SAFE AND SECURE: RE-ENGINEERING A SOFTWARE PROCESS SET FOR THE CHALLENGES OF THE 21ST CENTURY

K.R. Wallace

BAE Systems, Naval Ships, Broad Oak, Portsmouth, PO3 5PQ, UK
ken.r.wallace@baesystems.com

Keywords: Software, Process, Risk, Safety, Security.

Abstract

This paper discusses a risk-based approach to re-engineering a legacy software engineering process set in the context of a large-scale engineering enterprise responsible for the design and production of surface warships. The increasing integrity requirements on software deployed on modern naval platforms, principally in respect of safety and security, have been addressed through elicitation and analysis of key software integrity risks. The results of this analysis have been applied to assess the extent of mitigation of the identified risks provided in the legacy process set. This assessment provides a basis for the further development and improvement of the process set in respect of treatment of software integrity. More generally the approach provides a template for risk elicitation and analysis that can be extended to treat further categories of software-related risk such as acquisition/supply chain, legal and human factors.

1 Introduction

Legacy software engineering processes have the potential to present as many challenges and difficulties as the mature codebases that they have given rise to. This is particularly the case in instances where requirements for software integrity were either not considered during development of the processes themselves or the need to address integrity has been introduced retrospectively.

For system integrators, such as those in the defence sector, who deal with large numbers of suppliers of COTS products these problems can quickly become pronounced and costly. In many instances suppliers will have software engineering processes geared towards commercial imperatives rather than the attainment of stringent software integrity requirements.

The work reported in this paper addresses these issues by considering the re-engineering of a legacy software engineering process set through the introduction of risk-based approaches. While the approach is intended to identify and control any category of software engineering risk the well-defined and recognised nature of safety and security as sub-disciplines within software engineering has rendered these categories of risk suitable as early-adopters in order to develop and refine the approach prior to addressing additional categories of risk as part of the wider re-engineering activities.

1.1 Context

The BAE Systems - Maritime business comprises three business units: Maritime Services, Submarines and Naval Ships. Within the Maritime business, Naval Ships is recognised as the centre of specialism for software engineering. The Naval Ships business faces the engineering challenges of delivering complex surface warships such as the T45 Anti-Air Warfare destroyer and the Queen Elizabeth Class aircraft carrier. These platforms and their associated combat and mission systems are inherently and increasingly software-intensive systems. The software integrated onto these platforms is acquired either externally from an extensive supplier base or sourced internally from products developed by antecedent businesses; now part of Naval Ships following the merger of businesses that resulted in the formation of the Naval Ships business in 2012.

Within Naval Ships a population of several hundred software engineering staff work either on the development of new and existing software-based systems and products, or supporting the integration of externally sourced software onto platforms. Naval Ships also acts as a provider of software products and related engineering services to both the Maritime Services and Submarines business units.

1.2 Customer Requirements

Naval vessel requirements for survivability, resilience, and both defensive and offensive capabilities impose constraints and engineering challenges significantly beyond those associated with achieving the operational envelope of commercial shipping. These challenges apply as much to the software as to the physical platforms into which the software is integrated.

Recognising the increasing criticality of software to the effective operation of modern naval platforms the UK MoD and Defence Equipment and Support organisation (DE&S) have sought to address software integrity by building upon established safety approaches developed in the UK defence sector, most notably Defence Standard 00-56 and within the Maritime domain Joint Services Policy (JSP) 430: Ship Safety

Policy [1]. In furtherance of JSP430 and the security equivalent JSP440 a Naval Authority Notice (NAN) 09/2012 issued in 2012 to all "*MoD Procurement and Support Officers, Designers, Builders, Repairers, Class Societies and Surveyors*" is in force. This NAN is supported by an accompanying Software Integrity Policy and Guidance document which provides a basis for unifying the hitherto separate concerns in respect of software safety and software security. A revised version of this policy has recently been released for review.

These approaches mirror ongoing developments in the wider defence sector where a further revision of the currently obsolete Defence Standard 00-55, which addresses safety of software in defence equipment, is now being readied.

As the principal designer and builder of surface vessels for the Royal Navy the ability to respond effectively to such directives is clearly of fundamental importance to the Naval Ships business.

