Potts, M. W., Johnson, A., & Bullock, S. (2020). Evaluating the Complexity of Engineered Systems: A Framework Informed by a User Case Study. *Systems Engineering*, *23*(6), 707-723. https://doi.org/10.1002/sys.21558

Received: 20 May 2020 | Revised: 4 August 2020 | Accepted: 20 August 2020

DOI: 10.1002/sys.21558

WILEY

REGULAR PAPER

Evaluating the complexity of engineered systems: A framework informed by a user case study

Matthew W. Potts[1] | Angus Johnson[2] | Seth Bullock[3]

[1] Aerospace Department, The University of Bristol, Bristol, United Kingdom

[2] Thales Research, Technology and Innovation, Reading, Berkshire, United Kingdom

[3] Department of Computer Science, The University of Bristol, Bristol, United Kingdom

Correspondence
Matthew W. Potts, Aerospace Department, The University of Bristol, Bristol, Avonmouth, BS8 1TB, United Kingdom.
Email: matt.potts@bristol.ac.uk

Funding information
Engineering and Physical Sciences Research Council, Grant/Award Number: iCASE 16000139

Abstract

Evaluating the complexity of an engineered system is challenging for any organization, even more so when operating in a System-of-Systems (SoS) context. Here, we analyze one particular decision support tool as an illustratory case study. This tool has been used for several years by Thales Group to evaluate system complexity across a variety of industrial engineering projects. The case study is informed by analysis of semistructured interviews with systems engineering experts within the Thales Group. This analysis reveals a number of positive and negative aspects of (i) the tool itself and (ii) the way in which the tool is embedded operationally within the wider organization. While the first set of issues may be solved by making improvements to the tool itself, informed by further comparative analysis and growing literature on complexity evaluation, the second "embedding challenge" is distinct, seemingly receiving less attention in the literature. In this paper, we focus on addressing this embedding challenge, by introducing a complexity evaluation *framework*, designed according to a set of principles derived from the case study analysis; namely that any effective complexity evaluation activity should feature *collaborative* effort toward building an evaluation informed by a shared understanding of contextually relevant complexity factors, *iterative* (re-)evaluation over the course of a project, and *progressive* refinement of the complexity evaluation tools and processes themselves through linking project evaluations to project outcomes via a wider organizational learning cycle. The paper concludes by considering next steps including the challenge of assuring that such a framework is being implemented effectively.

KEYWORDS

complexity science, project planning/assessment/control, risk and opportunity management

1 | INTRODUCTION

Increasingly, organizations are confronted with the challenge of engineering a complex System-of-Systems (SoS), or engineering a system that operates in a complex SoS context.[1–5] System complexity poses several challenges to such organizations as complex systems are generally made up of a large number of diverse subsystems and components, interconnected and interdependent via different kinds of nonlinear relationship, which can lead to difficulty in predicting overall system behavior and performance.[6–9] In the context of SoS engineering, organizations face a compounding challenge of engineering systems that operate in conjunction with other diverse systems, often with some level of autonomy and emergent capabilities.[10–13] SoS engineering also confronts a broad scope of nontechnical challenges including politicinssal, economical, social, legislative, and environmental considerations.[10–12,14–20] In such contexts, complexity also presents management challenges that must be overcome in order to successfully navigate the delivery of such systems.[21–23] Further, several systems engineering contexts confront additional domain-specific challenges, for example, working with novel and cutting-edge technologies in the defense and space domains,[24] or having to meet exacting certification demands in the aerospace and healthcare domains.[25–28]

Organizations wishing to successfully engineer such systems may find themselves having to make several difficult technical and operational decisions at the start of the defined system life cycle, such as: Do we wish to bid on a "Request for Proposal"? If we do, how much risk are we exposing ourselves to? If we go on to design, deliver, and qualify the solution, how can we be confident that we have engineered the right system, and engineered the system right? In answering such questions, one important consideration for organizations may be to evaluate the *complexity* of their candidate systems, and assess the implications of this complexity for *understanding*[29] a System of Interest (SoI) and or in *realizing* it.[6,16,30] Organizations may rely on guidelines, instructions, and decision support tools to help inform this type of evaluation. Here, one particular industrial complexity evaluation decision support tool is reviewed as an illustratory case study in order to identify challenges for SoS complexity evaluation throughout a development life cycle. In this paper, we are particularly interested in distinguishing between the challenges involved in designing an effective tool, and the related challenges involved in operationally embedding this tool such that it is effective within an organization.

The paper is structured as follows: First, the literature related to challenges in system complexity evaluation is reviewed. Then, an industrial complexity evaluation decision support tool employed within the Thales Group is introduced as an exemplar of current industrial practice with respect to complexity evaluation. The paper then details the methodology employed for the semistructured qualitative interviews conducted with experts between March 2019 and September 2019 at various Thales Group locations within Europe and spanning multiple domains. From the results of the qualitative interviews, we identify positive and negative features of the (i) the tool itself and (ii) the way in which the tool is operationally embedded within the organization. From this analysis we derive foundational principles for a complexity evaluation framework within which any complexity evaluation tool can be embedded. Further work is identified to refine and eventually validate the framework before conclusions are drawn.

The purpose of this paper is twofold: (i) distinguish the different hurdles facing organizations hoping to successfully evaluate system complexity, so that they can enter into such a process with "eyes open," and (ii) advance the development, refinement, and validation of a holistic SoS complexity evaluation framework that deals explicitly with the "embedding challenge." The primary research question addressed by this research is therefore: "How should organizations embed complexity evaluation tools in order to derive benefits from system complexity evaluation and understand these benefits?"

2 | LITERATURE REVIEW

Increases in the complexity of an engineered system may have challenging consequences that include increases in the system's life cycle costs and increased difficulty in repairing and maintaining the system.[31] Evaluating the complexity of a system can usefully inform decision management and risk management processes throughout a system life cycle, and contribute to architecture evaluation and system analysis processes.[26] However, a significant challenge for those wishing to evaluate system complexity, and one that persists despite considerable research effort, is finding a single, agreed definition of system complexity.[6,7,30–35]

There are a range of perspectives on the term,[8] with some researchers arguing that engineering efforts should be concerned, primarily, with *structural complexity*[36–38] (see also *descriptive complexity*[39]), while others emphasize *dynamic complexity*,[40,41] or *sociopolitical complexity*.[42] The relationships between these types of complexity is described by Sheard and Mostashari.[8]

For a systems engineer or architect, *structural complexity* can quantify the complexity of a system or product architecture.[36–38] In such approaches, the structural complexity of an architecture depends on the heterogeneity and quantity of different architectural elements and their connectivity. The structural complexity metric (Equation 1, where C represents structural complexity) includes three terms: C_1 represents "component complexity," which is the sum of complexities of individual components; C_2 represents the total number of pair-wise interfaces and is pertinent to interface design activity; and C_3 represents the topological complexity of the architecture and is pertinent to systems integration activity. The terms in the expanded structural complexity metric (Equation 2) are defined in more detail in Refs. 36–38 and rely on adjacency matrices to represent the architecture (e.g., design structure matrices [DSMs]) to quantify the number of components (n) in the system and the connectivity between them.

$$C = C_1 + C_2C_3, \tag{1}$$

$$C = \sum_{i=1}^{n} \alpha_i + \left[\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{ij}A_{ij}\right]\gamma E(A), \tag{2}$$

where A_{ij} captures the connectivity between elements i and j, and interface complexity (β_{ij}) depends on the complexities of the pair-wise interfacing components (α_i and α_i) and a coefficient characteristic of the interface type (f_{ij}):

$$\beta_{ij} = f_{ij}\alpha_i\alpha_j, \tag{3}$$

where $\alpha_i, \alpha_j \neq 0$ and $\gamma = \frac{1}{n}$ and $E(A) = \sum_{i=1}^{n} \sigma_i$, where σ_i represents ith singular value.

However, care must be taken with such an approach as "the map is not the territory." While an adjacency matrix is certainly a useful *representation* of an engineered system for systems engineers, the complexity of this representation is not necessarily the complexity of the system itself.[43,44] Further, the constituent terms of Equation (1) themselves rely on estimations, impacting on the objectiveness of the quantification; for example, $C_1 = \sum_{i=1}^{n} \alpha_i$, where n is the number of components in the system and α_i is the complexity of the components, suggested to be estimated using judgments of technology readiness levels, which have their own limitations and challenges.[45]

Further, the implication of the proposed structural complexity metric is that a distributed architecture is inherently more complex than a centralized architecture. When measured in terms of the number of different system components and how connected these components are, such an implication makes sense. However, by not considering the *behavior* of the system, systems architects and engineers may be neglecting vital information in their evaluation of candidate architectures.

A complementary approach is to evaluate the "dynamic complexity" of a system, which principally concerns evaluating the amount of difficulty in predicting a system's behavior.[41] In such an approach, three elements determine the overall behavioral complexity of an engineered system: the system being observed, the capabilities of the observer, and the behavior the observer is attempting to predict. Behavioral complexity[41] at time t is then:

$$C_t = \sum_{i=1}^{I} - \ln(1 - p_{bi}), \tag{4}$$

where $(1 - p_{bi})$ is the estimate probability of failing to correctly predict behavior. By including confidence intervals such an approach includes consideration of the capabilities of the observer. While such an approach can include considerations of behavior relating to mission performance (e.g., component failures) and the wider system context (e.g., technology supportability), there is a challenge for systems engineers in not only determining which aspects are particularly important for evaluation, but also fundamentally in the reliance on subjective judgments of probabilities relating to all three of these elements (the system itself, the behavior of the system, and the capabilities of the observer). Further, by this definition of dynamic complexity, as systems become more *dynamically* complex, it necessitates a reduction in the confidence and accuracy of estimating this property.

While evaluating structural or dynamic complexity is nonetheless likely to be a useful approach for system architects and can provide important insights during system design and system analysis, to what extent can such approaches usefully inform other activities in a life cycle, for example, prebid/bid activity, or support evaluations of candidate projects, stakeholder, and operational contexts?

Contrast these two quantifiable perspectives on system complexity, "structural complexity" and "dynamic complexity," with "socio-political complexity," which emphasizes the effect of people on the complexity of a system.[42] This includes competing *perceptions* of how complex a system is due to the multiple diverse viewpoints of the system and the wider context by stakeholders. It also includes the behaviors of people, agents, or the system in relation to these, and the difficulty in predicting outcomes based on inputs when contrasted with simple systems. How can this kind of complexity be measured or quantified? While one can count the number of different stakeholders (or types of different stakeholders), or estimate metrics such as the degree to which stakeholders are aligned in their perceptions, a challenge remains for organizations to make sense of this subjective, contested property.

