SYSTEMS ENGINEERING NEWSLETTER

brought to you by

Project Performance International (PPI)

SyEN 51 – March 16, 2017

Welcome back SyEN! Project Performance International is delighted to offer to you the first Revival Edition of its Systems Engineering Newsletter (SyEN).

Past subscribers to SyEN, together with clients of PPI, will continue to receive SyEN. If you are unsure of your subscription status, or wish to subscribe, please register here.

Best regards,

Robert Halligan CPEng FIE Aust

Managing Director

Project Performance International

SyEN is an independent free newsletter containing informative reading for the technical project professional, with scores of news and other items summarizing developments in the field, including related industry, month by month. This newsletter and a newsletter archive are also available at www.ppi-int.com.

Systems engineering can be thought of as the problem-independent, and solution/technology-independent, principles and methods related to the successful engineering of systems, to meet stakeholder requirements and maximize value delivered to stakeholders in accordance with their needs and values.

If you are presently receiving this newsletter from an associate, you may receive the newsletter directly in future by signing up for this free service of PPI, using the form at www.ppi-int.com. If you do not wish to receive future Systems Engineering Newsletter, please unsubscribe by clicking on the link at the bottom of this email.

We hope that you find this newsletter to be informative and useful. Please tell us what you think. Email us at syen@ppi-int.info.

IN THIS EDITION

Quotations to Open On

Read More…

Feature Article

Avoiding Off-ramps on the Way to Good Systems Engineering, by David Long

Read More…

Article

The Root Causes of Poor Software Quality: Bad Quality Metrics, Incomplete Quality Data, Inadequate Quality Education, by Capers Jones

Read More…

Systems Engineering News

- National Academy of Engineering - Engineering Challenges of the 21st Century
- Research at the Stevens School of Systems of Enterprises (SSE)
- SSE Faculty Share Insights on Modern Systems
- Engineering Smart Factories of the Future
- Creating a Less Wasteful Economy
- 3 Principles to Guide Designing for Social Change
- IIBA releases new multi-level, competency-based Certification
- Short Comedy Sketch Concerning System Requirements – “The Expert”

Read More…

INCOSE

- Revised INCOSE Vision and Mission Statements
- INCOSE SE Vision 2025
- INCOSE Working Groups
- Empowering Women as Leaders in Systems Engineering (EWLSE)
- INSIGHT

Read More…

Featured Organizations

- IEEE Technology and Engineering Management Society (TEMS)
- International Association of Engineers (IAENG)

Read More…

Systems Engineering Tools News

- INCOSE Tool Integration and Model Lifecycle Management Working Group
- The COBIT 5/CMMI Practices Pathway Tool

Read More…

Systems Engineering Publications

- Systems Engineering Innovation and Design
- The Seeds of Innovation
- Team of Teams
- Guideline for Systems Engineering within the Dutch Civil Engineering Sector
- Systems Engineering Jr. Handbook

Read More…

Education and Academia

- Worcester Polytechnic Institute (WPI) Systems Engineering and Leadership Institute (SELI)
- Research positions available at the University of Toyko in Formal Methods and Cyber-Physical Systems

Read More…

Some Systems Engineering-Relevant Websites

Read More…

Standards and Guides

- Ford Chief Engineer Nand Kochhar named Vice Chair of SAE International Technical Standards Board
- Object Management Group Chairs take Standards Work to Next Level at Coronado Membership Meeting
- Techstreet Facilitates Communication Concerning Standards
- National Electrical Manufacturers Association

Read More…

Some Definitions to Close On

- Engineering
- Quality

Read More…

Conferences and Meetings

Read More…

PPI News

Read More…

PPI and CTI Events

Read More…

PPI Upcoming Participation in Professional Conferences

Read More…

QUOTATIONS TO OPEN ON

"Don't assume that the original statement of the problem is necessarily the best, or even the right one."

One pragmatic principle of Systems Engineering – INCOSE

"Systems Engineering focuses on ensuring the pieces work

together to achieve the objectives of the whole."

Systems Engineering Body of Knowledge (SEBoK)

"Amazing things are all around us – one needs only to glance through the

Systems Engineering Newsletter (SyEN) and *follow up* on something of interest!"

Adapted from google.com

Image Source

Avoiding Off-ramps on the Way to Good Systems Engineering

by David Long

15 December 2016

There is no doubt that systems engineering is growing. It is growing in applicability as the degree of complication and complexity rises across many different application domains – aerospace & defense, transportation, automotive, energy, medical, consumer products, and more. It is growing in importance as we seek to deliver value efficiently and effectively, free from unintended consequences. But as systems engineering grows, there is a groundswell of frustration as well – a groundswell that increasingly what people call systems engineering is not really systems engineering at all.

This complaint has increasingly been raised in private discussions, small meetings, and public forums, and I share that sense and frustration. It is often framed in terms of individual practitioners and organizations performing “partial systems engineering” or “checklist systems engineering”, sometimes tinged with a sense of nostalgia for the early days of the practice. Rarely do I hear it framed in an “us versus them” mindset but more often in a constructive question concerning how do we make good systems engineering more accessible to all those who could benefit from good application and good practice.

It starts by recognizing that there are multiple off-ramps on the way to good systems engineering. The first off-ramp is seeing, intellectually grasping, and implementing only a subset of systems engineering – most often requirements engineering. There is absolutely nothing wrong with requirements engineering

when it is performed well. It is an essential part of good problem solving and good systems engineering. However, it is a very limited subset and is by no means a substitute for systems engineering.

How do we fall into this trap – creating and then taking this off-ramp? Quite often it comes from returning to our reductionist roots, which have served us so well since the Scientific Revolution. We break problems – and disciplines – down into smaller and smaller pieces in order to increase understanding and advance. We hope to assemble the pieces into a working whole. This approach has served us well for centuries but breaks down when addressing the interconnected, interdependent, highly complicated, and often complex problems, solutions, and environments of today. Hence the birth of systems engineering, but even this practice based on holism is subjected to reductionism as people seek to break systems engineering down into its constituent parts, practice each independently, and then assemble the right outcome. The moment we break systems engineering into parts, we both violate its fundamental principles and run the risk of individuals substituting a part for the whole – knowingly or, more often, unknowingly. This is exactly the case with requirements engineering as a decoy for systems engineering.

The second off-ramp is substituting process for thinking and creating so-called “checklist systems engineering.” Again, process is a critical component of performing anything complex in an appropriate and repeatable manner. However, checklist engineering is neither purpose nor totality, only a partial enabler. Here we need to be very aware and very careful of the way we position our practice because we are part of the problem. INCOSE’s most cited definition of systems engineering is “an engineering discipline whose responsibility is creating and executing an interdisciplinary process to ensure that the customer and stakeholder's needs are satisfied in a high quality, trustworthy, cost efficient, and schedule compliant manner throughout a system's entire life cycle” – a process-centric mentality. The most cited systems engineering standard is ISO/IEC/IEEE 15288:2015 – Systems and software engineering -- System life cycle processes. We must stop enabling – if not encouraging – this misinformed substitution of process for practice.

For those lacking a deeper understanding, process easily becomes a proxy. The thinking is that if I follow the process, I am doing the practice and will deliver the results. If the process is tailored to the circumstance and performed properly in the context of principles, supporting methods, and experience, the conclusion is absolutely true. However, we all know that the process executed in isolation is useless at best, harmful at worst – and I suspect we have all observed practitioners doing just this in the name of systems engineering.

The most recent off-ramp to emerge between many practitioners and good systems engineering is the rise of model-based systems engineering – or more specifically the misunderstanding of MBSE. Like requirements engineering and process, MBSE done right is an enabler to the modern practice of systems engineering to address the reality of today’s problems and solutions. However, MBSE is easily misunderstood – as a distinct approach, as a constrained practice to be performed in specific ways with specific notations, or most often simply as a series of diagrams. As is the case with the first two off-

ramps, sometimes the misunderstanding is simply naivety on the part of the practitioner. In other cases, again we share the blame in the way we have framed (or miss-framed) this tool in our systems engineering toolkit.

These off-ramps emerge – and are taken – for a variety of reasons. The first is by far the most innocent but also the most damaging – simple misunderstanding. Misunderstanding on the part of the practitioner who believes he is doing the right thing but simply lacks the greater knowledge. Misunderstanding on the part of the individual, group, or organization who builds the off-ramp thinking that they are enabling good practice when in reality they are creating a false substitute. The second is more frustrating to those committed to advancing the state of the practice – personal or commercial gain. Approximately fifteen years ago the developer of a then-popular requirements management tool told the story of starting with systems engineering and then "dumbing the tool down" (his language, not mine) until procurement could understand what was being sold. Today, I know many spearheading the charge of MBSE within organizations who are knowingly incorrectly positioning the implementation and approach in furtherance of their interests at the ultimate expense of the organization. The third case is errantly substituting what is expedient or efficient for what is effective. Sometimes this is cloaked in the language of simplification (after all, systems engineering can be quite complex). Sometimes it begins with the thought "if I can only get them started, then I can correct the path later." Too often, later never comes.

There are many other off-ramps on the way to good systems practice. We could attempt to enumerate them all, but the moment we do so other off-ramps would emerge. The answer is not to identify every possible error in the hopes of avoiding them. The answer is to return to first principles which will keep us on track throughout the journey. The answer is also to reduce the barriers to entry for good systems engineering, simplifying without becoming simplistic in order to help practitioners stay on course.

Whereas the oft-cited INCOSE focuses on interdisciplinary process, I prefer descriptions that highlight an interdisciplinary or transdisciplinary approach and practice that can be appropriately tailored to different levels of complication and complexity. Jon Wade of Stevens Institute emphasize two critical aspects – the lifecycle perspective and the creative act of design noting that "there is art in engineering, not just science and analysis." Richard Beasley of Rolls Royce says "systems engineering collects and organizes all the information to understand the whole problem, explores it from all angles, and then finds the most appropriate solution." Stanley McChrystal in *Team of Teams* describes systems engineering and addresses examples from across many different application domains without resorting to "tech speak". We can learn from all of these individuals as we work not to reduce or simplify systems engineering but instead to make the critical concepts more accessible to those with different levels of experience and varied backgrounds.