2 Governance of Software

2.1 Common Software Engineering Processes

Central to the Naval Ships approach to software engineering is the Common Software Engineering Process set (CSEP). This provides a process based framework for the conduct of software engineering supporting accreditation to standards such as ISO 9001 and TickIT. As such the CSEP constitutes a key engineering asset in the Naval Ship Business Management System (BMS).

Inherited by Naval Ships in 2012 the origins of this legacy process set can be traced to the Radar division (RSD) of GEC-Marconi. A response to the emergence of Software Capability Maturity Management in the 1990s, the resulting Organisation Standard Software Process (OSSP) developed by RSD provided the basis for the first release of the CSEP (R1) in 2006, active development of which continued until 2010 (R11) transitioning thereafter to maintenance until 2012.

Since being inherited by Naval Ships a programme of sustainment, in the first instance, and latterly development of the CSEP to meet the evolving demands of the business has been put into effect. This addresses the requirements of both the Naval Ships business and Maritime Services which continues to employ the CSEP for the governance of software engineering conducted within that business unit.

Key CSEP processes include:

- Software Requirements Engineering
- Software Estimation
- Software Configuration Management
- Peer Review
- Software Defect Prevention
- Software Measurement
- Software Maintenance
- Software Safety
- Software Security

The current release of CSEP (R14.2) includes a total of 38 processes. These processes are supported by additional artefacts namely:

Artefact	Number
Checklists	15
Forms	8
Guidelines	93
Templates	43

Table 1: Supporting artefacts for R14.2 of the CSEP.

Of the 38 processes in R14.2, 13 are legacy processes inherited from another antecedent business (Surface Ships Limited) and are retained in the CSEP in support of existing projects where migration to other equivalent processes is either impractical or inappropriate.

Due to the origins of the CSEP software development has historically been and currently remains the focus for the CSEP. Consequently treatment of activities such as acquisition (procurement) of software and risk management is achieved though guidelines rather than the application of processes.

2.2 Application of the CSEP

The diversity of projects involving software within Naval Ships renders a one size fits all philosophy to the application of the CSEP neither practical nor desirable. Moreover the extent of detail required in the CSEP to enable compliance to be a realistic outcome renders such a goal unachievable. Instead conformance to the requirements of the CSEP, by reference to the relevant process, guideline or other artefact is the intended outcome with projects being able to demonstrate such conformance through the provision of appropriate evidence. This approach allows for tailoring and detailed elaboration of CSEP processes at the project level with the plan for conformance being established in either the project Software Management or Development Plan.

2.3 Limitations of the current CSEP

Although the constituent processes inherently define a modular architecture the legacy CSEP has a number of limitations. In addition to the previously noted historic focus on software development key issues relate to the significant amount of material in the supporting artefacts, most notably the 93 guidelines. As many of these guidelines have been derived from historic project-based material, issues of both consistency of coverage and currency of content are increasingly arising. These issues, which are exacerbated by the implementation of both guidelines and processes in HTML, will only become more intractable over time. Currently the CSEP comprises approximately 1000 source files, equivalent to a codebase of approximately 115K SLOC.

Complexity of navigation through the CSEP, the look and feel, and limitations on the search facilities available within it also present significant technical problems with the current

CSEP implementation. Cumulatively these factors contribute to a user experience (UX) that compares less than favourably with contemporary digital environments that users will experience beyond the boundaries of their normal activities workings as professional software engineers. These aspects represent significant barriers to the effective application of the CSEP.

In regard of these limitations work undertaken by experienced independent usability experts concluded of the CSEP that:

"*The user population is large and diverse, holding different roles, working across a range of project lengths and based in different locations. It is challenging to design a single tool that meets the needs of all these diverse groups of users*"

Such observations are clearly of importance and must be accounted for when considering how future development of the CSEP is to be addressed.

Given the demands on software engineering within Naval Ships and the noted limitations of the current CSEP it is evident that maintenance activities alone will not address the increasingly diverse range of activities that the CSEP is required to encompass. Recognising this, through a process of consultation with relevant stakeholders, adoption of risk-based software engineering (RBSE) has been identified as offering the most suitable means of addressing the current challenges facing the CSEP.

