on analysis of selected open standards. We also briefly discuss methods that will be adopted to empirically evaluate industrial work as described, and work as done.*

[1] matthew.osborne@york.ac.uk

[2] mark.nicholson@york.ac.uk

[3] richard.hawkins@york.ac.uk

1 Introduction –Good Practice for Software Safety Assurance

Currently, 'recognised good practice' for achieving software safety assurance is expressed through functional safety lifecycles denoted in various Open Standards (such as (BSI, 2010), (SAE Aerospace, 2010), (RTCA, 2011) and (SAE Aerospace, 1996)) and other guidance literature ((UK Ministry of Defence, 2016), (UK Ministry of Defence, 2017) and (Hawkins, Habli and Kelly, 2013) for example).

Anecdotal evidence suggests that this 'recognised good practice' is not being widely followed in industry, and there are potentially many reasons for this – including socio-technical impediments, perhaps.

To investigate this, we have embarked on a series of analytical and empirical data gathering exercises involving interviews and observations of software safety practitioners. This will help us to identify and characterise the impediments that are preventing the assurance and careful management of software safety requirements as they evolve through the lifecycle of a system.

Through our empirical studies we will model the 3 stages to ascertain where (and why) disagreements exist between work:

- As Desired
- As Described (both as described in Open Standards, and the practice interpreted by industry and mandated in internal processes)
- As Done (the processes implemented within organisations).

The identified differences and impediments will enable us to assess whether, and how, software assurance guidance and/or practices need to change. We aim to ensure our proposed approach will help mitigate these impediments in a way that is compatible with the 'work as done' profile we have identified in our empirical studies.

In this paper we focus on the first part of our study by describing the work we have done to create a model of software safety assurance work as desired, and an analysis of relevant open standards.

The rest of the paper is structured as follows. Section 2 discusses the development of the work as desired model. Section 3 describes the empirical research that will be undertaken to analyse work as described. Section 4 discusses the empirical research that will be undertaken to assess work as done. Sections 5 and 6 discuss the threats to empirical data validity and the differing outcomes of each stage of the empirical research, respectively. In Section 7 we draw conclusions and outline the next steps.

2 Work as Desired – Expressing and Representing Best Practice

The initial plan for the first part of the empirical research was to model a representation of best practice using the activities required of 3 selected Open Standards ((BSI, 2010), (SAE Aerospace, 2010), and (BSI, 2019)). Taking each standard in turn the lifecycles were to be modelled:

- As depicted pictorially (typically a flow chart or 'V'model)
- As conveyed by the accompanying and supporting text.

The decision to undertake this in two distinct steps was predicated on our observation that the main text of the standards do not necessarily match the simplistic overview portrayed by the visual representation – and often contradicts and/or confuses it (which may point to an impediment).

Each standard would be compared to the wider state of academic literature, enabling us to assess any identified shortfalls, vagaries, or disagreements – with each subsequent standard improving on the shortfalls and mitigating the vagaries of the last.

On completion, further recourse to academic literature and personal experience would be made to eliminate any residual shortfalls and clarify any remaining vagaries – before offering this as-desired model for review and feedback as a lifecycle representation that may be considered a referenceable benchmark in support of the Empirical Research.

As our research will demonstrate (with some of our observations against the first standard we have assessed discussed below), the processes and practices required of Open Standards cannot defensibly be asserted to constitute best practice, and an alternative strategy had to be adopted.

An effective functional safety lifecycle must clearly map all the activities to be carried out, at which stages of the overall lifecycle, along with the corresponding:

- Input requirements (and appropriate formats)
- Timing imperatives
- Independence requirements
- Methods that may be employed to fulfill the activity
- People required to undertake the tasks
- Resources required in support of the tasks
- Outputs from the activities (and appropriate formats).

To establish a more pragmatic, effective, and realistic lifecycle that is agnostic of sector and application, we must first look to the 4+1 Principles as the benchmark of best practice for software safety assurance. We assert that the as desired representation of software safety assurance must be predicated on the 4+1 Principles

as they are "constant across domains and across projects, and can be regarded as the immutable core of any software safety justification" (Hawkins, Habli and Kelly, 2013), (Hawkins and Kelly, 2013).

In our research, the 4+1 Principles are considered in turn, having defined the claims that any lifecycle must be capable of making to achieve compliance with them (Hawkins, Habli and Kelly, 2013), (Hawkins and Kelly, 2013).

3. Work as Described - Study 1

This phase of the study comprises 2 distinct research areas:

- That described by Open Standards
- That described by Companies' Internal Processes.

3.1 Open Standards

In this research area we are looking at internationally recognised Open Standards that have been created and endorsed by international committees of appointed subject matter experts.

The term 'open' refers to the fact that there is no Intellectual Property Rights (IPR) precluding paid access to them. 'Closed' standards refer to those created and used by an organisation – who restrict access to their employees; protecting the invested IPR held.