Resolutions are generally made in the New Year, but let us begin (early) with the end in mind. Let us resolve to reach out and welcome individuals and organizations to this systems engineering practice, recognizing that more today than ever before the world needs good systems engineering. Let us resolve to make systems engineering more intellectually accessible, not by creating decoys and becoming

simplistic but by emphasizing first principles and good communication. Let us resolve to combat off-ramps at every opportunity – refusing to take them ourselves, teaching and coaching those that we see moving off-track, and rejecting the creation of new off-ramps whatever the purpose.

If we are willing to make and follow through on these resolutions, we can move systems engineering from an oft misunderstood and misapplied process to an important and impactful practice, changing the world for the better. And isn't that exactly what resolutions – and systems engineering – are for?

ARTICLE

The Root Causes of Poor Software Quality: Bad Quality Metrics, Incomplete Quality Data, Inadequate Quality Education

by Capers Jones

Version 1.0 August 10, 2016 update 2nd January 2017

Abstract

The software industry has a bad and justified reputation for poor quality and low reliability. A key reason for poor quality is that quality measurements are embarrassingly bad. Metrics such as cost per defect and lines of code distort reality and conceal software quality. Software measures omit important data, use hazardous metrics, and are not sufficient to show the effectiveness of various quality control methods such as static analysis, inspections, and testing.

Software quality depends upon two important variables. The first variable is that of "defect potentials" or the sum total of bugs likely to occur in requirements, architecture, design, code, documents, and "bad fixes" or new bugs in bug repairs. The second important measure is "defect removal efficiency (DRE)" or the percentage of bugs found and eliminated before release of software to clients.

The metrics of Defect Potentials and Defect Removal Efficiency (DRE) were developed by IBM circa 1970 and are widely used by technology companies and also by insurance companies, banks, and other companies with large software organizations.

Web: www.Namcook.com

Email: Capers.Jones3@gmail.com

Introduction

Poor software quality is an endemic problem of the software industry. Many large software system projects are canceled due to poor quality. Almost all large system projects run late and are over budget due to poor quality. There are many lawsuits for poor quality. There are billions of dollars wasted by software customers due to customer damages caused by poor quality. There are also rapidly increasing numbers of cyber-attacks most due to poor software quality. Why do these endemic problems occur?

An important root cause is that most software quality companies are "one-trick-ponies" that only care about one subject. Some quality companies sell automated testing, some sell static analysis, some sell automated proofs, or whatever; no company sells a full suite of software quality tools and methods that encompass all sources of software defects and all forms of software defect removal.

Effective quality control needs a synergistic combination of defect prevention, pre-test defect removal, and formal testing with certified test personnel. Worse, most quality companies have zero empirical data as to the efficacy of their tools and methods. They make vast claims of better quality but provide no case studies or validated results.

Software quality data should be based on function point metrics because function points can show defects from all sources such as requirements, design, etc. Table 1 shows approximate U.S. averages for defect potentials for poor quality and high quality software projects.

If drugs and pharmaceutical products were released to the public with as little validation as software quality tools, the U.S. death rate would probably be twice what it actually is today.

Table 1 provides a list of 56 software topics that include defect origins, defect prevention methods, and defect removal stages that run from early requirements to post-delivery for a large system of a nominal 10,000 function points and 1,500 SNAP points in size in Java.

Note: all data in this article come from studies by the author and colleagues, which are based on about 26,000 projects from over 750 companies and 45 government organizations between 1984 and 2016. Additional facts and data were published in the author's 19 books and over 100 journal articles.

Table 1: Results of Poor Quality and High Quality Software

(Nominal 10,000 function points; 1,500 SNAP points)

		Poor Quality	High Quality
	U.S. Software Defect Potentials per Function Point 2016		
1	Requirements defects (functional and non-functional)	0.90	0.30
2	Architecture defects	0.20	0.05
3	Design defects	1.10	0.35
4	Code defects	1.55	0.60

5	Security flaw defects	0.35	0.10
6	Document defects	0.45	0.20
7	Bad fix defects (new bugs in bug repairs)	0.55	0.05
	Total	5.10	1.65
	Application Defect Removal Efficiency Results		
8	Defect removal efficiency (DRE)	91.00%	98.50%
9	Defects removed per function point	4.64	1.63
10	Defects removed - actual total	46,410	16,253
11	Defects delivered per function point	0.46	0.02
12	Defects delivered - actual total	4,590	247
13	High severity defects delivered per function point	0.46	0.02
14	High severity defects delivered - actual total	689	27
15	Security flaws delivered - actual total	55	1
		Poor Quality	**High Quality**
	Application Defect Prevention Stages		
16	Joint application design (JAD)	No	Yes
17	Prototype	Yes	Yes
18	Requirements models (primarily functional)	No	Yes
19	Quality function deployment (QFD)	No	Yes
20	Automated proofs	No	Yes
21	SEMAT (Software Engineering Methods and Theory)	No	Yes
22	Six-Sigma for software	No	Yes
23	Capability Maturity Model Integration (CMMI) - defense only	No	No
	Total	1	7
		Poor Quality	**High Quality**
	Application Pre-Test Defect Removal Stages		
24	Formal inspections of requirements	No	Yes
25	Formal inspection of architecture (large systems)	No	Yes
26	Formal inspections of design	No	Yes
27	Formal inspections of new/changed code	No	Yes
28	Formal quality assurance reviews	No	Yes
29	Pair programming (*not recommended*)	No	No
30	Independent verification & validation (defense only)	No	No
31	FOG readability index of requirements, design	No	Yes
32	Static analysis of application code and changed code	No	Yes
33	Ethical hackers on high-security software	No	Yes
34	Cyclomatic complexity analysis and reduction	No	Yes
35	SANS Institute defect category analysis and removal	No	Yes
	Total	1	10
		Poor Quality	**High Quality**
	Application Test Defect Removal Stages		
36	Unit test - automated	Yes	Yes
37	New function test	Yes	Yes
38	Regression test - automated	Yes	Yes

39	Stress/performance test	Yes	Yes
40	Usability test	No	Yes
41	Component test	Yes	Yes
42	Independent test (defense only)	No	No
43	Security test	No	Yes
44	System test - automated	Yes	Yes
45	Multi-platform test	Yes	Yes
46	Global nationalization test	Yes	Yes
47	Beta/acceptance test	Yes	Yes
	Total	9	11
		Poor Quality	**High Quality**
	Application Post-Release Quality Stages		
48	Static analysis of all code changes/bug repairs	No	Yes
49	Formal inspection of large changes	No	Yes
50	Cyber-attack defenses (firewalls, antivirus, etc.)	Yes	Yes
51	Penetration teams (high security applications)	No	Yes
52	Maintainability analysis of legacy applications	No	Yes
53	Test library analysis (defective test case removal)	No	Yes
54	Error-prone module (EPM) analysis and removal	No	Yes
55	Race-condition analysis and correction	No	Yes
56	Cyclomatic complexity analysis and correction	No	Yes
	Total	1	9
56	**TOTAL QUALITY CONTROL FACTORS**	**19**	**53**

All of these quality control factors are important for large systems in the 10,000 function-point size range. The problem today is that no known software quality company sells more than one or two of these 56 quality methods or even knows about the others!

Few quality companies and even fewer of their clients know about the other factors! A narrow focus on testing and basic ignorance of the suite of effective software defect prevention and defect removal methods is an endemic and chronic problem in the software industry.

If a reader wanted to learn about all 56 quality factors, he or she would probably need a dozen courses from at least half a dozen software quality training companies because none of the courses cover the full spectrum of effective quality tools and methods.

None of the major international standards such as ISO and IEEE standards on software quality are fully adequate because none of them ask for quantitative quality data and all ignore basic software quality factors such as defect potentials and defect removal efficiency (DRE).

Software Quality Education Curricula

The curricula of the major software quality training companies are embarrassing because of the gaps, omissions, and topics that are not covered. Even worse, not a single quality education company has actual quantified data on software defect origins, defect densities, defect prevention, or defect removal

efficiency (DRE) levels. You would have to go back almost 200 years in medical education to find such skimpy knowledge of the basic topics needed to train physicians as we have for training software quality and test personnel in 2016.

You might take quality courses from companies such as Construx, from CAST, from IBM, from ITMPI, from SQE, from QAI, from the SANS Institute, from Parasoft, from Smart Bear, and probably from other local educators, but these would likely be single-topic courses such as static analysis or automated testing. The courses, while useful by themselves, would not be part of a full software quality curriculum because none of the quality companies know enough about software quality to have effective overall curricula!

Worse, these courses even from major quality companies would lack quantitative data on defect potentials, defect removal efficiency (DRE), bad-fix injections, error-prone modules or any other of the critical topics that quality professionals should know about. The software industry is running blind due to a widespread lack of quantitative quality data.

Software quality data is available from some benchmark organizations such as Davids Consulting, Gartner Group, Namcook Analytics LLC, TIMetricas, Q/P Management Group/ QSM, and several others. But the combined set of clients for all current quality benchmark organizations is less than 50,000 customers in an industry employing close to 20,000,000 people on a global basis.

Software quality data can be predicted by some parametric software estimation tools such as Software Risk Master (SRM), KnowledgePlan, SEER, SLIM, and COCOMO, but the combined market for all of these parametric tools is less than 25,000 customers in an industry employing almost 20,000,000 people on a global basis.

In other words, even companies that offer accurate quality data have comparatively few clients who are interested in that data, even though it could save companies and governments billions of dollars in reduced defect repairs and reduced cyber-attack recovery costs!

It is professionally embarrassing about how unsophisticated software quality education is compared to medical school curricula for training physicians.

You probably could not take courses on this set of 56 topics from any university because their curricula tend to deal only with a few of the more common methods and concentrate primarily on testing. I have yet to see a university with quantitative data on software defect volumes, severity levels, origins, or effective defect removal methods with quantitative results.