Further, how is a complex system distinguished from a complicated system? Some argue a complicated system is one that "one can model and predict outcomes in a way that cannot be done with a complex system,"[5] while others, also recognizing emergent behavior as a distinguishing feature, instead emphasize the distinction in terms of how difficult a system is to understand or successfully realize, stressing that complexity is observer dependent.[16]

For a more extensive review of the range of definitions, descriptions of distinguishing characteristics of complexity, and the challenges they present, the interested reader is directed to Refs. 34, 35, 46. For a history of recent complexity theory development and implications for systems engineers, the reader is directed to the opening two chapters of Refs. 5.

For organizations wishing to evaluate system complexity, this myriad of definitions suggests that system complexity is dependent on perspective, on which aspects of a system are deemed important and for what reasons. For example, is system complexity considered from the perspective of "the system being observed," "the capabilities of the observer," or "the behavior the observer is attempting to predict."[41] Or is the complexity of a system considered in terms of how difficult the system is to comprehend ("cognitive complexity") or how difficult the behavior of the system is to predict ("behavioral complexity").[47,48] A further challenge for systems engineering practitioners is that such clear-cut distinctions in perspective are difficult to maintain in practice as these concepts may overlap. For example, is the difficulty in predicting the behavior of an autonomous system dependent on the observer of the system or an intrinsic property of the system?

Moreover, what counts as a reasonable approach to defining system complexity depends on what type of SoI is being considered; is it limited to the *technical system(s) being developed and deployed*, or does it also include the wider *sociotechnical system* involved in developing and deploying the technical systems?[29,49] Does an organization evaluate the *technical system to be developed* or the *project to realize the technical system*?[50–52] Does it include the processes of utilizing the system once deployed or the user's perceptions of how complex the system is (e.g., how familiar users of the system are with important features of the system)?[53–55] What is the boundary of the SoI; is it the physical context of the implemented system or does it also include the more extended strategic/business context?[56,57] A complication for organizations engaged in system complexity evaluation is that these concepts may in fact be interrelated and Sheard provides a useful chart showing how a large number of complexity concepts relate to systems engineering activity (SEA) and to each other (the Systems Engineering

Complexity Contexts [SECC]).[7] In the context of SoS engineering, opaque "authorities" and managerial and operational independence of constituent systems, create a compounding confusion on the perspective challenge identified here.[26,34,58] Again, clear-cut distinctions between different perspective on SoS complexity are difficult to maintain, exemplified in the overlapping "classifying dimensions" that discriminates between the "Complexity," "Dynamicity," and "Connectivity" of an SoS but which are, in reality, interrelated terms.[59]

As a consequence, the development of unambiguous and reliable *measures* of system complexity is a considerable challenge. Metrics such as cyclomatic complexity[60] and lines of code[61] have been used in software engineering to measure software complexity, and the number and connectivity of physical system components and interfaces are used to measure the complexity of a product architecture topology.[32,36,62–65] However, developing metrics for a diverse system as a whole remains a challenge.[66] While "the number of difficult requirements" and the amount of "cognitive fog" present in the project and the "relationships among stakeholders"[30] have been used as metrics of system complexity, accurately measuring and reporting them remain nontrivial tasks.

The "Cynefin framework"[67] from Snowden and Boone is often used to categorize operational context into "simple," "complicated," "chaotic," "complex," and "disordered." Here, the assumption is that simple and complicated contexts allow cause-and-effect relationships to be known, whereas complex and chaotic contexts have no immediately apparent cause-and-effect relationships. The suggestion is that different contexts have different characteristics and require different approaches. However, while a necessary first step for an organization is to acknowledge that their operational context is complex, the extent to which the suggested guiding principles can usefully inform systems engineering methodological approaches is not yet fully known.

Stevens[56] goes further in categorizing engineered systems complexity into the "system context," the "strategic context," the "implementation context," and the "stakeholder context." A complementary view to Steven's Profiler is provided by the SEA profiler.[68] The SEA profiler advocates for adapting SEA (typified as nine activities, e.g., analysis of alternatives, defining the system problem) based on the perceived complexity, utilizing a sliding scale to help systems engineering practitioners determine appropriate approaches for these activities.[68] An aggregate assessment can also be considered across all nine typical SEAs to help a team identify whether they should approach a problem using more traditional systems engineering approaches (e.g., emphasizing the establishment of system requirements and adapting to changes contrasted with trying to predict future enterprise needs and emphasizing discovery of needed mission capabilities for complex systems engineering). This profiler can be used by project teams to discuss and check whether their approach to the engineering of a system seems appropriate for the kind of challenges they are likely to face. While these decision support tool (and the Cynefin framework) may be useful for complexity evaluation and characterizing the system or the project, there remains a challenge in combining several perspectives, categorizations, and measures into a coherent whole. Complexity assessment tools and complexity categorization frameworks are discussed in more detail in Refs. 69, 70.

The "Complex Adaptive Systems Engineering (CASE)" methodology provides guidance on additional activities that support the engineering of complex SEAs (originally a set of 8, later updated to 25 activities). These activities can collectively supplement, re-enforce, and re-emphasize traditional activity, potentially contributing differentially to tackling the kind of challenges that engineering complex systems or SoS presents.[71,72] Practitioners also have at their disposal "principles for complex systems engineering" (e.g., embrace political, operational, economic and technical aspects, nurture discussions, enforce layered architecture), which provide additional useful mechanisms for organizations to manage system complexity.[22,73] Similarly, the "Complexity Primer for Systems Engineers" suggests candidate approaches to address complexity in the problem context or environment and to address system complexity.[6]

Considering that no single perspective is likely to address all the concerns of an organization and its stakeholders, it makes sense to recognize that there is a complicated landscape of complexity definitions and approaches.[74–76] Indeed, the "Evolving toolbox for complex project management" provides a rich guide to the various toolsets that aid the successful realization of complex systems, including, *inter alia*, the use of the aforementioned profilers and methodologies, cost estimation, systems thinking, and the use of social network analysis.[77]. Overall, while different toolsets and approaches continue to proliferate and evolve throughout the systems engineering literature, there is value in pursuing empirical questions related to the effective deployment of these ideas in order to shed more light on relevant enablers and obstacles to improved engineering practice.

3 | THE THALES GROUP "COMPLEXITY PROFILER"

Here, we introduce a version of the Thales Group proprietary "Complexity Profiler,"[78,79] a spreadsheet-based tool used by teams and individuals to evaluate the complexity of systems of interest and their operational environments during prebid/bid stages and also throughout a project, in support of technical governance actions.

The Thales Group "Complexity Profiler" was inspired by the work of Stevens[56] and developed by four senior experts within Thales Group, each having around 30 years of experience in systems engineering. The tool was introduced to encourage explicit evaluation of both technical and nontechnical risk as a result of system complexity, particularly early in a system life cycle (i.e., prebid, bid phase). It was intended to support not only risk and opportunity identification and evaluation, but also mitigation activity and to identify expertise and competency requirements specific to a particular project. During development of the tool, the system complexity factors, shown in Table 1, were amended from those in the Stevens Profiler[56] to be oriented toward the supply and provision of systems as opposed to the acquisition of systems.

TABLE 1 Description of the complexity factors used in the Thales Group Complexity Profiler, ©Thales Group 2020

Complexity factor	Description
Impact of environment on solution	Impact of physical environment on the properties of the solution (which includes operational processes).
Operational concept stability	Operational concept includes concept of operation, concept of use, and concept of employment. This factor is intended to evaluate the stability and predictability of each concept (purpose, goals, mission, activity objectives) along the solution life cycle (from solution conception to disposal).
User diversity	Expected number of users and their role diversity.
External stakeholder involvement	Level of confidence regarding stakeholder support during the execution of the contract.
Life cycle interlacing	Number of system/solution life cycles possibly interlaced in a global Programme shared between several contractors.
Systems engineering effort and criticality	Level of innovation and criticality of engineered parts.
System behavior stability and determinism	The ability to define system modes, system functions, system states and system performances, and to predict their evolution according to well-defined mathematical laws.
Engineering organization	Level of cooperation and subcontracting due to team size and number of organizational units.

The aim of the Complexity Profiler is to provide a "synthetic view that assists a team in quantifying the complexity of a particular solution. The profile helps to frame the decisions and direction that a bid/project has to take. Furthermore it helps to recognise the important differences in difficulty of a project, providing ability to compare the level of difficulty of one aspect of the project against another."[78] Using the tool is intended to guard against the risk that the organization underestimates the level of challenge inherent in developing a particular system, and hence underestimates the resource requirements needed to successfully realize and deliver the system within imposed constraints.

The Thales Group Complexity Profiler is used across a wide range of systems covering the diverse portfolio of Thales Group solution offers, including, but not limited to, Optronics systems, Command and Control (C2) systems, radar systems, radio systems, etc. The user guide for the Complexity Profiler states that "System" covers a system, equipment, platform, product or service. And, that "Solution" covers "system" and any enabling "system" necessary to sustain the "system" of interest during its life cycle. Systems engineering practitioners are likely well aware that every level of this hierarchy of systems can be complex. While the emphasis of this paper is on the natures of challenges that SoS present, as these kind of systems bring an additional layer of challenge, the findings may nonetheless be useful for systems engineering practitioners operating at other levels within such a hierarchy.

The profiling is performed in three stages: (i) assess the complexity of the SoI using the factors detailed in Table 1, (ii) conduct action analysis to define an action plan to manage complexity, and, finally, (iii) support effective decisions and implement action plans.

For each of the complexity factors detailed in Table 1, an integer score (1–4) is assigned corresponding to the level of difficulty or risk that the SoI presents. This score is arrived at by discussions within the project team, including the systems engineering manager, project managers, senior managers, and engineering teams. The rationale behind each score should be recorded in the Complexity Profiler as free text. Mitigating actions are then suggested by the Complexity Profiler, which highlights particular cells containing advisory text based on the value of the scores entered, each such action being triggered by a simple "if, then" relationship conditioned on a single complexity factor. Teams may identify other mitigating actions in free text if they deem them suitable. The profiler's complexity factors, mitigating actions, and the mapping between them that it employed, were developed using the collective experience of several senior Thales personnel, including technical and nontechnical personnel.

