



# SYSTEMS ENGINEERING NEWSJOURNAL

PPI SyEN 97 - January 2021

**Dedicated to inspiring and improving the practice of systems engineering**

brought to you by

**Project Performance International (PPI)**

*Systems engineering training and consulting for project success ...*

Welcome to the latest edition of PPI SyEN, PPI's monthly publication filled with informative reading for the technical project professional. In this issue, as well as in tons of archived issues available free at [www.ppi-int.com](http://www.ppi-int.com), you will find powerful ideas that may help you prosper in the highly competitive world of systems and product development.

Access a mix of news, quick tips, short reads and deep article content that will help you expand your professional skill set, keep up to date on events and activities of interest, and quite possibly unlock levels of project performance you've never before considered possible.

We hope that you find this newsletter to be informative and useful. As the leading provider of systems engineering training worldwide, PPI is passionate about improving project outcomes. Thousands of professionals in aerospace, medical, consumer and other major sectors subscribe to PPI SyEN, as do leading thinkers in academic, government and scientific organizations.

Our editors strive to bring you a diverse range of views and opinions each month, but please know that the views expressed in externally authored articles are those of the author(s), and are not necessarily those of PPI or its professional staff.

Have a topic you would like to learn more about, or possibly ideas you'd like to share with others? We'd love to hear from you! Just email us at [syen@ppi-int.info](mailto:syen@ppi-int.info)

We would also love it if you shared this edition with others who may benefit, and we encourage you to join our active community on [LinkedIn](https://www.linkedin.com). If a colleague sent you this copy, you can easily receive future newsletters directly by signing-up using the form at [www.ppi-int.com](http://www.ppi-int.com).

A PDF version of this newsletter can be found [here](#).

---

## IN THIS EDITION

---

### 1. Quotations to Open On

[Read More ...](#)

### 2. Feature Articles

- 2.1 Managing the “Black Hole” by *Gary A. Gack*
- 2.2 Why Systems Engineers should be Late Early Adopters of New Technology by *Barclay R. Brown*

[Read More ...](#)

### 3. Notable Articles, Webinars And Presentations

- 3.1 Some Tools for Use in Program Management and Systems Engineering Integration by *Robert Halligan*
- 3.2 SysML V2 Updates
- 3.3 INCOSE President Address: INCOSE International Workshop 2021 by *Kerry Lunney*
- 3.4 Systems Engineering as a Data-Driven and Evidence-Based Discipline by *Ron S. Kenett, Avigdor Zonnenshain, and Robert S. Swarz*
- 3.5 Analysis: A Love-Hate Relationship with “Systemness” by *Kevin Nortrup*
- 3.6 Ten Characteristics of an Outstanding Engineering Leader

[Read More ...](#)

### 4. Systems Engineering News

- 4.1 INCOSE Sector Updates—Asia-Oceania
- 4.2 Systems Engineering Body of Knowledge
- 4.3 India Systems Engineering Webinar Series
- 4.4 Named Professorship at Watson College (USA) Honors Systems Science Pioneer
- 4.5 SysML for Models of Digital Twins
- 4.6 INCOSE Annual Impact Report

[Read More ...](#)

## **5. Featured Organizations**

- 5.1 Center for Intelligent Systems (CIS)
- 5.2 Acquisition Innovation Research Center
- 5.3 Institute of Industrial and Systems Engineers (IISE)
- 5.4 Society for Health Systems (SHS)

[Read More ...](#)

## **6. News on Software Tools Supporting Systems Engineering**

- 6.1 Mentor Graphics Becomes a Part of Siemens EDA
- 6.2 Siemens RapidAuthor 13 Introduces New Functionality for Authoring and Illustration:

[Read More ...](#)

## **7. Systems Engineering Publications**

- 7.1 A Systematic Framework for Exploring World Views and its Generalization as a Multi-purpose Inquiry Framework
- 7.2 Systems Engineering Principles and Practice
- 7.3 Security Engineering: A Guide to Building Dependable Distributed Systems
- 7.4 Systems Engineering in the Fourth Industrial Revolution: Big Data, Novel Technologies, and Modern Systems Engineering
- 7.5 MITRE Systems Engineering Guide

[Read More ...](#)

## **8. Education and Academia**

- 8.1 Department of Systems Science and Industrial Engineering (SSIE)
- 8.2 Translating Systems Engineering for High School Teachers and Students: An Exploratory Study of Implementing some initial SE Concepts
- 8.3 Systems Modeling and Engineering Research Laboratory, Worcester Polytechnic Institute (USA)

[Read More ...](#)

## **9. Some Systems Engineering-Relevant Websites**

[Read More ...](#)

## **10. Standards and Guides**

10.1 IEEE 1028-2008 – IEEE Standard for Software Reviews and Audits

10.2 IEEE 42020-2019 – ISO/IEC/IEEE International Standard – Software, Systems and Enterprise — Architecture Processes

[Read More ...](#)

## **11. Some Definitions to Close On**

11.1 Systems Architecture

11.2 Enterprise Architecture

11.3 Hierarchy

[Read More ...](#)

## **12. Conferences and Meetings**

[Read More ...](#)

## **13. PPI and CTI News**

13.1 PPI Welcomes Eduardo to the Team

13.2 Successful First Live-Online CTI Course in China

13.3 PPI and INCOSE Systems Engineering Tools Database (SETDB) Activity at the INCOSE IW 2021

[Read More ...](#)

## **14. PPI and CTI Events**

[Read More ...](#)

## **15. Upcoming PPI Participation in Professional Conferences**

[Read More ...](#)

---

# 1. QUOTATIONS TO OPEN ON

---

*A systems engineering process standard is not a substitute for engineering competence, common sense, and hard work.*

**Robert John Halligan**



*Don't ask what the world needs. Ask what makes you come alive and go do it. Because what the world needs is people who have come alive.*

**Howard Thurman**



*Nothing ever becomes real until it has been experienced.*

**John Keats**



*Planning is bringing the future into the present so that you can do something about it now.*

**Alan Lakein**

---

## 2. FEATURE ARTICLES

---

### 2.1 Managing the “Black Hole”

by

Gary A. Gack | [g.gack.fl@gmail.com](mailto:g.gack.fl@gmail.com)

Version 1.0 December 13, 2020

#### Abstract

This article, based largely on my book, *Managing the Black Hole: The Executive’s Guide to Software Project Risk*, provides a concise summary of several “redeeming virtues” that can offset the “grim realities” to which software projects are susceptible. Some may find elements of my commentary controversial. I will be happy to provide additional detail and supporting rationale on a particular topic on request.

Copyright © 2020 by Gary A. Gack. All rights reserved.

#### Introduction

Software projects are risky and failures are common. Less than one-third of all software projects are fully successful. Success means delivered on-time, on-budget, with the needed features and functions. The average software project overruns its budget by around 50% and schedule by around 80%. The average project delivers less than 70% of planned features and functions. *These statistics have not significantly changed over at least the last 20 years!*

In particular, larger projects have high cancelation risk:

- Approximately 20% of projects costing \$1,000,000 to \$25,000,000 fail
- Approximately 40% of projects costing \$25,000,000 to \$200,000,000 fail
- Approximately 60% of projects costing more than \$500,000,000 fail (these few systems are mainly in defense, aeronautics, operating systems, and the ERP domain – these are built by very few organizations)

Approximate Cost by Size Range			
Size	Function Points	Typical Cost Range (US\$)	
Very large	100,000 +	\$200,000,000	\$1,200,000,000
Large	10,000 – 100,000	\$25,000,000	\$200,000,000
Medium	1,000 – 10,000	\$1,000,000	\$25,000,000
Small	< 1,000	\$100,000	\$800,000

Adapted from Capers Jones © 2007

Projects in this high risk group account for around 80 – 90% of total software spending, even though they constitute only around 8-10% of all software projects.

## Why Do Software Projects Fail?

### Lack of Risk Recognition

Executives need a realistic perspective on risk. Projects in the Medium (\$1,000,000 – \$25,000,000) and Large (\$25,000,000 – \$200,000,000+) size categories carry high risk – 20-40% of them fail. If and/or when such projects are actually delivered they are likely to be highly defective and only partially complete. Business as usual is insufficient for high risk projects, so early awareness of risk is essential. Forewarned will hopefully lead to forearmed.

### Sizing and Estimating Malpractice

Estimating is among the most severe weaknesses in the software field. Excessive optimism and a lack of understanding of software “laws of physics” often leads to “3 minute mile” estimates. The world record is around 3:40. Among many organizations the cost/schedule tradeoff is rarely a conscious decision based on facts and data. Like it or not, shorter schedules *always* mean higher risk and much higher cost – it’s a highly non-linear relationship *that can be quantified*.

#### **Speed Kills**

The impact of schedule pressure has been quantitatively modeled (based on thousands of completed projects) by several software cost estimating tools, including QSM ([www.qsm.com](http://www.qsm.com)) and Galorath ([www.galorath.com](http://www.galorath.com)). All of these models clearly show non-linear (exponential) relationships among project effort, delivered quality and the schedule itself.

The impact of the schedule is particularly dramatic for larger projects (generally those greater than 10 person years of effort). For larger projects it is clear that compression of the schedule can increase cost by a factor of 2 to 3, increase delivered defects by a factor of 5 or more, and greatly increase risk of failure. In larger projects you can realistically expect to get 3 for the price of 2 if you opt for least cost schedules – in a typical case that may mean 12 months instead of 9 (which probably won’t happen anyway).

In other words, a project that could be delivered in 12 months at a cost of around \$1,000,000 may cost \$2,000,000 or more to deliver in 9 months, and the quality will be dramatically lower. This is not speculation, but as near as we get to universal truth in software.

### Project Planning Deficiencies

Planning is a serious weakness in many software organizations. Plans lack adequate detail, end states are not clearly defined, and defect detection and rework activities are not explicitly planned. These failings lead to highly misleading reports of progress as supposedly completed work products contain significant numbers of defects that must be corrected later.

Many of these organizations utilize excessively complex systems for planning and reporting that often lead to very poor data quality – in reality “less is more”. Time reporting and project status tracking are both more effective and more accurate when these needs are met by different processes, rather than by a single integrated process. (*Heresy alert* – the first of several).

## Methods, Standards, and Tools Fetishes

Like the drunk looking for his keys under the street light “because that’s where the light is”, many immature software organizations are looking for salvation in the wrong place. New methods have not driven significant improvements in outcomes on high risk projects. *Actual performance* is what matters, not *compliance* with a particular approach, model, or standard. Many organizations can realize major gains using virtually any of the available methods and standards – as organizations approach Philip Crosby’s “level 4” Quality Management Maturity, choices of methods and standards become more important, but initially they are far down the priority list.

As with methods, tools are not going to turn lead into gold. They can be helpful in a stable process, but do not in themselves bring stability. Investment required to introduce and support tools can easily consume millions of dollars. The learning curve on the bleeding edge often carries a significant penalty and can increase risk. That does not mean methods, standards, and tools have no value, but it does mean other actions are much more essential. Reducing excessive task switching, for example, far outweighs the impact of any particular method. Estimating and planning, quality management, and project status tracking processes are all much more important than the methods, tools, and standards chosen.

## Product Quality Not Top Priority

Finding and fixing defects is by far the largest part of total software cost. Most immature organizations spend far more, and achieve far less, than is readily possible. Compared to an average immature organization that finds 85% of defects before delivery, a “best in class” organization finds 95% – and does so 20-30% cheaper and 10-15% faster. Finding defects before testing is key to significant improvement. An effective defect containment regimen will significantly improve delivered quality and *at the same time* reduce non-value-added costs. The methods proposed here are implementable by any organization.

## Measurement Myopia

The “big 4” measures in software groups include effort (time), project duration, defects, and size. Of these many groups measure effort and duration, but rarely do so accurately. Many collect more detail than is actually useful, and the additional detail contributes to inaccuracy that leads to misleading status reporting. Few immature organizations measure size at all – a major contributor to estimating errors.

