

Project Performance International

<http://www.ppi-int.com/newsletter/SyEN-040.php>

SYSTEMS ENGINEERING NEWSLETTER

brought to you by

Project Performance International (PPI)

SyEN 40 – January 19, 2012

Dear Colleague,

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 of systems engineering, 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 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 SE eNewsletters, please reply to the notifying e-mail with "Remove" in the subject line, from the same email address. Your removal will be confirmed, by email.

We hope that you find this newsletter to be informative and useful. Please tell us what you think. Email to: contact@ppi-int.com.

WHAT'S INSIDE:

A Quotation to Open On

[Read More...](#)

Feature Article: Evaluating Ten Software Development Methodologies

[Read More...](#)

Systems Engineering News

- Status of OMG Systems Modeling Language (OMG SysML) V1.3 and V1.4
- Working Group meetings at INCOSE IW2012 (with opportunity to meet at RAMS 2012)
- OMG and Business Process Incubator Announce Relaunch of Enhanced BPMN.org Website
- Get Ready for the Next Wave of Engineers • Should Computer Science Be Required in K-12?
- Paul Croll is Awarded 2012 IEEE Computer Society Hans Karlson Award

[Read More...](#)

Ask Robert Why would you want to measure requirements quality?

[Read More...](#)

Featured Societies – Americas Requirements Engineering Association (AREA)

[Read More...](#)

INCOSE Technical Operations:

South Africa Systems Engineering Training Working Group

[Read More...](#)

Systems Engineering Tools News

- UML/SysML Tool Vendor Model Interchange Test Case Results Now Available

[Read More...](#)

Systems Engineering Books, Reports, Articles, and Papers

- Modern Methods of Systems Engineering
- Enterprise Release Management
- Lean for Systems Engineering with Lean Enablers for Systems Engineering
- INCOSE INSIGHT, December 2011

[Read More...](#)

Conferences and Meetings

[Read More...](#)

Education and Academia

- UC Irvine Extension and Georgia Institute of Technology Announce Articulation Agreement between Systems Engineering Programs
- Stevens Software Systems Engineering Faculty Leads Efforts To Release Newest Version of the Worldwide Graduate Reference Curriculum for Systems Engineering
- Systems Engineering Scholarships Available
- INCOSE Webinars

[Read More...](#)

Some Systems Engineering-Relevant Websites

[Read More...](#)

Standards and Guides

Recent ISO/IEC/SC7 Systems Engineering Relevant Standards

[Read More...](#)

A Definition to Close On – Life Cycle

[Read More...](#)

PPI News

- PPI's Systems Engineering Goldmine (SEG) is Upgraded
- New Cognitive Systems Design Video by Dr. Gavan Lintern
- www.ppi-int.com Goes Mobile

[Read More...](#)

PPI Events

[Read More...](#)

A Quotation to Open On

“If I had asked people what they wanted, they would have said faster horses.”
— Henry Ford

Feature Article

Evaluating Ten Software Development Methodologies

Capers Jones
President, Capers Jones & Associates LLC
Email: [Capers.Jones3 \(@\) Gmail.com](mailto:Capers.Jones3@gmail.com)

Copyright © 2011 by Capers Jones & Associates LLC.
All rights reserved.

Abstract:

As this is written there are about 55 named software development methods in use, and an even larger number of hybrids. Some of the development methods include the traditional waterfall approach, various flavors of agile, the Rational Unified Process (RUP), the Team Software Process (TSP), V-Model development, Microsoft Solutions Framework, the Structured Analysis and Design Technique (SADT), Evolutionary Development (EVO), Extreme Programming (XP), PRINCE2, Merise, and many more. Only 10 methods are evaluated here, since evaluating all 50 would take too many pages.

In general selecting a software development methodology has more in common with joining a cult than it does with making a technical decision. Many companies do not even attempt to evaluate methods, but merely adopt the most popular, which today constitute the many faces of agile.

This article uses several standard metrics including function points, defect removal efficiency (DRE), Cost of Quality (COQ), and Total Cost of Ownership (TCO) to compare a sample of contemporary software development methods.

The data itself comes from studies with a number of clients who collectively use a wide variety of software methods.

INTRODUCTION

The existence of more than 55 software development methods, each with loyal adherents, is a strong message that none of the 55 is capable of handling all sizes and kinds of software applications.

Some methods work best for small applications and small teams; others work well for large systems and large teams; some work well for complex embedded applications; some work well for high-speed web development; some work well for high-security military applications. How is it possible to select the best methodology for specific projects? Is one methodology enough, or should companies utilize several based on the kinds of projects they need to develop?

Unfortunately due to lack of quantified data and comparisons among methodologies, selecting a software development method is more like joining a cult than a technical decision. Many companies do not even attempt to evaluate alternative methods, but merely adopt the most popular method of the day, whether or not it is suitable for the kinds of software they build.

When software methodologies are evaluated the results bring to mind the ancient Buddhist parable of the blind man and the elephant. Different methods have the highest speed, the highest quality, and the lowest total cost of ownership.

Combinations of Factors that Affect Software Projects

An ideal solution would be to evaluate a variety of methods across a variety of sizes and types of software. However that is difficult because of combinatorial complexity. Let us consider the major factors that are known to have an impact on software project results:

Factor	Number of Possibilities
Methodologies	55

Programming languages	50
Nature, class, and type of application	15
Capability Maturity Model Levels	5
High, average, and low team experience	3
Size plateau of application (small, medium, large)	3
Combinations of factors	1,856,250

Since the number of combinations is far too large to consider every one, this article will make simplifying assumptions in order to focus primarily on the methodologies, and not on all of the other factors.

In this article the basic assumptions will be these:

Application size	1000 function points
Programming languages	C and C++
Logical code statements	75,000
Requirements creep	Omitted
Deferred features	Omitted
Reusable features	Omitted
Team experience	Average
Cost per staff month	USD \$7,500

By holding size, languages, and team experience at constant levels it is easier to examine the impacts of the methodologies themselves. There are unfortunately too many methodologies to consider all of them, so a subset of 10 methods will be shown, all of which are fairly widely used in the United States.

Note that the actual applications being compared ranged from about 800 function points in size to 1,300. The author has a proprietary method for mathematically adjusting application sizes to a fixed size in order to facilitate side-by-side comparisons.

Methodologies in alphabetic order:

1. Agile with scrum
2. CMMI 1 with waterfall
3. CMMI 3 with iterative
4. CMMI 5 with spiral
5. Extreme programming (XP)
6. Object-oriented development
7. Pair programming with iterative
8. Proofs of correctness with waterfall
9. Rational unified process (RUP)
10. Team Software Process (TSP). Since not every reader may be familiar with every method, here are short descriptions of the ones in the article:

Agile with scrum: The term “agile” is ambiguous and there are many flavors of agile. For this article the term is used for projects that more or less follow the 1997 agile manifesto, have embedded users to provide requirements, use user stories, divide projects into discrete sprints that are developed individually, and use the scrum concept and daily status meetings. Minimizing paperwork and accelerating development speeds are top goals of agile.

CMMI 1 with waterfall: The Capability Maturity Model Integrated™ (CMMI) of the Software Engineering Institute is a well-known method for evaluating the sophistication of software development. CMMI 1 is the bottom initial level of the 5 CMMI levels and implies fairly chaotic development. The term “waterfall” refers to traditional software practices of sequential development starting with requirements and not doing the next step until the current step is finished.

CMMI 3 with iterative: The third level of the CMMI is called “defined” and refers to a reasonably smooth and well understood set of development steps. The term “iterative” is older than “agile” but has a similar meaning of dividing applications into separate pieces that can be constructed individually.

CMMI 5 with spiral: The 5th level of the CMMI is the top and is called “optimizing.” Groups that achieve this level are quite sophisticated and seldom make serious mistakes. The spiral model of software development was pioneered by Dr. Barry Boehm. It features ideas that also occur in agile, such as individual segments that can be developed separately. The spiral segments are often larger than agile segments, and are preceded by prototypes.

Extreme Programming (XP): This method falls under the umbrella of agile development but has some unique features. The most notable unique feature is to delay coding until test cases have been developed first. The XP method also uses reviews or inspections. Sometimes pair programming is used with XP but not always, so that is a special case. Quality of the final software is a major goal of the XP method.

Object-oriented development (OO): The OO method is one of the oldest in this article and has had many successes. It has also led to the creation of special languages such as Objective C. In this article OO analysis and design with use cases are used. The C++ language is also an OO language. OO analysis and design are somewhat different from conventional methods so a learning curve is needed.

Pair programming: The concept of pair programming is often part of the agile approach, but is not limited to it. In this paper pair programming is used with iterative development. The basic idea of pair programming is that two people take turns coding. While one is coding the other is watching and making suggestions. Sometimes the pair uses only one computer or work station between them.

Proofs of correctness: The concept behind proofs of correctness is that of applying formal mathematical proofs to the algorithms that will be included in a software application. It is obvious that the algorithms need to be expressed in a formal manner so that they can be proved. It is also obvious that the person who performs the proof has enough mathematical skills to handle rather complex equations and algorithms.

Rational Unified Process (RUP): The RUP methodology was originated by the Rational Corporation which was acquired by IBM in 2003 so it is now an IBM methodology. The RUP method includes aspects of both iterative and object-oriented development. Since RUP is now owned by IBM there are numerous tools that support the method. Use Cases and visual representations are standard for RUP applications, but the author’s clients usually include other methods as well such as decision tables.