Depending on the resultant overall complexity of the SoI, the Complexity Profiler will mandate that, as a minimum, teams discuss certain actions (e.g., a high score for "Impact of Environment on Solution" will mandate that teams discuss "Physical simulation (Mechanical, thermal, EMC, etc)"; a high score for "User Diversity" will mandate that teams discuss "Value & Cost analysis," "Concept of Operation," "Concept of Employment," and "Concept of Use"). However, actions suggested by the Complexity Profiler are only mandated to be *discussed*. The Complexity Profiler includes a section to annotate in free text the identified risks and the proposed actions to mitigate them. Finally, the team are expected to use the information they have captured to support relevant decisions for the particular development phase the team finds themselves in (prebid, bid, etc.) and to implement an agreed action plan, perhaps launching dedicated investigations and interventions in order to gain more knowledge about the SoI and its environment. As an example, consider a SoI at bid stage that scores highly for "Operational Concept Stability" and "User Diversity." As a result of using the Complexity Profiler the project team decides to undertake a dedicated work package to understand the operational complexity in more detail to support their bid, such as conducting mission analysis and producing or refining operational concept documents. As a further example, consider a synthetic system for which a complexity profile has been completed, shown in Figures 1 and 2. The complexity of this synthetic system appears to be predominately nontechnical, with low scores for "Impact of Environment on Solution" and "System Behaviour Stability" but high scores for "External Stakeholder Involvement" and

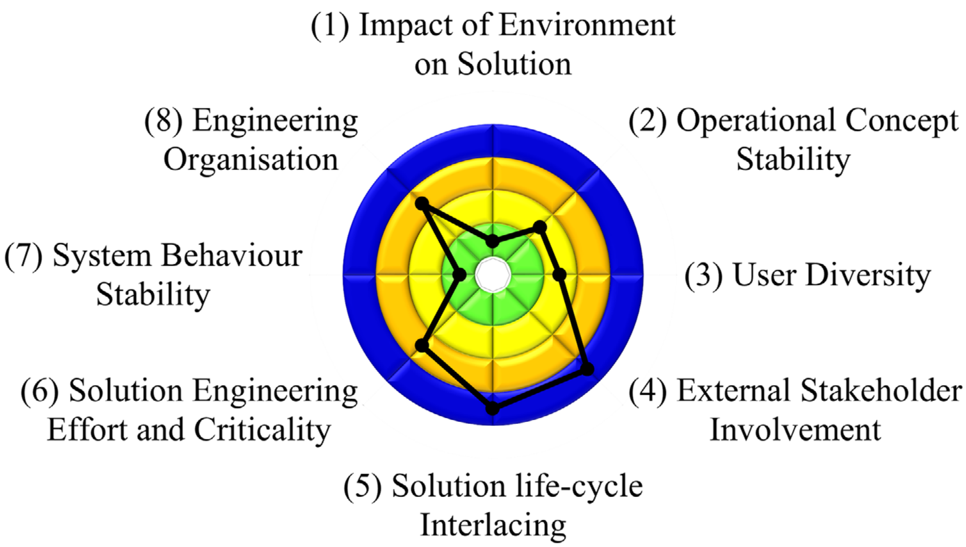
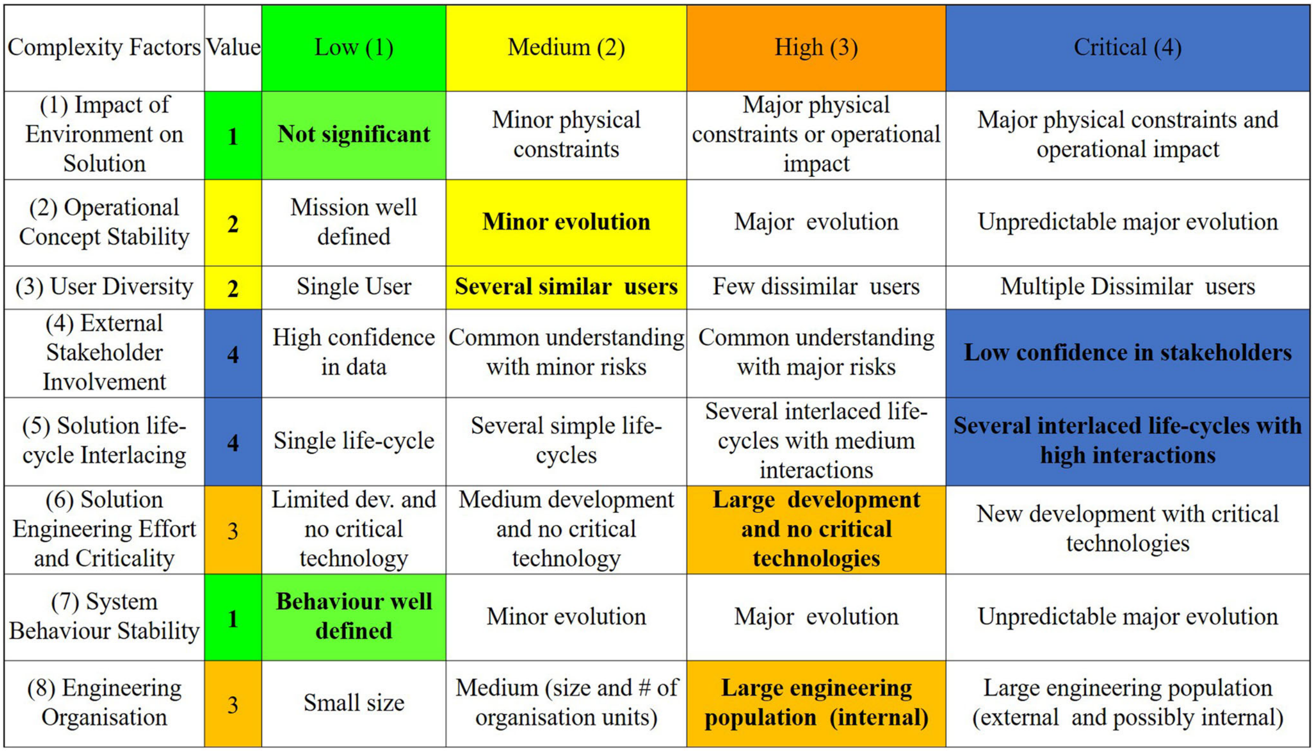


FIGURE 1 View of a completed complexity profile for a synthetic system demonstrating complexity, which appears to be predominately nontechnical, suggesting care and attention should be focused on project management, contracting, and commercial arrangements



Complexity Factors	Value	Low (1)	Medium (2)	High (3)	Critical (4)
(1) Impact of Environment on Solution	1	**Not significant**	Minor physical constraints	Major physical constraints or operational impact	Major physical constraints and operational impact
(2) Operational Concept Stability	2	Mission well defined	**Minor evolution**	Major evolution	Unpredictable major evolution
(3) User Diversity	2	Single User	**Several similar users**	Few dissimilar users	Multiple Dissimilar users
(4) External Stakeholder Involvement	4	High confidence in data	Common understanding with minor risks	Common understanding with major risks	**Low confidence in stakeholders**
(5) Solution life-cycle Interlacing	4	Single life-cycle	Several simple life-cycles	Several interlaced life-cycles with medium interactions	**Several interlaced life-cycles with high interactions**
(6) Solution Engineering Effort and Criticality	3	Limited dev. and no critical technology	Medium development and no critical technology	**Large development and no critical technologies**	New development with critical technologies
(7) System Behaviour Stability	1	**Behaviour well defined**	Minor evolution	Major evolution	Unpredictable major evolution
(8) Engineering Organisation	3	Small size	Medium (size and # of organisation units)	**Large engineering population (internal)**	Large engineering population (external and possibly internal)

FIGURE 2 View of complexity factor scores for the same synthetic system as shown in Figure 1, taken from the Thales Group Complexity Profiler, ©Thales Group 2020

"Solution life-cycle Interlacing" suggesting care and attention should be focused on project management, contracting, and commercial arrangements.

The integer scores assigned to the eight complexity factors in the Complexity Profiler are manually entered into a separate decision support tool. This separate tool considers nontechnical aspects of project delivery and is used to determine the governance and reporting arrangements for a project, based on factors such as commercial, financial, strategic, and technical risk, where technical risk is represented by the values from the Complexity Profiler. Thus, the scores assigned in the Complexity Profiler also affect project governance and reporting mechanisms.

4 | METHODOLOGY

Semistructured qualitative interviews were undertaken in order to collect information on complexity evaluation within Thales Group and the utilization of the proprietary "Complexity Profiler"; exploring perceptions of its strengths and weaknesses, opportunities for improvement, and challenges for its exploitation. A semistructured interview format involves using a number of open and closed questions that provide a formal structure. However, they also allow for further discourse as required to establish a depth of understanding.

The target population was personnel with over 10 years experience working in a systems engineering context within Thales Group who have experience evaluating the complexity of systems. The sample population is predominately systems engineering managers, systems architects, and enterprise architects, although the roles and jobs they undertake within Thales Group vary. The population was sampled using theoretical sampling; individuals were chosen as those who were in the best position to provide answers that were well informed and relevant. A sample size was not predetermined; instead, the interviews were conducted until more than a minimum number (10) had been carried out, and a saturation point was achieved, where no significant new views were uncovered during the interviews. Interviewees were also asked to nominate additional potential candidates as part of the interview.

The interview protocol was approved by the University of Bristol Ethics Committee on March 12, 2019 (application ID 81802). Interviewees were invited via email with a standard message and were given a participant information sheet. If participants agreed to participate, a mutually convenient date, time, and location for the interviews were arranged. Interviewees were given a participant consent form to read and sign. The interviews were guided by 17 questions (Supplemental Material).The research was conducted within an action research methodology,[80,81] in line with a systems thinking approach.[76,82]

A total of 16 interviews were conducted, with two additional preliminary pilot interviews. The pilot interviews were conducted to test and adjust the interview methodology, data from the pilot interviews are not reported here. The interviews took place between March 2019 and September 2019 at various Thales Group locations within the United Kingdom and France, representing multiple business units within the Group.

The interviews were audio recorded, and were then transcribed. A thematic analysis was conducted on the interview transcripts.[83,84] First, key messages were codified in a deductive way as contributions to the research question on the utility of complexity evaluation within Thales Group. The resultant codes were then refined to remove duplication and emphasize important messages. In the results section, we report the main findings and provide a reference to textual excerpts to support the findings where appropriate.

The documentation supporting this research: Introductory email, participant information sheet, participant consent form, interview questions, anonymized biography of participants, textual excerpts, are published as open data and can be accessed from Ref. 85. The analysis that follows signposts the reader to the relevant section of the textual excerpts documents by directing them to a specific section (A– K) in that document.

