PPI SyEN

SYSTEMS ENGINEERING NEWSJOURNAL

EDITION 105 | OCT 2021

Power through Models

AGILE METHODOLOGY PART 2
System architecture models

RETURN ON INVESTMENT
Enhancing projects with SE

SE TOOLS TAXONOMY
Data organisation of SE tools

IMPROVING SOPs WITH MBSE
Evaluating procedure efficacy

A PROJECT PERFORMANCE INTERNATIONAL PUBLICATION | PPI-INT.COM

Project Performance International
2 Parkgate Drive
Ringwood, Vic 3134 Australia
Tel: +61 3 9876 7345

Tel Brazil: +55 12 9 9780 3490
Tel UK: +44 20 3608 6754
Tel USA: +1 888 772 5174
Tel China: +86 188 5117 2867

www.ppi-int.com
contact@ppi-int.com

PPI SyEN (PPI Systems Engineering Newsjournal) is published monthly.

Archived editions and subscriptions to future editions are available for free at: https://www.ppi-int.com/syen-newsjournal/

WELCOME

Just like that we're into the fourth quarter of 2021! In the Featured Articles of this edition, we see a multitude of mechanisms for applying systems approaches for the purposes of development, analysis or management. With Juan Navas' article we read about the use of architectures in enabling agility in the development of a drone, with Wioletta Kowalczyk's article overviewing the work of Eric Honour we see the value of using systems thinking in analyzing the ROI of SE. With Robert Halligan's article, we read about the benefit of systems thinking in developing not only on the technical products, but also in developing the systems that manage the technical products. The case for the versatility of systems approaches applied beyond the confines of development of technology systems is cemented in Steven Dam's article on using modeling approaches to map out Standard Operating Processes.

In applying systems approaches, one can't take a stick and swing within a 5m radius of themselves without hitting someone who will utter the term 'MBSE' (try this at your own risk). Many of you will know that model-based systems engineering (MBSE) is one of the key vehicles to applying systems engineering to complex systems. Sad but true that I've seen many instances of models being applied without much systems thinking to the construct of the models of themselves – you've heard the claim, 'all models are wrong but some are useful' by George E. P. Box.

All models are wrong, because it is inherently impossible to capture every aspect of the system and every aspect of the environment completely and correctly in one model. Also, the model becomes useless when we fail to record the assumptions made when constructing the model, when we don't acknowledge the limitations of the language that we used to construct the model, when we don't fully understand the modeling language or the tools that we use to construct the models (mostly tools are not intelligent, they just execute instructions – including incorrect instructions - perfectly) and worst of all, when we don't understand the system that we are attempting to model. All of this can be tackled through the power of systems engineering. Systems engineering, not as a name but as the approach to development that relies on separation of problem definition and solution description, creating both very well, with a focus on reducing risk and maximizing value to stakeholders. In my books, MBSE could easily have been called, 'SBME' systems-based model engineering where systems engineering is the basis for constructing models that are used in the engineering of systems. I hope you enjoy this edition!

Regards,

René

Managing Editor, PPI SyEN

PPI Systems Engineering Newsjournal (PPI SyEN) seeks:	**PPI defines systems engineering as:**
➢ To advance the practice and perceived value of systems engineering across a broad range of activities, responsibilities, and job-descriptions ➢ To influence the field of systems engineering from an independent perspective ➢ To provide information, tools, techniques, and other value to a wide spectrum of practitioners, from the experienced, to the newcomer, to the curious ➢ To emphasize that systems engineering exists within the context of (and should be contributory toward) larger social/enterprise systems, not just an end within itself ➢ To give back to the Systems Engineering community	*an approach to the engineering of systems, based on systems thinking, that aims to transform a need for a solution into an actual solution that meets imperatives and maximizes effectiveness on a whole-of-life basis, in accordance with the values of the stakeholders whom the solution is to serve. Systems engineering embraces both technical and management dimensions of problem definition and problem solving.*

PPI SyEN FORUM

Selected correspondence from readers, authors, and contributors

PPI SyEN FORUM offers the opportunity for feedback and discussion on topics around systems engineering – especially those that have been (or should be) addressed in PPI SyEN.
Please send your email to **PPISyEN@ppi-int.com**

Defeating Complacency – Obstacle to Mission Success by Scott Jackson, PhD

On the day after the successful SpaceX mission, I ran into an old friend. "How are you?" he asked. Knowing that he intended his question to be a common courtesy, I did not give him a courteous answer. "Depressed" I answered. "Why? He replied. "I am depressed about the failure of the SpaceX mission." I said. "But" he replied, "the mission was a total success." "I know," I replied, "I am talking about the one around the corner. It has all the potential for failure."

As a part of my career in systems engineering, I have studied all the major system disasters, what caused them, how they can be prevented, and how can the system recover from a disruption. Obvious accidents include Columbia, Challenger, and Tenerife, the deadliest accident in aviation history. Systems that recovered include Apollo 11 and Apollo 13.

Returning to the topic of failures, Leveson [1] tells us that the two most common causes of failure are optimism and complacency. Optimism is simply the belief that a goal can be achieved that is beyond the capability of the current system. Complacency is more complex. It is the inattention to safety and other safeguards of success.

So how are these obstacles overcome and show us a path to success? That is difficult to say, and that is why I was depressed. It will be remembered that Columbia and Challenger were only 17 years apart. One would think that the lessons from the first disaster would make the second one virtually impossible. But that was not the case. What happened in between? Did complacency set in?

Following the splash-down of SpaceX the airwaves were packed with self-congratulation. The mood was euphoria. The question that was asked many times was: What is the main obstacle to doing it again? The answer was always the same: cost. It was pointed out that a new propulsion system would bring the cost down. With all respect, I cannot agree that cost is the main obstacle. I vote for complacency.

So, the real question is: how do you control complacency? There are no easy answers. One approach is to find a complacency metric, that is, a way to measure complacency and determine how close you are to having mission success. Without going into details, I can report that there are no clear complacency metrics. At least I have not found one.

What you can do is assure that you are following all the standard safety design procedures, such as the use of redundancy. But even there you have to make sure your process is complete and thorough. You cannot do that if you are suffering from complacency.

I don't want to leave you with the message that success is not possible. It certainly is, but you cannot relax.

Defeating Complacency

Among the various sources for managing or defeating complacency is Eikenberry [2]. Eikenberry provides six rules for defeating complacency. The first thing you might notice about these rules is that they are highly qualitative. They are easy to understand, but they provide no measurable way of knowing whether you have defeated complacency or not. Here are the six rules:

1. Recognize it (complacency). This one is intuitively correct. It is difficult to defeat complacency if you don't recognize what you are trying to defeat. Eikenberry [2] gives some clues on how you can recognize complacency. For example, are you (or your team members) doing less of things that have led to success in the past?
2. Put it in context. Eikenberry [2] recommends that you put the task in the context of success and remind your team of it. The SpaceX Mission described above is an example.
3. Set new goals. So, what is better than a successful space launch? The answer is a space launch with no hitches.
4. Keep your purpose clear. A successful launch is a clear goal. The measure of this goal is reliability. Check and re-check of reliability will meet this rule.
5. Create healthy competition. Eikenberry [2] suggests that you create goals to achieve that are greater than last year's. This is a difficult rule, but it has the look and feel of a beneficial step.
6. Remember history and human nature. Eikenberry [2] reminds us that complacency is part of the human condition. That is, people will naturally look for the easy way to do things. So, when you see it, focus on ways to overcome it.

In summary, defeating complacency is not easy, and there are no clear guidelines how to defeat it. Yet, its importance is not to be understated as pointed out by Leveson [1]. Still yet, defeating it will consume all your mental and physical energy. Past catastrophes will be a reminder how important it is, and the potential for future catastrophes will be ever present if no action is taken.

[1] N. Leveson, Safeware: System Safety and Computers. Reading, Massachusetts: Addison Wesley, 1995, p.434.

[2] K. Eikenberry. "6 Ways to Defeat Complacency." Bud to Boss. (accessed 29 September, 2021).

What are your thoughts on Scott Jackson's position on complaceny in complex system development? We'd love to hear from you!

SYSTEMS ENGINEERING NEWS

Recent events and updates in the field of systems engineering

INCOSE and SAE International Announce Joint Membership Agreement

INCOSE and SAE International have announced a joint membership model, whereby members of both organizations can pay discounted membership fees for each organization. INCOSE will offer a 15% discount for joint membership. SAE International will offer a 15% discount for joint membership. Joint membership is available to regular, senior, and student members of INCOSE and SAE International, with some special exclusions.

Both organizations are very active in advancing the systems engineering discipline within their respective spheres of concern.

More information:

• INCOSE: helpdesk

• SAE International: membership

Object Management Group RAAML Beta Version 1.0 Specification Defines Extensions to SysML

In September, international technology standards organization Object Management Group® (OMG®) issued the Beta version 1.0 of its Risk Analysis and Assessment Modeling Language (RAAML) specification. The standard defines extensions to the SysML open-source systems modeling language needed to support safety and reliability analysis.

A group of industry experts at the OMG has been working since 2016 to define the new specification, which provides the necessary capabilities.

"The need for a standardized Unified Modeling Language (UML) profile/library for addressing safety and reliability aspects emerged long ago," said Kyle Post, Systems Safety Technical Leader at Ford Motor Company and a member of the team that designed and wrote the standard. "Working group members have seen multiple commercial-grade model-based safety and reliability solution implementations being developed during the recent years and successfully used in practice. We drew upon that experience and expertise in the design of this spec."

Their new RAAML Beta Version 1.0 specification defines extensions to SysML needed to support safety and reliability analysis. It provides the modeling capabilities for tool vendors to build safety and reliability modeling tools that provide traditional representations (e.g. trees, tables, etc.) while using a modern model-based approach.

The RAAML specification can provide the foundation for conducting various safety and quality engineering activities including safety and reliability analysis methods. Besides the method support, linkages to the SysML model-of-interest are provided, enabling integration with and traceability to the analyses.

The spec describes the RAAML core concepts and shows:

That simple concepts are powerful enough to unite all safety and reliability information across a variety of analysis methods,

The approach to automating several safety and reliability analyses, which is built on leveraging existing SysML functionalities to ensure that the profile and library is usable with existing tooling,

Specific safety and reliability analysis methods and application domains that are supported, including Failure Mode and Effects Analysis (FMEA), Fault Tree Analysis (FTA), System Theoretical Process Analysis (STPA), Goal Structuring Notation (GSN), ISO 26262 Road Vehicles Functional Safety, and Extension Mechanisms that are typically needed by the industry to apply the specification in practice.

The most current version of the document is located here. OMG specifications address middleware, modeling and vertical domain frameworks. All OMG specifications are available from the OMG website.

SpesML - SysML Workbench for the SPES Methodology

The German Chapter of INCOSE (GfSE) conducted a webinar on 6 October, 2021 under the subject title. A summary of this significant work follows.

Scientific research results achieved together with partners from industry and academia by the Technical University of Munich show that the introduction of model-based system development of cyber-physical systems into the industrial development process offers a wide range of benefits, but requires a readjustment of the development. This readjustment concerns the development method, artifacts, tools and organization. So far, the industrial introduction of MBSE often uses the modeling language SysML as a quasi-standard, which is firmly established in practice by tool providers and standardization organizations. However, the analysis of the introduction projects shows that SysML is currently often used without a consistent development methodology, which (i) specifies which models are created and how these models are interrelated, and (ii) provides a precise understanding of the diagrams. Thus, crucial potential offered by MBSE remains unused, because ultimately the well-coordinated selection of methodology, modeling language and modeling tool is a crucial factor for a successful application of MBSE in industrial practice. The project "SysML Workbench for the SPES Methodology" (SpesML), funded by the German Federal Ministry of Education and Research (BMBF) and involving a total of eleven partners from industry and research as well as 2 active associated partners, aims to close this gap and to show ways for a successful implementation of the MBSE methodology in industry.

With the Software Platform for Embedded Systems (SPES), an end-to-end methodology for MBSE has been developed. On this basis, a SysML profile with a precise semantics that can be used for automation such as automated analysis and simulation, is defined in SpesML. This gives the widely-used modeling language SysML a methodological foundation and thus opens the way to comprehensive MBSE.

Key messages in the webinar were:

- Challenges in the implementation of MBSE
- Introduction into basic concepts of the SPES methodology
- Implementation of initial concepts in SysML within the SpesML Workbench based on MagicDraw
- MBSE maturity model as a proven approach for practical MBSE implementation.

Join GfSE (German Chapter of INCOSE) here

Tools: BigLever PLE Studio

BigLever Software has released the first of a three-part series "The Art & Science of Feature-based PLE" to subscribers to its PLE Insight newsletter. The series specifically concerns onePLE Studio, a component of BigLever's solution for Feature-based Product Line Engineering (PLE).

The paper and BigLever's home page link to additional resources such as PLE blogs, news releases, webinars, INCOSE's PLE Primer and ISO/IEC 26580:2021.

CAPELLA DAYS 2021 - November 15-18, 2021 (Free Online Event)

Capella Days is the annual event that regularly brings together the MBSE community of Arcadia and Capella practitioners, organized by Obeo, in partnership with Thales.

After a warm-up on the first day to start this event, attendees will learn from concrete case-studies of Arcadia and Capella in different industries: space, energy, railway, electronics, air traffic management, healthcare, education. Attendees will benefit from the experience of industrial adopters (such as Siemens or Thales) who have deployed an MBSE approach with Arcadia and Capella on their projects.

AGENDA

Day 1: Monday 15th of November

3:35 pm CET How I pack my suitcase

4:15 pm CET Capella Warmup - Introduction to CAPELLA/ARCADIA and NASA Systems Engineering Handbook: Modeling overview with the HUBBLE Space Telescope

Day 2: Tuesday 16th of November

4:05 pm CET The long way from Bid to project… supported by Capella

4:45 pm CET A STEP towards Model-based: Case Study covering Conceptual Design of a Fusion Power Plant

5:25 pm CET Using MBSE to integrate engineering undergraduate courses curriculum

Day 3: Wednesday 17th of November

4:05 pm CET Exploring the various roles of MBSE in the digital thread

4:45 pm CET Enhancing CubeSat design through ARCADIA and Capella: a concrete application

5:25 pm CET Where to start with MBSE when thousands of system requirements are already defined

Day 4: Thursday 18th of November

4:05 pm CET An example of model-centric engineering environment with Capella and CI/CD

4:45 pm CET Using MBSE and Capella to improve regulatory certification

5:25 pm CET How much time does modeling take? Experiences from modeling without experience

More about the event here: Capella Days 2021

Registrations: Capella Days 2021 - Register

Tom Sawyer Releases Perspectives 10.0

Tom Sawyer Perspectives is a low-code graph and data visualization and analysis development platform. Integrated design, preview interfaces and extensive API libraries allow developers to create custom applications that intuitively solve big data problems. Features include advanced edge labeling, precise shape clipping, port and connectors controls, and incremental layout to see the superstructure of data and produce visual graphs for understanding by domain experts and stakeholders alike.

This release includes new features, improvements, and architectural changes including:

- Interactive graphic schema editor
- Automatic integrator bindings for graph databases
- Native graph in-memory model
- Schema code generator tool
- Dynamic rulesheet domains
- Dynamic inspectors.

More information: Tom Sawyer Perspectives

INCOSE Loss-Driven SE Initiative (LDSE)

The International Council on Systems Engineering (INCOSE) in 2020 formed a Working Group with the objective of identifying commonality and synergy among loss-driven specialty areas. LDSE domains include reliability, sustainability, survivability, risk management, resistance, resilience, agility, safety, and security. The LDSE initiative aims to unify these loss-driven specialty areas with one another, and to integrate loss-driven activities into the normal systems engineering lifecycle involving development and use of systems. LDSE was featured via a number of articles on the topic in the January 2021 edition of INSIGHT, INCOSE's quarterly practitioners' magazine.

Contact John Brtis (jbrtis@johnbrtis.com) for more information.

“

SysML 2 will be a game-changer .

ROBERT HALLIGAN

CONFERENCES, MEETINGS & WEBINARS

Upcoming events of relevance to systems engineering

INCOSE Symposium 2022 - Call for Submissions

The INCOSE International Symposium is the premier international forum for systems engineering and systems approaches. INCOSE IS2022 will be held in Detroit, MI USA as a hybrid event over 25-30 June 2022. Participants network, share ideas, knowledge and practices, and learn more about the most recent innovations, trends, experiences and issues in all aspects of systems engineering from world-class thought leaders in the field. The theme for the INCOSE International Symposium 2022 is The Power of Connection.

Key dates for submissions are:

Paper, Panel, Tutorial Submission: November 14, 2021
Notification of Acceptance: February 20, 2022
Final Paper, Panel, Tutorial Submission: March 20, 2022

Submission links are:

Paper submissions: https://easychair.org/conferences/?conf=is2022papers
Panel and roundtable submissions: https://easychair.org/conferences/?conf=is2022panels
Tutorial submissions: https://easychair.org/conferences/?conf=is2022tutorials

More details on IS2022 can be found here.

