PROJECT PERFORMANCE INTERNATIONAL

CERTIFICATION TRAINING INTERNATIONAL
A PPI Company

SYSTEMS ENGINEERING NEWSLETTER

PPI SyEN 86 – 28 February 2020

Dedicated to inspiring and improving the practice of systems engineering

brought to you by

Project Performance International (PPI)

Systems engineering training and consulting for project success

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

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

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

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

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

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

Should you no longer wish to receive SyEN for any reason, simply unsubscribe by clicking the link at the bottom of this email.

IN THIS EDITION

1. Quotations to Open On

Read More…

2. Feature Articles

2.1 An Issue in Systems Integration: A Shortage of Timely Stitches *by James R. Armstrong*

Read More…

3. Additional Article

3.1 Systems Engineering Principles *by Project Performance International*

3.2 Adapting DevOps Processes to System Engineering *by Len Bass*

3.3 Whither Goest Systems Engineering? *by Ralph Young*

Read More…

4. Systems Engineering News

4.1 Gary Roedler Provides Review of INCOSE Activities

4.2 Special Issue Planned for the January/February 2021 *IEEE Intelligent Systems Journal* concerning Learning Complex Couplings and Interactions

4.3 IISE Launches New International Competition "The IISE Cup"

4.4 Job Opportunity: Senior Systems Engineer at Rolls Royce

4.5 Job Opportunity: Lead Engineer Systems Engineering with GE Aviation

4.6 Jama Software On-Demand Webinar: The High Cost of Poor Requirements Management

4.7 INCOSE UK Annual Systems Engineering Conference Call for Content

4.8 INCOSE Recognition Nominations: Deadline Extended Submissions for INCOSE Recognition Awards have been extended to 15 March 2020

4.9 Launch of the INCOSE Systems Engineering Tools Database Working Group

4.10 Launch of the INCOSE Social Systems Working Group

Read More…

5. Featured Organizations

5.1 INCOSE Institute for Technical Leadership

5.2 French Chapter of INCOSE: Association Française d'Ingénierie Système (AFIS)

Read More…

6. News on Software Tools Supporting Systems Engineering

6.1 Tom Sawyer Announces Perspectives 9.0

6.2 Modelio 3.8.1 Released

Read More…

7. Systems Engineering Publications

7.1 Systems Engineering: Reliability Analysis using k-out-of-n Structures 7.2 International Journal on Software and Systems Modeling (SoSyM)

7.2 Ten Requirements Traps to Avoid

7.3 Deployment and Operations for Software Engineers

7.4 Engineering Systems Integration

7.5 Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering

7.6 Joint Cognitive Systems: Patterns in Cognitive Systems Engineering

7.7 INSIGHT December 2019 – Volume 22: Issue 4 Released

7.8 Detecting Incoherent Requirements in Large Documents

7.9 Swipe to Unlock: The Primer on Technology and Business Strategy

Read More…

8. Education and Academia

8.1 UC Berkeley's Online Masters in Cybersecurity

8.2 Institute for Advanced Systems Engineering

Read More…

9. Some Systems Engineering-Relevant Websites

Read More…

10. Standards and Guides

10.1 AV Safety Standards Set to Arrive in 2020

10.2 System Life Cycle Processes Per ISO/IEC/IEE 15288 (2015)

10.3 The IEEE Software & Systems Engineering Standards Committee (S2ESC)

Read More…

11. Some Definitions to Close On

11.1 Logistics Support

11.2 Measures of Effectiveness

11.3 Mode

Read More…

12. Conferences and Meetings

12.1 The Annual Conference on Systems Engineering Research: March 19 - 21, 2020 – Redondo Beach (USA)

12.2 Host a Society of Women Engineers (WE) Event

Read More…

13. PPI and CTI News

Read More…

14. PPI and CTI Events

Read More…

15. Upcoming PPI Participation in Professional Conferences

Read More…

1. QUOTATIONS TO OPEN ON

“Only two things must be done in systems engineering practice: physical design and system integration. Everything else is discretionary - we decide whether to do it, if so, how much of it to do, and how to best do it to that degree.”

Robert John Halligan

‘“Leadership is an action, not a position.”

Donald H. McGannon

“Imagine a world in which every single person on the planet has free access to the sum of all human knowledge.*”*

Jimmy Wales, Founder of Wikipedia

“Do not communicate by sharing memory; instead, share memory by communicating.”

R. Pike

2. FEATURE ARTICLE

2.1 An Issue in Systems Integration: A Shortage of Timely Stitches

by

James R. Armstrong

Stevens Institute of Technology

Email: jimarmstrong29@aol.com

Editor's Note: Jim Armstrong has worked tirelessly to further, strengthen and improve the integration process. He has contributed articles previously for PPI SyEN: 1) in March, 2018, "The Impact of Integration on Reliability"; and, 2) in November, 2018, "Divergent Thinking in Systems Engineering Practice: Is There a Shortfall?" Both articles were thoughtful treatments and were well-received by subscribers. As Jim has stressed, "There are actions we can take as systems engineering practitioners, educators, and discipline definers to address integration issues. We must treat system integration as more than the assembly of components. And we must recognize that systems integration takes place throughout the entire life cycle from first concept to final disposal. In doing these two things, we can start a broader consideration of what integration should address through inclusion in early analysis, modeling, and trades related to a wider range of systems concerns".

Abstract

Most discussions on integration focus on the activities during the building of the product. Early actions are limited to defining and controlling the interface requirements. With the reality of the increased costs of fixing problems late in the program and the general recognition that there are usually problems at the interfaces, this seems to be a less than optimum approach. This paper looks at several ways to improve the overall integration process and find those problems much earlier. Examples are given of both need for earlier actions and their benefit.

I. Introduction

'A stitch in time saves nine.' – Proverb

Over the last three decades, I've posed a question to various groups of engineers asking whether anyone in the group has been part of a project where a hole has to be cut in a wall to install the system at deployment. If the hole was pre-planned, it doesn't count. No group of ten or more has failed to come up with a personal example. Further, in preparing for a discussion on early integration, I attempted to find an article from a few years ago about a Florida fire station that ordered a new fire engine that ended up being too large to fit in the facility. The solution was to add 10 feet to the front of the facility. After the extension was completed, there was insufficient clearance at the entrance to turn in or out of the area.

That article was not found; however, 11 other cases came up where the new fire engine was too big, too heavy, didn't fit in the streets in the town, or had similar problems. In each of these cases, the problem was discovered as a surprise upon delivery. Even when requirements were established, errors were made that showed up late. For instance, one group did measure the doorway height, but missed the fact that the door did not raise up to the full height of the doorway. At Boston's Boylston station, Figure 1, the new Engine 33 met the height requirements, but its width made it too tight a fit for the arched doorway (Armstrong, 2010).

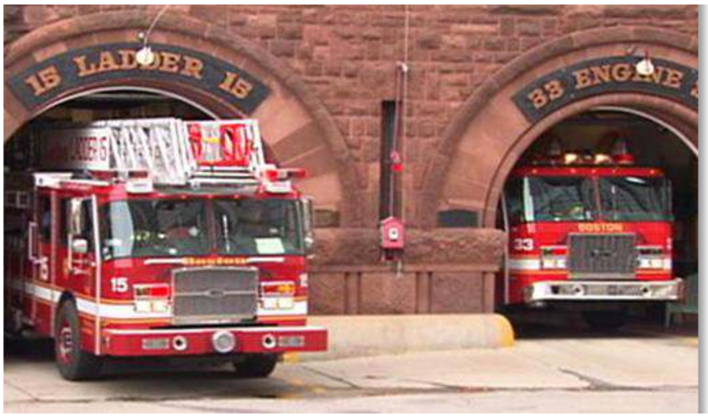


Figure 1: Arched Doors in Boston

It's easy to say this is obviously a problem due to people not accustomed to developing systems and integrating new products into a legacy system making the acquisitions. However, there are numerous examples in major programs with people who are supposed to know better in charge that are equally as preventable. The cost of changes to fix problems has been well known for a long time to increase significantly and exponentially as the program progresses. Finding problems late in the assembly sequence or during transition is easily an order of magnitude more expensive than prior to detail design. It would seem to make sense to find ways to prevent or at least detect more end-of-program problems earlier – timely stitches.

II. Guidance or Misguidance?

Do the various standards, handbooks, and texts provide the best guidance in addressing integration? Or, are they part of the problem? Even the best intentions may be guiding readers in a direction that results in problems when executed as presented.

Standards: ISO/IEC 15288 addresses integration as an assembly activity and has a separate process for transition that covers deployment activities. IEEE 1220 only addresses integration as the assembly of fabricated components and notes it as a late program activity following design. EIA 632 buries a minor mention of integration into the implementation process.

Textbooks: A review of the integration content of several well known texts finds that the treatment of integration is consistent with the standards and mostly silent on early activities, save development of interface requirements. In Blanchard and Fabrycky (2006), integration is conspicuously missing with the exception of integration of engineering disciplines and efforts. Buede (2000) clearly places integration in the right side of the V and limits it to the assembly of components. Kossiakoff and Sweet (2003) identify integration as part of a separate phase of development following design. Martin (1996) limits mention to a definition of integration as the assembly of components. Stevens, et al (1998) wraps integration and verification and shows them as the right side of the V. Installation is also addressed to complete the integration into the user environment. Larson, et al (2009) devotes a chapter to integration and includes a figure with a recommendation that integration have a long buildup overlapping design. However, specifics on what should be done during the overlap are not provided and the description of integration activities is limited to assembly of components. Grady (1994) comes the closest to addressing the concerns of this paper and does identify early integration related activities during requirements development and architecture phases.

Handbook: The current INCOSE Systems Engineering Handbook provides a context diagram for the integration process as shown in Figure 2. Interface requirements are mentioned in the requirements process and provided by the architecture process. The integration process does include the generation of a strategy and plan but its focus is on timely assembly of components. The Perform Integration activity only addresses that assembly. The handbook references ISO/IEC 15288:2008 which again focuses on the assembly of components. There is a Transition Process that separately addresses the integration of the system into the operational environment, also consistent with ISO/IEC 15288.

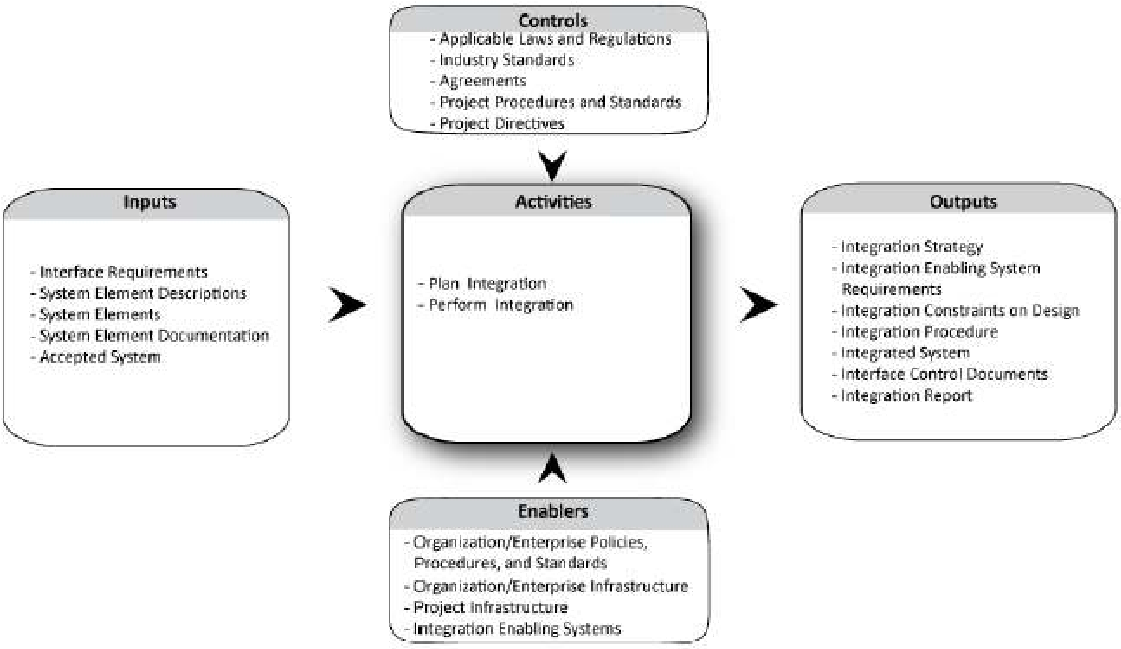

Controls
- Applicable Laws and Regulations
- Industry Standards
- Agreements
- Project Procedures and Standards
- Project Directives
Inputs
- Interface Requirements
- System Element Descriptions
- System Elements
- System Element Documentation
- Accepted System
Activities
- Plan Integration
- Perform Integration
Outputs
- Integration Strategy
- Integration Enabling System Requirements
- Integration Constraints on Design
- Integration Procedure
- Integrated System
- Interface Control Documents
- Integration Report
Enablers
- Organization/Enterprise Policies, Procedures, and Standards
- Organization/Enterprise Infrastructure
- Project Infrastructure
- Integration Enabling Systems

Figure 2: INCOSE SE Handbook Integration Process Context Diagram

Maturity Models One influence in the organizational processes on many companies has been the CMMI®. Although the initial versions kept the source models' initial practice of developing an integration strategy, version 2.0 of the CMMI changed to an initial practice of defining the assembly sequence. The impact is that processes developed in response to this version of the model focus on the actual assembly and skip the early integration opportunities. The difference is that a strategy focus defines the integration risks and defines mitigations to address them, including early activities well ahead of actual manufacturing of product. The latest revision has corrected this issue and returned the strategy to the first practice.

The "V" and Waterfall: The saying is that all models are wrong but some are useful. The "V" and waterfall models of systems development are useful in depicting the overall progress from problem to solution. However, they can also be misleading. The depiction in Figure 3 shows integration as part of the “right side of the V” and being fully placed after detail design has been completed. The waterfall unfolds the V but, in its basic form, still depicts integration as an activity occurring after detail design and component build. The rows behind the waterfall were added to indicate that all activities run in parallel throughout the development process although with more emphasis at various times (Armstrong, 1988). The “V” model prevents this depiction and leads to misconceptions of a sequenced process with each activity performed only once and minimal parallel activity. The key is to recognize that there are benefits to doing risk reduction activities on the left side to avoid late discovery and increased cost of fixes at the end of the program. The highlighted bar indicates this continuous performance of integration.