5 | RESULTS

The pertinent findings of the interviews are presented under two subheadings; (i) the tool itself (the Thales Group "Complexity Profiler") and (ii) the tool's embedding. From this analysis, key features of a complexity evaluation framework are derived within which any complexity evaluation tool might be embedded operationally.

5.1 | The tool itself

5.1.1 | Positives

Respondents generally found the "Complexity Profiler" easy to use; "[I wanted to ask you when you have used the Complexity Profiler, how easy is it to use overall?] It's not too difficult, I think. I find it relatively easy." [IF]. "[...Suggesting that it is quite an easy to use tool?] To me it is, yes." [IP]. "[In your opinion, how easy is the Complexity Profiler to use overall?] Yeah, pretty easy. [And the outputs are easy to understand?] Yeah, I quite like the type of polar plot. I've used that technique in quite a number of ways to describe risk and maturity levels of systems, so I think it's quite a neat way of capturing it." [IK]. Interviewees generally reported that the eight "complexity factors" were relevant and easy to understand (Section A), but were not exhaustive and were open to interpretation, points that are elaborated on later.

Further, several respondents claimed the "Complexity Profiler" is useful to them for different reasons; for some it was useful for surfacing risks that may otherwise be unnoticed (Section B), for others it could be useful for justifying project resources to mitigate identified risks (Section C), while others claimed it aided communication between technical and nontechnical personnel (Section D). Others suggested using the "Complexity Profiler" helped to demonstrate that a project team had considered the complexity of a candidate system prior to project reviews (Section E).

Below is an extract from an interview (IH), reporting that the most important feature of the "Complexity Profiler" was the identification of risk areas, which they felt was done well, with an acknowledgment that the tool will not manage risk *for* an organization on its own without further effort from personnel but that it will help with the identification of risk.

> If you recognise system complexity, and that's the key in most management, if you know what you've got to manage, it's one thing. That's not to say even when you've identified, it's not necessarily easy to manage but if you haven't identified the complexity in the first place, then frankly you've got no hope. So, the absolute key is you've got to identify the complexity, you've got to identify the risk areas. You've got to identify where you need

> to focus effort and then you can focus on managing those risks and as a program director, or [REDACTED] engineering manager, you can then look to what sort of team or what sort of resource or service you're going to need to – or what artefacts, you're going to need to develop, or expertise you're going to need to bring in to help develop those things. If complexity profiler points to you need high risk on the CONOPS [Concept of Operations] you're going to need to get in then, some users. It's pointless going on with glorious isolation without consulting any users, for example. It absolutely helps identifying it. It's not going to do the managing for you though.

Below is an extract from an interview (IE), where the "Complexity Profiler" has proven to be useful to justify resources in a project to conduct system modeling and simulation activity to de-risk the systems development project.

> It's quite often useful certainly for doing system modelling – solution modelling – quite often the business will ask – 'cause it can be quite time consuming to do a modelling activity – creating models in Enterprise Architect or something and following through. The justification sometimes is yes, but we've already demonstrated to you there's high complexity in this area, therefore the justification for doing the modelling is to address that complexity. If you don't have that, then they can push back and say, "Why do you need to spend all this time doing system modelling?"

When discussing the use of the "Complexity Profiler" to aid communication, a participant reported the following; "[What about communicating about system complexity? Does the complexity profiler help you communicate about system complexity?] Yeah, so I think that's probably something that it does do reasonably well because at the end of it there is a way of describing to the stakeholders of you of how complex we think the job is that we've got. That's the bit I think where it does have its use." The interview went on to discuss the role the "Complexity Profiler" plays in aiding communication of risk, from the engineering community to the nonengineering community.

> [What's the current value, and what do you think it could be?] I think the current value is limited as a communication tool from the engineering community to the non-engineering community... most people should be able to look at that and then say, 'Oh, look, everything is on the outer of the wheel,' or, 'Everything's on the inner of the wheel.' But I think the last half an hour has made me think about what the value really could be of it in enabling us to really put complexity at the heart of risk management and therefore, now we're talking about technical risk management, where we quite often think about it as a project management tool. The other thing that it could do is drive us to think about integration early. Thales doesn't have a strong culture of integration. We put systems together, but thinking about integration as a skill, as a capability, as a thing about everything that we do, this could perhaps help us to drive there. Either a really simple approach to integration and it's early, little and often. If you could use this tool to help us drive which things we're doing early, how we chunk up our complexity to do the integration often – so what are the little bits, and how are we gonna do it – we could use it to drive an integration plan with strong links to our risk and opportunity plan, and now we're managing through complexity throughout the life cycle of a programme.

5.1.2 | Negatives

Respondents identified several practical challenges and limitations that hinder the effectiveness of the tool. In evaluating a limited number of poorly defined, subjective properties on a crude numerical scale the Complexity Profiler masks the problems associated with measuring a disputed property. Consider how two individuals, with an equivalent amount of experience, might nevertheless draw on experiences with different characters, resulting in assessments that vary yet are, to first approximations, equally valid. Further, when evaluating a complexity factor such as "User Diversity," how does one characterize on a single four-point scale the impact of small differences in several different user profiles to significant differences in only a few different user profiles. Or consider the different meaning that "User Diversity" may have for individuals in the hardware domain contrasted with those in the software domain. See Section H. Below is an extract from an interview (IM) where this challenge is discussed.

> [How easy do you feel these factors are to understand?] You're asking the question that you should never ask in one sense because the minute you have a list of things around engineering that you should measure, every engineer will come up with a different view as to what that list should be. As a list I don't think it's too bad actually if I'm honest, but it wouldn't be the list I'd start off with. [Do you feel that this would have a different list?] Yeah. If you asked 20 engineers to come up with the eight complexity factors that they would use for assessment in a complexity profiler I think you would find that those eight would appear at some point, or most of those eight. You might have to sort of make a liberal interpretation of what the words that somebody used as being the same or similar but I think you would discover another 10 that are not on that list probably, which then says is eight enough? is there something missing that absolutely should be there?

A compounding issue is that the profiler is not sensitive to interactions between the defined properties, evidenced in the profiler's simple "if, then" mappings, each conditioning a suggested mitigating action on a single complexity factor score. Trying to evaluate the impact of complexity factors in this reductionist manner may miss compounding risks arising from the interaction between factors. See Section I. and below from an interview (IH) where this challenge is discussed. One recommendation to alleviate this challenge is to include an explicit column within the Thales Group "Complexity Profiler" that encourages consideration of compounding risks and issues as a result of a combination of system complexity factors. This issue is also applicable for other complexity assessment tools surveyed in the literature such as the "SEA Profiler"[68] and Steven's Profiler.[56]

> A wider review of it may have elicited a perhaps bigger scaling of the mitigation actions. If not the nature of the actions themselves, but as I say, rather than writing something off as oh yeah, it'll take a couple of weeks of effort to solve that, to implement that mitigation. It might have been a more, hang on a minute, if you look at the other dimensions of the complexity profile, when actually how many stakeholders you've got to engage, or how many different sites there are to engage with, actually that's not just an isolated thing, it's the power law effect of multiple dimensions of complexity really made things harder to mitigate than you might otherwise think. I think that's something that doesn't directly come out of the complexity tool, directly. You have to weave that in when you describe your mitigations.

While the interviewees generally suggested the current set of eight complexity factors included in the Thales Group "Complexity Profiler" to be a sufficient starting point, several other factors could also be included. Some of these suggestions include; systems integration effort [IH, IK], regulatory complexity ("The word integration doesn't appear anywhere in there. That and regulatory compliance I think are the two that would, I guess, because the regulatory compliance on both my current programme and a previous one did significantly drive the engineering..."' [IK], cyber security considerations [IM], and supply chain considerations [IM]:

> Increasingly we have to deal with flowing down complex requirements to suppliers to supply something to integrate into our systems, so we're not only having to manage the complexity of our own activities but also manage the complexity of the things we flow down to our suppliers to do for us and our ability to manage our supply chain from a technical perspective not from a pure procurement…perspective, is not so easy. I see more and more complexity and problems coming from the fact that we are a conductor of an orchestra rather than the guys that play all the instruments and that is not an easy…

Further, the literature surveyed additionally identified the following potentially relevant system complexity factors that could be suitable additions to the Thales Group "Complexity Profiler", for example; "requirement difficulty," "cognitive fog," and "stable stakeholder relationships,"[30] structural complexity (the number, diversity and connectivity of components, subsystems and systems alongside their connectivity),[36–38] "dynamic complexity" (the difficulty in predicting behavior)[41] difficulty conducting functional analysis and allocation,[4] and technology maturity.[45] The issue is also applicable for organizations who make use of other complexity assessment tools surveyed in the literature such as the "SEA Profiler"[68] and Steven's Profiler.[56]

A previous study collected judgments from current systems engineering practitioners on the relative and absolute importance of several different system complexity factors identified that additional relevant system complexity factors could include consideration of the number and diversity of system interfaces and dependencies, nonfunctional requirements, and "client/customer/user complexity (e.g., their understanding of the system, novelty of the system to them, willingness to accept change)."[86] For the Thales Group "Complexity Profiler," this finding implicates that the complexity factors encoded in the tool requires updating, preferably via a combination of literature survey and a survey of experienced systems engineering managers from across the organization. However, care must be taken with the addition of more system complexity factors to such tools to ensure that the factors are adequately and unambiguously defined. Providing additional clarity on the contested ontology of system complexity would be an additional improvement. While academic literature can detangle the term complexity from notions of complicatedness or volatility, the Thales Group "Complexity Profiler" does not achieve the same clear-cut distinctions. The lack of a consensus view within the systems engineering community on the relative importance of different system complexity factors causes an additional challenge. That systems engineers are not aligned with each other on how important system complexity factors are, and may not appreciate the extent or nature of these misalignments, may hinder efforts to effectively identify, evaluate, and manage sources of system complexity.[86]

Similarly, the limited number of complexity factors encoded in the profiler also leaves specific SoS considerations as a blind spot. While the SoS literature emphasizes *autonomy*, *diversity*, *connectivity*, and *emergence* as distinguishing characteristics of an SoS,[10,11,16,17,26,58,82] these characteristics are not explicitly evaluated in the Complexity Profiler. Neither are the challenges presented by the autonomy, or operational and managerial independence, of constituent systems, which may result in a lack of common authority for the SoS, full consideration of constituent system constraints, or end-to-end testing and validation of the SoS.[26,58] See Section I. Including these additional SoS complexity factors appears to be a straightforward improvement for the Thales Group "Complexity Profiler," although care must be taken to ensure these are carefully described within the tool to avoid the aforementioned challenge of ambiguous, subjective system complexity factors. These issues can be improved but a wider comparative piece of research is needed in order to widen the scope to consider a set of similar evaluation tools and perform a full analysis and make detailed recommendations.