2021 PDMA Innovators Conference and JPIM Research Forum Goes Virtual

After careful consideration, the Product Development Management Association (PDMA) Board of Directors and Conference Committee has decided to cancel the in-person PDMA Innovators Conference and JPIM Research Forum scheduled for November 13-16 in Baltimore. The organizers will pivot to a virtual event — to be held at a date to be announced soon. Previously registered participants will receive a full refund for their conference registration.

More details here.

Upcoming AI Events

The Systems Engineering community is seeing a burst of interest in the application of Artificial Intelligence (AI) to systems and products. Here are two near-term opportunities to consider if you want to grow your understanding and shape the future of AI in the engineering of intelligent systems that deliver increased value to your stakeholders.

Progress in Test and Evaluation of AI-enabled Systems in the DoD: On 27 October, the SERCTALKS webinar series on Test and Evaluation will focus attention on progress that has been made in the test and evaluation of AI-enabled systems. Dr. Yevgeniya "Jane" Pinelis of the DoD's Joint AI Center (JAIC) will highlight the differences in the science, practice, skills and infrastructure required to evaluate AI-Enabled Systems (AIES). Register here.

Systems Engineering Pathways to AI: INCOSE President, Kerry Lunney, will host this virtual three-hour AI mini-event on 9 November. SE and AI experts will address how SE is moving forward with incorporation of AI right now, and how systems engineering can be applied to AI-based applications and systems. Topics will include how to verify AI, understanding uncertainty quantification, where to find the data to progress AI, and how to break down the barriers from prototype to enterprise AI adoption. Check here for details.

INCOSE 7th Annual Systems Engineering in Healthcare Conference

The INCOSE Healthcare Working Group invites you to participate in the 7th Annual INCOSE Healthcare Systems Conference (2021). The conference will be fully virtual and will be free to both INCOSE members and non-members.

The theme for the conference is: "Advancing the Practice of Systems Engineering in the Healthcare Industry".

The conference will be held on Friday afternoons from 1:00 pm to 5:00 pm starting on October 29 and running through November 19. The conference sessions are as follows:

Friday, October 29, 2021, 1-5 pm USA Eastern Time

Session 1: Requirements Tools to Meet FDA Design Control Requirements

Friday, November 05, 2021, 1-5 pm USA Eastern Time

Session 2A: Systems Responses to COVID-19 and Future Pandemics

Session 2B: User View on Requirements Tools

Friday, November 12, 2021, 1-5 pm USA Eastern Time

Session 3A: Linking SE Models with Simulations for Device Development

Session 3B: System Approaches to Tracking Pandemic Responses

Friday, November 19, 2021, 1-5 pm USA Eastern Time

Session 4: Lean Healthcare Systems Engineering Applications

The intended audience is systems engineers, product developers and testers, and leaders of organizations developing complex healthcare products and services, from large Healthcare IT systems to medical devices to healthcare delivery organizations. Attendees from other domains interested in learning about systems methods in healthcare are welcome.

Details here.

2021 MBSE Cyber Experience

From 7-10 November, 3DS is hosting the MBSE Cyber Experience Symposium at the Marriott Courtyard in Allen, Texas. The event is geared to tracking advancements in technology and innovations in the world of MBSE including developments in SysML, PLM, PLE, Enterprise Architecture, Business Architecture, BPMN, Concept Modeling and Ontology. The call for presentations is now open! Apply to present or register your attendance here.

FEATURED ARTICLES

Architecture Models as Enablers of Agility

by Juan Navas, Thales Corporate Engineering

Email: juan.navas@thalesgroup.com

Based on the paper **Models as enablers of agility in complex systems engineering** presented at the INCOSE International Symposium 2020 by Juan Navas, Stephane Bonnet, Guillaume Journaux and Jean-Luc Voirin.

Abstract

This series of articles follows the engineering teams of the PythaDrone project during their endeavor of developing a drone-based product that addresses multiple market segments. The focus is on the use of system architecture models as the basis for several technical, management and organizational activities, in a context in which a company implements agility in its engineering processes. Although the PythaDrone project and the Pythagoras company are fictive, the practices described here are those put in place in some of our projects and business units.

Episode 2 – Run!

Previously on Episode 1

In Episode 1 of this series of articles, we introduced the fictive PythaDrone team in the also fictive Pythagoras company. The PythaDrone team is in charge of developing a lightweight drone-based product that will address different markets: agriculture, aircraft exterior inspection, and public security enforcement. The team is implementing both agility and MBSE practices in order to react faster to changes in the customers' expectations and ensure the consistency of the design.

We followed them through their initial "warm-up" activity, in which they built their engineering strategy and the first vision of their product. In this Episode 2 we will situate ourselves later in time, and we will focus on the Systems Architecture team while specifying a subset of the features to be developed in further iterations.

Defining the scope of design for Iteration 3

The Architecture team has been releasing incremental versions of the product architecture at each product-level iteration (i.e. every 3 months). The project started 6 months ago and the team is preparing the third iteration at the product-level. Until now the design has been focused on the visualization capabilities of the product in order to get feedback on user experience (UX) as soon as possible from operators. The table below summarizes the progress status of design, development and test per capability.

Capability	% designed	% developed	% validated
Manually pilot the drone	40%	20%	10%
Automatically follow a flight plan	40%	20%	20%
Manually acquire data	30%	10%	10%
Automatically acquire data	60%	40%	40%
Visualize data during mission execution	70%	70%	30%
Visualize data after mission execution	100%	50%	40%
Analyze data during mission execution	0%	0%	0%
Analyze data after mission execution	50%	20%	0%

During the “warm-up” activity of this iteration 3, the team identifies which product capabilities would be designed in the next 3 months. They do so by analyzing the product delivery milestones and the availability of resources to develop them, among other factors (cf. Episode 1 for details on how to run a “warm-up” of an iteration). The conclusions of the warm-up are:

- The team will focus on the Manually pilot the drone and Manually acquire data capabilities, which design has already started but needs to be completed soon in order to provide a first prototype to future customers
- The team will also have to update the design of the Visualize data after mission execution, which design was thought to be complete, but will need to be modified following the feedbacks from the UX specialists
- For the other capabilities, the software and tests teams already have enough to do with the past releases of the design

Now the team is ready to start the “Run” activity of the iteration.

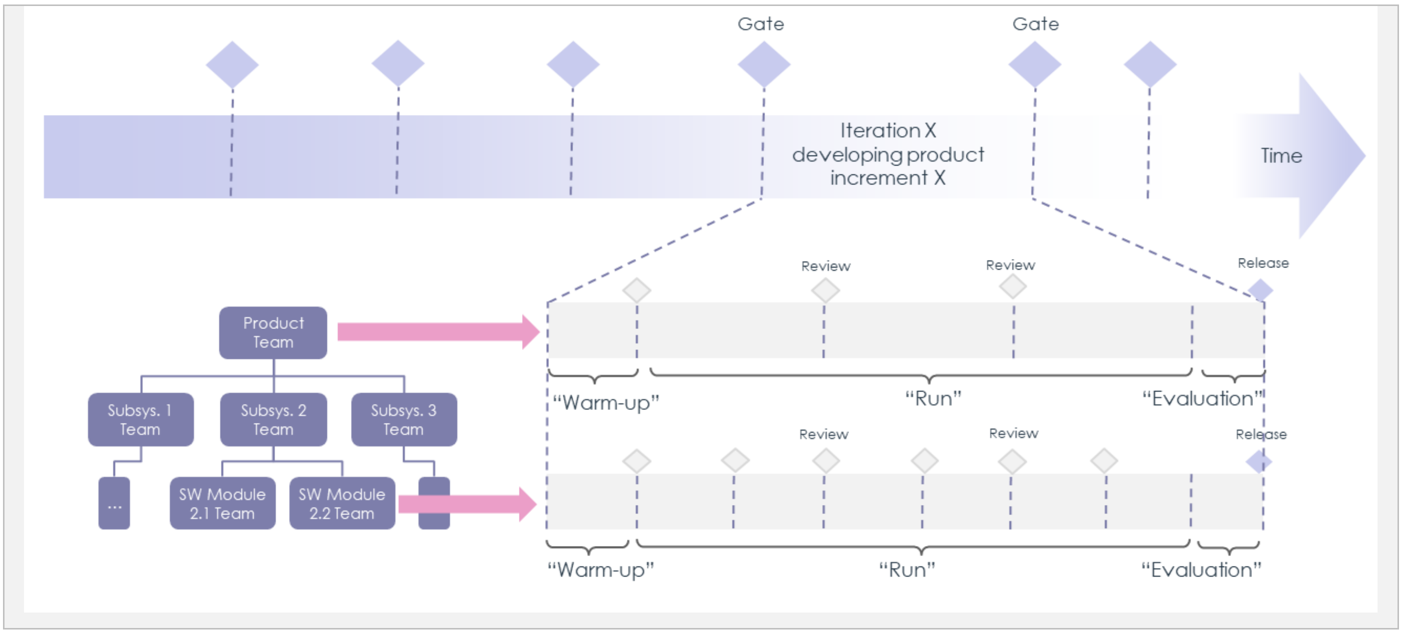
Run

“Run” is another kind of activity that you perform in an iteration, along with “Warm-up” and “Evaluation”, as presented in the previous episode. Using the sports analogy, in an iteration you need to prepare yourself (warm-up) before performing a continuous and strong effort (run), and if you want to improve you need to measure and analyze your performance (evaluate).

Warm-up Run Evaluate

For an engineering team, the “Run” activity is made of shorter iterations or sprints, aiming at implementing product features. This includes (non-exhaustive list) the detailed definition of product functions, the development of the system and subsystems' architecture; the development of the software and hardware implementing expected behavior; and the verification and validation of what will be delivered to the customer.

Sprints tend to have the same period, but their number inside an iteration may vary according to the life cycle of the project and the organization. For instance, teams devoted to product components may do 6 beats inside a 3-month iteration (one each 2 weeks), whereas product-level architecture teams may do 3 (one per month), as illustrated in the figure below.



Designing product capabilities

The MBSE approach that the Pythagoras company implements is based on three pillars: a set of *concepts* that are known and understood by the engineers in the company, a *method* for performing MBSE, and *engineering tools* that are consistent with the method. The concepts and methods are based on the Arcadia method [VOIRIN 2017], which has been slightly adapted to match the vocabulary of the company; the MBSE tool is the open source tool Capella.

Designing product capabilities with Arcadia and Capella MBSE

Arcadia is a model-based architecture method devoted to systems, software and hardware engineering. It allows architects to capture the needs of the stakeholders (customers, users, relevant regulations …), define and share the product architecture with all engineering stakeholders, validate the design early and justify it. Arcadia has been proven especially effective when designing products with strong constraints to be reconciled such as cost, performance, safety, security, reuse, consumption, weigh, etc. It has been applied in a large variety of contexts over the last ten years.

Arcadia defines a set of perspectives, in accordance with the [IEEE 1220] standard. A perspective is a way of analyzing the product. The method encourages the architect to adopt different perspectives and to run engineering tasks for each of them. Arcadia proposes four main perspectives: two of them aim at reaching a shared comprehension of the context in which the product will evolve and the needs that it shall satisfy (Operational Analysis and Systems Needs Analysis), the two others aim at defining the architecture design, both at conceptual and physical levels.

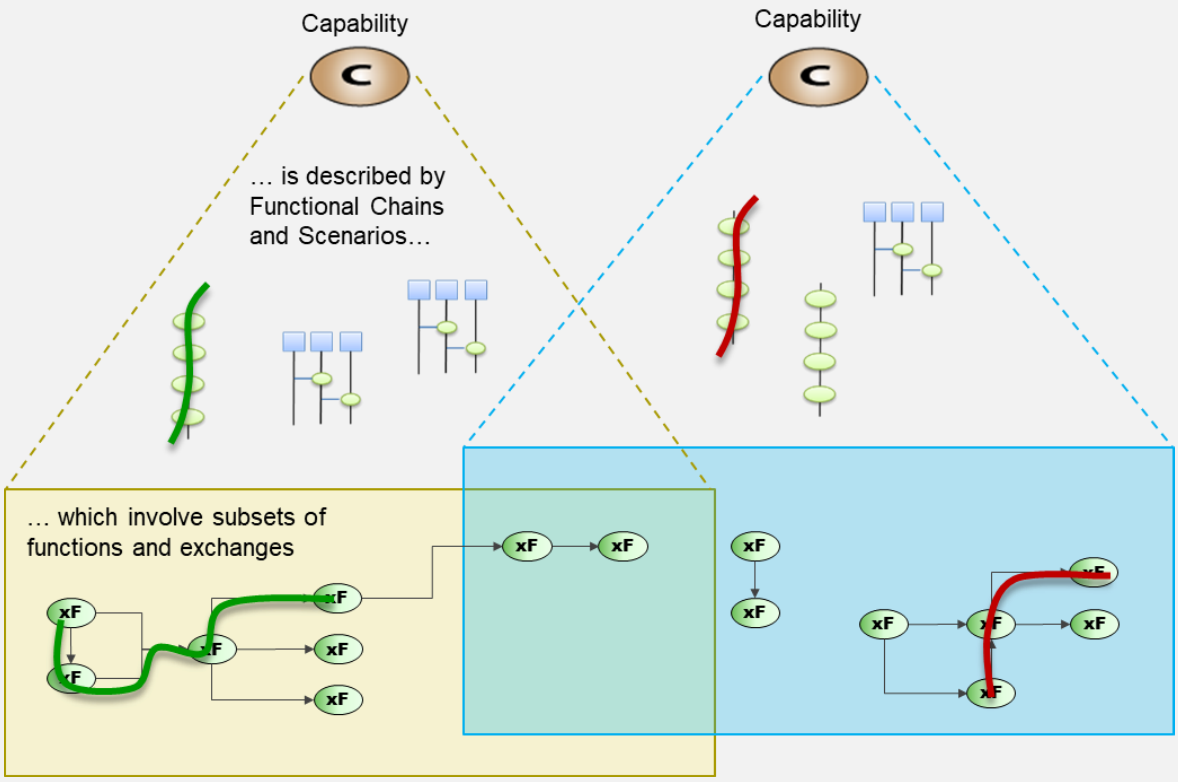
	Perspective	Objective
NEED & CONTEXT PERSPECTIVES	**Operational Analysis**	What the stakeholders need to accomplish
	System Needs Analysis	What the system has to accomplish for the stakeholders
SOLUTION PERSPECTIVES	**Conceptual Architecture**	How the system will work to fulfill expectations
	Finalized Architecture	How the system will be developed and built

One of the main concepts in Arcadia is the *Capability*, which represents the ability of the product to provide a service that will support the satisfaction of the stakeholders' expectations. As an illustration, the PythaDrone product, through offering the service of manual piloting, will support a farmer at surveying his agricultural field.

Capabilities are user-oriented, and hence they are a good vehicle for discussing the product with the customers. However, the service represented by a Capability usually needs to be provided in different ways depending on the usage contexts of the product. Arcadia defines two concepts that describe Capabilities in different usage contexts: *Functional Chains* and *Scenarios*. They both describe how the product, the product's components and/or the entities external to the system shall be involved in providing a service. Scenarios are specific in that they can describe dynamic, time-related interactions, whereas Functional Chains do not.

Functions represent what the product, product components and external entities are expected to do. *Functional Exchanges* represent the dependencies between Functions. Both are involved in different ways in different usage contexts (Functional Chains and Scenarios). In addition, a Function can contribute to more than one Capability, i.e. it can be involved in different usage contexts of different Capabilities. To refine Functions and Functional Exchanges, textual requirements are associated to them: textual requirements and model requirements complement each other, as presented in [BONNET, 2019].

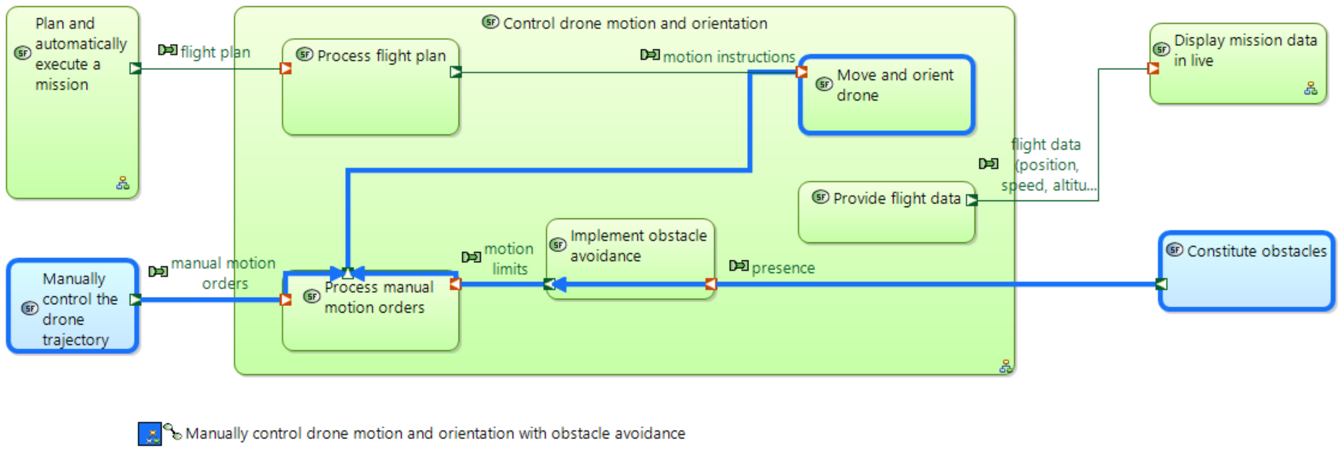
The Figure below summarizes the relationship between Capabilities, Functional Chains and Scenarios, and Functions and Functional Exchanges.