You might take some courses from non-profit associations such as the American Society for Quality (ASQ), the Society for Information Management (SIM) or the Project Management Institute (PMI). But no single organization in 2016 covers more than a small fraction of the total intellectual content of effective software quality control. None of the software non-profit organizations have quantitative data

on defect volumes, severity levels, or defect removal efficiency. To illustrate the kind of quality education that is needed, Table 2 shows a sample curriculum for software quality assurance testing:

Table 2: Software Quality Assurance Curricula Circa 2016

	Software Quality Assurance Courses	Days	Value
1	Hazardous quality metrics: cost per defect	0.50	10.00
2	Hazardous quality metrics: lines of code	0.50	10.00
3	Hazardous quality metrics: technical debt	0.50	10.00
4	Hazardous quality metrics: story points	0.50	10.00
5	Effective quality metrics: function points	0.50	10.00
6	Effective quality metrics: defect removal %	0.50	10.00
7	Effective quality metrics: defect severity levels	0.50	10.00
8	Effective quality metrics: defect origin analysis	0.50	10.00
9	Emerging quality metrics: SNAP points	0.50	10.00
10	Overview of Major Software Failures	1.00	10.00
11	Overview of Major Software Cyber-Attacks	1.00	10.00
12	Error Prone Module (EPM) Analysis	1.00	10.00
13	Software Defect Detection Efficiency (DDE)	1.00	10.00
14	Software Defect Removal Efficiency (DRE)	1.00	10.00
15	Software Defect Tracking	1.00	10.00
16	Software Defect Prevention (JAD, QFD, etc.)	1.00	10.00
17	Software Pre-Test Defect Removal	1.00	10.00
18	Software Test Defect Removal	1.00	10.00
19	Software Requirements Modeling	1.00	10.00
20	Functional and non-Functional Requirements	2.00	10.00
21	Software Static Analysis: text	1.00	10.00
22	Software Static Analysis: code	1.00	10.00
23	Software correctness proofs: manual	1.00	10.00
24	Software correctness proofs: automated	1.00	10.00
25	Software Security and Quality in 2016	2.00	10.00
26	Quality Benchmarks: Namcook, Q/P, etc.	1.00	10.00
27	Software Security Inspections	3.00	10.00
28	Security flaw removal (hacking, test, etc.)	3.00	10.00
29	Error Prone Module (EPM) Analysis	2.00	9.95
30	Software Test Case Design	2.00	9.75
31	Software Test Library Management	1.00	9.75
32	Reducing Bad-Fix Injections	1.00	9.75
33	Test Case Conflicts and Errors	1.00	9.75
34	Software Requirement Inspections	1.00	9.75
35	Software Design Inspections	2.00	9.50
36	Software Code Inspections	2.00	9.50
37	Software Test Inspections	2.00	9.50
38	Defect removal using Pair Programming	1.00	9.50
39	Defect removal using Container Development	1.00	9.50
40	Defect removal using DevOps	2.00	9.50
41	Defect removal using TSP/PSP	2.00	9.00
42	Defect removal using Agile	2.00	9.00
43	Defect removal using RUP	2.00	9.00

44	Automated software testing	2.00	9.00
45	Quality Assurance of Software Reuse	1.00	9.00
46	Quality Assurance of COTS and ERP	1.00	9.00
47	Quality Assurance of Open Source	1.00	9.00
48	Tools: Quality Assurance	1.00	9.00
49	Tools: Defect Prediction	1.00	9.00
50	Defect removal using Waterfall Development	1.00	8.00
51	Cost of Quality (COQ)	1.00	8.00
52	Overview of the CMMI	1.00	8.00
53	ISO and IEEE Quality Standards	1.00	7.00
54	Six Sigma: Green Belt	3.00	7.00
55	Six Sigma: Black Belt	3.00	7.00
	Total	**70.50**	**9.49**

In today's world, software quality assurance has an expanding role in cyber defenses and cyber-attack recovery. Software quality assurance personnel need much more knowledge on security topics in 2016 than they did 30 years ago in 1986.

Table 3: Software Testing Topics Circa 2016

	Software Topics Courses	**Days**	**Value**
1	Test Case Design Optimization	2.00	10.00
2	Test Cases - Design of Experiments	2.00	10.00
3	Test Cases - Cause/Effect Graphing	2.00	10.00
4	Test cases and requirements	2.00	10.00
5	Risk-based test case design	2.00	10.00
6	Analysis of gaps and errors in test case designs	2.00	10.00
7	Cyclomatic complexity and test coverage	2.00	10.00
8	Test Library Control	2.00	10.00
9	Security Testing Overview	2.00	10.00
10	Advanced Security Testing	3.00	10.00
11	Test Schedule Estimating	1.00	10.00
12	Software Defect Potential Estimating	1.00	10.00
13	Defect Removal Efficiency (DRE) Measurement	1.00	10.00
14	Software Build Planning and Control	1.00	10.00
15	Big Data Test Design	2.00	10.00
16	Cloud Test Design	2.00	10.00
17	Removal of Incorrect Test Cases	1.00	10.00
18	Test Coverage Analysis	1.00	9.50
19	Identifying Error-Prone Modules (EPM)	2.00	9.50
20	Data Base Test Design	1.00	9.50
21	Test Case Conflicts and Errors	1.00	9.25
22	Static analysis and testing	1.00	9.00
23	Reducing Bad-Fix Injections	1.00	9.00
24	Basic black box testing	1.00	9.00
25	Basic white box testing	1.00	9.00
26	Basic gray box testing	1.00	9.00
27	Fundamentals or Risk-Based Testing	1.00	9.00
28	Fundamentals of Unit Testing	1.00	9.00

	Software Topics Courses	Days	Value
29	Fundamentals of Regression Testing	1.00	9.00
30	Fundamentals of Component Testing	1.00	9.00
31	Fundamentals of Stress Testing	1.00	9.00
32	Fundamentals of Virus Testing	2.00	9.00
33	Fundamentals of Lab Testing	1.00	9.00
34	Fundamentals of System Testing	2.00	9.00
35	Fundamentals of External Beta Testing	1.00	9.00
36	Fundamentals of Acceptance Testing	1.00	9.00
37	Testing Web Applications	1.00	9.00
38	Tools: Automated Testing	2.00	9.00
39	Tools: Test Case Design	1.00	9.00
40	Tools: Test Library Control	1.00	9.00
41	Tools: Defect Tracking	1.00	9.00
42	Tools: Complexity Analysis	0.50	9.00
43	Tools: Test Coverage Analysis	0.50	9.00
44	Fundamentals of reusable test materials	1.00	9.00
45	Testing Cloud, SOA, and SaaS	2.00	8.80
46	Testing COTS Application Packages	1.00	8.75
47	Testing ERP Applications	1.00	8.75
48	Testing Reusable Functions	1.00	8.75
49	Supply Chain Testing	1.00	8.50
50	Function Points for Test Measures	1.00	7.00
	TOTAL	**67.00**	**9.31**

Software testing has become a barrier to cyber-attacks, so special attention is needed for testing software security flaws. Table 3 lists topics that should be addressed in software testing courses.

Between software quality assurance training and software test personnel training there is a need to expand on both university curricula and the limited curricula from software quality companies, neither of which are fully adequate as of 2016.

If you wanted to acquire actual supporting tools for these 56 quality topics you would probably need to go to at least 15 commercial quality companies, static analysis companies, and test tool companies and another half dozen open source quality groups.

Nobody in 2016 sells all the tools that are needed to control software quality! Most quality tool vendors don't even know about effective quality tools other than the ones they sell. There are no software quality companies in 2016 that have the depth and breadth of medical companies such as McKesson or Johnson & Johnson.

The static analysis companies only sell static analysis; the testing companies only sell test tools; to get quality metrics and measurement tools you need additional vendors; to get ordinary defect tracking tools you need still other vendors; to get quality benchmark data you need another set of vendors; to get software quality predictions via commercial estimating tools you need yet another set of vendors.

The software industry has nothing like a Mayo Clinic where advanced medical treatments are available for a wide spectrum of medical conditions. In fact software quality control in 2016 is closer to the days 200 years ago when doctors were also barbers and sterile surgical procedures had not yet been discovered. Software has nothing even close to CAT scans and MRI exams for finding quality and security problems.

No known software company in 2016 covers the full spectrum of software quality tools, technologies, topics, and effective quality methods, although a few large companies such as IBM, Microsoft, and Hewlett Packard may sell perhaps 12 to 15 out of the set of 56.

No university in 2016 has a truly effective software quality curriculum. In fact many universities still teach courses using "cost per defect" and "lines of code" metrics and hence have no accurate quality data available at all, since these metrics distort reality.

Of course no pharmaceutical company sells medicines for all diseases and no physicians can treat all medical conditions; however, physicians at least learn about almost all common medical conditions as a basic part of their education. There are also specialists available who can deal with uncommon medical conditions.

Medicine has the Index Medicus that provides an overall description of the use of thousands of prescription drugs, their side effects, and dosage. There is no exact equivalent to the Index Medicus for software bugs and their treatment, but the closest is probably Capers Jones' and Olivier Bonsignour's book on The Economics of Software Quality, published in 2012.

Medicine also has many wide-ranging books such as Control of Communicable Diseases in Man published by the U.S. Surgeon General's office which show the scope of common infectious diseases such as polio and smallpox as well as their known treatments. Software has nothing like the breadth and depth of the medical literature.

A book recommended by the author to all clients and colleagues is The Social Transformation of American Medicine by Paul Starr. This book won a Pulitzer Prize in 1982. It also won the Bancroft Prize. This book provides an excellent guide concerning how medicine was transformed from a poorly educated craft into one of the top learned professions in world history.

Surprisingly, at one time about 150 years ago, medicine was even more chaotic than software is today. Medical schools did not require college degrees or even high-school graduation to enter. Medical students never entered hospitals during training because the hospitals used private medical staff. There was no monitoring of medical malpractice and quacks could become physicians. There were no medical licenses or board certifications.

There was no formal evaluation of prescription drugs before release and harmful substances such as opium could be freely prescribed. (A Sears-Roebuck catalog in the 1890's offered liquid opium as a balm for quieting noisy children. This product was available without prescription.)

Paul Starr's excellent book shows how the American Medical Association (AMA) transformed itself and also medical practice to improve medical education and introduce medical licenses and board certifications.