5.2 | The tool's embedding

5.2.1 | Positives

Participants felt that asking questions such as "how does the complexity of a candidate system affect our methodologies, process and outcomes?" or "what is the point of taking a complex systems perspective" starts a deliberate act of thinking and discourse within the organization. Without starting to ask these questions they would not have the "Complexity Profiler" in place and would not be asking personnel to consider certain issues relating to system complexity during systems development activity. That many respondents found the "Complexity Profiler" to be useful is a positive indicator, and while the tool is flawed, it is encouraging that the organization has started this line of enquiry and can make improvements in the future. It is a positive feature that the organization has invested effort in understanding the impact of complexity on their systems, given that many systems engineers have not been systematically engaged in an evaluation activity like this as part of their practice or training.[86]

5.2.2 | Negatives

The Complexity Profiler takes an inherently "divide-and-conquer" approach, where a single person is tasked with completing a single profile at a single point in time. The Complexity Profiler User Guide suggests that a completed profile will provide a general understanding of the complexity of the system. A more realistic outcome would instead be that such a profile would convey a general understanding of the complexity of the system *as interpreted by an individual at a single point in time during its development*. Although the Profiler recommends work is conducted collaboratively, in reality, the tool appears to be designed to be completed by an individual and is not structured in a way that allows multiple perspectives to be captured or compared. This is evidenced by the fact that the User Guide makes explicit that a single role is tasked with filling in the action analysis and defining an action plan to manage complexity, including securing resources and scheduling. Similarly, the Profiler asks for the "name of the person responsible for the profiling activity" to be recorded.

Given the varied definitions of complexity, system boundary, etc., that we have discussed earlier in the literature review, a collaborative tool capable of representing and drawing together multiple perspectives in a conversation to build shared understanding would be more appropriate. See Section G. Below is an extract from an interview (IB) where this challenge is articulated. A potential improvement for the Thales Group "Complexity Profiler" would be to redesign the tool, and supporting documentation, to encourage an initial collaborative activity that elicits and represents a discussion centered on different stakeholder's individual perspectives of how complex the SoI is. Similarly, evaluations of the *impact* of these perspectives and the subsequent suggestions for mitigating actions should be captured in the tool alongside the provenance of these discussions.

> The issue with complexity profiler is it tends to be filled in by the PDA [Project Design Authority] or the project manager in some cases but it then becomes their personal perspective on where they think the complexity is. Value from the complexity profiler comes when the delivery team or the leadership team do it as a joint effort in their own way because then they get to discuss when why they think the complexity is where it is and how complex it actually becomes. So what it does is it aligns people on the same baseline so you have a single statement of truth from the complexity of a profile as perceived by the delivery team as opposed to individual perspectives on complexity… [So who do you think would tend to be filling in the complexity profiler?] It would be done by the PDA or the project manager as an individual.

Further, the Complexity Profiler is an inherently punctate, discrete tool applied at a point in time but purporting to evaluate a system overall (i.e., timelessly), with no reference to previous or subsequent evaluations. As a project progresses, more information is likely to be uncovered about the environment the system will operate in, about other systems the SoI will interact with, etc., which may significantly change the evaluation of complexity. Similarly, factors identified during an early-stage evaluation could influence or steer subsequent evaluations, ensuring that warning signs are attended to, for example. While the Complexity User Guide suggests the Complexity Profiler should be used at several stages of a system development process, the reality is that any revision activity is discretionary and is likely to only focus on that specific point in time. "[The complexity profiler user guide suggests it should be used for our system lifecycle? Is this generally the case in your experience?] I've never seen it used throughout lifecycle" [IL]. "What tends to happen is the complexity profiler is done once and then left" [IE]. "[The complexity profiler suggests it should be used throughout the system life cycle but you're suggesting that that's generally not the case in your experience?] I think it tends to get looked at briefly to see if it still makes sense. If I'm being honest, I don't think it really gets re-evaluated thoroughly in my experience" [IH]. Instead, an evaluation of system complexity should, ideally, evolve alongside the project it relates to by being revisited periodically, and a view should be taken on the trajectory of these evaluations, rather than just considering each isolated point along the way. It is not just the destination system that should be assessed, or key waypoints, but the evolving project journey, overall. See Section J.

Without a wider learning system, new, relevant aspects of complexity may go unnoticed, and the process may not be tailored and adapted to respect the role context plays in system and project outcomes, which may in turn prevent the sharing of lessons and better practice across different projects. Although the Complexity Profiler is completed as a standard part of executing design and engineering projects, and the profiler itself mandates that certain mitigating actions are to be *discussed*, there is no process or structure to support the evolution of the evaluation process or to connect evaluations with project outcomes, either positive or negative. If an organization was to adopt a similar approach to the Thales Group Complexity Profiler or other complexity assessment tools, such as "SEA Profiler"[68] or Steven's Profiler[56], careful consideration must be given to how they are embedded. As a consequence, the use of the such tools does not straightforwardly result in improvements to the organization's ability to make the most effective decisions. While this approach could be reasonable in a context where the system evaluation being attempted was grounded in a mature and consensually agreed upon set of principles or theories (e.g., evaluating the load that a device could be expected to tolerate), this is not currently the case for complexity and is unlikely to be the case for the forseeable future. This is due to the diversity of systems, domains, operating environments, contexts, etc., that system complexity evaluation is undertaken within and the current relative lack of maturity in the underpinning theory. While software engineers have developed principles and measures for complexity evaluation,[60,87] determining overall principles and theories for the a diverse SoS as a whole remains a significant challenge. See Section K.

Finally, the Complexity Profiler is a fire-and-forget system that sits inside business processes rather than spanning them. The opportunity for the tool to impact on the evolution of business processes would be more effectively realized if it were embedded in a wider organizational learning system. The lack of such a process with which to learn from complexity evaluations was identified during an interview (IM):

> That's the thing with any of these systems, if you're really going to try and use them there should be some element of calibration of how much their predictions at the start are lined up with what you experienced at the end. [Why do you think that calibration isn't happening?] I think there's a general… I think it's just the fact that projects on the whole are not good at doing that retrospective lessons learnt at the end and if they do, they tend to do it in a sort of very wordy what went well, what didn't go well type sort of analysis rather than something quantifiable. I may be wrong, maybe someone can find one or two examples but, in my experience, I have not seen somebody go back and say well now we've done it where was it really complex, what did we say at the beginning and what's the difference? Maybe there are some benefits in doing that and it maybe that as I've said, the things that are really complex are things that are not totally covered by the complexity profile in the first place which could be part of the answer but I don't have enough evidence to say that it's that or it's we did the complexity profiler, we completely underestimated how complex certain characteristics that the complexity profiler provides were.

These issues are going to be relevant to any complexity evaluation tool, no matter how well it solves the kind of challenges described in the previous list. Any tool will only be effective and valuable if it is embedded effectively within the operation that is deploying it, as such there is a need for a framework within which complexity evaluation tools should reside.

6 | FOUNDATIONS OF A COMPLEXITY EVALUATION FRAMEWORK

While the analysis above centered on the positive and negative features of the tool, here we use the analysis to derive several features of an effective complexity evaluation framework. An effective complexity evaluation framework should provide clarity on language where significant ambiguity is present, found both in the surveyed literature and the case study. It should enable long-term organizational learning and should evolve as a result where necessary. It should dovetail with good governance to ensure it is executed effectively as part of an iterative, whole-life cycle approach. A multiperspective evaluation of system complexity, and the risks that this complexity presents, should be enabled and integrated. Mitigating strategies appropriate to these risks should be mandated in support of organizational decision making. A preliminary sketch of such a framework is provided in Figure 3. The framework has objectives to: (i) promote discussions of the role of system complexity,and (ii) facilitate shared understanding, in order to (iii) provide enhanced decision support at every life cycle phase. Every design principle of the complexity evaluation framework, Figure 3, is informed by the earlier analysis.

By encouraging an evaluation of the complexity of the SoI, organizations can gain an additional insight into decisions during pre-bid and bid phases, such as "Do we wish to bid on a "Request for Proposal"? If we do, how much risk are we exposing ourselves to?," or to support analysis of alternatives, system architecture evaluation, and system design evaluation. Further, the evaluation of system complexity may be useful to help scale the level of effort required on operational concept development or technical derisking activity such as modeling and simulation.

For the purposes of this paper, a framework is defined as "a structure … that can be used as a tool to structure thinking, ensuring consistency and completeness." [88] A complexity evaluation framework should define and support a standardized way to go about evaluating system complexity within an organization, one that promotes effective decision making, improves project outcomes, supports communication between stakeholders (internal and external), and also enhances an organization's understanding of the evaluation process itself, its strengths and weaknesses and its value or impact for the organization.[88] The benefit of such a framework is in ensuring that relevant decision makers engage appropriately with considerations of system complexity and can be shown to have so engaged. Further, the cyclical exchange of information and collaboration promote understanding of the SoI, which can reduce errors caused by the wrong interpretation of the system interfaces. We argue that while it may be useful

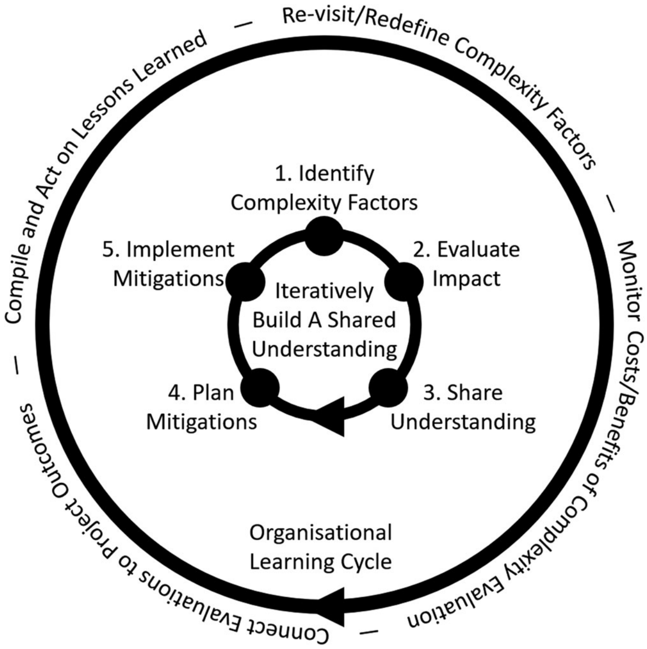

Re-visit/Redefine Complexity Factors
Monitor Costs/Benefits of Complexity Evaluation
Connect Evaluations to Project Outcomes
Compile and Act on Lessons Learned
1. Identify Complexity Factors
2. Evaluate Impact
3. Share Understanding
4. Plan Mitigations
5. Implement Mitigations
Iteratively Build A Shared Understanding
Organisational Learning Cycle