We will take a closer look to the Manually pilot the drone capability (from this point on, the name of model elements shown in the diagrams are written in grey). Drone piloting refers to controlling the motion and orientation of the drone. One of the variabilities being considered at this point is whether to offer the ability to automatically avoid obstacles during navigation. Marketing identified markets for which it would be an attractive feature (security), but also markets in which it would be a superfluous one. The Architecture team decides to design both and ensure that the feature can be removed if necessary.

Let us zoom in on one of the Functional Chains of this capability: Manually control drone motion and orientation with obstacle avoidance. The Architecture team decides to work on the System Needs Analysis perspective to formalize the functions that the product as a *whole* shall perform in order to provide the capability. The dataflow below is one of the diagrams shared with other engineering teams, it shows:

- Three major functions of the product: Plan and automatically execute a mission, Control drone motion and orientation, and *Display mission data in live*; shown in green
- Two functions that have been allocated to external entities: Manually control the drone trajectory which has been allocated to the Operator, and Constitute obstacles, which merely represents the existence of external obstacles and may carry the expected characteristics of those; they are shown in blue
- The result of the functional analysis of the Control drone motion and orientation function, which has been decomposed in 5 sub functions
- The dependencies between these functions
- The functions and functional exchange that are involved in the functional chain Manually control drone motion and orientation with obstacle avoidance, which are either highlighted in blue or implicit by the fact that their input / output functional exchanges are involved in it.



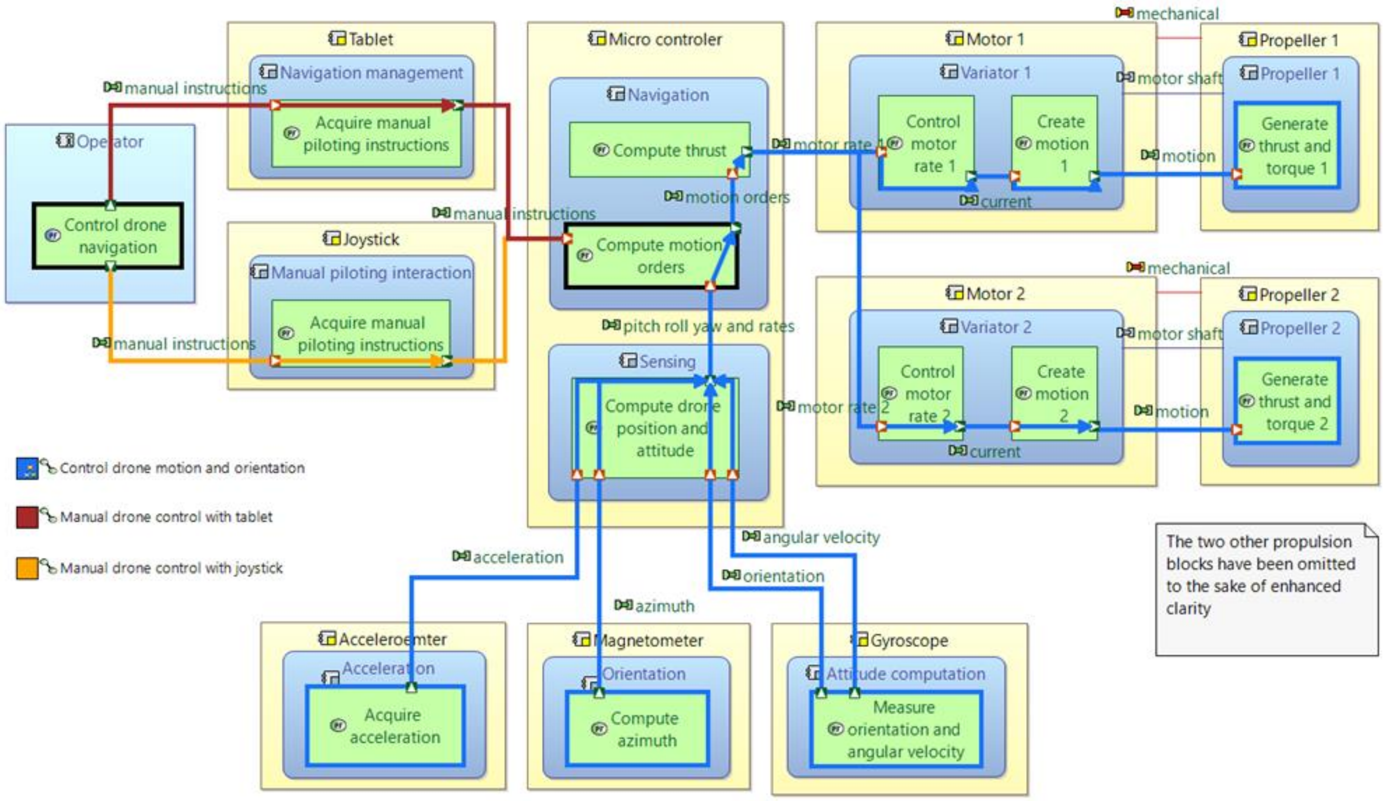
Functions have been further refined through textual requirements, not shown above. Functional Exchanges have been further described using other available concepts in Arcadia and Capella such as Exchange Items and Information models, not shown here neither.

In previous iterations, the Architecture team had developed a reference architecture for the product, which slightly evolves at each iteration. One of the variabilities that have been already considered concerns the device that will be used by the operator to control the motion of the drone. Some markets require fine control of the motion (security and inspection markets) whereas others do not (agriculture). The first ones will be provided a dedicated joystick, and the second ones will control the motion using a tactile tablet, which is also used for visualization purposes.

Because of these architectural decisions, when Architects work in the Physical Architecture perspective, they will need to define several variants of the functional chains describing the Manually pilot the drone capability. The figure below is one of the diagrams that will be presented to software development and tests teams as a specification of what is expected from them in the next iterations. It shows:

- The physical components of the architecture involved in this capability: Tablet, Joystick, Microcontroller, Accelerometer, Gyroscope, Magnetometer, and Motors and Propellers; shown in yellow
- The expected behavior of these physical components, which is represented by the functions (in green), grouped in behavioral components (in blue) that are deployed in the physical components
- The Operator external entity (in light blue) and its function related to the capability
- Three of the Functional Chains that realize the capability in the Physical Architecture perspective: Control drone motion and orientation, Manual drone control with tablet, and Manual drone control with joystick; highlighted in blue. The functions highlighted in black are involved in more than one functional chain. The end-to-end service is represented by the composition of functional chains: the joystick variant is represented by the Manual drone control with joystick + Control drone motion and orientation functional chains, the tablet variant is represented by Manual drone control with tablet + Control drone motion and orientation functional chains.

In fact, as the automated piloting of the drone was already addressed in previous iterations, the functional chain Control drone motion and orientation was already been developed and tested. The iteration hence focused on designing the upstream functional chains.



Were the architects working alone?

Of course not! Co-engineering is a well-established practice in Pythagoras. Focusing only on the teams considered in this series of articles:

- Integration, Verification and Validation (IVV) representatives worked with Capability Leaders and Architecture team to secure the testability of product-level functional chains. In fact, in parallel they translated Functional Chains in System Needs Analysis perspective in corresponding product-level test procedures. Each test "tells the story" of its corresponding functional chain.

- Representative members of the software teams participated to the regular reviews of the architecture increments. Their role was to anticipate the feasibility and to make sure the product-level vision of the solution is compatible with the current software architecture.

Next episode

In the third and last episode of this series of articles, we will see what happens at the end of this iteration and how the architecture increment is delivered to other engineering teams. We will also take a closer look at the activities performed by IVV and software team in further iterations, and to the “Evaluation” activity.

List of Acronyms Used in this Paper

Acronym	Explanation
MBSE	Model Based Systems Engineering
PLE	Product Line Engineering

References

Dove, Schnidel 2018. Agile Systems Engineering Life Cycle Model for Mixed Discipline Engineering. 28th Annual INCOSE International Symposium 2018.

INCOSE, 2014, ‘A World in Motion – Systems Engineering Vision 2015’.

ISO/IEC/IEEE 15288, 2015, ‘Systems and software engineering – System life cycle processes’.

IEEE 1220, 2005, IEEE Standard for Application and Management of the SE Process

Voirin J-L, 2017, ‘Model-based System and Architecture Engineering with the Arcadia Method’, ISTE Press, London & Elsevier, Oxford, 2017

Capella, 2019, Capella Polarsys Website: https://www.polarsys.org/capella/

Bonnet S, Voirin J-L, Normand V , Exertier D, 2015, ‘Implementing the MBSE Cultural Change: Organization, Coaching and Lessons Learned’, 25th INCOSE International Symposium

Navas, J, 2019, ‘How MBSE practices support the effective implementation of a Product Line Engineering approach’, 29th Annual INCOSE International Symposium, 2019.

Bonnet S, Voirin J-L, Navas J., 2019, ‘Augmenting requirements with models to improve the articulation between system engineering levels and optimize V&V practices’, 29th Annual INCOSE International Symposium, 2019.

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

About the Author

Juan Navas is a Systems Architect with 15 years’ experience on performing Systems Engineering activities and implementing innovative engineering practices in multiple organizations. He currently leads the Modelling & Simulation team at Thales Corporate Engineering and dedicates most of his time to expertise and consulting for Thales business units and other organizations worldwide, accompanying managers and architects when implementing MBSE practices. He holds a PhD on embedded software engineering, a MSc Degree on control and computer science, and Electronics and Electrical Engineering Degree.

Overview of a Systems Engineering ROI Thesis by E. Honour

by Wioletta Kowalczyk BSc BS MS (PPI)

Email: wkowalczyk@ppi-int.com

This short paper overviews a PhD thesis written by Eric C. Honour of research conducted by the author at the University of South Australia [1]. The thesis aims to provide a fresh perspective and overview of the topic of systems engineering (SE) return on investment (ROI).

The main goals of the research were to determine the degree of the correlation between systems engineering activities and program/project success, and to calculate the optimum amount and type of systems engineering activities based on a definable parameters of a program/project.

As a result of detailed ontology work, the author has provided his detailed definitions for eight main SE activities.

- Total SE (SE) is considered as total effort expended across the eight system level activities defined below.
- Mission/purpose Definition (MD) is a starting point for creating a new system or modifying an existing one. The artifacts created during this activity use system users' language, rather than technical language.
- Requirements Engineering (RE) is a core activity for the systems engineering discipline. According to the author, RE encompass requirements creation, management and definition of the system capabilities, characteristics and quality factors.
- System Architecture (SA) is the design aspect of the SE activities including the definition of a system in terms of its components and their relationships.
- System Integration (SI) is the next system-level activity that occurs after detailed activity of design, purchase, creation and testing of system components.
- Verification and Validation (VV) is defined by the author as the comparison of a system or developmental artifact with its requirements through the use of specific methods: inspection, analysis, demonstration, test or other.
- Technical Analysis (TA) is defined as the assessment of system performance against the requirements.
- Scope Management (SM) is area of SE activity that focuses on the contractual relationships with clients and subcontractors.
- Technical Leadership/Management (TM) is said to be an asset for every "systems engineer" and is defined to include the abilities of project planning, technical progress assessment, technical control, team leadership, inter-discipline coordination, risk and interface management.

The researcher over a 3 year period interviewed staff of several companies and programs/projects, in order to understand the key thesis questions by determining the following factors in each program/project: types and amounts of SE activities used in the program/project, level of success of the program/project, and program/project characterization parameters. The minimum number of programs/projects needed for adequate accuracy of correlation was calculated by the researcher to be 43.

The thesis provides an additional perspective on SE ROI, building upon previous studies that were mainly anecdotal in nature. These previous works indicate underlying trends that support the possibility of calculating SE ROI. The earlier studies referred to in the thesis are a Boundary management study (Ancona 1980s), NASA project definition (Gruhl 1992), Impact of SE on quality and schedule (Frantz 1995), Large engineering projects study (Miller 2000), a landmark SEI study (Elm et al 2008) [2] and many other studies including INCOSE research on the impact of SE on the development of complex systems. The thesis does not embrace a later, substantial SEI study conducted in a similar timeframe to the research conducted by the author [3].

The research itself reported on in the thesis was designed in a careful and well thought out manner, defining main research questions (RQA and RQB), definitions of terms that were used subsequently, activities to be performed within the scope of the research, ethics considerations, observations and findings analyses.

The research focused on collecting the basic types of data in order to answer questions:

1. RQ_A: Is there a quantifiable correlation between the amount, types and quality of systems engineering efforts used during a program/project and the success of the program/project?
2. RQ_B: For given program/project, can an optimum amount, type and quality of systems engineering effort be predicted from the quantified correlations?

First step in the research was to gather data on program/project success, measured in cost, schedule, and technical quality, these combined into an overall success measure. Second was to understand and gather the data on systems engineering effort, measured in SE effort costs in each of the eight SE activity categories against the program/project total cost. Third was to gather data for program/project characterization (size, complexity, quality) to parameterize the expected correlation of systems engineering effort with program/project success.

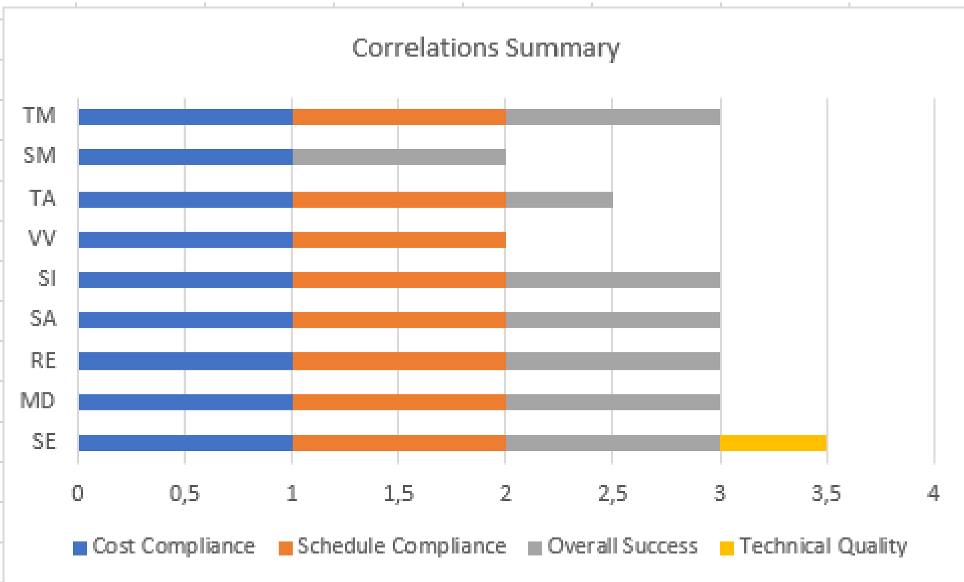


Figure 1 Correlations summary graph

The thoroughly developed model and data set were used in statistical modeling to answer the two questions. The main conclusions were:

- all of the defined SE activities exhibited a significant positive correlation with cost compliance,
- all of the defined SE activities except Scope Management exhibited a significant positive correlation with schedule compliance,
- all of the defined SE activities (except Mission/purpose Definition, Verification and Validation, and Technical Analysis) exhibited a significant correlation with overall success,
- none of the defined SE activities exhibited a significant correlation with technical quality.

Therefore, one can conclude that there is a quantifiable relationship between systems engineering effort and program/project success. This finding serves as a caution to programs/projects, meaning that the level of SE effort matters to the likely success of the program/project, as does the mix of that effort across the constituent activities of SE. As defined below, the SE activities have a significant and quantifiable ROI.

No correlation though, was found between the SE effort and system technical quality. It was determined that there is an optimum amount of SE activities for highest program/project success, and programs/projects typically use less SE effort than is optimum. A method of calculation is provided in the thesis for the optimum. Values are provided in this synopsis.

For estimation of SE effort, some program/project characterization parameters were found to be of much greater importance than others. This finding shows that when estimating program/project SE effort it is advisable to particularly focus on level of definition at start, development autonomy, level of integration, system size etc. as the most on the important program/project factors. One of the SE factors, Technical leadership/management (TM), was shown to be unique in providing optimum program/project success simultaneously in cost, schedule and stakeholder satisfaction.