This book by Paul Starr provides a full guide for the set of steps needed by the software industry in order to become a true profession.

One of the interesting methods used by the AMA was reciprocal membership with all state medical societies. This had the effect of raising AMA membership from below 800 to more than 80,000 which finally gave physicians enough political clout to lobby for medical licenses. It would be interesting to see the impact if the Institute of Electrical and Electronic Engineers (IEEE), Society for Information Management (SIM), Association for Computing Machinery (ACM), and other software professional organizations also had reciprocal memberships instead of more or less competing with one another.

Poor software quality is a sociological problem as well as a technology problem. Starr's book showed how medicine gradually improved both the sociology of medical practice and the underlying technology of medical offices and hospitals over about a 50-year period.

With Starr's book as a guide, software engineering might be able to accomplish the same results in less than 25 years instead of the 50 years required to professionalize medicine.

The three major problems facing the software industry in 2016 are these:

1. Software has poor quality control due to lack of knowledge of effective software quality techniques.
2. Software has embarrassingly poor education on software quality due to lack of empirical data.
3. Software has embarrassingly bad and incomplete quality data due to use of ineffective and hazardous metrics such as "cost per defect," combined with the failure to use effective metrics such as function points and defect removal efficiency (DRE).

The sad thing about poor software quality is that all three of these problems are treatable conditions that could be eliminated in less than 10 years if one or more major software companies became proactive in 1) Effective quality metrics and measures, 2) Fact-based education with quantitative data, and 3) Expanded quality control that encompassed effective quality measures, effective defect prevention, effective pre-test defect removal, and effective formal testing.

The author's hope is that vaccinations and effective treatments for poor software quality will be developed soon such as vaccinations for smallpox and polio were developed, and anti-biotics were developed for many bacterial infections. There are still untreatable medical conditions, but overall medicine has made huge advances in prevention and control of hundreds of formerly serious and common diseases. Software has not yet made any major advances in quality control although some modern methods such as static analysis hold promise.

Author's end note: The author's first job after college was editing a medical journal and medical papers for the Office of the Surgeon General of the U.S. Public Health Service. The author's first programming job was also for the USPHS working on medical applications for the National Institutes of Health (NIH).

As an editor of many medical papers and then an author and editor for many software papers, the differences between medical writing and software writing are striking. In medical papers about a third of the text is devoted to descriptions of the measurement methods used to collect the data that form the main opinions of the articles and lead to the key conclusions.

In software papers, including refereed papers for major journals, there is barely a mention of measurement methods. Worse, many software articles use flawed metrics such as cost per defect and lines of code (LOC) without any apparent awareness that these metrics distort reality. Some software articles that use LOC metrics do not even state whether physical lines or logical statements were used by the author in spite of the 500% variances between the two!

The reason that the author has developed such a large collection of quantitative software data is due to my early experience with medical reports and medical writing, where quantitative results are the norm and not a rare exception as they are with the software literature.

The author's first book on software quality in 1993, Software Quality: Analysis and Guidelines for Success, followed the same format as the medical book, Control of Communicable Diseases in Man, and the format worked quite well.

The author's most recent book in 2014 on The Technical and Social History of Software Engineering, was inspired by Paul Starr's book on The Social Transformation of American Medicine, although it did not use quite the same format. For one thing, software has not yet been transformed from a craft into a true profession while Starr's book was written after the transformation of medicine from a craft into a top profession.

There is much the software industry could learn from the structure and nature of medical education and the medical literature which is far more professional than the software literature and software education. Software could also learn from sociological medical factors such as reciprocal memberships among professional associations, and the need for licenses and board certifications for software specialists.

Above all, software should learn that progress depends upon accurate quantified information and not upon unproven claims and questionable assertions. It has been over 100 years since a new medicine could be released to physicians without testing and validation. New software methods and new programming languages come out almost every month without any validation at all.

List of Acronyms Used in this Paper

Acronym	Explanation
ACM	Association for Computing Machinery
ACR	Applied Computer Research
AMA	American Medical Association
ASQ	American Society for Quality
CAST	CAST is a multinational technology corporation, with headquarters in France and New York City.
COCOMO	Constructive Cost Model
Construx	Construx is a company in Bellevue WA that provides training and consulting services concerning software development.
CMMI	Capability Maturity Model Integration
COQ	Cost of Quality
COTS	Commercial Off-The-Shelf
DRE	Defect Removal Efficiency
EPM	Error Prone Module
ERP	Enterprise Resource Planning
IBM	International Business Machines
IEEE	Institute of Electrical and Electronic Engineers
ISBSG	International Software Benchmarking Standards Group
ISO	International Standards Organization
IT	Information Technology
ITMPI	Information Technology Metrics and Productivity Institute
JAD	Joint Application Design
JAVA	Java is a programming language and computing platform first released by Sun Microsystems in 1995.
LOC	Lines of Code

NIH	National Institutes of Health
Parasoft	Parasoft is a company headquartered in Monrovia CA that researches and develops software solutions.
PMI	Project Management Institute
PSP	Personal Software Process
QAI	Quality Assurance Institute
QFD	Quality Function Deployment
Q/P	Q/P Management Group, Inc., located in Stoneham MA, is a provider of software measurement, benchmarking, quality and productivity consulting services.
QSM	Quantitative Software Management is a company located in McLean VA that provides software measurement and estimation services.
RUP	Rational Unified Process
SaaS	Software as a Service
SANS	System Administration, Networking, and Security Institute
SEER	A tool utilized for software estimation
SEMAT	Software Engineering Methods and Theory
SIM	Society for Information Management
SLIM	A systems and software project estimation tool
Smart Bear	A company headquartered in Somerville MA that provides software testing tools.
SNAP Points	**Software sizing** is an activity in software engineering that is used to estimate the **size** of a software. The non-functional **size** is measured by **SNAP Points.**
SOA	Service Oriented Architecture
SPR	Software Productivity Research
SRM	Software Risk Manager
SQE	Software Quality Engineering
TSP	Team Software Process
USPHS	United States Public Health Service

References and Readings in Software Quality Control

Abrain, Alain. *Software Estimating Models*. Wiley-IEEE Computer Society, 2015.

Abrain, Alain. *Software Metrics and Metrology*. Wiley-IEEE Computer Society, 2010.

Abrain, Alain. *Software Maintenance Management: Evolution and Continuous Improvement*. Wiley-IEEE Computer Society, 2008.

Abrain, A. and P. N. Robillard. "Function Point Analysis: An Empirical Study of its Measurement Processes." *IEEE Transactions on Software Engineering*, Vol. 22, No. 12, December 1996, pp. 895-909.

Albrecht, Allan. *AD/M Productivity Measurement and Estimate Validation*. Purchase, NY: IBM Corporation, May 1984.

Austin, Robert D. *Measuring and Managing Performance in Organizations*. New York, NY: Dorset House Press, 1996, 216 pages. ISBN 0-932633-36-6.

Beck, Kent. *Test-Driven Development*. Boston, MA: Addison Wesley, 2002, 240 pages. ISBN 10: 0321146530.

Black, Rex. *Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing*. Wiley, 2009, 672 pages. ISBN-10 0470404159.

Boehm, Barry. *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice Hall, 1981, 900 pages. ISBN-10 0138221227.

Brooks, Fred. *The Mythical Man-Month*. Reading, MA: Addison-Wesley, 1974, revised in 1995.

Chelf, Ben and Raoul Jetley. "*Diagnosing Medical Device Software Defects Using Static Analysis.*" San Francisco, CA: Coverity Technical Report, 2008.

Chess, Brian and Jacob West. *Secure Programming with Static Analysis*. Boston, MA: Addison Wesley, 2007, 624 pages. ISBN 13: 978-0321424778.

Cohen, Lou. *Quality Function Deployment: How to Make QFD Work for You*. Upper Saddle River, NJ: Prentice Hall, 1995, 368 pages. ISBN 10: 0201633302.

Crosby, Philip B. *Quality is Free*. New York, NY: New American Library, Mentor Books, 1979, 270 pages. ISBN-10 0070145121.

Charette, Bob. *Software Engineering Risk Analysis and Management*. New York, NY: McGraw Hill, 1989.

Charette, Bob. *Application Strategies for Risk Management*. New York, NY: McGraw Hill, 1990.

Constantine, Larry L. *Beyond Chaos: The Expert Edge in Managing Software Development*. ACM Press, 2001, ISBN-10 0201719606.

DeMarco, Tom. *Peopleware: Productive Projects and Teams*. New York, NY: Dorset House, 1999, 245 pages. ISBN 10: 0932633439.

DeMarco, Tom. *Controlling Software Projects*. New York, NY: Yourdon Press, 1982, 284 pages. ISBN 0-917072-32-4.

Everett, Gerald D. and Raymond McLeod. Software Testing. Hoboken, NJ: John Wiley & Sons, 2007. ISBN 978-0-471-79371-7, 261 pages.

Ewusi-Mensah, Kweku. Software Development Failures. Cambridge, MA: Massachusetts Institute of Technology Press, 2003, ISBN 0-26205072-2, 276 pages.

Flowers, Stephen. Software Failures: Management Failures, Amazing Stories, and Cautionary Tales; John Wiley & Sons; 1996, ISBN-10 0471951137.

Gack, Gary. Managing the Black Hole: The Executives Guide to Software Project Risk. Thomson, GA: Business Expert Publishing, 2010, ISBN10: 1-935602-01-9.

Gack, Gary. *Applying Six Sigma to Software Implementation Projects* http://software.isixsigma.com/library/content/c040915b.asp.

Galorath, Dan and Michael Evans. Software Sizing, Estimation, and Risk Management: When Performance is Measured, Performance Improves. Philadelphia, PA: Auerbach, 2006.

Garmus, David and David Herron. Function Point Analysis – Measurement Practices for Successful Software Projects. Boston, MA: Addison Wesley Longman, 2001, ISBN 0-201-69944-3, 363 pages.

Gibbs, T. Wayt. “Trends in Computing: Software’s Chronic Crisis”. *Scientific American Magazine*, 271(3), International edition, pp 72-81, September 1994.