FIGURE 3 A preliminary complexity evaluation framework comprising an inner, iterative cycle of sense making that will be completed multiple times over the course of a single project, embedded within an outer cycle of organizational learning that will be completed multiple times across successive projects. By defining the sense-making as a collaborative and iterative process, the framework avoids the challenges associated with the contested definition of system complexity. By embedding complexity evaluation within a wider learning cycle an organization can revisit or redefine the aspects of the complexity that are most pertinent to their contexts, enabling lessons to be identified, learned and shared across system domains and contexts. Organizations can then monitor the costs/benefits of conducting complexity evaluation activity and understand how evaluations relate to eventual project and system outcomes

to have proformas, decision support tools, user guides, and processes, their value will be largely determined by their place within a wider process that ensures personnel interact with them appropriately. A framework in this sense is essentially a way to make people stop and think in a carefully structured way.[89]

6.1 | Collaborative

The core of the framework is an emphasis on *iteratively building a shared understanding* to handle the necessarily subjective, contested, and evolving definitions of system complexity, system boundary, level of abstraction, etc.[8,49,56] Multiple perspectives need to be drawn together and integrated in a collaborative manner in order to develop an accurate and actionable view on the system and its realization project. The need for multiple perspectives when evaluating the complexity of an engineered system is also well advocated for in academic literature.[23] One mechanism in which personnel could achieve this is through collaborating on completing and maintaining a "Complexity Register," which would require conversations that make explicit the assumptions, rationale and perspectives of different stakeholders. The emphasis on collaboration, iteration and shared understanding is a key departure from the punctate nature of a decision support tool completed by a single author at a single point in time. Alternatively, the Thales Group "Complexity Profiler" could be redesigned to promote these kind of conversations and capture a greater level of detail from discussions on the complexity of a SoI, the impact of this complexity and potential mitigating actions. In doing so, an organization can begin the process of evaluating system complexity without being hamstrung by the limited number of poorly defined, subjective properties that currently hinder the "Complexity Profiler."

Rather than employing a fixed set of complexity factors to be evaluated, the five-step evaluation process commences with a team collaborating on the identification of system complexity factors that are relevant to the current context and domain. In evaluating the impact of these contextually relevant complexity factors, and then communicating the resultant shared understanding of their impact, an organization stands a better chance of dealing with the ambiguity and opacity of a term like system complexity. The lack of consensus on complexity needs to be foregrounded and confronted explicitly by the framework, rather than relying on a limited number of defined properties treated in isolation. This approach also bakes in an assumption that complexity is an operational concept that is likely to evolve. Rather than culminate in a set of advisory mitigating actions that must be discussed, the planning and subsequent implementation of mitigations is designed to inform (and be tracked by) the next cycle of complexity evaluation.

6.2 | Iterative

Each of the five steps in the framework's inner cycle could be prefixed with "re-" to indicate that these steps are taken repeatedly: reidentify, re-evaluate, etc. An evaluation of system complexity needs to evolve alongside the project it relates to by being revisited periodically. Moreover, an organization should be concerned with monitoring the project charged with developing an SoS, and it is only in applying the complexity evaluation framework throughout a system development life cycle that this can be achieved. The evidence from the case study suggests that, despite the intention at the creation of the "Complexity Profiler" tool that it should be used an iterative tool, the reality is that the tool has largely become a "fire-and-forget" activity. While mandating revisions to the "Complexity Profiler," or other complexity assessment tool in the case of other organizations, throughout an SoI life cycle may encourage *activity* in this regard, organizations must acknowledge potential reticence for this, as one respondent described feelings of "process for the sake of process." Instead, organizations may need to establish a robust cost-benefit assessment, or value proposition, for system complexity evaluation, which can be achieved through the final design principle—the need for organizational learning.

6.3 | Progressive

Finally, complexity evaluation only has the potential to improve business practices if it is embedded within a wider organizational learning cycle, depicted in the outer loop of the framework. Evaluations of system complexity must be conducted in a contextually sensitive manner otherwise collected evaluations are unlikely to be useful in sharing lessons identified and better practice long term. In this way organizations can continually explore what aspects of complexity are pertinent in their context, and how they impact system and project outcomes.[73] It is during this broader activity of embedding complexity evaluations in organizational activity that focus can turn to recording and ensuring that the "stop and think" process is taking place effectively, and that the framework is enabling consistency and completeness. By comparing the trajectories of system development projects; comparing the initial plan with later planning, the actual activities undertaken and the eventually deployed systems, organizations can determine if the framework is helping to uncover, and subsequently mitigate, relevant complexity factors. It is only in comparing the *outcomes* of organizational decisions with the information available at the time the decisions were taken, that an assessment can be made on the effectiveness of complexity evaluation in terms of costs, benefits, impact, etc.

For Thales Group, a stronger linkage between system complexity evaluations, project outcomes, and organizational learning may have allowed new, contextually relevant system complexity factors to emerge, including SoS considerations, that effect that compounding risks and issues between system complexity factors have on projects, and would encourage the collection of data to assess the efficacy of system complexity factors and their predictive power for project outcomes. In doing so, Thales Group could have a stronger cost–benefit analysis or value proposition for their complexity evaluation activity, promoting wider and more appropriate engagement with their proprietary tool.

While this framework requires deployment and operational validation, it takes the necessary first steps toward addressing the challenges and opportunities that were derived from analysis of the case study presented here. The intention is to provide useful insights to practitioners who currently conduct complexity evaluation, or those who wish to instigate their own evaluations, and to establish a frame within which further academic analyses of the role of complexity in SoS engineering can take place. The future work section discusses further how progress may be made with this complexity evaluation framework.

7 | DISCUSSION

There are several hurdles that tend to interfere with an organization's ability to realize the intended benefits of the Thales Group "Complexity Profiler" or similar complexity assessment tools. These challenges include into tool-specific problems that can be improved by making the tool better (e.g., avoiding using a limited number of poorly defined criteria rated against a crude scale, ensuring the tool is sensitive to compounding risks and SoS considerations), and organizational embedding challenges that must be addressed separately (e.g., avoiding a divide-and-conquer approach to system complexity evaluation, and ensuring that the tool itself is updated and integrated within a wider organizational learning cycle). Addressing the second set of challenges is not a totally new idea, organizations will have been solving it in various ways. However, more literature seems to be concerned with addressing the first category of challenges, how to approach evaluation, etc., while there seems to be less work on the second category of challenges concerning how such tools are embedded. It is therefore valuable to draw attention to addressing the second category of challenges as a research question distinct from the question of how organizations define, measure and evaluate complexity.

While individual decision support tools, such as the Thales Group "Complexity Profiler," "Steven's Profiler,"[56] or the "SEA Profiler,"[68] have the potential to be useful to organizations, they need to be embedded within a wider framework in order to mitigate the challenges described here. Further, given the contested definition and subjectivity of the term system complexity evidenced in the literature, organizations need to think about system complexity in a more holistic way. An instantiation of one such complexity evaluation framework is introduced here. It seeks to achieve robust and effective evaluations through the collaborative identification and evaluation of contextually relevant system complexity factors, with continuous re-evaluations encouraged and supported by a wider emphasis on organizational learning. These design principles are well aligned, but not fully mapped, with wider principles for the engineering of complex adaptive systems.[5,71]

There are several challenges that remain for organizations wishing to evaluate the complexity of their candidate systems. First, *identifying* risks arising from the complexity of a system cannot be taken to mean these risks have been *managed*. Second, how can organizations ensure their guiding tools, processes, and frameworks are engaged with in an appropriate way. Finally, and perhaps most importantly, how can organizations be sure that the "stop and think" process is actually happening? Appropriate engagement with the Thales Group "Complexity Profiler" was difficult for personnel engaging with the tool to ensure, for example avoiding biases or even completing the "Complexity Profiler" itself and the same challenges also apply for the framework proposed here.

Evaluating the complexity of a candidate system should not be an end in itself, rather, we argue that it must directly inform an organization's substantive decision making. Complexity evaluation must trigger mitigating action to reduce the likelihood and/or impact of the identified complexity. While the proposed framework encourages organizations to relate their evaluations to suitable mitigating actions and eventual project outcomes, the challenge of implementing this should not be underestimated. Additionally, given the apparent relationship between system complexity evaluation and project risk, care must be taken to ensure complexity evaluation activity is integrated with an organization's wider through-life Risk Management process.[26,90] Organizations should also be cognizant that, while we have emphasized the treatment of *risk* here, consideration should also explicitly include the treatment of *opportunities*.[91]. As others have suggested: "In complex (enterprise) environments like that of an SoS, it is better to have an opportunity exploration mindset as opposed to a risk mitigation mindset."[23]

Aligning system complexity evaluation within a broader suite of organizational decision management processes is a further challenge as such evaluations are only one aspect of these processes, whether at the concept or development stages of a life cycle.[26] Various other factors also need to be considered such as, *inter alia*, strategic direction, technology roadmaps, development strategy, risk exposure, etc. Ensuring that system complexity evaluation is complementary to other existing system analyses remains a further challenge for organizations. The relationship between system complexity and the type of system is a compounding challenge for organizations designing and developing diverse systems; the kind of risks and issues presented by the complexity of an autonomous system are different to the kind of risks and issues presented by the development of a state-of-the-art thermal imaging system.

There are some methodological limitations of the study. The case study reported here is specific to one organization and could be usefully augmented by more empirical studies of complexity evaluation in situ. Similarly, the sample of interviewees was drawn from one organization and may not be representative of a wider array of organizations facing the challenge of evaluating system complexity. Further, the number of participants interviewed ($n = 16$) is low considering the number of senior systems engineering managers within Thales Group, along with the size and global footprint of the Group.

