PPI SyEN

SYSTEMS ENGINEERING NEWSJOURNAL

EDITION 104 | SEPT 2021

SE in Society

AGILE METHODOLOGY
System architecture models

DESIGN THINKING
Proven SE principles

LOGICAL MODELING
A comparison of tools

REQUIREMENTS OVERVIEW
The ReqIF standard

A PROJECT PERFORMANCE INTERNATIONAL PUBLICATION | PPI-INT.COM

PPI SyEN

EMAIL: PPISyEN@PPI-Int.com

PUBLISHER

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

Welcome to the September 2021 edition of PPI SyEN!

It feels like just the other day that I was I turning the digital pages while reviewing the August edition of PPI SyEN but in the blink of an eye here we are with the September edition of PPI SyEN. This edition contains something for everyone in our readership. Whether you're looking for upcoming events in systems engineering, the latest on your favorite SE software tools or are thirsty for thought-provoking articles – this edition has you covered!

In this edition of PPI SyEN we also introduce a new section titled 'SE in Society'. This section will cover notable organizations, academic institutions and societies in systems engineering. In future expect content about youth programmes dedicated to spreading the benefits of systems engineering, university competitions encouraging SE development of engineering projects and much more. If you're involved in a systems engineering-related initiative we'd love to hear from you! You could bring some exposure to your community with a feature in an upcoming PPI SyEN edition. Send as an email at ppisyen@ppi-int.com with a brief description of the initiative and we'll be in touch.

It really does take a village (well kind of) to make PPI SyEN happen every month so as always a special thanks goes to our authors and contributors. This month we have riveting content by Juan Navas on 'Architecture Models as Enablers of Agility', the first in a 3-part series in which Juan unpacks a very palatable case study as an illustration of how to make use of applying architecture models as part of agile approach to development. I thoroughly enjoyed Episode 1 in this edition and look forward to Episode 2 and 3 coming in October and November respectively.

John Fitch takes us on a journey of design thinking with 'A Fresh Look at Design Thinking in Light of Proven Systems Engineering Principles'. As someone who has done a short course in design thinking out of interest for this hot topic, this article really paints a clear picture of what design thinking is and what it isn't.

PPI Contributors Alwyn Smit and Wioletta Kowalczyk provide us wth an 'Overview of the Requirements Interchange Format (ReqIF)' and 'A Comparison of the Usability of Fiev MBSE Language/Tool Combinations' – two quick and informative reads for those on the move. Convinced yet that there is something in this edition for everyone?

René King
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**

Addressing Important Concepts by Robert Halligan

Recently I had a delegate on a course that asked for clarificaion on the following concepts. These concepts are often confused/misunderstood in the SE world so I thought it would be worthwhile to repeat them here.

1. The difference between CONOPS, CONUSE and OCD

For a given system, a CONUSE and an OCD are the same thing, a system-centric description of intended use in terms of users and their characteristics, uses, for each user and use how it is to be used, and the external conditions intended or expected during use.

For a given Capability/Enterprise/Business system, a CONOPS is an operational solution description, a description of that part of the solution that serves an end-use purpose, in terms of key solution elements, their key characteristics and the concept of their interoperation to achieve key Capability/Enterprise/Business outcomes. The term is only used for systems that are a mixture of humans and technology (usually) or entirely human (occasionally).

2. The difference between: Allocated Baseline, Functional Baseline and Product Baseline

The baselines you mention are always with reference to a given system (system of interest) that may in context be a subsystem.

Functional Baseline: a terrible but common name for a baselined problem definition.

Allocated Baseline: a design baseline one physical level below the system that is the subject of the baseline, populated by the requirements specifications of the solution elements at that level, including the data describing how the solution elements are to be configured with respect to one another to comprise the whole.

Product Baseline; a design baseline containing all the necessary information from which the system can be built, populated by the requirements specifications of the solution elements at all physical levels, including the data describing how the solution elements are to be configured with respect to one another to comprise the whole, through the physical levels. In practice, this means the identification and specification of all parts and materials and associated configuration data, including, for developmental software elements, software construction down to lines of code and compilation data. The Product Baseline excludes the design of the production system, unless the system of interest is a production system!

These are general engineering definitions. Each of these baselines may or may not include specification of the requirements for evidence of the system being in compliance.

SYSTEMS ENGINEERING NEWS

Recent events and updates in the field of systems engineering

Checklists for ARP4754A Civil Aircraft Development Standard

SAE standard ARP4754A on Guidelines for Development of Civil Aircraft and Systems naturally has a strong systems engineering orientation. SAE International, a society with the mission of connecting and educating engineers while promoting, developing and advancing aerospace, commercial vehicle and automotive engineering, has recently published a series of checklists to this important standard. The objective for the checklists is to help ease the work associated with compliance with the ARP4754A standard. Here are some examples of the checklists:

ARP4754A Objectives Checklist

ARP4754A Project Specific Certification Plan (PSCP) Checklist

ARP4754A System Development Plan Checklist

ARP4754A System Process Assurance Plan Checklist

ARP4754A Requirements Management Plan Checklist

More information: Shitong Xing, email: shitong.xing@sae.org

INCOSE SEH 5th Edition in German

According to SSSE (Swiss Chapter of INCOSE), SSSE and GfSE (German Chapter of INCOSE) have agreed to join forces for the translation of the next edition, Edition 5, of the INCOSE Systems Engineering Handbook (SEH). The SEH 5th Edition is scheduled for publication mid-2023.