Below we provide a justification for the Open Standards used in this study:

- **ARP 4754A** (SAE Aerospace, 2010): Representative of Civil Aerospace system safety recommended practice – and an acceptable means of compliance for certification by regulatory bodies (e.g. Federal Aviation Administration (FAA) and European Organization for Civil Aviation Equipment (EUROCAE)). Its use is not merely confined to the aerospace sector, however. Despite representing a system safety process, ARP 4754A is relevant to this research as it identifies the contribution made to system safety by software. It is therefore in accord with Principle 1 of the 4+1 Principles (Hawkins, Habli and Kelly, 2013) which are reflected in UK Defence Standard 00-055 (UK Ministry of Defence, 2016).
- **BS EN 61508 (BSI, 2010)**: Unified and generic standard for all functional safety lifecycle activities, that is designed to facilitate development of application- and sector-specific standards in a parent/child relationship. Although a functional safety standard, BS/EN 61508 considers the functional safety of electrical/electronic/programmable electronic safety-related systems.
- **ISO/TC 215 N 2750 – IEC/CD 62304.3** (BSI, 2019): A (DRAFT) software lifecycle process for the development of 'Health Software'.

The first of these Open Standards – ARP 4754A, is discussed below.

3.1.1 Illustrative Example: ARP 4754A

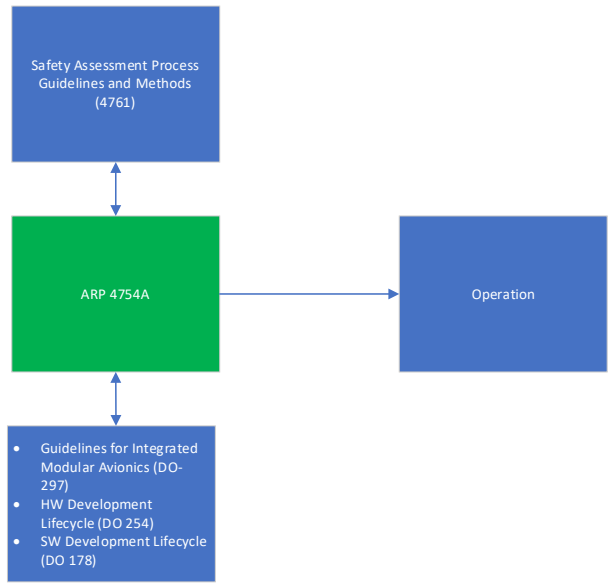

Fig. 1. Illustration of ARP 4754A's Relationships with Other Documents

Figure 1 illustrates how the ARP has relationships with other documents in a suite of documentary artefacts that combine to create lifecycle processes for the safety-assured development of systems, electronic hardware, and software for use in civil aviation. The development of aircraft and systems is expressed in ARP 4754A in the form of a 'traditional V-model' lifecycle that aims to show the interaction between safety and development processes – as shown in Figure 2.

This V-model process is defined as being employed "in an iterative and concurrent fashion using both top-down and bottom up strategies"(SAE Aerospace, 2010). It focuses "on the top-down aspect since it provides the necessary links between aircraft safety and system development" (ibid).

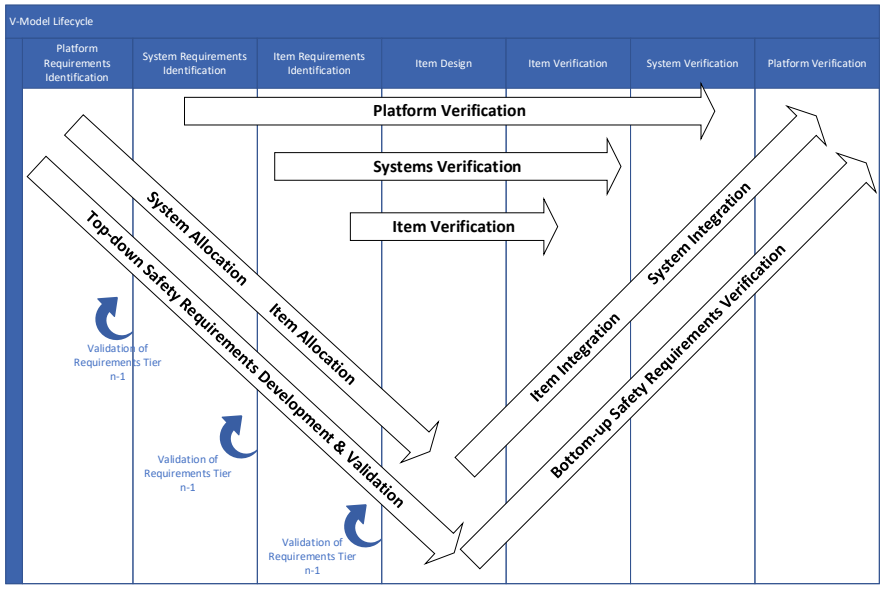

Fig. 2. Typical V-Model Lifecycle

3.2 Graphical Representation of As-Described Processes

To represent a model of practice that is pan-domain, we have selected the Functional Resonance Analysis Method (FRAM). We use this with minimal adaptations to suit our use (Hollangel, 2012), and have selected it because of its simplicity, and inherent ability to represent the pre-requisite conditions (aspects) for each function (see Figure 3). The aspects employed in the original FRAM are:

- Input,
- Output,
- Precondition,
- Resource,
- Time and
- Control.