As the focus for Naval Ships is on the design and production of naval vessels, rather than the subsequent operational phase of a platform, the initial scope of CSEP re-engineering has been aligned to the Concept – Manufacture phases of the MoD CADMID lifecycle, collectively denoted herein as the development phase. The In-service phase of the lifecycle will be considered at a later date.

3 Risk-Based Re-engineering of the CSEP

3.1 Risk-based Approaches To Software Engineering

Risk-based approaches to software engineering can be traced to early work by Boehm [2] and Charette [3]. Since then a slow but steady growth in reports in the literature regarding such approaches is apparent.

In exploring how RBSE might be applied to re-engineering of the CSEP existing proven approaches and supporting methods were the preferred route these being in themselves lower risk and hence potentially more cost-effective.

For a project of any significant size it was recognised that the risk profile will be both substantial and likely complex. Accordingly attempting to address all types of risks simultaneously even on smaller projects was considered impractical. Instead a number of prototype risk categories associated with software projects were identified by groups of subject matter experts (SMEs) and from these categories safety and security were identified as being test cases due to:

- the absolute importance of software integrity, noting that a compromise to security has the potential to compromise safety
- an explicit assumption that safety and security categories of risk were both relatively well defined
- the structured nature and similarity of existing CSEP treatment of both safety and security
- the observation that engineering safe and secure software in many respect equates to good software engineering, thereby offering the potential to leverage practice developed in dealing with these categories more widely in follow-on work.

It should be noted that in respect of software safety the current CSEP does not support software development above SIL2 (or equivalent) and the re-engineering of CSEP will maintain the exclusion on any software engineering exceeding this threshold.

To identify applicable techniques for RBSE and elicit individual software engineering risks literature reporting results derived from Evidence-Based Software Engineering (EBSE) investigations of RBSE was identified and scrutinised. First proposed by Kitchenham et. al. in 2004 [4] EBSE offers an adaptation of the methodologies of evidence based medicine for the needs of software engineering that [5]:

"aims to improve decision making related to software development and maintenance by integrating current best evidence with practical experience and human values".

EBSE emphasises the importance of systematic and extensive review of available primary (empirical) literature through the technique of systematic literature review (SLR) and associated Research Questions (RQs).

For current purposes the starting point was identification and review of SLRs addressing aspects of RBSE, augmented thereafter with limited consideration of primary studies if considered relevant. Attainment of completeness and consistency of review, which for EBSE research would be a high priority to ensure validity of findings, was less of a concern in the context of the present work.

3.2 Practical Questions

Analogous to the research questions employed by EBSE researchers the approach adopted was to define practical questions (**PQ**) to which answers were then sought. Specifically:

PQ1: Is there evidence of successful application of risk-based approaches in comparable contexts?

PQ2: What are the risks to be mitigated?

PQ3: Are there approaches in the CSEP to mitigate identified risks?

PQ4: Are applicable approaches in the CSEP good practice or better?

As PQ3 is a question that could only be answered by reference to the CSEP, no contribution to this question was sought from sources external to the Maritime business.

Taking the practical questions in turn:

PQ1: Is there evidence of successful application of risk-based approaches in comparable contexts?

Risk-based approaches to software engineering by standards bodies such as ISO and IEEE and advocacy of such by prominent practitioners and institutes, notably, the Software Engineering Institute have driven research into RBSE. Nevertheless reports on state-of-practice regarding real-world applications in industrial contexts were limited, appearing to have peaked in the early part of the last decade [6]. This surge in interest may be attributable to preceding work on risk management undertaken by the Software Engineering Institute at Carnegie Mellon University which was subsequently incorporated into the CMMI [7], [8], [9].

Successive papers by Kitchenham and collaborators [10], [11] reporting on the results of systematic literature reviews of systematic literature reviews (referred to as tertiary studies) identified no SLRs on RBSE within the respective periods, up to and including 2008, considered by these reviews. Three subsequent systematic reviews of RBSE were identified [12] [13], [14]. In addition a related systematic review addressing software process simulation aspects of RBSE was also considered [15]. Cumulatively inspection of the data sources in these secondary and tertiary studies was sufficient to engender confidence that RBSE was a valid approach to pursue. Evidence supporting the reliability of systematic reviews as a tool in software engineering [16] further contributed to this confidence.