Software Measurement Maturity	Crosby's "Quality Management Maturity" Stages, adapted to software projects				
	1:Uncertainty	2:Awakening	3:Enlightenment	4:Wisdom	5:Certainty
<b>Management Understanding</b>	No comprehension of quality as a management tool	Quality might be valuable, but not funded	Learning, becoming supportive and helpful	Participating, recognizing personal role	Quality is an essential part of effective management
<b>Problem Handling</b>	Fire fighting, lots of yelling and blaming	Teams address critical issues, quick fix bandaids, short term focus, problems recur	Corrective action focus, problems are faced openly and resolved in an orderly process	Problems are identified early, all participants are open to improvement	Most problems are prevented
<b>Non-Value-Added % of effort</b>	60-70%		50%	40%	20-30%
<b>Pre-Release Defect Containment %</b>	60-80%		85%	90%	95%+
<b>% of Software Organizations (estimate)</b>	65%		20%	10%	5%

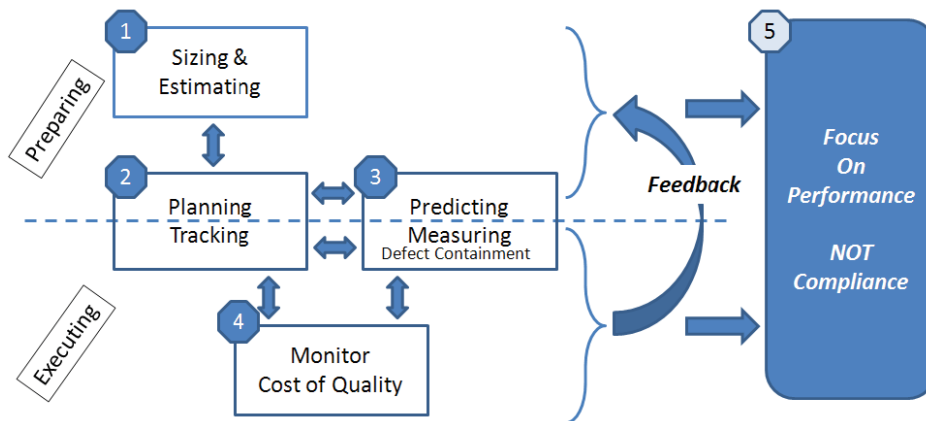


Generally inadequate measurement of defects among many groups is the equivalent of night driving without headlights. Finding and fixing defects are the largest single cost elements in nearly all software projects – all of that effort is “non-value-added”. Many organizations rely on a “test only” strategy that increases project risk, leads to lower delivered quality, and is an important factor in project delays. Most of these organizations devote little or no effort to finding defects in requirements and design before construction begins, resulting in around 40% of all delivered defects traceable to requirements and design errors and omissions. Here follows a more effective approach that will lead to significantly reduced costs.

## What Can You Do to Improve Outcomes?

Given the limited length of this paper I have elected to briefly summarize key “virtues” and introduce a software “econometric” model that will enable quantification of the expected impact of quality related process changes.

Here’s how the “virtues” fit together:



### 1. Rationalize Sizing and Estimating

“Triage” always comes first – you are off on a trip to Las Vegas – think through your risk tolerance – find out how much is likely to be at stake. Capers Jones has defined a multi-dimensional analogy approach to first approximation estimates (called Software Risk Master)<sup>1</sup>.

First approximations are just that – expect the initial estimate to change significantly. Make sure those doing sizing are up on the latest approaches – sizing is critically important, but don’t spend more than necessary.

Realistic “bottom up” plans cannot be developed until requirements are reasonably well understood – don’t stint on involving key staff if you want to get it right – pay now, or pay a lot more later. Don’t commit beyond requirements until you have a solid bottom up plan.

Get independent “should cost” estimates developed using “top down” methods such as those available from Galorath<sup>2</sup> or QSM<sup>3</sup>. Make sure assumptions are explored and bottom up estimates converge with top down. Don’t commit to the next phase until independent estimates have been sanity checked and reconciled to bottom up estimates prepared by the development team.

<sup>1</sup> <http://www.namcook.com/>

<sup>2</sup> <https://www.linkedin.com/company/galorath-incorporated/>

<sup>3</sup> <https://www.qsm.com/>

Make certain estimates are developed using adequate expertise and tools appropriate to the task. This is not the place to rely on amateurs.

Hold development teams accountable for quality as a primary consideration. Don't focus on schedule. Ultimately you'll get it sooner and at lower cost if teams are evaluated on quality first.

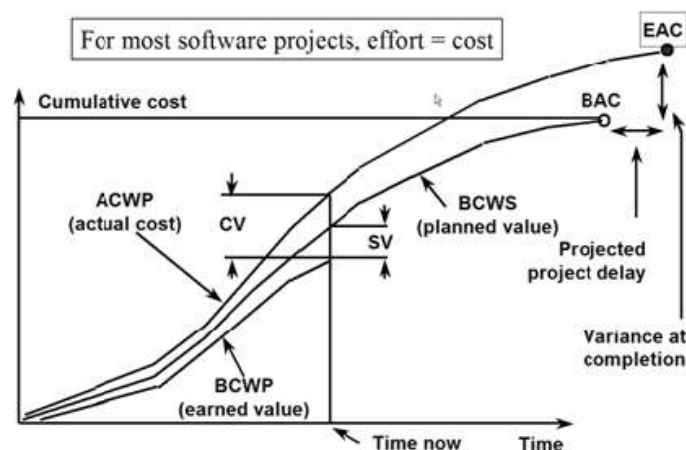
## 2. Professionalize Planning and Tracking

Expect to spend about 5% of total project effort on project planning and tracking – an “out of control” project is one that does not have time to plan.

Always require plans developed using the Critical Path Method. Require a detailed Work Breakdown Structure that identifies “small” tasks – generally most should be no more than one week in duration. Require *explicit* planning for Appraisal and Rework tasks that are in aggregate reasonable in relation to industry benchmarks or locally documented past experience.

Mature organizations *may* be able to implement best in class status tracking using the Earned Value method<sup>4</sup> (“EVMS”, an ANSI standard) – highly recommended, but feasible only if data collection is highly accurate, often not the case.

### Earned Value



You will notice a set of abbreviations on the chart – here are the definitions along with the associated values based on our simple example above:

- BAC: Budget at completion = 100 hours
- EAC: Estimate at completion =  $BAC / CPI = 100 / .57 = 175$  (75% overrun!)
- SV: Schedule variance =  $BCWP - BCWS = 20 - 30 = -10$
- CV: Cost variance =  $BCWP - ACWP$  (positive is favorable) =  $20 - 35 = -15$
- $SV\% = SV / BCWS = -10 / 30 = -33\%$
- $CV\% = CV / BCWP = -15 / 20 = -75\%$

CPI: Cost Performance Index =  $BCWP / ACWP = 20/35 = .57$

SPI: Schedule Performance Index =  $BCWP / BCWS = 20/30 = .67$

Earned Value shows us how our actual rate of development (“productivity”) compares to the rate that was assumed when we prepared our estimates. In this illustration it’s a pretty dismal picture – we’re 75% over budget and 33% behind schedule as of “time now”.

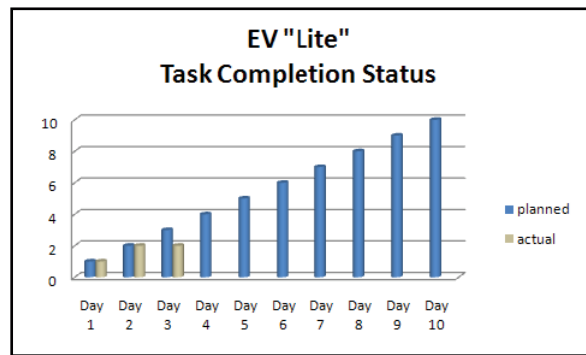
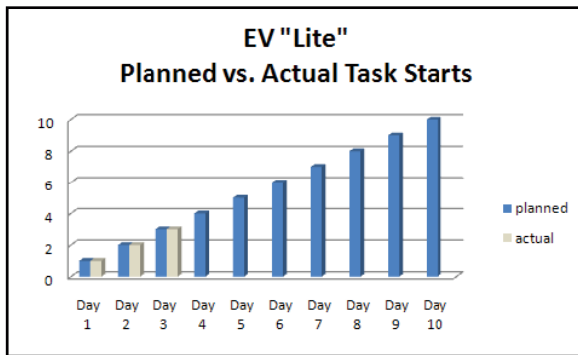
<sup>4</sup> See Paul J. Solomon and Ralph R. Young, [Performance-Based Earned Value](#), for an authoritative treatment of PBMS.

A less challenging approach I call “Earned Value Lite” can alternately be used to track project status. This approach is much simpler and yet delivers most of the benefit of full EVMS.

Earned Value “lite” requires a reasonably detailed project plan that includes expected start and finish dates for each task, but does not specifically require either critical path or full-blown EVMS. Many lower maturity organizations can do this. It does not require task level time tracking, but it does require objective monitoring of tasks planned to start and end vs. tasks actually started and ended.

The chart on the left shows planned (darker bar) and actual (lighter bar) *starts*, while the one on the right is planned and actual *finishes* against a timeline, updated at “time now” (day 3 in this example) to reflect actual status of tasks. Data used here is the same as the EVMS example above.

Together these charts give a very clear picture of real status – when the light and dark bars diverge they clearly indicate whether the project is ahead or behind.



EV “Lite” makes several important simplifying assumptions:

- ACWP = total effort charged to the project as of time now. In our example, as of day 3, 35 hours have been charged to the project in total.
- We do not attempt to track actual time at the task level. i.e., BCWS = the sum of the estimated effort *planned* for those tasks *planned* to be completed as of time now.
- If a task is late the actual effort expended is *assumed* to be proportionate to the ratio of tasks that were scheduled to be completed / the number actually completed. i.e., BCWP = BCWS \* (# tasks *actually* completed / # tasks *scheduled* to complete)

Example: BCWP = 30 \* (2/3) = 20 – the same result we get from the more formal method *provided* the simplifying assumptions are in fact valid. Most of the time they are very close.

Establish an independent “Project Office” function to monitor and objectively report status. Never allow the fox in with the chickens – always ensure the Project Office is independent of the development team.

Set high standards for outsource providers. Require detailed information, metrics, and honesty in reporting. Specify penalties for misleading or incomplete information.

### 3. Monitor Cost of Quality

Many organizations collect task level time accounting information that is highly inaccurate and rarely used. A simplified data collection chart of accounts will prove more useful and more accurate.

A simple “Cost of Quality” chart of accounts will collect time in broader categories: Value-Added, Appraisal, Rework, and perhaps Prevention. Monitoring the value-added percentage gives a solid indication of overall improvement.

The “Chief Measurement Officer” should facilitate definition of an appropriate chart of accounts and report objectively on the data, but responsibility for data accuracy and completeness must rest with those reporting.

The politics of implementation are complex and likely to meet resistance. Clear executive level sponsorship is essential, as is a carefully considered implementation plan.

#### **Less is More**

From an efficiency improvement perspective task level time detail is not useful, because that data is not comparable across projects. What is common to all projects, and far more useful for measuring efficiency, is a “Cost of Quality (CoQ)” view of time expenditures. For software and IT activities we commonly use a simplified three part CoQ scheme consisting of Appraisal (testing, inspections, any activity performed to find defects), Rework (all effort expended to correct defects detected by Appraisals and by customers), and Value Added, which is simply total time less Appraisal and Rework – collectively referred to as “internal” CoQ.

Significant additional costs, known as “external” CoQ, include all costs incurred by the development and/or support organizations associated with diagnosis and correction of defects released to customers. External CoQ should also include the costs experienced by the customer as a consequence of defects, but in practice these are very difficult to measure and are often ignored. Some estimates suggest these costs are roughly equal to internal CoQ.

In many instances this approach will mean *less* time reporting, not more – fewer items in the “Chart of Accounts” (list of time-charge categories). Alternately, if you are committed to more detailed task level time accounting you can structure your chart of accounts so that the detail is explicitly mapped to CoQ categories.

#### 4. Predict and Measure Defect Containment

To increase value-added, the single most important thing any software organization can do is to apply an appropriate amount of resource, using appropriate Appraisal methods, at every stage of the development process.