Gilb, Tom and Dorothy Graham. Software Inspections. Reading, MA: Addison Wesley, 1993, ISBN 10: 0201631814.

Harris, Michael D.S., David Herron, and Stasia Iwanacki. The Business Value of IT: Managing Risks, Optimizing Performance and Measuring Results. Boca Raton, FL: CRC Press, Auerbach Publications; ISBN-10 1420064748, 2009.

Hill, Peter R. Practical Software Project Estimation: A Toolkit for Estimating Software Development Effort & Duration. New York: McGraw Hill, 2010, ISBN -10 0071717010.

Hill, Peter, Capers Jones, and Don Reifer. The Impact of Software Size on Productivity. Melbourne, Australia: International Software Benchmarking Standards Group (ISBSG), September 2013.

Howard, Alan (Ed.). Software Metrics and Project Management Tools. Phoenix, AZ: Applied Computer Research (ACR), 1997, 30 pages.

Humphrey, Watts. Managing the Software Process. Reading, MA: Addison Wesley, 1989.

International Function Point Users Group (IFPUG). IT Measurement – Practical Advice from the Experts. Boston, MA: Addison Wesley Longman, 2002, ISBN 0-201-74158-X; 759 pages.

Jacobsen, Ivar, Martin Griss, and Patrick Jonsson. Software Reuse - Architecture, Process, and Organization for Business Success. Reading, MA: Addison Wesley Longman, ISBN 0-201-92476-5, 1997, 500 pages.

Jacobsen, Ivar et al. The Essence of Software Engineering; Applying the SEMAT Kernel. Boston, MA: Addison Wesley Professional, 2013.

Jones, Capers. Software Risk Master (SRM) Tutorial. Narragansett RI: Namcook Analytics LLC, 2015.

Jones, Capers. Software Defect Origins and Removal Methods. Narragansett RI: Namcook Analytics LLC, 2015.

Jones, Capers. The Mess of Software Metrics. Narragansett RI: Namcook Analytics LLC, 2015.

Jones, Capers. The Technical and Social History of Software Engineering. Boston, MA: Addison Wesley, 2014.

Jones, Capers and Olivier Bonsignour. The Economics of Software Quality; Boston, MA: Addison Wesley, 2011, ISBN 978-0-13-258220-9, 587 pages.

Jones, Capers. Software Engineering Best Practices. New York, NY: McGraw Hill, 2010, ISBN 978-0-07-162161-8, 660 pages.

Jones, Capers. Applied Software Measurement. New York, NY: McGraw Hill, 3rd edition 2008, ISBN 978-0-07-150244-3, 662 pages.

Jones, Capers. Critical Problems in Software Measurement. Information Systems Management Group, 1993, ISBN 1-56909-000-9, 195 pages.

Jones, Capers. Software Productivity and Quality Today -- The Worldwide Perspective. Information Systems Management Group, 1993, ISBN -156909-001-7, 200 pages.

Jones, Capers. Assessment and Control of Software Risks. Englewood Cliffs, NJ: Prentice Hall, 1994, ISBN 0-13-741406-4, 711 pages.

Jones, Capers. New Directions in Software Management. Information Systems Management Group, ISBN 1-56909-009-2, 150 pages.

Jones, Capers. Patterns of Software System Failure and Success. Boston, MA: International Thomson Computer Press, December 1995, ISBN 1-850-32804-8, 292 pages.

Jones, Capers. "Sizing Up Software." *Scientific American Magazine*, December 1998:279(6)104-111.

Jones, Capers. Conflict and Litigation between Software Clients and Developers. Narragansett, RI: Software Productivity Research Technical Report, 2007, 65 pages.

Jones, Capers. Software Quality – Analysis and Guidelines for Success. Boston, MA: International Thomson Computer Press, ISBN 1-85032-876-6, 1997, 492 pages.

Jones, Capers. Estimating Software Costs, 2nd edition. New York, NY: McGraw Hill, 2007, ISBN-10 1850328676700, 700 pages.

Jones, Capers. "The Economics of Object-Oriented Software." Burlington, MA, Software Productivity Research: SPR Technical Report, April 1997, 22 pages.

Jones, Capers. "Software Project Management Practices: Failure versus Success." *Crosstalk*, October 2004.

Jones, Capers. "Software Estimating Methods for Large Projects." *Crosstalk*, April 2005.

Kan, Stephen H. Metrics and Models in Software Quality Engineering, 2nd edition. Boston, MA: Addison Wesley Longman, ISBN 0-201-72915-6, 2003, 528 pages.

Land, Susan K; Douglas B Smith, and John W Walz. Practical Support for Lean Six Sigma Software Process Definition: Using IEEE Software Engineering Standards. Wiley-IEEE Computer Society Press, 2008, ISBN 10: 0470170808, 312 pages.

McConnell, Steve. Software Project Survival Guide. Redmond, WA: Microsoft Press, 1997, ISBN-10 1572316217, 288 pages.

Mosley, Daniel J. The Handbook of MIS Application Software Testing. Englewood Cliffs, NJ: Yourdon Press, Prentice Hall, 1993, ISBN 0-13-907007-9, 354 pages.

Nandyal; Raghav. Making Sense of Software Quality Assurance. New Delhi, India: Tata McGraw Hill Publishing, 2007, ISBN 0-07-063378-9, 350 pages.

Pressman, Roger. Software Engineering – A Practitioner's Approach, 6th edition. New York, NY: McGraw Hill, 2005, ISBN 0-07-285318-2.

Radice, Ronald A. High Quality Low Cost Software Inspections. Andover, MA: Paradoxicon Publishing, ISBN 0-9645913-1-6, 2002, 479 pages.

Royce, Walker E. Software Project Management: A Unified Framework. Reading, MA: Addison Wesley Longman, 1998, ISBN 0-201-30958-0.

Starr, Paul. The Social Transformation of American Medicine. New York, NY: Basic Books; Perseus Group, 1982, ISBN 0-465-07834-2. **AUTHOR'S NOTE: This book won a Pulitzer Prize in 1982 and is highly recommended as a guide for improving both professional education and professional status. There is much of value for the software community in this book.**

Strassmann, Paul. Information Payoff. Stamford, CT: Information Economics Press, 1985.

Strassmann, Paul. Governance of Information Management: The Concept of an Information Constitution; 2nd edition (eBook). Stamford, CT: Information Economics Press, 2004.

Strassmann, Paul. Information Productivity. Stamford, CT: Information Economics Press, 1999.

Weinberg, Gerald M. The Psychology of Computer Programming. New York, NY: Van Nostrand Reinhold, 1971; ISBN 0-442-29264-3, 288 pages.

Weinberg, Gerald M. Becoming a Technical Leader. New York, NY: Dorset House, 1986, ISBN 0-932633-02-1, 284 pages.

Weinberg, Gerald M. Quality Software Management - Volume 2 First-Order Measurement. New York, NY: Dorset House Press, ISBN 0-932633-24-2, 1993, 360 pages.

Wiegers, Karl A. Creating a Software Engineering Culture. New York, NY: Dorset House Press, 1996, ISBN 0-932633-33-1, 358 pages.

Wiegers, Karl E. Peer Reviews in Software – A Practical Guide. Boston, MA: Addison Wesley Longman, ISBN 0-201-73485-0, 2002, 232 pages.

Yourdon, Ed. Outsource: Competing in the Global Productivity Race. Upper Saddle River, NJ: Prentice Hall PTR, ISBN 0-13-147571-1, 2005, 251 pages.

Yourdon, Ed. Death March - The Complete Software Developer's Guide to Surviving "Mission Impossible" Projects. Upper Saddle River, NJ: Prentice Hall PTR, ISBN 0-13-748310-4, 1997, 218 pages.

SYSTEMS ENGINEERING NEWS

National Academy of Engineering: The Engineering Challenges of the 21st Century

A diverse committee of experts from around the world, some of the most accomplished engineers and scientists of their generation, proposed 14 challenges. Rather than attempt to include every important goal for engineering, the panel chose opportunities that are both achievable and sustainable to help people and the planet thrive. The panel's conclusions were reviewed by more than 50 subject-matter

experts. The effort received worldwide input from prominent engineers and scientists and the general public. Here are the challenges:

- Make solar energy economical
- Provide energy from fusion
- Develop carbon sequestration methods
- Manage the nitrogen cycle
- Provide access to clean water
- Restore and improve urban infrastructure
- Advance health informatics
- Engineer better medicines
- Reverse-engineer the brain
- Prevent nuclear terror
- Secure cyberspace
- Enhance virtual reality
- Advance personalized learning
- Engineer the tools of scientific discovery

More information

Research at the Stevens School of Systems of Enterprises (SSE)

Stevens Institute of Technology faculty recently gathered in New York City to discuss research priorities and thrusts in the coming years. Activities during the offsite research meeting included faculty members presenting their research focus areas, discussions on research proposals, and an Accenture presentation on global trends and analytics, among others. With the theme, “Understanding Complexities in Modern Systems,” the 2016 / 2017 Research Review, the School’s first, summarizes the core research areas established at the meeting, as well as the current work of our research centers and SSE faculty profiles.

More information

SSE Faculty Shares Insights on Modern Systems

Thoughtful insights on modern systems are shared by SSE faculty in their new blog: Systems Perspectives. In the latest installment, “The Long Road to Fully Autonomous Vehicles,” Professor Yeganeh Hayeri discusses the impact of autonomous vehicle systems on society.

More information

Engineering Smart Factories of the Future

Smart factories of the future will need to be innovative, nimble and smart; constantly changing, and improving on the back of intelligent use of data. By integrating technology and information in real time, traditional factories will turn from cost centers into profitable innovation centers. Cyber-Physical Systems (CPS) will monitor the physical processes within modular structured factories, and a virtual copy of the physical world will be mined for data in real time, enabling decentralized decisions. The increasing availability and use of distributed industrial CPS devices and systems, if aligned with the Internet of Things (IoT) and Internet of Services (IoS), could radically change the nature of manufacturing and provide new opportunities to develop more-effective, finer-grained, and self-configuring automation systems. To achieve this, manufacturers will need to make changes. To realize effective CPS for industrial automation implies the need for engineering tools capable of supporting distributed systems. This is coupled with a major shift in emphasis from traditional monolithic, specialism-based, isolated engineering tools and methods, towards integrated, cloud-based infrastructure based around an IoS and associated data.