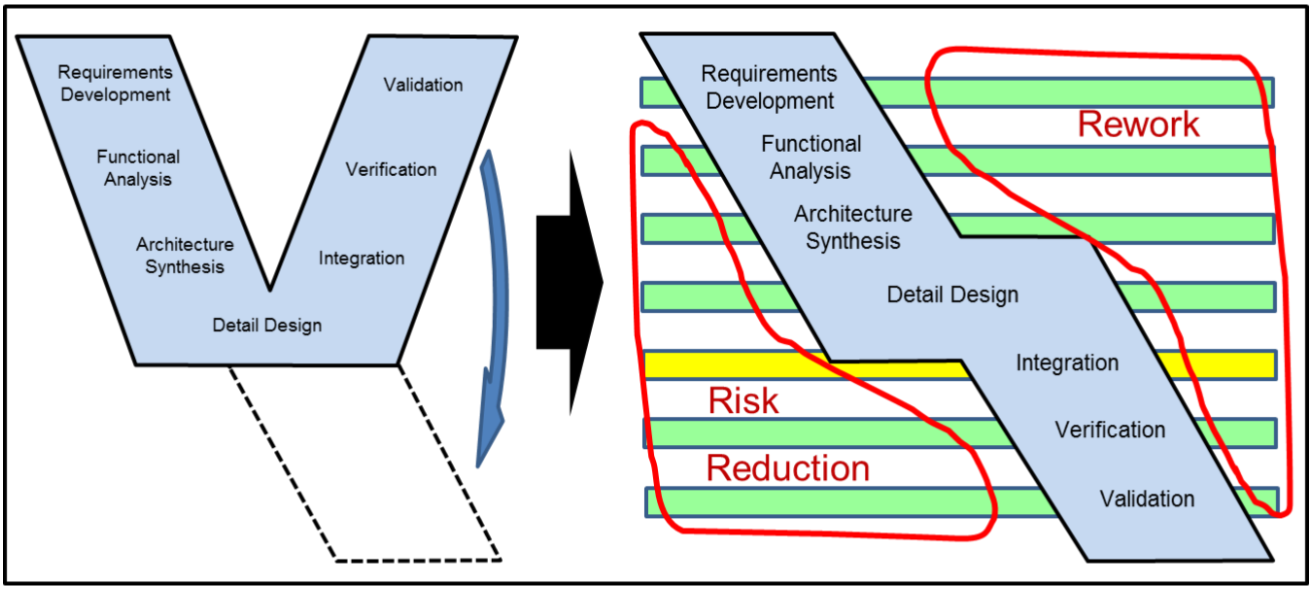


Figure 3: The “V” model unwrapped to a waterfall with continuous activities shown.

III. Better Option – Early Action

There are several early actions that can be taken to discover and avoid integration problems. A common theme is that they provide specific integration emphasis during the processes that are already defined in the “front side” of the V model. They include attention to Conway’s Law, recognition that interface information appears in places other than ICDs, use of models and thread analysis for early integration, more integration concern in developing architectures, better recognition of hidden allocations, and early concern for life-cycle integration concerns such as deployment and maintenance.

Risk Based Strategy. One of the key elements in the revised CMMI® practice is the identification of integration risks and developing an overall strategy that works to mitigate these risks in a timely fashion. The principal difference here is not waiting until the assembly step that might discover the problem but taking action early to reduce the cost of fixing it. One example was a program that was building a large satellite in California to be launched in Florida. Around the time of Preliminary Design Review, a full-scale wooden version of the shipping container was built and loaded into the cargo plane that would carry it to the launch facility. Someone had identified the tight fit as a potential problem that would be painful to discover at time of transport. One can debate whether this is the right mitigation or note that there are better ways to address it today. However, it does represent the basic difference between a strategy that looks for early risk mitigation and waiting for the assembly sequence to uncover issues.

Conway's Law. In his 1968 paper "How Do Committees Invent?" Melvin Conway posits that the architecture of the system is tied to the architecture of the organization that produces it. The organization of the design team is in itself a constraint on the design decisions in both structure and design decisions. He further discusses the relationship of the communications within the organization and the communications or interfaces within the system. At first this may sound strange. But what is the likelihood that a structure with two design groups for a transmitter and a receiver will produce a single transceiver module?

With regard to integration, this raises the need to integrate the design team architecture early in the development of a new system. Potential communications problems are very likely to end up as integration issues later. The Mars Climate Orbiter error in two different sets of units being used can also be mapped to two different organizations developing the space and ground components. In a pair of tactical telephone switches being developed by two different companies under contracts with two different services, the small detail of whether the parity check would be odd or even was discovered in field testing. Both were seemingly simple mistakes but are the type that gets lost in the overall communications between organizational units. The analogy to herding cats is very applicable to being an integrator on a large system. Recognition of the impact of Conway's Law can be very helpful in establishing an organization and communications process that will reduce the integration risks.

Requirements Process. The identification of interface requirements in the form of Interface Specifications or Interface Control Documents (ICDs) is mentioned in the INCOSE handbook. Also noted is the identification of operational constraints from other systems. However, a more focused emphasis in this phase is warranted. The initial emphasis should be on external interfaces of all types including enabling systems. Some of these concerns will be addressed later in this paper.

One other impact on the requirements process that must be remembered is that interface requirements are not limited to the content of ICDs. The environment section of a specification also provides the definition of the interface with nature. The human interface section is also not an ICD. Outside of the specification, there may be interface requirements or constraints in memorandums of understanding, teaming agreements, service agreements, operational concepts, policies, legal requirements, or other sources. All of these must be identified and addressed in the integration strategy.

Architecting Process. There are several integration issues that can be addressed at the architecture level of abstraction. The first is looking at the interfaces that the architecture is creating and identifying the risks involved. Modeling and analysis can often determine whether a given architecture is likely to come together and perform as needed. In another article in this issue of PPI SyEN, Bass (2020) discusses the benefit of a microservice architecture to reduce integration risks that can delay large quantity and rapid deployment of software changes.

N2 diagrams. A basic approach to depicting the interactions between system elements is the N2 diagram. Figure 4 shows how this diagram can be used to compare the interactions between two alternative solutions that allocate functionality to different physical elements. The example is quite basic. It shows what might be the result if functions 4 and 5 can be kept in a central computer (option 1) or handled in remote processors (option 2), perhaps multiple smart actuators to take advantage of their computational ability. The potential difference in interface complexity is evident in the diagrams. For software integration, a similar approach called Design Structure Matrices is used to improve the placement of functionality in modules.

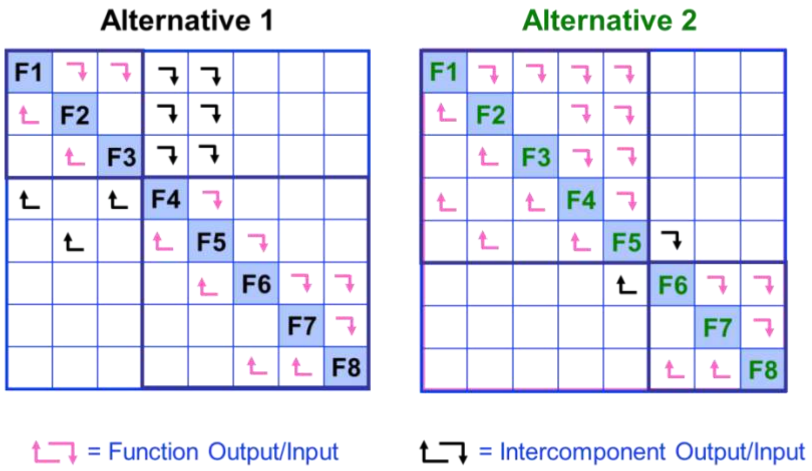


Figure 4: N2 Example problem

Cross-Branch integration. Integration problems across branches of the hierarchy are part of the common complaint about the perceived systems engineering process. In its most basic form, the process calls for a top down flow of requirements with allocation of derived details to smaller parts of the system. The individual parts are independently built and assembled with the implicit assumption that all the allocations were perfect. Whether the interfaces are external or internal, the problem is that there are often interfaces that do not show up until late in the integration sequence if a strict bottom up sequence is adhered to.

This was part of the problem with the two tactical telephone switches mentioned previously in the discussion of Conway's Law. Since each element of the overall tactical communications system was developed separately by different acquisition groups and at different times, the first time they came together was at the systems test range at Fort Huachuca, AZ. Figure 5 shows this in the typical wishbone integration sequence diagram.

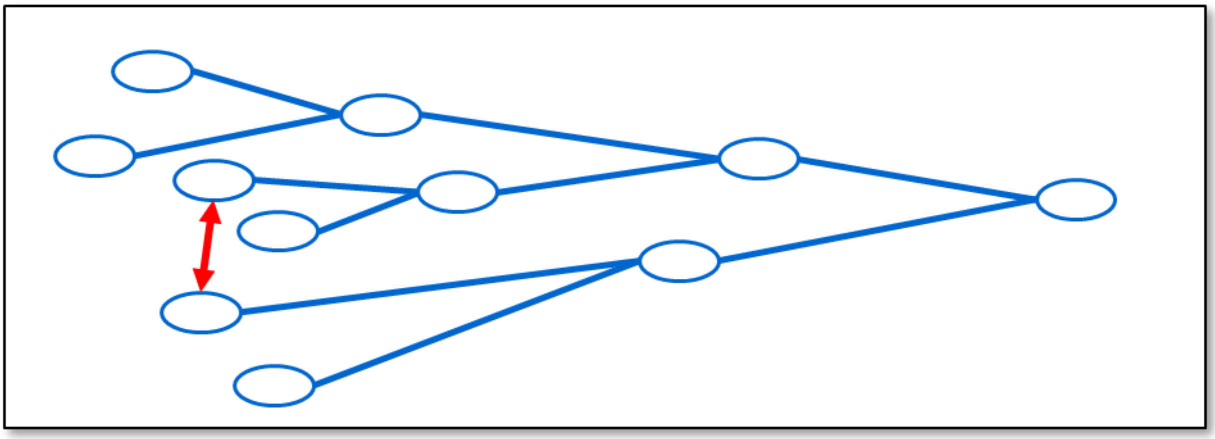


Figure 5: Cross-Branch Integration Opportunities

These are the type of interfaces that will not show up as problems until late in the program. There are often ways that they can be verified early in the process by arranging to integrate the component or subsystems directly at an early point in their development. In the case of the switches, communications could have been checked through remote links or taking a partially functioning component to a test site much earlier. Would it have been worth it for this one issue? Not necessarily. However, the consideration of this type of early, lower level integration activity should be part of the discussion in developing the integration strategy.

External Interfaces. One of the most common integration errors is the focus on internal interfaces and the late or missing treatment of external interfaces. Engineers seem to have a natural tendency to put blinders on and limit their field of view to the scope of the problem put before them. One example is a study that was done to look at potential future upgrades to the US Air Force tactical command and control system. The intent was to look twenty years into the future and try to identify technologies that should be investigated to be prepared for a possible upgrade. After a long technical briefing to headquarters staff, a Wing Commander Colonel in the audience asked the briefer, “Where is the output?” The answer was, “The entire combat picture is in front of the General here in the command post.” The Colonel was not happy with that answer and repeated the question, with additional emphasis. The briefer had to be directed to a prior chart that showed a small airplane in the corner that would receive the output. The briefer had been so involved in the internal workings of the command and control system that he had entirely forgotten its purpose.

Models and Integration. One solution to this mindset is to not limit analysis and understanding of external interfaces to just the inputs and outputs. The external systems that are interacted with need to be specified and their behavior with the outputs and generating the inputs understood. One example to show this in a simple diagram is provided in Figure 6. To merely say the ATM inputs requests and outputs money is a very limiting view. The model in the figure uncovers a user expectation to retry which the ATM model does not address.

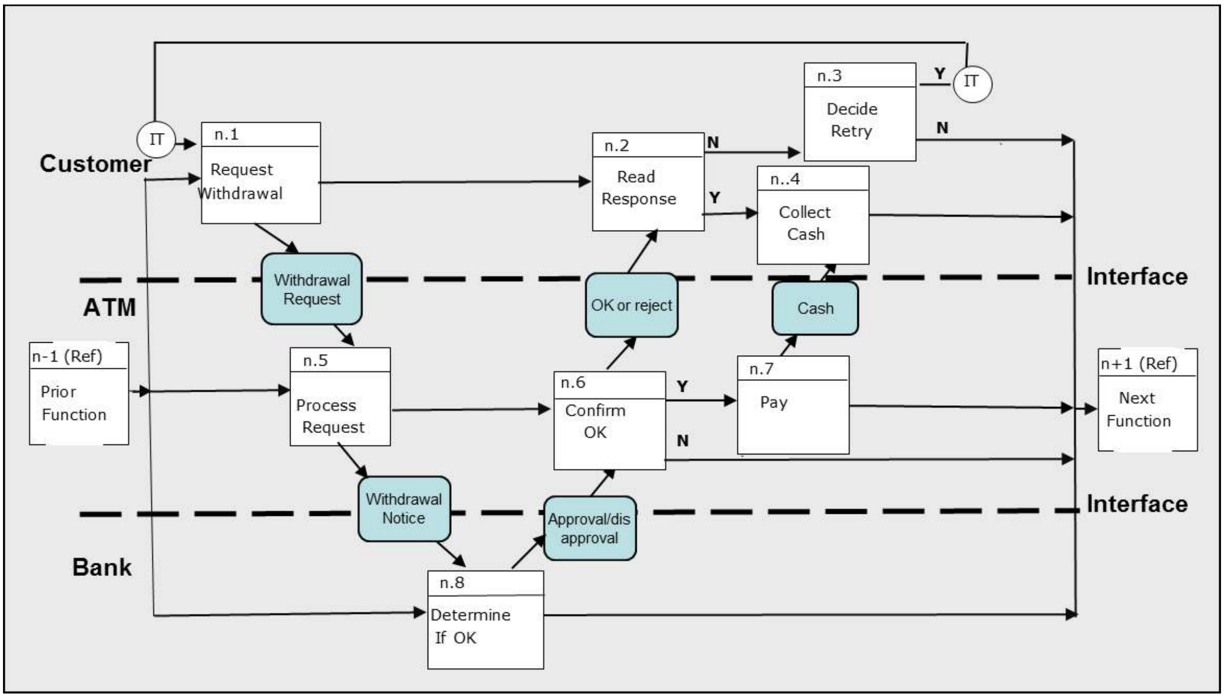


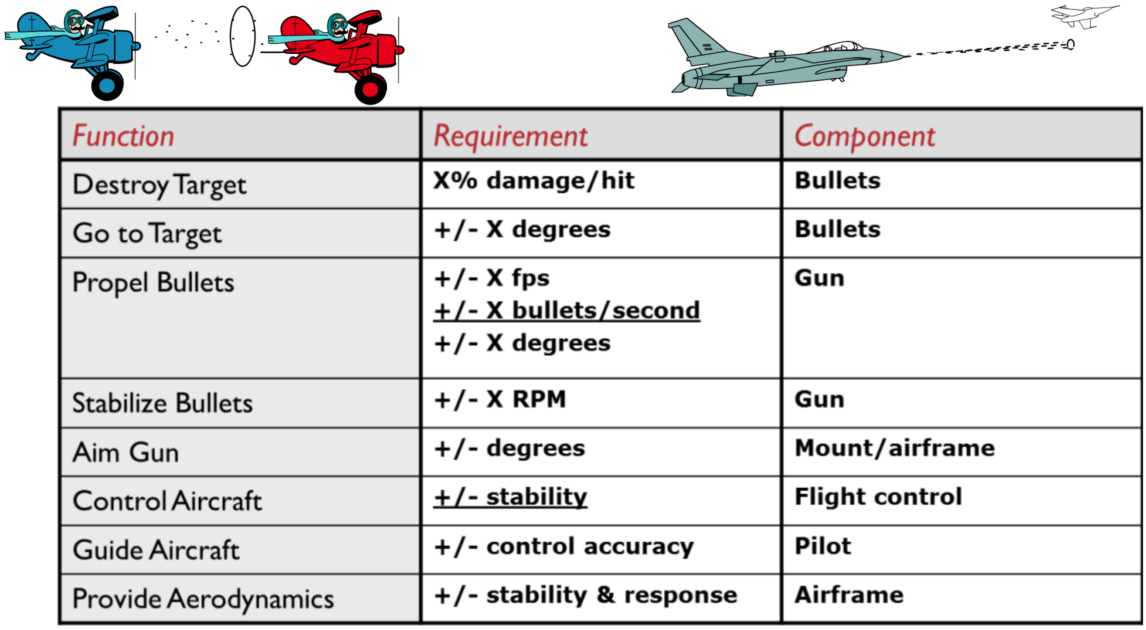
Figure 6: Integration with Models

In addition to the functionality shown here, the systems engineer should also have an understanding of who the potential customers are and the circumstances under which they will be using the system. In the ATM case, is it a single user alone? Is it a shopper with kids in tow? Is the ATM indoor or outside? If outside, what will the weather be? Understanding this will provide a solution better integrated into the users' environment.