Team Software Process (TSP): The TSP method was developed by the late Watts Humphrey, who was IBM’s director of programming and later created the assessment method used by the Software Engineering Institute (SEI) capability maturity model. TSP is very focused on software quality. All bugs are recorded; inspections are used, and high quality is the main goal on the grounds that bugs slow down development. The TSP method has some unusual aspects such as self-governing tools and a coach that serves the role of manager. TSP is now endorsed and supported by the SEI.

Three Kinds of Methodology Evaluation and 10 Metrics

Even with the number of methods limited to 10 there are still a great many results that need to be evaluated. However, from working with hundreds of clients, the topics that have the greatest importance to development managers and higher executives are these:

Speed-related metrics:

1. Development schedules
2. Development staffing
3. Development effort
4. Development costs

Quality-related metrics:

5. Defect potentials
6. Defect removal efficiency (DRE)
7. Delivered defects
8. High-severity defects

Economic-related metrics:

9. Total Cost of Ownership (TCO)
10. Cost of Quality (COQ).

Even with only 10 methodologies and 10 topics to display, that is still quite a significant amount of information.

This article will attempt to compare methodologies in three major categories:

1. Speed: Development schedules, effort, and costs
2. Quality: Software quality in terms of delivered defects
3. Economics: Total Cost of Ownership (TCO) and Cost of Quality (COQ).

Note that the technique used in this article of holding application size constant at 1000 function points means that the data cannot be safely used to determine the best methods for large systems of 10,000 function points or massive systems of 100,000 function points. However, applications in the 1000 function point size range are very common, and are large enough to show comparative results in a fairly useful way.

Some of the data in this article was prepared using the author's Software Risk Master (SRM) tool, which is designed to perform side-by-side comparisons of any development methodology, any CMMI level, and any level of team experience. Some of the tables are based on SRM outputs, although derived from earlier measured applications.

Speed: Comparing Methodologies for Development Schedules and Costs

The first comparison of methodologies concerns initial development speeds, costs, and short-term issues. Among the author's clients, the most frequent request when estimating software projects is to predict the development schedule. Because schedules are viewed as critical to a majority of software managers and executives, Table 1 is sorted by the speed of development.

Methodologies	Schedule Month	Staff	Effort Months	FP Month	Development Cost
1 Extreme (XP)	11.78	7	84	11.89	\$630,860
2 Agile/scrum	11.82	7	84	11.85	\$633,043
3 TSP	12.02	7	86	11.64	\$644,070
4 CMMI 5/ spiral	12.45	7	83	12.05	\$622,257
5 OO	12.78	8	107	9.31	\$805,106
6 RUP	13.11	8	101	9.58	\$756,157
7 Pair/iterative	13.15	12	155	9.21	\$1,160,492
8 CMMI 3/iterative	13.34	8	107	9.37	\$800,113
9 Proofs/waterfall	13.71	12	161	6.21	\$1,207,500
10 CMMI 1/waterfall	15.85	10	158	6.51	\$1,188,870
Average	13	8.6	112.6	9.762	\$844,852

Table 1: Software Schedules, Staff, Effort, Productivity

As can be seen, the software development methods that yield the shortest schedules for applications of 1000 function points are the XP and Agile methods, with TSP coming in third.

Quality: Comparing Defect Potentials, Defect Removal, and Delivered Defects

The next topic of interest when comparing methodologies is that of quality. The article considers four aspects of software quality: defect potentials, defect removal efficiency, delivered defects, and high-severity defects.

The phrase “defect potential” refers to the sum of defects found in requirements, design, source code, documents, and “bad fixes.” A bad fix is a new defect accidentally injected during an attempt to repair a previous defect (about 7% of attempts to fix bugs include new bugs).

The phrase “defect removal efficiency” refers to the combined efficiency levels of inspections, static analysis, and testing. In this article, six kinds of testing were included: 1) unit test; 2) function test; 3) regression test; 4) performance test; 5) system test; 6) acceptance test. There are about 40 total kinds of testing, but the specialized forms of testing are outside the scope of this article.

When quality is evaluated, readers can see why the parable of the blind man and the elephant was cited earlier:

Methodologies	Defect Potential	Defect Removal	Defect Delivered	Hi Sev. Defects
1 TSP	2,700	96.79%	87	16
2 CMMI 5/ spiral	3,000	95.95%	122	22
3 RUP	3,900	95.07%	192	36
4 Extreme (XP)	4,500	93.36%	299	55
5 OO	4,950	93.74%	310	57
6 Pair/iterative	4,700	92.93%	332	61
7 Proofs/waterfall	4,650	92.21%	362	67
8 Agile/scrum	4,800	92.30%	370	68
9 CMMI 3/ lter.	4,500	91.18%	397	73
10 CMMI 1/waterfall	6,000	78.76%	1,274	236
Average	4,370	92.23%	374	69

Table 2; Software Defect Potentials, Removal, and Delivery

When the focus of the evaluation turns to quality rather than speed, TSP, CMMI 5, and RUP are on top, followed by XP. Agile is not strong on quality so it is only number 8 out of 10. The Agile lack of quality measures and failure to use inspections will also have an impact in the next comparison.

Economics: Total Cost of Ownership (TCO) and Cost of Quality (COQ)

Some of the newer methods such as Agile and XP have not been in use long enough to show really long-range findings over 10 years or more. In this article, TCO is limited to only five years of usage, because there is almost no data older than that for Agile.

The figures for TCO include development, five years of enhancement, five years of maintenance or defect repairs, and five years

of customer support. While the Software Risk Master tool predicts those values separately, in this article they are all combined together into a single figure.

The figures for COQ consolidate all direct costs for finding and fixing bugs from the start of requirements through five years of customer usage.

Methodologies	TCO	COQ	Percent
1 TSP	\$1,026,660	\$298,699	29.09%
2 CMMI 5/ spiral	\$1,034,300	\$377,880	36.53%
3 Extreme (XP)	\$1,318,539	\$627,106	47.56%
4 RUP	\$1,360,857	\$506,199	37.20%
5 Agile/scrum	\$1,467,957	\$774,142	52.74%
6 OO	\$1,617,839	\$735,388	45.45%
7 CMMI 3/ Iter.	\$1,748,043	\$925,929	52.97%
8 Pair/iterative	\$2,107,861	\$756,467	35.89%
9 Proofs/waterfall	\$2,216,167	\$863,929	38.98%
10 CMMI 1/waterfall	\$3,944,159	\$2,804,224	71.10%
Average	\$1,784,238	\$866,996	44.75%

Table 3; Total Cost of Ownership (TCO): Cost of Quality (COQ)

Because applications developed using the TSP, CMMI 5, and RUP methodologies are deployed with low numbers of defects it is fairly easy to enhance them, maintain them, and support them. Therefore, the 5-year total cost of ownership clearly favors the quality-related methods rather than the speed-related methods.

Agile is not bad, but with a COQ of more than 50%, Agile needs to take quality more seriously up front.

The COQ percentages reveal a chronic problem for software applications. We have so many bugs that finding and fixing bugs is the major cost of both development and total cost of ownership.

The Methods that Achieve Top Rankings in all Categories

To continue with the metaphor of the blind men and the elephant, here are the top methods in each of the 10 categories:

1. Development schedules	Extreme programming (XP)
2. Development staffing	Agile/scrum (tied)
3. Development effort	CMMI/5 spiral
4. Development costs	CMMI/5 spiral
5. Defect potentials	TSP
6. Defect removal efficiency (DRE)	TSP

7. Delivered defects	TSP
8. High-severity defects	TSP
9. Total Cost of Ownership (TCO)	TSP
10. Cost of Quality (COQ)	TSP

Table 4: Top Methods in 10 Categories

The phrase “be careful of what you wish for because you might get it” seems to be appropriate for these methodology comparisons. Methods such as Agile that focus on speed are very quick. Methods such as TSP, RUP, and CMMI 5 that focus on quality have very few defects.

Why Some Methods Compare Poorly for Speed, Quality, and Economics

As can be seen, the various methodologies fluctuated in their effectiveness on the speed, quality, and economic dimensions. However, three methodologies were near the bottom for all three evaluations. These laggards were the waterfall method, which was in last place, the proof of correctness method, and the pair programming method. It is useful to explain the probable reasons for the low placements of these three methodologies.

Waterfall and CMMI 1

It is no secret that about 35% of software projects for more than 50 years have been cancelled due to poor quality or overruns. Most of these used waterfall development and either were at CMMI level 1 or did not use the CMMI at all.

At the 1000 function point size range used in this example for waterfall, the percentage of time and effort devoted to finding and fixing bugs is about 25.71%. The number of projects that run late or exceed their budgets is about 50%. These are not very large applications, but with waterfall they are often troublesome.

It should be mentioned that the primary motivation of most of the newer methods is to overcome the historical problems associated with waterfall development.

There have been a few successes with waterfall projects but these tend to be those done by expert teams.

Pair Programming

Unfortunately, pair programming is an expensive mistake. The idea of letting two people take turns programming while one watched is a theoretical idea but weak in practice. The evidence for pair programming is flawed. There are assertions that pairs create software with fewer bugs than individual programmers. Unfortunately, the individuals were using basic waterfall methods. Capable individual programmers who use static analysis and participate in formal code inspections of their work produce better code for about half the cost of pair programming.

Further, there are some 90 different software occupations. Why double up programmers and nobody else? If the idea of pair programming worked as asserted, then architects, business analysts, testers, quality assurance and everybody might be doubled. Why not use two project managers instead of one?

The usage of pair programming is symptomatic of very poor measurement practices and also a failure to understand the distribution of talent among large populations. If a company were building a large system with 500 programmers, it would not be possible to bring in or hire 500 more to pair up with them.