More information

Creating a Less Wasteful Economy

The current human footprint exceeds the earth’s bio-capacity by more than 50 per cent. Humankind has exceeded the planetary boundaries for biodiversity loss, climate change and many other vital parameters. Mainstreaming the norms of “circular economy”— a conceptual value-preserving model — can help transit towards a sustainable future where instead of extraction of natural resources, as in most linear industrial models, resources are reintegrated, regenerated and reutilized. Drawing upon principles from approaches such as cradle-to-cradle and biomimicry, the ‘circular economy’ emphasizes eliminating the concept of waste, and the use of renewable energy and systems thinking. Through the design of products consisting of biological materials or recyclable constituents, every component of a product can either be looped into its natural cycle or used as a resource for new products. Today, recycling activities in India are dominated by the informal sector and recycling quotas are not yet fully exploited. Secondary raw material usage in the manufacturing industries remains low (20-30 per cent) and much needs to be done in closing the material loop. A vibrant circular economy will help businesses integrate life cycle thinking and resource optimization and change the way they design, buy, make, sell, and collect their

products; consumers will be more conscious of sustainability impacts and contribute to building a better working world.

More information

---

3 Principles to Guide Designing for Social Change

Complex problems can easily swallow our collective ambition with their magnitude. Historically, they have been tackled through systems thinking approaches by policy makers, economists, and civic organizations. This approach of laws, government initiatives, and global nonprofits has worked for some of the massive social changes in our recent past, such as equal voting rights and the near-eradication of polio. The outcome of this approach is a sweeping wide change, initiated from "above." Effective for huge movements, the success of this approach often is undermined by slow adoption or cultural inappropriateness. That is where human-centered design (HCD) comes in. We have seen a recent focus on approaching the same challenges through the lens of the individual's experience. HCD ensures that the solutions take into account the motivations, needs, and values of the impacted individuals.

More information

IIBA Releases New Multi-Level, Competency-Based Certification

To meet the growing demands of the global marketplace in the field of business analysis, IIBA has launched a new global Certification program. This enhanced, multi-level, competency-based certification program recognizes the BA professional's knowledge and skills, and supports IIBA's lifelong BA career progression. Learn more about how IIBA certification can help you become recognized for your skills, knowledge, and expertise within the business analysis profession.

More Information

Short Comedy Sketch Concerning System Requirements – "The Expert"

View YouTube video.

INCOSE

Revised INCOSE Vision and Mission Statements

The INCOSE Vision and Mission statements were updated in the latter part of 2016, the statements are:

- Vision: A better world through a systems approach.

- Mission: To address complex societal and technical challenges by enabling, promoting, and advancing Systems Engineering and systems approaches.

More information

INCOSE SE Vision 2025

With its 25-year history in the systems engineering universe, INCOSE is able to inspire and guide the direction of systems engineering and envision the future state as this century continues to unfold. Vision 2025 is a publication produced by a team of leaders from industry, academia, and government with the intent that it will be used by people working in many domains who will add their unique perspectives to the role systems engineering plays in serving our world's many complicated demands.

Vision 2025 addresses:

- The Global Context for Systems Engineering
- The Current State of Systems Engineering
- The Future State of Systems Engineering

Following is a summary of the Vision:

- Relevant to a broad range of application domains, well beyond its traditional roots in aerospace and defense, to meet society's growing quest for sustainable system solutions to providing fundamental needs, in the globally competitive environment.
- Applied more widely to assessments of socio-physical systems in support of policy decisions and other forms of remediation.
- Comprehensively integrating multiple market, social and environmental stakeholder demands against "end-to-end" life cycle considerations and long-term risks.
- A key integrating role to support collaboration that spans diverse organizational and regional boundaries, and a broad range of disciplines.
- Supported by a more encompassing foundation of theory and sophisticated model-based methods and tools allowing a better understanding of increasingly complex systems and decisions in the face of uncertainty.
- Enhanced by an educational infrastructure that stresses systems thinking and systems analysis at all learning phases.
- Practiced by a growing cadre of professionals who possess not only technical acumen in their domain of application, but who also have mastery of the next generation of tools and methods necessary for the systems and integration challenges of the times.

More information

INCOSE Working Groups

Technical Operations Working Groups create the resource practitioners need. Discuss, collaborate, and share in person and online across more than 40 Working Groups with a wide diversity of interests. INCOSE Working Groups create products, present panels, develop and review standards. By participating in INCOSE's Working Groups, you will:

- Bring value to other INCOSE stakeholders in your interest area.
- Build expertise and contacts.
- Help develop and review international standards.
- Share information across Working Groups.
- Create products to advance the state, art and practice of systems engineering.

More Information

Empowering Women as Leaders in Systems Engineering (EWLSE)

The vision of this new INCOSE initiative is for men and women to work together as advocates for women as leaders in systems engineering. It was launched at the 2015 INCOSE International Symposium. At present, EWLSE continues its outreach to membership at the INCOSE international and regional events.

More information

INCOSE INSIGHT

INSIGHT is the magazine of INCOSE (International Council on Systems Engineering). It is published four times per year and features informative articles dedicated to advancing the state of practice in systems engineering and to close the gap with the state of the art. INSIGHT is now a Practitioners' Magazine published by Wiley and can be found in Wiley Online Library and in Connect. *You must be logged in through the* INCOSE Member *Login in order to gain access to these Member Only Areas:* Wiley Online Library *or* INCOSE Connect*.*

FEATURED ORGANIZATIONS

IEEE Technology and Engineering Management Society (TEMS)

The Engineering Management Society (EMS) was founded in 1951, becoming the Technology Management Council (TMC) in 2007. In 2015 they transitioned to the Technology and Engineering Management Society.

TEMS encompasses the management sciences and practices required for defining, implementing, and managing engineering and technology. Specific topics of interest include, but are not limited to: technology policy development, assessment, and transfer; research; product design and development; manufacturing operations; innovation and entrepreneurship; program and project management; strategy; education and training; organizational development and human behavior; transitioning to management; and the socioeconomic impact of engineering and technology management.

Please visit the website for more information.

International Association of Engineers (IAENG)

The International Association of Engineers (IAENG) is a non-profit international association for the engineers and the computer scientists. IAENG was founded by a group of engineers and computer scientists in 1968, originally as a private club network for its founding members. Later, with the efforts from its members, IAENG membership became open to all the members in the engineering and computer science community. Nowadays, the IAENG members include research center heads, faculty deans, department heads, professors, research scientists/engineers, experienced software development directors and engineers, and university postgraduate and undergraduate students, etc., from over one hundred different countries.

Their goals are to promote the co-operation between the professionals in various fields of the engineering and to cultivate an environment for the advance and development of the technology. Objectives include:

- Promoting the interactions between the engineers;
- Advancing the application of engineering techniques from the academics to the industry; and
- Facilitating the exchange of information and ideas among the engineers and scientists freely.

More information

SYSTEMS ENGINEERING TOOLS NEWS

INCOSE Tool Integration and Model Lifecycle Management Working Group

The Tools Integration and Interoperability Working Group and the MBSE Model Life cycle Management Activity Team have merged to synchronize the objectives to enable Model Based Systems Engineering in an Integrated Systems Engineering Environment.

The Tool Integration and Model Lifecycle Management Working Group (TIMLM WG) mission is to capture best practices and guidelines for using computer based tools, exchanging data between tools and be able to operate on the data without human intervention. The TIMLM WG provides a forum for discussion and information dissemination on tool integration, process implementation, operational behavior and model management to promote the development, validation and deployment of standards that advance the interoperability of systems engineering toolsets.

Objectives:

- To characterize systems engineering tool integration and interoperability requirements.
- To specify requirements and assess solutions for model life cycle management.
- To foster efficiency of systems engineering process execution through integrated processes, tools and environments including design, manufacturing and sustainment life cycle systems.
- To provide forums for discussion and information dissemination regarding tool integration, process implementation and operational behavior.
- To promote the development, validation and deployment of standards that advance the interoperability of systems engineering toolsets.
- To address model life cycle management concerns and to establish scenarios and best practices that address the concerns of the community.
- To determine efficient modeling structures to use. Structures include the repository, the models, and data within the models.

More information

The COBIT 5/CMMI Practices Pathway Tool

CMMI Institute is pleased to announce a new free tool connecting ISACA's COBIT® 5 framework and CMMI® capability improvement models. The COBIT 5/CMMI Practices Pathway Tool provides business IT practitioners with guidance on how to deliver additional value to stakeholders by strengthening

process designs. Users will be able to utilize organizational resources more effectively, measure performance more accurately, and lower costs through stronger governance.

More information

SYSTEMS ENGINEERING PUBLICATIONS

Systems Engineering Innovation and Design

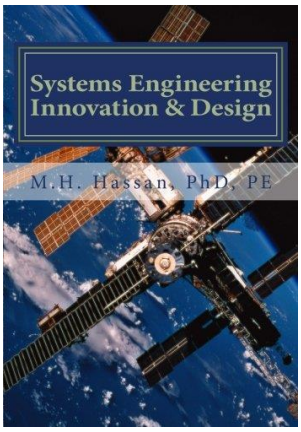

Image source

by M. H. Hassan

Book Description (from the Amazon website):

This book examines innovations and evaluates the components of the systems engineering which best drive the innovation process to develop new systems and products. Disciplined processes, trade studies, risk activities, and others are cross-examined for their value proposition to innovation and design. The book then asks the question as to whether we can do better, including establishing the proper expectations and plans, choosing the right participants, providing the enabling environment, defining the valued outputs, and predicting outcomes. It then presents a vision for tomorrow's systems engineering which involves newer, emerging techniques and design technologies that can lead to higher levels of innovation along with perhaps reduced complexity, both in the engineered systems and the systems engineering process itself. The significant features of this text include: Solid coverage of the fundamental concepts and theory of systems engineering and model-based systems engineering; A strong emphasis on systematic innovation procedures to produce new systems; Application systems engineering tools to realistic problems; Detailed introduction to requirements engineering; Comprehensive coverage to the principles of design; Detailed methodologies to requirement-driven design; Coverage to risk-driven design; Coverage to human-centered design; Detailed review to model-based design; Every chapter concludes with problems for the readers to carry out.

