Complexity Measures to Predict System Development Project Outcomes

Sarah A. Sheard, Ph.D.[1]
sarah.sheard@gmail.com
Stevens Institute of Technology
Castle Point on Hudson
Hoboken, New Jersey 07030-5991 USA

Ali Mostashari, Ph.D.
amostash@stevens.edu
Stevens Institute of Technology
Castle Point on Hudson
Hoboken, New Jersey 07030-5991 USA

Abstract. While it is broadly accepted that complexity makes system development harder, there is no concrete understanding of which types of complexity have the most significant impact. Looking beyond current literature which describes complexity or measures the complexity of a system, this research seeks complexity measures that directly affect development project outcomes: project cost overrun, project schedule delay, and system performance shortfalls. A set of complexity measures was developed based on a comprehensive literature analysis and ranking via a trade study. The effect of those measures on project outcome was studied for 75 systems development efforts, primarily in the aerospace and defense sector. The findings indicate that among the dozens of complexity measures discussed in the literature, the three measures with the most significant impacts on development outcomes in these projects were: number of hard-to-meet requirements, degree of cognitive fog, and stability of stakeholder relationships.

Introduction

In 2009 DARPA requested industry participation in an effort to improve the way technological systems are built (Eremenko, 2009). One intent was to elicit better complexity metrics than "Part Count + Source Lines of Code." Reports from twelve year-long studies proposed complexity measures ranging from information entropy (applied to two resistors and three voltages) (Willcox, Allaire, Deyst, He, & Sondecker, 2011) to computations of predicted labor and material costs for massive development efforts. (Stuart, Mattikalli, DeLaurentis, & Shah, 2011) It is clear that respondents (and in fact, literature from industry, government and academia) have not come to a consensus about how to measure systems engineering complexity. Furthermore, very little has been studied about whether such complexity measures would be able to predict project success.

In this work, which is more fully discussed in the Ph.D. dissertation of one of the authors (Sheard 2012), we studied 75 completed system development projects by means of retrospective survey of senior systems engineers and project managers. We asked 39 complexity-related questions about these projects, plus six outcome-related questions, a few demographic questions, and questions about project management methods. We tested correlation among the resulting variables using t-tests (the set of responses to one question is a variable). Three of the variables correlated to project cost, project schedule, and system

[1] Now at the Software Engineering Institute, Carnegie Mellon University, Pittsburgh PA.

performance. Twenty others correlated to some but not all of cost, schedule, and system performance.

In this paper we show that:

- Today's literature lacks substantial conclusions on how best to measure complexity in systems engineering.
- Measurement of complexity for systems engineering could have useful benefits.
- It is possible to demonstrate qualitatively that some "complexity" measurements are more strongly correlated to project outcome than others, and in fact differences in outcome appear that are statistically significant.
- A large number of further directions can be taken related to this research.

The research began with two working hypotheses that provided guidance for some of the methodological decisions: 1) Complex projects, however complexity is defined, will have less successful outcomes (i.e., they will have more cost overrun, schedule delay, and performance shortfall than less complex projects) and 2) Most complexity measures will not directly correlate with project outcomes, but some will.

After the data was collected, a statistically testable hypothesis was generated and tested. The results are presented in the "Correlation to project outcomes" section.

Background: Literature Review

No one definition of complexity successfully captures all aspects of a topic that has many interlocking facets; in other words, a topic that is complex. Complexity has been described as uncertainty (Clara, 2007), interrelated dynamics such as coupled nonlinear oscillators (Strogatz, 2003), chaotic regions of activity, emergent behaviors at the top level that arise from generative rules at a lower level (Phelan 2001), and significant amounts of information processing (Surowiecki, 2004). McCabe describes the complexity of software in terms of the number of cycles the flow chart contains and shows how to measure it using more easily-counted nodes and edges (McCabe, 1976). The variety of meanings of complexity is daunting.