Proofs of Correctness

The idea of proofs of correctness is an academic construct and is more theoretical than real. In order to prove algorithms in software they need to be formally expressed and the personnel doing the proofs need considerable mathematical sophistication. Even then, errors will occur in many proofs.

In the sample used in this article for 1000 function points there were about 690 specific requirements that need to be proved. This is why even small applications that use proofs take a long time, because proofs are time consuming.

It would be essentially impossible to use proofs of correctness on an application of 10,000 function points because there would be 7.407 specific algorithms to be proved and that might take several years, during which the requirements would have changed

...), for specific algorithms to be proved and validated over several years, during which the requirements would have changed so much that the earlier proofs might no longer apply.

Matching Software Methodologies with Projects

Since no method is top-ranked in every category, readers may well ask how to select methods that match the needs of their projects.

For smaller applications of 1000 function points or less where speed of delivery is the most critical parameter, then XP, Agile, and TSP are all very good choices.

For complex applications that might need FDA approval, operate weapons systems, or control sensitive financial data, high quality levels are mandatory. In this class TSP and CMMI 5 are the top choices, with XP as another possible method. Agile has been used for such applications but needs to be bent and twisted so much that it no longer is very agile.

For applications that might last for more than 10 years or which require very frequent enhancements and therefore need well designed interior structures, TSP would be the top choice, with CMMI 5 and XP also possibilities.

SUMMARY AND CONCLUSIONS

As this article is written, the software industry has about 55 different development methodologies. This is too large a number to compare in a short article.

For the 10 methods compared here, most have had some successes and most have had a few failures too.

Overall, the Agile family and the methods that emphasize speed have achieved their goal, and they are fairly quick.

The methods that emphasize quality such as TSP and CMMI 5 have also achieved their goals, and deliver very few defects.

No single method appears to be a universal panacea that can be successful on every size and kind of software application.

This article attempts to show the methods that give the best fit to three important factors: 1) speed; 2) quality; 3) long-range economic value.

REFERENCES AND READINGS

Jones, Capers; "Early Sizing and Early Risk Analysis"; Capers Jones & Associates LLC; Narragansett, RI; July 2011.

Jones, Capers and Bonsignour, Olivier; The Economics of Software Quality; Addison Wesley Longman, Boston, MA; ISBN 10: 0-13-258220-1; 2011; 585 pages.

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

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

Jones, Capers; Estimating Software Costs; McGraw Hill, New York, NY; ISBN-13: 978-0-07-148300-1; 2007.

Jones, Capers; Software Assessments, Benchmarks, and Best Practices; Addison Wesley Longman, Boston, MA; ISBN 0-201-48542-7; 2000; 657 pages.

Jones, Capers; Conflict and Litigation Between Software Clients and Developers; Software Productivity Research, Inc.; Burlington, MA; September 2007; 53 pages; (SPR technical report).

Systems Engineering News

Status of OMG Systems Modeling Language (OMG SysML) V1.3 and V1.4

The OMG Systems Modeling Language (OMG SysML™) is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. In particular, the language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models. The current version is 1.2, with Version 1.3 at Beta 2 status.

SysML is defined as an extension of the OMG UML 2 Superstructure specification. SysML is intended to be supported by two evolving interoperability standards including the OMG XMI 2.1 model interchange standard for UML 2 modeling tools and the ISO 10303 STEP AP233 data interchange standard for systems engineering tools.

SysML supports the OMG's Model Driven Architecture (MDA) initiative by its reuse of UML and related standards.

The OMG SysML Revision Task Force for SysML version 1.3 was chartered in September 2009 and is co-chaired by Roger Burkhart and Rick Steiner. The scope of revisions through the Revision Task Force is limited by OMG policy. Major revisions are handled through a new request for proposal.

SysML 1.3 has both added some features and deprecated others. Details can be found in the book *A Practical Guide to SysML: The Systems Modeling Language*, by Sanford Friedenthal, Alan Moore, and Rick Steiner, Elsevier, 28 October 2011, especially in Chapter 7.

The SysML 1.3 Revision Task Force (RTF) process generated the following documents:

- ptc/2011-08-08 (RTF Report - full record of RTF votes and issue resolutions)
- ptc/2011-08-07 (Submission inventory document)
- ptc/2011-08-09, ptc/2011-08-10 (Beta "convenience document," with and without change bars)
- ptc/2011-08-11, ptc/2011-08-12 (Normative and non-normative XMI).

Work is also progressing on a SysML V1.4, as reported by Roger Burkhart (Deere & Company) and Conrad Bock (NIST) at a meeting of the OMG Systems Engineering Domain Special Interest Group (SE DSIG) at Santa Clara, CA, USA over 12-13 December 2011. The program for development of SysML V1.4 was reported in SyEN 037, October, 2011.

A list of issues put before the OMG SysML Revision Task Force (RTF) and their status is at <http://www.omg.org/issues/sysml-rff.html>.

Notes:

- OMG, Object Management Group, SysML are trademarks of Object Management Group. All other trademarks are the property of their respective owners.
- SysML User Group is now Open Group
- James Byrne's SysML User Group on LinkedIn is now an open group. All discussions are fully visible, searchable, and shareable on the Web. Discussions prior to the date of going open are now closed in a members-only archive.

[More information](#)

Working Group Meetings at INCOSE IW2012 (with opportunity to meet at RAMS 2012)

Working Group meetings at INCOSE IW2012 (with opportunity to meet at RAMS 2012). See: http://www.linkedin.com/e-dw4hik-qxqfse3w-21/vaq/89416928/1218517/-1/view_disc/?hs=false&tok=2u2oaiqq6XoB41

The INCOSE Reliability Engineering Working Group Charter has recently been approved, and meetings will be held on 21 & 22 January 2012 in Jacksonville, Florida, to plan and initiate working group activities (<http://www.incose.org>). Since I plan to attend RAMS 2012 directly after IW2012, interested persons can contact me for informal meetings from 24 to 26 January 2012 in Reno, Nevada. (<http://rams.org>). Please do not hesitate to contribute to this working group and send me your comments, suggestions, ideas, etc.

OMG and Business Process Incubator Announce Relaunch of Enhanced BPMN.org Website

OMG® and the Business Process Incubator (BPI) have joined forces to enhance the Business Process Model and Notation specification's website, BPMN.org, with more content and tools. The site is now live at <http://www.bpmn.org>.

BPMN is a graphical notation that depicts the steps in a business process. BPMN depicts the end to end flow of a business process. The notation has been specifically designed to coordinate the sequence of processes and the messages that flow between different process participants in a related set of activities. This widely used and implemented standard (73 current and three planned implementations to date) is in use in both industry and government projects. BPMN.org provides information to organizations looking to implement the standard.

Note: OMG, Object Management Group, BPMN are trademarks of Object Management Group. All other trademarks are the

Note. OMG, Object Management Group, BPMN are trademarks of Object Management Group. All other trademarks are the property of their respective owners.

The Grand Opening of the Global Association For Systems Thinking

A group of 22 Global Thought Leaders have come together to take Systems Thinking mainstream and around the world with the opening of the Global Association for Systems Thinking (GAST). It is intended to become the premier global Membership alliance and world-wide clearinghouse for people from all walks of life that have come together to share their knowledge, experience and expertise in the field of Systems Thinking and its universal applications.

[More Information](#)

Get ready for the next wave of engineers—they may be a bit different than what you're used to

In 2011, engineering firms have had difficulty finding engineers to fill two out of five open positions. While this is disheartening, enrollment and degrees awarded have been increasing for the past 10 years, but two-thirds of engineering graduates are moving into nontechnical fields. Why are so many young engineers moving away from their education and the engineering industry?

[More Information](#)

Should Computer Science Be Required in K-12?

Computer science is not widely taught, even though programming may be one of the most important skills of the 21st century. While most schools do recognize the importance of helping students learn how to use new technologies, you'll still find scant opportunities in K-12 classes for students to learn how to actually *build* those very technologies. A [report](#) issued last year by the Association of Computing Machinery (ACM) found that very few states of the United States offer K-12 computer science education at all. Just nine states allow CS courses to count towards graduation requirements for mathematics or science. And no U.S. states require computer science for graduation.

[More Information](#)

Paul Croll Is Awarded 2012 IEEE Computer Society Hans Karlsson Award

IEEE Computer Society Vice President for Technical and Conference Activities Paul Croll has been named the 2012 recipient of the organization's Hans Karlsson Award, which commemorates achievement in computer standards. Croll was recognized "for dedicated leadership of the IEEE Systems and Software Engineering Standards Committee, and for his diplomacy and collaboration in facilitating the development of a collection of high-quality standards." Croll is a Fellow in CSC's Defense Group, where he is responsible for researching, developing, and deploying systems and software engineering practices, including those for cybersecurity. He also serves as chief scientist for CSC's Defense & Maritime Enterprise Technology Center. As vice president for Technical and Conference Activities, Croll oversees the work of 35 Technical Committees and Task Forces and more than 200 conferences. He chairs the IEEE Software and Systems Engineering Standards Committee and is vice chair of ISO/IEC JTC1/SC7, which covers software and systems engineering. Croll has been active in the development of ISO/IEC/IEEE 15026, the benchmark standard for system and software assurance in the lifecycle processes, and ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207, the framework standards for system and software lifecycle processes, as well as several other ISO/IEC/IEEE standards. In addition, through an INCITS/CS1 Ad Hoc committee, he is active in the ISO/IEC SC 27 effort to develop new standards for cybersecurity in supply-chain risk management.