To effectively manage defect containment, the organization must predict the number of defects likely to be “inserted” at each stage of the development process. Most will rely initially on industry benchmarks and revise as experience is gained.

A count of defects discovered can be compared to the estimate of defects present to approximate a “containment rate” at each stage of development. This rate is a leading indicator that provides useful insight into product quality as the project progresses. This metric, in effect, provides a “quality adjusted” view of project status.

Available Appraisal methods include not only testing, widely used in most organizations, but also a variety of pre-test methods, including formal software inspections (defined by an IEEE Standard) and, where applicable, static analysis. Pre-test methods are dramatically more effective and efficient than the “test only” approach commonly used. In addition a relatively new state of the art test planning approach based on Design of Experiments should be used to maximize test coverage while minimizing the number of test cases.

Collection and analysis of defect data provides critical feedback essential to improvement. The details of how to do that and ensure data quality require you assign a “defect czar” to monitor, maintain, analyze, and report. Require reporting of defect related efficiency and effectiveness metrics, including Total Containment Effectiveness (TCE) and Appraisal Containment Effectiveness (ACE) for each Appraisal method.

Require use of the Defect Containment Econometric Models proposed here, or equivalent. These models enable forecasting of the consequences of different defect insertion and detection rates, rework costs, and alternative Appraisal strategies. In aggregate these models forecast and monitor the impact of defect containment strategies as they drive non-value-added cost. *A copy of this model is available from the author on request (including guidelines and suggested initial parameters for use).*

Let’s take a look at what we can learn from using these models to play out a series of scenarios that embody different appraisal strategies.

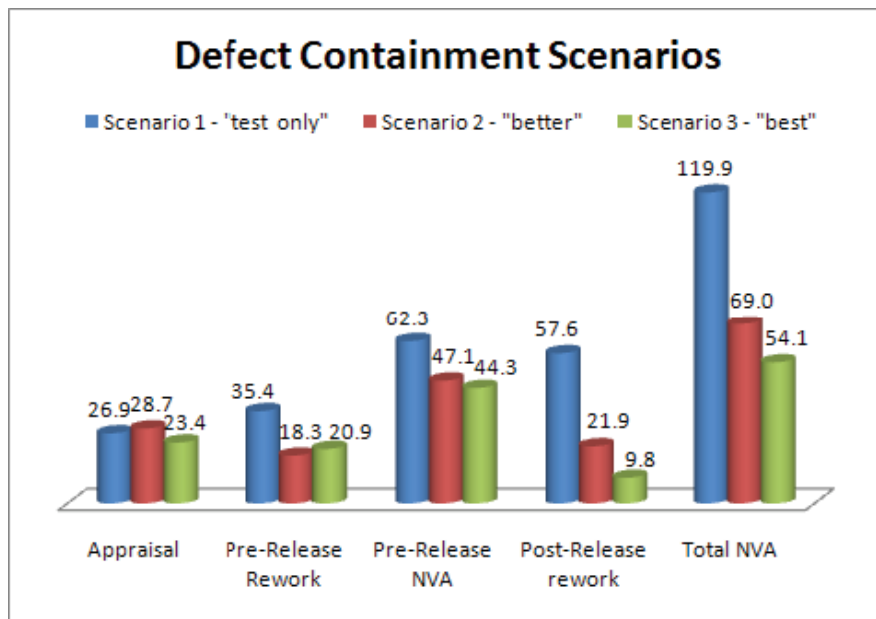
*Scenario 1 – common practice today – “Test only” – five types of tests are used*

*Scenario 2 – “better” – 50% of Code is inspected and the 5 test types are continued*

*Scenario 3 – “best” – Requirements, Design, and 20% code inspections are used, Static Analysis is added, and the first 2 test types are discontinued*

I have defined a “reasonable” set of parameters common to all three scenario models – common containment rates, unit costs for rework, etc. What varies across the different scenarios is the mix of appraisal types included – *nothing else changes.*

Here are the results in person months for a hypothetical 1,000 function point project:



The “best” scenario reduces total NVA by more than 50% compared to “test only” and delivered quality (TCE) improves from 80.5% to over 95%!

*Even more importantly, post-release rework is reduced by over 80%! This saves costs in the software team, but also greatly improves the customer experience – fewer interruptions for software defects, more time achieving business goals!*

This is NOT speculation – THIS CAN BE DONE – better, faster, AND cheaper.

#### 5. Focus on Performance

A focus on performance is a longer term culture change for many organizations. Quantitative literacy takes time and experience to mature into a way of life.

Implementing the recommendations offered here will establish a solid foundation for additional substantial improvement. Applying state of the art methods such as Lean and Six Sigma will lead to additional continuous improvement.

#### **Summary and Conclusions**

I know from personal experience that the ideas expressed here are implementable in any software organization (given meaningful executive support), and will be particularly beneficial to those early in the process improvement journey. I also believe some of these ideas will prove valuable even among high maturity organizations. It would be most interesting to know the extent to which high maturity groups have local experience data necessary to accurately populate the Cost of Quality model described. Perhaps some of our readers will be interested in taking a look at the model and giving me some feedback. I look forward to your comments.

## List of Terms/Acronyms Used in this Paper

Term	Description
<b>Appraisal</b>	Any activity whose principal purpose is to find defects. Includes a variety of testing and other pre-test methods such as formal inspections and static analysis.
<b>ACE - Appraisal Containment Effectiveness</b>	The percentage of defects present that are removed by a specific Appraisal type.
<b>Benchmark</b>	A point of reference such as averages, means, and distributions derived from industry studies of actual results.
<b>Process Maturity</b>	A rating, generally on a 1 to 5 scale, that characterizes the extent to which an organization uses defined and repeatable processes appropriate to the domain of activity. Established scales include those defined by Philip Crosby, by the Software Engineering Institute, and by ISO Standard 15504.
<b>TCE – Total Containment Effectiveness</b>	The percentage of software defects removed prior to release of software for customer use. $(\text{defects found pre-release}) / (\text{defects found pre-release} + \text{defects found post-release})$
<b>Defect</b>	Any exception or deficiency in software sufficiently important to justify correction. Defects are often classified as “Major” (Severity 1 or 2) or “Minor” (Severity 3 or 4)
<b>Defect Severity</b>	1 – Software does not run 2 – Major function disabled 3 – Minor function disabled 4 – Cosmetic error
<b>Defect Potential</b>	An estimate of the number of defects likely to have been “inserted” (present) at a given point in the development process.
<b>Value-Added Effort</b>	The percentage of total effort NOT devoted to Prevention, Appraisal, or Rework.
<b>Rework</b>	Any effort devoted to correction of defects both pre- and post-release.
<b>Embedded Software</b>	Software included within (as part of) another product not sold as software per se.
<b>Package Software</b>	General purpose software not developed solely for a specific customer.
<b>TQM</b>	Total Quality Management
<b>Six Sigma</b>	A process improvement methodology focused on reduction of defects and variance. Developed at Motorola during the 1980s.
<b>Lean</b>	A set of tools and concepts focused on reduction of waste and cycle time. Derived from the Toyota Production System.
<b>Lean Six Sigma</b>	In practice ideas from both have largely merged into a unified approach to process improvement, cycle time reduction, and elimination of waste.
<b>SEI</b>	The Software Engineering Institute. Established at Carnegie Mellon University. Initially funded by the US Dept. of Defense to focus on improvement performance of government software contractors. Authors of the <i>Capability Maturity Model Integration</i> and several variants thereof.
<b>CMMI®</b>	Capability Maturity Model Integration
<b>Prevention</b>	Any activity intended to reduce defect insertion and related rework. Typically includes training and process improvement initiatives.
<b>Cost of Quality</b>	A system of measurement that focuses on distinguishing non-value-added activities (primarily Appraisal and Rework) from value-added activities.



Term	Description
<b>Agile Methods</b>	A family of approaches to software development that attempt to embody Lean principles and practices.
<b>Function Points</b>	A software sizing method defined by the International Function Point User Group (IFPUG)
<b>Critical Path Method</b>	An “industrial strength” approach to project planning.
<b>Earned Value</b>	An “industrial strength” approach to project status reporting. Defined by an ANSI standard.

## References

Barry Boehm, *Software Engineering Economics*, Prentice Hall (1981), ISBN 0-13-822122-7 (and Richard Turner) *Balancing Agility and Discipline: A Guide for the Perplexed*, Addison Wesley (2004) ISBN 0-321-18612-5

Capers Jones:

*Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies*, McGraw-Hill (2009), ISBN-13: 978-0071621618.

*Applied Software Measurement 3rd Ed.*, McGraw-Hill (2008) ISBN 978-0-07-150244-3.

*Estimating Software Costs 2nd Ed.*, McGraw-Hill (2007) ISBN: 978-0-07-148300-1.

*Assessment and Control of Software Risks*, Prentice Hall (1994) ISBN 0-13-741406-4.

Frederick Brooks, *The Mythical Man-Month: Essays on Software Engineering 2nd Ed.*, Addison Wesley (1995) ISBN: 978-0201835953.

Gary Gack, *Managing the Black Hole: The Executive’s Guide to Software Project Risk*. Business Expert Publishing (2010) ISBN: 1-935602-01-2.

Karl Wiegers, *Peer Reviews in Software*, Addison Wesley (2002) ISBN 0-201-73485-0.

Larry Putnam and Ware Myers, *Five Core Metrics*, Dorset House (2003), ISBN 0-932633-55-2.

Paul J. Solomon and Ralph R. Young, *Performance-Based Earned Value*. The IEEE Computer Society and John Wiley & Sons Publications, 2007. ISBN 978-047172188-8.

Peter Senge, *The Fifth Discipline: The Art and Practice of the Learning Organization*.

Ron Radice, *High Quality Low Cost Software Inspections*, Paradoxicon (2002) ISBN 0-9645913-1-6.

Stephen Kan, *Metrics and Models in Software Quality Engineering, 2nd Ed.*, Addison Wesley (2003) ISBN 0-201-72915-6.

Tarek Abdel-Hamid and Stuart Madnick, *Software Project Dynamics* Prentice Hall (1991) ISBN 0-13-822040-9.

W. Edwards Deming, *Out of the Crisis*, MIT CAES (1982), ISBN 0-911379-01-0.

Watts Humphrey:

*Managing the Software Process*, Addison Wesley (1989) ISBN 0-201-18095-2.

*A Discipline for Software Engineering*, Addison Wesley (1995) ISBN 0-201-54610-8.

*Winning with Software: An Executive Strategy*, Addison Wesley (2002) ISBN 0-201-77639-1.



## About the Author



### **Gary A. Gack**

#### Education and Credentials

- MBA, The Wharton School, 1986; Six Sigma Black Belt, ASQ Certified Software Quality Engineer, PMI Project Management Professional, Certified Scrum Master, ITIL Foundation Certification

#### Work History (50+ years in software/IT)

2014 – Present: semi-retired, occasional informal consultant

2007 – 2014: self-employed independent consultant – Process-Fusion.net

Client engagements included:

- MBUSA Quality Evaluation Center – Six Sigma Green Belt training for 20 engineers – coached candidates for certification through team process improvement projects dealing with product quality operations.
- Broadridge Corp. – Six Sigma and software process best practices training for 16 software/IT professionals
- Ulticom – Six Sigma and software best practices, and formal inspections training and coaching for a team of approximately 20 software engineers and managers building telecom / VOIP software
- Gap, Inc. – problem project turnaround coaching and facilitation in support of the Real Estate department – performed an assessment to determine root causes of two prior failures, assisted them in reforming and re-motivating a joint business / IT team, aided them in vendor selection and team building. Facilitated requirements gathering and high level test planning.
- SAP Corporate (Waldorf, Germany) – lead and facilitated a process improvement assessment and planning activity in conjunction with the internal Six Sigma quality team

2003-2007: Partner, Six Sigma Advantage

Client engagements included:

- SAP (Waldorf, Germany) – Six Sigma Yellow Belt, Green Belt and Black Belt training for approximately 100 candidates in the corporate software quality organization. Provided training on software inspections and other software best practices. Coached Green and Black Belt candidates in execution of several dozen improvement projects required for certification.
- VF Corporation – performed an assessment of root causes of a “runaway” SAP deployment effort; facilitated re-planning for the project, resulting in a 10,000 task critical path network.
- Provided Six Sigma and software best practices training to software specialists in Motorola, IDX Software, SaraLee, Seagate, Trane, National City Bank, and others

1993 – 2002: self-employed independent consultant – IT Effectiveness, Inc.