Ideally, organizations would have strong empirical evidence of the efficacy of the relevant system complexity factors as predictors of system realization project outcomes. While other studies have identified some system complexity factors that have a significant impact on generally desirable project outcomes,[30] the same kind of investigation has not yet begun within Thales Group to understand the impact of the eight factors in their "Complexity Profiler." While future research projects can examine this, there is a considerable methodological challenge in securing accurate data to confidently assess the predictive power of these factors in project outcomes given, *inter alia*, the long time horizon projects, the multitude of other variables that affect project outcomes, and that even if statistically significant correlations are found between project outcomes and system complexity factors, these relationships do not infer causation. Nonetheless, the success of similar previous studies in this direction suggest a positive outlook for future investigations.

8 | FUTURE WORK

In order to further refine the proposed framework, principles for system complexity evaluation need to be developed in order to deal with, *inter alia*, SoS-specific considerations, life cycle tailoring, defining complexity within and between the contexts of different individual business units, providing clarity on terminology and governance. The problem structuring methods associated with Soft Systems Methodology (SSM) appear to be particularly well suited to developing these strands of future work.[89] An SSM investigation would aid understanding of the human activity system that undertakes complexity evaluation within an organization in more detail and would identify and inform the implementation of desirable and feasible improvements to the framework. Such an investigation may also offer insight into how the framework can be assured, that is, how an organization could establish confidence that the framework is being implemented effectively. With a mature framework specified, attention could turn to deploying it within organizations in order to further refine, and eventually validate it.

One way to mobilize the complexity evaluation framework is to create a "Complexity Register" with which to support the collaborative identification of contextually relevant complexity factors and provide a store of data supporting system complexity evaluation. Developing such a tool, along with guidance for users, prompts to support complexity evaluation, and ensuring integration with other engineering management artifacts and risk management processes remains as further work. Such a mobilization would also require deployment and eventual validation.

Empirical evidence should be sought to determine if the expected value of conducting system complexity evaluation throughout a life cycle can be realized and at what cost. While gathering empirical data to support this is fraught with challenges, it is necessary in order to demonstrate the utility of system complexity evaluation.

9 | CONCLUSION

Despite efforts by scholars and practitioners to provide clarity on the term "system complexity," the term is still loaded with ambiguity and opacity, making any efforts to implement tools and processes for SoS complexity evaluation challenging. One particular tool is evaluated here informed by semistructured interviews with senior personnel within Thales Group.

While organizations may use such decision support tools to aid complexity evaluation, we argue that such tools may not deliver value unless they are embedded within an appropriate complexity evaluation framework. Such a framework must support structured thinking that respects the organizational context within which complexity evaluation sits.

A preliminary framework is introduced here, combining the three key features derived from analysis of the use case explored in this paper: being *collaborative*, *iterative*, and *progressive*. It is centered on an iterative five-step complexity evaluation process (identification of system complexity factors, collaborative evaluation of their impact, communication of the resultant shared understanding, planning mitigations, and implementing these mitigations) embedded within a larger organizational learning cycle that interrogates and progressively improves the process of complexity evaluation and monitors its net benefit.

Notwithstanding the significant challenges that currently exist, if organizations can establish the utility of complexity evaluation in this fashion, they stand to gain, at the very least, a greater awareness of likely risks for their SoS development projects, and, more optimistically, may articulate new accurate predictors of project outcomes.

DATA ACCESS STATEMENT

The data supporting this research (Introductory email, participant information sheet, participant consent form, interview questions, anonymized biography of participants, textual excerpts) are available at the University of Bristol data repository, data.bris, at https://doi.org/10.5523/bris.pji8xwa0q6ue27lcu8gp62k0q.[85]

ACKNOWLEDGMENTS

The authors would like to thank Jean-Luc Garnier and Dave Harvey within Thales Group for their guidance and support. The authors would also like to thank the anonymous reviewers for their insightful feedback and contributions.

ORCID

Matthew W. Potts https://orcid.org/0000-0002-4266-0862

REFERENCES

1. Hartmann R, Belhoff B, Oster C, et al. *A World In Motion, Systems Engineering Vision 2025 [Journal Article]*. San Diego, CA: INCOSE; 2014.
2. Punzo G, Tewari A, Butans E, et al. Engineering Resilient Complex Systems: The Necessary Shift Toward Complexity Science. *IEEE Syst J*. 2020;14(3):3865–3874. https://doi.org/10.1109/JSYST.2019.2958829.
3. Mayfield M, Punzo G, Beasley R, Clarke G, Holt N, Jobbins S. Challenges of complexity and resilience in complex engineering systems. ENCORE Network+ White Paper. 2018.
4. Suh NP. Complexity in engineering. *CIRP Ann*. 2005;54(2):46–63.
5. Gorod A, White BE, Ireland V, Gandhi SJ, Sauser B. *Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering*. Boca Raton, FL: CRC Press; 2014.
6. Sheard S, Cook S, Honour E, et al. *A complexity primer for systems engineers [White paper]*. San Diego, CA: INCOSE; 2015. [Available online at https://www.incose.org/docs/default-source/ProductsPublications/a-complexity-primer-for-systems-engineers.pdf.]
7. Sheard SA. Systems engineering complexity in context. *INCOSE Int Symp*. 2013;23(1):1145–1158. [Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2334-5837.2013.tb03077.x.]
8. Sheard SA, Mostashari A. A complexity typology for systems engineering. *INCOSE Int Symp*. 2010;20(1):933–945. [Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2334-5837.2010.tb01115.x.]
9. Bullock S, Cliff D. Complexity and emergent behaviour in ICT systems. Technical Report HP-2004-187, Hewlett-Packard Labs; 2004. [This report was commissioned by the Foresight Programme of the UK's Office of Science and Technology (DTi). However, its findings are independent of government and do not constitute government policy].
10. ISO/IEC/IEEE International Standard. *Systems and Software Engineering: System of Systems (SoS) Considerations in Life Cycle Stages of a System*. Piscataway, New Jersey: ISO/IEC/IEEE 21839:2019(E). 2019:1–40.
11. Boardman J, Sauser B. System of systems—The meaning of of. In: *IEEE/SMC International Conference on System of Systems Engineering*. IEEE Press; 2006:118–126.
12. Gorod A, Sauser B, Boardman J. System-of-systems engineering management: A review of modern history and a path forward [Journal Article]. *IEEE Syst J*. 2008;2(4):484–499.
13. Keating C, Rogers R, Unal R, et al. System of systems engineering. *Eng Manag J*. 2003;15(3):36–45.
14. Keating CB, Gheorghe AV. Systems thinking: Foundations for enhancing system of systems engineering. In: *2016 11th System of Systems Engineering Conference (SoSE)*. IEEE; p. 1–6.
15. Dahmann JS. Systems of systems characterization and types [Journal Article]. *Systems of Systems Engineering for NATO Defence Applications (STO-EN-SCI-276)*. 2015; p. 1–14.
16. Luzeaux D, Wippler JL. *Complex Systems and Systems of Systems Engineering*. Hoboken, NJ: John Wiley & Sons; 2013.
17. Dahmann J, Baldwin K. Implications of systems of systems on system design and engineering. In: *2011 6th International Conference on System of Systems Engineering (SoSE)*. IEEE;. 131–136.
18. Jamshidi M. System of systems engineering—New challenges for the 21st century [Journal Article]. *IEEE Aero El Sys Mag*. 2008;23(5):4–19.
19. Abbott R. Open at the top; open at the bottom; and continually (but slowly) evolving. In: *IEEE/SMC International Conference on System of Systems Engineering*. IEEE Press; 2006. 41–46.
20. Cocks D. How should we use the term "System of Systems" and why should we care? In: *INCOSE International Symposium*. Vol. 16. Wiley Online Library; 2006. 427–438.
21. White BE. On leadership in the complex adaptive systems engineering of enterprise transformation. *J Enterp Transform*. 2015;5(3):192–217.
22. McCarter BG, White BE. *Leadership in Chaordic Organizations*. Boca Raton, FL: CRC Press; 2016.
23. McCarter BG, White BE. In: Rubin CRJM S, ed. *Emergence of SoS, Sociocognitive Aspects*. Boca Raton, FL: CRC Press; 2009.
24. Baldwin KJ, Lucero S. Defense system complexity: Engineering challenges and opportunities. *ITEA J Test Eval*. 2016;37(1):10–16.
25. Griffin PM, Nembhard HB, DeFlitch CJ, Bastian ND, Kang H, Muñoz DA. *Healthcare Systems Engineering*. Hoboken, New Jersey: John Wiley & Sons; 2016.
26. Walden DD, Roedler GJ, Forsberg K, Hamelin RD, Shortell TM. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Hoboken, NJ: Wiley; 2015.
27. Kossiakoff A, Sweet WN, Seymour SJ, Biemer SM. *Systems Engineering Principles and Practice*. Vol. 83. Hoboken, NJ: John Wiley & Sons; 2011.
28. ISO/IEC/IEEE International Standard - Systems and software engineering – System life cycle processes. ISO/IEC/IEEE 15288 First edition 2015-05-15. 2015 May; p. 1–118.
29. Mostashari A, Sussman JM. A framework for analysis, design and management of complex large-scale interconnected open sociotechnological systems. *Int J. Decision Supp Syst Technol. (IJDSST)*. 2009;1(2):53–68.
30. Sheard SA. *Assessing the Impact of Complexity Attributes on System Development Project Outcomes*. Hoboken, NJ: AAI; 2012.
31. Ameri F, Summers JD, Mocko GM, Porter M. Engineering design complexity: An investigation of methods and measures. *Res Eng Des*. 2008;19(2-3):161–179.
32. Alkan B, Vera DA, Ahmad M, Ahmad B, Harrison R. Complexity in manufacturing systems and its measures: A literature review. *Eur J Ind Eng*. 2018;12(1):116–150.
33. Efatmaneshnik M, Ryan MJ. A general framework for measuring system complexity. *Complexity*. 2016;21(S1):533–546.
34. Lloyd S. Measures of complexity: A nonexhaustive list. *IEEE Control Syst Mag*. 2001;21(4):7–8.
35. Ladyman J, Lambert J, Wiesner K. What is a complex system? *Eur J Phil Sci*. 2013;3(1):33–67.
36. Sinha K, de Weck OL. Empirical validation of structural complexity metric and complexity management for engineering systems. *Syst Eng*. 2016;19(3):193–206.
37. Sinha K, et al. Structural complexity and its implications for design of cyber-physical systems. *Massachusetts Institute of Technology*; 2014.
38. Sinha K, de Weck OL. A network-based structural complexity metric for engineered complex systems. In: *2013 IEEE International Systems Conference (SysCon)*; 2013. p. 426–430.