[More Information](#)

Ask Robert

Q. Why would you want to measure requirements quality?

A. There are several reasons:

- The most basic reason: do we have a requirements problem, and if so, how big a problem do we have? (i.e., as an input to estimating the amount of work to do a requirements analysis)
- We are a manager and have staff developing a requirements specification or doing a requirements analysis. Are they working effectively?
- We set a standard for our requirements specifications in terms of the value of a requirements quality metric, actually, more probably three values for different circumstances:
 - Original novel development in demanding circumstances

- Original novel development in demanding circumstances
- Original novel development in non-demanding circumstances
- Very routine development.
- In contracting with a service provider to develop a requirements specification or do a requirements analysis to a defined standard as a professional service provider
- Providing professional advice on requirements quality, in terms of the value of a quality metric, followed an interpretation of the consequences of the quality value in the circumstances which apply
- In requirements-related litigation.

Of all the metrics in use in engineering, I find a requirements quality metric to be one of the most valuable, addressing as it does what is historically the single biggest problem in engineering: developing the wrong thing.

- Robert Halligan FIE Aust.

Featured Society

Americas Requirements Engineering Association (AREA)

AREA is presented to readers of SyEN in the form of a Press Release dated December 5, 2011 from founder of the Association, Gary Gack.

For additional information, contact Gary Gack:

Email: [gqack \(at\) A-RE-A.org](mailto:gqack(at)A-RE-A.org)

Tel: +1 904 579-1894

Requirements Engineering (also known as “business analysis”) is a critically important element of successful software engineering. Unfortunately, requirements are an important source of software defects. According to industry data, approximately 17% of all software defects originate in requirements and on average only 77% will be removed before delivery (Jones2008, p. 434). In other words, about 33% of all delivered defects originate in requirements. This Association (“AREA”) has been established in an effort to encourage improvements in this critical aspect of software engineering.

Membership

Membership is available at no cost to anyone involved in any way with the field of requirements engineering. Our focus is North, Central, and South America, but we welcome anyone involved in these countries regardless of country of residence. Membership currently includes residents of Argentina, Bolivia, Brasil, Canada, United States, and Venezuela.

Association Objectives:

- Promote effective software requirements engineering practices throughout North, Central, and South America. Parties outside the Americas who may provide products or services to the Americas are equally welcome to participate.
- Increase industry awareness of requirements engineering standards, certifications, tools, and “good” practices.
- Provide a forum for a free interchange of ideas and opinions related to software requirements engineering. A LinkedIn group, “Americas Requirements Engineering Association” has been established to provide a forum for discussion. Members of the LinkedIn group are automatically members of the Association and are listed in the membership section of www.A-RE-A.org
- Maintain a catalog of open source software requirements engineering tools. (See the web site for a current list – suggested additions are welcome)
- Maintain a comprehensive bibliography of software requirements engineering texts and articles.

Association Advisors

A distinguished group of industry experts have agreed to act as advisors to the Association. Advisors, among others, include Capers Jones - author of 18 books covering metrics, estimating, and best practices as well as numerous articles (see bibliography); Patricia McQuaid, PhD, President, ASTQB (American Software Testing Qualifications Board), Professor of Information Systems at California Polytechnic State University; Tom Love, PhD, CEO, ShouldersCorp; Martin Tornquist, Professor of Software Engineering at UFRGS (southern Brazil’s most prestigious public university); and Ralph Young, CSEP, Editor, the Systems Engineering Newsletter.

[More information](#)

INCOSE Technical Operations

INCOSE South Africa Chapter Systems Engineering Training Working Group

Charter developed by INCOSE SA, PO Box 67018, Highveld Park, 0169, South Africa

INCOSE SA Communications Officer - Phone: +27 83 302 6528 - Fax: +27 86 536 8122 Email: [info \(at\) incose.org.za](mailto:info@incose.org.za)

Web site: <http://www.incose.org.za>

DOCUMENT TITLE: SYSTEMS ENGINEERING TRAINING WORKING GROUP CHARTER

DOCUMENT NUMBER: INCOSE SA 2011-01 (slightly abridged for SyEN)

1. PURPOSE Systems Engineering (SE) training in South Africa is currently based on the initiative and perception of individual educational institutions and training providers. The real industry need for SE training and the requirements for such SE training are unknown. The purpose of the INCOSE SA SE Training Working Group (WG) is to establish a consolidated SE training need for SA and to provide guidance on a SE reference curriculum.

2. GOALS The goals of the INCOSE SA SE Training WG are to: a. Establish the local need for SE training (from companies connected to the CHAPTER through corporate or individual membership). b. Identify the current local offerings on SE training (from major academic institutions or service providers involved in SE training in South Africa). c. Present a consolidated SE training need to local training providers. d. Discuss the mapping of their respective offerings to the consolidated need with each academic institution or training provider individually. e. With inputs from the BKCASE1 project, provide guidelines on what a SE reference curriculum should include. f. Establish a sustainable process for periodic review of SE training needs and guidelines. 1 The BKCASE (Body of Knowledge and Curriculum to Advance Systems Engineering) project is an initiative of the Stevens Institute of Technology together with the Naval Postgraduate School and the US Department of Defense. The BKCASE project scope is to define a Systems Engineering Body of Knowledge (SEBoK) and use the SEBoK to develop a Graduate Reference Curriculum for Systems Engineering (GRCSE, pronounced "Gracie").

3. SCOPE The scope of the INCOSE SA SE Training WG is to consolidate SE training requirements for South Africa and provide guidance to educational institutions and training providers operating in South Africa on a SE reference curriculum that will address the consolidated SE training requirements.

4. INPUT REQUIREMENTS a. List of companies connected to the CHAPTER through corporate or individual membership (from Membership Officer). b. List of major academic institutions or service providers involved in SE training in South Africa. c. Cooperation will be required from companies connected to the CHAPTER through corporate or individual membership, for the establishment of a consolidated need. d. Cooperation will be required from major academic institutions or service providers involved in SE training in South Africa for: i. The establishment of current offerings. ii. Individual discussions on mapping of current offerings to the consolidated need. iii. Individual discussions on how current offerings can be tailored to meet the consolidated need. e. INCOSE SA financial support for visiting individual organizations. f. Support from employers of participating WG members to allow time for visiting individual organizations.

5. MEMBERS, ROLES AND RESPONSIBILITIES Lead: Alwyn Smit Responsibilities: Status Reporting to INCOSE SA President Chair at WG Meetings Recruitment of Members Working Group member database Working Group web page on INCOSE SA website Owner of LinkedIn INCOSE SA SE Training Group Co-Lead(s): To be determined later Members: To be listed in member database

6. OUTCOMES (PRODUCTS/SERVICES) a. The documented local need for SE training (from companies connected to the CHAPTER through corporate or individual membership). b. The documented list of current local offerings on SE training (from major academic institutions or service providers involved in SE training in South Africa). c. The documented mapping of respective offerings to the consolidated need (Confidential to individual institutions or service providers). d. Documented guidelines on what a SE reference curriculum should include.

7. MODUS OPERANDI The SE Training WG will be conducting its business as follows: a. A yearly meeting at the INCOSE SA Chapter Conference. b. Continuous electronic communication throughout the year. c. Monthly electronic reporting to the Chapter President. d. Ad hoc meetings with individual companies, major academic institutions and training providers. e. Decision making will be based on broad consensus by the WG members.

8. MEASURES OF SUCCESS The WG should ultimately be measured by the contribution it makes towards the establishment of SE training in SA that meets the documented consolidated training need. Specific measures of success will include: a. At least one member from each identified company, academic institution and training provider. b. Members should include representation from all industry sectors represented in INCOSE SA. c. General acceptance of the consolidated training needs by academic institutions and service providers d. Adoption of the guidelines by academic institutions and service providers.

9. RESOUCRE REQUIREMENTS It is envisaged that INCOSE SA will fund travel expenses in South Africa for WG members that have to visit specific companies or institutions. It is possible that formal cooperation with academic institutions and training providers may require Memoranda of Understanding or Memoranda of Agreement.

10. DURATION The SE Training WG will remain in effect until rescinded by the signatories

Systems Engineering Tools News

UML/SysML Tool Vendor Model Interchange Test Case Results Now Available

Object Management Group ([OMG's®](#)) [Model Interchange Working Group \(MIWG\)](#) announced on December 1, 2011 the public availability of the vendor test case results that demonstrate their UML® and OMG SysML™ model interchange capability. Six tool vendors, specifically Atego, IBM, NoMagic, Sodus (supporting IBM Rhapsody), SOFTEAM, and Sparx Systems, supporting six tools, participated in the model interchange testing using XMI® as the interchange standard. The test results encompass a test suite of sixteen test cases that provide test coverage of a majority of the commonly used UML and SysML functionality.

Sandy Friedenthal, chair of the MIWG, noted that "the ability to interchange models offers the potential to significantly improve productivity, quality, and the long term retention of models. The MIWG test suite demonstrates a broad interchange capability that includes the interchange of executable activity models, and the interchange of domain specific models using profiles."

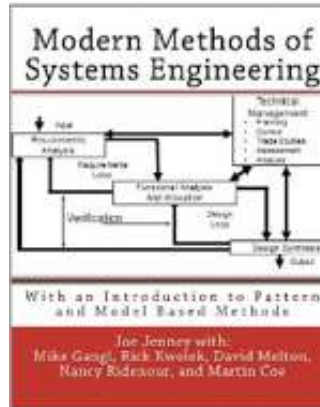
Most importantly, the vendor participants have worked together to make model interchange a success. Over a series of incremental releases of the MIWG test suite, vendors have collaboratively interchanged each of the MIWG tests between their tools, identified issues and resolved them in subsequent tool releases. As a result, the versions of these tools now on the market are far more interoperable than they were three years ago when the MIWG effort started.