New Graduate Certificate in Systems Engineering from Australian National University (ANU)

The ANU College of Engineering and Computer Science, based in the Australian capital city of Canberra, ACT, has introduced a Graduate Certificate of Systems Engineering. This Graduate Certificate can be completed as a standalone postgraduate qualification or used as a pathway to a Master of Engineering at ANU. The next intake for applications is Semester 1, 2022.

There are many other ways to study systems engineering in Australia (apart from PPI and CTI's training!). Existing Australian university providers of systems engineering education are:

Undergraduate:

- Bond University – Systems Thinking & Management Modelling for Projects (single unit)
- Curtin University – Industrial and Systems Engineering (Major)
- James Cook University – Introduction to Systems Engineering and Project Management (single unit)
- University of Adelaide – Defence Systems (Major)
- University of South Australia – Principles of Systems Engineering (single unit)

Postgraduate:

- University of Melbourne – High Integrity Systems Engineering (single unit)
- University of New South Wales – Master of Systems Engineering
- University of South Australia – Graduate Certificate in Defence Systems Integration
- University of South Australia – Master of Defence Systems Integration

UK Study on the Value of MBSE Underway

Those who have developed complex systems with and without the aid of soundly-conceived MBSE know its value. But scientific-quality evidence of the value of MBSE is rare to non-existent. The UK Chapter of the International Council on Systems Engineering (INCOSE UK) is undertaking a study that aims to define the value MBSE can bring to organization. Specific objectives of the study are:

- To collate publications and research findings relating to the business case for MBSE
- To examine the role MBSE does or could play in adding value in various 'engineering' business models.

The MBSE Value team currently comprises James Towers of Scarecrow Consultants (lead), Andrew Pemberton of Thales, Alex Toth of Tothal Engineering/Jaguar Land Rover, Lucy Berthoud of University of Bristol and Joe Gregory of University of Bristol.

The group is actively engaged in the collation of research relating to the value, benefit and return-on-investment (ROI) provided by systems engineering and Model-Based Systems Engineering in particular. An experiment to collect quantitative and qualitative data in order to compare an MBSE and Document-Centric SE task is planned.

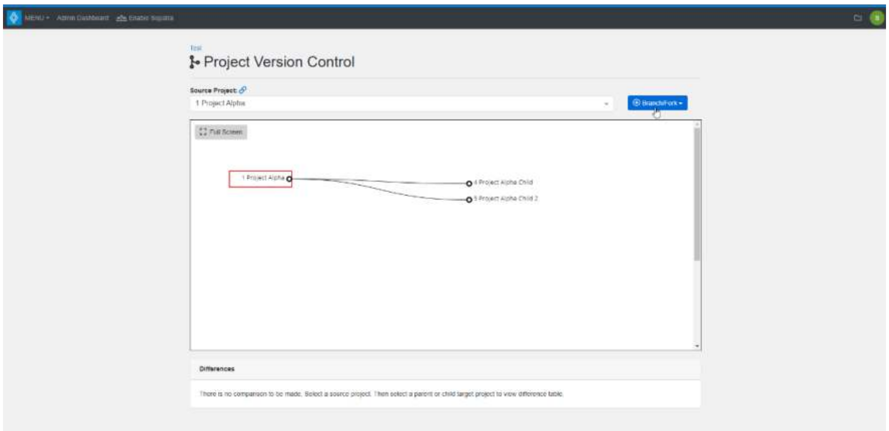
Links to 16 papers on the subject are provided at https://www.incosewiki.info/Model_Based_Systems_Engineering/index.php?title=MBSE_Value

Innoslate 4.5 Announcement

MBSE tool, Innoslate, was developed by SPEC Innovations in 2013. SPEC Innovations has launched a major feature release in September 2021 with Innoslate 4.5. Innoslate users can now utilize project management features such as Kanban boards, branching and forking, calendar, and Gantt charts. In addition, significant upgrades to the Action Diagram and simulators provide enhanced useability and reporting. Screenshots of some of the new features and enhancements are provided below:

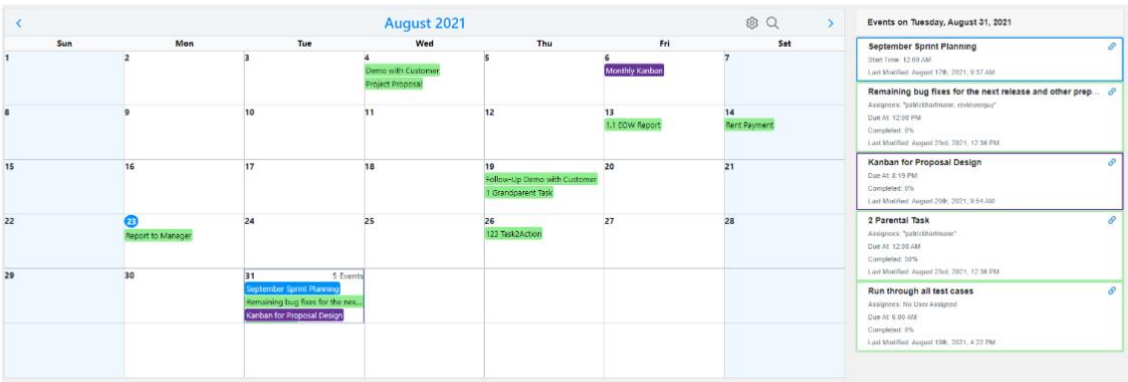
- Branching and Forking

Users can now branch and fork their projects as well as merge changes between linked projects.