39. Broniatowski DA, Moses J. Measuring flexibility, descriptive complexity, and rework potential in generic system architectures. *Syst Eng*. 2016;19(3):207–221.
40. Bar-Yam Y, Jeffrey RS. *Dynamics Of Complex Systems*. New York City, New York: Addison-Wesley; 1997. Available from: https://books.google.co.uk/books?id=LH_wAAAAMAAJ.
41. Fischi J, Nilchiani R, Wade J. Dynamic complexity measures for use in complexity-based system design. *IEEE Syst J*. 2017 11(4):2018–2027.
42. Sillitto H. *Architecting Systems: Concepts, Principles and Practice*. London: College Publications; 2014.
43. Potts MW, Sartor P, Johnson A, Bullock S. A network perspective on assessing system architectures: Foundations and challenges. *Syst Eng*. 2019;22(6):485–501.
44. Potts MW, Sartor PA, Johnson A, Bullock S. A network perspective on assessing system architectures: Robustness to cascading failure. *Syst Eng*. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/sys.21551.
45. Olechowski AL, Eppinger SD, Joglekar N, Tomaschek K. Technology readiness levels: Shortcomings and improvement opportunities. *Syst Eng*. 2020;23(4):395–408.
46. Watson M, Anway R, McKinney D, Rosser LA, MacCarthy J. Appreciative methods applied to the assessment of complex systems. In: *INCOSE International Symposium*. vol. 29.Wiley Online Library; 2019. p. 448–477.
47. Simpson JJ, Simpson MJ. System of systems complexity identification and control. In: *2009 IEEE International Conference on System of Systems Engineering (SoSE)*. IEEE; 2009. p. 1–6.
48. Simpson J, Simpson M. Complexity reduction: A pragmatic approach. *Syst Eng*. 2011;14(2):180–192.
49. Martin JN. The Seven Samurai of Systems Engineering: Dealing with the complexity of 7 interrelated systems. *INCOSE International Symposium*. 2004;14(1):459–470.
50. Mirza E, Ehsan N. Quantification of project execution complexity and its effect on performance of infrastructure development projects. *Eng Manag J*. 2017;29(2):108–123.
51. Ellinas C, Allan N, Durugbo C, Johansson A. How robust is your project? From local failures to global catastrophes: A complex networks approach to project systemic risk. *PLOS One*. 2015 11;10(11): 1–21.
52. Ellinas C, Allan N, Johansson A. Toward project complexity evaluation: A structural perspective. *IEEE Syst J*. 2018 12(1):228–239.
53. Rouse WB. Complex engineered, organizational and natural systems. *Syst Eng*. 2007;10(3):260–271.
54. Tolk A, Diallo S, Mittal S. *Complex Systems Engineering and the Challenge of Emergence*. Hoboken, NJ: John Wiley & Sons, Ltd.; 2018:78–97.
55. Ham D, Park J, Jung W. A framework-based approach to identifying and organizing the complexity factors of human-system interaction. *IEEE Syst J*. 2011;5(2):213–222.
56. Stevens R. Profiling complex systems. In: *2008 2nd Annual IEEE Systems Conference*. IEEE; 2008. p. 1–6.
57. Stacey R. *Tools and Techniques of Leadership and Management: Meeting the Challenge of Complexity*. London: Routledge; 2012.
58. Dahmann J. System of systems pain points. In: *INCOSE International Symposium*. Vol. 24. Wiley Online Library; 2014. p. 108–121.
59. Cook SC, Pratt JM. Typology dimensions for classifying SoSE problem spaces. In: *2016 11th System of Systems Engineering Conference (SoSE)*; 2016. p. 1–6.
60. McCabe TJ. A complexity measure. *IEEE Trans Softw Eng*. 1976;(4):308–320.
61. Kafura D, Reddy GR. The use of software complexity metrics in software maintenance. *IEEE Trans Softw Eng*. 1987;(3):335–343.
62. Summers JD, Shah JJ. Mechanical engineering design complexity metrics: Size, coupling, and solvability. *J Mech Des*. 2010;132(2):021004.
63. Min G, Suh ES, Hölttä-Otto K. System architecture, level of decomposition, and structural complexity: Analysis and observations. *J Mech Des*. 2016;138(2):021102.
64. Kim G, Kwon Y, Suh ES, Ahn J. Correlation between architectural complexity of engineering systems and actual system design effort. *J Mech Des*. 2017;139(3):034501-1–034501-5. https://doi.org/10.1115/1.4035319.
65. Dwyer M, Cameron B, Szajnfarber Z. A framework for studying cost growth on complex acquisition programs. *Syst Eng*. 2015;18(6):568–583.
66. Allaire D, He Q, Deyst J, Willcox K. An information-theoretic metric of system complexity with application to engineering system design. *J Mech Des*. 2012;134(10):100906-1–100906-10. https://doi.org/10.1115/1.4007587.
67. Snowden DJ, Boone ME. A leader's framework for decision making. *Harv Bus Rev*. 2007;85(11):68.
68. White BE. Systems engineering activities (SEA) profiler. In: *8th Conference on Systems Engineering Research (CSER)* March 17–19, 2010, Hoboken, NJ; 2010. Paper Number 1569264775.
69. Beale D, Tryfonas T. Exploration of the complex ontology. In: *INCOSE International Symposium*. Vol. 28. Wiley Online Library; 2018. p. 1549–1563.
70. Beale D, Tryfonas T, Young M. Evaluating approaches for the next generation of difficulty and complexity assessment tools. In: *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*;. p. 227–233.
71. White BE. Complex adaptive systems engineering (CASE). *IEEE Aero El Sys Mag*. 2010;25(12):16–22.
72. White B. A complex adaptive systems engineering (CASE) methodology: The ten-year update. In: *2016 Annual IEEE Systems Conference (SysCon)*. IEEE; 2016. 1–8.
73. White BE. On principles of complex systems engineering complex systems made simple. *INCOSE International Symposium*. 2011;21(1):1590–1844.
74. Sillitto H, Griego R, Arnold E, et al. A fresh look at systems engineering—What is it, how should it work? In: *INCOSE International Symposium*. Vol. 28. Wiley Online Library; 2018. 955–970.
75. Blockley D, Godfrey P. *Doing It Differently: Systems for Rethinking Infrastructure*. London: ICE Publishing; 2017.
76. Checkland P. *Systems Thinking, Systems Practice: Includes a 30-Year Retrospective*. Chichester, UK: Wiley; 1999.
77. Gorod A, Hallo L, Ireland V, Gunawan I. *Evolving Toolbox for Complex Project Management*. Boca Raton, FL: CRC Press; 2019.
78. Garnier JL. *Complexity Profiler*; 2013. Paris, France: Thales Group Internal.
79. Potts M, Sartor P, Johnson A, Bullock S. Deriving key features of a system-of-systems complexity evaluation framework from an industrial case study analysis. In: *2019 International Symposium on Systems Engineering (ISSE)*; 2019. 1–8.
80. Burns D. How change happens: The implications of complexity and systems thinking for action research. *Sage Handbook Action Res*. 2015;3:434–445.
81. Bradbury H. *The Sage Handbook of Action Research*. Thousand Oaks, CA: Sage; 2015.
82. Boardman J, Sauser B. *Systems Thinking: Coping With 21st Century Problems*. Boca Raton, FL: CRC Press; 2008.
83. Braun V, Clarke V. Using thematic analysis in psychology. *Qual Res Psychol*. 2006;3(2):77–101.
84. Patton MQ. *How To Use Qualitative Methods in Evaluation*. Thousand Oaks, CA: Sage; 1987.
85. Potts M, Bullock S. Data from Study_How does complexity affect the successful engineering of systems throughout a system lifecycle; 2020. Available at https://doi.org/10.5523/bris.pji8xwa0q6ue27lcu8gp62k0q.

86. Potts M, Sartor P, Johnson A, Bullock S. Assaying the importance of system complexity for the systems engineering community. Available at https://onlinelibrary.wiley.com/doi/abs/10.1002/sys.21550.
87. Chidamber SR, Kemerer CF. A metrics suite for object oriented design. *IEEE Trans Softw Eng*. 1994;20(6):476–493.
88. The Open Group. *TOGAF® Version 9.1*. 's-Hertogenbosch, Netherlands: Van Haren Publishing; 2011.
89. Checkland P, Poulter J, Poulter J. *Learning for Action : A Short Definitive Account of Soft Systems Methodology and Its Use for Practitioner, Teachers, and Students*. Hoboken, NJ: Wiley; 2006.
90. Risk management: Guidelines. Geneva, CH: International Organization for Standardization; 2018.
91. Rebovich Jr. G, White BE. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL: CRC Press; 2016.

AUTHOR BIOGRAPHIES

Matthew W. Potts is a PhD research student at the University of Bristol, United Kingdom, within the Aerospace Engineering Department. He received the BEng degree from the University of Southampton, United Kingdom, in Electromechanical Engineering (2008–2011). He served in various roles in the Royal Air Force (2011–2016) as an engineering officer and has worked as an independent consultant. He is a registered incorporated engineer with the Institute of Engineering and Technology.

Angus Johnson is the Thales Head of Systems Research and the Industry lead on the Thales–Bristol Partnership in Hybrid Autonomous Systems Engineering. He has a background in radar spectrum sensing and multifunction RF systems engineering.

Seth Bullock is Toshiba Chair in Data Science and Simulation in the Department of Computer Science at the University of Bristol, United Kingdom. He holds a first degree in cognitive science (1993) and a doctorate in evolutionary simulation modeling (1997) from Sussex University, UK. His research expertise lies in complex systems simulation modeling. He has won over 28m in research and infrastructure funding in this area and has undertaken consultancy for the UK Government on complexity in ICT (2004) and financial systems (2012). He publishes in international peer-reviewed journals spanning health, social science, economics, biology, architecture, engineering, environmental science, computing, and physics. He was twice elected to the Board of Directors of the International Society for Artificial Life and serves on the editorial boards of the MIT Press journals *Adaptive Behavior*, *Artificial Life*, and *Frontiers in Robotics & AI*. He has given invited keynote lectures in London, Paris, Athens, Melbourne, Madrid, Granada, and Tokyo.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: Potts MW, Johnson A, Bullock S. Evaluating the complexity of engineered systems: A framework informed by a user case study. *Systems Engineering*. 2020;23:707–723. https://doi.org/10.1002/sys.21558