More information

The Seeds of Innovation

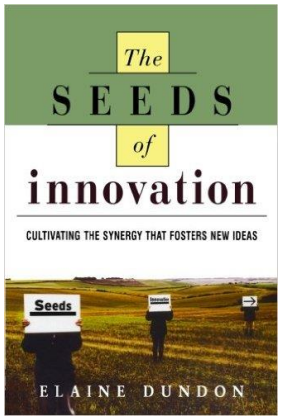

Image source

by Elaine Dundon

Book Description (from the Amazon website):

Breakthrough innovation is a prerequisite for success in almost any organization, yet the actual management of innovation has only recently begun to receive the attention it deserves. Here, innovation thought leader Elaine Dundon offers a ""how-to"" prescription for building creative and strategic innovation skills at all levels of an organization (rather than focusing on decision-making levels only) -- and explains how to produce measurable results that translate directly to the bottom line. Using field-tested concepts and practical examples, and featuring easy-to-apply processes and concrete thinking tools, this straight-talking book provides a broadly applicable guide to innovation -- one that's not limited to a specific industry sector. Today's most comprehensive, one-stop innovation resource, it describes: * The three necessary components of innovation -- creative, strategic, and transformational thinking * Methods for applying innovative thought to existing products, processes, and business models * 90 great innovations and 90 trends to consider.

More information

Team of Teams

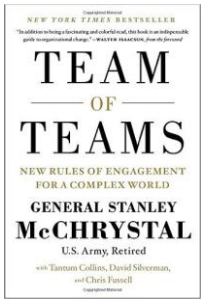

Image source

by General Stanley McChrystal

Book Description (from the Amazon website):

In this powerful book, McChrystal and his colleagues show how the challenges they faced in Iraq can be relevant to countless businesses, nonprofits, and other organizations. The world is changing faster than ever, and the smartest response for those in charge is to give small groups the freedom to experiment while driving everyone to share what they learn across the entire organization. As the authors argue through compelling examples, the team of teams strategy has worked everywhere from hospital emergency rooms to NASA. It has the potential to transform organizations large and small. It's no secret that in any field, small teams have many advantages—they can respond quickly, communicate freely, and make decisions without layers of bureaucracy. But organizations taking on *really* big challenges can't fit in a garage. They need management practices that can scale to thousands of people.

More information

Guideline for Systems Engineering within the Dutch Civil Engineering Sector

Version 3 of the Guideline for Systems Engineering (SE) within the Dutch Civil Engineering Sector is available. The following collaborating parties, who incorporate SE in their contracts and designs, have joined forces to create a national platform that helps improve the dialog between clients and contractors:

- ProRail (the Dutch Rail Infrastructure Manager)
- Rijkswaterstaat (the Directorate-General for Public Works and Water Management)
- Bouwend Nederland (the Dutch construction industry)
- Vereniging van Waterbouwers (the associated Dutch Hydraulic Engineering firms)
- NLingenieurs (the associated Dutch engineering firms)
- Uneto VNI (the Dutch electronic installation industry)

The Systems Engineering methodology is rapidly growing within the Dutch civil engineering sector. Each involved collaborating party is currently translating the guiding principles to match their own specific business environment. The Guideline for Systems Engineering offers a broad collective framework, containing a set of common terms, definitions, and process descriptions.

The first edition of the Guideline for Systems Engineering was released in 2007. At this early stage, the main priority was to develop a common language for SE throughout the Dutch civil engineering sector. This first Guideline was well received and appreciated by all contributing parties. After the publication of the Guideline for Systems Engineering version 2 (in 2009), the support for SE has broadened further, because all parties realized that SE indeed helps to make projects manageable. It created a more thorough understanding of the SE principles and how they improve communication about client requirements. Version 3 of the Guideline (released in November 2013) reflects the experiences gained in

recent years through the application of SE. It outlines the current situation of SE in the civil engineering sector and the challenges for the future. Furthermore, it focuses on the measures that need to be taken at an organizational level for the introduction of SE. Finally, it aims to build a bridge between theory and practice. It's all about cohesion.

Download: The Guideline for Systems Engineering within the Dutch Civil Engineering Sector

---

Systems Engineering Jr. Handbook

The Los Angeles Chapter of INCOSE and The Creativita Institute collaborated to produce the Systems Engineering Jr. Handbook. “Welcome to Systems Engineering!” In this student handbook, the subject of systems engineering is explored with the reader to build a powerful set of concepts with processing tools for future system design, development, and management projects.

Systems engineering is a way of thinking as well as a way of doing. Its roots began with the development of large-scale complex technical systems in the mid-20th century such as the Manhattan project. Since then many “super” projects for the engineering of large-scale and complex technical systems continue to evolve. Many of these projects, e.g., space station and Mars exploration mission, continue to inspire us.

According to a study from the Government Accountability Office (GAO) titled “Assessing the Relationship between Education and the Workforce” published on June 9, 2014, both the number of science, technology, engineering, and mathematics (STEM) degrees awarded and the number of jobs in STEM fields increased in recent years. Many government programs, such as the Networking and Information Technology Research and Development (NITRD) were established to help adopt the education and practice of systems engineering (SE). They all recognized the critical need for the skills of engineering and systems.

To respond to the call, SE is introduced in this student handbook to convey how the concept can apply to a variety of STEM projects. In the second part of the handbook, SE applications on Technology Competitions such as Robotics Contests are addressed.

Download: The Systems Engineering Jr. Handbook

EDUCATION AND ACADEMIA

Worcester Polytechnic Institute (WPI)

Systems Engineering and Leadership Institute (SELI)

The SELI Mission: The advancement of the art, science, practice and research of systems engineering and leadership integrated across the entire engineering spectrum.

WPI believes that all engineers should have some knowledge of systems engineering and leadership methods to be properly prepared for 21st century realities. The strategic goal of the institute is to advance systems engineering education, research, and leadership through partnerships with professionals from government, industry, and academic communities. They seek to:

- Foster crucial needs analysis for today's complex global realities.
- Develop and deploy solutions to best meet those needs.
- Nurture leadership skills necessary to integrate vital systems thinking.

More information

Research Positions Available at the University of Toyko in Formal Methods and Cyber-Physical Systems

Ichiro Hasuo of the Department of Computer Science, The University of Tokyo, and Fuyuki Ishikawa of the National Institute of Informatics are seeking 10+ senior researchers and postdocs to support a new 5.5-year research project (ERATO MMSD, Metamathematics for Systems Design). This broad project aims to extend the realm of formal methods from software to cyber-physical systems (CPS), with particular emphases on logical/categorical metatheories and industrial application (especially in the automotive industry). The project covers diverse areas that include: formal methods, programming languages, software science, software engineering, control theory, machine learning, numerical optimization, user interface, mathematical logic and category theory.

More information

SOME SYSTEMS ENGINEERING-RELEVANT WEBSITES

For more information on Systems Engineering-relevant websites, please proceed to our website.

STANDARDS AND GUIDES

Ford Chief Engineer Nand Kochhar Named Vice Chair of SAE International Technical Standards Board

SAE International announces that Nand Kochhar, Global Safety Systems Chief Engineer for Ford, has been named the Vice Chair of Technical Standards Board (TSB). Kochhar will serve as TSB Vice-Chair beginning January 1, 2017; he will then become TSB Chair on January 1, 2018 and serve a two-year tenure. The SAE International Technical Standards Board is the largest developer of technical standards for Mobility including land, sea, air and space. Industry, governments and the public are served by standards that are used for design, manufacturing, testing, quality control and procurement. It is the duty

of the Technical Standards Board to promote and supervise Technical Committee activities of SAE, including participation in technical committees of other organizations and to approve and issue technical reports including standards, recommended practices, and information reports resulting from its activities.

More information

Object Management Group Chairs Take Standards Work to Next Level at Coronado Membership Meeting

Task Force chairs of the Object Management Group® (OMG®) reported about technology processes underway at OMG's quarterly membership meeting from December 5-9 in Coronado, California. OMG is one of the largest and longest-standing not-for-profits, open-membership consortia that develop and maintain computer industry specifications. Its international community of more than 300 member companies is comprised of end-users, vendors, government agencies, universities and research institutions. Many members are thought leaders and experts in their fields who have collaborated on technology and vertical standards over the past 27 years.

More information

Techstreet Facilitates Communication Concerning Standards

Techstreet is part of Thomson Reuters, a leading source of information for businesses and professionals. Techstreet's IP & Science group provides content and tools to help customers support innovation, protect their intellectual assets, and maximize the value of their intellectual property. Techstreet facilitates communication concerning standards and getting to market faster with better, safer products and services. Techstreet standards and management tools help business, industry, government, and academic organizations achieve national and international compliance, global competitiveness, and speed to market.

More information

National Electrical Manufacturers Association (NEMA)

NEMA is the largest U.S. trade organization representing manufacturers of products used in the generation, transmission, distribution, control, and end-use of electricity. Many of NEMA's 350+ standards have been approved as American National Standards or adopted by the federal government.

More information

SOME DEFINITIONS TO CLOSE ON

Engineering

The word "engineering" is derived from the Latin "ingenium", meaning something like a brilliant idea or flash of genius. It was created in the 16th century and originally described a profession that we would probably call an artistic inventor. Engineers apply the knowledge of the mathematical and natural sciences (biological and physical), with judgment and creativity to develop ways to utilize the materials and forces of nature for the benefit of mankind. The subjects are diverse and include names like bioengineering, computer engineering, electrical and electronics engineering, financial engineering, industrial engineering, internet engineering and systems engineering, etc.

Source: www.iaeng.org/about_IAENG.html

Quality

Quality:

"Degree of excellence”

Source: www.merriam-webster.com

"The standard of something as measured against other things of a similar kind; the degree of excellence of something"

Source: https://en.oxforddictionaries.com/definition/quality

"The totality of features and characteristics of a product or service that bear
on its ability to satisfy stated or implied needs. Not to be mistaken for "degree of excellence" or "fitness f or use" which meet only part of the definition.