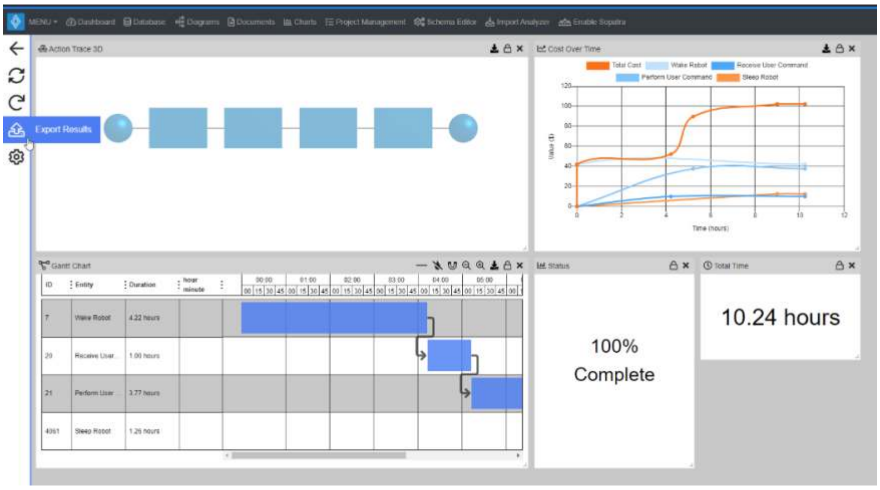
- New Project Management Dashboard

The Project Management dashboard offers a way for project managers and other users to view important upcoming dates and deadlines, Calendar events, track Kanban Board progress, and view hierarchical breakdowns of Kanban Boards. The dashboard can be customized with a built-in calendar, hierarchy, and board progress widgets.



- Simulator Updates

The Discrete Event Simulator and Monte Carlo User Interface has new configurable graphs and reports. Simulation results can be exported directly into an Innoslate artifact that will store the .SIM file. The artifact will present the finished simulation of the exported ‘Discrete Event’ simulation (or the ‘Monte Carlo’ simulation). This allows users to track or compare changes to a project’s simulation results.



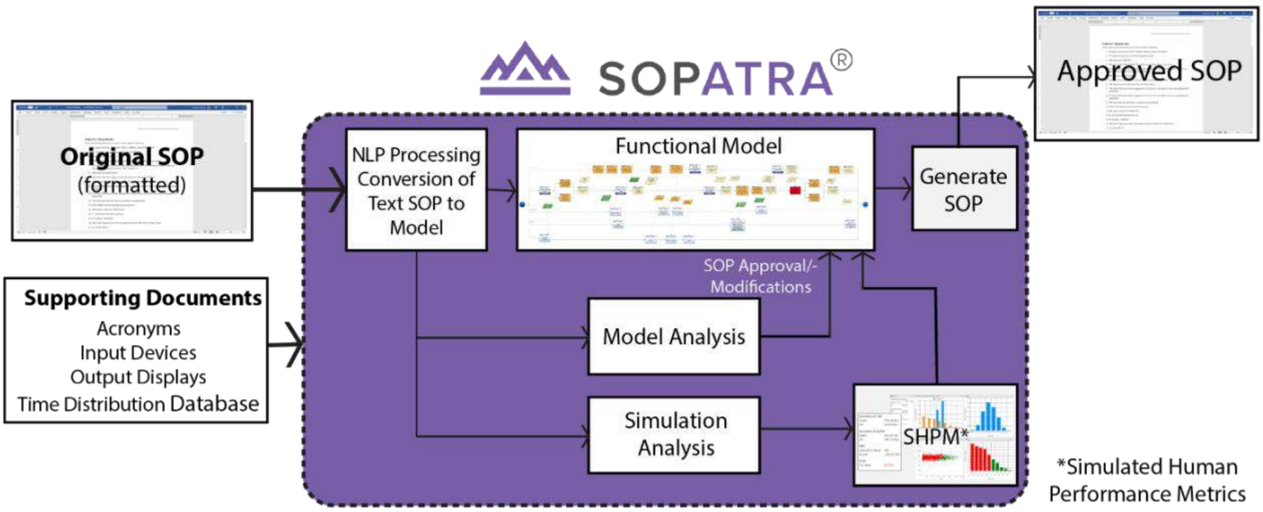
Find out more about the 4.5 release here.

Innoslate Launches a New Product: Sopatra

Sopatra presents a model-based approach to develop standard operating procedures for more effective and reliable SOPs. Sopatra uses artificial intelligence to convert your written processes into diagrams. The Monte Carlo simulator and discrete event simulator provide human performance metrics, including Sopatra's unique metrics: Allowable Operational Time Window (AOTW), Time of Procedure (ToP), and Procedure Buffer Time (PBT).

How Sopatra Works:

1. Import Standard Operation Procedure and support documents into Sopatra.
2. Natural Language Processing converts the text into a functional model.
3. Simulate the model using the Discrete Event Simulate to ensure the procedure executes.
4. Then use the Monte Carlo Simulator to verification.