Client engagements included:

- A variety of training, coaching and consulting engagements related to software and IT best practices in requirements, project management, methodologies, quality assurance, process improvement, and software metrics. Clients included VF Corporation, Rohm and Haas, Okidata, Bell Canada, Citizens Telecom, Mastercard, Cap Gemini, and others

## 2.2 Why Systems Engineers should be Late Early Adopters of New Technology

by

Barclay R. Brown, Ph.D., ESEP | [Barclay.brown@incose.net](mailto:Barclay.brown@incose.net)

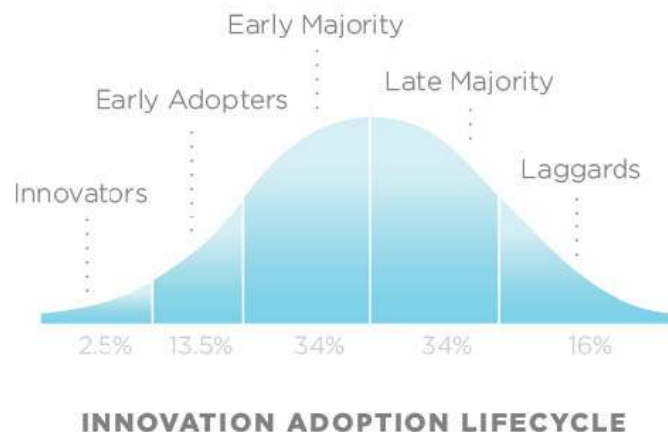
Engineering Fellow, Raytheon Technologies

INCOSE CIO-Elect, 2021-2023

<https://www.linkedin.com/in/barclayb/>

Copyright © 2020 by Barclay R. Brown. All rights reserved.

Systems engineers deal with two kinds of technology—the technologies they choose to build into the systems they create, and the technology they use in their own systems engineering work. Since the days of pencils and typewriters, systems engineers have used various forms of technology to reason about, document, and communicate their systems engineering work. In this article I take up the question of where systems engineers should be on the technology adoption curve when it comes to their own use of technology.



I'll make the case that systems engineers should be among the mid-to-late early adopters, eagerly taking hold of new technology, but only after it's been proven to work reliably, and long before the early and late majority groups have picked it up.

My generation saw the first mass-produced microprocessor introduced as we began college, and the personal computer and I began our careers at about the same time. With about two billion personal computers in use today, my career-cousin has been quite successful. No matter your age, you probably have an idea where you were on the personal-computer adoption lifecycle. Were you the first on your street to get one, or the last? Did you hold onto Windows 3.1 (or XP or 7) as long as possible, going to Windows 10 only when you were forced? Or did you get a Macintosh so you could run that first version of PageMaker and do your own layout and avoid expensive manual stripping and film creation? When did you move beyond dial-up Internet access in your home? As soon as it was available, or only after friends and family laughed at you for still dialing into America Online?

As systems engineers, we tend to think in a linear systems development lifecycle, where the first step is gathering requirements, followed by architecture and design, and only later, implementation technology selection. Gathering requirements has two major components to it—first, finding out our users' and stakeholders' wants, needs, and expectations, and second, writing down what users and stakeholders want the system to do. It is only a naive systems engineer who assumes these two are the same thing. What human beings want, need, or expect is not a fixed set of information, like a specific vitamin deficiency, where the solution follows directly from the need. Our wants, needs, and expectations are fluid, changing as we change and influenced heavily by what we view as possible or available. No one wanted or needed a smart phone until it was invented by a visionary and presented as available. Then the adoption curve began. Smart phones achieved one of the fastest adoption rates in technology history, but there were still clearly identifiable innovators, early adopters, early majority, and late majority, and there are still some who don't have them and don't (yet) want them—the laggards.

Figuring out what system is needed, and how it should evolve and change over time, is what we do as systems engineers. We must balance many factors including the risk of adopting a technology too early with the cost of being too late. We balance the disruption of changing to a newer technology with the missed benefits of staying with the current system. As painful as it can be, switching technologies is often necessary. A company that has built a successful business with a product created years ago, often cannot afford to rebuild it from the ground up with newer technologies and approaches. To do so could risk alienating their potential late adopter customers, so the company becomes vulnerable to a startup or spin-off which can disrupt by creating a new product (based on concepts proven at the original company's expense, but using newer technology).

## **The Technology Pendulum**

Successful systems are conceived in a pendulum-like pattern. Systems engineers, users, and stakeholders swing back and forth between the consideration of wants, needs, and expectations on the left, and what technology and solutions are available on the right. Back and forth they go, until the pendulum comes to rest somewhere in the middle, with a choice optimized to the situation. The best choice is not likely to be the very latest technology since it may not have proven itself sufficiently. But the right choice is unlikely to be the oldest and most-proven technology either, since it may not be the most efficient or long-lasting solution.

Engineers, whose career it is to design and build new things, can fall into the trap of assuming that the right solution is to build something new. After all, that's a proven approach. To take an extreme example, an author planning to write a book could decide that the safest approach is to use a completely proven technology, the C++ programming language, to write a word processor. After all, Microsoft Word could lack features, or have bugs that the engineer can't fix. But the years it would cost for the engineer to create a new word processor aren't likely to be worth it. Probably, the would-be author will abandon the book project before the word processor is completed. To extend the example, an author who considers all options, might opt for a newer technology than MS Word, something specifically designed for authoring books, like Scrivener<sup>5</sup>, concluding that the benefits outweigh the risks of the newer and less-proven technology. An author unaware of the existence of Scrivener or overly skeptical of new technology, might not even consider it, but it's probably the most effective and efficient way to get the new book written.

There are several key reasons why systems engineers should make the effort to remain informed on innovative new technology, especially information technology, and try to be in the mid-to-late early adopter group when it comes to putting new technology to work:

---

<sup>5</sup> <https://www.literatureandlatte.com/scrivener/overview>

1. Systems engineers are called upon to have a broad view of new systems, to make strategic choices about functionality, architecture, and even implementation. As the pendulum illustrates, knowing what technology is available (or will be available soon) can dramatically influence how a system is conceived and architected. When Netflix began streaming video as an addition to its DVD rental business, only some homes had high-speed internet sufficient to effectively stream content. Most relied on cable or satellite for high quality video delivery. But Netflix was prescient, knowing that faster and more widely distributed high speed Internet service was coming, able to deliver 100 Mbps+ speeds, and built a streaming system that now streams even 4k content successfully to many homes—something hard to imagine in the days of 5 Mbps service. To build a system based only on today's technology is to build a system that will be outdated before its time.
2. Systems engineers are optimizers, working to optimize both the systems they create, and also the systems used in their own work. It seems natural for a systems engineer to be continually evaluating the tools, methods, and technologies used to perform systems engineering and select those that have proven effective enough to adopt, at both organizational and personal levels. Years ago, I discovered that I'm more productive with several large monitors connected to my computer, a mechanical (gaming) keyboard, and a precision laser mouse. For a few hundred dollars, I have multiplied my productivity and enjoyment of working! (My latest personal work technology find: DisplayLink technology which allows the connection of up to six monitors to any computer using a USB 3 port.)
3. Systems engineers should be lifelong and broad-based learners. While single-discipline engineers go deeper over time, systems engineers need to go broad, but deep enough to understand technology and make good choices about its use. There are still systems engineers who refuse to learn SysML, even though it has been successfully applied to systems engineering for over ten years. Their resistance robs them of the ability to make a reasoned decision about its applicability to a systems development effort. Learning a new technology doesn't mean you have to use it, but refusing to learn it removes the option.

My early-career computer programming was in languages that haven't been used for years so when I wanted to pick up programming again, I selected Python, an easy to learn but very powerful language widely used in data science and artificial intelligence development. I wanted to learn how to develop AI algorithms, not because I wanted to develop them as a career, but because I wanted that deep intuitive sense for how they work. Now that I've built a few AI applications, I understand enough to operate as a systems engineer for systems that include AI subsystems.

### **Fear Not the Tech**

As we age and mature, we sometimes get out of practice learning new things, falling into habits and patterns of working and thinking we developed long ago. We may even defend them, without evidence, as being better than new ways of working. But it's never been a better time to be a lifelong learner. Virtually anything you want to learn is available online, usually in well designed and beautifully presented courses, for free or nearly free. Even better, a learner can select a learning style that matches his or her background, prior knowledge, and learning style preferences. I like audio books and courses with video lectures—others may prefer reading printed (or on screen) books. When I decided to learn Python I was a bit scared—I had not written a line of code in decades. I found a very slow-paced, methodical course on the most elemental basics of Python programming, which I could follow easily, building my confidence until I could handle more advanced courses and reference books.

When faced with a new technology, I like to find beginner tutorials that take me from the very beginning. In INCOSE, I am championing a number of new technologies, and since many of them are in use by millions worldwide already, there are lots of tutorials and guides readily available (pro tip: use words like *tutorial* and *beginner* when searching YouTube for help on getting started with a new technology.) Cultivate the Zen beginner-mind, by being open to learning. Be the ready student and the master will appear. You might be surprised how many highly technical people love beginner tutorials on new subjects. Have you noticed how popular the “Dummies” book series is? People crave good, simple explanations that can take you from where you are to where you want to go. Apply game theory and balance the *exploration* of new technology through learning and research, with the *exploitation* of the technology in practice. I tend to watch beginner tutorials and lessons until I’m itchy to try it out myself, then I start playing with the new technology or product. When I exhaust my knowledge, I go back to school to learn more, until I’m itchy to try out that new information.

Fear is the mind killer, cautions a character in Frank Herbert’s *Dune*<sup>6</sup>. Replace fear with a systematic approach to learning, taking the time to be a student for a while. Soon the knowledge will be yours, and it won’t matter how long you took to learn it. Don’t hide behind excuses like “I’ve never been good with computers,” or “I like doing it the old way,” especially if you’ve never tried learning the new way! For some great thinking and motivation to explore new technology, try Kevin Kelly’s excellent book, “The Inevitable: Understanding the 12 Technological Forces That Will Shape Our Future.”

As the newly elected CIO of INCOSE for the 2021-2023 term, my plan is to bring new technology to the organization, and make it available to those ready to learn and enjoy its benefits. Our safe and small scale environment, compared to the large enterprises that employ many of our members, enables us to move faster, experiment with new technology, and give our members a jump on learning it that will help in their careers.

### **Vision for INCOSE Information Technology**

The world of IT moves and changes rapidly, and INCOSE must also move and change to be the most effective. INCOSE IT should lead in this area, and challenge the organization to increasing levels of automation, efficiency, and communication. Every function within INCOSE can be improved by the application of current IT technologies. In 2020, as I became the INCOSE Assistant Director for Online Collaboration, I immediately pushed for the acquisition of Zoom as the primary technology for live online communication. The push came at a time when headlines were accusing Zoom of security issues, problematic information routing and storage locations, and many other sensationalistic issues. We persevered, evaluating such claims on their own detailed technical merits. As Zoom is increasingly adopted worldwide, and was used successfully to deliver the INCOSE International Symposium online in 2020, this choice has been confirmed. INCOSE stands to save thousands of dollars each month in communication costs, while achieving high quality worldwide audio and video communication through the use of Zoom.

At the same time, we began a push for the implementation of Microsoft Teams, now Microsoft’s fastest growing business application ever with over 100 million users, implemented on an enterprise cloud-based platform as part of Microsoft 365 (formerly Office 365). This choice too has been validated as Microsoft is now positioning Teams as the replacement for Skype for Business, used by many major corporations and organizations worldwide. The intent is to continue to leverage the latest Microsoft enterprise-level technology for INCOSE communication and collaboration. The Microsoft 365 platform is more than a set of office and communication products—it includes capabilities to quickly extend and build specialized applications and connections to support the way INCOSE works. It’s an enabling technology for INCOSE’s forward progress.