Complexity is most often considered in terms of the properties of the system being built: how many pieces of hardware does it contain, or how many interfaces must it include? Simon (1996) suggests that complexity is not inherent in the system itself, but rather in its representation: "How complex or simple a structure is depends critically upon the way in which we describe it. Most of the complex structures found in the world are enormously redundant, and we can use this redundancy to simplify their description. But to use it, to achieve the simplification, we must find the right representation." Warfield (2001) goes a step further and claims that complexity is not at all a property of a thing but solely a property of the human's conception of it. "Complexity is the frustration that occurs when people cannot understand a problematic situation that is of great importance to them." While most practicing systems engineers have experienced this frustration, it seems intuitive that some difficulty arises from attributes of the system, the project, and the environment, and not all important aspects of complexity are psychological. Furthermore, prevention of such difficulty requires taking a step back and addressing its causes, namely the objective rather than only subjective forms of complexity (Sillitto, 2009).

Complexity can therefore refer to any of a large number of characteristics of a system, or the representation of the system, or the emotional experience of a human. Which of these might be useful for systems engineering?

Systems engineering became popular in the 1960s (Hall, 1962; Hughes, 1998) because it breaks up large problems into smaller, solvable ones (e.g., the Vee model) (Haskins, 2006) and makes the implications of different decisions explicit (e.g., modeling) so that the best choice can be made. Hall observed, "Most writers…emphasize increasing complexity as the principal causative factor." Thus systems engineering was created to manage complexity in the development of technological systems.

More recently, capability models (Bate et al., 1995, Software Engineering Institute, 2010) suggest that the purpose of systems engineering is not so much to reduce complexity as to optimize the solution that can be built for the budget and schedule. Others suggest systems engineering's purpose is to make an enterprise more efficient than its competitors. Even so, complexity is blamed for problems with product life cycle costs, difficulty getting engineering changes made (General Accounting Office, 1994), difficulty in servicing (leading to many failure modes), management and logistical problems in the supply chain (Press Association, 2011) and the need for a costly design process (Ameri, Summers, Mocko, & Porter, 2008).

Since in the past forty years a new science has evolved that studies complexity (Lawrimore, 2004; Waldrop, 1992), it is reasonable to see whether usable principles for the engineering of complex systems can be derived from the science. (Sheard, 2009) Simon (1996) describes the motivation for the current popularity of complexity studies as "the growing need to understand and cope with some of the world's large-scale systems—the environment, for one, the world-wide society that our species has created, by itself, and organisms, for a third."

Some ideas for how to incorporate complex systems ideas into today's engineering practices are already published. The numerical modeling of complex adaptive systems (Miller & Page, 2007) and theoretical fields such as nonlinear dynamics and networks (Newman, Barabási, & Watts, 2006) are potentially superior to earlier methods. Boisot and McKelvey (2007) show how analysis and management techniques such as risk assessment should change when distributions in the real world resemble power laws more than Gaussian distributions. DuPreez and Smith (2004) apply complexity theory to project management of information and to communications technology system development. Sheard and Mostashari (2009) provide principles of complex systems and show how they should be applied to systems engineering, at a general level.

Although none of these recommendations are widely incorporated into systems engineering practice at this time, there are suggestions that industry and government are interested in measuring complexity. The DARPA effort mentioned above is one source; another is the Systems Engineering Leading Indicators Guidebook effort (Roedler, Rhodes, Schimmoller, & Jones, 2010), which sought a leading indicator of complexity for Version 2, but was unable to find a suitable mature candidate.

Complexity measurement for systems engineering must be implemented on top of a basic foundation of systems engineering measurement. Significant work has been done in this field by INCOSE (Roedler & Jones, 2005) and PSM (McGarry, 2002). What is needed is statement of the exact data quantities (say, hours charged or number of parts) to be measured, when they should be measured (day of the week and program phase), how they should be measured (units, and including or not including overhead), as well as algorithms for combining such raw data into indicators for decision making. (ISO/IEC, 2007) If any of these questions have been answered for any systems engineering complexity measurement, they are not yet generalized and usable by others.

Essentially, complexity in systems engineering is not well defined, and measurement of complexity is addressed to date only from a series of unique perspectives. Most conceptions of complexity measures in the theoretical literature could not be applied to systems engineering easily, and many concepts of complexity within engineering do not have a close tie to theory.