Watch a video demonstrating how the software works here: https://www.youtube.com/watch?v=eGRp9Rc3Ero

Sopatra will be available September 14th, 2021, visit https://sopatra.innoslate.com/ for more information.

CONFERENCES, MEETINGS & WEBINARS

Upcoming events of relevance to systems engineering

SERC Sponsor Research Review

The Systems Engineering Research Center is hosting its annual Sponsor Review set to take place from 2-4 November 2021. The hybrid (physical-virtual) conference will be hosted in Washington DC on 2 November and hosted virtually on 3 and 4 November. This three-day event unites sectors of the systems engineering and acquisition research communities–government, industry, and academia–and provides an opportunity to share progress on research addressing the most challenging issues facing the Department of Defense (DOD) and other federal departments and agencies.

To view the full program or register for the event, visit the following site: https://sercuarc.org/research-reviews/2021-serc-annual-research-review/

International Congress of Engineering and Complex Systems (ICECS 2021)

Sidi Mohamed Ben Abdellah University (USMBA) and the Moroccan Association of Innovative Technologies (AMTI) are organizing the first edition of the International Congress of Engineering and Complex Systems (ICECS 2021) from 27-29 October 2021. This congress offers a forum and an excellent environment for researchers, university professors, students, industrialists and actors in the field of engineering and complex systems.

The main objectives of this congress are:

1. Focus on the main subjects in the theoretical and applied engineering and computer science subjects.
2. Share the state of the art on the one hand and discuss the problems on the other hand.
3. Bring together the scientific and technical community, Moroccan and foreign, in order to exhibit and publish the results of their research and their approaches.

Find out more about the conference here.

ENERGY 4.0: Digital Transformation Technology Exhibition

The world's demand for clean and sustainable energy production, new qualified workforce & performance while improving safety, process-operations planning and results is growing in today's industry. The development of new technologies is at the core of real solution to this complex scenario. Energy 4.0, presented by NRG events , is an exhibition dedicated to covering new technologies paving the way for the future. Register to hear from industry leaders at this exhibition who will be discussing :

- Digital Twins & OTS to optimize operation and maintenance of physical assets, systems, and processes.
- AR/VR & XR: Immersive learning for the future engineering workforce

- Artificial Intelligence, IOT & Machine Learning: Big data as a core to re-shape the industry
- Blockchain: A true disruptor for the energy industry
- Nuclear Small Modular Reactors SMRs and Advanced Reactors ARs as a flexible power generation solution

EMEA Workshop 2021

The INCOSE EMEA Workshop 2021 taking place from 28-29 October is the event for Systems Engineers from the (Europe, Middle East and Africa) region to contribute to the state of art in Systems Engineering. It is a biennial event. Unlike the International Symposium and the national conferences, there are no paper, panel, or tutorial presentations. Instead, attendees will spend two days working alongside fellow Systems Engineers who are there to make a difference.

This fourth INCOSE EMEA Workshop on Systems Engineering is organized by the Chapters of the EMEA Sector of INCOSE and hosted by AEIS (The Spanish Chapter of INCOSE).

View the event schedule here.

Webinar: Integrated MBSE Approach for Architecture Analysis & Requirement Validation

Date: October 20th

Time: 3pm-4pm GMT

Presenter: Alexandre Luc (Phoenix Integration)

Abstract from the website GfSE website:

Over the last few years, system engineers in all industries have been increasingly turning to Model Based Systems Engineering (MBSE) to meet the expectations placed upon them. This helps them designing ever more complex systems while reducing development cost and time, maximizing system performance, and improving the safety of the product. By aligning people, processes, and technology around a single vision of a product, MBSE promises to dramatically reduce the development cost and risk of complex systems. The challenge is the gap between systems engineers and engineering analysts that prevents MBSE from achieving its full potential.

In this presentation, we will discuss a new approach to unlock the promise of MBSE by connecting the system model with virtually any analysis/simulation model or workflow, assuring that the product vision remains in sync with the underlying analysis throughout the product lifecycle. This allows engineers to validate requirements, simulate system behavior, and carry out Multi-Disciplinary Analysis & Optimization (MDAO) to optimize the system design at any time during the design process. Development costs and risk are reduced because design problems can be identified and corrected early in the design lifecycle before they become too costly to fix. Rigorous traceability between requirements, design, and analysis results in improved quality.

Register for the webinar here:
https://www.gfse.de/events/165-integrated-mbse-approach-for-architecture-analysis-requirement-validation.html

FEATURED ARTICLES

Architecture Models as Enablers of Agility

By Juan Navas, Thales Corporate Engineering

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 1 – Let's warm-up

The context

The (fictive) company *Pythagoras* aims to develop and sell lightweight drone-based products for different markets: agriculture, aircraft exterior inspection, and public security enforcement. In addition to the unmanned autonomous vehicles themselves, such a product embeds features such as mission control and data analysis, manual and automated piloting, data acquisition, live data processing, data recording, live and post mission data analysis.

Marketing and business teams have done a great job identifying the targeted markets and customers, their expectations and their operational constraints. Nevertheless, there is still a lot of uncertainty regarding the scope of the product and the services that are to be provided to satisfy future customers. Moreover, drones' markets change very rapidly, and customers' expectations may vary accordingly. **Pythagoras has therefore decided to implement agility in the engineering process**: agility being defined as the ability to adapt to new circumstances quick enough.