Source: ISO8402:1995

Comment from Robert Halligan: This definition of quality raises some significant distinctions.

"Degree to which a set of inherent characteristics (3.10.1) of an object (3.6.1) fulfils requirements (3.6.4)”

Source: ISO 9000:2015(en) Quality management systems — Fundamentals and vocabulary

Comment from Robert Halligan: Note that the ISO 9000:2015 definition of requirement "need or expectation that is stated, generally implied or obligatory" is problematic, in that it is inconsistent with OED and usual usage of the term. Needs just exist. A valid requirement come into existence as a result of a decision by a human to satisfy a need. Requirements can be invalid, i.e. inconsistent with need. A stakeholder can have an expectation that a project will fail, or a product will perform poorly. Under the ISO 9000:2015 definition, these are requirements!

"A subjective term for which each person or sector has its own definition. In technical usage, quality can have two meanings: 1. the characteristics of a product or service that bear on its ability to satisfy stated or implied needs; 2. a product or service free of deficiencies. According to Joseph Juran, quality means "fitness for use;" according to Philip Crosby, it means "conformance to requirements."

Source: http://asq.org/glossary/q.html

Quality Assurance:

"The planned and systematic activities implemented in a quality system so that quality requirements for a product or service will be fulfilled."

Source: American Society for Quality

"A program for the systematic monitoring and evaluation of the various aspects of a project, service, or facility to ensure that standards of quality are being met."

Source: www.merriam-webster.com

"The maintenance of a desired level of quality in a service or product, especially by means of attention to every stage of the process of delivery or production."

Source: https://en.oxforddictionaries.com/definition/quality_assurance

"Part of quality management (3.3.4) focused on providing confidence that quality requirements (3.6.5) will be fulfilled."

Source: ISO 9000:2015(en) Quality management systems — Fundamentals and vocabulary

"QA is defined as a procedure or set of procedures intended to ensure that a product or service under development (before work is complete, as opposed to afterwards) meets specified requirements."

Source: www.whatis.techtarget.com/

"In developing products and services, quality assurance is any systematic process of checking to see whether a product or service being developed is meeting specified requirements."

Source: searchsoftwarequality.techtarget.com/definition/quality-assurance

"Often used interchangeably with quality control (QC), QA is a wider concept that covers all policies and systematic activities implemented within a quality system. QA frameworks include (1) determination of adequate technical requirement of inputs and outputs, (2) certification and rating of suppliers, (3) testing of procured material for its conformance to established quality, performance, safety, and reliability standards, (4) proper receipt, storage, and issue of material, (5) audit of the process quality, (6) evaluation of the process to establish required corrective response, and (7) audit of the final output for conformance to (a) technical (b) reliability, (c) maintainability, and (d) performance requirements."

Source: www.businessdictionary.com/definition/quality-assurance-QA.html

Quality assurance (QA) is a way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to customers; which ISO 9000 defines as "part of quality management focused on providing confidence that quality requirements will be fulfilled"

Source: en.wikipedia.org/wiki/Quality assurance

Quality Control:

"Quality Control: The observation techniques and activities used to fulfill requirements for quality."

Source: American Society for Quality

"An aggregate of activities (as design analysis and inspection for defects) designed to ensure adequate quality especially in manufactured products."

Source: www.merriam-webster.com

"A system of maintaining standards in manufactured products by testing a sample of the output against the specification."

Source: https://en.oxforddictionaries.com/definition/quality_assurance

"A part of quality management focused on fulfilling quality requirements."

Source: ISO 9000:2005, Clause 3.2.10

"Part of quality management (3.3.4) focused on fulfilling quality requirements (3.6.5)."

Source: ISO 9000:2015(en) Quality management systems — Fundamentals and vocabulary

"A system for verifying and maintaining a desired level of quality in an existing product or service by careful planning, use of proper equipment, continued inspection, and corrective action as required."

Source: www.dictionary.com

"Quality control (QC) is a procedure or set of procedures intended to ensure that a manufactured product or performed service adheres to a defined set of quality criteria or meets the requirements of the client or customer."

Source: www.whatis.techtarget.com/

"An aspect of the quality assurance process that consists of activities employed in detection and measurement of the variability in the characteristics of output attributable to the production system, and includes corrective responses."

Source: www.businessdictionary.com

Comment from Robert Halligan: Note that under this definition, quality control is a subset of quality assurance.

Quality management:

"Management (3.3.3) with regard to quality (3.6.2)

Source: ISO 9000:2015(en) Quality management systems — Fundamentals and vocabulary

"Quality management includes all the activities that organizations use to direct, control, and coordinate quality. These activities include formulating a quality policy and setting quality objectives. They also include quality planning, quality control, quality assurance, and quality improvement."

Source: http://www.praxiom.com

"Management activities and functions involved in determination of quality policy and its implementation through means such as quality planning and quality assurance (including quality control)."

Source: http://www.businessdictionary.com

CONFERENCES AND MEETINGS

For more information on systems engineering related conferences and meetings, please proceed to our website.

PPI NEWS

PPI Celebrates 25 Years

Join us in celebration of 25 years of service by PPI to the systems engineering community.

For PPI, founded in 1992 by Robert Halligan, 2017 is our Silver Anniversary. Over the 25 years, we have grown from three people to an international company with employees in Australia, Brazil, China, South Africa, the United Kingdom and the United States. We have a wholly-owned subsidiary company in the United States (PPI-USA), and another wholly-owned subsidiary Certification Training International (CTI). CTI delivers mainly CSEP training.

We have delivered project-related training or consulting in 38 different countries, to delegates working in almost every imaginable sector. Our training on systems engineering and its various facets has been delivered so far to over 13,500 professionals worldwide, presently at the rate of over 1150 people a year (5-day courses). To this can be added very many shorter courses.

Alwyn Smit Presents at the INCOSE Netherlands Chapter Event

On Thursday 19 January 2017, Alwyn Smit delivered an evening tutorial to the INCOSE Netherlands Chapter on Getting the Most out of "Work" Breakdown Structure. The tutorial has hosted by the Engineering department of the City of Amsterdam. A group of about 20 engineering professionals attended this presentation.

The tutorial got under way after a short presentation by Dick Terleth, the Chapter President, on the Chapter as well as a short presentation on the activities of the Engineering Department of the City of Amsterdam by Thomas Munster.

What made this tutorial rather unique is that here we had an Australian company delivering a tutorial in English to a Dutch audience by an Afrikaans speaking presenter. Fortunately the Dutch audience did not have too much of a problem with the South African accent and a few statements in Afrikaans were even well understood.

Something that caught my eye during Dick's talk was the Chapter's outreach to young system engineers referred to as "JONG INCOSE" or “YOUNG INCOSE”.

The INCOSE Netherlands is of course also well known for the book it produced on the application of systems engineering in the construction industry - something that led to additional discussion during the social part of the evening when the event was concluded with wine and snacks.

Image source

Alwyn Smit (second from the left), Aysun Sancar Yılmaz (second from the right)

PPI Leader becomes President of INCOSE South Africa

PPI’s Professional Development Manager, Suja Joseph-Malherbe, was elected as President-elect of INCOSE South Africa in 2016. She has taken over as President of the Chapter as of 1 January 2017. Her term of office ends 31 December 2018. INCOSE South Africa was founded in 2002 and is the first African chapter of INCOSE, International Council on Systems Engineering. The International Council on

Systems Engineering (INCOSE) is a not-for-profit membership organization founded in 1990. Its mission is to address complex societal and technical challenges by enabling, promoting, and advancing Systems Engineering and systems approaches.

PPI is delighted with this wonderful news and congratulates Suja on her important new role.

PPI AND CTI EVENTS

Systems Engineering Public 5-Day Courses

Upcoming locations include:

- Canberra, Australia
- Oslo, Norway
- São José dos Campos, Brazil

Requirements Analysis and Specification Writing Public Courses

Upcoming locations include:

- Amsterdam, the Netherlands
- Bremerton, Washington, United States of America
- London, United Kingdom

Systems Engineering Management Public 5-Day Courses

Upcoming locations include:

- Bristol, United Kingdom
- Eindhoven, the Netherlands
- Orlando, Florida, United States of America

Requirements, OCD and CONOPS Public 5-Day Courses

Upcoming locations include:

- Amsterdam, the Netherlands
- Melbourne, Australia
- Pretoria, South Africa

Human Systems Integration Public 5-Day Courses

Upcoming locations include:

- Sydney, Australia

CSEP Preparation 5-Day Courses (Presented by Certification Training International, a PPI company)

Upcoming locations include:

- Denver, Colorado, United States of America
- Orlando, Florida, United States of America
- North Ryde, New South Wales, Australia

PPI UPCOMING PARTICIPATION IN PROFESSIONAL CONFERENCES

PPI will be participating in the following upcoming events.

IEEE International Systems Engineering Conference (SysCon)

24 - 27 April 2017

Montreal, Canada

IISE Annual Conference and Expo

20 - 23 May 2017

Pittsburgh, PA

15th Annual Conference on System Engineering Research (CSER)

23 - 25 March 2017

Redondo Beach, CA

27th Annual INCOSE International Symposium (IS2017)

15 – 20 July 2017

Adelaide, Australia

13th INCOSE SA Conference 2017

11 – 13 October 2017

Pretoria, South Africa

Kind regards from the SyEN team:

Robert Halligan, Editor-in-Chief, email: rhalligan@ppi-int.com

Ralph Young, Editor, email: ryoung@ppi-int.com

Suja Joseph-Malherbe, Managing Editor, email: smalherbe@ppi-int.com

Project Performance International

2 Parkgate Drive, Ringwood North, Vic 3134 Australia Tel: +61 3 9876 7345 Fax: +61 3 9876 2664

Tel Brasil: +55 11 3958 8064

Tel UK: +44 20 3608 6754

Tel USA: +1 888 772 5174

Web: www.ppi-int.com

Email: contact@ppi-int.com

Tell us what you think of SyEN. Email us at syen@ppi-int.info.