The research concluded that there is a quantifiable relationship between levels of systems engineering effort and program/project success demonstrated by high correlation coefficients, well in excess of test values for significance level of 0.05.

The research showed that the ROI for SE effort can be as high as 7:1 for programs/projects expending little to no SE effort. For programs/projects expending median level of SE effort, the ROI was 3.5:1.

The major finding of the research was that the optimum amount of SE can be predicted based on all nine hypotheses (the hypotheses relating to total SE and the eight individual SE activities). The evidence was achieved by selecting a prediction methodology for optimum levels of SE activities, then demonstrating that the bounds on the selected methodology proved false the null hypotheses.

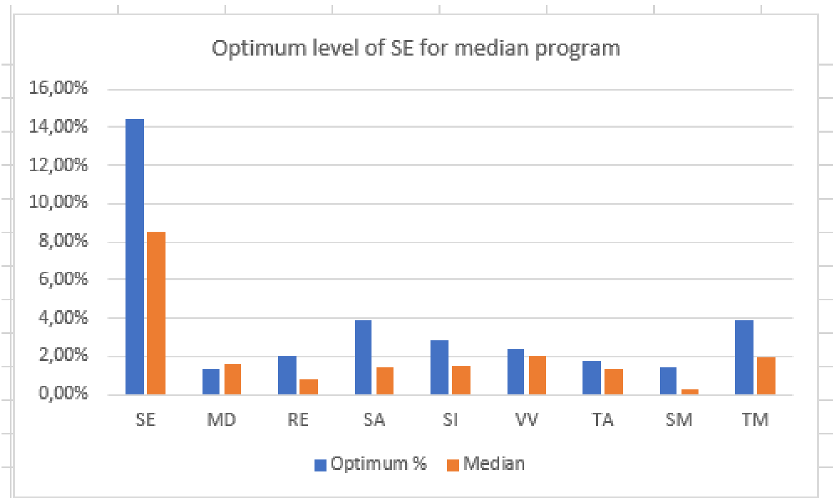


Figure 2 Optimum level of SE for median program

As stated previously, the research concluded that there is an optimum amount of systems engineering effort for maximum program/project success. For a program/project of median characterization parameters, the optimum is 14.4% of the total program/project cost. Examples of use of the estimating method were provided for space system development (optimum level of total SE 15.6%) and airborne training system development (optimum level of total SE 13.2%).

In summary, the research concluded that a method for estimation is available to determine the optimal levels of SE effort for a given set of program/project characterization parameters. Variation in the program characterization changed the optimum within the range 8% to 19% of total program/project cost.

[1] Systems engineering return on investment by Eric C. Honour BSSE MSEE, A thesis submitted for the degree of Doctor of Philosophy, Defence and Systems Institute, School of Electrical and Information Engineering, University of South Australia, January 2013

Available online at: http://www.hcode.com/seroi/documents/SE-ROI%20Thesis-distrib.pdf

[2] Elm, J, et al. (2008). A Survey of Systems Engineering Effectiveness. Special Report CMU/SEI-2008-SR-034. Pittsburgh, PA, USA: Carnegie Mellon Software Engineering Institute.

[3] Elm & Goldenson (2012). The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey. Special Report CMU/SEI-2012-SR-009. Pittsburgh, PA, USA: Carnegie Mellon Software Engineering Institute.

“

Proceeding in ignorance of what is already known or knowable is rarely a part of the formula for producing the best result .

ROBERT HALLIGAN

A SYSTEMS ENGINEERING TOOLS TAXONOMY

by Robert J. Halligan FIE Aust CPEng IntPE(Aus)

Email: rhalligan@ppi-int.com

Abstract

Utilization of software tools and cloud-based services to support a systems approach to the engineering of systems continues to grow apace. This reliance led to Project Performance International (PPI) and the International Council on Systems Engineering (INCOSE) establishing under a Memorandum of Understanding a project to develop a systems engineering tools database (SETDB). One aspect of the work on this project has been the development of a systems engineering (SE) tools taxonomy to assist in organizing data on SE tools. The development built upon work previously done by the author. This paper describes the taxonomy that has resulted and is used with the SETDB (see systemsengineeringtools.com).

Principles for defining tool categories

The following principles were defined for use in developing the taxonomy. These principles were derived from a study of the intended users and users of the database, the aim being to maximize ease of use of the SEDB:

Principle 1. The taxonomy should accommodate stand-alone software tools, families of software tools, and cloud-based services supporting systems engineering.

Principle 2. Each tool category should possess a significant orientation towards one or more practices commonly associated with systems engineering, viz. problem definition/problem solving/verification/validation, and management of these activities, as contrasted to a more general application.

This criterion places, for example, technical management software such as EVM as being within scope, whereas general project management software would be out of scope. Similarly, a requirements authoring tool would be within scope, whereas a word processing application would not be.

Principle 3. Consistent with the nature of systems engineering, the tool taxonomy should be predominantly agnostic to application domains and solution technologies. However, provision should be made for domain-specific tool categories – both application domain and technology domain, at a low level of granularity. This principle would place within scope, for example, regulatory compliance software for medical device development.

Principle 4. Tool categories should be oriented towards tool functionality (capability of the tool), not necessarily the application of the tool. The reasoning behind this principle was that many tool functionalities can be used for a multitude of purposes having different degrees of likelihood, making classification of tools on this later basis problematic.

Principle 5. The taxonomy may incorporate hierarchy reflecting levels of abstraction of classification.

Principle 6. The taxonomy should employ consistent logic of navigation in its structure.

Principle 7. Where hierarchy is used, a subcategory must be a member, not just a possible functionality, of a parent category.

Principle 8. Parent categories may have no subcategories.

Principle 9. Language used in naming categories should be in mainstream use by intended users, in the application domain of the tool.

Principle 10. Tool categories should be keyworded to accommodate variation in usage of language across different technology, application and geographic domains.

Principle 11. Keywords should be defined on the basis of likelihood of the user searching for tools against the term, not on the relevance of the term to the application of the tool.

The Tool Taxonomy

1 Decision Support 1.1 **Analytical Hierarchy Process (AHP)** 1.2 **Analytic Network Process (ANP)** 1.3 **Conjoint Analysis** 1.4 **Decision and Event Trees** 1.5 **Multiple Attribute Utility Theory (MAUT)** 1.6 **Quality Function Deployment (QFD** 1.7 **Decision Support – Other**	2 Design Thinking 3. Domain-Specific Engineering 3.1 **Domain Specific – Application** 3.2 **Domain Specific – Technology (excluding software)** 3.3 **Software**
4 Engineering Management 4.1 **Engineering Capability Appraisal** 4.2 **Configuration Management (CM)** 4.3 **Defect Tracking** 4.4 **Documentation Management** 4.5 **Earned Value Management (EVM)** 4.6 **Engineering Cost Estimation** 4.7 **Engineering Performance** 4.8 **Engineering Planning** 4.9 **Issue Management** 4.10 **Knowledge Management (KM)** 4.11 **Organizational Change Management** 4.12 **Process Mining** 4.13 **Regulatory Compliance** 4.14 **Technical Performance Measurement (TPM)** 4.15 **Technology Readiness Level (TRL)** 4.16 **Verification & Validation Management** 4.17 **Engineering Management – Other**	5 Lifecycle Management 5.1 **Application Lifecycle Management (ALM)** 5.2 **Product Lifecycle Management (PLM)** 5.3 **Lifecycle Management – Other** 6 Mathematical Analysis & Modeling 7 Meta-Modeling (Modeling Language Workbenches) 8 Modeling & Simulation,excluding CAD, Math & Value Modeling. (M&S) 9 Product Line Engineering and Management (PLE) 10 Prototyping
11 Physical Design 11.1 **Brainstorming** 11.2 **CAD** 11.3 **Design for Six Sigma (DFSS)** 11.4 **Physical Design – Other**	12 Requirements Engineering 13 Quality Management 14 Risk Management 15 SE Training Software 16 SE Tool Integration Software

17 Specialty Engineering 17.1 **Aesthetics** 17.2 **Durability** 17.3 **Environmental Engineering** 17.4 **Fashion & Style** 17.5 **Human Factors Engineering** 17.6 **Information Security Engineering** 17.7 **Lifecycle Cost Analysis** 17.8 **Maintainability Engineering** 17.9 **Producibility Engineering** 17.10 **Product Costing** 17.11 **Recyclability Engineering** 17.12 **Reliability Engineering** 17.13 **Safety Engineering** 17.14 **Thermal Engineering** 17.15 **Specialty Engineering – Other**	18 Systems Thinking 19 Tailoring 20 Testing (see Verification) 21 Verification and Validation (V&V) 21.1 **Formal Methods** 21.2 **Integrity Analyzers** 21.3 **Interface Compliance** 21.4 **Performance Analysis** 21.5 **Test Coverage Analysis** 21.6 **Verification & Validation – Other** 22 Visualization 22.1 **Augmented Reality** 22.2 **Graph Visualization** 22.3 **Data Visualization** 22.4 **Visualization – Other**

Tool Category Scoping Definitions and Keywords

Leaf tool categories are listed alphabetically and defined below, with keywords

Tool Category	Scope of Tool Category	Keywords (if any)
Aesthetics	Software tools specifically intended for use in engineering the property of aesthetic appeal into a product. Aesthetics is a discipline that defines a design's pleasing qualities. In visual terms, aesthetics includes factors such as balance, color, movement, ...	beauty
Analytic Network Process (ANP)	Software tools which implement Saaty's decision support methodology ANP. See https://en.wikipedia.org/wiki/Analytic_network_process	AHP
Analytical Hierarchy Process (AHP)	Software tools which implement Saaty's decision support methodology AHP. See. https://en.wikipedia.org/wiki/Analytic_hierarchy_process	ANP
Application Lifecycle Management (ALM)	Software tools which are intended to be used primarily for management of the lifecycle of a software application, from concept to delivery to maintenance.	"digital mockup", "issue management", "Corrective and Preventative Actions (CAPA)", "regulatory compliance and management", "product configuration", visualization, "tool costing", PLM, "report & specification generation", "requirements capture & authoring", "data exchange (RIF/REQIF)", "requirements allocation", "requirements tracelinking", "Product Lifecycle Management", "program planning", "project management"

Tool Category	Scope of Tool Category	Keywords (if any)
Augmented Reality (AR)	Software tools for the use real-time in engineering of information in the form of text, graphics, audio, and other virtual enhancements integrated with real-world objects.	AR, 3D, hologram
Brainstorming	Software tools intended primarily for use in group idea-generation, often problem-solving, that involves the spontaneous contribution of ideas from all members of the group.	“idea generation”, “mind mapping”, EBS, “nominal group”, “group passing”, “team idea mapping”
CAD	Software tools used primarily for the performance of computer-aided design, especially mechanical design. Includes virtual physical prototyping. Excludes modeling and simulation and decision support tools, and augmented reality tools.	Computer-aided design, CADD, "electronic design automation”, EDA, "mechanical design”, MDA, "computer-aided drafting”, "engineering drawings”, 3D, “computational geometry”, "computer graphics”, "discrete differential geometry”, "geometric models”, "computer-aided geometric design”, CAGD
Configuration Management (CM)	Software tools used primarily for supporting the technical and administrative activities concerned with identifying, recording, and controlling change to the attributes of a product of engineering activity during the creation, maintenance, and use of the product.	baseline, PCA, FCA, RSA, “change control”, BOM, “change management”
Conjoint Analysis	Software tools supporting the performance of conjoint analysis - a survey-based statistical technique used in determining how people value different attributes possessed by a product or service.	“best–worst scaling”, “decision support”, “decision analysis”
Data Visualization	Software tools intended for use primarily in the graphic representation of engineering-related data.	“data visualisation”
Decision and Event Trees	Software tools supporting the use of decision and event trees. (see Ang and Tang, 1984)	“decision tree”, “event tree”, “decision analysis”
Decision Support	Software tools intended to be used primarily to understand the merits of different decision alternatives for the purpose of deciding between these alternatives, based on their attributes. Excludes tools for the estimation of the value of any single attribute, i.e. decision support tools involve multiple criteria.	MCDA, “decision analysis”, DSS, “multi-objective optimization”, “multi-objective programming”, “vector optimization”, “multicriteria optimization”, “multiattribute optimization”, “Pareto optimization”, “decision support system”, MCDM

Tool Category	Scope of Tool Category	Keywords (if any)
Decision Support - Other	Software tools intended to be used primarily to understand the merits of different decision alternatives for the purpose of deciding between these alternatives, based on their attributes, but excluding a. Analytical Hierarchy Process (AHP) b. Analytic Network Process (ANP) c. Conjoint Analysis d. Decision and Event Trees e. Multiple Attribute Utility Theory (MAUT) f. Quality Function Deployment (QFD)	MCDA, "decision analysis", DSS, "decision support system", "multi-objective optimization", "multi-objective programming", "vector optimization", "multicriteria optimization", "multi-attribute optimization", "Pareto optimization", MCDM
Defect Tracking	Software intended to be used primarily for the tracking of the existence, status and resolution of defects in engineering work products.	FRACA, FRACAS, bug, "failure reporting", "fault reporting", CAR, "corrective action", "quality management"
Design for Six Sigma (DFSS)	Software tools supporting Design for Six Sigma including quality function deployment (QFD), axiomatic design, TRIZ, Design for X, design of experiments (DOE), Taguchi methods, tolerance design, robustification and Response Surface Methodology. DFSS is an engineering design process, business process management method related to traditional Six Sigma.	DMAIC, DMADV, "quality function deployment", QFD, "axiomatic design", TRIZ, "Design for X", "design of experiments", DOE, Taguchi, "tolerance design", robustification, "Response Surface Methodology"
Design Thinking	Design thinking is a human-centered approach to innovation that draws from a toolkit to integrate the emotions of people, needs of people, the possibilities of technology, and the requirements for business success. As such, it transcends tool categories of requirements engineering, physical design, decision support, and validation. Software tools and cloud services in this category are limited to those explicitly and substantially identified with design thinking in online resources and in the literature.	"empathy map", "feedback grid", ideation, "Impact/Effort Matrix", "Impact/Power Matrix", innovation, "journey map", "rapid prototyping", "stakeholder map", UX
Documentation Management	Software tools intended primarily for use in the capture, storage and retrieval of engineering documents.	"document management", "documentation control", "document control", "configuration management", "structured content management"
Domain Specific – Application	Software tools intended to be used in performing engineering activities in specific application domains, for example aviation, medical, rail, public infrastructure,	aviation, medical, rail, infrastructure, healthcare, transportation, biomedical, aviation, avionics, communication, LSA, "logistic support analysis", PSA, "product support analysis"

Tool Category	Scope of Tool Category	Keywords (if any)
Domain Specific - Software	Software tools intended to be used in software engineering.	programming, "software development"
Domain Specific – Technology (excluding software)	Software tools intended to be used in performing engineering activities in specific technologies, for example electronics, radio frequency engineering, chemicals, materials, but excluding software.	electronics, radio frequency, chemical, materials, mechanical, civil, hydraulic, mechatronics",
Domain-Specific Engineering	Software tools intended to be used in performing engineering in specific application or technology domains.	
Durability	Software tools specifically intended for use in engineering the property of durability into a product. Durability is the quality of being able to resist wear, decay, damage and similar over time.	corrosion, "fatigue life", deterioration, "forensic engineering", PBDE
Earned Value Management (EVM)	Software tools implementing earned value management, based on the comparison of work performed and work planned.	EVMS, "earned value analysis", EVA, CSCS, EVPM, DFARS, "C/SCSC"
Engineering Capability Appraisal	Software intended mainly for use in assessment of the capability of an organization or person to perform engineering activities, by evaluation of current practice against an objective framework.	CMMI, "capability maturity", CMM, "EIA-731"
Engineering Cost Estimation	Software tools supporting the estimation of the cost of conducting the engineering and its management, including WBS-based and parametric engineering cost estimating. Excludes tools intended primarily for estimation of product cost in a quantity production sense.	"engineering economic analysis", "engineering cost estimating"
Engineering Management	Software tools intended to support engineering management specifically. Excludes tools intended to support management in general, and project management in general.	"systems engineering management", SEP, "project management", SEMP
Engineering Management - Other	Software tools intended to support engineering management but specifically excluding: a. Engineering Capability Appraisal b. Organizational Change Management c. Configuration Management d. Defect Tracking e. Documentation Management f. Earned Value Management	"systems engineering management", "project management"