There are two aspects of models such as the one shown above and integration. The first is the use of models to do integration as described. The second is the integration of the models. If the models developed by different system element teams do not work together, it is likely the products produced won't either.

Threads. One defense against cross-branch issues is the use of threads to model and integrate functions across the system. In this approach, the parts that support a key function can be integrated through physical, simulated, or mixed modes at early stages to learn how those critical functions are working. A simple example for the command and control site is simply the function of placing or receiving a phone call. The full complement of components need not be created. Only those that are involved in the function of interest need be activated. Similar integration is used in the avionics arena when the software that drives actuators is paired with actual hardware and simulated loads. The entire aircraft is not needed to learn at least the basics of how the software and the hardware interact.

Allocations. One risk during architecture and design is the hidden or assumed allocation. Designers often focus on individual components and their optimization without recognizing the impact on the top-level objective. One example is the reported instance of the F-16 not being able to hit targets in the initial air-to-air tests. The issue of how to get a spread of bullets similar to a shotgun, but not too much spread, has been a challenge since air-to-air combat started. The spread of bullets was primarily a combination of the movement of the aircraft and the time between bullets leaving the gun. On the F-16, the avionics smoothed out the flight and the Gatling guns reduced the time between bullets. The result was too little spread in the pattern. The initial fix was to add a shaker to the gun and later modify the mount to be more flexible. Figure 7 shows how the allocation might have been developed had systems engineering done so.



Function	Requirement	Component
Destroy Target	X% damage/hit	Bullets
Go to Target	+/- X degrees	Bullets
Propel Bullets	+/- X fps +/- X bullets/second +/- X degrees	Gun
Stabilize Bullets	+/- X RPM	Gun
Aim Gun	+/- degrees	Mount/airframe
Control Aircraft	+/- stability	Flight control
Guide Aircraft	+/- control accuracy	Pilot
Provide Aerodynamics	+/- stability & response	Airframe

Figure 7: Undefined Allocations Example

The reality is that this reported issue has not been validated as factual. However, in discussing this as a type of example, others have come forward as having seen the same or similar issues in the F-104, A-10 (reverse problem), Huey gunship, and the B-52 tail gun. While this seems to be a modern technology issue, a history buff noted that the Romans had a similar problem with a catapult that was intended to launch a large sheaf of arrows at once. The problem being that they all came down together.

One program that did address this and other factors that affect overall performance was the Comanche helicopter which modeled the systems performance to also include the difference in early or late mission, age of airframe, pilot variation, etc. In that way, individual component improvements could be analyzed for their impact on the integrated system performance.

Human Interface. In explaining the downing of Iran Air 655, a Washington Post headline stated that the system worked but the operator made a mistake. The problem with that comment is that the operator is part of the system. Most definitions of systems integration refer only to the assembly of hardware and software components. This leads to reduced or partial consideration of the human interface. Engineers often ignore or minimize their consideration of functions and requirements allocated to the user. Several efforts have been made to change this behavior and multiple techniques exist to address the human interface. One way to increase consideration is to ensure that the human component appears in early architectural diagrams.

Maintenance Interfaces. There are at least two considerations for maintenance impact that should be included as part of integration. The easy one is the interface with test equipment. The more often forgotten is the actual consideration of integration of the maintenance activity. One such issue occurred with an air traffic control radar that the Air Force purchased. At the same time as the new equipment was being purchased by a project office in Boston, the facilities were being upgraded by an office in Sacramento. The new upgrade replaced maps on the wall, phones hanging from the ceiling, etc. with an integrated console. The radar scope fit nicely in the new console. However, when it came time to do maintenance, the large multi-chip board on top of the scope hit the console when it was swung up for access to the bottom side. At first, this could be chalked up to Conway's law. However, a check of the new scopes in the existing mobile radar van revealed the same problem in a well-known and unchanged environment. Similar stories of difficult or impossible maintenance are easy to find. Many are available in the automotive sector where engines have to be dropped for simple access to oil filters or spark plugs.

The answer is that maintenance actions must also be addressed in early integration activities and maintenance interfaces included in the interface requirements. Use of 3-D modeling tools for maintenance access is another part of the early integration approach to this risk.

Mische's fourth level. Mische (1998) identifies four levels of integration. The first three are commonly used. They are interconnection of the components, interoperability of the components, and semantic consistency across the interface. Even the most experienced engineers sometimes miss this basic approach of starting with the physical connection level and working up as we troubleshoot. It should not be a surprise that the help desk's first question is whether or not the device is plugged in. In assisting the planning for the installation of equipment in an underground bunker, one of the first questions to the lead of each system was whether they knew where in the facility it would be plugged in. One system manager answered that the signal wire was located at a specific location but had not considered a basic power source. At the other end, the question of semantics is generally recognized to be a common problem in systems talking to each other.

While these are often recognized and respected issues, the fourth level of convergent operation addresses the more complex issue of whether or not the system actually works for the user. Does it work with their business processes? Does it fit with their human resources? Does it do the job? This is a harder question to answer because it is not a collection of technical parameters that can be verified and easily measured. However, it is probably the most important aspect of integration and should be a focus of the integration strategy. Many of the customer and environment awareness techniques associated with validation can be applied in answering this question. Site presence early in the project is a particularly useful tool.

Management Actions. The common approaches from a management view for early systems integration are to have an active interface control working group or systems engineering integration team. These are effective and work much better than another common approach which is to have the leads of separate efforts coordinate among themselves.

Configuration Management. One specific process issue that management should look at is the change process within configuration management. In reviewing these processes during multiple process appraisals, it was very common to find that the process was written assuming that the change would be internally generated and approved. When asked how a change with an external interface would be handled, it was often found not to be addressed. The fix is relatively easy. It just has to be considered.

Reviews. Another concern is the manner in which interfaces are addressed in reviews. If the reviews, particularly Preliminary and Critical Design Reviews, are held independently at the component level, the result may be that each component is reviewed for its ability to meet interface requirements but both sides are not looked at together. In the early days of the Defense Support Program satellite system, this was recognized as an issue. Even though the space segment had performed system level reviews, the ground operational segment direction had been initiated later and the two schedules were somewhat out of sync. A decision was made to add a system-level integration design review. All parties involved with the space and ground segments met for a week. The focus was not on whether a given system element did what it was supposed to do but rather whether interfacing segments worked together. Not every system needs this extra review. However, every system should give serious consideration to where and when such discussions with all sides of an interaction present take place.

IV. Summary

The common interpretation of systems integration as an assembly task that begins after design is completed can lead to significant issues late in the program that could and should have been resolved much earlier. There are methods that can be used to reduce this problem. The discussion above provides a number of concerns that should be addressed early in the program life cycle. The critical change from the usual discussion of integration is to see systems integration as starting on day one. The first step is to develop a risk-based strategy and approach that will take early actions to reduce the most serious integration risks. In developing this strategy and approach, the concerns discussed above should be addressed. If this is done, the program will likely see better results with lower costs to budget, schedule, and performance. We may be able to bring a halt to the parade of lessons relearned and not continuously have to cut holes in the wall to install or other late-program discovery of costly integration issues.

List of Acronyms Used in this Paper

Acronym	Explanation
ATM	Automated Teller Machine
CMMI	Capability Maturity Model Integration
EIA	Electronic Industries Alliance
ICD	Interface Control Document or Drawing
IEC	International Electrotechnical Commission
INCOSE	International Council on Systems Engineering
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
US	United States

References

Armstrong, J, New Fire Truck Too Big for the Boylston St. Station, WBZ-TV, http://boston.cbslocal.com/2010/11/15/new-bpd-fire-truck-too-big-for-the-boyston-st-station/, November 15, 2010.

Armstrong, J. R., *Integrating the Stovepipes Course*. Defense Systems Management School, Ft. Belvoir, VA. 1988.

Armstrong, J. R., *Systems Integration: He Who Hesitates Is Lost.* INCOSE International Symposium, Henderson, NV. 2014.

Bass, Len. "Adapting DevOps Processes to Systems Engineering." *Project Performance International Systems Engineering Newsletter*, February 2020.

Blanchard, B. S. and Fabrycky, W. J., *Systems Engineering and Analysis*, Pearson Prentice Hall, Upper Saddle River, NJ, 2006.

Buede, D. M., *The Engineering Design of Systems, Models, and Methods*, John Wiley & Sons, Inc., 2000.

Capability Maturity Model Integration, Development, Version 1.2, Carnegie Mellon University, August, 2006.

EIA 632, *Process for Engineering a System,* Coordination Draft, Electronic Industries Alliance, Arlington, VA, May 1998.

Grady, Jeffrey O., *System Integration*, CRC Press, Boca Raton, FL, 2020.

Grady, Jeffrey O. *System Engineering Planning and Enterprise Identity*, CRC Press, Boca Raton, FL, 1995.

Haskins, C., ed. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities.* Version 3.1. Revised by K. Forsberg and M. Krueger. San Diego, CA (US): INCOSE, 2007.

Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. Version 3.2. Revised by M. Krueger, D. Walden, and R. D. Hamelin. San Diego, CA (US): INCOSE, 2010.

IEEE 1220, *Standard for Application and Management of the Systems Engineering Process*, IEEE, 345 East 47th Street, New York, NY 10017, 1999.

ISO/IEC 15288, *Systems engineering – System life cycle processes*, ISO, 1 rue de Varembe, CH-1211 Geneva 20, Switzerland, 2002.

Kossiakoff, A. and Sweet, William M., *Systems Engineering, Principles and Practices*, John Wiley & Sons, Inc., 2003.

Langford, Gary O., *Engineering System Integration: Theory, Metrics, and Methods*, CRC Press, Boca Raton, Florida, USA, 2017.

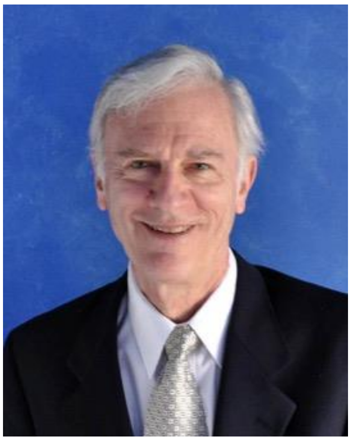
Martin, J., *Systems Engineering Guidebook: a Process for Developing Systems and Products*, CRC Press, Boca Raton, FL, 1996.

Mische, M.A. *Defining Systems Integration*, in M.A. Mische, Editor, *Reengineering Systems Integration Success*, CRC Press, Boca Raton, FL, 1998.

Stevens, R. et al, *Systems Engineering – Coping with Complexity*, Prentice Hall Europe, Hertfordshire, England, 1998.

Summers, Boyd L., *Effective Methods for Software and Systems Integration*, CRC Press, Boca Raton, Florida USA, 2012.

About the Author



Jim Armstrong has practiced systems engineering for 52 years, performing various roles including configuration management, test, deployment, chief engineer, program manager, and program element monitor. For the last 30 years, he taught, consulted, and appraised systems engineering in industry and government. He is currently an Adjunct at Stevens Institute of Technology primarily teaching Systems Integration. Jim was on the author teams for several of the standards and models discussed in this paper. He has a BS in Mechanical Engineering from Rensselaer Polytechnic Institute, an MS in Systems Management from the University of Southern California, and a PhD in Systems Engineering from Stevens Institute of Technology. He has an INCOSE Expert Systems Engineering Professional certification and is also certified in the use of the Myers-Briggs Type Indicator. Jim is active in INCOSE and chairs the Integration, Verification, and Validation Working Group.

3. ADDITIONAL ARTICLES

3.1 Systems Engineering Principles

by

Robert Halligan
Project Performance International

Melbourne, Australia

Email: rhalligan@ppi-int.com

Editor's Note: A concerted effort is ongoing within INCOSE to formalize and publish a set of Systems Engineering Principles. Project Performance International (PPI) has developed a set of Systems Engineering Principles over the past 20 years of training, education, and consulting in the field and discipline of systems engineering and has contributed it to INCOSE's Team that is addressing this area. Following is PPI's set of Systems Engineering Principles:

1. Capture and understand the requirements, measures of effectiveness and goals (the problem) before committing to the solution.

2. Try to ensure that the requirements are consistent with what is predicted to be possible in solutions, at the time of required supply, i.e. are feasible.

3. Treat as goals, desired characteristics that *may* not be feasible, but not at the expense of the requirements. *Note: "treat as goal" means that effort will be expended to achieve the goal which is related to the importance of the goal and the probability of success. Where conflicts between goals exist, the goals will be traded off to maximize overall effectiveness.*

4. Define system requirements, measures of effectiveness, goals and solutions with regard to the whole of the (remaining) life cycle of the system of interest.

5. Maintain a distinction between the statement of the problem and the description of the solution to that problem for the system of interest, and for each subsystem/component/system element of that system. *Note: "Maintain a distinction" means "ensure that each is separately identifiable".*

6. Baseline each statement of the problem (requirements, measures of effectiveness, and goals set) and description of the solution to that problem (design). Control changes to requirements and design.

7. Identify and develop descriptions of solution alternatives (designs) that are both feasible (i.e. can meet requirements) and potentially are the most overall effective. Put aside from further consideration, as potential solutions, all other alternatives (unless the assessment of that potential

solution changes). *Note: MOEs could include development cost, time to market or other measures unrelated to system capabilities.*

8. Develop solution descriptions for enabling systems concurrently, and in balance with, the solution description for the system of interest. *Note: an "enabling system" is a system that enables some phase of the life cycle of the system of interest*. *The internal design of an enabling system must be related to the internal design of the system of interest; they are not independent, each simply against a set of requirements.*

9. Except for simple solutions, develop logical solution descriptions (descriptions of how the system solution is to meet requirements) as an aid to developing physical solution descriptions (descriptions of how to build the system).

10. Be prepared to iterate in design to drive up overall effectiveness, but not at the expense of the requirements. *Note: as stated above, MOEs contributing to overall effectiveness could include development cost, time to market, or other measures unrelated to system capabilities.*

11. Decide between feasible solution alternatives based on evaluation of the overall effectiveness of each of these alternatives. Limit alternatives to be evaluated to those which have potential to be overall the most effective. Take risk and opportunity into consideration in the evaluation.

12. Subject to level of risk, independently verify work products (is the job being done right?). *Note: "independently" in this context means "not by the developer of the work product".*

13. Subject to level of risk, validate work products from the perspective of the stakeholders whom the work products serve (is the right job being done?).

14. The act of managing is needed to plan and implement the effective and efficient transformation of requirements and goals into solution descriptions and solutions – this is unlikely to *just happen*

3.2 Adapting DevOps Processes to System Engineering

by

Len Bass

Carnegie Mellon University

Email: lenbass@cmu.edu

Abstract

More and more, the software portions of systems are being updated frequently. Automobiles are having updates sent over the internet. Satellites are having updates sent while they are in orbit. These updates can happen at arbitrary times and, potentially, without concern for the state of the system being updated. Systems designed for frequent updates utilize digital networks and are not memory limited.

DevOps (a portmanteau of Developer and Operations) is a set of processes designed to speed the placing of software updates into production. Some of these processes are immediately adaptable to

systems, others are constrained by system environmental concerns. In this paper, we discuss DevOps processes based on the use of microservice architecture and how they are applicable to System Engineering. The use of microservice architecture can reduce errors during the integration process and speed up the development of a version of the software being readied for deployment.

Introduction

"DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality" [1]. These practices can be divided into those intended to speed up the deployment process and those intended to speed up incident handling after an initial deployment.