In reviewing existing literature regarding RBSE, regardless of category (primary, secondary or tertiary) a deliberately wide interpretation of the term risk was adopted to maximise inclusivity of source in the first instance. In particular no distinction was made between risked-based approaches applied in Information Technology applications and those explicitly addressing software development or engineering; both categories being considered to be potentially of comparable utility in guiding the re-engineering approach.

PQ2: What are the risks to be mitigated?

Sources, both internal and external, mined for safety and security risks are summarised in Table 2. Keyword searches were constrained to simple search terms such as 'software AND risk'. Interviews with Subject Matter Experts (SMEs) and experienced practitioners took a semi-structured approach with an initial briefing combined with the use of Bow-tie diagrams [17] to promote discussion and assist interviewees to consider possible Threat-Risk-Consequence sequences. These sessions were followed thereafter by written submissions of the risks to allow for a level of reflection, synthesis and review prior to submission. SMEs were asked to prioritise submissions by concentrating on a list of top ten [18] risks in the first instance. The team undertaking the analysis included experienced software developers with significant and substantial knowledge of software engineering within the Maritime businesses and beyond, hence correlation and calibration of risks identified with those the business had experienced previously was possible. In practice the majority of sources yielded similar risks both in terms of the nature of the risk and importance. Variability of description and terminology was evident in some instances necessitating further analysis of the root cause of the cited risks in order to establish equivalence or otherwise, however, the need for this additional step was exceptional. As the use of a standard risk taxonomy is not a discipline universally practiced or one reported consistently in the literature no attempt was made to apply such an approach either in the interviews or to the subsequent written submissions.

A total of 23 key risks were elicited of which 22 were assessed as being within the scope of the current study. The one remaining risk while recognised as being of importance was sentenced as being exclusive to the In-service rather than development phase.

Source	Method
Learning From Experience	Interview
Lessons Learnt	Keyword search
Literature Review	Keyword search
Risk Register	Keyword search
Subject Matter Expert	Interview

Table 2: Sources used for risk identification

PQ3: Are there approaches in the CSEP to mitigate identified risks?

Following risk identification a mapping of risks to the CSEP was undertaken by answering the questions:

PQ3.1: Can the identified risks be mitigated by the CSEP?

PQ3.2: If so what means of mitigation of the risk is provided in the CSEP?

Mitigation in CSEP	Extent
None	9%
Partial	41%
Complete	50%

Table 3:Results for PQ3.1

The results for PQ3.1 are given in Table 3. Of the artefacts in the CSEP only processes were rated capable of providing a complete mitigation of the risk. Guidelines possessing less, if any, prescription were assessed as being more susceptible to error in comprehension, interpretation, and hence execution. Accordingly they were rated as being only a partial mitigation regardless of the amount of information and extent of any prescription individual guidelines did contain. Checklists and templates were not considered as offering any mitigation of a risk without a supporting process or guideline. This reflects a deliberately conservative approach to the assessment of artefacts, particularly in respect of guidelines. It also takes into account the previously noted lack of consistency of

content which will reduce the overall efficacy of guidelines in mitigating risks. Both processes and guidelines were rated as capable of providing partial mitigation. The 13 legacy processes retained in the CSEP in support of enduring but complete projects were excluded from the assessment exercise.

For the risks for which mitigation did exist in the CSEP the results for PQ3.2 are given in Table 4.

Artefact Type	Risk Mitigation
Guideline	40%
Process	60%

Table 4: Results for PQ3.2

PQ4: Are applicable approaches in the CSEP good practice or better?

A qualitative assessment of mitigations was undertaken to address PQ4. The results for this question are given in Table 5. This incorporated existing information on aspects of the CSEP known to require improvement, augmented by SME assessment of the mitigation available and comparison with comparable approaches reported in the literature. In particular for safety the Software Safety Assurance Principles proposed by Kelly and co-authors [19] were used as a test to assess whether the mitigation provided by CSEP qualified as good. No assessment of whether Good practice equates to Best practice was undertaken in the present work. Even for mitigations where the practice was assessed as 'Good' it is recognised that further improvement may, or may not, be required to achieve best practice assuming that such a benchmark exists.