Benefits of Measuring Complexity

If there were a well-understood way to quantify the complexity of a design or a development effort, engineers could identify troublesome areas, quantify risk, make predictions, and provide inputs to decision analyses such as trade studies. Engineers could express quantitatively the relative complexities of a number of different designs, and trade study analyses could include complexity measurements as well as cost, schedule, and performance. Measurements could help project managers apply resources to the riskiest areas and better mitigate risks. Acquirers could select projects, contractors, and designs that are likely to succeed because they are lower in the types of complexity that matter, and could make acquisition and project decisions consistent with improved project outcomes.

Methodology

The methodology is split into four phases. These are described in more detail by Sheard (2012).

Phase 1. Measure collection.

This research first interpreted the literature of complexity, defining a complexity taxonomy that breaks down types of complexity into structural characteristics (size, connectivity, and inhomogeneity), dynamic characteristics (short- and long-term), and sociopolitical characteristics. (Sheard & Mostashari, 2010) This typology was augmented by an understanding of entities whose complexity should be assessed, namely System, Project, Environment, and possibly Cognitive. (Sheard & Mostashari, 2011)

In order to determine measures that may apply to systems engineering, over 300 expressions of what complexity means were collected from complex systems sciences journals and books, and from systems engineering papers and books. A taxonomy was created as reported by Sheard and Mostashari (2010), summarized in Tables 1 and 2. Specific measures in each area of the taxonomy were proposed and then down-selected to a set of 39, using criteria shown in Table 3.

Table 1. Six types of complexity

Six Types	
Structural: Size	Number of elements, number of instances, total cost, total number of requirements
Structural: Connectivity	Number of connections, density of connections, strengths of relationships, amount of conflict
Structural: Inhomogeneity	Number of types of entities, number of types of relationships, number of different areas within a space, diversity of sizes of elements or contractors or stakeholders
Dynamic: Short-term	Safety-criticality, tendency to blow up in operational time frame, seriousness of consequences of a mishap
Dynamic: Long-term	Evolution of purpose of an item, co-evolution of a variant and its environment, how much different the next iteration of a system might be
Socio-political	Fraction of stakeholder interests that are political, amount of disagreement among stakeholders, number of layers of management, changes of opinion of management or stakeholders, number of different cultures working together on a project.

Table 2. Four entities that can be more or less complex

Four Entities	
[Technical] System being built	Product, system, system-of-systems, tank, squadron, database, sensor, software algorithm.
Project or organization doing the building	Project, organization, program, tasks, team
Environment, both external systems and people	Customers, buyers, market, external technological system, future systems that need to interface with product
Cognitive: capacity of humans to understand, build and operate the system.	Learning curve, uncertainty, confusion, operator skill set

Table 3. Trade-study criteria used to down-select potential measures

Priority	Weight	Criterion
A	30	Projects will have this data already
A	30	It is important whether this measure correlates or does not correlate
B	10	Applies to projects broadly (e.g., software, weapons, civil)
B	10	Existing and real
B	6	Covers an otherwise difficult-to-cover category (DL, SI, diversity)
B	6	Reproducible
C	2	Uses items of interest to projects
C	2	Basic to complexity and/or chaos (nonlinearity, connectivity,
C	2	Basic to system dynamics behavior
C	2	Basic to uncertainty

Phase 2, Survey creation. Questions (including answer scales) were prepared for a survey of finished projects. For the purpose of clarifying progress through the survey, the questions were sorted into four groups: Project basics (such as domain, years, and total cost), Demographic and respondents, Outcomes (cost, schedule, performance, etc.), and Complexity measures (subdivided by entity: System, Project, and Environment; a single Cognitive question was included within the Project questions). The complexity measures touched on all six types of complexity for the System, Project, and Environment entities (see Table 4).

Table 4. Types and Entities of Survey Questions*

	Number of Survey Questions		
Type Entity:	System	Project	Environment
Structural Size: How large/ how many pieces?	3	5	1
Structural Connectivity: How many connections?	3	1	2
Structural Inhomogeneity: What kind of structure?	1	1	1
Dynamic short term: How rapid must the system react?	1	1	1
Dynamic long term: How much evolving is occurring?	1	1	3
Socio-political: How much socio-political complexity?	1	2	2

* One question about the Cognitive entity was included with the Project questions.

Survey testing. The questions and answer choices were tested by several engineers from a Delphi group (Linstone & Turoff, 1975) of senior systems engineers. Adjustments were made to reduce ambiguity prior to developing the survey in the Survey Monkey tool. The actual survey was tested by the authors first and then by the first few respondents; adjustments were very minor in this last test.