[More information](#)

Systems Engineering Books, Reports, Articles and Papers

Modern Methods of Systems Engineering

[Joe Jenney](#) (Author), [Mike Gangl](#) (Contributor), [Rick Kwolek](#) (Contributor), [David Melton](#) (Contributor), [Nancy Ridenour](#) (Contributor), [Martin Coe](#) (Contributor)



The publishers of *Modern Methods of Systems Engineering* state that the book reviews the fundamentals of systems engineering and shows how these fundamentals can be integrated with modern model and pattern based methods to achieve reuse of systems engineering as well as reuse of hardware and software designs. The methods and tools presented are said to be complementary to the (U.S.) DoD Systems Engineering Fundamentals, the NASA Systems Engineering Handbook, and the INCOSE Systems Engineering Handbook. The publishers state that exercises are included so that the book is a guide to self-training for engineers new to systems engineering, and to experienced systems engineers wanting to learn new and better methods.

[Additional information](#)

Enterprise Release Management: Agile Delivery of a Strategic Change Portfolio

Author: Louis Taborda



Published by: [Artech House](#)

ISBN: 978-1-60807-168-5

Publication Date: September 30, 2011

Binding(s): Hardcover

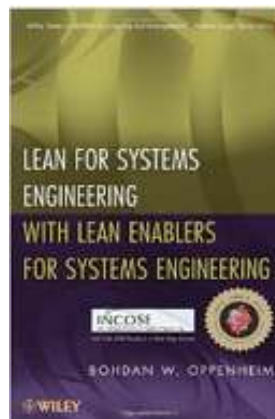
Abstract: Based on a review on Amazon by John R. Vacca of Pomeroy, Ohio.

The author begins by making the case for release management at the enterprise level. Next, he reviews the two apparently opposed management viewpoints of plan-driven versus agile development that respectively addresses change and complexity. Then, he discusses the ubiquitous project paradigm, identifying conditions in the enterprise that can reduce the effectiveness of project management. He identifies the enterprise pattern that inevitably results when multiple projects impact complex enterprise architecture. He also identifies the release as an important factor in selecting a change portfolio for execution. Then, he looks at Enterprise Release Management (ERM) practices, in terms of a release-centric perspective to an industry study on business alignment versus IT efficacy. The author then introduces the matrix notation that resolves the most complex enterprise pattern and offers a representation for the multiple stakeholder viewpoints present in an enterprise release. Next, he is concerned with the matrix model that provides a multi-project release life cycle model. Finally, the author discusses the implementation of ERM in the enterprise and the implications of the new approach and its relationship to different practices in the enterprise.

[More Information](#)

Lean for Systems Engineering with Lean Enablers for Systems Engineering Wiley Series in Systems Engineering and Management

Author: Bohdan W. Oppenheim - founder and Co-Chair of the Lean Systems Engineering Working Group of INCOSE and leader of the development effort of Lean Enablers for Systems Engineering.



ISBN: 978-1118008898

Publisher: John Wiley

Publication Date: September 27, 2011

The book describes the field of Lean for Systems Engineering and the new intellectual product Lean Enablers for Systems Engineering, LEFSE, a comprehensive checklist of 194 practices of Systems Engineering, focused on creating value without waste in large technological programs.

Key Topics Covered:

- Introduction.
- A Brief History of Recent Management Paradigms.
- Lean Fundamentals.
- Lean in Product Development.
- From Traditional to Lean Systems Engineering.
- Development of Lean Enablers for Systems Engineering (LEfSE).
- Lean Enablers for System Engineering.
- General Guidance for Implementation.

[More information](#)

INCOSE INSIGHT, December 2011, Vol 14 - Issue 4

The December 2011 INSIGHT is ready to view or download by INCOSE members on INCOSE Connect <https://connect.incose.org/default.aspx>. Special Feature: Systems Engineering Research Challenges in French Universities. This issue is devoted to coverage of the French Systems Engineering Academia-Industry Forum, organized by AFIS, the French chapter of INCOSE, with the support of the University of Bordeaux. The objective of the forum was to develop strong relationships between industry and academia. The theme section begins with an account of the RobAFIS student competition. The 14 papers that follow are expanded from poster presentations by PhD students during the workshop on "Learning Systems Engineering while Doing Research," translated into English and improved through a peer-review process. The articles give insight into current academic doctoral research that provides advanced methods for managing complex, multi-national enterprises.

Conferences & Meetings

6th International Conference on Design Principles & Practices

January 20 - 22, 2012, Los Angeles, CA, USA

[More information](#)

INCOSE International Workshop (IW) 2012

January 21 - 24, 2012, Jacksonville, FL, USA

[More information](#)

Ontologies Influences in Systems Engineering

January 23 - 27, 2012 Carlos III, Madrid University Spain

[More information](#)

Eighth Asia-Pacific Conference on Conceptual Modelling (APCCM 2012)

January 30 - February 2, 2012, RMIT, Melbourne, Australia

[More information](#)

IBM Systems and Software Engineering Symposium COMPLEX NEW

February 8, 2012, Orlando, FL, USA

[More information](#)

Systems Engineering for Defense Symposium

February 15 - 16, 2012, Defense Academy of the United Kingdom, Shrivenham, UK

[More Information](#)

ESSoS12 - International Symposium on Engineering Secure Software and Systems

February 16 - 17, 2012, Eindhoven, The Netherlands

[More information](#)

16th GfSE Workshop 2012

February 17, 2012, Hannover, Germany

[More information](#)

IBM Systems and Software Engineering Symposium NEW

March 1, 2012, Dallas, TX, USA

[More information](#)

IEEE CogSIMA 2012 – 2nd International Conference on Cognitive Methods in Situation Awareness and Decision Support

March 6 - 8, 2012, New Orleans, LA, USA

[More information](#)

28th Annual National Test & Evaluation Conference 

March 12 - 15, 2012, South Carolina, USA

[More information](#)

2nd International Requirements Engineering Efficiency Workshop (REEW 2012) 

March 19, 2012, Essen, Germany on the 19th of March

[More information](#)

16th International GI/ITG Conference on Measurement, Modelling and Evaluation of Computing Systems and Dependability and Fault-Tolerance (MMB & DFT 2012)

March 19 - 21, 2012, Kaiserslautern, Germany

[More information](#)

CSER 2012 – Conference on Systems Engineering Research

March 19 - 22, 2012, St Louis, Missouri, USA

[More information](#)

The 9th ENTERPRISE ENGINEERING Track at ACM-SAC 2012

March 25 - 29, 2012, Riva del Garda, Trento, Italy

[More information](#)

Fifth Edition of the Requirements Engineering Track (RE-Track'12)

Part of the 27th ACM Symposium on Applied Computing (SAC 2012)

March 25 - 29, 2012, University of Trento, Trento, Italy

[More information](#)

2nd International Workshop on Model-driven Approaches for Simulation Engineering

Part of the Symposium on Theory of Modeling and Simulation, (SCS SpringSim 2012)

March 26 - 29, 2012, Orlando, FL, USA

[More information](#)

Symposium On Theory of Modeling and Simulation, TMS'12

Part of the 2012 SpringSim - Spring Simulation Multi-Conference

March 26 - 29, 2012, Orlando, FL, USA

[More information](#)

Software for Theory of Modeling & Simulation at TMS/DEVS'12

March 26 - 29, 2012, The Florida Hotel, Orlando, FL, USA.

[More Information](#)

2012 SpringSim - Spring Simulation Multi-Conference

March 26 - 30, 2012, Orlando, FL, USA

[More Information](#)

Applied Ergonomics Conference 2012

March 26 - 29, 2012, Gaylord Opryland Resort and Convention Center, Nashville, TN, USA

[More information](#)

The 31st International Conference on Modelling, Identification and Control

April 2 - 4, 2012, Phuket, Thailand

[More information](#)

Fourth NASA Formal Methods Symposium (NFM 2012)

April 3 - 5, 2012, Norfolk, VA, USA

[More Information](#)

9th IEEE International Conference and Workshop on Engineering of Autonomic and Autonomous Systems (EASe 2011)

April 11 - 13, 2012, Novi Sad, Serbia, Europe

[More Information](#)

Workshop on Requirements Engineering (WER'12)

April 24 - 27, 2012. Buenos Aires This workshop will be held in parallel with ClbSE'12 and ESELAW'12.