DevOps as a technology area is about a decade old and its practices have been widely adopted by software development organizations [3]. DevOps practices are driven by a desire to reduce the time to market for new features or for modifying existing features. Traditionally, new versions of a software system were scheduled into releases. One new release per month/per quarter/or per six months was a common schedule. In the internet world, these times are too slow and statistics such as hundreds or thousands of deployments per day are reported for companies such as Netflix, Amazon, and Facebook [4].

The Systems Engineering world is not immune to the time to market pressures and for this community, the questions have become: which DevOps practices are appropriate for systems involving specialized hardware and what are the production hardware requirements to support DevOps practices?

We propose that the production hardware must support distribution and network communication. Given those two properties of the hardware, systems can be developed using the microservice architectural style and this style supports both faster deployment and faster incident handling. Such hardware is now common on automobiles as well as other high-end hardware/software systems. AUTOSAR, for example, assumes a collection of ECUs connected by a bus structure [5].

We begin by defining the microservice architectural style [2] and then describe how microservice architectures support some DevOps practices.

Microservice Architectural Style

Around 2002, Amazon promulgated the following rules for their developers. Although the term microservices came later, the core concepts trace back to these rules:

- All teams will henceforth expose their data and functionality through service interfaces.
- Teams [software] must communicate with each other through these interfaces.
- There will be no other form of inter-process communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, and no back-doors whatsoever. The only communication allowed is via service interface calls over the network.
- It doesn't matter what technology the services use.

- All service interfaces, without exception, must be designed from the ground up to be externalizable, ready to be exposed to developers outside of your organization.

The key points for the System Engineering community are that functionality is packaged as services and all communication is through messages between these services. Implicit in the packaging is that services can be deployed onto independent processors.

Now we will see how use of this architectural style supports some DevOps practices.

Reducing Time for Deployment

Continuous Development is the name given to automatically preparing the code for deployment and placing it on a staging server in readiness for deployment.

In either case, use of a microservice architecture allows services to be deployed independently. If the service is a maintenance release, then it can be deployed without consideration of other services. If the service implements functionality for a new feature, then appropriately architecting it will allow it to co-exist with services that have not yet implemented the functionality for that feature [2].

One source of delay in deployment is errors that occur during integration. Errors can be caused by inconsistent choices of technology. Suppose one service uses version 2.3.1 of a library and another service uses version 2.3.2. These two versions may be incompatible. This incompatibility will be discovered during integration. Once it is discovered, it must be resolved. The discovery and resolution of errors takes time.

Since services communicate only via messages, if two independent services make two different choices of technology, not only versions of libraries but also development language, and other technology choices, as long as both services can produce and consume messages, no errors will occur as a result of these choices. Protocol buffers is a name given to a technology that supports message passaging among services using heterogeneous languages.

Reducing Time for Incident Handling

The time for incident handling partially depends on the information available to the incident handler. Since services are small chunks of functionality and communicate via messages, recording the data sent in the messages will provide a wealth of information. Two points are important with respect to the recording of messages:

1. The use of protocol buffers means that the data on entry and exit to a service can be recorded without additional developer work. If every message goes through a protocol buffer on entry and exit from a service, the protocol buffer software can automatically log this information. Thus, if an incident occurs, data is available to recreate the actions of the various services involved.

2. Reserving a server as a repository for logging allows the incident handler to know where to find the information necessary to diagnose the incident. In this case, there should be an agent to move

the log data from the location where the protocol buffer stores it to the repository. This agent can then delete the log data from its source server freeing up resources used for data storage.

Summary and Conclusions

Using DevOps practices makes the frequency of releases a business decision, not a technical one. We have focused on microservices as a means for implementing DevOps practices. Microservice architectures support the reduction of time to deployment, the reduction of errors in deployed systems, and the reduction of time to handle incidents.

Yet the use of microservice architecture is neither necessary nor sufficient to support DevOps practices. Continuous deployment and development can be done using a three-tier architecture [1]. Speeding up incident handling can be done by having specialized teams for incident handling [6] regardless of the underlying architecture. Automated tests speed up deployment time, although for safety critical portions of a system, automated tests may need to be supplemented by manual testing.

The main messages of this paper are that DevOps practices can speed up deployment, that one DevOps practice – the use of microservice architecture – has multiple benefits, and that your organization should not automatically reject the adoption of DevOps practices if time to market is important.

List of Acronyms Used in this Paper

Acronym	Explanation
DevOps	A portmanteau of Developer and Operations.
AUTOSAR	AUTomotive Open System ARchitecture

References

1. Bass, L., Weber, I., and Zhu, L. *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, 2015. ISBN 9780134049847.

2. Bass, L and Klein, J. *Deployment and Operations for Software Engineers*. Independently Published, 2019, ISBN 1096269600.

3. Puppet 2019 State of DevOps Report.

4. Techbeacon https://techbeacon.com/devops/10-companies-killing-it-devops

5. AUTOSAR https://www.autosar.org/

6. Beyer, B., Jones, C., Petoff, J., and Murphy, N.R. (Eds) *Site Reliability Engineering*, O'Reilly Media, ISBN 149192912X.

About the Author

Len Bass has been in the computer business for more than 55 years. He spent 15 years as a Professor at the University of Rhode Island, 25 years as a researcher at the Software Engineering Institute of Carnegie Mellon University, and three years as a researcher at the National ICT, Australia. During his career he has written 8 books and over 100 papers. He is currently an adjunct professor at Carnegie Mellon University where he teaches a course in DevOps.

3.3 Whither Goest Systems Engineering?

by

Dr. Ralph R. Young,

Systems Engineer

Editor, *Project Performance International Systems Engineering Newsletter* (2011 – 2020)

Much has been accomplished world-wide by the systems engineering (SE) community with regard to the SE discipline. A huge number of people and organizations have individually and collaboratively moved SE forward. What are the major initiatives currently underway? How are we doing? What seem to be our major concerns and needs now?

"Traditional Systems Engineering has flourished since the United States first entered the space race and developed Intercontinental Ballistic Missiles. In ensuing decades, more complex systems and/or systems-of-systems have resulted from continuous advancements in the technological world. Industry has been challenged by the constantly evolving needs for managing the growing number of systems and interfaces."[1]

Garry Roedler, outgoing President of INCOSE, provided a comprehensive review of INCOSE's current activities in the December 2019 *INCOSE Newsletter*. Everyone who has had an opportunity to work with Garry knows that he is a very effective leader. His review is included in the Systems Engineering News section of this issue, below. The list of accomplishments and activities underway is impressive and reflects great efforts and credits to INCOSE and its internationally-based members.

Kerry Lunney, incoming President of INCOSE, will provide her vision in an article in an upcoming issue of this Newsletter. We look forward to her exceptional insight and leadership in taking INCOSE forward.

In addition to INCOSE, there are many other societies and organizations worldwide with strong relevance to systems engineering, including among many others, the Institute of Electrical and Electronics Engineers (IEEE), the Systems Engineering Research Center (SERC), the International Institute of Business Analysis (IIBA), the Institute of Industrial and Systems Engineers (IISE), the Systems Dynamics Society (SDS), and the Project Management Institute (PMI).

David Long, president of Vitech Corporation (USA) and past president of INCOSE, wrote recently:

[1] *Advances in Systems Engineering*, 2016, American Institute of Aeronautics and Astronautics.

'Today's challenges continue to grow in complexity, and the pace of change continues to accelerate. Traditional engineering disciplines are evolving to meet these challenges but struggle to do so successfully from within their silos. At the same time, systems engineering seeks to transform itself to model-based approaches, but far more is required. If systems engineering remains as it is today, we are on the short road to irrelevance. It's time to move beyond the art of systems engineering and our traditional base of practice and engineer our place in meeting the grand challenges of the 21st century.'

David has agreed to provide an article for this Newsletter in the mid-year timeframe elaborating on his concern and providing additional suggestions.

INCOSE's Sector Directors have provided leadership for systems engineering deployment throughout the world: Tony Williams, Director, Americas Sector; Paul Schreinemakers, Director, EMTA Sector; and Serge Landry, Director, Asia-Oceania Sector. A terrific report concerning these efforts was provided as an addendum to INCOSE's December 2019 Newsletter, "Focus on International Conferences."

Among our current challenges, one was explored in depth by Eric Rebentisch and many colleagues in *Integrating Program Management and Systems Engineering: Methods, Tools, and Organizational Systems for Improving Performance* (Wiley, 2017, available here). The book elaborated on the concern that improvement is required in integrating program management and systems engineering if systems engineering is to contribute effectively to society. This Newsletter provided a series of articles that summarized each of the chapters of this book, in an effort to reach out to subscribers and enlist support for needed initiatives. A summary of these articles is available here. Unfortunately, interest in, attention to, and actions addressing the concern including needed initiatives have not yet taken hold.

Related to the program integration challenge is the ability to effectively lead complex systems development projects and programs. A lot has been written concerning leadership; yet creating empowered people and organizations remains elusive. A successful approach has been developed by Charles Markert, who facilitates a process called Partnering. Partnering is a proven, powerful process that project and program managers can use to secure the commitment of all stakeholders to help ensure project success. It is also a tool that formalizes otherwise happenstance teamwork and cross organization collaboration. Charlie is developing an article for PPI SyEN that will be shared with subscribers later this year.

Pyster, Hutchinson, and Henry provided a book, *The Paradoxical Mindset of Systems Engineers: Uncommon Minds, Skills, and Careers,* in 2018 (Wiley, available here). It offers unique insights into systems engineers that are well worth exploring and digesting, including how to further strengthen and improve one's effectiveness. The book is a guide to help systems engineers embrace the values that are most important to themselves and their organizations.

Related work on the INCOSE SE Competency Framework is being used to help drive SE professional development. Alan Harding, head of information systems engineering at BAE Systems-Air and a past-president of INCOSE, noted that the latest issue of the INCOSE SE Competency Framework was created in partnership between INCOSE and leading industry and academic partners and covers the full breadth of capability required for today and tomorrow's systems engineers (core, professional, management, technical, and integrating competencies). With its mapping to ISO/IEC/IEEE: 15288, SE processes, and

INCOSE SEP competency areas it is at the heart of INCOSE's evolving professional development offering.

SE has been criticized for not having a set of guiding principles. Project Performance International (PPI), sponsor of this Newsletter, has developed a set of SE Principles over the past 30 years – it is provided as an Article in this issue**.** There is a collaborative team that spans multiple organizations, led by INCOSE that is working to formalize systems engineering principles. The team is aware of and analyzing previous efforts to articulate SE principles and will ensure the resulting set of guiding principles for systems engineering is comprehensive and truly foundational.

Randy Iliff pointed out in an editorial published in *PPI SyEN* in November 2019 that we don't always *act* like SE:

Given our training and theoretically enhanced ability to apply SE, we should be among the most tightly networked professional organizations on the planet. If we truly applied our systems thinking to the organization itself:

- We would seize every opportunity to connect members with each other and the outside world.
- We would value and maintain a robust, strategically designed set of alliances with other organizations.
- All of these interfaces would be recognized, honored, and enabled as a cultural norm.
- We would be guardians of SE's function, only loosely coupled to the recipe de jour, and in constant conversation about how to do more.

I would charitably observe that we continue to fall spectacularly short of these ideal behaviors. My thoughts on why that remains true after nearly thirty years are not intended to embarrass or indict, merely to point out areas that could benefit from improvement.

In Randy's view, we remain inwardly-focused rather than market-focused. Thus, the opportunity scope we apply to our vision of SE is limited, compared with the true region of "development effort" that stands to benefit from our practices.

INCOSE and many other individuals and organizations have been making huge efforts in the recent three to four years to reach out to both those in SE related work (such as ISSS, IEEE, NDIA, and SERC) as well as those in other disciplines and specific domains (such as SAE International, the International Association for the Engineering Modelling, Analysis, and Simulation [NAFEMS] Community, and others). A few examples include an initiative on the Future of Systems Engineering (FuSE), which includes about a dozen collaborating organizations under INCOSE leadership; collaboration on the integration of SE and PM (as described above); and developing guidance for the application of modeling and simulation at all levels of the system across the entire life cycle.

The Systems Engineering Handbook (SEH) is a guide for system life cycle processes and activities, for the new systems engineer, for the engineer in another discipline who needs to perform systems engineering, and for the experienced systems engineer who needs a convenient reference. It explains what each systems engineering process activity entails in the context of designing for affordability and performance. On some projects, a given activity may be performed very informally (e.g., on the back of an envelope, or in an engineer's notebook); or, on other projects, a more formal response is required with interim products under formal configuration control. This book provides tools that lead to project success in various circumstances. The Fourth Edition of the SEH is available from INCOSE. As a member benefit, the SE Handbook Fourth Edition digital copy is available for download from the INCOSE Store to members, employees of CAB organizations, and students of Academic Council members.

Dr. Donna Rhodes is Director and Principal Research Scientist at the Systems Engineering Advancement Research Initiative (SEAri) at the Massachusetts Institute of Technology (MIT USA). SEAri is comprised of researchers and affiliated faculty, visiting scholars and students, and undergraduate research assistants. Collaborative research efforts facilitate the transfer of knowledge, skills, and creativity across faculty and students within MIT, and with other universities. SEAri works collaboratively with its research sponsors to enrich research, achieve impactful outcomes, and enable deployment. Donna plans to provide an article for PPI SyEN in the mid-year time frame.

Extensive work has been done in recent years by the Systems Engineering Research Center (SERC) that is managed by Stevens University for the US Department of Defense (DoD). The SERC includes 24 universities that collaborate in research. In addition, work led by Jon Wade is performed with support from the Defense Acquisition University (DAU), to develop the *Experience Accelerator*, an initiative to create, validate, and transition technology and content for the use of experiential technology to educate systems engineers and technical leaders.

I'll conclude this exploration of the current status of systems engineering by highlighting the collaborative effort INCOSE is leading concerning the Future of SE. This initiative recognizes the need for us to understand the changes that SE must embrace and address as we go forward. It also recognizes the changes and needs articulated in INCOSE's SE Vision. We are basically in a race with accelerating technology development and adoption, as well as the increasing complexity and interconnectedness of systems. We must explore and evolve engineering nondeterministic, dynamic systems.

In answer to the question posed above, *how are we doing*, one might conclude amazingly well or one might observe that we have a long way to go. In my judgment, both are accurate characterizations of our current situation, although we must be mindful of Randy Iliff's admonition that our best efforts and energies are required.

With respect to the current major concerns and needs of SE, several have been identified in this article. Given the incredible talent engaged and the extensive efforts underway and planned, the SE discipline has huge potential as well as heavy moral, intellectual, and practical responsibilities to thoughtfully and energetically apply ourselves to contribute to a healthy, effective world.

Feedback concerning this editorial is welcome and should be sent to ryoung@ppi-int.com.

The ideas expressed in this editorial are those of the author.

4. SYSTEMS ENGINEERING NEWS

4.1 Gary Roedler Provides Review of INCOSE Activities

Editor's note: Gary Roedler, Outgoing President of INCOSE, provided an insightful Review of INCOSE activities in the December 2019 *INCOSE Newsletter*.

It is hard to believe that this is my final President's Corner article. The 4 years as President-Elect and President have flown by. During my term, I stressed the need to look forward to identify and prepare for the future needs of systems, systems engineering, and INCOSE. Over the course of the last four years, we have made several accomplishments worth highlighting.

Future of Systems Engineering (FuSE) Initiative. We established an industry-wide, global collaborative initiative focused on defining the needs and developing a roadmap to address systems engineering for dynamic, nondeterministic, evolutionary systems. The initiative has over a dozen organizations and associations working together. The team spans all the INCOSE sectors. Projects have been established for: artificial intelligence used in systems engineering and systems engineering for systems applying artificial intelligence; systems engineering foundations (more below on this); and horizon scanning to track advancements in technology that will impact the conduct of systems engineering. FuSE will initiate other projects as they define prioritized needs.