Phase 3, Survey administration. Senior systems engineers were asked to commit to filling out the survey. Additional respondents were sought after the initial set of committed respondents returned too few surveys.

Phase 4, Analysis. After data cleaning, analysis consisted of splitting each question into two groups (more and less complex responses) and identifying each question's polarity, as described below.

Splitting into two groups for t-tests. The t-test splits all projects into two groups, one group consisting of those projects that had higher-numbered answers on a question called the "split variable" and the other group consisting of projects that had lower-numbered answers. The means of the two groups' answers to all the other questions are then compared to see if the split variable makes a difference to each other variable. For example, projects with more requirements would be put into one group and projects with fewer requirements into another group, to see if the number of requirements has an effect on the outcome variables such as cost overrun, schedule overrun, and performance shortfall. Means were compared for all variables, including the other complexity variables and project management techniques used. (Linear means of the response codes were used, with 1-5 available for most questions; this approximated geometric means for the exponential scales.)

Variable polarities. One question to be answered was whether complexity can be helpful. For example, natural systems get more complex as they get more capable, and more capable systems evolve because they better compete for resources. Is such a phenomenon evident in systems engineering? Although the hypothesis is that complexity leads to problems in general, does the data contradict this for some types of complexity?

In order to examine this, "more complex" had to be defined for each question. This determines the "variable polarity." If the responses to two questions indicated that complexity goes up in both together, then the two questions are "congruent" and the intersection of these two variables in a matrix will be colored green. If one goes up while the other goes down (say the more complex projects have better outcomes), then the box will be colored red and this will be obvious.

For many questions, determining which answer represented lowest complexity and which was highest complexity was easy. The main criteria used to select one end of a variable's responses as "higher complexity" were: larger size, more connectivity, more inhomogeneity, more change or faster change, and more difficult socio-political environment (these are the six types). This rule applied to complexity variables.

For outcome variables (project cost, project schedule, and system performance, among others), the "higher complexity" end was selected as the poorer outcome end. This was per the first assumption, which would be proven wrong if many red boxes appeared.

For some questions, however, it is not obvious which end of the spectrum should be considered higher complexity. Decisions made for these variables, and polarity conclusions, are described in (Sheard 2012).

Correlation to project outcomes

Of the 39 complexity measures, three correlated in a statistically significant manner to all three outcome measures (project cost overrun, project schedule overrun, and system performance shortfall).

Table 5 shows the wording of these three complexity measures (variables) and the outcome measures (variables).

Table 5. Top Three Complexity Variables and Outcome Variables

		Answer Choices					
#	Variable Name, Question	1	2	3	4	5	6
Complexity Variables							
16d	Requirements, Difficult Approximately how many system-level requirements did the project have initially? Difficult requirements are considered difficult to implement or engineer, are hard to trace to source, and have a high degree of overlap with other requirements. How many system requirements were there that were Difficult?	1-10	10-100	100-1000	1000-10,000	Over 10,000	
32	Cognitive Fog 'The project frequently found itself in a fog of conflicting data and cognitive overload.' Do you agree with this statement?	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	
38f	Stakeholder Relationships "Where did your project fit, on	Relation-ships	New Rela-tion-	Resist-ance to			

		Answer Choices					
#	Variable Name, Question	1	2	3	4	5	6
	a scale of Traditional, Transitional, or Messy Frontier, in the following eight attributes?” 38f. “Stakeholder relationships: 1: Relationships stable; 2: New relationships; 3: Resistance to changing relationships.	stable	ships	Chang-ing Rela-tion-ships			
Outcome Variables							
9	**Cost Overrun** At the point of finishing, how much did the project cost, compared to the initially predicted cost for delivery?	Under cost	At cost, +/- 5%	5-20% over plan	20-50% over	50-100% over	More than 100% over plan
10	**Schedule Delay** At the point of finishing, how long had the project taken, compared to the initially scheduled development time?	Ahead of schedule	On time within 5%	5-20% late	20-50% late	50-100% late	More than 100% late
11	**Performance Shortfall** At the point of finishing, how was the project performance, compared to the initially specified performance? (Please consider the average performance of *mission critical* features, and add any qualifiers in Notes.)	Higher than specified	Same as specified, within 5%	Low by 5-20% (fewer features or waived require-ments)	Low by 20-50%	Low by more than 50%, or project was cancelled	

Table 6 shows the statistical significance of these correlations. The difference in mean cost overrun between the programs of less than 100 difficult requirements and programs of more than 100 difficult requirements was statistically significant at the $p<0.001$ level (called “much more likely” below). The difference between these two groups in schedule delay and performance shortfall was statistically significant as well, but at the $p<0.05$ level (called “more likely” below; only these two levels of significance were discussed).