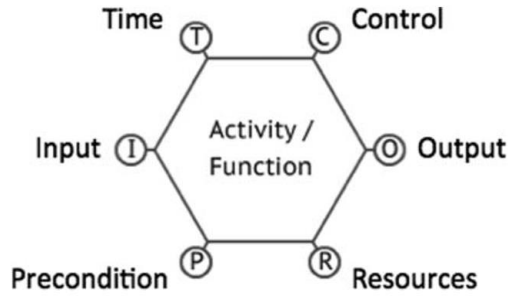

Fig. 3. FRAM Notation

In our model we retain the majority of the FRAM ontology, but have modified some of the syntax, adopted a 'layered modelling approach', and introduced some new concepts to adapt and enhance the utility of FRAM for our specific purpose. Our adapted version of FRAM removes the aspect 'precondition' but adds 'Resource' as an aspect.

FRAM's functions are further decomposed into sets of sub-functions – as shown in the example instantiation (Furniss, Curzon and Blandford, 2016) in Figure 4. We use the same modelling principle for activities, as it enables the linking of activities together to define processes throughout the development lifecycle.

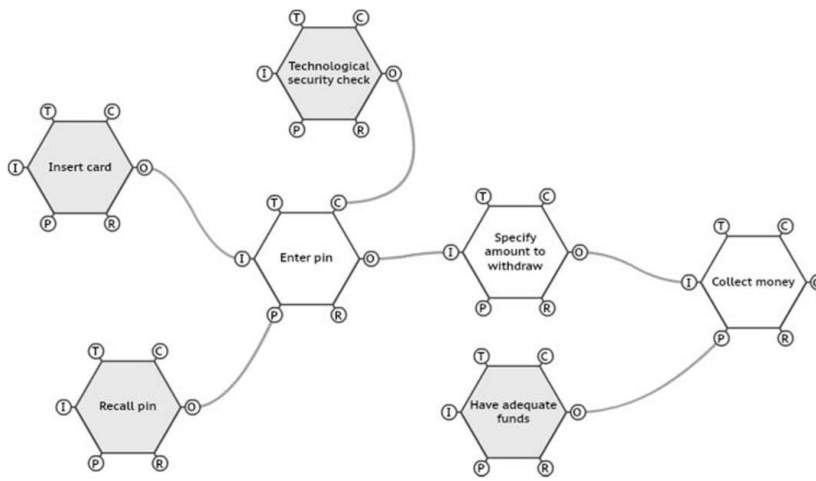

Fig. 4. FRAM Instantiation Example

3.2.1 Artefacts

As the lifecycle model progresses, a level of abstraction is reached at which the inputs to and/or outputs from an activity will not require further process consideration. In this case artefacts are employed as outputs from or inputs to an activity.

An artefact represents a deliverable or item that supports or constrains an activity or is the result of an activity. As such, it is the lowest level of abstraction that our representation will model.

To ensure we adequately model the required aspects of all instantiations of an artefact, we model the following:

- **Time** - the temporal relationship between the artefact and the activity (when the artefact should be available to facilitate the activity),
- **Quality criteria** – Quality criteria for artefacts are numerous (completeness, consistency, independence, method etc., see for example (Object Management Group (OMG), 2018). We use a single quality aspect that facilitates multiple instances of differing criteria for differing artefacts.
- **Existence** (positive/negative) – Artefacts may be an input or an output of an activity/sub-activity. Since this is defined by the input/output aspects of the activities themselves, we only need to represent the existence of the artefact.

For artefacts, we do not need to model 'Resource' as resources are expended by the supported activity, and an artefact is the lowest abstraction that we consider in our model.

As only 3 aspects of an artefact are required, a simple triangle is used – as shown in Figure 5 below.

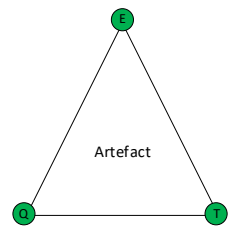

Fig. 5. Artefact Symbol

3.2.2 Colour Coding

In the 'As Described' models, a comparison is firstly made between Open Standards and the wider state of academic and further guidance literature. In this context, lines

that link activities and other entities are colour-coded to represent the completeness and efficacy of a link. The colour-coding of lines denotes the strength of the link itself (as shown in Figure 6):

- **GREEN:** A link between the activities or sub-activities is established by the lifecycle under consideration, which is considered sufficient compared to the state of the literature.
- **AMBER:** A link between the activities or sub-activities is inferred by the lifecycle or the text describing the lifecycle under consideration, but not positively stated. Or the links required by the lifecycle under consideration are not considered to be sufficient when compared to the state of the literature.
- **RED:** No link is established between the activities or sub-activities; yet one would expect to be described from what is established by the state of the literature. In the interest of maintaining a readable model, we rely on the colour-coding of the aspects to signify the lack of an established link.