Tool Category	Scope of Tool Category	Keywords (if any)
	g. Engineering Cost Estimation h. Engineering Performance Engineering Planning i. Issue Management j. Knowledge Management k. Portfolio Management l. Process Mining m. Regulatory Compliance n. Technical Performance Measurement o. Technology Readiness Level p. Verification & Validation Management,	
Engineering Performance	Software tools intended to be used primarily in recording, tracking or assessing engineering performance (not product performance).	LEPM, Lean, TMP, TPM, "Technical Performance Measurement", EVM, "earned value"
Engineering Planning	Software tools intended to be used primarily in planning engineering activities. Excludes project planning in general.	scheduling, costing, WBS, "Work Breakdown Structure", PBS, "Project Breakdown Structure", SEMP, SEP, 'Vee Model", "Wedge Model"
Environmental Engineering	Software tools specifically intended for use in engineering the property of environmental resilience or minimization of environmental impact into a product.	"Environmental Impact Assessment", sustainability, pollution, waste, "hazardous materials", "ecological engineering"
Fashion & Style	Software tools specifically intended for use in engineering the property of fashion or style appeal into a product.	aesthetics, trend, "wearable technology", beauty
Formal Methods	Software tools intended for use in engineering using mathematically rigorous languages and techniques.	trust, "trusted systems", verification
Graph Visualization	Software tools intended mainly for the engineering application of graph visualization. Graph visualization is the visual representation of the nodes and edges of a graph. Dedicated algorithms, called layouts, calculate the node positions and display the data on two (sometimes three) dimensional spaces.	"graph visualization", "tree diagram", "hierarchical data", treemap
Human Factors Engineering	Software tools specifically intended for use in engineering the property of usability into devices and systems for human use, having regard to the physiological and psychological characteristics of humans.	ergonomics, "human systems integration" "cognitive systems engineering", HFE, "man-machine interface", UX, "user experience", comfort

Tool Category	Scope of Tool Category	Keywords (if any)
Information Security Engineering	Software tools specifically intended for use in engineering the property of information security into devices and systems. Incorporates communications, computing and cyber-security. Excludes formal methods (see separate category).	“IT security”, trust, COMSEC, ITSEC, SIGINT, “network security”, “ICT security”, SSECMM, “SSE-CMM”, “Covert Channel Analysis”
Integrity Analyzers	Software tools intended primarily for analysis of the integrity of design or engineering data in general. Integrity analyzers find defects in design or data in general.	“Control flow Analyzers”, “Data Use Analyzers”, “Information Flow Analyzers”, “Semantic Analyzers”, “Compliance Analyzers”
Interface Compliance	Software tools that evaluate compliance of interfaces with requirements derived from applicable interface standards.	SCMI
Issue Management	Software tools that help monitor, manage and resolve issues in development.	“bug tracking”, “help desk”
Knowledge Management (KM)	Software tools that help engineering organizations classify, organize, control and share knowledge.	KM, “lessons learned”, patent, “intellectual property”, IP, SECI, security
Lifecycle Cost Analysis (LCC)	Software tools for lifecycle cost analysis – the cost of ownership of a system or product over its life cycle. Excludes domain-specific LCC software tools. Includes LCC tools using Monte Carlo simulation to address uncertainty.	LCC, LCCA, “life cycle cost”
Lifecycle Management	Software that specifically supports management of the life of a system of any nature from initial concepts through to end of life.	ALM, PLM
Lifecycle Management - Other	Software that specifically supports management of the life of a system of any nature from initial concepts through to end of life, excluding product and software application lifecycle management.	ALM, PLM, LCC
Maintainability Engineering	Software tools specifically intended for use in engineering the property of maintainability into devices and systems. Maintainability is the relative ease and economy of time and resources with which an item can be retained in, or restored to, a specified condition when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources.	RAM, RAMS, Availability, Repairability, “R&M”, “maintainability prediction”, LSA, “logistic support analysis”
Mathematical Analysis & Modeling	Software tools that specifically support the application of mathematics to engineering. See also Modeling and Simulation.	“numerical analysis”, Monte Carlo”, “Latin Hypercube”, “Finite Element Analysis”, “statistical analysis”, correlation, matrix inversion

Tool Category	Scope of Tool Category	Keywords (if any)
Meta-Modeling (Modeling Language Workbenches)	Software tools for application to the design of domain-neutral and domain-specific modeling languages.	DSL, “domain-specific language”
Modeling & Simulation (excluding CAD, Math & Value Modeling)	Software tools for modeling and simulation (M&S) that support representation of phenomena, either statically (modeling) or over time (modeling and simulation), to develop data as a basis for making technical decisions. M&S attempts to represent the logic of operation of real-world processes, equipment, people, activities and environments. For the SETDB, excludes CAD, Math & Value Modeling, and system dynamics modeling tools.	“M&S”, “behavior modeling”, “behavior modelling”, “discrete-vent”, “digital twin”, “functional design”, “functional analysis”, “logical design”, Workflow Management, Simulation, Management, Thread Analysis, Scenario Validation, Model Validation, Simulation Validation, SysML, OPM, “object process methodology”, BPMN, SDL, AADL, “system design”
Multi-Attribute Utility Theory (MAUT)	Software tools which implement Multi-Attribute Utility Theory (MAUT). MAUT is a structured methodology designed to handle the trade-offs among multiple objectives. (Keeney and Raiffa, 1976)	MCDS. MCDA, MCDM, “multiple attribute utility theory”, “multiple attribute utility technique”
Organizational Change Management (OCM)	Software tools intended for application to organizational change management. Organizational change management (OCM) is a framework for managing the effect of new enterprise processes, changes in organizational structure or cultural changes within an enterprise, i.e. the social side of change.	OCM, “organisational change management”
Performance Analysis	Software tools intended for engineering application in predicting the performance (how well a function is to be performed) in the engineering of a system, or in understanding why a system is not performing as expected.	
Physical Design	Software tools intended for use in designing the structure of a system, in terms of the system elements, their individual characteristics, their interfaces and their other relationships.	“system hierarchy”, “functional allocation”, “system breakdown structure”, “system block diagram”, SBD, “internal block diagram”, “interface engineering”, IBD, “block definition diagram”, BDD, “Parametric Diagram”, “package diagram”, SysML, OPM, “object process methodology”, “DFSS”, ‘CAD”, ‘design for six sigma”, “brainstorming”, “system design”
Physical Design – Other	Software tools intended for use in designing the structure of a system, in terms of the system elements, their individual characteristics, their interfaces and their other relationships. For the purpose of the	“system hierarchy”, “functional allocation”, “system breakdown structure”, “system block diagram”, SBD, “internal block diagram”, “interface engineering”, IBD, “block

Tool Category	Scope of Tool Category	Keywords (if any)
	SETDB, excluding Brainstorming, CAD and Design for Six Sigma (DFSS). Also excludes decision support and formal methods, which are separately listed.	definition diagram", BDD, "Parametric Diagram", "package diagram", SysML, OPM, "object process methodology"
Process Mining	Software tools intended to be used primarily in process mining. Process mining is a family of techniques in the field of process management that support the analysis of processes based on event logs. During process mining, specialized data mining algorithms are applied to event log data in order to identify trends, patterns and details contained in event logs recorded by an information system. Process mining aims to improve process efficiency and understanding of processes.	hyperautomation, "business process engineering", BPMN, "Petri net", "data mining", "business process management"
Producibility Engineering	Software tools specifically intended for use in engineering the property of producibility into devices and systems. Producibility, also called manufacturability, is Ease of manufacturing an item (or a group of items) in large enough quantities.	DFA, DFM, DFMA
Product Costing	Software tools specifically intended for use in engineering the property of cost into devices and systems Product cost management software provide a means to track and control design, process costs and materials costs in manufactured goods or other development-intensive products. The software is used to locate cost drivers and compare various designs for cost-feasibility, and cost efficiency.	profitability, BOM, yields, DFMA, Lean, DfC/DfP
Product Lifecycle Management (PLM)	Software tools which are intended to be used primarily for management of the lifecycle of a product, from concept to delivery to use and maintenance, to disposal.	Digital Mockup, BOM Management, Issue Management, Corrective and Preventative Actions (CAPA), Artwork Design and Proofing, Document Design and Publishing, Regulatory Compliance and Management, Product Configuration, Visualization, Tool Costing, ALM, Report & Specification Generation, Requirements Capture & Authoring, Data Exchange (RIF/REQIF), Requirements Allocation, Requirements Tracelinking, Software Lifecycle Management, Program Planning, Project Management, Search, Navigation, Reporting & Analytics

Tool Category	Scope of Tool Category	Keywords (if any)
Product Line Engineering & Management	Software tools which are intended to be used primarily for product line engineering and management. Product Line Engineering (PLE) is a way to engineer a portfolio of related products in an efficient manner, taking advantage of the products' similarities while respecting and managing their differences. PLE includes the activities involved in planning, producing, delivering, deploying, sustaining, and retiring products in a product line. PLE works by considering a portfolio as a single entity to be managed, as opposed to multiple separate products to be managed individually.	MODA, MODE, PLE, ISO/IEC 26550
Prototyping	In the context of the SETDB, software tools intended to be used in development for creation of early versions of products, often non-functional, that can be reviewed before going to development of the actual product. For this SETDB taxonomy, excludes virtual physical prototyping, which is included under CAD, and excludes augmented reality software tools, which are listed separately.	"HMI Prototyping"
Quality Function Deployment (QFD)	Software tools supporting the technique of Quality Function Deployment. Quality Function Deployment aims to be a process and set of tools used with the aim of defining customer needs and requirements and converting them into detailed engineering specifications and plans to produce the products that satisfy those needs and requirements.	"House of Quality", VOC, "Voice of Customer"
Quality Management	Software tools that assist the process of minimizing product defects. Excludes verification and validation and defect tracking, which are separate categories.	TQM, Quality Assurance, Issue Management, Corrective and Preventative Actions (CAPA)
Recyclability Engineering	Software tools specifically intended for use in engineering the property of recyclability into devices and systems.	DFA, reuse, "sustainable design"
Regulatory Compliance	Software tools specifically intended for use in engineering and managing the development of a product to ensure regulatory compliance.	FDA, FCC, airworthiness, "compliance assurance", "compliance engineering", "forensic engineering", ITAR
Reliability Engineering	Software tools specifically intended for use in engineering the property of reliability into devices and systems. Reliability is the inverse of the propensity of the device or system to fail.	MTBF, RAM, RAMS, MTTF, FTA, ETA, FMEA, DFMEA, FMECA, "forensic engineering"

Tool Category	Scope of Tool Category	Keywords (if any)
Requirements Engineering	Software tools specifically intended for use in authoring, eliciting, capturing, analysing, classifying, valuing, validating, specifying, communicating, measuring the quality of, verifying and/or managing requirements. See also modeling and simulation.	Goals, specification, "requirements traceability
Risk Management	In the context of the SETDB, software tools specifically intended for use in managing risk, viz. the expected loss with respect to one or more desired outcomes. Risk management has the ingredients of threats, vulnerabilities, probabilities, value of achievement of desired outcomes, and the value of avoidance of undesirable outcomes.	"risk analysis", "risk log", ERM
Safety Engineering	Software tools specifically intended for use in engineering the property of safety into devices and systems. System Safety Engineering is an engineering discipline that employs specialized knowledge and skills in applying scientific and engineering principles, criteria, and techniques to identify hazards and then to eliminate the hazards or reduce the associated risk when the hazards cannot be eliminated.	Hazards, HAZOPS, "hazards analysis", SSE, QRA, PRA, FTA, ETA, MILSTD-882, DEF STAN 00-56, AS61508, ARP4761
SE Tool Integration Software	Software tools specifically intended to be used in the integration of systems engineering tools.	"tools interoperability", "model exchange", "data exchange"
SE Training Software	Software tools specifically intended to be used in the development or delivery of systems engineering training.	Learning, "training needs analysis"
Software	Software tools specifically intended to be used in performing engineering activities in software development and maintenance.	programming
Specialty Engineering	Software tools specifically intended for use in engineering into devices and systems qualities beyond function and performance, commonly referred to as the "ilities", but with names not necessarily ending in "ility".	DFX, "Design for X"
Specialty Engineering - Other	Software tools specifically intended for use in engineering into devices and systems qualities beyond function and performance, commonly referred to as the "ilities", but with names not necessarily ending in "ility", and excluding the following, which have their own entries: a. Aesthetics b. Durability c. Environmental Engineering	comfort, "perception of safety", "perception of value for money", "mass properties", "electromagnetic compatibility", EMC, DFT, "Design for Test", DFX, Design for X

Tool Category	Scope of Tool Category	Keywords (if any)
	d. Fashion & Style e. Human Factors Engineering f. Information Security Engineering g. Lifecycle Cost Analysis h. Maintainability Engineering i. Producibility Engineering j. Product Costing k. Recyclability Engineering l. Reliability Engineering m. Safety Engineering n. Thermal Engineering. Examples are engineering for the perception of safety, engineering for perception of value for money, and trust engineering.	
Systems Thinking	Software tools intended primarily for stimulating systems thinking, including system dynamics tools. Systems thinking is a general term for looking at things systemically and thinking in terms of feedback. One of the major tools of systems thinking is the causal loop diagram, and another is the system archetype.	“system dynamics”, “causal loops”, influence diagrams”, “stocks & flows”
Tailoring	Software tools intended primarily for tailoring broader standards and defined practices to a specific application, by deleting unnecessary activities or adding extra activities.	
Technical Performance Measurement (TPM)	Software tools specifically intended to be used in implementing Technical Performance Measurement (TPM). TPM is a technique of predicting the future value over time of a key technical parameter for a product under development, and tracking and reporting current achievement against plan.	IPM, “integrated performance measurement”
Technology Readiness Level (TRL)	Software tools specifically intended to be used in using technology readiness levels in engineering management. TRLs are a method for estimating the maturity of technologies.	TRL, “technology maturity”, “technology risk”, TRA, “Technology Readiness Assessment”
Test Coverage Analysis	Software tools for calculating and managing test coverage. Test coverage analysis is a technique that determines the proportion of system behavior that has been or will be tested by a defined test program or set of test cases.	“test management”, “verification management”
***Testing* (*see Verification*)**		

Tool Category	Scope of Tool Category	Keywords (if any)
Thermal Engineering	Thermal Engineering is a specialized sub-discipline of mechanical engineering and chemical engineering that deals with the movement of heat energy and transfer. The energy can be transformed between two media or transferred into other forms of energy	thermodynamics, "fluid mechanics", fluid dynamics", "heat transfer"
Verification & Validation - Other	Software tools specifically intended for performance of verification or validation, but excluding tools for: a. Formal Methods b. Integrity Analysis c. Interface Compliance d. Performance Analysis e. Test Coverage Analysis.	"Risk Reduction", Testing, "quality management", "IV&V", "V&V"
Verification & Validation Management	Software tools specifically intended for implementation of verification or validation recording, traceability and reporting.	"verification traceability", RTEM, VCRI, VCRM, RVTM, VRTM, QRTM, RVC, VCM, VTM, TEMP
Verification and Validation (V&V)	In the context of the SETDB, software tools intended to be used for the primary purpose of establishing the correctness of an engineering work product (requirements, design, system element, system, or something else) – verification. Or for validation, software tools intended to be used for the primary purpose of establishing that an engineering work product (requirements, design, system element, system, or something else) satisfies the need for that work product.	"Risk Reduction", Testing, "quality management", "IV&V", "V&V"
Visualization	Software tools intended to be used primarily for communicating engineering information visually.	"engineering graphics", "computer graphics", 2D, 3D, "information visualization", "knowledge visualization", "product visualization", "visual communication", "visual analytics"
Visualization - Other	Software supporting the use of visualization in engineering, excluding: a. Augmented Reality b. Graph Visualization c. Data Visualization.	"engineering graphics", "computer graphics", 2D, 3D, "information visualization", "knowledge visualization", "product visualization", "visual communication", "visual analytics"

Applying MBSE Techniques to SOP Analyses

By Steven H. Dam, SPEC Innovations

Email: steven.dam@specinnovations.com

ABSTRACT

In this paper, we describe the process for applying model-based systems engineering (MBSE) techniques to create and analyze standard operating procedures (SOP). This work is based on research by George Mason University (GMU) for creating high quality standard operating procedures (SOPs) and using that information to modify a commercial off-the-shelf MBSE tool, Innoslate®, to create a new tool, Sopatra®, that aids SOP developers in creating and verifying that the procedures work within the Allowable Operational Time Window (AOTW). The tool compares the AOTW with the Time on Procedure (ToP) to create new metrics: Procedure Buffer Time (PBT) and Probability of Failure to Complete (PFtC). A new Natural Language Processing (NLP) technique was developed as well to enable better conversion from the text-based SOPs to behavioral models.

Keywords

Standard Operating Procedures, SOP, AOTW, ToP, PBT, PFtC, NLP, LML

INTRODUCTION

In many domains, particularly those that involve human safety, Standard Operational Procedures (SOPs) are the "glue" that holds the human-machine system together [1]. Merriam-Webster's online definition for SOP states: "established or prescribed methods to be followed routinely for the performance of designated operations or in designated situations [17]." SOPs enable flexible crew pairing and crew substitution, reliable performance of energy workers, quality production of goods, and safe operation of mass transit among many applications. SOPs are the basis for worker training and post-incident liability analysis. For these reasons, developing high quality SOPs is critical for the success of any operation.

Historically, these SOPs are developed by subject matter experts, such as pilots, in a text-based approach. Many assumptions are made by the experts that make learning and applying these SOPs difficult, particularly in an emergency situation. They also frequently contain logical errors that may not be detected until the SOP is used in a situation where little time is available for trial and error.