What this means is that projects which had many difficult requirements according to the COSYSMO definition (Valerdi, 2008) were much more likely to have cost overruns and were also more likely to have schedule delays and performance shortfall than projects that had fewer difficult requirements (as shown in Table 5). Projects that “frequently found [themselves] in a fog of conflicting data and cognitive overload” were much more likely to have performance shortfall and were also more likely to have cost overruns and schedule delay than projects that did not have this cognitive fog. Projects whose stakeholder relationships were changing and that were experiencing resistance to that change were more likely to have problems in all three outcomes than projects whose stakeholders had stable relationships.

Table 6. Significance of correlations of top three complexity variables

Complexity Variable	Outcome Variable Cost Overrun	 Schedule Delay	 Performance Shortfall
Q16d—Requirements Difficult			
Low (Under 100) group mean	3.37	3.30	2.26
High (Over 100) group mean	5.00	4.64	3.60
p-value	0.00027	0.00165	0.00163
Significance	**p<0.001**	p<0.05	p<0.05
Q32—Cognitive Fog			
Low (D-SD) group mean	3.03	2.97	2.00
High (A-SA) group mean	3.89	4.11	3.53
p-value	0.0395	0.0120	0.00074
Significance	p<0.05	p<0.05	**p<0.001**
Q38f—Stakeholder Relationships			
Low (Stable) group mean	3.30	3.11	2.15
High (Resistance) group mean	4.50	4.19	3.27
p-value	0.0209	0.0243	0.0245
Significance	p<0.05	p<0.05	p<0.05

The goal of this research, to identify measures that help predict improved project outcomes, is achieved by showing that these three complexity measures correlate to project cost, project schedule, and system performance.

Additional Discussion

Continued analysis provided additional interesting information, which is not discussed here but can be found in (Sheard 2012). For example, there are 20 other measures that correlate to either system performance outcomes (did the system do what it was supposed to?) or to project cost and schedule outcomes (did the project proceed as expected?). For example, as the experience level of the project staff goes up, so does the performance of the system. Experienced people probably will neither cost less nor reduce delays, but they will be able to solve the difficult problems of the system and get it to work.

The project management methods used to improve outcomes, namely more planning and control, did not result in better outcomes, and in fact higher use correlated *negatively* with outcomes. However, a conclusion should not be drawn from this, since programs that were expected to do badly may have been given more direction to keep tight control on plans, biasing the high-control group to include more risky programs.

Figure 1 is provided as one possible causal chain that could start with the three complexity variables and end with cost, schedule, and performance problems. While this does not show the only reason there could be outcome problems from these complexity variables, it does show feasibility of the relationships.