Fig. 6. Colour Coding

Colour coding of the linking lines does not offer a solution to representing optionality and multiplicity, however. One notation that does offer this is Goal Structuring Notation (The Assurance Case Working Group, 2018), using the extensions shown in Figure 7. Where optionality or multiplicity is to be denoted, we have incorporated the same methodology in our linking lines.

n	A solid ball is the symbol for many (meaning zero or more). The label next to the ball indicates the cardinality of the relationship.
	A hollow ball indicates 'optional' (meaning zero or one).

Fig. 7. Multiplicity and Optionality Extensions in GSN

As shown in Figure 6, colour coding is also used for aspects of activities. Colour-coding of the activity aspects indicates the strength of the aspect:

- **GREEN:** The quality or existence of the aspect is sufficient as compared to the state of literature regarding the characteristics required, OR no further consideration of this aspect is required for this level of abstraction.

- **AMBER:** Some consideration of quality or existence of the aspect is made in the process being modelled, but there are perceived gaps as compared to the state of the literature regarding the characteristics required.
- **RED:** No consideration has been given to the required quality or existence of the aspect; yet one would expect this when compared to the state of the literature, benchmark model, or industrial practice considering the characteristics required.

After comparisons have been made between the Open Standards and the state of academic and further guidance literature, the standards will be assessed against the 'As Desired' model to assess the levels of compliance. Colour coding will again be used, and we will outline the scheme in future publications.

During the modelling of the 'As Desired' representation it emerged that Open Standards often infer an activity or artefact, or the reader must assume an input or artefact for an activity to take place. An example of this (from ARP 4754A) is the note that "additional assumptions" will emerge. Although not explicitly clear from the text, this requires the creation of an artefact (introduced below) entitled 'System Assumptions'. As this is only inferred, the artefact and associated aspects are all colour-coded red – as shown in Figure 8.

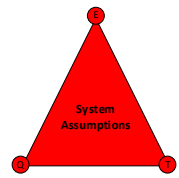

Fig. 8. Using Colour to Denote Assumed Existence of an Element

3.2.3 Graphical Representation of the ARP 4754A Process

Using our modified FRAM, we created a representation of the ARP 4754A lifecycle model. This is presented in Figure 9 at the end of the paper and is included only to highlight the complexity of the model[4].

A potential pitfall for the unwary reader of ARP 4754A would be to simply infer that the simplistic view of the V-lifecycle (in Figure 2) implies a chronological (although iterative and recursive) series of steps that are sequential according to the level of granularity or decomposition of design. The text of the ARP 45754A must

[4] The full model will be provided in a readable format in future publications.

be read and thoroughly understood to meet the intent of the ARP – the depicted lifecycle representations being but a 'readable' precis of the process.

The mismatch between a simplistic view (Figure 2) and the mapped process borne out by the text of the ARP (Figure 9) serves as a stark indicator of the differences between the represented and the documented lifecycles within ARP 4754A itself.

Documenting and representing 'recognised good practice' in open standards is a significant challenge; it must be representative of the necessary steps, yet be portrayed in a manner that makes achievement possible and logical with the right organisational and commercial structure. Open standards such as ARP 4754A are therefore prone to different interpretations by different users (e.g. readers, implementers, regulators). Varying levels of interpretation are not unique to the suite of ARP 4754A artefacts. We are also aware (through experience and recourse to literature) that there exists no clear pan-industry agreement on what constitutes recognised best practice for software safety assurance. This may be a contributor to the variable levels of interpretation and implementation evident in practice (and we aim to ascertain this, and other impediments through our empirical studies).

3.2.4 Issues with ARP 4754A as a Basis for the As Desired Model

Modelling ARP 4754A has identified shortfalls and vagaries in using it as the basis of the As Desired model. In this section we discuss a number of these issues under the headings of 'Model Observations' (those arising from our modelling of the ARP 4754A lifecycle), 'Guidance Observations' (comments made on specific aspects of ARP 4754 as noted in the academic literature), and 'Characterisation' (a critique of this style of 'V-model' lifecycle taken from academic literature).

3.2.4.1 Model Observations

- **Insufficient Quality Attributes:** Recognised good practice must be portrayed at a level of granularity that enables those charged with its implementation to ensure the activities are carried out, and the artefacts are created and delivered to the right quality. This should describe for example who needs to know what, by when, in what format, to what quality, and…to what end etc.). ARP 4754A does not provide this level of information. More significantly, the ARP offers no consideration of the resource(s)[5] consumed by activities or the quality criteria of such resources (e.g. qualifications, experience, and authority).