Measurement of time provides the means for evaluating procedures for management and control of complex, dynamic systems. Time provides a common unit of measurement of worker performance of tasks and provides a common basis of describing both natural and experimental conditions [14]. Time can also be used to correlate errors [5][11][13], predict operational efficiency [7], and adjust task loading [6] to meet the needs of the operation.

Several approaches to evaluating the performance of a procedure have been developed using the time dimension. The approaches include:

- analytical methods, such as timeline/workload analysis [11], and the identification of the sequence of events and the time frame for each event
- simulations, such as state-transition models [4] and keystroke level models [2], and
- workload models. [8][16]

Each of these methods calculate the time on task based on the sum of the time for each action in the sequence of actions that constitute the procedure.

The sequence of actions, as discussed in recent work by Kourdali and Sherry [9][10] demonstrates the use of Model-Based Systems Engineering (MBSE) to design and validate SOPs. This work also defines several metrics for using time. These metrics include: Allowable Operational Time Window (AOTW), Time on Procedure (ToP), Procedure Buffer Time (PBT), and the Probability of Failure to Complete (PFtC). Figure 1 below shows a graph of how these metrics relate to one another.

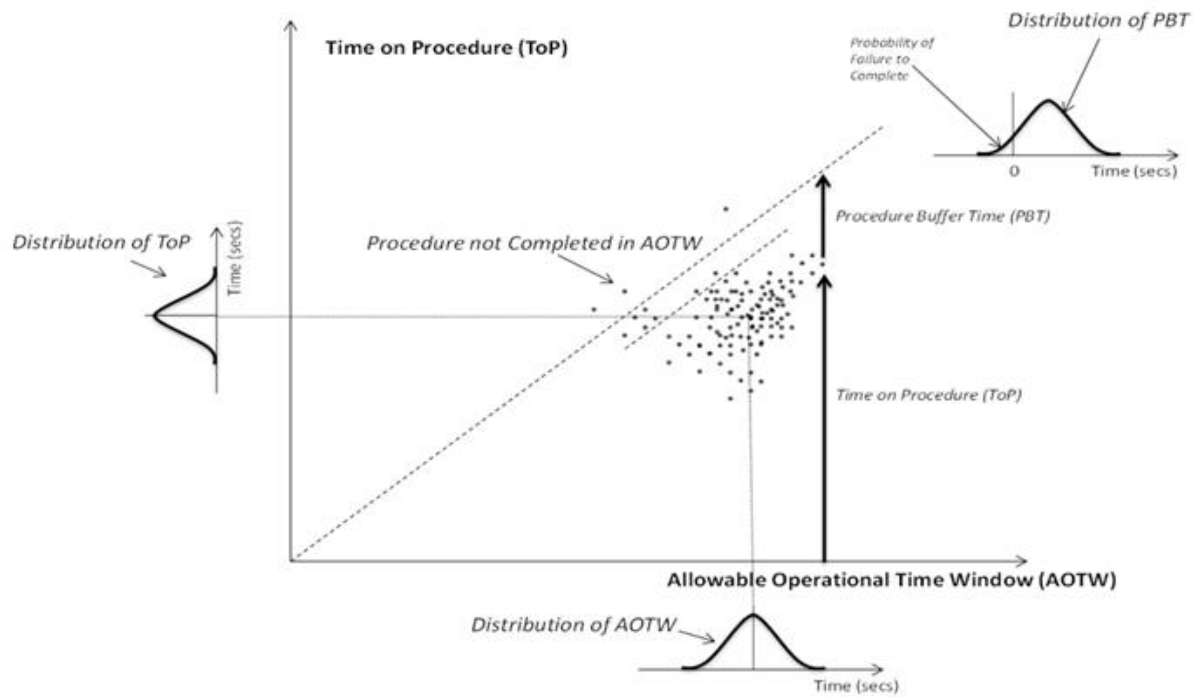


Figure 1. PBT provides a key metric for determining SOP success

The PBT is simply the difference between AOTW and the ToP. The PFtC unveils itself if the left side of the PBT distribution curve falls below zero. The PFtC becomes the measure of reliability for the SOP.

MBSE APPROACH TO SOP DEVELOPMENT

One MBSE technique, the Lifecycle Modeling Language (LML) [12], provides a simple way to understand the process/behavior modeling diagram, the Action Diagram (see Figure 2 below for an example), that can be used to model and develop SOPs.

Figure 2 was derived from the SOP for flight take-off and only represents the first 16 steps of a 77-step process. Those 16 steps are:

1. ATC gives clearance to Takeoff: “XXX123, RNW16, cleared for takeoff”.
2. PF reads back clearance “Cleared for takeoff, XXX123”.
3. PM announces “TAKE-OFF”.

4. PM announces “YOUR CONTROLS” simultaneously holds ailerons into wind.
5. PF puts right hand on the nose wheel steering control and simultaneously keeps left hand on lap, and simultaneously confirms “MY CONTROLS”.
6. PM advances throttle levers.
7. PM checks that all 4 engines accelerate symmetrically beyond 50% N1.
8. PM activates auto throttles by means of TOGA buttons.
9. PM checks FMA auto-throttle engagement: A/T green arc and FADEC trim arrow extinguished (if applicable).
10. PF checks FMA auto-throttle engagement: A/T green arc and FADEC trim arrow extinguished (if applicable).
11. PM checks that take-off thrust is set before reaching 80 kts.
12. Note: Needs time/aircraft dynamics awareness.
13. PM reports “TAKE-OFF THRUST SET”.
14. PF verifies that takeoff thrust is set.
15. PF confirms “CHECKED”.
16. PM checks engine parameters throughout the take-off toll to be within limits.

The 16 steps above translate into the 44-steps in the Action Diagram below through the use of the Sopatra® tool. Operator refers to the Pilot Flying (PF) and Operator 2 refers to the Pilot Monitoring (PM).

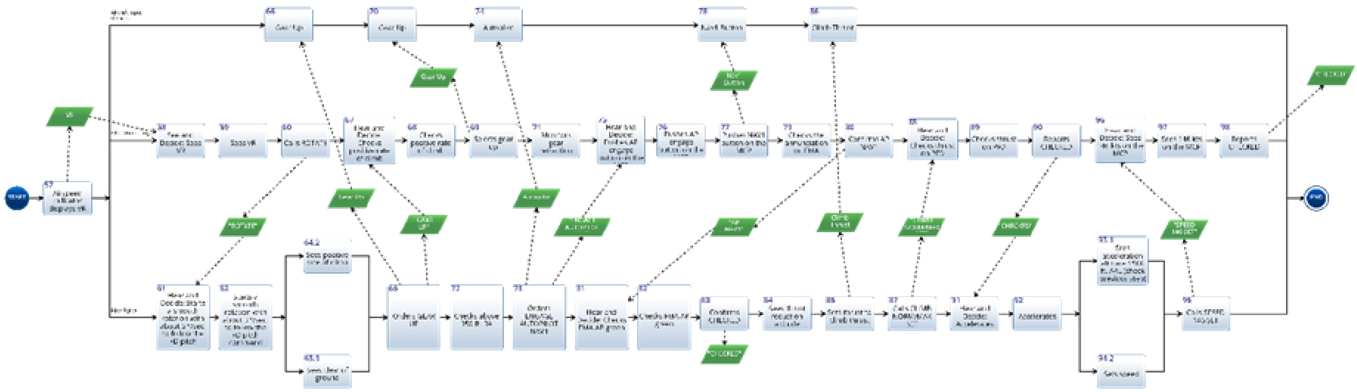


Figure 2. Detailed Level of Airplane Takeoff Process Model Shows the Complexity Required.

The Sopatra® tool uses the Innoslate® Action Diagram to create the process shown in Figure 2 from the text above. LML’s Action Diagram provides a significant simplification from other diagramming languages, such as SysML or BPMN, in that it does not use abstract constructs for logical operations. Instead the Action Diagram uses a special type of Action that represents the decision point required, as indicated by the superimposed diamond on the basic Action block (rounded rectangle). By making this an Action, we can then allocate that decision point to who or what performs it. That simplification ensures better modeling for security, command and control, and management functions by embedding them into the design where they are needed. Although many processes, such as the one above, do not contain these decision points, the user can add them as needed.

The GMU research indicated several best practices that needed to be implemented in any tool that aids in the creation of SOPs as models. The first best practice that can be seen in Figure 2 is the use of “Hear and Decide” actions, which the tool automatically adds when it detects a need to show where a human needs to respond to an input, either verbal or from the instruments. The “Hear and Decide” actions require a certain amount of time for response that may not be realized when developing this kind of SOP.

Another best practice is the addition of long-term memory (LTM) actions. Sopatra® includes the capability to add these as part of a template for the actors (Assets in LML), including the operators. If no LTM actions are identified, then the user can delete the LTM line, as was done in the diagram above.

The Input/Output (I/O) entities in these diagrams are symbolized by a parallelogram. I/Os can act as a trigger that allows an enabled Action to execute. In this way, I/Os can control the sequencing of Actions executed by individual actors or Assets, which is the class in LML for physical systems or people.

By using discrete event and Monte Carlo simulation, the user can also validate the execution of the process to ensure that it meets the operational goals of the system. The commercial-off-the-shelf tool, Innoslate® implements LML and has a built-in discrete event and Monte Carlo simulation capability [15]. In the following sections, we discuss our approach to developing the SOP tool, Sopatra®, to implement this important research.

SPEC Innovations' MBSE METHODOLOGY

SPEC Innovations has been working in the Model-Based Systems Engineering (MBSE) discipline since its inception in 1993. Before this approach was called MBSE, the engineering community called it "Computer-Aided Systems Engineering" or CASE. The underline for CASE was to differentiate it from the "Computer-Aided Software Engineering" approach or "CASE." Tools such as RDD-100 and Slate used this term to describe their capabilities. SPEC Innovations created its Innoslate® systems engineering and program management tool to meet the emerging needs of the community for MBSE and digital engineering in cloud computing environments. MBSE was formally defined by the International Council on Systems Engineering in 2007 as "*the **formalized application of modeling** to support system requirements, design, analysis, verification and validation, beginning in the conceptual design phase and continuing throughout development and later life cycle phases*." This definition does not require a database management tool, but MBSE has come to be seen as a way to manage all the data and information needed to describe, build, and validate the system. That capability requires some form of a database management tool to capture the information. It also requires a means to view the information in a variety of diagram types or models.

SPEC Innovations uses LML in conjunction with the classic systems engineering process shown in Figure 3. Details of this process are described in a recent book [3].

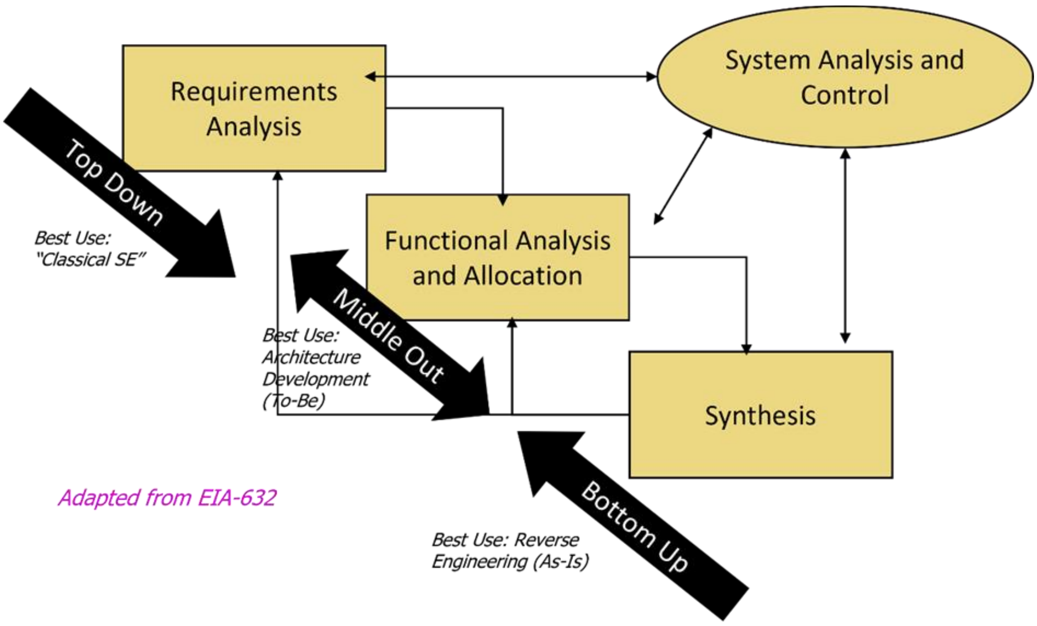


Figure 3. SPEC's Design and Analysis Process Provides a Proven Systems Engineering Approach.

This process was used to derive the functional requirements needed for software development. Those functional requirements were then passed on to the software developers to create prototypes of the software. Since Innoslate® already contained the basic functionality needed to create the Action Diagrams and simulate them, it made sense to build the prototype using Innoslate®'s application programming interfaces (APIs).

AGILE SOFTWARE ENGINEERING APPROACH

The prototype development and productization followed our software development process shown in Figure 4 below.

The process in Figure 4 is the same process SPEC Innovations uses in the development of Innoslate®. The functional requirements and any user feedback from previous versions are used to evolve the code, moving it closer and closer to the end goals set by the user needs.

For SOP development, we realized early that most SOPs evolved from a text-based approach. Each step in the process often represented a set of Action and I/Os. As such we need a way to import the text-based SOP and other data, such as standard performance distributions, acronyms, and common Assets, such as the environment that is being worked in, functions of the plant or system that is being analyzed, input mechanisms, output displays, and the operators themselves. As such, a specialized import analyzer was developed to ingest these data elements into Innoslate®. These additional files, which can be tailored to particular domains, enable better parsing of the text-based information.

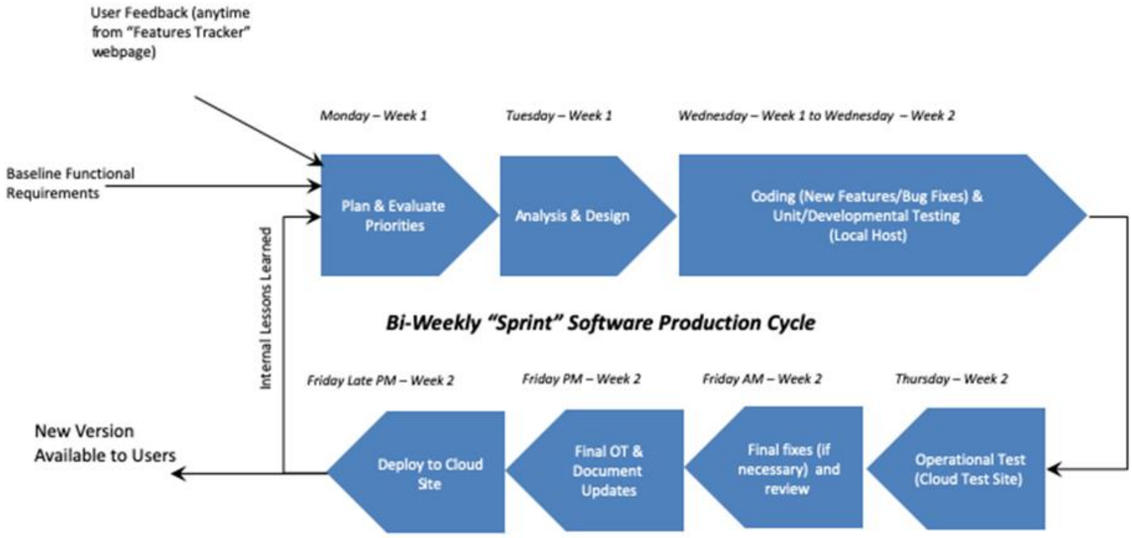


Figure 4. SPEC Innovations' Agile Development Process Creates Software Products Rapidly

NLP libraries were initially used to translate the text-based SOP steps into the appropriate set of Actions and I/Os. We determined that those libraries were insufficient for this purpose. As a result, specific new NLP algorithms were developed as part of the process to enable more accurate translation of the text-based steps into the Action Diagram. Modification of the Monte Carlo simulator output also was necessary to show the graphs for ToP vs. AOTW and PBT vs. Time, as well as the PFtC calculation.

Once the prototype was developed and initial testing was performed, productization began. To create the new tool, Sopatra®, it was determined that it was advantageous to directly integrate the new functionality into Innoslate® as it already contained most of the functionality required. However, to improve the usability of the tool for the SOP developers, we needed to hide a lot of the functionality of Innoslate®. New logos and themes were also adjusted as part of this process. The end result provides the users with three options: 1) Innoslate® only; 2) Sopatra® only; and 3) combined Innoslate® and Sopatra®.