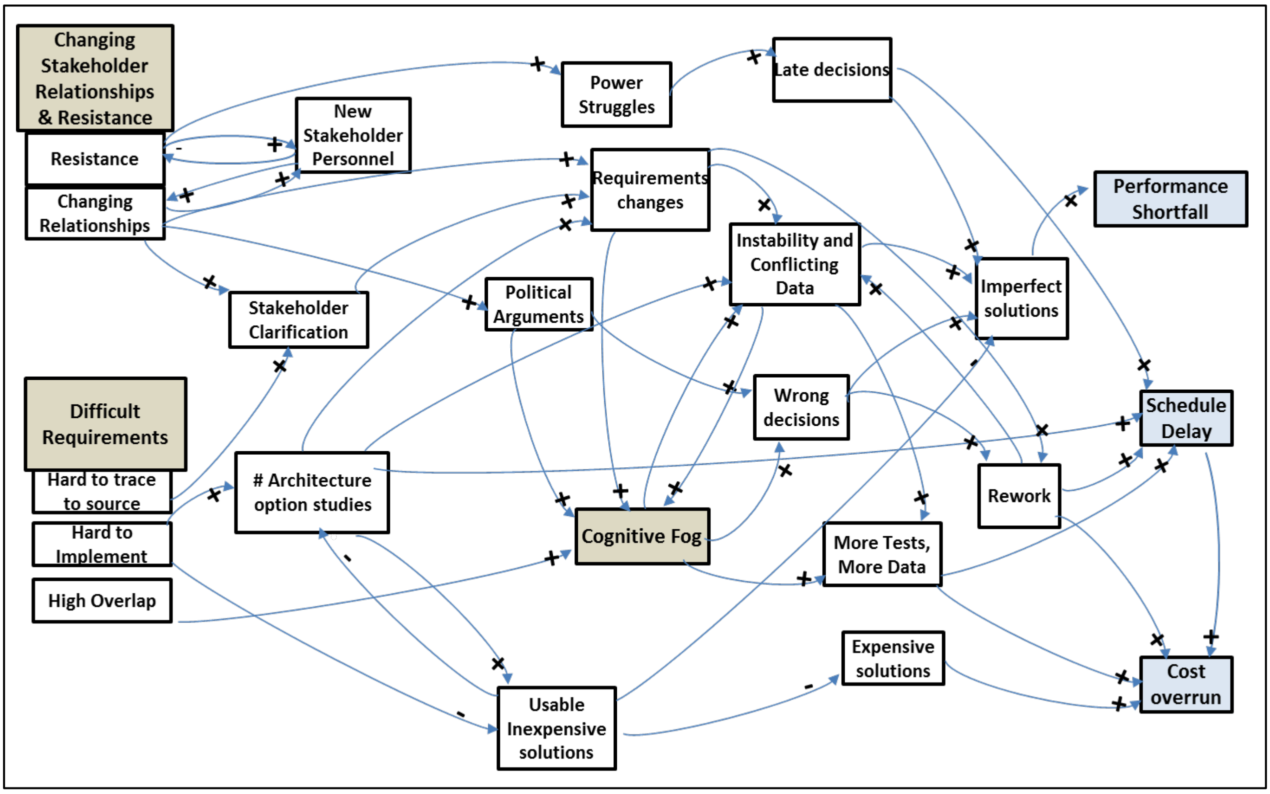

Figure 1. Connection of Complexity Variables to Outcomes

Further Research

Many directions can be taken related to this research. Questions that show correlation with outcomes ought to be refined by specifying data to be gathered and conditions for gathering it, according to templates such as in PSM (McGarry, 2002) and the Systems Engineering Leading Indicators guidebook (Roedler et al., 2010). Additional questions should be vetted, perhaps on acquisition, software, or technology maturity.

Methods to reduce complexity (Wade, Heydari, & Mostashari, 2010) ought to be tied to the kinds of complexity that they reduce. Similarly, project management methods should be analyzed to determine what kinds of complexity they reduce, which they allow, and what happens to problem space complexity that is eliminated from solutions and from plans…is it simply allocated to human operators to figure out, and how does that work?

Additional theoretical work should address the best representations of complexity both for measurement purposes and for understanding and management. In particular, "socio-political complexity" should be better defined in terms of what is important in systems development projects.

Heuristics can be derived that guide project managers in what kinds of complexity should be allowed and what kind should be resolved in each phase of a program. How should complexity be mitigated, and when? Are there "knees of the curve" where further reduction of complexity below, say, 100 difficult requirements is not as effective as reduction from 1000 to 100? Are there modifications to project processes or to systems engineering processes that should be instituted specifically to reduce complexity? Perhaps architecture complexity, requirements complexity, stakeholder complexity, and test complexity should be measured specifically and tested for relationships to project outcomes.

Conclusion

In this paper we have shown that:

- The question of how to measure complexity in systems engineering is largely unresolved today.
- Measurement of complexity for systems engineering could benefit engineers, project managers, and acquirers.
- It is possible to demonstrate qualitatively that some "complexity" measurements are more strongly correlated to project outcome than others, and in fact differences in outcome appear that are statistically significant.
- A large number of further directions can be taken related to this research.