[More information](#)

SETE APCOSE 2012

April 30 - May 2, 2012, Brisbane Convention and Exhibition Centre, Brisbane, QLD, Australia

[More information](#)

Software Engineering Institute Architecture Technology User Network (SATURN) 2012 Conference

May 7 - 11, 2012, St. Petersburg, FL, USA

[More Information](#)

1st Annual Systems Engineering in the Washington Metropolitan Area Conference (SEDC 2012)

May 14 - 16, 2012, George Mason Inn and Conference Center, Washington, USA

[More information](#)

IIE Annual Conference and Expo 2012

May 19 - 23, 2012, Hilton Bonnet Creek, Orlando, FL, USA

[More information](#)

Risk Engineering Society Conference: RISK 2012

May 23 - 24, 2012, Lovedale, NSW, Australia

[More information](#)

12th International Design Conference Design 2012

May 21 - 25, 2012, Dubrovnik, Croatia

[More information](#)

Australian System Safety Conference 2012

May 23 - 25, 2012, Brisbane, Australia

[More information](#)

12th International SPICE Conference on Process Improvement and Capability dEtermination in Software, Systems Engineering and Service Management

May 29 - 31, 2012, Palma de Mallorca, Spain

[More Information](#)

Engineering Leadership Conference (ELC 2012)

May 30 - June 2, 2012, Adelaide, Australia

[More information](#)

International Conference on Software and Systems Process (ICSSP) 2012

June 2 - 3, 2012, Zurich, Switzerland (co-located with ICSE 2012)

[More Information](#)

119th American Society for Engineering Education (ASEE) Annual Conference & Exposition NEW

June 10 - 13, 2012, San Antonio, Texas, USA

[More information](#)

iFM2012 ABZ 2012 - Abstract State Machines

June 18 - 22, 2012, CNR Research Area of Pisa, Italy

[More information](#)

12th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Model-Driven Engineering (SFM-12:MDE) NEW

June 18 - 23, 2012

[More Information](#)

PETRI NETS 2012 - 33rd International Conference on the Application and Theory of Petri Nets and Concurrency

June 25 - 29, 2012, Hamburg, Germany

[More information](#)

12th International Conference on Application of Concurrency to System Design (ACSD 2012)

June 27 - 29, 2012, Hamburg, Germany

[More Information](#)

8th European Conference on Modelling Foundations and Applications

July 2 - 5, 2012, Technical University of Denmark, Denmark

[More information](#)

INCOSE International Symposium (IS) 2012

July 9 - 12, 2012, Rome, Italy

[IS2012 Call for Papers](#): Deadline for draft papers, and proposals for panels and tutorials for IS2012 is November 8th, 2011.

[More information](#)

Interdisciplinary Network for Group Research (INGRoup) - Seventh Annual Conference

July 12 - 14, 2012, Chicago, IL, USA

[More information](#)

International Conference of the System Dynamics Society, 2012

July 22 - 26, 2012, St. Gallen, Switzerland

[More Information](#)

4th Improving Systems & Software Engineering Conference (ISSEC) 2012

August 15 - 16, 2012, Melbourne, Australia

[More information](#)

20th IEEE International Requirements Engineering Conference

September 24 - 28, 2012, Chicago, Illinois, USA

[More information](#)

MODELS 2012, ACM/IEEE 15th International Conference on Model-Driven Engineering Language & Systems - Call for Papers - Deadline 19 March 2012

Sept. 30th - Oct. 5th, 2012 - Innsbruck/AUSTRIA

[More Information](#)

Human Factors and Ergonomics Society HFES 2012 Annual Meeting

October 22 - 26, 2012, Boston, MA, USA

[More information](#)

The World Congress on Engineering and Computer Science 2012

October 24 - 26, 2012, San Francisco, USA

Building Business Capabilities (BBC) 2012

October 28 - November 2, 2012, Fort Lauderdale, FL, USA

[More information](#)

3rd International Conference on Complex Systems Design & Management (CSD&M 2012)

December 12 - 14, 2012, Cité Internationale Universitaire, Paris (France)

[More information](#)

Education and Academia

US Irvine Extension and Georgia Institute of Technology Articulation Agreement Between Systems Engineering Programs Partnership Creates New Educational Opportunities for Systems Engineering Professionals

The U.S. University of California, Irvine Extension and the Georgia Institute of Technology today announced their agreement to articulate earned credits for systems engineering classes in order to open new pathways for students pursuing systems engineering. Through the new articulation agreement, course participants who complete designated UC Irvine Extension Systems Engineering courses may receive academic credit toward the Georgia Institute of Technology Professional Master's in Applied Systems Engineering (PMASE) degree program. Both programs are delivered primarily online. Georgia Tech's program does require a few campus visits over the two-year duration.

For more information about the articulation agreement or to enroll in UC Irvine Extension's Systems Engineering certificate program visit or call 949-824-5380. For information about the Georgia Institute of Technology's PMASE program, or to apply, visit <http://www.pmase.gatech.edu>

[More Information](#)

Stevens SSE Faculty Lead Efforts to Release Newest Version of the Worldwide Graduate Reference Curriculum for Systems Engineering "GRCSE 0.5"

The Graduate Reference Curriculum for Systems Engineering (GRCSE™) is the first ever worldwide release of an "official" reference curriculum for a systems-centric master's level program in systems engineering. GRCSE provides a standardized set of recommendations for graduate level systems engineering programs, together with implementation guidance for a university to satisfy those recommendations. The newest version of GRCSE™ is available at <http://www.bkcase.org/grcse-05/> for comment.

The "GRCSE project is ready for review and early adoption," said Dr. Art Pyster of Stevens Institute of Technology, principal investigator on the GRCSE™ project. "GRCSE was motivated by the huge diversity in systems engineering graduate programs and the need to create a common core of systems engineering content in worldwide graduate programs."

[More Information](#)

Systems Engineering Scholarships Available

Three partner organizations at the Velocity center are teaming up with Oakland University (USA) to offer scholarships for OU students pursuing degrees in science, technology, engineering or mathematics. The intent of the Product Lifecycle Management Scholarship Program is to provide 32 four-year scholarships at \$4,800 per year for students seeking careers in Industrial and Systems Engineering, and Mechanical Engineering.

[More Information](#)

INCOSE Webinars

The International Council on Systems Engineering (INCOSE) provides a free monthly web-based service for the greater Systems Engineering community. Once a month, recognized experts in the field present the "State of the Art" in Systems Engineering. These presentations are conducted live, via the internet. Each lasts for one hour, including opportunity for discussion and questions. Members of INCOSE can access the web archive and access recordings and materials. For example, there is a recording of Professor Avraham Shtub's December 2011 Presentation, "Training System Engineers in Project Management – A Simulation-based Training (SBT) Approach". The tool developed for this endeavor won the 2008 Best Product Award from the Project Management Institute (PMI).

[More Information](#)

Some Systems Engineering-Relevant Websites

<http://themanagersguide.blogspot.com/>

The blog aims to teach the fundamentals of modern systems engineering and introduce new methods that reduce the time and costs to prepare systems engineering documentation for complex systems.

http://www.sebokwiki.org/index.php/Main_Page

This Wiki site contains version 0.5 of the Guide to the Systems Engineering Body of Knowledge (SEBoK). The SEBoK 0.5 Introduction contains information about the Purpose of the SEBoK, Scope of the SEBoK, and the Uses of the SEBoK. This SEBoK is the product of the work of many contributors: sponsor, partner organizations, core team, authors, reviewers, and participants. Primary leadership of the project was provided by Stevens Institute of Technology and the Naval Postgraduate School, working together through the U.S. Department of Defense Systems Engineering Research Center. The primary funding sponsor was the office of the Deputy Assistant Secretary of Defense for Systems Engineering (DASD/SE).

Standards

Recent ISO/IEC/SC7 Systems Engineering Relevant Standards

ISO/IEC/IEEE 24765:2010 provides a common vocabulary applicable to all systems and software engineering work. It was

prepared to collect and standardize terminology. ISO/IEC/IEEE 24765:2010 is intended to serve as a useful reference for those in the information technology field, and to encourage the use of systems and software engineering standards prepared by ISO and liaison organizations IEEE Computer Society and Project Management Institute. ISO/IEC/IEEE 24765:2010 includes references to the active source standards for each definition so that the use of the term can be further explored.

ISO/IEC TR 24748-1:2010 Systems and software engineering -- Life cycle management -- Part 1: Guide for life cycle management

ISO/IEC TR 24748-1:2010 provides information on life cycle concepts and descriptions of the purposes and outcomes of representative life cycle stages. It also illustrates the use of a life cycle model for systems in the context of ISO/IEC 15288 and provides a corresponding illustration of the use of a life cycle model for software in the context of ISO/IEC 12207. ISO/IEC TR 24748-1:2010 additionally provides detailed discussion and advice on adapting a life cycle model for use in a specific project and organizational environment. It further provides guidance on life cycle model use by domains, disciplines and specialties. ISO/IEC TR 24748-1:2010 gives a detailed comparison between prior and current versions of ISO/IEC 12207 and ISO/IEC 15288, as well as advice on transitioning from prior to current versions and on using their application guides. The discussion and advice are intended to provide a reference model for life cycle models, facilitate use of the updated ISO/IEC 15288 and ISO/IEC 12207, and provide a framework for the development of updated application guides for those International Standards. ISO/IEC TR 24748-1:2010 is a result of the alignment stage of the harmonization of ISO/IEC 12207 and ISO/IEC 15288.

ISO/IEC TR 24748-2:2011 Systems and software engineering -- Life cycle management -- Part 2: Guide to the application of ISO/IEC 15288 (System life cycle processes)

ISO/IEC TR 24748-2:2011 is a guide for the application of ISO/IEC 15288:2008. It addresses system, life cycle, process, organizational, project, and adaptation concepts, principally through reference to ISO/IEC TR 24748-1 and ISO/IEC 15288:2008. It then gives guidance on applying ISO/IEC 15288:2008 from the aspects of strategy, planning, application in organizations, and application on projects. ISO/IEC TR 24748-2:2011 is intentionally aligned with both ISO/IEC TR 24748-1 and ISO/IEC TR 24748-3 (Guide to the application of ISO/IEC 12207) in its terminology, structure and content.

ISO/IEC TR 24748-3:2011 Systems and software engineering -- Life cycle management -- Part 3: Guide to the application of ISO/IEC 12207 (Software life cycle processes)

ISO/IEC TR 24748-3:2011 is a guide for the application of ISO/IEC 12207:2008. It addresses system, life cycle, process, organizational, project, and adaptation concepts, principally through reference to ISO/IEC TR 24748-1 and ISO/IEC 12207:2008. It gives guidance on applying ISO/IEC 12207:2008 from the aspects of strategy, planning, application in organizations, and application on projects. ISO/IEC TR 24748-3:2011 is intentionally aligned with both ISO/IEC TR 24748-1 and ISO/IEC TR 24748-2 (Guide to the application of ISO/IEC 15288) in its terminology, structure and content.