[5] Resources in this context refers specifically to personnel.

- **Assumed Artefacts and Activities:** A recurring theme across our model of ARP 4754A is the existence of activities and artefacts that must exist for the guidance to be implemented, but which are not explicitly discussed in the ARP.
- **Optionality/Multiplicity:** Although sub-activities may be linked to activities (such as a System FHA or CCA carried out as a contributor to requirements elicitation activities), the ARP offers no guidance on how many should be carried out.
- **Interpretation:** It is not explicitly clear whether the link between the activities Item Design and Item Hardware Design, Item Software Design and Item Mechanical Design is that of an overarching activity that produces 3 artefacts; an activity that enables 3 sub-activities; or 3 supporting sub-activities that contribute to a parent activity. It is also of note that there are no discussions on Item Mechanical Design in any documents within this suite of artefacts.

3.2.4.2 Guidance Observations

Some of the more significant observations of the guidance are:

- Stipulation of what constitutes 'recommended practice' is avoided. This offers what can only be described as 'helpful advice', without making any stipulations on whether or how it would be beneficial to follow it. This is done under the explicit assumption that a regulator or certifying body will agree the analyses, techniques, and practices with the developer. It should be noted that in many cases, a list of recommendations is offered with a caveat that they may not, in fact, constitute best practice. If accepted as an acceptable means of compliance, the terminology of the ARP swap from 'could' and 'may' to 'shall' and 'should'. The caveat that it may not constitute best practice hints at the existence of more suitable practice, however.
- It highlights the need for "extra rigor" [sic] when interfaces span organisational or contractual boundaries, but offers no guidance on how such commercial, communication, legal, and contractual complexities can be managed. Such shortcomings in open standards are highlighted, with guidance on how they can be mitigated by Menon (Menon, 2010).
- Section 5.3.2 notes that certain safety assessments will derive safety requirements, but again offers only vague guidance on what types of safety assessments are required, and the measures of performance or constraints that can be allocated against such requirements. It offers no assertions as to how these requirements should be managed as they evolve. Nor does it describe what constitutes a set of reasonable assessment techniques (pertinent to the level of design abstraction or stage in the safety lifecycle) that may be undertaken to elicit such safety requirements.
- Section 5.4 of the ARP considers the validation of requirements and presents a list of generic considerations that "may be helpful", leaving the specific format

of requirements validation to the developer. This high-level guidance can only be effective and robust when relying on a certification authority or regulator to endorse the sub-set of developmental activities. Otherwise, the vagaries of the standard are left to the developer to mitigate to a commensurate level of assurance, relying instead on sister publications such as ARP 4761.

- There is also a lack of consideration concerning the entry and exit points in the lifecycle for pre-existing software (or hardware elements).

3.2.4.3 Characterisation

One of the key aspects of the ARP process is the V-Lifecyle model employed. Models of this type have been criticised, because:

- A sequential, hierarchical lifecycle that guides the reader through a strict sequencing of requirement elicitation activities is not supported by the wider literature, as software safety requirements are considered at a far higher level of abstraction and earlier phase of the design lifecycle than is suggested by such a lifecycle. Furthermore, in complex socio-technical systems, emergent properties (and therefore hazards) emerge from a complex web of interdependencies that span systems, sub-systems, components, and the environment (Stålhane, 2013). Such iterative, yet chronological models are insufficient for mitigating and managing such complexity.
- In 2001, McDermid and Pumfrey observed that some software projects that have been developed to "certification standards" were, in essence, "developed 3 times" with the rework due to "late discovery of requirements or design flaws" (McDermid and Pumfrey, 2001). They further highlighted that development standards assume a lifecycle for "completely new systems, and generally ignore the change and development of existing systems"; and also that processes such as ARP 4761 would be enhanced by mandating an extension to the classic functional failure analysis to the software level (through techniques such as a Software HAZOP) [ibid].
- As with the majority of 'software safety standards', no consideration is given to the timing constraints or requirements of activities that derive artefacts; nor to any contractual principles (between acquirer and supplier) or limitations thereof (Hawkins et al., 2013). Vilela et al (Vilela et al., 2017) note further that "safety standards…do not explicit [sic] highlight which information should be specified early in the development process". Examples of such information may include human interaction with the software, and integrity constraints (which may also inform an organisation's make/buy decision at the simplest level).
- Boehm urged a re-consideration of development lifecycles as far back as 1988 (Boehm, 1988), warning that "many software projects…have come to grief because they pursued their various development and evolution in the wrong order". A more recent Systematic Literature Review (Vilela et al., 2017), highlighted the importance of identifying (software safety) requirements as early

as possible in the lifecycle in order to prevent the propagation of safety issues through subsequent phases of development. This would also reduce costs significantly – by addressing the issue when it is cheapest to do so.