The three measures that correlated with project cost, schedule, and performance outcomes were the number of difficult requirements (per the COSYSMO definition), the project's level of cognitive fog, and the stability of stakeholder relationships. All of the nine correlations were significant at the $p<0.05$ level and two were significant at the $p<0.001$ level.

References

Ameri, F., Summers, J. D., Mocko, G. M., & Porter, M. 2008. "Engineering design complexity: an investigation of methods and measures." *Res Eng. Design* 19, 161-170. doi: 10.1007/s00163-008-0053-2

Bate, R., Kuhn, D., Wells, C., Armitage, J., Clark, G., Cusick, K., Garcia, S., Hanna, M., Jones, R., Malpass, P., Minnich, I., Pierson, H., Powell, T., and Reichner, A. 1995. *A Systems Engineering Capability Maturity Model, Version 1.1.* Pittsburgh, PA (US): Software Engineering Institute.

Boisot, M., and Bill McKelvey. 2007. "Extreme events, power laws, and adaptation: towards an econophysics of organization." In Academy of Management Proceedings, vol. 2007, no. 1, pp. 1-6. Academy of Management, 2007.

Clara, M. M. 2007. *The brink of chaos*. College Station, TX: Virtualbookworm.com.

Software Engineering Institute. 2010. *CMMI ® for Development, Version 1.3*. Pittsburgh, PA: Carnegie Mellon University.

duPreez, L. J., & Smith, A. J. 2004. "The application of complexity theory in the development of large-scale ICT systems." Paper presented at the annual international symposium of the International Council on Systems Engineering, Toulouse, France. June 20-24.

Eremenko, P. 2009. Broad agency announcement, META. Federal Business Opportunities Solicitation Number DARPA-BAA-10-21. Retrieved 3/14/2013 from https://www.fbo.gov/?s=opportunity &mode=form&id=dd65d945682e6a95b474bd8d33e21660&tab=core&_cview=0.

General Accounting Office. 1994. "Advanced automation system: implications of problems and recent changes" (GAO/T-RCED-94-188). Retrieved 3/14/2013 from www.gao.gov/assets/110 /105482.pdf.

Hall, A. D. III. 1962. *A Methodology for Systems Engineering*. Princeton, NJ: Van Nostrand.

Haskins, C., Ed. (2006). *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Seattle Washington, INCOSE (International Council on Systems Engineering).

Hughes, T. 1998. *Rescuing Prometheus*. New York: Pantheon Books.

Lawrimore, E. W. B. 2004. "The new science of complexity vs. old science." Retrieved 8/3/2012 from http://www.codynamics.net/science.htm.

Linstone, H. A., & Turoff, M. 1975. "The Delphi method: techniques and applications." Reading, Mass.: Addison-Wesley Pub. Co.

McCabe, T. J. 1976. "A complexity measure." IEEE Transactions on Software Engineering SE-2(4), 308-320.

McGarry, J. 2002. *Practical software measurement: objective information for decision makers*. Boston, MA: Addison-Wesley.

Miller, J. H., & Page, S. E. 2007. *Complex adaptive systems: An introduction to computational models of social life.* Princeton, NJ: Princeton University Press.

Newman, M., Barabási, A.-L., & Watts, D. J. (Eds.). 2006. *The Structure and Dynamics of Networks*. Princeton NJ: Princeton University Press.

Phelan, S. E. (2001). What is complexity science, really? Emergence, 3(1), 130.

Press Association. 2011. "North Sea helicopter crash report says gearbox failed after maintenance error," The Guardian, 23 November. Retrieved 3/14/2013 from http://www.guardian.co.uk /uk/2011/nov/24/ north-sea-helicopter-crash-report.

Roedler, G. J., & Jones, C. 2005. "Technical measurement, INCOSE-TP-2003-020-01." Seattle, WA: INCOSE.

Roedler, G. J., Rhodes, D. H., Schimmoller, H., & Jones, C. 2010. *Systems Engineering Leading Indicators Guide, Version 2.0*. Cambridge, MA: Lean Advancement Initiative.

Sheard, S. A. 2009. "Complex adaptive systems in systems engineering and management." In A. P. Sage & W. B. Rouse (Eds.), *Handbook of Systems Engineering and Management* (pp. 1283-1318). Hoboken, New Jersey: Wiley.