---

<sup>6</sup> [https://en.wikipedia.org/wiki/Dune\\_\(novel\)](https://en.wikipedia.org/wiki/Dune_(novel))



As systems engineers and systems thinkers, we should be able to see how our IT systems function as parts of the larger INCOSE system, analyze communication paths, and optimize the overall system's performance.

The world of systems engineering, and INCOSE membership, skews to (shall we say) pre-millennial generations, and many of us did not grow up with anything like current online technology. Even as engineers in highly technical fields, we may often be behind the times in our own knowledge and use of digital technology. Unfamiliarity and lack of knowledge may breed unwarranted fear in areas like security, privacy, and perceived value. But we need not use our fear of hacking, GDPR<sup>7</sup>, or email spam to resist or avoid new technology.

My strategy here is education—there is no substitute for learning how to handle technology safely and securely, while still achieving maximum benefit. I plan to prioritize the education of INCOSE members on the IT technologies available in the world and in INCOSE. As an example, did you know that the vast majority of headline-making hacking incidents involve social engineering—the digital version of a “con game” where hackers take advantage of users’ lack of knowledge about how security works, to dupe them into revealing passwords or installing malicious software? We should probably fear fallible humans more than we fear technology!

Learning new technology can be fun and highly productive. I believe that systems engineers, when we combine our systems knowledge with the ability to learn and use current and future technologies, can be key contributors to realizing *a better world through a systems approach*.

## About the Author



**Barclay R. Brown** is an Engineering Fellow at Raytheon Technologies, focusing on model based systems engineering and artificial intelligence. Before joining Raytheon in 2018, he was with IBM for 14 years, serving in the public sector practice of IBM Global Business Services, where he was the lead systems engineer for some of IBM's largest development projects. He also served as the worldwide lead for the IBM systems engineering software business in the aerospace and defense industry. Prior to IBM, he managed large software development and web site projects and consulted on software development methods.

Dr. Brown has been a practitioner, consultant, and speaker on systems engineering and software development methods for over 25 years. He received a bachelor's degree in Electrical Engineering followed by master's degrees in Psychology and Business and a Ph.D. in Industrial and Systems Engineering. He is a certified Expert Systems Engineering Professional (ESEP), Certified Systems Engineering Quality Manager, OMG Certified Systems Modeling Professional, the former INCOSE Director for the Americas, and an adjunct professor at several universities.

---

<sup>7</sup> [General Data Protection Regulation](#)

---

## 3. NOTABLE ARTICLES, WEBINARS AND PRESENTATIONS

---

### 3.1 Some Tools for Use in Program Management and Systems Engineering Integration

*A presentation by Robert Halligan*

The quest for integration of program managers' and chief systems engineers' mindsets integration continues. Although a lot of attention has been given to this critical need, achievement of the needed perspective remains elusive. Robert Halligan has provided a set of useful tools that should be leveraged. What can you do to further strengthen and improve the practice of systems engineering?

[View the Presentation](#)

### 3.2 SysML V2 Updates

At regular intervals, the SysML Submission Team (SST) publishes a release of the pilot implementation of the new SysML v2, including implementations for Jupyter Notebook and for Eclipse, the current state of development of the SysML v2 specification and client implementations of the new SysML v2 API & Services. The releases give a deep but not easy insight into the current state of SysML v2. The Podcast unpacks a release and explains and tries out the individual parts.

[SysML2 Public Incremental Release 2020-10](#)

A very useful overview Podcast of the under-development MBSE language SysML V2 is at <https://www.youtube.com/watch?v=He6p9OOVido>. The Podcast is led by Tim Weilkiens and Christian Muggeo.

MBSE Audio-Podcast: [Spotify](#)

iTunes: <https://podcasts.apple.com/de/podcast...>

Links: [MBSE-Podcast Website](#)

### 3.3 INCOSE President Address: INCOSE International Workshop 2021

*by*

**Kerry Lunney**

INCOSE's first virtual workshop.

[View the Presentation](#)

## 3.4 Systems Engineering as a Data-Driven and Evidence-Based Discipline

by

**Ron S. Kenett, Avigdor Zonnenshain, and Robert S. Swarz**

Data and information are considered today as the “new oil” or the “new gold” in almost all aspects of life and economic domains, such as industry, healthcare, education, entertainment and more. The so-called 4th Industrial Revolution is based on the digital transformation derived from the Big Data revolution, through the capability of storing huge amounts of data and performing very sophisticated analytics. In this paper, we present opportunities for Systems Engineering (SE) to evolve towards a data-driven and evidence-based discipline, thereby making better systems and engineering decisions. We discuss how systems engineers can apply data-driven characteristics through systems engineering processes and programs. The classical Model-Based Systems Engineering (MBSE) approaches are presented here as powerful tools to collect, generate, and analyze data on systems under development. In addition, the “Digital Twin” concept is presented here in the context of system design. We highlight the challenges for systems engineering to become an evidence-based discipline. Moreover, we emphasize research and development in systems engineering processes using statistical techniques in the design and analysis of systems testing, and Model-Based Systems Engineering (SE) as a source for evidence-based engineering decisions. The success of data driven SE in organizations depends on the information and data analytics infrastructure in these organizations. An information quality framework is proposed for evaluating organizational information infrastructure. In addition, it is proposed to assess the data analytics maturity level in organizations. The level of data analytics is the basis for planning implementation programs of data-driven and evidence-based systems engineering. The paper concludes with a case study based on a real-life complex project, and lessons learned for effective data analytics implementation.

[Access the Paper](#)

## 3.5 Analysis: A Love-Hate Relationship with “Systemness”

by

**Kevin Nortrup** | [Kevin@SugarCreekSolutions.com](mailto:Kevin@SugarCreekSolutions.com)

CSEP, CPHIMS, LSSGB, FHIMSS, DSHS

Version 1.0 December 25, 2020

**Copyright © 2020 by Kevin Nortrup. All rights reserved.**

### Introduction

The first time that I encountered the term, “systemness”, I experienced something akin to an allergic reaction. That reaction only increased with continued exposure both to the term and to its application. At one point, I characterized “systemness” as a terminological weed that had invaded the lawn of systemic discussion: an intrusive, distractive nuisance whose ongoing spread was difficult to control. (“Hey, Kevin, tell us what you really think.”) I suspect that others in the systems community might have felt similarly.



However, one person can find utility and beauty in what another sees only as a useless weed. For example, dandelions are commonly regarded as weeds in North America, yet their greens can make salads that are tasty and nutritious, their blossoms can make a deliciously dry wine, and their brilliant yellow flowers and white seed heads can bring a lovely aesthetic accent to a meadow, pasture, or lawn.

So: can the systems community find something useful, nutritious, and even attractive in “systemness” as well?

### **What’s wrong with “systemness?”**

To those with a penchant for precision, the construction of “systemness” itself might seem awkward. Typically in the English language, the suffix, “-ness,” attaches to adjectives to transform them into nouns: for example, awkwardness is the state, characteristic, or extent of being awkward. Since “system” is not an adjective, “systemness” must be understood as the state, characteristic, or extent of being a system. In the absence of accepted equivalents or alternatives, perhaps one can come to toler-ate such nonstandard construction.

However, the common applications of the term can be frustrating to those who work with systems routinely.

In the early years after its coinage, “systemness” was ambiguous in its definition and inconsistent in its usage. Then for a while, it was associated with information technology (IT), whose are the first (and sometimes only) systems that many people can identify easily. Eventually, it came to describe optimizing and leveraging corporate-level strengths (such as economies of scale and integrated, centralized, and standardized operation across strategic business units or physical campuses) – particularly in the healthcare and education industries.

At first glance, this may not appear to be problematic. It certainly is important to recognize a corporate entity as a sociotechnical system – to approach, operate, and improve it holistically as an integrated organization, not merely as a loose amalgam of siloes. Acknowledging that “the whole is greater than the sum of its parts” and prioritizing that greater good is critical to successful improvement of the operation of any system.

However, while a corporate entity is a sociotechnical system, so is every strategic business unit, physical campus, department, floor, team, even workspace or treatment room: each a hierarchical subsystem. Similarly, the industry-segment itself is a super system of all such organizations, and the collective economy of all industries is a still greater super system. Finally, process-improvement, innovation, and research are metasystems that create and refine those hierarchical systems.

Human/systems integration, capability systems, mission systems, enterprise systems, and purposeful human activity systems: there are multiple systems approaches and models that can help to manage the complexity of how organizations pursue their objectives in an increasingly complex world. The full power of a systemic approach extends far beyond its application solely to the enterprise or corporate level of an organization. Unfortunately, if “systemness” precludes that greater scope of applicability, then it can discourage systemic inquiry, analysis, synthesis, and improvement in areas and at levels where such is desperately needed.

In summary, “systemness” can be little more than a thin “system” veneer atop familiar, insufficiently scoped reductionism. A tiny dose of constrained systems thinking may vaccinate people against openness to more widely applicable systems thinking, causing them to be resistant or even immune to perceiving the need for a consistently holistic approach to analysis and interventions throughout their organization.

## “Yes – and...”

Despite its potential for counter productivity, “systemness” is widespread and still growing in its usage – it even has a [Wikipedia article](#). It has the allure of most hot new trends: it sounds vaguely familiar yet exotic simultaneously; it has some basis in genuine, effective principles; it hints at substantially better results through minimally disruptive changes; and it has been boiled down into simplistic, almost formulaic approachability. For better or worse, it’s not going away anytime soon.

What, then, should be the response of the systems community to “systemness”? Clues for such may be found in an unexpected domain: that of improvisational theatre (improv).

In improv, there is no authoritative, over-arching script. The players take the germ of an idea and build upon it sequentially in real time. At every turn, each player is faced with a decision on how to respond to the storyline that has been inherited:

1. “No – but...” – This refutes and redirects what has gone before. Since it kills momentum and risks devolving into a tug-of-war between conflicting ideas, this approach is regarded as least productive.
2. “Yes – but...” – This acknowledges what has gone before but still closes it down. Although this approach is less disruptive than “No – but...”, the audience still may find it disjointed and unsatisfying.
3. “Yes – and...” – This not only acknowledges what has gone before but explicitly builds further upon it. This approach is widely regarded as the essential foundation for successful improv.

Therefore, if the systems community cannot beat back “systemness”, perhaps they can join it – or at least, walk alongside it and gently steer it.

**Yes**, “systemness” is dealing effectively and holistically with the top-level enterprise – **and** such a systems approach can apply to (and produce substantial improvements in) any subsystem, supersystem, or metasystem of that enterprise – indeed, any purposeful collaboration of people, process, and technology. In fact, there’s an extensive body of inquiry and practice regarding systems and their characteristics, behaviors, analysis, and design. Would you like to know more?

Can we in the systems community engage and expand the discussion of “systemness” so that it opens, not closes, the door to greater understanding and application of systemic concepts and approaches? Can we do so, without confounding the audience and the other players?

Meanwhile, perhaps I’ll search on-line for recipes with dandelion greens...

## About the Author



**Kevin Nortrup**, CSEP, CPHIMS, LSSGB, FHIMSS, DSHS, graduated *summa cum laude* from the University of Illinois in computer engineering and has extensive experience in designing, implementing, analyzing, and troubleshooting complex sociotechnical systems. As Principal at Sugar Creek Solutions, he champions a transdisciplinary systems approach to transform process-improvement into systems-improvement in healthcare and in other industries. He is a Certified Systems Engineering Professional and holds a Lean Six Sigma Green Belt in healthcare.

Kevin is a member of INCOSE, IISE and SHS; a director of IISE-Indiana and of ISSCSH; chair of the Enterprise Systems Working Group of INCOSE; and a HIMSS Fellow and an SHS Diplomate.

## 3.6 Ten Characteristics of an Outstanding Engineering Leader

This article from the Case School of Engineering at Case Western Reserve University (USA) advises that engineering leadership requires certain skills that do not always come naturally to those who have decided to pursue a career in Science, Technology, Engineering, and Math (STEM).