3.3 Company Processes

To evaluate 'Work as Described' as represented in companies' internal processes, we will be engaging with a number of organisations that develop software in safety-related applications. To represent this phase of 'Work as Described' we will carry out a desktop review of the organisation's lifecycle artefacts and model the process using our adapted FRAM – and use the same comparisons and colour schemes as for our study of Open Standards.

This desktop exercise will then be used to facilitate follow-up interviews with representatives of the organisation whose processes are under analysis to identify the reasons for disagreements with the wider state of academic literature, and any non-compliances with the 'As Desired' model.

4. Work as Done

Having established 'Work as Desired' and 'Work as Described' we plan to determine and compare 'Work as Done'.

4.1 Study 2 - Organisation specific Work as Done

The aim of the second empirical study is to establish the process as employed 'in practice' by those charged with implementing a company's software safety lifecycle process.

This 'Work as Done' phase of the study involves an open interview with a representative of each organisation (software safety engineer or software engineer with functional safety responsibilities).

To ensure the description of work done provided by the interviewee is not influenced by the models created in the first two stages of the research, the interview will be conducted from a single initiating request:

"*Please describe how safety-related software is developed in your organisation*".

This will enable a comparison to be made between the model of work as described, and the model of work as done. This comparison then facilitates follow-up interviews with representatives of the company whose processes are under analysis to identify:

- The reasons for limited areas of limited agreement
- The reasons why there are areas of no agreement
- Positive or negative reinforcement of the validity of the 'As Described' model (when used as a means of conforming with an Open Standard),
- Any impediments behind the areas of limited or no agreement (assuming a positive validity of the 'As Described' model as expressed by Open Standards)
- Whether any areas of limited/no agreement contribute to meeting any shortfalls with a company process and that of an 'As Desired' model.

It is entirely plausible that different domains (Aerospace, Maritime, Medical, Automotive etc.) may have different 'As Described' and 'As Done' factors or peculiarities. These will be investigated and revealed as we continue our empirical studies, considering the following:

- Whether the factors are unique to the organisation or domain under analysis
- Where domain-exclusive factors exist, whether the factors could be used to enhance a pan-domain model of best practice, or
- Whether entry and exit points for domain-specific factors need to be created in a pan-domain model of best practice.

5 Data Validity

The threats to validity of this empirical study concern that of invalid data. This may be caused by commercially or reputationally sensitive information contained in an organisation's lifecycle description, constraints placed on interviewees, or their own social biases influencing their responses. We aim to mitigate this through carrying out a pilot study. This will enable us to amend and re-test the question structure or alter the discipline or seniority of the respondents before re-testing as appropriate.

Care will also be taken with regards to the ethical construct of the interviews and a clear question structure that avoids the use of leading questions.

6. Discussion

As our empirical studies move from modelling the 'As Desired' software safety assurance process to the 'As Described' phase, the 'As Described' model may be:

- Equivalent to our ‘As Desired’ representation of best practice
- Different to our ‘As Desired’ representation of best practice.

Equivalency may point to the fact that the impediments to achieving software safety assurance do not manifest from the description or representation of the practice itself but may point to impediments to implementation. Study 2 will investigate this through the empirical observations of work in practice.

Should the ‘As Described’ of a company’s model improve on our representation of that expressed in Open Standards, or our representation of best practice in the ‘As Desired’ model, this may suggest that the practices required of Open Standards or an as-desired model themselves may be a root cause of impediments in practice, or perhaps that the state of practice is implicitly or explicitly aware of the shortcomings of such standards and has evolved in isolation of the standards. In this case we may improve our ‘As Desired’ representation and/or use this as a mechanism to suggest amendments to the practices extolled in Open Standards.

An outcome that suggests the ‘As Described’ practice of a company is worse than our ‘As Desired’ model, or that of an Open Standard (against which the company process has been designed to comply with) may indicate that impediments manifest in the interpretation of Open Standards into organisation-described processes. Study 1 will investigate this through targeted follow-up interviews with the organisation.

As our empirical studies move from modelling ‘As Described’ processes to the modelling of the ‘As Done’ phase, the ‘As Done’ model may be:

- Equivalent to the ‘As Described’ representation of organisational practice
- Different to the ‘As Described’ representation of organisational practice.

It may also be:

- Equivalent to the ‘As Desired’ representation of organisational practice
- Different to the ‘As Desired’ representation of organisational practice.

Assuming that there is at least equivalency of the organisation’s processes with our model of best practice, equivalency between ‘As Described’ and ‘As Done’ will mean that the ‘As Described’ process is being fully implemented. However, follow-up interviews as part of Study 2 will identify difficulties in implementing the process, and the existence of these may suggest issues with the company practice itself.

Through follow-up questions and/or further interviews as part of Study 2, we will aim to identify, characterise, and suggest mitigations to any impediments or difficulties.