Although the company can count on strong technical skills, and has in its catalog some of the building blocks of the future product, it has never integrated these building blocks with new ones to provide turnkey solutions. Such a mission will require adoption of more customer-oriented point of views, more rigor in performing systems engineering tasks and managing engineering data, and an enhanced comprehension of what the product is, and of the contribution of its parts to the product goals. **Pythagoras has therefore decided to implement Model-Based Systems Engineering (MBSE) practices** within its engineering processes.

The team

The development of the product will require the involvement of a significant amount of Pythagoras resources. In this case study, we put the spotlight on three of them:

- The product architecture team is strongly involved in the needs capture. Architects are in charge of formalizing the needs and designing the overall solution. They are accountable for the proper integration of constituent parts.
- The Integration Verification and Validation (IVV) team at system level is in charge of producing the test procedures, specifying the test means, running the actual tests and organizing the validation campaigns.
- The control software development team is made of embedded software engineers, specialized in motion control laws.

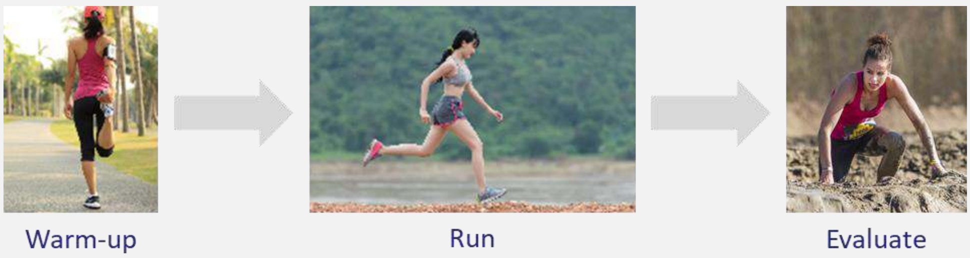
In this episode and the following ones, we will follow these three teams on their endeavors in developing the "PythaDrone Product", Pythagoras' lightweight drone-based product addressing multiple markets.

Where to start? First, let's warm-up

Why? Because you do not want to get hurt while doing an (engineering) effort!

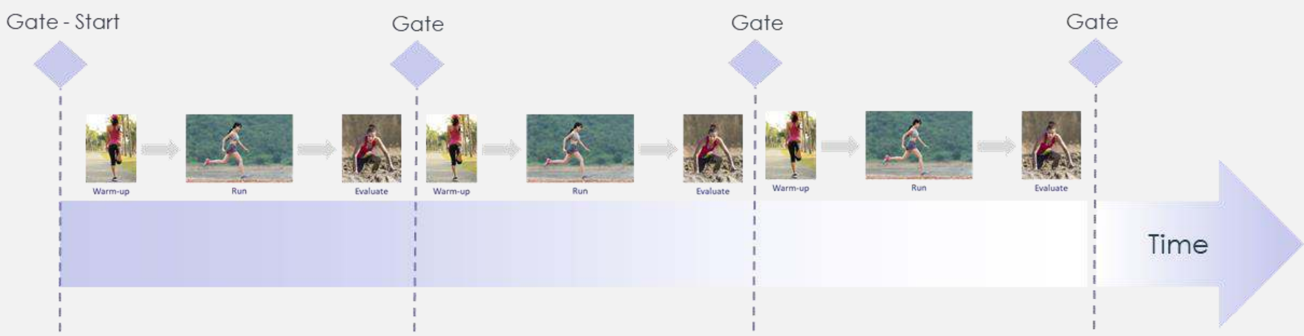
A word about warm-up, run and evaluate activities

The "warm-up" activity refers to the type of tasks that will reduce the risk of the engineering efforts that will be done afterwards. It is one of three types of activities that will be presented in this article:



Using the sports analogy, you need to prepare your body (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). In a long-term effort, such as preparing yourself for a competition, you perform these activities several times.

In engineering, an equivalent of such long-term efforts is the engineering effort that is done between project's major gates. Gates can be associated to major releases of the product or system, major technical or management milestones, or regular time slots.



Depending on the engineering physical level on which we are working (either the product as a whole, a subsystem of the product or a software component), time between gates may vary (1 year, 3 months or 3 weeks). The time allocated to warm-up, run and evaluate activities varies accordingly.

For instance:

- Systems teams will deliver major releases of the product every 18 months, they define gates every 3 months and at each gate they produce increments on the system engineering data (specifications, tests…) and the system itself (e.g. prototypes demonstrating value to the stakeholders)
- Software teams can release new versions every 3 months, so they define gates every 3 weeks (called sprints) and they produce increments of the working software

But these are only indications: the effort and time length allocated to the warm-up activity between two gates is defined by the team.

In any case, warm-up is about:

- Capture, selection and prioritization of the scope of work to be done so that it meets the objectives of the next gate, providing the expected value to the stakeholders
- Estimation of efforts required to do it
- Definition of the schedule to do it

Last but not least: warm-up, run and evaluate activities are not necessarily sequential, they can and are often executed in parallel: e.g. some key members of the team can “warm-up” by defining the scope, while others can “run” and pay technical debt that needs to be done at that moment.