ISO/IEC NP 24748-4 Systems engineering -- Application and management of the systems engineering process This is a new standard with a target publication date of 15 December, 2014.

ISO/IEC 26702:2007 Systems engineering -- Application and management of the systems engineering process

ISO/IEC 26702:2007 defines the interdisciplinary tasks which are required throughout a system's life cycle to transform customer needs, requirements and constraints into a system solution. In addition, it specifies the requirements for the systems engineering process and its application throughout the product life cycle. ISO/IEC 26702:2007 focuses on engineering activities necessary to guide product development, while ensuring that the product is properly designed to make it affordable to produce, own, operate, maintain and eventually dispose of without undue risk to health or the environment. This is an ISO-badged version of IEEE 1220.

ISO/IEC TR 15026-1:2010 Systems and software engineering -- Systems and software assurance -- Part 1: Concepts and vocabulary

ISO/IEC TR 15026-1:2010 defines terms and establishes an extensive and organized set of concepts and their relationships, thereby establishing a basis for shared understanding of the concepts and principles central to ISO/IEC 15026 across its user communities. It provides information to users of the subsequent parts of ISO/IEC 15026, including the use of each part and the combined use of multiple parts. Coverage of assurance for a service being operated and managed on an ongoing basis is not covered in ISO/IEC 15026.

ISO/IEC 15026-2:2011 Systems and software engineering -- Systems and software assurance -- Part 2: Assurance case

ISO/IEC 15026-2:2011 specifies minimum requirements for the structure and contents of an assurance case to improve the consistency and comparability of assurance cases and to facilitate stakeholder communications, engineering decisions, and other uses of assurance cases. An assurance case includes a top-level claim for a property of a system or product (or set of claims), systematic argumentation regarding this claim, and the evidence and explicit assumptions that underlie this argumentation. Arguing through multiple levels of subordinate claims, this structured argumentation connects the top-level claim to the evidence and assumptions. Assurance cases are generally developed to support claims in areas such as safety, reliability, maintainability, human factors, operability, and security, although these assurance cases are often called by more specific names, e.g. safety case or reliability and maintainability (R&M) case. ISO/IEC 15026-2:2011 does not place requirements on the quality of the contents of an assurance case and does not require the use of a particular terminology or graphical representation.

quality of the contents of an assurance case and does not require the use of a particular terminology or graphical representation.

Likewise, it places no requirements on the means of physical implementation of the data, including no requirements for redundancy or co-location.

ISO/IEC 15026-3:2011 Systems and software engineering -- Systems and software assurance -- Part 3: System integrity levels

ISO/IEC 15026-3:2011 specifies the concept of integrity levels with corresponding integrity level requirements that are required to be met in order to show the achievement of the integrity level. It places requirements on and recommends methods for defining and using integrity levels and their integrity level requirements, including the assignment of integrity levels to systems, software products, their elements, and relevant external dependences. ISO/IEC 15026-3:2011 is applicable to systems and software and is intended for use by:

- definers of integrity levels such as industry and professional organizations, standards organizations, and government agencies;
- users of integrity levels such as developers and maintainers, suppliers and acquirers, users, and assessors of systems or software and for the administrative and technical support of systems and/or software products.

One important use of integrity levels is by suppliers and acquirers in agreements; for example, to aid in assuring safety, economic, or security characteristics of a delivered system or product. ISO/IEC 15026-3:2011 does not prescribe a specific set of integrity levels or their integrity level requirements. In addition, it does not prescribe the way in which integrity level use is integrated with the overall system or software engineering life cycle processes. ISO/IEC 15026-3:2011 can be used alone or with other parts of ISO/IEC 15026. It can be used with a variety of technical and specialized risk analysis and development approaches. ISO/IEC TR 15026-1 provides additional information and references to aid users of ISO/IEC 15026-3:2011. Assurance cases are covered by ISO/IEC 15026-2. ISO/IEC 15026-3:2011 does not require the use of assurance cases but describes how integrity levels and assurance cases can work together, especially in the definition of specifications for integrity levels or by using integrity levels within a portion of an assurance case.

ISO/IEC 15026-4:2011 Systems and software engineering -- Systems and software assurance -- Part 4: Assurance in the life cycle

This standard is under development.

ISO/IEC 29155-1:2011 Systems and software engineering -- Information technology project performance benchmarking framework -- Part 1: Concepts and definitions

ISO/IEC 29155-1:2011 identifies a framework for information technology (IT) project performance benchmarking (e.g. development or maintenance productivity) and related aspects (e.g. data collection and software classification). The framework consists of activities and components that are necessary to successfully identify, define, select, apply, and improve benchmarking for IT project performance. It also provides definitions for IT project performance benchmarking terms. The target audience of ISO/IEC 29155-1:2011 are the stakeholders of IT project performance benchmarking. ISO/IEC 29155-1:2011 does not prescribe how to organize benchmarking. It is out of the scope of ISO/IEC 29155-1:2011 to prescribe the name, format, or explicit content of the documentation to be produced from the benchmarking process.

ISO/IEC TR 24774:2010 Systems and software engineering -- Life cycle management -- Guidelines for process description

An increasing number of international, national and industry standards describe process models. These models are developed for a range of purposes including process implementation and assessment. The terms and descriptions used in such models vary in format, content and level of prescription. ISO/IEC TR 24774:2010 presents guidelines for the elements used most frequently in describing a process: the title, purpose, outcomes, activities, task and information item. Whilst the primary purpose of ISO/IEC TR 24774:2010 is to encourage consistency in standard process reference models, the guidelines it provides can be applied to any process model developed for any purpose.

ISO/IEC/IEEE 29148:2011 Systems and software engineering -- Life cycle processes -- Requirements engineering

ISO/IEC/IEEE 29148:2011 contains provisions for the processes and products related to the engineering of requirements for systems and software products and services throughout the life cycle. It defines the construct of a good requirement, provides attributes and characteristics of requirements, and discusses the iterative and recursive application of requirements processes throughout the life cycle. ISO/IEC/IEEE 29148:2011 provides additional guidance in the application of requirements engineering and management processes for requirements-related activities in ISO/IEC 12207:2008 and ISO/IEC 15288:2008. Information items applicable to the engineering of requirements and their content are defined. The content of ISO/IEC/IEEE 29148:2011 can be added to the existing set of requirements-related life cycle processes defined by ISO/IEC 12207:2008 or ISO/IEC 15288:2008, or can be used independently.

ISO/IEC 16085:2006 Systems and software engineering -- Life cycle processes -- Risk management

ISO/IEC 16085:2006 defines a process for the management of risk in the life cycle. It can be added to the existing set of system and software life cycle processes defined by ISO/IEC 15288 and ISO/IEC 12207, or it can be used independently. ISO/IEC

16085:2006 can be applied equally to systems and software. Risk management is a key discipline for making effective decisions and communicating the results within organizations. The purpose of risk management is to identify potential managerial and technical problems before they occur so that actions can be taken that reduce or eliminate the probability and/or impact of these problems should they occur. It is a critical tool for continuously determining the feasibility of project plans, for improving the search for and identification of potential problems that can affect life cycle activities and the quality and performance of products, and for improving the active management of projects.

ISO/IEC/IEEE 16326:2009 Systems and software engineering -- Life cycle processes -- Project management

ISO/IEC/IEEE 16326:2009 provides normative content specifications for project management plans covering software projects, and software-intensive system projects. It also provides detailed discussion and advice on applying a set of project processes that are common to both the software and system life cycle as covered by ISO/IEC 12207:2008 (IEEE Std 12207-2008) and ISO/IEC 15288:2008 (IEEE Std 15288-2008), respectively. The discussion and advice are intended to aid in the preparation of the normative content of project management plans. ISO/IEC/IEEE 16326:2009 is the result of the harmonization of ISO/IEC TR 16326:1999 and IEEE Std 1058-1998.

ISO/IEC TR 16337 Systems Engineering Handbook

This number has been designated for the INCOSE Systems Engineering Handbook

ISO/IEC/IEEE 15289:2011 Systems and software engineering -- Content of life-cycle information products (documentation)

ISO/IEC/IEEE 15289:2011 provides requirements for identifying and planning the specific information items (information products, documentation) to be developed and revised during systems and software life cycles and service processes. It specifies the purpose and content of all identified systems and software data records and life cycle information items, as well as records and information items for information technology service management. The information item contents are defined according to generic document types (description, plan, policy, procedure, report, request, and specification) and the specific purpose of the document. For simplicity of reference, each information item is described as if it were published as a separate document. However, information items may be unpublished but available in a repository for reference, divided into separate documents or volumes, or combined with other information items into one document. ISO/IEC/IEEE 15289:2011 is based on the life cycle processes specified in ISO/IEC 12207:2008 (IEEE Std 12207-2008) and ISO/IEC 15288:2008 (IEEE Std 15288-2008), and the service management processes specified in ISO/IEC 20000-1:2005 and ISO/IEC 20000-2:2005.