OPERATIONAL TEST AND EVALUATION

The first part of the operational test and evaluation (OT&E) was carried out by employees of George Mason University (GMU) and SPEC Innovations to provide a first round of user feedback to ensure the appropriate functionality of Sopatra®. The next phase of testing included using a class of students from George Mason University. These students were studying usability in systems as part of their graduate work. They identified many issues that the SPEC Innovations software developers were able to resolve quickly.

As part of testing process, SPEC Innovations and GMU also held a tutorial workshop at the INCOSE International Symposium 2021. This virtual workshop included the opportunity for attendees to use the beta version of Sopatra®. Feedback from users was used to adjust some part of the User Interface, enhancing the usability of the tool.

A final set of users from NASA, the US Navy, and participants from the Airline industry also used the tool and provided valuable feedback.

CONTINUED DEVELOPMENT

The final product, Sopatra® v.1.0, provides users the benefit of a rapid way to convert text-based SOPs into an Action Diagram model and verify them using the discrete event and Monte Carlo simulations. Figure 5 shows the output of the Monte Carlo calculations for 1,000 iterations of a complex airline model. In this particular example the failure rate was quite high (75%), hence the large amount of red dots and bars in the ToP vs AOTW and PBT charts respectively. This result may mean that significant work is required to enhance the SOP to perform the tasks within the time available.

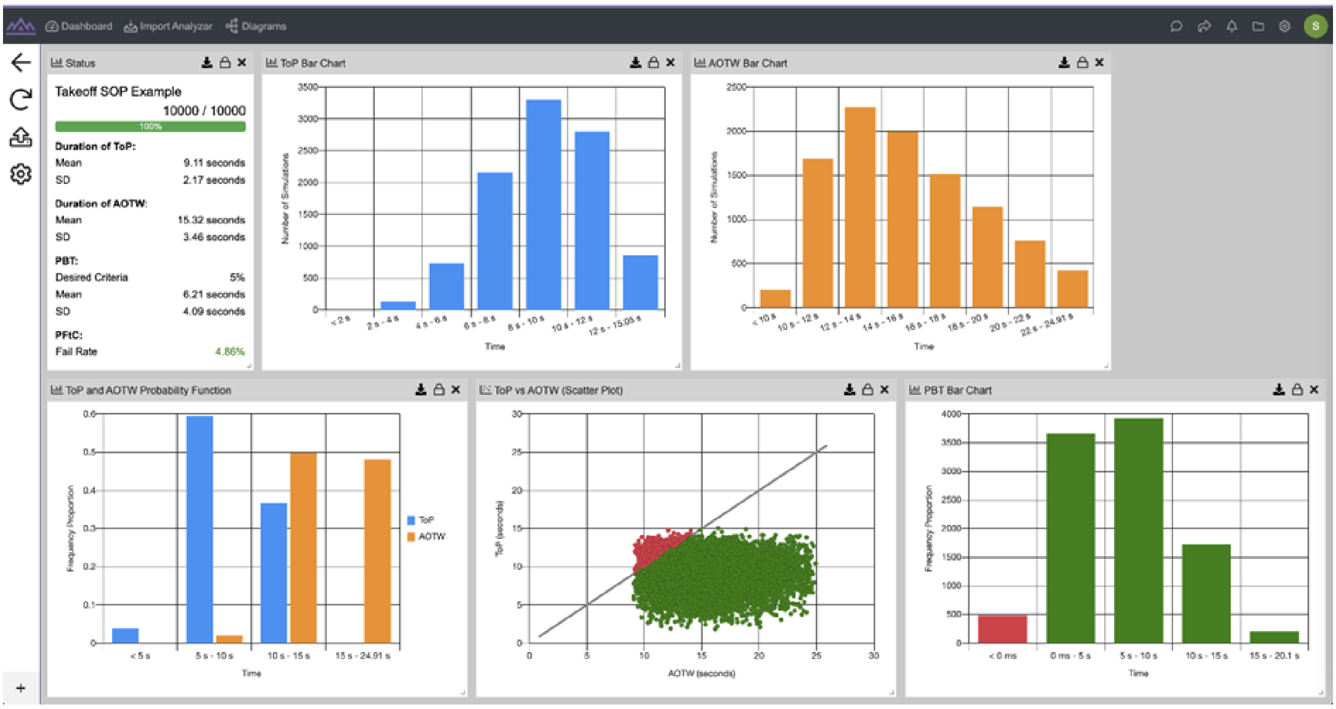


Figure 5. Sopatra® provides a means for SOP developers to rapidly assess the feasibility of their processes.

We will continue to explore more complicated forms of SOPs and new technologies, such as machine learning (ML) as part of a Phase 2 effort. Ideally, over time the community will move from a text-based to a model-based approach thus aiding the SOP designers in more rapidly developing processes and procedures that meet the time-critical, safety-critical mission needs, while instilling greater confidence in results. Further research into the application of machine-learning, verbal input, and holographic mixed reality will enhance these capabilities and enhance SOP development even more.

ACKNOWLEDGMENTS

We thank our colleagues from George Mason University, Dr. Lance Sherry and Ms. Jomana Bashatah, whose research formed the basis for the development of Sopatra®. We also thank the primary software developer, Mr. Patrick Hartmann, and marketing director, Ms. Elizabeth Steiner for their contributions.

REFERENCES

[1] Barshi, I., Mauro, R., Degani, A., Loukopoulou, L. (2016). Designing flightdeck procedures (NASA/TM-2016–219421). Washington, DC: NASA.

[2] Card, S. K., Moran, T. P., Newell, A. (1980). Communications of the ACM, 23, 396–410.

[3] Dam, S.H. Real MBSE: Model-Based Systems Engineering using LML and Innoslate®, 2019. Available at www.amazon.com.

[4] Asaf Degani, Alex Kirlik. (1995) Modes in human-automation interaction: Initial observations about a modeling approach. In the proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century, Vol. 4, 3443-3450.

[5] Keyser, V. (1995). Time in ergonomics research. Ergonomics, 38, 1639–1660.

[6] Gawron, V. J. (2000). Human performance measures handbook. Mahwah, NJ: Lawrence Erlbaum.

[7] Hollnagel, E. (1998). Cognitive reliability and error analysis method: CREAM. Oxford, UK: Elsevier.

[8] John, B., Vera, A., Matessa, M., Freed, M., Remington, R. (2002). Automating CPM-GOMS. In Proceedings of CHI '02 (pp. 147–154). New York, NY: ACM.

[9] Kourdali, H., Sherry, L. (2017). Available Operational Time Window (AOTW): A Method for Evaluating and Monitoring Airline Procedures. Journal of Cognitive Engineering and Decision Making, August 31, 2017.

[10] Kourdali, H., Sherry, L. (2016). A system engineering method for analysis and simulation of standard operational procedures. Paper presented at 2016 HCIAero, Paris, France.

[11] Laudeman, I. V., Palmer, E. A. (1995). Quantitative measurement of observed workload in the analysis of aircrew performance. International Journal of Aviation Psychology, 5, 187–198.

[12] Lifecycle Modeling Language (LML) Standard, v1.1, LML Steering Committee (2014). Available at www.lifecyclemodeling.org.

[13] Raby, M., Wickens, C. D. (1994). Strategic workload management and decision biases in aviation. International Journal of Aviation Psychology, 4, 211–240.

[14] Rothrock, L. (2001). Using time windows to evaluate operator performance. International Journal of Cognitive Ergonomics, 5, 1–21.

[15] Sherry, L., Bashata, J., Dam, S. H., Flenniken, L. (2020). Analyzing Standard Operating Procedures Using MBSE/LML Action Diagrams.

[16] Stimpson, A. J., Ryan, J. C., Cummings, M. L. (2016). Assessing pilot workload in single-pilot operations with advanced autonomy. In Proceedings of the Human Factors and Ergonomics Society 60th Annual Meeting (pp. 675–679). Santa Monica, CA: Human Factors and Ergonomics Society.

[17] Merriam-Webster Online, www.merriam-webster.com, accessed June 21, 2021.

SYSTEMS ENGINEERING RESOURCES

INCOSE INSIGHT Practitioners Magazine Sep. 2021 Released

Another excellent edition of INSIGHT, INCOSE's Practitioner Magazine published by Wiley, has just been released. Electronic subscriptions to INSIGHT are available as a member benefit to INCOSE members. Hard-copy subscriptions to INSIGHT are available for purchase by INCOSE members for one membership year, and to the public. Click here to join INCOSE!

The issue theme is the social dimensions of systems. Content includes:

"Perceived Conflicts of Systems Engineering in Early-Stage Research and Development" by Michael DiMario, Gary Mastin, Heidi Hahn, Ann Hodges, and Nick Lombardo discusses the difficulty of introducing systems engineering to the research and early development process and the inclination perspectives of researchers, engineers, and managers. The article offers potential means to manage the cultural transformation of early adoption of right-sized systems engineering in Early Stage Research & Development (ESR&D) and reverse the attitudinal positions.

"Incorporating the Role(s) of Human Actors in Complex System Design for Safety and Security" by Elizabeth Fleming and Adam Williams outlines the system context lenses to understand how to include various roles of human actors into systems engineering design. Several exemplar applications of these organizing lenses are summarized and used to highlight more generalized insights for the broader systems engineering community.

"An Agile Systems Engineering Analysis of Socio-technical Aspects of a University-built CubeSat" by Evelyn Honoré-Livermore, Joseph L. Garrett, Ron Lyells, Robert (Rock) Angier, and Bob Epps presents the results of an exploratory case study on a university CubeSat team developing an earth observation satellite. Formal analysis of agile systems engineering helps improve success throughout the CubeSat lifecycle. The authors apply the INCOSE Agile SE WG decision guidance method for applying agile system engineering method to identify areas in which the project organization can improve to become more agile in three specific problem spaces: customer problem space, solution space, and product development space. The analysis process leads to valuable insights about how the project organization of an academic project differs from that of industry. Additionally, the results indicate that areas such as stakeholder management and support environment can be factors that would benefit more from agile responsiveness.

"To Get Systems Engineers Interested in Social Dimensions, Give Them a Social Optimization Problem" by Tom McDermott and Molly Nadolski presents a case study on student-led implementation trades for urban electrical microgrids that optimize community sustainability and resilience. In this case study, the students used formal models of non-traditional socioeconomic variables such as availability, energy burden on residents, and local jobs created. The case study presents a straightforward process that considers typically overlooked social requirements and metrics in systems engineering design. The authors present this as both an example learning framework and a broader call to define and standardize systems engineering methods, processes, and tools for increased integration of social dimensions as functional requirements in future systems.

"Applying Behavioral Science to Agile Practice Evolution" by Larri Rosser and Brian Ganus states the observation that certain approaches work well in agile realization of products and services is not accidental, but rooted in the study of psychology, sociology, and human performance. For example, the "ideal agile team size" of 7 plus or minus 2 not only works but is supported by psycho-social

theories such as the Ringleman effect, social channel capacity and short-term memory limitations. Examples of similar relations between behavioral science and agile patterns abound – preferred planning horizons, methods of estimating effort and approaches to scaling agile all relate to our understanding of human behavior individually and in groups. This article explores such relationships with the intent to provide agile practitioners with information about the underpinning of practices, and social scientists with examples of how their work contributes to the improvement of agile practices.

"Detecting and Mitigating Social Dysfunction within Systems of Systems" by Mike Yokell elaborates on a means of assessing the managerial relationships between the organizations that own constituent systems within a system of systems (SoS), with a goal of detecting social dysfunction that could adversely affect operations. For each of the relationship types, or affinity options, tangible, actionable guidance is offered that could help mitigate the social and operational dysfunctions. Results from a case study are included to illustrate the application, detection, and successful mitigation of social dysfunction within a system of systems.

"The emergent properties of an ethical leadership when aligned with the Systems Engineering Handbook and Code of Ethics" by Anabel Fraga analyzes the definitions found in the current Code of Ethics and Handbook regarding ethical leadership, its implications, and its application. These definitions are explained and aligned to the ethical systems engineering idea. Also, examples of ethical behavior are introduced to explain emergent properties. It exemplifies that applying ethical leadership works in favor of the development of successful systems.

"Application of Model-Based System Architecture Process (MBSAP) to a Complex Problem with Social Dimensions: Utilization in Outpatient Imaging Centers" by Jill Speece and Kamran Eftekhari Shahroudi provides a comprehensive and visually understandable framework for system development. The primary social dimensions in outpatient imaging are the customer dimension, planning dimension, operations dimension, and technical dimension. Each of these dimensions has stakeholders with a diverse set of needs that must be well-understood and incorporated into the requirements. This paper presents an architecture for a system that utilizes all available exam time slots without a dependency on modifying patient behavior to prevent same day missed appointments. The MBSAP artifacts are the starting point for making the system a reality with stakeholders and finding the right balance between separate social dimensional measures.

"Bridge the Partisan Divide and Develop Effective Policies with Systems Engineering" by Jim Hartung describes a simple six-step systems engineering process for optimizing social, economic, and political systems. Second, he illustrates this process with two examples: (1) development of a nonpartisan tax reform proposal that balances the U.S. federal budget and addresses key societal problems without increasing the economic burden on taxpayers and (2) development of a nonpartisan plan for the United States to achieve the United Nations Sustainable Development Goals and address other urgent problems. Third, he discusses how lawmakers and policymakers can incorporate systems engineering into the lawmaking process. Many of the ideas presented here also apply to other countries.
More information

CESAM Pocket Guide = CESAMES Systems Architecting Method

The CESAM Pocket Guide = CESAMES Systems Architecting Method, is now available in an updated short (38-page) version. The CESAM Framework is widely used, said by CESAM to have been used with more than 2,000 systems/projects in a variety of industries in 20 countries. Under the framework, three core visions can be used to specify any complex system/project.

A PDF file of the CESAM Pocket Guide, in English, is free to download here. A version in simplified Chinese is downloadable at the same location.

Readers are also reminded of PPI's 5-day Architectural Design course. This course provides an integrated approach to the set of technical design process disciplines conducive to development success. These disciplines combine with technology knowledge to contribute to the satisfaction of requirements and maximization of system effectiveness, enhancing value delivery and reducing risk to the enterprise. See detail here.

Definitions and Standards: Building Information Model (BIM)

The abbreviation 'BIM' can be used in several ways. One definition is "A BIM is a digital representation of physical and functional characteristics of a building. As such, it serves as a shared knowledge resource for information about a building, forming a reliable basis for decisions during its life-cycle from inception onward." Source: BuildingSMART

Authors of Australia's 'National Guidelines for Digital Modelling' describe some characteristics of BIM in the following way: A model needs only two essential characteristics to be described as a BIM model. The first is that it must be a three-dimensional representation of a building (or other facility) based on objects, and second, it must include some information in the model or the properties about the objects beyond the graphical representation. Source: National Guidelines for Digital Modelling, CRC Construction Innovation.

Building Information Model - Product: "An object-based digital representation of the physical and functional characteristics of a facility. The Building Information Model serves as a shared knowledge resource for information about a facility, forming a reliable basis for decisions during its lifecycle from inception onward". Source: Department of Veterans Affairs (USA)

Building Information Modeling - Process: "A collection of defined model uses, workflows, and modeling methods used to achieve specific, repeatable, and reliable information results from the model. Modeling methods affect the quality of the information generated from the model. When and why a model is used and shared impacts the effective and efficient use of BIM for desired project outcomes and decision support". Source: Department of Veterans Affairs (USA)

"Building Information Modeling (BIM) is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition. A basic premise of BIM is collaboration by different stakeholders at different phases of the life cycle of a facility to insert, extract, update or modify information in the BIM to support and reflect the roles of that stakeholder." Source: NBIMS - National Building Information Model Standard Project (USA)

Building information modeling (BIM) is a process supported by various tools, technologies and contracts involving the generation and management of digital representations of physical and functional characteristics of buildings. Building information models (BIMs) are computer files (often but not always in proprietary formats and containing proprietary data) which can be extracted, exchanged or networked to support decision-making regarding a built asset. BIM software is used by individuals, businesses and government agencies who plan, design, construct, operate and maintain buildings and diverse physical infrastructures, such as water, refuse, electricity, gas, communication utilities, roads, railways, bridges, ports and tunnels.

The concept of BIM has been in development since the 1970s, but BIM only became an agreed term in the early 2000s. Development of standards and adoption of BIM has progressed at different rates in different countries; standards developed in the United Kingdom from 2007 onwards have formed the basis of international standard ISO 19650, Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling. Part 1 of this standard, Concepts and principles, was released in December 2018.

Other parts of this standard are:

- Part 2: Delivery phase of the assets (December, 2018)
- Part 3: Operational phase of the assets (July 2020)
- Part 4: Information exchange (Under development, currently well-advanced at DIS - Enquiry stage, draft available for purchase)
- Part 5: Security-minded approach to information management (June, 2020)
- Part 6: Health and Safety (Under development as ISO/AWS 19650-6, currently 20 - Preparatory stage, in development by a Working Group)

In a systems engineering context, the BIM is an instantiation of the digital twin, and has a direct relation to the digital thread, even more so in the context of a built structure as an element of a larger system.

More information: “BIM in Practice”, Australian Institute of Architects, is downloadable here.