Systems of Systems Engineering (SoSE). We have published the Systems of Systems (SoS) primer (free to download from the INCOSE store); led the development of three new international standards for SoSE, the first in the industry; been a technical cosponsor of the SoSE Conference; and continued the highly popular SoSE webinar series.

Systems Engineering Foundations. The need to strengthen the foundations of systems engineering is a priority in both the INCOSE Vision 2025 and the FuSE initiative. In mid-2016, the INCOSE Fellows took on a project to review and refine the definitions for system and systems engineering to better support the future needs. That effort included comprehensive analysis of the world views and resulted in a revised set of definitions and a supporting paper presented to the Board of Directors for approval and published in the beginning of 2019. Concurrently, INCOSE participated to develop a set of systems engineering principles. The project team leveraged work started by NASA and others. After a workshop in December 2018, a team documented, presented, and published a draft set of systems engineering principles in 2019. Although there is still additional work needed to mature and evolve the principles, it provides an initial baseline.

Digital Engineering. As INCOSE has been working towards the transformation of systems engineering into a model-based discipline, we have realized that we also need to evolve in a way to integrate with the other disciplines to have the ability to share the engineering information and analysis across the boundaries of the disciplines—i.e., an integrated and holistic engineering capability. One significant success in this area is the establishment in 2018 of the Digital Engineering Information Exchange Working Group (DEIXWG). This is an INCOSE-led joint working group with the National Defense Industrial Association (NDIA) Systems Engineering Division (SED) and the US Department of Defense

(DoD). This group is identifying and defining the needs, especially the digital artifacts, required across the life cycle to support system definition and communication among stakeholders.

Agile Systems Engineering. As we continue to move more into evolutionary and dynamic environments with respect to system capabilities, it is essential to build an understanding of and guidance for agile systems and agile systems engineering. Here “agile” means the ability to deal with change. INCOSE conducted a series of workshops during 2016-2017 to analyze industry experiences and extract a set of fundamental principles. This resulted in a set of papers and webinars to help others gain from learnings in this area.

Systems Engineering Vision. We have planned the kick-off of a revision project for the INCOSE Vision 2025. We established a diverse Core Team for INCOSE Vision 2035, and the official project kick-off will be at the INCOSE International Workshop 2020 in January.

Systems Engineering Grand Challenges. We are collaborating with international organizations on applying systems engineering and systems approaches to help address grand challenges (such as availability of clean water and sanitation) and other societal issues and system needs. INCOSE is considering all these highlighted in the upcoming revisions of the INCOSE Handbook, the Systems Engineering Body of Knowledge (SEBoK), and the systems engineering-related international standards, such as ISO/IEC/IEEE 15288, Systems Life Cycle Processes. We need to continue to move the key systems engineering reference and standards forward so that the systems engineering discipline will evolve and continue to be a relevant factor in future system development and life cycle management. When I ran for President-Elect, I stressed four focus areas, one was the advancement of Systems Engineering, addressed above. Our INCOSE team has been doing an excellent job in looking forward and putting things in place to advance systems engineering. The other three focus areas are: INCOSE In-reach to build a more integrated, global INCOSE; INCOSE Outreach to build beneficial collaborations and alliances with other industry associations, professional societies, and other organizations towards to achievement of common objectives; and Professional Development to build a comprehensive capability to provide assistance in the professional development of those performing systems engineering. The dedicated teams of volunteers in INCOSE have helped to make significant progress in each of these areas.

For INCOSE in-reach, we have revised our policies and procedures for global operations of INCOSE that include how we interact. Additionally, we have ensured that the key projects of INCOSE reach across the geographical boundaries, using chapter efforts and including diversity in the team representation. The results have been superior. Finally, we have recently worked to update the agreements with all chapters to reflect our organization as it is today and where we are heading soon.

For INCOSE Outreach, we have made huge steps forward in building a set of alliances and collaborations that will positively affect both INCOSE and the systems engineering discipline for years to come. Many of the collaborations focus on the advancement of systems engineering, as described above. For more on this, see the President’s Corner article in the final 2018 issue of the INCOSE Newsletter.

For Professional Development, we have an initiative in place to develop an INCOSE Professional Development Portal (PDP) that is moving in a very encouraging direction. This initiative is on track to supply resources to enable a systems engineering community to interact in a platform environment that facilitates the exchange of ideas, information, training assets, etc. For more on this, see the President's Corner article in the May issue of the INCOSE Newsletter.

In looking forward, it is also essential to focus on the evolution of INCOSE as an organization from a business perspective. Although we are a non-profit organization, we need to ensure the health of the organization from more than its technical focus. Over the past 4 years, we have reviewed all aspects of INCOSE and identified areas that needed addressing to ensure a solid organization that is prepared for growth and ongoing change. The efforts include review and revision (as needed) of the following: all policies; bylaws; global operations; financial models; IT infrastructure; leadership roles; support contracts; membership value; membership types; value streams; agreements with others; and several other things.

Your INCOSE Newsletter Q4 3 Board of Directors is a highly dedicated and diverse group who have worked diligently to move INCOSE into a stronger position for the future. I want to thank them for their professionalism and contributions. The level of their efforts is not typically known by the average INCOSE member. Thank You! My opportunity to serve as the INCOSE President has given me a chance to build many lasting relationships. We have a great organization of highly skilled and dedicated people who believe in the mission and vision of INCOSE enough to volunteer their time to help make a difference. The teamwork is impressive. It has been my honor to be a leader in INCOSE and to be a part of the many teams that make INCOSE special. Thank you!

4.2 Special Issue Planned for the January/February 2021 *IEEE Intelligent Systems Journal* concerning Learning Complex Couplings and Interactions

The world is becoming increasingly complex, with a major drive towards incorporating increasingly intricate coupling relationships and interactions between entities, behaviors, subsystems, and systems, and their dynamics and evolution of any kinds and in any domains. Effective modeling of complex couplings and interactions in complex data, behaviors and systems directly addresses the core nature and challenges of complex problems, and is critical for building next-generation intelligent theories and systems to improve machine intelligence and understand real-life complex systems. This Special Issue on learning complex couplings and interactions will collect and report the latest advances in artificial intelligence and data science theories, models and applications of modeling complex couplings and interactions in big data, complex behaviors, and complex systems.

The Special Issue will solicit the recent theoretical and practical advancements in learning complex couplings and interactions in areas relevant but not limited to the following:

- Explainable learning of complex couplings and interactions
- Representation learning of complex couplings and interactions
- Coupling learning of complex relations, interactions and networks and their dynamics
- Interaction learning of activities, behaviors, events, processes and their sequences, networks, and dynamics

- Learning couplings and interactions in hybrid intelligent systems and problems
- Learning natural, online, social, economic, cultural, and political couplings and interactions
- Learning real-time, dynamic, high-dimensional, heterogeneous, large-scale, and sparse couplings, dependencies, and interactions
- Probabilistic and stochastic modelling of coupling and interaction uncertainty and dependency
- Deep learning of complex couplings and interactions in big data and complex behaviors
- Large-scale simulation of complex couplings and interactions
- Visual analytics and visualization of complex couplings and interactions
- Impactful applications and tools for modeling complex couplings and interactions
- Human understandable explanations of complex couplings and interactions

Submission and Timeframe (https://mc.manuscriptcentral.com/is-cs):

All submissions must comply with the Submission Guidelines of *IEEE Intelligent Systems* and will be reviewed by research peers. The schedules are as follows:

- Paper Submission Date: March 30, 2020
- First Round of Reviews Date: May 30, 2020
- Revision Date: July 30, 2020
- Final Accept/Reject Date: October 2, 2020
- Camera Ready Copy Date: October 30, 2020
- Publication Date: January/February 2021

Guest Editors

- Dr. Can Wang (Griffith University, Australia)
- Prof. Fosca Giannotti (University of Pisa, Italy)
- Prof. Longbing Cao (University of Technology Sydney, Australia)

Inquiries

Inquiries about this Special Issue can be sent to Dr Can Wang (can.wang@griffith.edu.au), Professor Fosca Giannotti (fosca.giannotti@isti.cnr.it) or Professor Longbing Cao (longbing.cao@uts.edu.au).

4.3 IISE Launches New International Competition “The IISE Cup”

The Institute of Industrial and Systems Engineers (IISE) is launching a new international competition that provides an exciting opportunity for companies to highlight their successful innovation solutions. The IISE Cup will recognize organizations for innovative and effective implementation of industrial and systems engineering principles and practices that deliver exemplary business performance improvement. Starting in 2020 IISE will conduct the award competition to promote successful implementation of industrial & systems engineering practices by recognizing best in class industries and sharing their stories with others. This will be one of IISE's most prestigious awards to be given to a team and their industry. The award will be bestowed to the top three teams and be presented at the upcoming IISE Annual Conference & Exposition.

Awarding process

- All entries will be pre-screened to select **10 finalist teams**.
- All finalist teams will be required to attend the IISE Annual Conference & Expo and present their cases.
- **Top 3 teams** will be selected and presented at the IISE Annual Conference & Expo. There will be a 1st place gold, 2nd place silver and 3rd bronze awarded
- Conference attendees will also vote to select the Excellence Award.

Evaluation criteria

- The industrial and systems engineering implementation initiative is innovative.
- The industrial and systems engineering implementation initiative had clear measurable performance objectives and they were met.
- The industrial and systems engineering implementation approach is simple and straightforward.
- The industrial and systems engineering initiative used ISE methods, tools and techniques.
- The industrial and systems engineering initiative derived significant and measurable outcomes such as cost reduction, supply chain complexity reduction, lead time reduction, quality improvement, safety improvement, etc.
- The industrial and systems engineering implementation is sustainable.
- The quality of the presentation is excellent including problem statement, clear descriptions, and metrics with visuals.

More Information

4.4 Job Opportunity: Senior Systems Engineer at Rolls Royce

Rolls Royce in Peterborough, Ontario, Canada is looking for a Senior Systems Engineer who will work with the Chief Design Engineer to ensure the technical efficacy, reliability, quality, and safety of the product. The appointed person will be responsible for the technical leadership, coaching, and guidance of deployed engineers in an Integrated Program Team (IPT) for Naval Handling Equipment. The Senior Systems Engineer will also be accountable for the coordination and delivery of technically challenging work packages, contributing directly to defense objectives from within a large cross-functional team. The role will be the prime interface between large numbers of deployed engineers and IPT leadership, providing the opportunity to work with senior leadership in concert with the day-to-day responsibility for personal and team engineering task delivery.

Requirements

- Bachelor's OR Master's degree in Mechanical Engineering or relevant Science or Mathematics degree subject with suitable experience
- Appropriate domain design/functional domain knowledge to facilitate competent oversight of deployed engineering delivery
- Demonstrable program leadership of successfully delivered technical projects/tasks to time, cost, quality
- Proven, extensive experience in the management of technical work packages or projects, of increasing complexity.
- Experience of managing or coordinating projects or programs in unfamiliar contexts or subjects
- A good level of knowledge and experience in all the technical skills recognized as being required for the management of technical work packages
- Proven ability to apply logical, analytical and innovative thinking on a range of technical problems and to make balanced decisions in complex situations and with incomplete information
- Proven, extensive experience in provisioning optimized design/function solutions
- Has a minimum of five years of engineering leadership experience
- Demonstrable understanding of process driven product change control including (but not limited to) PCB, DDRAM, IPPR/PPAP processes and Product Introduction Lifecycle Management experience, etc.
- Has an excellent working-level knowledge and experience of all relevant technical skills and is widely recognized as such
- Experience applying Systems Engineering approach or equivalent (Robust Design, Black belt, PELA etc.); advocate of OPS and positive change

Click here to apply

4.5 Job Opportunity: Lead Engineer Systems Engineering with GE Aviation

General Electric in Evendale, Ohio in the United States is looking for a Lead Engineer who will demonstrate leadership in communicating business goals, programs, and processes for an area or business segment. In this role, you will utilize your experience or expertise to solve problems, develop and execute objectives for self and others, and have the ability to effect short-term and some long-term business goals.

Requirements

- Bachelor of Science in Engineering, Physics, Chemistry, Mathematics, or Computer Science from an accredited university or college (Or a high school diploma / GED with a minimum of 4 years of IT, engineering, scientific or other technical experience)

- A minimum of 3 years of experience in an engineering position.

Desired Characteristics

- Ability to obtain and/or maintain government security clearance
- Thermodynamics experience
- Experience with coding and analysis
- Strong technical aptitude, including applicable engineering tools and systems
- Solid verbal and written communication skills
- Strong interpersonal and leadership skills
- PC proficiency
- A minimum of 5 years of experience in an engineering position

Click here to apply

4.6 Jama Software On-Demand Webinar: The High Cost of Poor Requirements Management

Most product development teams think their requirements management process is just fine, according to a recent Engineering.com report. The problem is, the same research found most teams dealt with excessive product outcome failures, blown budgets, and regulatory sanctions due to poor requirements management.

Whether teams are working in silos or using internally developed requirements management tools, the negative outcomes that manifest as a result don’t have to just be the price of doing business.

Watch this on-demand webinar to learn how a modern requirements management solution can provide teams with a single source of truth, reduce risk and uncertainty, and increase efficiency and collaboration.

Watch this webinar now

4.7 INCOSE UK Annual Systems Engineering Conference Call for Content

The Annual Systems Engineering Conference is the UK’s premier Systems Engineering event, attracting a wide range of industry professionals, international presenters and practitioners, which provides a distinguished platform for networking, learning and sharing ideas.

ASEC 2020 Call for Content
The ASEC 2020 call for content document has now been released.
For further information about how to submit content for ASEC 2020 click the link!
#ASEC2020UK
INCOSEUK

Image Source

Access more information: https://incoseuk.org/Documents/Group_Documents/UK%20Chapter/ASEC_2020_Call_for_Content.pdf

Call For Papers

INCOSE UK is inviting individuals and organizations to submit papers to be delivered at ASEC 2020. The theme for this year's ASEC is "The Challenges of Contemporary Systems Engineering". Within this INCOSE will be exploring the following subthemes:

1. Integrating with new disciplines
2. Developing new competencies
3. Evolving current practices
4. Exploiting emerging technologies and techniques

However, INCOSE is open to proposals that do not necessarily fall within a rigid theme, and are keen to accept papers based around a variety of interesting applications in Systems Engineering. All papers must be able to fill a 40-minute presentation timeslot, and should include a five-minute window to allow presenters to answer any questions from their audience.

Call for Tutorials

ASEC includes tutorial sessions which are 2 hours 15 minutes in length. The format of a 'tutorial' can cover a range of options: from a 'set of instructions to complete a task' to 'an interactive problem-solving workshop'. A typical ASEC tutorial attracts between ten and thirty delegates who are interested in trying something new or perhaps finding a different perception on something already familiar to them (and maybe looking for a break from back-to-back sessions!)

INCOSE UK is looking for tutorial options which will be selected from the proposals received. Proposers are welcome to submit more than one proposal, but each proposal must be self-contained so that it can be assessed independently. It is vital that the tutorial style is explicitly expressed so that delegates clearly understand whether they will be expected to plaster flip charts with Post-it Notes or simply sit and listen.

Successful presenters will be awarded a single one-day pass for ASEC 2020. If you wish to field more than one presenter, then additional presenters must register via the ASEC 2020 booking system at a standard delegate price. You should be prepared to present on either 17 or 18 November 2020. Tutorial Presenters will be asked to provide a description of their activity for publication in the event brochure.

Call for Posters

The Systems Research Showcase has become a familiar part of the Annual Systems Engineering Conference (ASEC) over the past few years, and this year aims to be even better.