Practice Rating	Assessment
For Improvement	32%
Good (or better)	68%

Table 5: Results for PQ4.

In answering the foregoing questions no consideration was given to human factors such as errors and violations which might affect the integrity of application of the CSEP, other than, as previously noted, to recognise that guidelines are more susceptible to error than processes.

3.3 Findings

To a very considerable extent the findings of the present work succeeded in confirming aspects of the CSEP known either for strength or weakness. For example, management of software acquisition is recognised currently to have insufficient treatment in the CSEP for reasons previously identified and this situation was reflected in the present findings. This is an aspect of the CSEP where work is already being undertaken to leverage practice [19] that has been developed by the T26 Global Combat Ship programme within Naval Ships.

Commonality of risk in safety and security was significant confirming that equivalence of treatment by means of addressing both as aspects of software integrity is, to a first approximation at least, a valid approach.

Describing risks, even for categories which might be considered as well recognised and bounded as safety and security presented some issues. This is perhaps unsurprising given that risk identification is the key activity that supports all subsequent analysis. As previously noted risk taxonomies do exist, however, as the present work did not make explicit use of such taxonomies in any detail whether their use would have assisted remains an open question. It should be noted, however, that as the present work made some use of literature regarding RBSE, indirect use of risk taxonomies as applied in primary studies is possible.

The use of systematic literature reviews as one starting point for risk identification permitted a wider perspective than that which could have been achieved through the use of internal sources alone. In practice there was considerable agreement across all sources as to key risks, even if the descriptions of the risk varied. In dealing with only key risks in safety and security it is recognised not only that the potential for divergence between the categories has been limited but that such limiting would become less appropriate the greater the extent of identification and elaboration of risks, particularly for those of a technical nature. This presents the prospect of the total number of risks identified increasing, possibly very substantially. The potential for RBSE to be overwhelmed by the number of risks is one of a number of barriers to RBSE reported by Odzaly et al. [20].

In the same study these authors reported other barriers some of which were also in evidence in the present work. The absolute imperative of safety and necessity for security within the defence sector are such that other key barriers identified in the study did not constitute any impediment to the present work. It is recognised, however, that such barriers do exist, particularly in environments where the need for consideration of safety or security are not as stringent or as well established.

As a final test of the validity of the findings an assessment of the extent of coverage of software integrity aspects for three projects with differing profiles for safety and security was undertaken. The results of that review correlated well with the present results thereby providing further confidence in the robustness of the findings.

4 Conclusions and the Way Ahead

Managing a legacy software engineering process set presents challenges not dissimilar to dealing with legacy codebases. Additional complexity arises, however, from the fact that many, rather than a single project, are dependent upon the process set. Increasing demands in respect of software integrity intensify the extent and nature of these challenges.

In responding to these challenges the present work has adopted a risk-based approach and applied it to key safety and security risks. This has facilitated an assessment of whether the legacy process set can address identified risks and allowed for an initial evaluation of how the approach can be used to

define the future development and improvement of the process set.

The results obtained have been encouraging serving to validate the approach and confirm much that was already known, but not necessarily quantified, about the CSEP. As such, the utility of the approach has been demonstrated sufficiently to merit extension. Future work will seek to refine and expand the approach through application to further categories of risk, notably human factors, legal and regulatory, supply chain and resourcing.

Currently the approach is entirely qualitative in nature, and will remain so while the scope, in terms of risk categories, is fully established. Nevertheless the potential for using the approach to support quantitative analysis is already apparent and will be considered in more detail as the approach is refined.

One aspect of future work that has been deliberately omitted thus far but which will need to be accounted for as the approach grows in scale is that of the underlying Information Architecture (IA). Related to this issue are questions as to the most appropriate models for supporting adoption by projects in a consistent and effective manner. Implementation aspects such as these will ultimately contribute to determining the success or failure of the approach just as much as the intrinsic strengths, or weaknesses, embodied within it.