The PythaDrone team is at the very early stages of the project, hence it performs an overall, product-level warm-up with the following objectives:

- Define the engineering strategy
- Elaborate the first vision of the product to be developed.

The team knows that, depending on which stage of the project they are in and on which engineering level they are working, the warm-up tasks and objectives will differ. They hope that in the future they will rely on the strategy and the organization put in place to focus on backlog management, for instance.

Define the engineering strategy

The team works on both the PythaDrone Product Roadmap, and on a plan describing how the systems engineering effort will be managed and conducted – this document is called Systems Engineering Management Plan – SEMP in the Pythagoras engineering process. The team defines:

- The engineering life-cycle and the synchronization mechanisms at each engineering physical level
- The product roadmap, including the major milestones and releases and the corresponding integration, verification and validation strategies
- The engineering organization to put in place
- The model-based engineering strategy, including the engineering practices and tools that will be used and how they will be tailored if necessary

Regarding the life cycle and the roadmap, the project can rely on previous training and experiences on implementing agility, and on a high-level Master Schedule.

What does "agility" mean here?

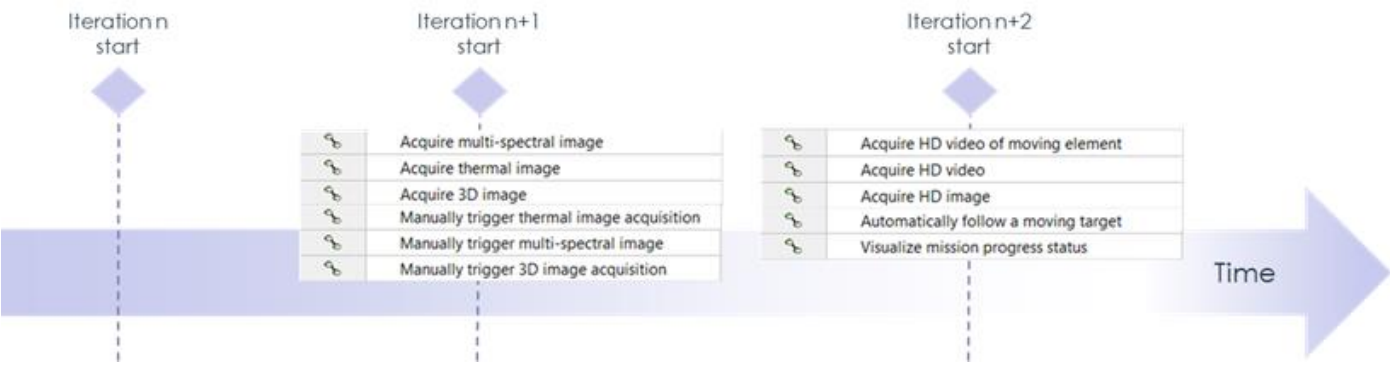
Agility in systems engineering refers here to an engineering effort in which teams are able to adapt to new circumstances (e.g., changes to or new stakeholders needs, technological innovations, findings in the development process ...) while meeting the customer expectations in terms of schedule, quality and cost. The ability to achieve agility must involve not only the systems engineering process but also the characteristics of the system under development, and the properties of the enterprise(s) that carry on the development effort.

Agility can be achieved through *increments* and *iterations*. An increment is a subset of the system (or of the systems engineering artifacts) that is delivered at the end of a time box. An increment is the addition of value to the system stakeholders built on top of an existing baseline. This value can be knowledge, risk reduction, new features, enhanced performance, etc. Increments may be divided into finer increments. Iterations are usually fixed-length time boxes in which engineering teams create value for the stakeholders by producing an increment. Iterations can also be divided into shorter ones. A backlog is an area where the upcoming needs to be fulfilled by the system are held so that it provides value to the stakeholders.

Gates consist of decisions, control reviews, milestones or system releases. These gates are key events that represent an expected progress in the systems engineering effort. Gates are hence used to assess the quality, costs and delays of the systems engineering effort, to synchronize teams work, and to feed the evaluation of the risk and opportunities of pursuing next systems engineering activities. A gate represents a point on time on which value is obtained and provided to the stakeholders if applicable; this value can take different forms: a consolidated conceptual design, a set of studies required for certification purposes, a simulation of the system, and the deployment of a subset of system capabilities, among others.

A *Master Schedule* provides a preliminary definition of the gates and of the foreseen increments of the system that will lead to the satisfaction of the stakeholders' expectations. The systems engineering effort performed between two gates leads to a *macro increment* of the system, and that these macro-level increments are possibly themselves divided into finer increments to address the required agility and reactiveness to changes.