ISO/IEC TR 90005:2008 Systems engineering -- Guidelines for the application of ISO 9001 to system life cycle processes

ISO/IEC TR 90005:2008 provides guidance for organizations in the application of ISO 9001:2000 to the acquisition, supply, development, operation and maintenance of systems and related support services. It does not add to or otherwise change the requirements of ISO 9001:2000. The guidelines provided in ISO/IEC TR 90005:2008 are not intended to be used as assessment criteria in quality management system registration or certification. ISO/IEC TR 90005:2008 adopts ISO/IEC 15288 systems life cycle processes as a starting point for system development, operation or maintenance and identifies those equivalent requirements in ISO 9001:2000 that have a bearing on the implementation of ISO/IEC 15288. ISO/IEC TR 90005:2008 is appropriate to systems that are part of a commercial contract with another organization, a product available for a market sector, used to support the processes of an organization, embedded in a hardware product, or related to software services.

ISO/DIS 10303-233 Industrial automation systems and integration -- Product data representation and exchange -- Part 233: Application protocol: Systems engineering data representation This standard is under development.

ISO 31000:2009 Risk management -- Principles and guidelines

ISO 31000:2009 provides principles and generic guidelines on risk management. ISO 31000:2009 can be used by any public, private or community enterprise, association, group or individual. Therefore, ISO 31000:2009 is not specific to any industry or sector. ISO 31000:2009 can be applied throughout the life of an organization, and to a wide range of activities, including strategies and decisions, operations, processes, functions, projects, products, services and assets. ISO 31000:2009 can be applied to any type of risk, whatever its nature, whether having positive or negative consequences. Although ISO 31000:2009 provides generic guidelines, it is not intended to promote uniformity of risk management across organizations. The design and implementation of risk management plans and frameworks will need to take into account the varying needs of a specific organization, its particular objectives, context, structure, operations, processes, functions, projects, products, services, or assets and specific practices employed. It is intended that ISO 31000:2009 be utilized to harmonize risk management processes in existing and future standards. It provides a common approach in support of standards dealing with specific risks and/or sectors, and does not replace those standards. ISO 31000:2009 is not intended for the purpose of certification.

ISO/IEC 31010:2009 Risk management -- Risk assessment techniques

IEC 31010:2009 is a dual logo IEC/ISO, single prefix IEC, supporting standard for ISO 31000 and provides guidance on selection and application of systematic techniques for risk assessment. This standard is not intended for certification, regulatory or contractual use. NOTE: This standard does not deal specifically with safety. It is a generic risk management standard and any references to safety are purely of an informative nature. Guidance on the introduction of safety aspects into IEC standards is laid down in ISO/IEC Guide 51

ISO 19439:2006 Enterprise integration -- Framework for enterprise modeling

ISO 19439:2006 specifies a framework conforming to requirements of ISO 15704, which serves as a common basis to identify and coordinate standards development for modeling of enterprises, emphasizing, but not restricted to, computer integrated manufacturing. ISO 19439:2006 also serves as the basis for further standards for the development of models that will be computer-enactable and enable business process model-based decision support leading to model-based operation, monitoring and control. In ISO 19439:2006, four enterprise model views are defined in this framework. Additional views for particular user concerns can be generated but these additional views are not part of this International Standard. Possible additional views are identified in ISO 15704.

ISO 15704:2000 Industrial automation systems -- Requirements for enterprise-reference architectures and methodologies

ISO 15704:2000/Amd 1:2005 Additional views for user concerns

A Definition to Close On

Life Cycle

Life cycle: The scope of systems or product evolution beginning with the identification of a perceived customer need, addressing development, test, manufacturing, operation, support and training activities, and continuing through various upgrades or evaluations until the product disposal.

Source: EIA/IS-731.1 Document2 - Systems Engineering Capability Model

Life cycle: The system or product evolution initiated by a perceived stakeholder need through the disposal of the products.

Source: IEEE Std 1220-2005, IEEE Standard for Application and Management of the Systems Engineering Process; ISO/IEC 26702

Life cycle: The period of time that begins when a system is conceived and ends when the system is no longer available for use.

Source: IEEE STD 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology

Life cycle: The period of time between starting the design or modification of a hardware item and completing the design or modification up as far as transition to production.

Source: RTCA - Design Assurance Guidance for Airborne Electronic Hardware, 2000

Life cycle: The course of events that brings a new product into existence and follows its growth into a mature product and into eventual critical mass and decline.

Source: <http://www.investopedia.com/terms/l/lifecycle.asp#axzz1jdaZqIkI>

Product life cycle: A marketing theory in which products or brands follow a sequence of stages including: introduction, growth, maturity, and sales decline.

Source: <http://marketing.about.com/od/marketingglossary/g/prodlifedef.htm>

Systems development life cycle: The systems development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.

Source: <http://searchsoftwarequality.techtarget.com>

System life cycle: The useful life of an information system.

Source: http://www.pcmag.com/encyclopedia_term

Project lifecycle: The term project lifecycle models how a project is planned, controlled, and monitored from its inception to its completion.

Source: <http://infolific.com/technology/definitions/>

PPI News (see www.ppi-int.com)

PPI's Systems Engineering Goldmine (SEG) is Upgraded

PPIs Systems Engineering Goldmine, a free resource of downloadable systems engineering documents and searchable systems engineering definitions, has been upgraded for the benefit of PPI clients and the engineering community in general. Improvements include:

- faster searching
- broader scope of indexing of documents, giving a higher probability of finding information
- improved hosting, leading to even lower downtime

The team at PPI looks forward to all our professional friends and colleagues continuing to benefit from the Systems Engineering Goldmine.

New Cognitive Systems Design Video by Dr. Gavan Lintern

The developer and presenter of PPI's 5-day Cognitive Systems Engineering Course (see below), Dr. Gavan Lintern, has uploaded his video "Why Cognitive Systems Design?" to his website at www.cognitivesystemsdesign.net. To view this video or to download it, go to the workshops page and follow the link under webcasts in the right-hand column.

Dr. Lintern covers three issues in this presentation. First, he considers the question of why complex socio-technical systems work and points out that there are a number of technological myths about this issue. He contrasts those myths with several technological realities. He then reviews a few remarkable success stories and draws lessons from them about how we should go about designing a robust socio-technical system. He emphasizes that the resulting system must be an effective cognitive system at the individual, team and organizational levels. Finally, he outlines the basic strategy of cognitive design and suggests that we get ourselves into trouble in this area partly because of a language problem. The language we commonly allow for technological artifacts implies functionality that those artifacts do not have, and leads to the neglect of potent functionality that human participants inevitably contribute to the operation of the system.

Also available on the site is a video by Dr. Lintern on Cognitive Performance Assessment.

How well do technical systems support the human cognition essential to the performance of those systems? This is a 30-minute tutorial in which Dr. Lintern leverages from some work on cognitive indicators from the naturalistic decision-making area to develop system evaluation tools based on the abstraction-decomposition space of work domain analysis and the decision ladder of work task analysis.

www.ppi-int.com Goes Mobile

The PPI website is now mobile-friendly - users can visit our website via their mobile device to get up-to-date information on our latest course deliveries, news, view our monthly newsletter SyEN and more.

PPI Events (see www.ppi-int.com)

Systems Engineering Public 5-Day Courses:

Upcoming locations include:

- Melbourne, Australia
- Las Vegas, USA
- London, UK
- Sydney, Australia

Requirements Analysis and Specification Writing Public Courses:

Upcoming locations include:

- Melbourne, Australia
- Amsterdam, The Netherlands
- Adelaide, Australia
- Austin, USA
- Stellenbosch, South Africa

Systems Engineering Management 5-Day Courses:

Upcoming locations include:

- London, UK
- Stellenbosch, South Africa

Software Engineering Public 5-Day Courses:

Upcoming locations include:

- Sydney, Australia
- Pretoria, South Africa
- Amsterdam, The Netherlands

Requirements Engineering 5-Day Courses:

Upcoming locations include:

- São José dos Campos, Brasil
- Buenos Aires, Argentina
- Santiago, Chile

OCD/CONOPS Public Courses:

Upcoming locations include:

- Melbourne, Australia
- Pretoria, South Africa
- Las Vegas, USA
- Brasilia, Brasil

Cognitive Systems Engineering Courses:

Upcoming locations include:

- Adelaide, Australia
- Las Vegas, USA

CSEP Preparation Course (Presented by PPI subsidiary Certification Training International)

Upcoming locations include:

- San Diego, USA
- Amsterdam, The Netherlands
- San Jose, USA
- Las Vegas, USA

PPI Upcoming Participation in Professional Conferences:

PPI will be participating in the following upcoming events. We look forward to chatting with you there.

- Pacific 2012 IMC | Participating | Sydney, Australia (31 January – 2 February 2012)
- Singapore Airshow | Exhibiting | (14-19 February 2012)
- SETE/APCOSE 2012 | Exhibiting | Brisbane (30 April – 2 May 2012)
- INCOSE IS 2012 | Exhibiting | Rome, Italy (9 - 12 July, 2012)

Kind regards from the SyEN team:

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

Raloh Youna, Editor, email: rvouna@ppi-int.com

Steph Halligan, Production, email: shalligan@ppi-int.com

Project Performance International
2 Parkgate Drive, Ringwood, Vic 3134 Australia
Tel: +61 3 9876 7345
Fax: +61 3 9876 2664
Tel Brasil: +55 11 3230 8256
Tel UK: +44 20 3286 1995
Tel USA: +1 888 772 5174
Web: <http://www.ppi-int.com>
Email: contact@ppi-int.com

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

Tell us what you think of SyEN: email to contact@ppi-int.com

If you do not wish to receive a monthly copy of SyEN in future, please reply to this e-mail with "SyEn" in the subject line. All requests are acknowledged and actioned; you may wish to contact PPI if acknowledgement is not received within 7 days.