The Systems Research Showcase is a competition for research posters, that provides a wider remit to participants than similar competitions and gives a platform for contributors to display and present their work. This will be a unique opportunity to present ideas and research from people across Systems Engineering. INCOSE UK encourage submissions from:

- Individual full-time students (75% of time spent in education), who are researching Systems Engineering topics.
- Group contributions from academic research groups in universities.
- Project contributions such as EU and EPSRC projects.
- Industry research teams.

Posters will be displayed in the networking areas for delegates to view during breaks and lunch times. Authors should be available at the conference to answer questions from delegates. Poster entrants will be in with the chance of winning a prize which will be presented at the ASEC 2020 conference dinner.

4.8 INCOSE Recognition Nominations: Deadline Extended

Submissions for INCOSE Recognition Awards have been extended to 15 March 2020

As a non-profit professional engineering organization, the success of the International Council on Systems Engineering (INCOSE) is entirely due to the efforts of its leaders and the many other volunteers who share their time, expertise, and knowledge to nurture the organization and advance the practice of systems engineering. Through the Hall of Leaders, INCOSE recognizes and remembers the many individuals who have made the organization and systems engineering what it is today.

INCOSE strives to recognize those who contribute to the advancement of the systems engineering profession as well as the advancement of our organization. Criteria for our INCOSE awards are shown below. Click each award for more information:

- Pioneer Award
- Founders Award
- Fellows Award
- Outstanding Service Award

- Working Group Award
- Chapter Award

Note:

All award nomination packages should be sent electronically by their respective deadlines to awards@incose.org. An acknowledgment of receipt will be sent within one week. If you do not receive a confirmation, please resend.

4.9 Launch of the INCOSE Systems Engineering Tools Database Working Group

INCOSE has chartered a Systems Engineering Tools Database Working Group (SETDB WG) to complete development and to operate for INCOSE the SETDB in partnership with PPI . The WG is Chaired by John Nallon, with René King (PPI) and Stéphane Lacrampe as Co-Chairs.

4.10 Launch of the INCOSE Social Systems Working Group

INCOSE has chartered a new Working Group: Social Systems Working Group (SocWG). The WG has its first meeting at the INCOSE International Workshop (IW) in Torrance, CA, USA over January 25 to 28, 2020.

The SocWG has defined some initial focus areas for its first year, that include:

Defining the basics of Social Systems:
SEBoK development on Social Systems
Possible inclusion of social systems in the new version of the SE handbook that is under development

Sustainability/Sustainable Development Goals (SDGs) and SE:
Developing a core team of people to lead key SDG thematic areas (agriculture, water, poverty, education, etc.) or SDG ontology.

The WG is Chaired by Erika Palmer and Co-Chaired Randy Anway. If you are interested in participating in the work of the group, contact Erika Palmer at erika.palmer@ruralis.no.

5. FEATURED ORGANIZATIONS

5.1 INCOSE Institute for Technical Leadership

"A development program for active INCOSE members seeking to improve their leadership skills in an open, collaborative environment."

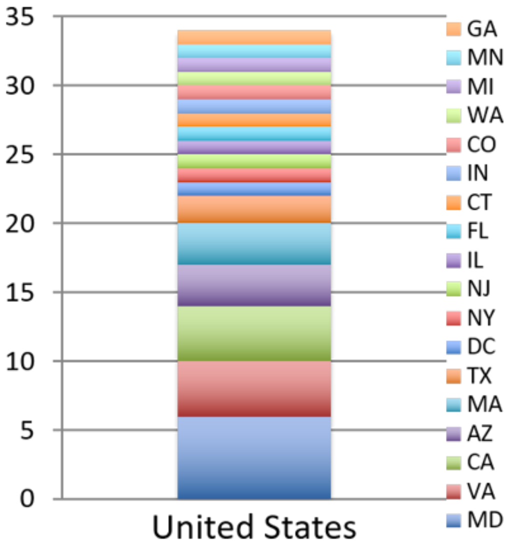
Outline

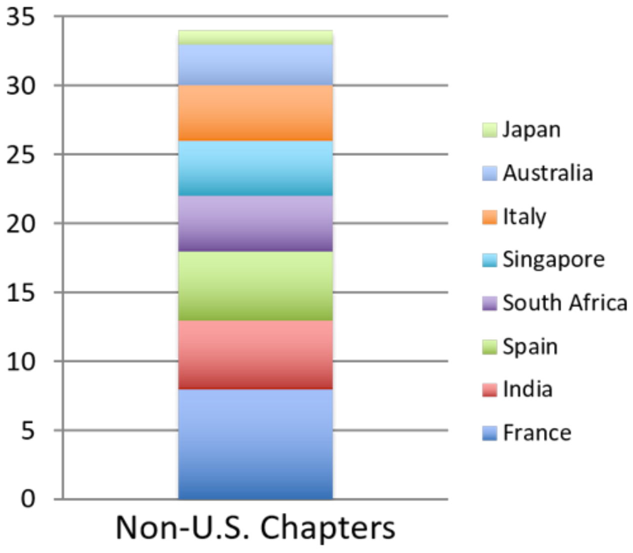
- INCOSE has a growing pool of qualified leaders to draw on and an enhanced international reputation for SE leadership.

- Individual members become more capable leaders and join an international network of systems engineering leaders.

- Sponsoring organizations obtain non-proprietary, tuition-free technical leadership training for their future SE leaders.

Four cohorts have now been formed, building a growing network of 68 leaders from 5 continents, 9 countries and 18 U.S. States!





INCOSE leaders nominate full INCOSE members for participation in the INCOSE Institute for Technical Leadership during February-March of each year. Interested members should discuss their interest with their Chapter President or Regional Director.

For further information, please contact Michael Pennotti at michael.pennotti@stevens.edu

View The INCOSE Institute for Technical Leadership booklet:

https://www.incose.org/docs/default-source/about/incose_institute_booklet.pdf?sfvrsn=d29c96c6_2

5.2 French Chapter of INCOSE: Association Française d'Ingénierie Système (AFIS)

The Association Française d'Ingénierie Système (AFIS) mission consists of the following objectives:

- Promote System Engineering by presenting and explaining its principles and its multidisciplinary approach with a view to achieving successful equipment and systems.
- Foster and ensure the development and use of System Engineering with public and private companies and state organizations.
- Promote training in System Engineering to all educational communities: students, teachers, training organizations, research laboratories, etc.
- Encourage exchanges between users and thus increase the knowledge base of System Engineering, while allowing its adaptation to the just needs of the different application sectors.
- Ensure the professional representation of users of System Engineering at national, European and international level.

AFIS hosts a number of events including conferences and forums.

Find out more about the activities of AFIS here

6. NEWS ON SOFTWARE TOOLS SUPPORTING SYSTEMS ENGINEERING

6.1 Tom Sawyer Announces Perspectives 9.0

Features:

The new Tom Sawyer Launch Center makes it easy to get started with Tom Sawyer Perspectives. With a single click, you can start Tom Sawyer Graph and Data Visualization, Tom Sawyer Graph Database Browser, Tom Sawyer Business Process, Tom Sawyer Model-Based Engineering, and the documentation for each module. You can also use the Launch Center to view licensing details and stop all the web servers from a single dialog.

In Tom Sawyer Graph and Data Visualization, customers can now use schema extraction for Microsoft Excel data sources and enjoy greatly reduced edge crossings in orthogonal layout. In addition, application start time is reduced by up to 30% and drawing view model update performance is up to 5% faster.

For customers with advanced Model-Based Systems Engineering (MBSE) needs, Tom Sawyer Model-Based Engineering includes many more customization options including the ability to custom color objects by metaclass and allow to filter the diagram tree to show only diagrams that were previously saved by Tom Sawyer Perspectives. There is also special rendering for bus nodes and the ability to group elements by attribute and place them into a nested drawing structure. On the access front, an Internet connection is no longer needed to use the Tom Sawyer Model-Based Engineering web application.

Tom Sawyer Business Process can now automatically create lanes based upon the departments or roles of task owners. And, Tom Sawyer Graph Database Browser has a user-friendly embedded query editor and improved appearance rules.

This release also includes a new Web Application Developer’s Guide that describes how to create a web application from a Tom Sawyer Perspectives Designer project

Find out more about improvements in quality and features of each module here

6.2 Modelio 3.8.1 Released

The latest version of Modelio released in early 2019 now includes the following features:

Evolutions:

- It is now possible to clone graphic properties from a diagram to another.
- It is now possible to save the Audit results or copy them into the clipboard.
- It is now possible to unmask BPMN Tasks in Class Diagrams.
- It is now possible to mask related diagrams in BPMN diagrams.
- The multiple elements selection box has been revamped.
- The Modelio log is now automatically refreshed.
- It is now possible to select a Combined Fragment which is behind another.
- “0.0” coordinates are now indicated in diagrams.
- The related diagram icon is now 24x24 instead of 48x48.
- A separator has been added between the Save and Undo/Redo buttons.

Fixes/Improvements:

- Fixed issue with Use Cases edition box.
- Fixed issue with model import on MacOS.
- Fixed issue when renaming projects.
- Fixed random issue with project creation on laptops.
- Fixed issues with the XMI import/export.

Download Modelio 3.8.1 here: http://www.modelio.org/downloads/download-modelio.html

Download the language packs here: http://www.modelio.org/downloads/language-packs.html

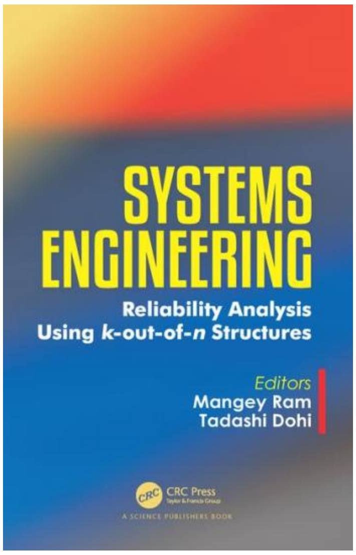
Access the Modelio forum here for more information on the tool.

7. SYSTEMS ENGINEERING PUBLICATIONS

7.1 Systems Engineering: Reliability Analysis using k-out-of-n Structures

by

Mangey Ram and Tadashi Dohi



From the Amazon.com Website:

A substantial amount of research has been conducted on consecutive *k*-out-of-*n* and related reliability systems over the past four decades. These systems have been used to model various engineering systems such as the microwave stations of telecoms network, oil pipeline systems, and vacuum systems in an electron accelerator. As such, studies of reliability properties of consecutive *k*-out-of-*n* structures have attracted significant attention from both theoretical and practical approaches. In the modern era of technology, the redundancies are employed in the various industrial systems to prevent them from failure/sudden failure or to recover from failures. This book is meant to provide knowledge and help engineers and academicians in understanding reliability engineering by using *k*-out-of-*n* structures. The material is also targeted at postgraduate or senior undergraduate students pursuing reliability engineering.

Publisher: Publisher: CRC Press; 1 edition (April 18, 2019)

Format: Kindle, Hardcover

ISBN-10: 1138482927

ISBN-13: 978-1138482920

More Information

7.2 Ten Requirements Traps to Avoid

by

Karl Wiegers

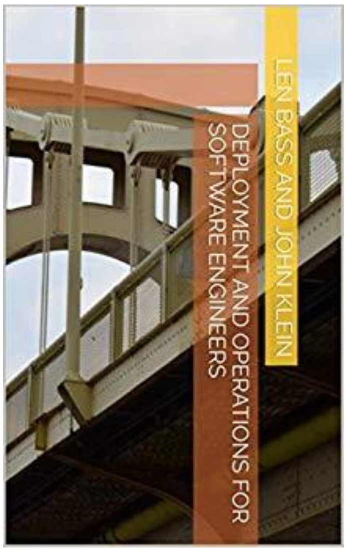
The path to quality software begins with excellent requirements. Slighting the processes of requirements development and management is a common cause of software project frustration and failure. This article describes ten common traps that software projects can encounter if team members and customers don't take requirements seriously. Several symptoms that might indicate that one is falling victim to each trap are described. Several solutions to control the problems are provided.

Read the article

7.3 Deployment and Operations for Software Engineers

by

Len Bass and John Klein



From the Amazon.com Website:

Software engineering practices require knowledge of the environment in which an application is to be run. In the modern world, this means knowledge of virtualization, containers, networking, the cloud, and security techniques for the internet. A developer should also know about microservices, configuration management, the deployment pipeline, monitoring and post production, disaster recovery, and how to develop secure applications. These topics, and more, are all covered in this book. The book includes exercises and discussion questions to facilitate classroom or group learning.

Sold by: Amazon.com Services LLC (Published May 6, 2019)

Format: Kindle, Paperback

ASIN: B07QKLWWFJ

More Information

7.4 Engineering Systems Integration

by

Gary O. Langford

From the Amazon.com webpage:

Engineering Systems Integration: Theory, Metrics, and Methods looks at the fundamental nature of integration, exposes the subtle premises to achieve integration, and posits a substantial theoretical framework that is both simple and clear. Offering systems managers and systems engineers the framework from which to consider their decisions in light of systems integration metrics, the book isolates two basic questions, 1) Is there a way to express the interplay of human actions and the result of system interactions of a product with its environment?, and 2) Are there methods that combine to improve the integration of systems? The author applies the four axioms of General Systems Theory (holism, decomposition, isomorphism, and models) and explores the domains of history and interpretation to devise a theory of systems integration, develops practical guidance applying the three frameworks, and formulates the mathematical constructs needed for systems integration.

The practicalities of integrating parts when we build or analyze systems mandate an analysis and evaluation of existing integrative frameworks of causality and knowledge. Integration is not just a word that describes a best practice, an art, or a single discipline. The act of integrating is an approach, operative in all disciplines, in all we see, in all we do.

Publisher: Publisher: CRC Press; 1 edition (May 9, 2012)

Format: Kindle, Hardcover, Paperback

ISBN-10: 143985288X

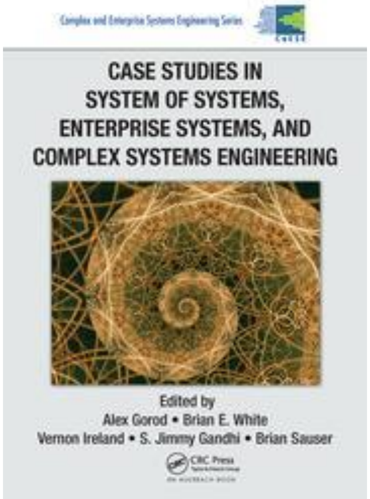
ISBN-13: 978-1439852880

More Information

7.5 Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering

Edited by

Alex Gorod, Brian E. White, Vernon Ireland, S. Jimmy Gandhi, and Brian Sauser



From the Amazon.com Website:

Suitable as a reference for industry practitioners and as a textbook for classroom use, *Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering* provides a clear understanding of the principles and practice of system of systems engineering (SoSE), enterprise systems engineering (ESE), and complex systems engineering (CSE).

Multiple domain practitioners present and analyze case studies from a range of applications that demonstrate underlying principles and best practices of transdisciplinary systems engineering. A number of the case studies focus on addressing real human needs. Diverse approaches such as use of soft systems skills are illustrated, and other helpful techniques are also provided. The case studies describe, examine, analyze, and assess applications across a range of domains, including:

- Engineering management and systems engineering education.
- Information technology business transformation and infrastructure engineering.
- Cooperative framework for and cost management in the construction industry.
- Supply chain modeling and decision analysis in distribution centers and logistics.
- International development assistance in a foreign culture of education.
- Value analysis in generating electrical energy through wind power.
- Systemic risk and reliability assessment in banking.
- Assessing emergencies and reducing errors in hospitals and health care systems.

- Information fusion and operational resilience in disaster response systems.
- Strategy and investment for capability developments in defense acquisition.
- Layered, flexible, and decentralized enterprise architectures in military systems.
- Enterprise transformation of the air traffic management and transport network.