Let's take a look to an extract of the resulting PythaDrone roadmap:

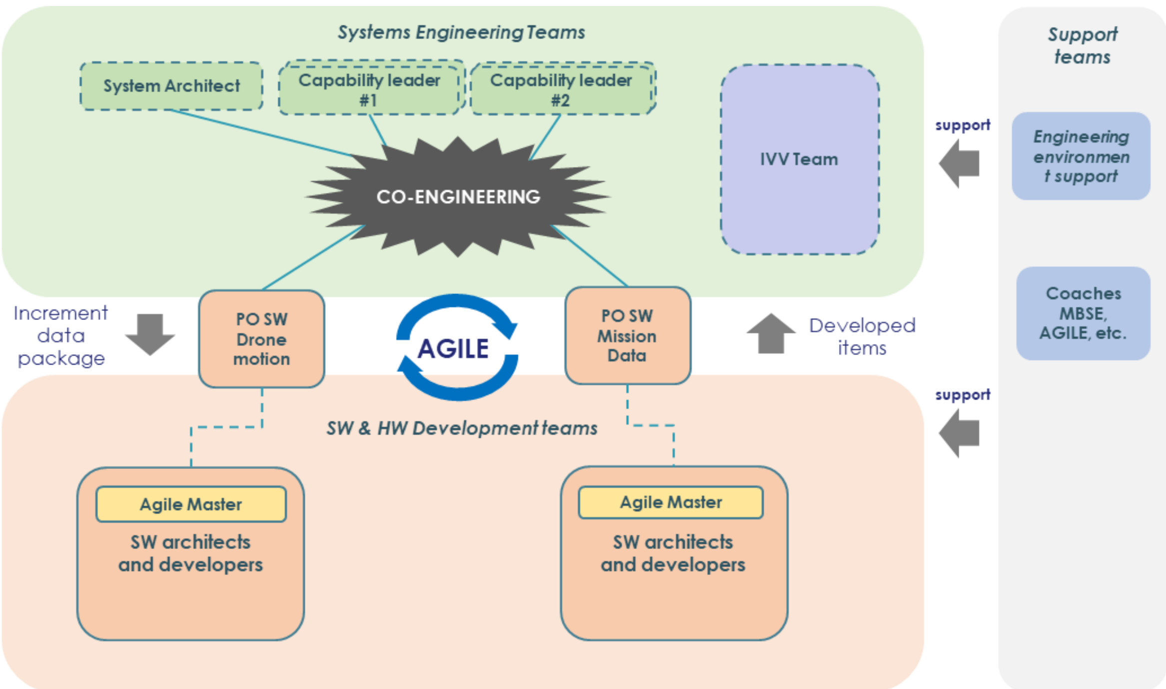


Iteration n+1 is foreseen to be focused on delivering features such as "acquire multi-spectral image" and "acquire thermal image". Next iteration would be focused on features such as "acquire HD video" and "visualize mission progress status". Choice of features to be developed at each iteration follows their interdependencies, which come from an initial architecture design activity (see Chapter: Elaborate the first vision of product on page 16).

The features for each iteration are defined to a certain extent. Starting from Gate TBD (foreseen 18 months from now) it is useless in defining the content of iterations: the team rather focuses on defining the expected delivery and product maturity at those stages. Similarly, some features already identified are not allocated to any iteration, they are added to the backlog of the product, waiting to be allocated in future warm-ups.

This is only the first vision of the roadmap: the PythaDrone team knows that it will adapt the roadmap to take advantage of opportunities or to mitigate development risks, that the architecture design will continue to evolve and will impact the roadmap as well, and that the feedback from early-adopter customers will also impact the content of the features and their delivery. This is part of what agility means! Or as D. Eisenhower stated: "In preparing for battles, I have always found that plans are useless, but planning is indispensable".

In order to manage the agile process, PythaDrone has put in place a customer-oriented organization. Each major functionality (aka capability) of the product is assigned a leader who coordinates the collaborative engineering between architects, IVV practitioners and software teams through iterations. The project has collectively decided that the length of the iterations at the product engineering level is to be 12 weeks.



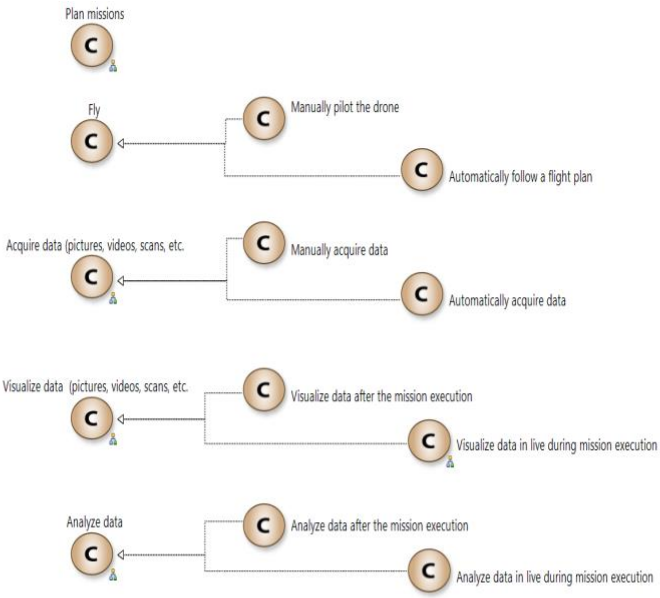
The PythaDrone team also defined how Pythagoras standard engineering practices and tools will be tailored for the project:

- The engineering data model: e.g. what are the relationships between textual requirements and model elements, how are contractual documents produced, how are the IVV artifacts related to textual requirements and models.
- The articulation between engineering teams: e.g. what is a “contract” between engineering teams made of, what are the outputs from/inputs to each team, what is the development pace (length of iterations at each engineering physical level).
- The model-based engineering strategy: what is the purpose of each model view? How will the views be structured? Are there existing building blocks to assemble? E.g. the Architecture team will (slightly) adapt the Arcadia MBSE method to design the product architecture.
- The engineering tools and how they will be configured: e.g. the Architecture team (of 8 people) will use Capella MBSE tool and will access and edit the model simultaneously to accelerate the design pace.