[Read the Article](#)

---

## 4. SYSTEMS ENGINEERING NEWS

---

### 4.1 INCOSE Sector Updates—Asia-Oceania

**Masaatsu Kusunoki** | [mazkusunoki@gmail.com](mailto:mazkusunoki@gmail.com)

JCOSE, Japan Chapter of International Council on Systems Engineering, hosted a virtual symposium between 2-3 September 2020; 100 participants across 7 different countries from the Asia-Oceania region attended. Attendees highly praised the keynote speeches delivered by David Walden, Takashi Kono (Architecture Design Department, IPA) and Dr. Yoshiki Yamagata (Center for Global Environmental Research, National Institute for Environmental Studies). The diverse invited speakers' lineup, and the well-received virtual café led attendees to request a regular JCOSE café session to continue.

### 4.2 Systems Engineering Body of Knowledge

**Robert Cloutier** | [rcloutier@southalabama.edu](mailto:rcloutier@southalabama.edu)

Version 2.3 of the SEBoK went live the first week of November 2020. This is the 14th release of the SEBoK since the fall of 2012.

Version 2.4 is scheduled for April/May 2021. If you have plans, thoughts, or interest in contributing content to the next release, please let [me](#) know.

## 4.3 India Systems Engineering Webinar Series

Stueti Gupta | [stueti.gupta@gmail.com](mailto:stueti.gupta@gmail.com)

The India Chapter of INCOSE continues to build momentum among the members in India via the *India Webinar Series*. Thus far, we have organized 15 webinars with 560 participants. The webinars have encouraged learning and networking among systems engineering professionals. The webinar recordings are available in INCOSE YouTube Channel—INCOSE India playlist here: ([https://www.youtube.com/watch?v=y3k\\_CQbSbFM&list=PL3wD0Sb1jb1LxR+0sP61G5E4WFXcf1ZVgW](https://www.youtube.com/watch?v=y3k_CQbSbFM&list=PL3wD0Sb1jb1LxR+0sP61G5E4WFXcf1ZVgW))

- Functional Safety—Automotive/Avionic Systems, Session II – Mr. Reveendra Menon, CSEP, principal member, technical staff, SENZOPT Technologies PVT LTD, Bangalore.
- Knowledge-Centric Integrated Systems Modelling – Dr. Swaminathan Natarajan, chief scientist, Tata Consultancy Services Research and INCOSE systems science working group co-chair, ISO 30103 product quality achievement standard editor, and a SysML v2 standardization contributor.
- INCOSE Systems Engineering Professional Certification and CTI preparation course overview – René King, managing director, Certification Training International
- The Surprising Benefits of Creating a Failure Resume – Tim Boyd, senior systems manager, Northrop Grumman
- Using System Dynamics to Understand and Model Complex Problems – Büşra Atamer, researcher and consultant, PhD (Middle East Technical University, Turkey)

If you would like to connect with the INCOSE India Chapter, please email the chapter at [IncoselIndiaChapter@gmail.com](mailto:IncoselIndiaChapter@gmail.com).

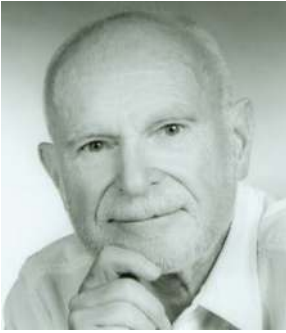
You can follow the Chapter activities on [LinkedIn](#) and [Twitter](#).

## 4.4 Named Professorship at Watson College (USA) Honors Systems Science Pioneer



Luis Rocha, PhD '97, to start next fall as first George Klir Professor in Systems Science

In the fall of 2021, thanks to the generosity of an anonymous donor, Binghamton University will inaugurate the [George Klir](#) Professor in Systems Science. The position, in honor of Klir's groundbreaking work in the field of complex systems, is the first named professorship for the [Thomas J. Watson College of Engineering and Applied Science](#), and it will be part of the [Department of Systems Science and Industrial Engineering \(SSIE\)](#).



Fittingly, the first to fill the professorship will be a former Klir student: [Luis Rocha, PhD '97](#). “George Klir was a true pioneer in his discipline; his work evolved from systems modeling and simulation to intelligent systems and fuzzy logic,” said [Executive Vice President for Academic Affairs and Provost Donald Nieman](#). “He was the epitome of what a distinguished professor should be. It says a lot about the program that Dr. Klir and his successors have built that we are able to recruit a scholar of Dr. Rocha’s stature back to Binghamton to carry on the tradition that his mentor established.”

Klir, who [passed away in 2016 at age 84](#), earned his PhD in computer science in his native country at the Czechoslovakia Academy of Sciences in 1964. After emigrating to the U.S. with his wife, Milena, he joined the then-SUNY Binghamton faculty in 1969 and became a pioneer in the emerging field of complex systems and systems science. He served as the chair of Department of Systems Science (1978-94) and director of the Center for Intelligent Systems (1994-2000), attaining the rank of SUNY distinguished professor in 1984. Before his retirement in 2007, more than 30 doctoral students had studied with him.

Because researchers were still formulating the discipline of systems science, Klir literally wrote the book on it — well, 23 books to be precise, including the seminal textbook *Facets of Systems Science* (1991). He also founded the *International Journal of General Systems* in 1974 and served as its editor-in-chief until 2014. In addition, he edited ten volumes in the International Book Series on Systems Science and Engineering, sponsored by the International Federation on Systems Research.

In more than 300 research papers and other articles, Klir explored topics such as systems modeling, logic design, computer architecture, discrete mathematics, intelligent systems, soft computing, fuzzy set theory and fuzzy logic.

[More Information](#)

## 4.5 SysML for Models of Digital Twins

Systems Modeling Language (SysML®) is claimed by some to be the future of Model-Based Systems Engineering (MBSE) and can be applied to a wide range of tasks in industry, engineering, construction, information technology, science, and education.

The Digital Twin concept is being increasingly embraced as a driving technology for future building and construction projects, and it has a big role to play in development and implementation of environmentally friendly technologies and the ongoing operation of sustainable buildings.

In its simplest form, a Digital Twin manages a *digital entity* that encapsulates and replicates a real-world *physical entity* (such as a building) as closely as possible. Data from sensors and regular measurements of a deployed physical entity are used to keep the digital entity in sync. *Variants* spawned from the digital entity may also be used to plan optimizations and drive the physical entity into a preferred state (such as for reducing energy consumption).

A screencast “The Webel Digital Twin Pattern for SysML: Part 1: Simulating acquisition or creation of physical assets using Activities and StateMachines in Cameo Simulation Toolkit.” takes this subject further.

[View the Video](#)

## 4.6 INCOSE Annual Impact Report

Every year INCOSE impacts the world of systems engineering. This 30th year of INCOSE's existence was no exception. INCOSE Annual Impact Report's recaps 2020 where INCOSE navigated international challenges while providing a valuable membership experience that serves a global community of engineers:

Kerry Lunney said in a press release from November 2020:

'We look forward to working with you in 2021 to continue to grow, improve, and inspire the global systems engineering community and INCOSE in our fourth decade.

We wish you the best year's end and start to 2021.'

---

## 5. FEATURED ORGANIZATIONS

---

### 5.1 Center for Intelligent Systems (CIS)

The École polytechnique fédérale de Lausanne (EPFL) (Switzerland) Center for Intelligent Systems brings together researchers working on various aspects of creating intelligent systems.

The fundamental research and its applications cross the boundaries of schools at EPFL and create opportunities for direct interactions with Swiss industry.

The Center for Intelligent Systems (CIS) at EPFL, a joint initiative of the EPFL schools [ENAC](#), [IC](#), [SB](#) and [STI](#), seeks to advance research and practice in the strategic field of intelligent systems. Such systems are embedded with elements of artificial intelligence and are emerging in response to the convergence of the virtual and physical worlds through the development of autonomous systems, wearable technologies, robotic co-workers, smart devices, drone technology, augmented reality, intelligent houses, digital twins, intelligent manufacturing, and many other applications. When available, such technologies will have profound implications for many areas including manufacturing, transportation, commerce, employment, healthcare, government, legal, security, privacy, and education.

#### CIS Goals

- To bring people together within EPFL under the umbrella of Intelligent Systems
- To encourage ambitious collaborative projects to build Intelligent Systems
- To act as the point of contact and collaboration for Swiss industry

[More Information](#)

### 5.2 Acquisition Innovation Research Center

The Acquisition Innovation Research Center (AIRC) is a new center, hosted within the Systems Engineering Research Center (SERC) at the Stevens Institute of Technology (USA), to support the continued transformation of the U.S. defense acquisition system to help the DoD innovate and respond to the rapid technological advancements critical to our national security interests.

[More Information](#)



## 5.3 Institute of Industrial and Systems Engineers (IISE)

Founded in 1948, the Institute of Industrial and Systems Engineers (IISE) is the world's largest professional society that is dedicated to the application, education, training, research, and development of industrial and systems engineering. Its members seek to improve the ability of their organizations to solve complex and critical problems around the world and across industries.

IISE defines Industrial and Systems Engineering (ISE) as follows:

Industrial and systems engineering is concerned with the design, improvement, and installation of integrated systems of people, materials, information, equipment, and energy. It draws upon specialized knowledge and skill in the mathematical, physical, and social sciences together with the principles and methods of engineering analysis and design, to specify, predict, and evaluate the results to be obtained from such systems.

IISE has over 300 university and professional chapters around the world, and its specialty communities include 3 societies, 12 divisions, and several technical interest groups. IISE offers conferences, publications, webinars, on-line courses, social networking, and a variety of other educational and leadership opportunities for industrial and systems engineers at all stages of their education and professional career.

[IISE Website](#)

## 5.4 Society for Health Systems (SHS)

The Society for Health Systems (SHS) is a specialized healthcare community within the Institute of Industrial and Systems Engineers (IISE). SHS aspires to improve lives through better healthcare delivery, by creating a culture of continuous improvement within healthcare organizations. Its action-plan towards those ends includes:

- (1) Cultivate a community of healthcare improvement professionals that is passionate about making a difference;
- (2) Foster innovative problem solving through systems thinking;
- (3) Develop healthcare leaders, professionals, clinicians, and students; and
- (4) Influence leaders and decision makers on the future design of healthcare.

SHS encourages the exchange of ideas and innovative techniques among healthcare professionals through conferences, webinars, newsletters, and a variety of members-only content through an on-library of tools and resources. Its members include management engineers, nurses, CEOs, directors of continuous improvement, administrators, clinicians, physicians, department managers, and students – all of whom can benefit from opportunities for volunteering, leadership, and professional development.

The annual Healthcare Systems Process Improvement Conference of SHS offers speakers, workshops, educational sessions, poster sessions, networking opportunities, and vendor exhibits. Conference attendees have access to the latest in operational and quality improvement tools, methods, and concepts – such as Lean, Six Sigma, productivity, benchmarking, simulation, and project management.

[SHS Website](#)

---

## 6. NEWS ON SOFTWARE TOOLS SUPPORTING SYSTEMS ENGINEERING

---

### 6.1 Mentor Graphics Becomes a Part of Siemens EDA

Following the acquisition of Mentor Graphics by Siemens in 2017, Mentor will now officially become Siemens EDA, a part of Siemens Digital Industries Software, effective January 2021. Mentor, a Siemens Business, provides software products supporting electronic design automation (EDA).

### 6.2 Siemens RapidAuthor 13 Introduces New Functionality for Authoring and Illustration

RapidAuthor for Teamcenter enables technical authors to work in the PLM environment and interact directly with engineering design geometry and bills of material (BOM) and create 2D/3D interactive parts catalogs, maintenance manuals, training materials, work instructions that reflect the product description.