Supplying the reader with a better understanding of SoSE, ESE, and CSE concepts and principles, the book highlights best practices and lessons learned as benchmarks that are applicable to other cases. If adopted correctly, the approaches outlined can facilitate significant progress in human affairs.

The study of complex systems is still in its infancy, and it is likely to evolve for decades to come. While this book does not provide all the answers, it does establish a platform through which analysis and knowledge application can take place and conclusions can be made in order to educate the next generation of systems engineers.

Series: Complex and Enterprise Systems Engineering

Paperback: 808 pages

Publisher: CRC Press; 1 edition (February 27, 2017)

Language: English

ISBN-10: 1138199834

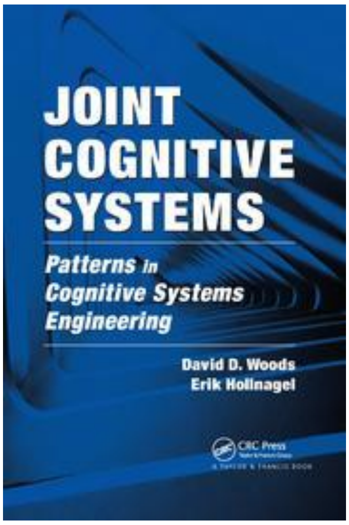
ISBN-13: 978-1138199835

More Information

7.6 Joint Cognitive Systems: Patterns in Cognitive Systems Engineering

by

David D. Woods and Erik Hollnagel



From the CRC Press Website:

Our fascination with new technologies is based on the assumption that more powerful automation will overcome human limitations and make our systems 'faster, better, cheaper,' resulting in simple, easy tasks for people. But how does new technology and more powerful automation change our work?

Research in Cognitive Systems Engineering (CSE) looks at the intersection of people, technology, and work. What it has found is not stories of simplification through more automation, but stories of complexity and adaptation. When work changed through new technology, practitioners had to cope with new complexities and tighter constraints. They adapted their strategies and the artifacts to work around difficulties and accomplish their goals as responsible agents. The surprise was that new powers had transformed work, creating new roles, new decisions, and new vulnerabilities. Ironically, more autonomous machines have created the requirement for more sophisticated forms of coordination across people, and across people and machines, to adapt to new demands and pressures.

This book synthesizes these emergent patterns though stories about coordination and discoordination, resilience and brittleness, affordance and clumsiness in a variety of settings, from a hospital intensive care unit, to a nuclear power control room, to a space shuttle control center. The stories reveal how new demands make work difficult, how people at work adapt but get trapped by complexity, and how people at a distance from work oversimplify their perceptions of the complexities, squeezing practitioners. The authors explore how CSE observes at the intersection of people, technology, and work, how CSE abstracts patterns behind the surface details and wide variations, and how CSE discovers promising new directions to help people cope with complexities. The stories of CSE show that one key to well-adapted work is the ability to be prepared to be surprised.

Publisher: Publisher: CRC Press (March 31, 2020)

Format: Paperback, Hardback, eBook, and eBook rental

ISBN-13: 9780367864156

More Information

7.7 INSIGHT December 2019 - Volume 22: Issue 4 Released

A publication of INCOSE

INSIGHT
This Issue's Feature:
AFIS Doctoral Symposium: New Challenges and Advances in Systems Engineering at French Universities
DECEMBER 2019
VOLUME 22 / ISSUE 4
A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING
INCOSE

- AFIS Doctoral Symposium: New Challenges and Advances in Systems Engineering at French Universities
- Review of the AFIS 2018 Academy-Industry Meetings in Nancy- The Celebration of the 20th Anniversary of AFIS!
- RobAFIS Student Competition Actuality: Safety & Security Interactions Between Operators and with the System
- Extended Enterprise Model for PSS Within a Systems Engineering Perspective
- Management of the Design Process: Human Resources Allocation and Project Selection in Factories of the Future
- A Monitoring Strategy for Industry 4.0: Master Italy s.r.l Case Study
- Challenges for Autonomous Vehicles (AVs) Engineering: Safety Validation of Functional Performance Limitations
- Systems Engineering and Dependability: Methodology of Model Synchronization between System Architecture Models and Risk Analysis
- A Model-Based Approach to Design, Organize, and Monitor Dismantling and Decommissioning of Nuclear Facilities
- On the Mastering of Modelling Activities Development in Engineering
- Towards a Maturity Assessment Scale for the Systems Engineering Assets Valorization to Facilitate Model-Based Systems Engineering Adoption

- Evaluation of Systems Contractor's Ability to Deliver a Solution to Offer During an Engineer-To-Order Bidding Process

- Coordination of Multi-Underwater Drones: Towards an Integrated Object-Oriented Methodology in an Open-Source Environment.

View this Issue (Requires login to the INCOSE Collaboration Center “Connect”)

Scroll to the bottom of this page:

https://connect.incose.org/Library/InsightMagazine/default.aspx

7.8 Detecting Incoherent Requirements in Large Documents

by

Dr. Patrick Saint-Dizier

Senior researcher at CNRS Toulouse in Computational Linguistics

Published in RE Magazine

Abstract

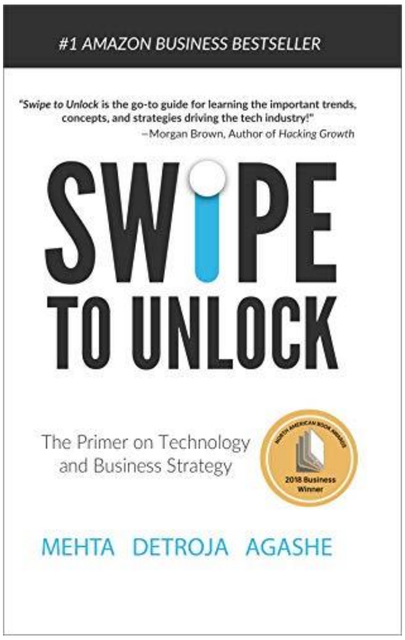
The different protagonists involved in requirements development are a source of mismatches, inconsistencies, incoherence, and redundancies. Stakeholders, technical writers, managers, users, and manufacturers all play different roles and have different views concerning requirements. These discrepancies are developed in technical writing principles (Schriever, 1989), (Unwalla, 2004), (Weiss, 1991), and (Kuhn 2014), where a large synthesis is proposed. In an attempt to overcome these problems, most industrial sectors have defined authoring recommendations (e.g., IEEE 830 and variants, ISO 29148, methods (IREB Handbook of Requirements Modelling (Cziharz et al. 2016), Elicitation (Häußer et al., 2019) and tools to elaborate, structure, and write requirements of various types. These specify how correct requirements should be elaborated. For that purpose, they provide attribute elicitation techniques and discuss the iterative application of requirements processes within life cycles. The result is easier traceability, control, and update.

Read the Article

7.9 Swipe to Unlock: The Primer on Technology and Business Strategy

by

Neel Mehta, Aditya Agashe, and Parth Detroja


#1 AMAZON BUSINESS BESTSELLER
"Swipe to Unlock is the go-to guide for learning the important trends, concepts, and strategies driving the tech industry!"
–Morgan Brown, Author of Hacking Growth
SWIPE
TO UNLOCK
The Primer on Technology and Business Strategy
2018 Business Winner
MEHTA DETROJA AGASHE

From the Amazon.com Website:

Authored by 3 Product Managers at Google, Facebook, and Microsoft, *Swipe to Unlock* is a comprehensive guide on the must-know concepts of technology and business strategy. It is a must-read for anyone pursuing product management, design, marketing, consulting or business strategy roles in the tech industry.

Swipe to Unlock was updated in 2019 to include over 40 pages of new content to cover the latest developments in the world of tech.

This #1 Amazon Business Bestseller won a medal from the North American Book Awards and has been featured in The Wall Street Journal, Forbes, and Business Insider. *Swipe to Unlock* was touted as "our generation's Rosetta Stone for enabling anyone to peer into the technology changing everyday life" by Jeremy Schifeling.

- How does Spotify determine what songs to recommend to you?
- How did KaiOS become the third largest mobile operating system in just two years since launching?
- Why does Amazon offer free shipping with Prime if it loses them money?
- How did a single typo take down 20% of the internet?
- Why did Microsoft acquire LinkedIn?

By answering real-world questions such as these, *Swipe to Unlock* gives you a peek under the hood of the technology you use every day, decodes tech's biggest buzzwords, and shows you how technology is changing the society we live in for better or for worse.

Topics Covered: Software Development, Business Models and Strategies, Economics, Hacking and Security, Hardware and Robots, The Internet, Cloud Computing, Big Data, Technology Policy, Emerging Markets, Future Trends, and others.

Featured Companies: Google, Facebook, Microsoft, Amazon, Apple, Spotify, Uber, WeChat, Yelp, Tinder, Washington Post, Grab, Toyota, GoJek, Samsung, Salesforce, M-Pesa, Quora, KaiOS, Twitter, Tesla, ByteDance, Airbnb, Robinhood, Adobe, Alibaba, Netflix, Paytm, Target, and others.

Publisher: CreateSpace Independent Publishing Platform (September 20, 2017)

Format: Kindle, paperback

ISBN-10: 1976182190

ISBN-13: 978-1976182198

More Information

8. EDUCATION AND ACADEMIA

8.1 UC Berkeley’s Online Masters in Cybersecurity

The University of California (UC) Berkeley School of Information’s (I School) Master of Information and Cybersecurity (MICS) is an accredited online program that prepares students with the cybersecurity skills needed to assume leadership positions in private-sector technology companies as well as government and military organizations.

UC’s holistic approach to cybersecurity develops students’ understanding of information security technologies as well as the economic, legal, behavioral, and ethical impacts of cybersecurity. Students graduate as competitive candidates in the job market with connections to UC Berkeley alumni and professionals in the San Francisco Bay Area technology hub.

- Complete your Master’s in as few as 20 months
- Attend live classes and complete course work online
- Connect with classmates at immersion experiences
- Network with leaders at the heart of Silicon Valley

Find out more

8.2 Institute for Advanced Systems Engineering

In 2013, the University of Connecticut (UConn) USA founded the United Technologies Corporation (UTC) Institute for Advanced Systems Engineering (UTC-IASE) with financial support from the United Technologies Corporation. UTC's initial pledge of $10 million, spanning five years, helps the Institute to reach its goals on both short- and long-term bases:

- Establish several educational programs, including certificate programs and a Master of Engineering (MEng) program in Systems Engineering (SE).
- Conduct a series of competitively selected research projects with a funding goal of $500,000 per year (total), to be conducted by UConn research scientists and students in collaboration with UTC.
- Create UTC-IASE fellowships to attract highly qualified students and to enhance research activities.
- Establish industry outreach activities; including a seminar series, a distinguished lecture program, and an international conference; with the objective of hosting internationally renowned scholars to engage a broad community of scientist and engineers within UTC, UConn, and beyond.

The UTC-IASE now serves as a hub for world-class research, project-based learning by globally distributed teams of students, and industrial outreach activities focused on model-based systems engineering (MBSE) of complex systems that are built from, and are dependent on, the synergy of computational and physical components. The convergence of computation, communications, control, and intelligence enables cyber physical systems (CPS) to have learning and predictive capabilities able to adapt to changing situations. Motivated by the increasing complexity of advanced products and the digital revolution, the Institute trains engineers in urgently needed CPS-related disciplines that are pivotal to innovation and product enhancement in the globally competitive economy. The Institute is positioned to advance the science base of CPS and to accelerate its technological translation into sustained industrial growth.

More Information

9. SOME SYSTEMS ENGINEERING-RELEVANT WEBSITES

Systems Engineering Approaches to Address Cybersecurity Challenges

A webpage containing information regarding how to apply systems engineering approaches to address cybersecurity challenges of the electric grid. The page outlines SE activities that may be useful in addressing real cyber threats to utility operations.

https://electricenergyonline.com/energy/magazine/1221/article/Systems-Engineering-Approaches-to-Address-Cybersecurity-Challenges-of-the-Electric-Grid.htm

Future of Systems Engineering

A downloadable .pdf about the INCOSE FuSE initiative – an ongoing project to better characterize the future and address the anticipated challenges using systems engineering approaches. The presentation contains data evidencing the growing complexity of systems and paints the picture of SE in 2020 and beyond.

http://www.kr.inf.uc3m.es/wp-content/uploads/2019/12/AI4SE-Schreinemakers.pdf

SE Research Consortium

The NASA Systems Engineering Research Consortium (NASA Consortium) was co-founded in the fall of 2010 to develop approaches to systems engineering that focus on the consistent development and operations of elegant system. This website contains principles, documents, links and other resources that emerged out of the consortium activities.

https://www.nasa.gov/consortium

10. STANDARDS AND GUIDES

10.1 AV Safety Standards Set to Arrive in 2020

From EE Times:

In 2020, a host of new industry standardization initiatives for artificial intelligence will roll out in earnest. Their common mission is to develop safety standards for AI-driven systems in autonomous vehicles and robotics.

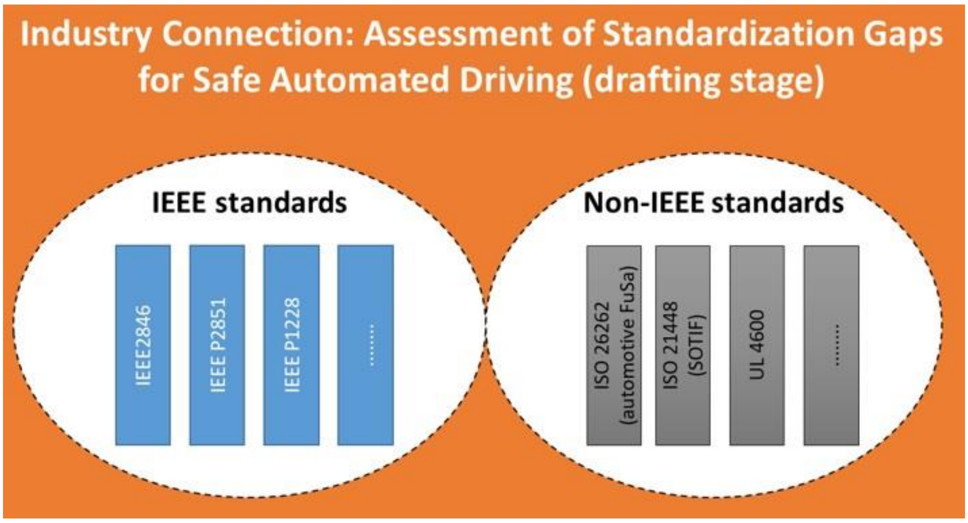
With so many efforts pursuing the same critical goal, questions loom. How much overlap will there be among the emerging safety standards? How well are these efforts coordinated? Will they leave any safety gaps unfilled? The answers aren't yet clear, but the rollout announcements offer some clues, and EE Times has reached out to people close to the efforts for insight.

IEEE alone is kick-starting several freshly approved working groups aimed at AV safety: IEEE P2846, IEEE P2851 and IEEE P1228.

IEEE P2846, initiated by Intel/Mobileye, seeks a "formal model for safety consideration in automated vehicle decision-making." IEEE P2851's job, meanwhile, is a much-needed interoperable "data format for safety verifications of IP, SoC and mixed-signal ICs," according to the working group. IEEE has reactivated an IEEE standard originally developed for nuclear power plants. IEEE P1228 now will address AV software safety throughout the vehicle's life cycle.

IEEE isn't alone. Underwriters Laboratories' draft of UL 4600 is currently on the ballot. Billed as "The First Comprehensive Safety Standard for Autonomous Products," it is a little different from other industry standards.

Instead of prescribing how to do safety by following certain steps, UL 4600 offers a guide to "build the safety case" for an AV design. Intel's Jack Weast, chair of IEEE's new P2864 working group, sees UL 4600 complementary to IEEE P2864, calling UL 4600 the "low-hanging fruit."