———, 2012. "Assessing the impact of complexity attributes on system development project outcomes." PhD diss., Stevens Institute of Technology (Hoboken, NJ, US).

Sheard, S. A., & Mostashari, A. 2009. "Principles of complex systems for systems engineering." *Systems Engineering*, 12(4).

———, 2010. "A Complexity typology for systems engineering." Paper presented at the annual international symposium of INCOSE, Chicago IL (US), 12-15 July.

———, 2011. "Complexity in large-scale technical project management." *International Journal for Complexity in Leadership and Management*. doi: 10.1504/IJCLM. 2011.042550

Sillitto, H. G. 2009. “On Systems architects and systems architecting: some thoughts on explaining the art and science of system architecting.” Paper presented at the annual international symposium of INCOSE, Singapore, Singapore, 20-23 July.

Simon, H. A. 1996. *The sciences of the artificial*. Cambridge, Mass.: MIT Press.

Strogatz, S. H. 2003. *Sync : the emerging science of spontaneous order*. New York: Hyperion.

Stuart, D., Mattikalli, R., DeLaurentis, D., & Shah, J. 2011. “META II complexity and adaptability (Boeing): DARPA.” Retrieved 3/14/2013 from www.darpa.mil/uploadedFiles/Content/Our_Work/TTO/Programs/AVM/Boeing%20META%20Final%20Report.pdf.

Surowiecki, J. 2004. *The wisdom of crowds*. New York: Doubleday.

Valerdi, R. 2008. *The constructive systems engineering cost model (COSYSMO): Quantifying the costs of systems engineering effort in complex systems*. Saarbrücken, Germany: VDM Verlag.

Wade, J., Heydari, B., & Mostashari, A. 2010. “Some simple thoughts on complex systems.” *Research Colloquium on Complex Systems: Exemplar – Financial Systems*, Stevens Institute of Technology, 24 June.

Waldrop, M. M. 1992. *Complexity: the emerging science at the edge of order and chaos*. New York: Simon & Schuster.

Warfield, J. N. 2001. “Measuring complexity.” Integrative Sciences, Inc.

Willcox, K., Allaire, D., Deyst, J., He, C., & Sondecker, G. 2011. “Stochastic process decision methods for complex cyber-physical systems (MIT Willcox): DARPA.” Retrieved 3/14/2013 from www.darpa.mil/.../Content/Our_Work/TTO/Programs/AVM/MIT%20Willcox%20META%20Final%20Report.pdf.

Biography

Sarah Sheard, Fellow of INCOSE and of the Lean System Society, earned INCOSE’s 2002 Founder’s Award and the CSEP certification. Dr. Sheard, a member of INCOSE since 1992, has served in both technical and administrative leadership roles. The most famous of her 40+ papers are *Twelve Systems Engineering Roles,* the *Frameworks Quagmire* and *Principles of Complex Systems for Systems Engineering*.

At the Software Engineering Institute of Carnegie Mellon University, Dr. Sheard researches software engineering process and measurement and brings software engineering tools and technologies to government clients. Previously she was a consultant and teacher at Third Millennium Systems and at the Systems and Software Consortium, and a systems engineer at Loral/IBM Federal Systems and Hughes Aircraft Company.

Dr. Sheard has a Ph.D. in Enterprise Systems at the Stevens Institute of Technology, a master’s degree from the California Institute of Technology, and a bachelor’s degree from the University of Rochester.

Dr. Ali Mostashari is the Director of the Complex Adaptive Sociotechnological Systems (COMPASS) Research Center, and a Research Associate Professor of Systems Engineering at Stevens Institute of Technology's School of Systems and Enterprises. Also he is a senior organizational strategy consultant for the United Nations Development Programme.

Earlier, Dr. Mostashari served as a strategic advisor leading an oversight and performance management function and helping drive many strategic initiatives at the United Nations Development Programme (UNDP). He was nominated by the UNDP Assistant Secretary General for Africa to receive the World Economic Forum's Young Global Leaders 2008 award. In 2007 he was selected as a Asia 21 Young Leader. He was also a finalist in UNDP's Leadership Development Programme.

Dr. Mostashari authored a book on stakeholder engagement in sociotechnical systems design and has contributed multiple book chapters and other publications on complex systems, sustainable development, energy and environment.