The new features of RapidAuthor include:

- Integration with Teamcenter Active Workspace, including support for creation of RapidAuthor projects; import of engineering data into projects; and project updates
- A new Viewpoints window that improves the management of project viewpoints
- Coaxial alignment of objects using the new cylindrical surfaces detection feature of the 3D manipulator
- 2D editing has new features for creating projections, and improvements in creating and editing polygons, circles and ellipses
- Support for additional CAD formats: SolidWorks 2020, CATIA 5-6 R2019 (R29), Parasolid 32.0, NX 1899, Revit 2020
- Import of textures and texture coordinates from SolidWorks, Autodesk and several other CAD formats

Find out more [here](#)

Watch the video Rapid 13: [What's New in Authoring Process](#)



---

## 7. SYSTEMS ENGINEERING PUBLICATIONS

---

### 7.1 A Systematic Framework for Exploring World Views and its Generalization as a Multi-purpose Inquiry Framework

by David Rousseau and Julie Billingham

Centre for Systems Philosophy, Surrey KT15 1EL, UK

Published: 10 July 2018 in *Systems*

*Special Issue: Systems Thinking*

<https://doi.org/10.3390/systems6030027>

#### Abstract

Systems science methodologies do not have a consistent way of working with worldviews, even though determining stakeholder perspectives is central to systems thinking. In this paper, we propose a comprehensive “Worldview Inquiry Framework” that can be used across methodologies to govern the process of eliciting, documenting, and comparing the worldviews of stakeholders. We discuss the systemicity of worldviews and explain how this can help practitioners to find the roots of stakeholders’ disagreements about value judgements. We then generalize the structure of the Worldview Inquiry Framework to produce a “General Inquiry Framework” that can be used to govern an inquiry process in other contexts. We show that the presented Worldview Inquiry Framework is a special case of this General Inquiry Framework and show how the General Inquiry Framework can be tailored for other contexts such as problem solving, product design, and fundamental research.

[Read the Article](#)

#### About the Systems Journal

*Systems* (ISSN 2079-8954) is an international **peer-reviewed** open access journal on systems theory in practice, including fields such as systems engineering management, systems based project planning in urban settings, health systems, environmental management and complex social systems, published quarterly online by MDPI. **The International Society for the Systems Sciences (ISSS)** is affiliated with *Systems* and its members receive a discount on the article processing charges.

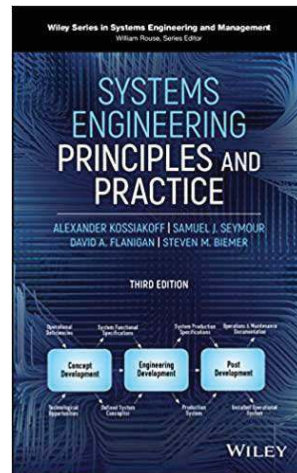
- Open Access—free for readers, with article processing charges (APC) paid by authors or their institutions.
- High Visibility: Indexed in the Emerging Sources Citation Index (ESCI – Web of Science) and in DBLP Computer Science Bibliography.
- Rapid Publication: manuscripts are peer-reviewed and a first decision provided to authors approximately 20.4 days after submission; acceptance to publication is undertaken in 3.8 days (median values for papers published in this journal in the first half of 2020).

Recognition of Reviewers: reviewers who provide timely, thorough peer-review reports receive vouchers entitling them to a discount on the APC of their next publication in any MDPI journal, in appreciation of the work done.

[Imprint Information](#)

## 7.2 Systems Engineering Principles and Practice

by [Alexander Kossiakoff](#), [Steven M. Biemer](#), [Samuel J. Seymour](#), and [David A. Flanigan](#)



**From the Amazon.com Webpage:**

*Systems Engineering: Principles and Practice, 3rd Edition* is the leading interdisciplinary reference for systems engineers. The up-to-date third edition provides readers with discussions of model-based systems engineering, requirements analysis, engineering design, and software design. Freshly updated governmental and commercial standards, architectures, and processes are covered in-depth. The book includes newly updated topics on:

- Risk
- Prototyping
- Modeling and simulation
- Software/computer systems engineering

Examples and exercises appear throughout the text, allowing the reader to gauge their level of retention and learning. *Systems Engineering: Principles and Practice* was and remains the standard textbook used worldwide for the study of traditional systems engineering. The material is organized in a manner that allows for quick absorption of industry best practices and methods.

Throughout the book, best practices and relevant alternatives are discussed and compared, encouraging the reader to think through various methods like a practicing systems engineer.

Publisher: Wiley; 3rd edition (July 8, 2020)

Format: Kindle, hardcover

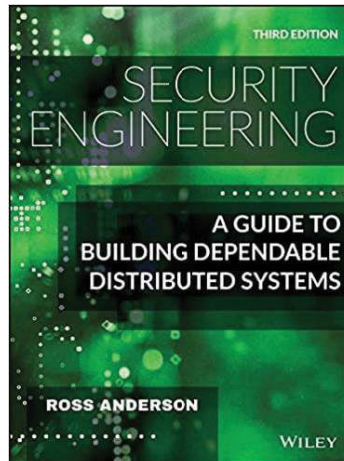
ISBN-10: 1119516668

ISBN-13: 978-1119516668

[More Information](#)

## 7.3 Security Engineering: A Guide to Building Dependable Distributed Systems

by Ross Anderson



### From the Amazon.com Website:

In *Security Engineering: A Guide to Building Dependable Distributed Systems, Third Edition* Cambridge University professor Ross Anderson updates his classic textbook and teaches readers how to design, implement, and test systems to withstand both error and attack.

This book became a best-seller in 2001 and helped establish the discipline of security engineering. By the second edition in 2008, underground dark markets had let the bad guys specialize and scale up; attacks were increasingly on users rather than on technology. The book repeated its success by showing how security engineers can focus on usability.

Now the third edition brings it up to date for 2020. As people now go online from phones more than laptops, most servers are in the cloud, online advertising drives the Internet and social networks have taken over much human interaction, many patterns of crime and abuse are the same, but the methods have evolved. Ross Anderson explores what security engineering means in 2020, including:

- How the basic elements of cryptography, protocols, and access control translate to the new world of phones, cloud services, social media and the Internet of Things
- Who the attackers are – from nation states and business competitors through criminal gangs to stalkers and playground bullies
- What they do – from phishing and carding through SIM swapping and software exploits to DDoS and fake news
- Security psychology, from privacy through ease-of-use to deception
- The economics of security and dependability – why companies build vulnerable systems and governments look the other way
- How dozens of industries went online – well or badly

Publisher: Wiley; 3rd edition (November 25, 2020) 32 of 41

Format: Kindle, hardcover

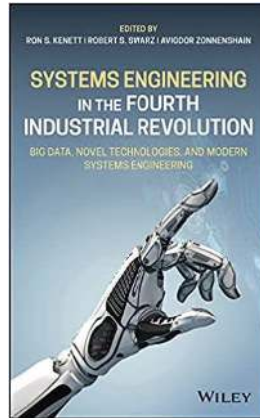
ISBN-10: 1119642787

ISBN-13: 978-1119642787

[More Information](#)

# 7.4 Systems Engineering in the Fourth Industrial Revolution: Big Data, Novel Technologies, and Modern Systems Engineering

by [Ron S. Kenett](#) (Editor), [Robert S. Swarz](#) (Editor), [Avigdor Zonnen-shain](#) (Editor)



## From the Amazon.com Website:

*Systems Engineering in the Fourth Industrial Revolution: Big Data, Novel Technologies, and Modern Systems Engineering* offers a guide to the recent changes in systems engineering prompted by the current challenging and innovative industrial environment called the Fourth Industrial Revolution—INDUSTRY 4.0. This book contains advanced models, innovative practices, and state-of-the-art research findings on systems engineering. The contributors, an international panel of experts on the topic, explore the key elements in systems engineering that have shifted towards data collection and analytics, available and used in the design and development of systems and also in the later life-cycle stages of use and retirement.

The contributors address the issues in a system in which the system involves data in its operation, contrasting with earlier approaches in which data, models, and algorithms were less involved in the function of the system. The book covers a wide range of topics including five systems engineering domains: systems engineering and systems thinking; systems software and process engineering; the digital factory; reliability and maintainability modeling and analytics; and organizational aspects of systems engineering. This important resource:

- Presents new and advanced approaches, methodologies, and tools for designing, testing, deploying, and maintaining advanced complex systems.
- Explores effective evidence-based risk management practices.
- Describes an integrated approach to safety, reliability, and cyber security based on system theory.
- Discusses entrepreneurship as a multidisciplinary system.
- Emphasizes technical merits of systems engineering concepts by providing technical models.

Written for systems engineers, *Systems Engineering in the Fourth Industrial Revolution* offers an up-to-date resource that contains the best practices and most recent research on the topic of systems engineering.

Publisher: Wiley; 1st edition (February 6, 2020)

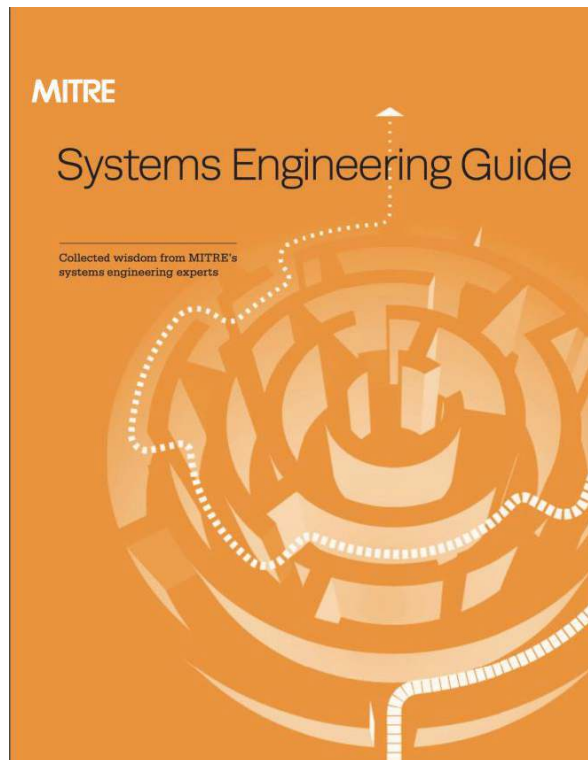
Format: Kindle, Hardcover

ISBN-10: 1119513898

ISBN-13: 978-1119513896

[More Information](#)

## 7.5 MITRE Systems Engineering Guide



MITRE is a federally funded USA organization with R&D centers and public-private partnerships. MITRE works across government to tackle challenges to the safety, stability, and well-being of the USA. The MITRE Systems Engineering Guide (SEG) was first launched in March 2010 as an internal MITRE resource. In late 2010, a government-only version was rolled out in response to many requests from MITRE staff to use it as a shared resource with their customers. In June 2011, an HTML version was published on [www.mitre.org](http://www.mitre.org) as a contribution to the wider systems engineering community. The rollout of the public SEG resulted in requests for it to be made available for popular mobile platforms, and in September 2012 an eBook version was posted on [www.mitre.org](http://www.mitre.org) in formats for the iPad, iPhone, Android, Kindle, and compatible devices. The SEG has been visited hundreds of thousands of times by individuals across the world.

Now it is available in hardcopy form. The SEG is the result of an effort involving nearly 200 individuals from across MITRE. The seminal idea for capturing the corporation's accumulated wisdom on a wide variety of important and timely systems engineering topics in a single, central location came from MITRE Corporate Chief Engineer Dr. Louis S. Metzger. He inspired the vision of the SEG as a resource that provides an "in the trenches" view of the typical problems, pitfalls, conundrums, and tight corners that practicing systems engineers are likely to find themselves in, together with best practices and lessons learned to avoid or mitigate the problems and enhance the likelihood of success. In this way, the SEG complements the many excellent systems engineering resources currently available.

The 726-page MITRE Systems Engineering Guide is downloadable at <http://www.mitre.org/sites/default/files/publications/se-guide-book-interactive.pdf>

---

## 8. EDUCATION AND ACADEMIA

---

### 8.1 Department of Systems Science and Industrial Engineering (SSIE)

**Thomas J. Watson College of Engineering and Applied Science**

**Binghamton University USA**

The SSIE Department has 26 faculty members, is in the top ten in terms of faculty size and research expenditures in systems science/industrial engineering departments in the United States, and has the second-largest doctoral program. According to U.S. News & World Report, SSIE also has the highest-ranked graduate program at Binghamton University.