Industry Connection: Assessment of Standardization Gaps for Safe Automated Driving (drafting stage)

Source: IEEE

The standard offers a checklist against which AV designers can argue the safety case for their design elements. Finally, there are existing safety standards. One is ISO/PAS 21448 (SOTIF) covering the "safety of the intended functionality." The standard focuses on guaranteeing the safety of the intended functionality in the absence of a fault. It thus stands in contrast to traditional functional safety described in ISO 26262, which aims at mitigating risk due to system failure.

This sudden rush to set standards for safety and reliability in AI-based systems is a marked departure from the practices of the very recent past. AV developers have long kept their methods close to the vest, disclosing scant data to the public (except for miles driven and disengagements — deactivation of the autonomous mode when a failure of the autonomous technology is detected) and treating safety in self-driving vehicles as a wedge for competitive advantage.

Today, neither industry nor government can assess the safety of self-driving cars. Without tools or common yardsticks, tech suppliers are working in the dark. By extension, the media and the public are told to take AV developers' word that self-driving cars are safe.

Here's the 64 million-dollar question: Are these developers finally acknowledging that the safety of autonomous vehicles must be a collaborative affair?

Figure: Riccardo Mariani (VP of industry safety at Nvidia)

EE Times recently asked Riccardo Mariani, vice president of industry safety at Nvidia, to walk us through each of the newly proposed IEEE standards. Mariani is an IEEE senior member and is the 2020 first vice president of the IEEE Computer Society.

Further, Mariani is in an ideal position to see advancements by a variety of working groups in IEEE. He is the chair of the P2851 and the co-chair of IEEE Special Technical Community. He is also the proposed chair of Industry Connection Activity.

We wanted Mariani to explain the changes in the AV industry's attitude, and how — or if — these new standards are supposed to work together.

First, Mariani reminded us that in addition to these emerging industry standards, a few private initiatives were launched in 2019.

For example, Safety First for Automated Driving (SaFAD) is led by Aptiv, Audi, Baidu, BMW, Continental, Daimler, FCA US LLC, HERE, Infineon, Intel, and Volkswagen. The group published a white paper last summer on how to build, test, and operate a safe automated vehicle.

Separately, SAE International, along with Ford, General Motors (GM), Toyota, and Uber, launched the Automated Vehicle Safety Consortium. The members are "working on the development of a series of safety principles for SAE Level 4 and 5 automated driving systems focusing on testing before and during operation of AVs on public roads; data collection, protection and sharing; and interactions between AVs and other road users," according to the consortium.

Standardization gaps

While some of these private consortia provide best practices, their work product tends toward higher-level guidance. "The market needs more detailed instructions" to implement safety in autonomous systems, said Mariani.

Those could be forthcoming in the emerging standards, which appear to be pursuing different aspects of safety for AI-based systems. But a formal process to investigate potential overlaps or gaps in the standards might have to wait until IEEE establishes a new working group called "Industry Connection Activity." Mariani is the proposed chair of Industry Connection Activity.

Article Source:

https://www.eetasia.com/news/article/AV-Safety-Standards-Set-to-Arrive-in-2020

10.2 System Life Cycle Processes Per ISO/IEC/IEE 15288 (2015)

ISO/IEC/IEEE 15288:2015 establishes a common framework of process descriptions for describing the life cycle of systems created by humans. It defines a set of processes and associated terminology from an engineering viewpoint. These processes can be applied at any level in the hierarchy of a system's structure. Selected sets of these processes can be applied throughout the life cycle for managing and performing the stages of a system's life cycle. This is accomplished through the involvement of all stakeholders, with the ultimate goal of achieving customer satisfaction.

ISO/IEC/IEEE 15288:2015 also provides processes that support the definition, control, and improvement of the system life cycle processes used within an organization or a project. Organizations and projects can use these processes when acquiring and supplying systems.

ISO/IEC/IEEE 15288:2015 concerns those systems that are man-made and may be configured with one or more of the following system elements: hardware, software, data, humans, processes (e.g., processes for providing service to users), procedures (e.g., operator instructions), facilities, materials and naturally occurring entities.

More Information

10.3 The IEEE Software & Systems Engineering Standards Committee (S2ESC)

The IEEE Software & Systems Engineering Standards Committee (S2ESC) is chartered by the IEEE Computer Society Standards Activities Board to codify the norms of professional software engineering practices into standards, including the standardization of processes, products, resources, notations, methods, nomenclatures, techniques, and solutions for the engineering of software and systems dependent on software.

S2ESC promotes the use of software engineering standards among clients, practitioners, and educators. S2ESC harmonizes national and international software engineering standards development, and promotes the discipline and professionalization of software engineering. S2ESC also promotes the coordination with other IEEE initiatives.

More Information

11. SOME DEFINITIONS TO CLOSE ON

11.1 Logistics Support

The supply and maintenance essential to proper operation of a system.

Source: James N. Martin, *Systems Engineering Guidebook*, CRC Press.

11.2 Measure of Effectiveness (MOE)

A property of a system, beyond requirements, whereby more or less of that property represents a better system solution in accordance with the values of one or more stakeholders. MOEs can be of any nature whatsoever, including operationally related properties such as performance, and programmatic properties such as cost and timing of availability for sell-off or use.

Source: Robert Halligan.

11.3 Mode (of Operation)

A function or a grouping of functions that has a significant relationship to a significant aspect of use, and has been designated for reference as a mode.

Source: Robert Halligan.

Editor's Note: PPI advocates, and adopts itself, a policy that definitions of engineering terms must be:

- consistent with dictionary definitions (although possibly more precise)
- in mainstream use
- conducive to efficient and effective engineering.

12. CONFERENCES AND MEETINGS

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

The featured event for this edition is:

The Annual Conference on Systems Engineering Research

March 19 - 21, 2020 – Redondo Beach (USA)

The Annual Conference on Systems Engineering Research (CSER) focuses on theoretical work in systems engineering and its translation to practical application. The conference includes research papers, plenary speakers, panels, and interactive sessions where attendees can engage in discussions and idea generation.

Since its inception, CSER has become the primary conference for disseminating systems engineering research and germinating new research ideas. The topic for this year's conference is:

'Recent Trends and Advances in Model-based Systems Engineering'

To register for the conference or to find out more, click here.

12.2 Host a Society of Women Engineers (WE) Event

WE Local is a Society of Women Engineers (SWE) program that was developed to bring the excitement and energy of SWE's annual conference – on a smaller scale – to SWE members' backyards around the globe through local conferences. WE Local brings together participants in all stages of their collegiate and professional journeys. Figure 1 depicts the highlights from SWE local events in 2019.

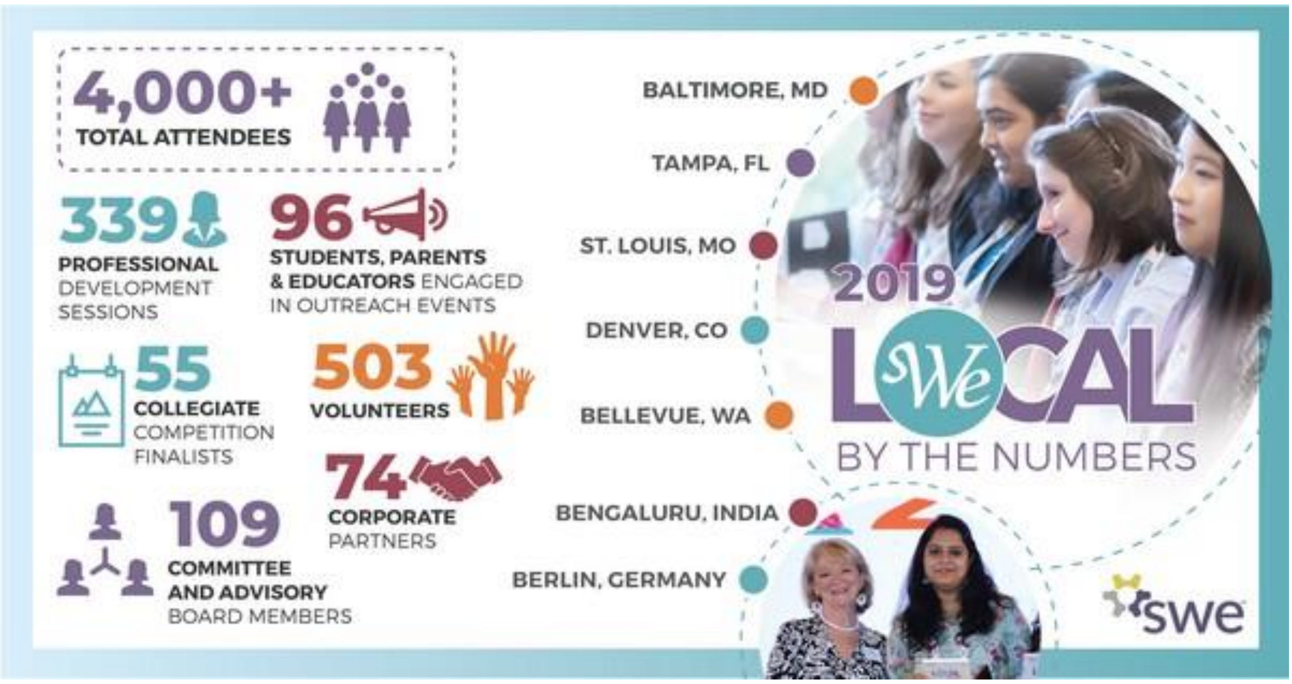


Figure 1: Highlights of SWE local events in 2019

Image Source

WE Local is now accepting applications for the 2020 WE Local conference season. Submissions may be received January 1-March 31, 2020. Applications will be reviewed April 1-June 30, 2020 with site visits taking place throughout the summer. The approval process and decision making is expected to be complete by the end of 2020.

Application Sample Questions

- Who may comprise the WE Local Host Committee?
- In a few sentences, explain the reasons why you are recommending this location.
- List any universities/SWE Collegiate Sections in close proximity of the proposed location.
- Is there a proposed WE Local conference venue that can host 500 attendees including a 100 attendee SWENext Outreach event?

Host City Criteria

Prior to applying, WE Local has outlined criteria and a set of questions for conference site selection to include the following, but not limited to, items:

1. Universities (how many universities are within a three-hour distance so undergrad/graduate students can attend?

2. Conference venue (is there appropriate function space/square footage?)

3. Accessibility (is the host city easily accessible and affordable by plane, train or automobile?)

4. Section/region members (did a member nominate this city?)

5. WE Local will not select a host city with a (high) gambling or alcohol presence due to the demographics of an underage population

For more questions regarding WE Local city selection or to register your city's interest in hosting a WE Local event, contact welocal@swe.org.

13. PPI AND CTI NEWS

13.1 Bijan Elahi's Book a 'Runaway Success'

We are delighted to celebrate the success of PPI course presenter Bijan Elahi's book titled 'Safety Risk Management for Medical Devices'. In a message from Bijan's publisher, Bijan's book was acknowledged to have sold five times as fast as similar books on the market in its first year. Elsevier thanks Bijan for adding much value to Elsevier's Biomedical Engineering list. Well done, Bijan!

Bijan Elahi is an expert on a world scale in risk management and systems engineering. He was presented with Educator of the Year Award by the International System Safety Society. Bijan presents the MDRM 3-day course all over the world. Visit the MDRM3D page for more information, or to register for an upcoming course.

https://www.ppi-int.com/training/mdrm3d/

13.2 René King Visits Australia

Although the PPI headquarters are in Melbourne, Australia, as the 'I' in PP*I* suggests, Project Performance International is a global company in that it serves and employs people from all around the world. In December 2017, René King who hails from Johannesburg, South Africa joined the PPI family and has worked remotely and travelled substantially in her role as senior engineer and CTI Managing Director. In February 2020, after more than 2 years of working via correspondence with the team in Melbourne, René finally met with the Melbourne team and had the pleasure of working in the office with her teammates, balancing hard work and fun the PPI way! Below is a photograph taken after an evening of mini-golf and an enjoyable dinner with the team.

Left to right: Caitlyn Liu, René King, Robert Halligan, Jess Cuffe, Kevin Blazé, Helen Hele, (front) Karen Deveson, John O'Kelly, Bridget Petty, (back left to right) Benjamin Bryant, Michael Petty, Helen Chen, Rebeca Carneiro.

13.3 SETDB Prototype Demo at the INCOSE IW 2020

A prototype version of the Systems Engineering Tools Database (SETDB) was demonstrated to the participants in the INCOSE International Workshop (INCOSE IW) held at Torrance, CA, USA over 25 to 28 January, 2020. The IW is an annual gathering of the 50 or so INCOSE Working Groups and leadership committees that works towards INCOSE's mission to make the world a better place through systems engineering.

The SETDB is being jointly developed by PPI and INCOSE under a MOU between the organizations. Release in fully developed form is scheduled for July 2020. The SETDB will be accessible by logon from PPI's Systems Engineering Goldmine, and to logged-in INCOSE members from the INCOSE website.

The picture below shows PPI Managing Director Robert Halligan and PPI SETDB Project Manager René King beside a SETDB screen display.

Experience the first Functional Prototype of the Systems Engineering Tools Database
We welcome your feedback !
INCOSE

14. PPI AND CTI EVENTS

On-site systems engineering training is delivered worldwide throughout the year. Below is an overview of public courses. For a full public training course schedule, please visit https://www.ppi-int.com/course-schedule/

Systems Engineering 5-Day Courses

Upcoming locations include:

- London, United Kingdom (P006-803)

  02 Mar – 06 Mar 2020

Requirements Analysis and Specification Writing 5-Day Courses

Upcoming locations include:

- London, United Kingdom (P007-502)

  18 May – 22 May 2020

Systems Engineering Management 5-Day Courses

Upcoming locations include:

- Berlin, Germany (P1135-190)

07 Sep – 11 S 2020

Requirements, OCD and CONOPS in Military Capability Development 5-Day Courses

Upcoming locations include:

- Melbourne, Australia (P958-62)

  16 Mar – 20 Mar 2020

Engineering Successful Infrastructure Systems (ESIS5D)

Upcoming locations include:

- Detroit, MI USA (P2005-5)

  15 Jun – 19 Jun 2020

Architectural Design 5-Day Course

Upcoming locations include:

- Adelaide, South Australia (P1768-27)

  04 May – 08 May 2020

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

Upcoming locations include:

- Bristol, U.K. (C002-96)

  27 Apr – 01 May 2020

Medical Device Risk Management 3-Day Course

Upcoming locations include:

- Berlin, Germany (P1848-8)

  07 Apr – 09 Apr 2020

Other training courses available **on-site** only include:

- Project Risk and Opportunity Management 3-Day
- Managing Technical Projects 2-Day
- Integrated Product Teams 2-Day

- Software Engineering 5-Day.

15. UPCOMING PPI PARTICIPATION IN PROFESSIONAL CONFERENCES

PPI will be participating in the following upcoming events. We support the events that we are sponsoring, and look forward to meeting old friends and making new friends at the events at which we will be exhibiting.

Institute of Industrial and Systems Engineering Annual Conference 2019

(Exhibiting)

Date: 30 May – 02 June, 2020

Location: New Orleans (LA), United States

The INCOSE International Symposium 2020

(Exhibiting)

Date: 18 – 23 July, 2020

Location: Cape Town, South Africa

Kind regards from the PPI SyEN team:

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

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

René King, Managing Editor, email: rking@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 12 9 9780 3490

Tel UK: +44 20 3608 6754

Tel USA: +1 888 772 5174

Tel China: +86 188 5117 2867

Web: www.ppi-int.com

Email: contact@ppi-int.com

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