References

[1] A. D. Bain and S. Dobson, "Safety Cases for Legacy Warships: A Systematic Approach," in *3rd IET International Conference on System Safety*, Birmingham, UK, (2008).

[2] B. Boehm, IEEE Tutorial on Software Risk Management, Los Alamitos, California, USA: IEEE Computer Society Press, (1989).

[3] R. Charette, Software Engineering Risk Analysis and Management, New York, USA: McGraw-Hill, (1989).

[4] B. Kitchenham, T. Dyba and M. Jorgensen, "Evidence-based software engineering," in *26th International Conference on Software Engineering (ICSE 2004)*, Edinburgh, UK, (2004).

[5] T. Dyba, B. A. Kitchenham and M. Jorgensen, "Evidence-based software engineering for practitioners," *IEEE Software,* vol. 22, no. 1, pp. 58-65, (2005).

[6] J. Ropponen and K. Lyytinen, "Components of software development risk: how to address them? A project manager survey," *IEEE Transactions on Software Engineering,* vol. 26, no. 2, pp. 98-112, Feb (2000).

[7] M. Chrissis, M. Konrad and S. Shrum, CMMI: Guidelines for Process Integration and Product Improvement, Second ed., Upper Saddle River, NJ, USA: Addison-Wesley, (2006).

[8] M. Carr, S. Konda, I. Monarch, F. Ulrich and C. Walker, "Taxonomy-Based Risk Identification (CMU/SEI-93-TR-6)," Carnegie Mellon University, Pittsburgh, USA, (1993).

[9] R. Higuera and Y. Haimes, "Software Risk Management (CMU/SEI-96-TR-012)," Carnegie Mellon University, Pittsburgh, USA, (1996).

[10] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey and S. Linkman, "Systematic literature reviews in software engineering - A systematic literature review," *Information and Software Technology,* vol. 51, no. 1, pp. 7-15, (2009).

[11] B. Kitchenham, R. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi and S. Linkman, "Systematic literature reviews in software engineering - A tertiary study," *Information and Software Technology,* vol. 52, no. 8, pp. 792-805, (2010).

[12] J. Li, O. Slyngstad, M. Torchiano, M. Morisio and C. Bunse, "A State-of-the-Practice Survey of Risk Management in Development with Off-the-Shelf Software Components," *IEEE Transactions on Software Engineering,* vol. 34, no. 2, pp. 271-286, March (2008).

[13] P. L. Bannerman, "Risk and risk management in software projects: A reassessment," *Journal of Systems and Software,* vol. 81, no. 12, pp. 2118 - 2133, (2008).

[14] I. I. Nurdiani, R. Jabangwe, D. Smite and D. Damian, "Risk Identification and Risk Mitigation Instruments for Global Software Development: Systematic Review and Survey Results," in *Sixth IEEE International Conference on Global Software Engineering Workshop (ICGSEW 2011)*, Helsinki, Finland, (2011).

[15] D. Liu, Q. Wang and J. Xiao, "The role of software process simulation modeling in software risk management: A systematic review," in *3rd International Symposium on Empirical Software Engineering and Measurement (ESEM 2009)*, Lake Buena Vista, Florida, USA, (2009).

[16] S. S. MacDonell, M. Shepperd, B. Kitchenham and E. Mendes, "How Reliable Are Systematic Reviews in Empirical Software Engineering?," *IEEE Transactions on Software Engineering,* vol. 36, no. 5, pp. 676-687, (2010).

[17] S. Lewis and K. Smith, "Lesson Learned From Real World Application of the Bow-Tie Method," in *6th Global Congress on Process Safety*, San Antonio, Texas, USA, (2010).

[18] B. Boehm, "Software risk management: principles and practices," *IEEE Software,* vol. 8, no. 1, pp. 32-41, (1991).

[19] R. D. Hawkins and T. P. Kelly, "A framework for determining the sufficiency of software safety assurance," in *7th IET International Conference on System Safety, incorporating the Cyber Security Conference*, Edinburgh, UK, (2012).

[20] E. Odzaly, D. Greer and P. Sage, "Software risk management barriers: An empirical study," in *3rd International Symposium on Empirical Software Engineering and Measurement (ESEM 2009)*, Lake Buena Vista, Florida, USA, (2009).