[More Information](#)

### 8.2 Translating Systems Engineering for High School Teachers and Students: An Exploratory Study of Implementing some initial SE Concepts

*by Rashmi Jain, Mercedes McKay, Beth McGrath and Debra Brockway*

#### **Abstract**

Systems engineering is a life cycle approach to engineering design: the integration of numerous technical and non-technical disciplines toward the development of new products, systems and services. This paper describes the experiences of the authors in designing and implementing a three-year project to engage high school classes in a geographically-distributed systems engineering design project that addresses relevant, social challenges of interest to students worldwide. Collaborating with others around the world to develop a solution to an engineering problem, students are introduced to systems-thinking, team work, effective communication and other 21st century workforce skills. This innovative project aims to increase the number of students interested in pursuing engineering as a career and to increase the pool of teachers familiar with engineering design and systems thinking.

[More Information](#)

### 8.3 Systems Modeling and Engineering Research Laboratory, Worcester Polytechnic Institute (USA)

The Systems Modeling and Engineering Research Laboratory (SMERL) focuses on two key research areas, policy modeling and model-based analysis for complex systems. The work focuses on 1) Building evidence-based policy analysis and demonstrating the value of using systems engineering methods in policy design, development, and implementation, and 2) Leveraging model based systems engineering to integrate system design with safety analysis.

[Read More](#)



---

## 9. SYSTEMS ENGINEERING-RELEVANT WEBSITE

---

### The Unwritten Laws of Systems Engineering

The Unwritten Laws of Systems Engineering is a website where David F. McLinton describes, in his terms, the spoken laws that abound in any engineering project. The laws are really simple basic truths that arise in typical engineering discussions. We didn't hear them in engineering school but they came through loud and clear in that school of hard knocks, the engineering project. The initial three laws are

- (1) Everything interacts with everything else;
- (2) Everything goes somewhere; and
- (3) There is no such thing as a free lunch.

[Several other laws](#) capture the lessons learned in various activities in the systems engineering process.

#### 9 Trends to Watch in Systems Engineering and Operations

An [outline of trends](#) for businesses that rely on systems engineering and operations to be mindful of in these times.

---

## 10. STANDARDS AND GUIDES

---

### 10.1 IEEE 1028-2008 – IEEE Standard for Software Reviews and Audits

IEEE 1028-2008 defines five types of software reviews and audits, together with procedures required for the execution of each type. This standard is concerned only with the reviews and audits; procedures for determining the necessity of a review or audit are not defined, and the disposition of the results of the review or audit is not specified. Types included are management reviews, technical reviews, inspections, walk-throughs, and audits. The standard was withdrawn in November 2019.

The standard, although no longer current, contains much content on technical reviews useful in relation to both software and physical systems. The home of the standard was the C/S2ESC Committee – Software & Systems Engineering Standards.

## 10.2 IEEE 42020-2019 – ISO/IEC/IEEE International Standard – Software, Systems and Enterprise — Architecture Processes

This standard complements the architecture-related processes identified in ISO/IEC/IEEE 15288, ISO/IEC/IEEE 12207 and ISO 15704 with activities and tasks that enable architects and others to more effectively and efficiently implement architecture practices. Implementing these practices can help ensure that the architecture has greater influence on business and mission success. It specifies a coherent set of processes for governance, management, conceptualization, evaluation and elaboration of architectures, and activities that enable these processes. Users of this document can apply these processes in the context of: (1) understanding, development and evolution of entities through their life cycle stages such as conception, development, implementation, operation, sustainment, decommissioning, and disposal; (2) organization(s) acting as users, customers and providers of the solution specified by the architecture description; and (3) architecting of entities.

---

## 11. SOME DEFINITION TO CLOSE ON

---

### 11.1 Systems Architecture

1. System Architecture is abstract, conceptualization-oriented, global, and focused to achieve the mission and life cycle concepts of the system. It also focuses on high-level structure in systems and system elements

**Source:** [Wikipedia](#)

2. The structural design of elements.

**Source:** [Simplifiable](#)

### 11.2 Enterprise Architecture

1. A well-defined practice for conducting enterprise analysis, design, planning, and implementation, using a comprehensive approach at all times, for the successful development and execution of strategy

**Source:** [Wikipedia](#)

2. The vehicle for integrating the resources necessary to create a complete view of the organization, as well as to provide products and services to facilitate the organization's transition to an integrated environment with optimized processes that are responsive to change and to the delivery of the business strategy.

**Source:** [Real IRM](#)



## 11.3 Hierarchy

1. A series of ordered groupings of people or things within a system

**Source:** [Vocabulary.com](https://www.vocabulary.com)

2. A graded or ranked series

**Source:** [Merriam Webster](https://www.merriam-webster.com)

---

## 12. CONFERENCES AND MEETINGS

---

For more information on systems engineering related conferences and meetings, please proceed to [our website](#).

**The feature conference for this edition is:**

### The INCOSE International Workshop

**29-31 January 2021, Virtual Event**

From the INCOSE website:

INCOSE's International Workshop is the event of the year for systems engineers to contribute to the state of the art. Unlike INCOSE's annual International Symposium and other conferences, there are no paper, panel or tutorial presentations. Instead, attendees spend some days working alongside fellow systems engineers who are there to make a difference. Systems Engineers at all levels and from all backgrounds are encouraged to engage in working sessions, and contribute their knowledge and experience to take the discipline forward.

IW2020 facilitates working meetings for groups engaged in INCOSE's major projects and in international standards development, workshops to explore the Systems Engineering challenges in new sectors, opportunities for chapter leaders to meet and share best practice, support sessions to help you get the most out of INCOSE's shared working environment and a broad range of other technical meetings. Planned sessions will be published on the website as these become available.

There are two kinds of working group sessions at the IW:

Working sessions, where the focus is on improving and completing working group products. Working sessions are ideal for contributing with and learning from the real experts in the field. Attendees planning to attend Working sessions are encouraged to contact the relevant session leaders before the event to facilitate planning

Outreach sessions, where the focus is on disseminating the current state of the art to Workshop attendees with no or little previous exposure to the working groups. Attending an Outreach session is also the ideal opportunity to influence the future direction of a working group and perhaps the entry point for deeper working group involvement.

To register for the IW 2021, click [here](#).

---

## 13. PPI AND CTI NEWS

---

### 13.1 PPI Welcomes Eduardo to the Team



**Eduardo Muñoz**, based in Campinas, SP, Brazil, has joined the PPI team in the role of Staff Engineer. Eduardo holds a BEng in Mechanical Engineering, MSc – Research Masters in Aeronautical and Mechanical Engineering and a MBA from ITA – Aeronautics Institute of Technology, São José dos Campos, SP, Brazil.

### 13.2 Successful First Live-Online CTI Course in China

In December 2020, CTI David Mason and CTI China Business Developer Victoria Huang presented two courses to Chinese delegates in live-online format for the first time ever. David and Victoria have delivered dozens of courses in China in in-person format but never before has this course been presented with presentation and translation, virtually. Typically, a PPI or CTI live-online delivery would find all delegates sitting with a laptop in front of them and both presenters and delegates working remotely. In the case of this adapted live-online delivery, there was a webcast of David's presentation from where he is based in San Diego with Victoria's interpretation taking place the same room as the delegates. A lot of effort was put into reformatting the standard material to be suitable for this live-online delivery and a technological solution was employed. CTI China can be very proud to have scored 8s and 9s for this presentation and looks forward to delivering many more courses in China in a live-online format.

### 13.3 PPI and INCOSE Systems Engineering Tools Database (SETDB) Activity at the INCOSE IW 2021

At the virtual INCOSE International Workshop taking place from 29-31 January 2021, PPI and INCOSE will be conducting work sessions to overview the current status of the SETDB website and develop supporting operational procedures. The INCOSE-PPI SETDB project has been ongoing since January 2018 and it is wonderful to see the project winding down after three years of consistent effort from both organizations to reinstate this popular resource. PPI welcomes you to join us at the IW conference, and partake the in the open SETDB working sessions.

Here is the SETDB schedule for the IW 2021:

<b>Title</b>	<b>Day &amp; Date</b>	<b>Start (GMT +1:00)</b>	<b>Finish (GMT +1:00)</b>	
<b>SETDB v0.9 Review</b>	Monday 25 Jan	16:30	19:00	The SETDB WG CCB will be reviewing the SETDB v0.9 readiness for production release. JIRA issues will be reviewed for high priority and blocking issues.
<b>Establishing SETDB Tool Vendor Accounts</b>	Monday 25 Jan	17:00	18:30	SE Tool Vendor Briefing: After a brief introduction to the SE Tools Database the SETDB Team will execute a Use Case depicting a tool vendor requesting a Vendor Account to obtain login credentials. Once logged in, we will review the features to use to enter you company and product information and capture your product data for publication. Open to any interested SE Tool Vendor.
<b>SETDB Operational Procedure Development</b>	Wednesday 27 Jan	18:00	21:00	Working session to focus resources on the development of the SETDB Operational Procedures. The procedures in development include: Access Management, Mapping Tool Categories to SE Processes, Tool Vendor Registration, Tool Information Management, Configuration Management, Stakeholder Communications and Lifecycle Management.
<b>A Systems Engineering Tools Database Walkthrough</b>	Thursday 28 Jan	18:00	20:00	Walking through the SETDB: a live walk through of the Systems Engineering Tools Database from the initial access to the exploration of all SETDB functionality. Potential users, tool vendors and stakeholders are encouraged to attend. Questions and comments are encouraged.
<b>Walking Through the SETDB System</b>	Saturday 30 Jan	19:00	20:30	The SETDB Project Leaders will introduce the SETDB functionality including the Website landing page, Home page, Tool, Tool Vendor and Taxonomy browsing and searching. This will be a live session with interactive questions and answers. Stakeholders, users and tool vendors are encouraged to attend.
<b>Systems Engineering Process Mapping to SETDB Tool Categories Live Demonstration</b>	Sunday 31 Jan	13:00	15:00	The SETDB Project Team will demonstrate the Systems Engineering Tools Database Tool Category Explorer features. They will provide a quick overview and do a live walk through the SETDB Taxonomy Browser and the category mapping to the processes defined in the INCOSE SEH and the PPI Process Elements. Questions and suggestions are encouraged.

---

## 14. PPI AND CTI EVENTS

---

<https://www.ppi-int.com/ppi-live-online/>

For a full public PPI training course schedule, please visit <https://www.ppi-int.com/course-schedule/>

To enquire about PPI Live-Online™ training for your organization, please visit <https://www.ppi-int.com/corporate-training/>

For a full public CTI Live-Online™ INCOSE SEP Exam Preparation course schedule, please visit <https://certificationtraining-int.com/incose-sep-exam-prep-course/>

To enquire about CTI Live-Online™ INCOSE SEP Exam Preparation Training for your organization, please visit <https://certificationtraining-int.com/on-site-training/>

---

## 15. UPCOMING PPI PARTICIPATION IN PROFESSIONAL CONFERENCES

---

PPI will be participating in the following upcoming events. We support many events, and look forward virtually or physically meeting old friends and making new friends at the events at which we will be participating.

### **[The INCOSE International Workshop 2021](#)**

Date: 29 – 31 January 2021

Location: Virtual

### **[The INCOSE International Conference 2021](#)**

Date: 17 – 22 July 2021

Location: Honolulu, USA

***Kind regards from the PPI SyEN team:***

**Robert Halligan**, Editor-in-Chief, email: [rhalligan@ppi-int.com](mailto:rhalligan@ppi-int.com)

**Dr. Ralph Young**, Editor, email: [syen@ppi-int.com](mailto:syen@ppi-int.com)

**René King**, Managing Editor, email: [rking@ppi-int.com](mailto:rking@ppi-int.com)

**Project Performance International**

2 Parkgate Drive, Ringwood, Vic 3134 Australia

Tel: +61 3 9876 7345

Tel Brasil: +55 12 9 9780 3490 (Breno Bacci)

Tel UK: +44 20 3608 6754

Tel USA: +1 888 772 5174

Tel China: +86 188 5117 2867 (Victoria Huang)

Web: [www.ppi-int.com](http://www.ppi-int.com)

Email: [contact@ppi-int.com](mailto:contact@ppi-int.com)

Copyright 2012-2021 Project Performance (Australia) Pty Ltd, trading as  
Project Performance International

Tell us what you think of PPI SyEN. Email us at [syen@ppi-int.info](mailto:syen@ppi-int.info)