Should the 'As Done' models improve on the 'As Described' representation of organisational practice, this may suggest that (assuming equivalency of the 'As Described' and 'As Desired' models) those charged with implementing the organisation's processes are aware of the limitations, inefficiencies, inaccuracies, or unrealistic expectations of their organisation's processes and have adopted factors to compensate. Through targeted follow-up questions as part of Study 2, we will aim to identify any impediments or difficulties that have led to a circumvention of process; and characterise and suggest mitigations accordingly.

By asking operatives what the impediments or difficulties are within the process, we may identify instances where those charged with carrying out the 'As Described' processes carry out processes that they know are inadequate (referred to by Dekker as instances of 'Malicious Compliance' (Dekker et al., 2017)). We may also identify examples where activities add no value (from a safety perspective), and perhaps other activities would be more useful with finite resources. Such instances may point to the inadequacy of practice within an organisation and will trigger follow-up interviews with the organisation (as part of Study 2) to identify, characterise, and mitigate the impediments or difficulties with the organisational process(es).

Some aspects of the 'As Desired', 'As Described', and 'As Done' software safety assurance processes may differ across domains. Cognisant of this we have chosen our Open Standards and selected our organisations from different domains to explore and characterise any differences with the aim of establishing a unified model of best practice.

7. Conclusions

Our empirical studies will develop a model of recognised best practice that is capable of pan-domain implementation.

Through professional experience and recourse to academic literature, we are aware that there may be socio-technical impediments to the transfer of knowledge between extant recognised good practice, and/or the state of literature, and industrial practice at large.

By modelling:

- Work as Desired (predicated on the 4+1 Principles)
- Work as Described (as represented in Open Standards and industrial processes)
- Work as Done (work as implemented by those charged with implementing industrial processes)

…we can identify, characterise, and mitigate shortfalls and vagaries, and suggest how the impediments at their root cause can be eliminated or mitigated.

This form of ethnographic study is in line with the recently-published manifesto to reality-based safety science (Rae et al., 2020) which we pledge to support and uphold.

We now repeat the modelling and characterisation of ARP 4754A with BS EN 61508 to enhance our representation and knowledge of documented extant recognised good practice, before repeating this process again with ISO/TC 215 N 2750 – IEC/CD 62304.3.

Having identified an initial list of organisations across the domains who have kindly and generously pledged their support, we progress our empirical studies by commencing the reviews of their processes – followed up by targeted interviews with those charged with their implementation.

Disclaimer

This paper is an updated version of one presented at the Safety Critical Symposium and Conference in February 2021:

https://scsc.uk/rp161.20:1

References

Boehm, B. (1988) 'A Spiral Model of Software Development and Enhancement', *IEEE Computer*. Available at: http://www.dimap.ufrn.br/~jair/ES/artigos/SpiralModelBoehm.pdf.

BSI (2010) 'Functional safety of electrical / electronic / programmable electronic safety related systems Parts 1-7'.

BSI (2019) 'DRAFT INTERNATIONAL STANDARD: Health Software - Software Life Cycle Processes', (0.3).

Charmaz, K. and Bryant, A. (2015) 'Grounded Theory', *The Blackwell Encyclopedia of Sociology*. John Wiley & Sons. doi: 10.1002/978405165518.wbeosg070.pub2.

Dekker, S. *et al.* (2017) 'HindSight25'.

Furniss, D., Curzon, P. and Blandford, A. (2016) 'Using FRAM beyond safety: a case study to explore how sociotechnical systems can flourish or stall', *Theoretical Issues in Ergonomics Science*. Taylor & Francis, 17(5–6), pp. 507–532. doi: 10.1080/1463922X.2016.1155238.

Hawkins, R. *et al.* (2013) 'Assurance cases and prescriptive software safety certification: A comparative study', *Safety Science*. Elsevier Ltd, 59, pp. 55–71. doi: 10.1016/j.ssci.2013.04.007.

Hawkins, R. D., Habli, I. and Kelly, T. (2013) 'The Principles of Software Safety Assurance', *International System Safety Conferene (ISSC)*. Available at: http://www-users.cs.york.ac.uk/~ihabli/Papers/2013Habli_ISSC.pdf.