Functional Mock-up Interface (FMI) Standard

The Functional Mock-up Interface (FMI) is a free standard that defines a container and an interface to exchange dynamic models using a combination of XML files, binaries and C code zipped into a single file. FMI is supported by more than 150 engineering software tools and is maintained as a Modelica Association Project on GitHub. The code is released under the 2-Clause BSD license, the documentation under the CC-BY-SA License. Access and use of the FMI standard is free of charge. However, tool vendors may charge for their tools to support the FMI standard.

FMI is currently at V2.0.2, with V3 in Beta.

FMI is being used by companies for model exchange and as a co-simulation format at system level, enabling the exchange of models with internal and external partners using different modeling tools. FMI is becoming an important building block in the efficient creation of interdisciplinary, multi-level digital twins - from rail and gas turbine engineering to virtual commissioning in the process industry and operational support in manufacturing plants. Saab has stated "we see the FMI standard as an enabler for scalable and tool neutral integration of simulation models from different technical disciplines, developed by different internal teams or by external partners”.

More information here

Website: https://resources.jamasoftware.com/

This website of tool vendor Jama Software contains scores of white papers, webinars, eBooks and case studies on aspects of systems engineering. Although the content (naturally) has a marketing orientation, it is much more than just marketing, representing a treasure trove of potentially valuable information. A small sample of titles includes:

- The Complete Guide to ISO 13485 for Medical Devices
- Best Practices to Accelerate Your Automotive Spice (ASPICE) Capabilities
- Reduce Project Risk in the Product Development Process
- MBSE Made Easy - Overcoming the Organizational Challenges

Book: Safety Risk Management for Medical Devices 2nd Edition

Following the success of the 1st edition of 'Safety Risk Management for Medical Devices' by PPI 's Bijan Elahi, the book returns in a 2nd, expanded edition. The book is due to be released on 6 December 2021 but eager readers can secure their copy now on pre-order from Elsevier.

From the Elsevier website:

Safety Risk Management for Medical Devices, Second Edition teaches the essential safety risk management methodologies for medical devices compliant with the requirements of ISO 14971:2019. Focusing exclusively on safety risk assessment practices required in the MedTech sector, the book outlines sensible, easily comprehensible, state-of the-art methodologies that are rooted in current industry best practices, addressing safety risk management of medical devices, thus making it useful for those in the MedTech sector who are responsible for safety risk management or need to understand risk management, including design engineers, product engineers, development engineers, software engineers, Quality assurance and regulatory affairs.

Graduate-level engineering students with an interest in medical devices will also benefit from this book. The new edition has been fully updated to reflect the state-of-the-art in this fast changing field. It offers guidance on developing and commercializing medical devices in line with the most current international standards and regulations.

Paperback ISBN: 9780323857550

Imprint: Academic Press

Page Count: 424

Order your copy today

Website: https://www.ssse.ch/content/mbse

This page, the Model-based Systems Engineering (MBSE) knowledge exchange of the Swiss Chapter of INCOSE, offers a number of downloadable presentations on aspects of MBSE, from "where to start" to advocacy of Arcadia/Capella.

The IIBA Online Library

The 27,000 member International Institute of Business Analysis is a non-profit professional association formed in 2003 with the purpose of supporting and promoting the discipline of business analysis. IIBA helps business analysts develop their skills and further their careers by providing access to relevant content.

IIBA's member-only online reading library includes access to a wide selection of books on different topics including agile methodologies and techniques; business process modeling and management; business rules; data modeling; enterprise analysis; elicitation techniques; enabling change; structured analysis methods; UML; and underlying competencies (facilitation, decision analysis, negotiation and conflict resolution, leadership and influencing).

All material available in the library has been reviewed and selected by the IIBA Professional Development team and includes most of the books used in the development of the Business Analysis Body of Knowledge (BABOK®) Guide, making it a key reference for anyone who is studying for the Certified Business Analysis Professional™ (CBAP®) exam.

The IIBA® has now added on-demand videos to the library. With the addition of this new service, members have access to over 300 professional development titles covering business analysis and related topics and leadership solution videos featuring global thought-leaders (CEOs, Executives and Top Business Authors) that support BA professional development. IIBA has selected videos that offer different learning formats including reflective and actionable style videos to support leaders at all levels.

Access to the IIBA online library is a benefit included in its Membership fee. You may join here.

INCOSE UK "Don't Panic" Series Available as eBooks

INCOSE UK's "Don't Panic!" series is now available in eBooks format. The first book in the series, Don't Panic! The Absolute Beginner's Guide to Model-Based Systems Engineering, was published in 2017 as a paperback. Since then, INCOSE UK has added three more books to the series. The books that are now available as eBooks are:

- Don't Panic! The Absolute Beginners Guide to Model-Based Systems Engineering
- Don't Panic! The Absolute Beginners Guide to Managing Interfaces, by PPI Principal Consultant and training presenter Paul Davies. This eBook complements PPI's training course *Interface Engineering and Management*, delivered worldwide over two days or four half-days as both open-registration and corporate training. More details here.
- Don't Panic! The Absolute Beginners Guide to Architecture Frameworks
- Don't Panic! The Absolute Beginners Guide to Architecture and Architecting
- Don't Panic! The Absolute Beginners Guide to Model-Based Systems Engineering.

INCOSE UK's first publication, Implementing MBSE Into Your Business—The Trinity Approach is also available in eBook format.

The eBooks and paperbacks are available here.

Webpage: https://www.gfse.de/newsletter-nichtmitglieder.html

This webpage contains an archive of the monthly newsletters of the GfSE, the German Chapter of INCOSE.

SYSTEMS ENGINEERING IN SOCIETY

Featured Society: The International Association for the Engineering Modelling, Analysis and Simulation Community (NAFEMS)

NAFEMS is an individual membership-based international society that aims to provide knowledge, international collaboration and educational opportunities for the use and validation of engineering simulation.

The specific goals of NAFEMS are to:

- Be the recognised independent authority and trusted source for communicating engineering simulation knowledge, and for sharing best engineering modeling, analysis, and overall simulation practices in developing reliable products and innovative solutions.
- Facilitate unbiased worldwide communication and collaboration between industries, academia, and government organizations for the advancement of best practice in multidisciplinary engineering simulation expertise.
- Develop and deliver training and personal educational opportunities that are aligned with the rapidly-advancing engineering simulation technologies.
- Have a strong impact on product quality, development efficiency and safety.

NAFEMS working groups play a major role in the activities of the organization. Drawn from experienced international membership, the technical working groups identify areas of interest to the community, gaps in educational materials, requirements for further research, and opportunities for collaboration in engineering analysis and simulation. The groups draw together a blend of leading engineering practitioners, academic researchers, and software vendors, giving independent insight and perspective into every aspect of engineering analysis and simulation.

The Society engages with the analysis and simulation community with more than fifty events each year, including conferences, seminars, workshops, open forums and webinars.

NAFEM's PSE (Professional Simulation Engineer) Certification allows engineers and analysts to demonstrate competencies acquired throughout their professional careers. Independently assessed by NAFEMS, the Certification enables individuals to gain recognition for their level of competency and experience, as well as enabling industry to identify suitable and qualified personnel.

The NAFEMS Resource Centre is a database that contains thousands of presentations, books, webinar recordings, magazine articles, journals, and more, which have been categorized and tagged in one central accessible location. There are currently over 1,500 resources available, with this number growing weekly. Members have access to numerous types of resources as part of their membership, including presentations from previous events, conference papers, webinar archives, and much more.

Over the past 35 years, NAFEMS has produced over 200 books covering the cutting-edge of engineering analysis and simulation. Many of the standards used today were tested against benchmarks produced by NAFEMS, whilst most of its publications have been written by industry experts, reviewed extensively by its working groups, and are available at significant discounts to its members. Each year, NAFEMS produces a plethora of how-to guides, industry surveys, technical textbooks, academic journals, best-practice guidelines and more.

NAFEMS and the International Council on Systems Engineering (INCOSE) have established a relationship for mutual participation and collaboration for the advancement of engineering simulation and model based systems engineering. This collaboration includes the implementation of a joint cross-organizational working group on Systems Modeling & Simulation. The mission of the SMSWG is to develop a vendor-neutral, end-user driven consortium that not only promotes the advancement of the technology and practices associated with integration of engineering simulation and systems engineering, but also acts as the advisory body to drive strategic direction for technology development and international standards in the space of complex engineering.

More information about this working group is available here:
https://www.omgwiki.org/MBSE/doku.php?id=mbse:smswg

See also the NAFEMS Systems Modeling & Simulation (SMS) Working Group home page:
https://www.nafems.org/community/working-groups/systems-modeling-simulation/

More information on NAFEMS: https://www.nafems.org/

Standards: IEEE 7000™-2021, IEEE Standard Model Process for Addressing Ethical Concerns during System Design

Avoiding risk is a key concern for any organization but focusing in product development solely on physical harms won't provide a full understanding of an end user's experience of what you build. Artificial Intelligence Systems (AIS) driving development of many products and services today are based on algorithms invisible to users. These algorithms affect users' data, identity, and values. Despite the best intentions of a developer, without having a methodology to analyze and test how an end user interprets a product, service or system, a design process will likely prioritize the values of its creators, not its users. IEEE 7000-2021 provides a values-oriented methodology that complements the other tools of systems engineering, facilitating responsible innovation in the algorithmic era.

IEEE 7000-2021 standard provides:

- a system engineering standard approach integrating human and social values into systems engineering and design.
- processes for engineers to translate stakeholder values and ethical considerations into system requirements and design practices.
- a systematic, transparent, and traceable approach to address ethically-oriented regulatory obligations in the design of autonomous intelligent systems.

IEEE 7000-2021 standard is intended for use by:

- large, medium, and small businesses dedicated to including broader ethical value criteria and concerns while designing, developing or operating AI or other technical systems.
- innovating organizations engaged in concept exploration, system requirements definition, or development of new or revised products or services. The standard is intended to help organizations to build products with a more refined and nuanced value proposition and with less risk.
- A review by PPI of IEEE 7000-2021 will appear in a subsequent edition of PPI SyEN. This review will compare IEEE 7000-2021 against already-used means of including ethical and social values in product development, in particular Multiple Attribute Utility Theory.

More information here.

FINAL THOUGHTS, FROM SYENNA

Dear reader (assuming I have one),

This month I'd like to tell you about Irma, a good friend of mine who was on holiday in Copenhagen. It so happens that she went to school with the Queen of Denmark, so she sought an invitation to catch up with her old friend, which was indeed granted by the Amalienborg Castle Secretariat.

After the initial hugs and reminiscences of school days etc., the Queen asked Irma what she was up to these days. "Well I run a consultancy company, and if you're interested I can tell you about our latest offering, which we call IRMA (Integrated Requirements Management and Analysis)". "Oh yes", said the Queen with practised enthusiasm, "do tell me something about that".

Irma proceeded to talk around Figure 1 in some detail. "The thing is that many companies invest in requirements management; buying licenses, training their people, setting up procedures etc. They think this will fix their requirements problems but of course it doesn't because they end up doing a great job of managing rubbish requirements. What they also need to think about is investing in requirements analysis; which is more fundamental because it means you solve the right problem…"

IRMA – Integrated Requirements Management and Analysis

Quality of management (attributes)	Quality of expression (characteristics): Rubbish	Good	Great
Great	A great job of managing rubbish requirements	Overkill except for the most critical requirements	Overkill except for the most critical requirements
Good	A good job of managing rubbish requirements	A good job of managing good requirements	A good job of managing great requirements
Rubbish	A rubbish job of managing rubbish requirements	A rubbish job of managing good requirements	A rubbish job of managing great requirements

Figure 1 – IRMA overview

By this time the Queen's eyelids, despite decades of training, had lost the battle with gravity. Luckily the clock struck the hour and she came round again. "So interesting!", said the Queen. "You must tell me more next time!".

"OK", said Irma, "Let me go away and do a small demo for you. Here's a question: if you could live anywhere in the World, what would it be like?"

"Well", said the Queen, "it would have to be exactly like Copenhagen, but if there was one thing I could change, I'd love to have longer daylight hours. I don't mind the cold, but I do suffer from lack of light, especially in the winter".

A few months later, Irma returned to the Castle, ready to show the results of the demo to the Queen.

“I really enjoyed your last visit”, said the Queen. “What have you got to show me for our demo?”

Irma fired up her laptop, saying: “All you have to do is type in the name of your city, and it will show you where it could be in the World whilst meeting your requirements, which we captured in IRMA. Now, if you just enter ‘Copenhagen, Denmark’ into the user-friendly field, see what happens next....” (see Figure 2).

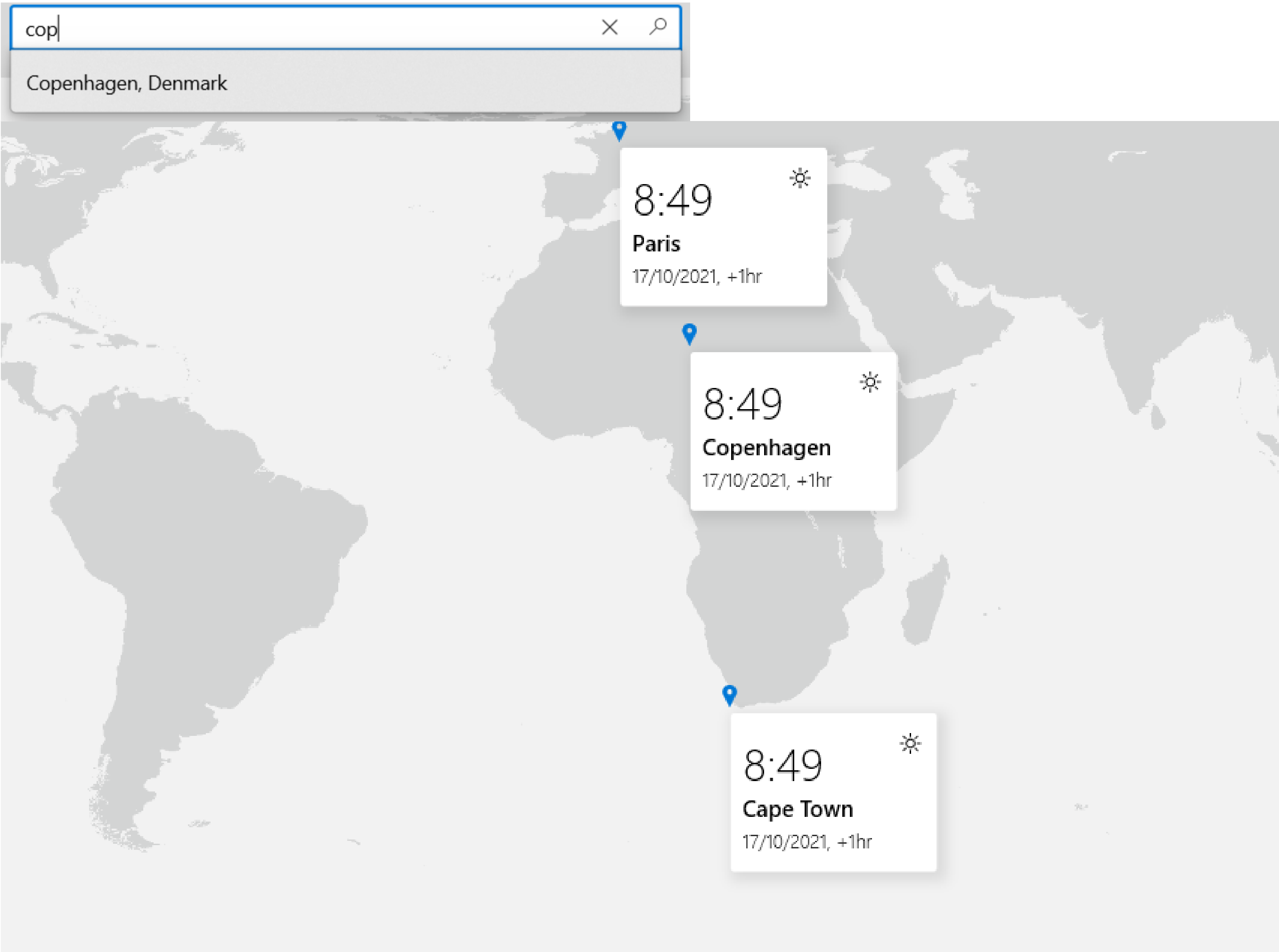


Figure 2 – Suggested location for Copenhagen

“Well”, said the Queen, “that would certainly give me longer daylight hours, which was a goal, but what about my requirement to be otherwise just like the actual Copenhagen?”

Irma, with some embarrassment, replied that this was just a first pass (although it had already been released to millions of customers as part of the company’s evolutionary approach). They had regrettably spent most of their efforts on the Queen’s goal, at the expense of the requirements. Doubtless they could find a better solution next time. The good news was that IRMA would now be enhanced, to differentiate between requirements and goals. The Queen had been most helpful by taking part in the company’s learn-do-learn process.