Hawkins, R. and Kelly, T. (2013) 'A Software Safety Argument Pattern Catalogue', p. 32. Available at: http://www.cs.york.ac.uk/ftpdir/reports/2013/YCS/482/YCS-2013-482.pdf.
Hollangel, E. (2012) *FRAM: The Functional Resonance Analysis Method*, *FRAM: The Functional Resonance Analysis Method*. Farnham: Ashgate. doi: 10.1201/9781315255071.
McDermid, J. A. and Pumfrey, D. J. (2001) 'Software Safety: Why is there no Consensus?', *Proceeds of the International System Safety Conference (ISSC)*.
Menon, C. (2010) *Expressing Software Safety Requirements Across Organisational and Contractual Boundaries*. York. doi: SSEI-TR-000074.
Object Management Group (OMG) (2018) 'Structured Assurance Case Metamodel (SACM) specification V2.0', (March), p. 60. Available at: https://www.omg.org/spec/SACM/2.0/.
Rae, A. *et al.* (2020) 'A manifesto for Reality-based Safety Science', *Safety Science*. Elsevier, 126(January), p. 104654. doi: 10.1016/j.ssci.2020.104654.
RTCA (2011) *Software Considerations in Airborne Systems and Equipment Certification*, *RTCA DO-178C*. RTCA DO-178C.
SAE Aerospace (1996) *Aerospace Recommended Practice Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*, *Society*.
SAE Aerospace (2010) *Aerospace Recommended Practice (R) Guidelines for Development of Civil Aircraft and Systems*.
Stålhane, T. (2013) 'Safety standards and Scrum–A synopsis of three standards', *Nbl.Sintef.No*. Available at: https://nbl.sintef.no/upload/IKT/9013/Safety standards and Scrum_May2013.pdf.
The Assurance Case Working Group (2018) 'Goal Structuring Notation Community Standard'.
UK Ministry of Defence (2016) 'Defence Standard 00-055 Part 1 Requirements for Safety of Programmable Elements (PE) in Defence Systems Part 1 : Requirements and Guidance', (4).
UK Ministry of Defence (2017) 'Defence Standard 00-56, Issue 7: Safety Management Requirements for Defence Systems. Part 1: Requirements', (7).
Vilela, J. *et al.* (2017) 'Integration between requirements engineering and safety analysis: A systematic literature review', *Journal of Systems and Software*, 125, pp. 68–92. doi: 10.1016/j.jss.2016.11.031.

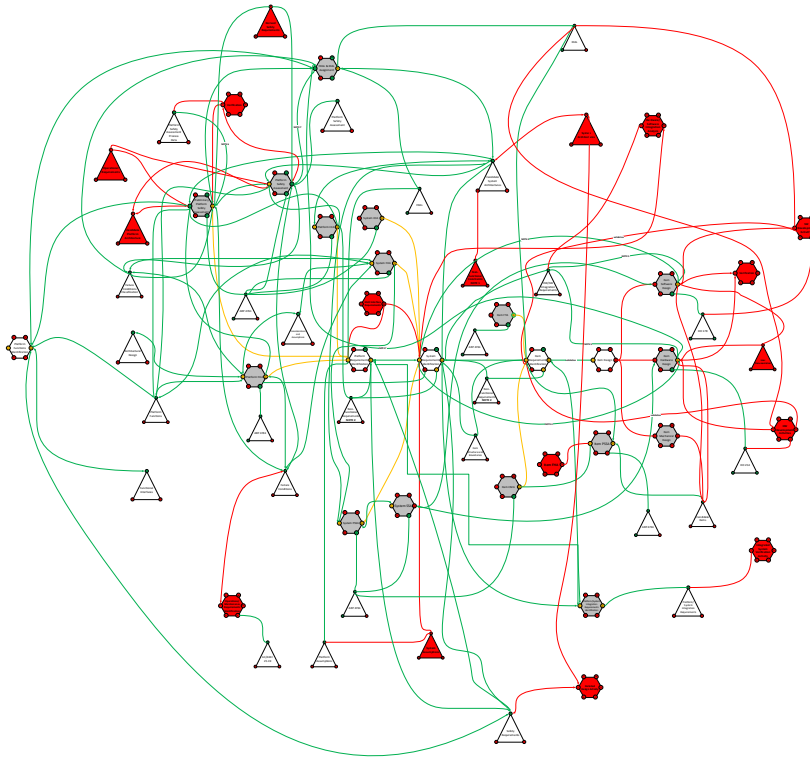

Fig. 9. An Illustration of the Complexity of the ARP 4754A Lifecycle Model using modified FRAM

……..

Title of Article: An "As Desired" Lifecycle Model for Software Safety Assurance

Name of Author(s): Matt Osborne, Mark Nicholson, Richard Hawkins

Name of Copyright Owner (if not author): University of York

Address of Copyright Owner:

Department of Computer Science
University of York
Deramore Lane
York
YO10 5GH

1. By signing this form, you (the copyright owner or author) agree to grant to the Safety Critical Systems Club (the publisher) the non-exclusive right to publish, distribute or broadcast your paper in printed or digital form. You agree that we may publish your paper in the SCSC Newsletter, SCSC books and make this available online throughout our website.
2. You promise that the paper is your original work. If it contains material which is someone else's copyright, you promise that you have obtained the unrestricted permission of the copyright owner and that the material is clearly identified and acknowledged in the text. You also promise that the paper does not, to the best of your knowledge, contain anything which is libellous, illegal or infringes anyone's copyright or other rights.
3. You assert your Moral Rights to be identified as the author. We promise that we will respect your rights as author and will make sure that your name is always closely associated with the paper.
4. Copyright remains yours and we will acknowledge this. However, you authorize the Safety Critical Systems Club, if we wish, to act on your behalf to defend your copyright if anyone should infringe it. You also retain the right to use part or all of your own paper in any way you wish.

…… ……..