Elaborate the first vision of the product

In parallel, the Architecture team defines the main capabilities of the product, encompassing all phases of the product life cycle, and focused on the product operations, from preparation to post-mission data analysis. When doing this they can count on the contributions from marketing and business teams, and on software team leaders to address some of the technical risks identified at this point.

The figure below presents an extract of the capabilities of the PythaDrone product, in the form of a Capability diagram in Capella tool. The capabilities at the left side are some of the core capabilities of the product. They are specialized by the capabilities shown at the right side of the diagram.



Each capability is analyzed and different usage contexts of the capability are considered. These capabilities are the basis of the features that will be specified, developed and delivered in the future iterations according to the roadmap (see Define the engineering strategy, page 13).

In order to develop these capabilities, the Architecture team could rely on an operational analysis performed with marketing and business teams, in which the needs of the future customers were identified, analyzed and prioritized based on the value expected by the customers. The capabilities of the product are those that provide the value expected by the customers.

How to get better at warming-up?

We often see athletes who warm up for only a few minutes and still achieve high performance without injury. They get there by warming-up, running and evaluating their performance regularly, and maintaining an excellent condition.

For engineering teams, there are ways to maintain a "good shape" as well, and to warm-up very quickly and still execute engineering tasks safely:

- Products and building blocks that have previously-developed and maintained reference architectures are more effective in defining how the current state satisfies future expectations and in defining the gaps to be filled. Reference architectures capture the essence of the architecture of a product family and provide guidance to tailor it.
- Agile reference architectures are even more effective. Here we refer to the capacity of the product architecture to adapt to new circumstances. Pattern-based architectures with loosely-coupled components and well-defined and standardized interfaces are considered as agile ones.

- Projects that have put in place Product Line Engineering (PLE) organizations, techniques and tools, can identify rapidly the building blocks that can be reused for a new configuration, and hence can compress their delivery roadmaps.
- Companies that regularly perform and maintain operational analysis of their customers and the operational contexts can react faster to new expectations and to changes in the markets, and can rapidly launch new projects and products.

Next Episodes

Episode 2 will focus on a future iteration of the product roadmap and on the "Run" activities of the Architecture team. This Episode will present the model-based architecture concepts that are used by the Architecture team when specifying the content of the increments.

Episode 3, the last Episode will focus on an even later iteration and on both the "Run" and "Evaluate" activities of Software and IVV teams. This Episode will address the cases where an evolution is requested by the marketing teams, affecting the architecture, software and IVV engineering artifacts.

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.

A Fresh Look at Design Thinking in the Light of Proven Systems Engineering Principles

By John Fitch (PPI Principal Consultant and Course Presenter)

The set of principles and disciplines that we know as systems engineering was birthed in the telecommunications sector and matured in the second half of the 20th century to address the novelty, complexity and scale of defense and aerospace systems. Systems engineering is now practiced in virtually every definable sector; however, the driving force behind many of today's most innovative products, particularly in the consumer space, are product development methodologies that arose in relative isolation from (and in some cases rejection of) SE disciplines. Design Thinking is one such methodology.

Design Thinking isn't a singular universally-defined and standardized process; rather it has evolved into a movement with many champions who share some common principles. One of the early champions, Nigel Cross, of the United Kingdom's Open University, summarized a core concept of a "designerly" way of solving problems [1]:

"A central feature of design activity, then, is its reliance on generating fairly quickly a satisfactory solution, rather than on any prolonged analysis of the problem. In [Herbert] Simon's inelegant term, it is a process of 'satisficing' rather than optimising; producing any one of what might well be a large range of satisfactory solutions rather than attempting to generate the one hypothetically-optimum solution."

Cross identified five aspects of what he called "designerly ways of knowing":

- Designers tackle 'ill-defined' problems.
- Their mode of problem-solving is 'solution-focused'.
- Their mode of thinking is 'constructive'.
- They use 'codes' that translate abstract requirements into concrete objects.
- They use these codes to both 'read' and 'write' in 'object languages'.

Concerning "codes", Cross elaborates: "The designer learns to think in this sketch-like form, in which the abstract patterns of user requirements are turned into the concrete patterns of an actual object. It is like learning an artificial 'language', a kind of code which transforms 'thoughts' into 'words'". The modern MBSE practitioner can justifiably substitute 'models' for 'codes' in this context.

Tim Brown, CEO of IDEO and leading proponent of Design Thinking [2], defines it as "a methodology that imbues the full spectrum of innovation activities with a human-centered design ethos. By this I mean that innovation is powered by a thorough understanding, through direct observation, of what people want and need in their lives and what they like or dislike about the way particular products are made, packaged, marketed, sold and supported." Brown claims (perhaps a bit creatively) that Design Thinking is the "lineal descendant" of Thomas Edison's Menlo Park R&D laboratory as a "discipline that uses the designer's sensibility and methods to match people's needs with what is technologically feasible and what a viable business strategy can convert into customer value and market opportunity. Like Edison's painstaking innovation process, it often entails a great deal of perspiration."

While Design Thinking is a methodology, the community places high value on the characteristics of a designer. Brown lists five characteristics of the designer persona:

- Empathy
- Integrative thinking
- Optimism
- Experimentalism
- Collaboration.

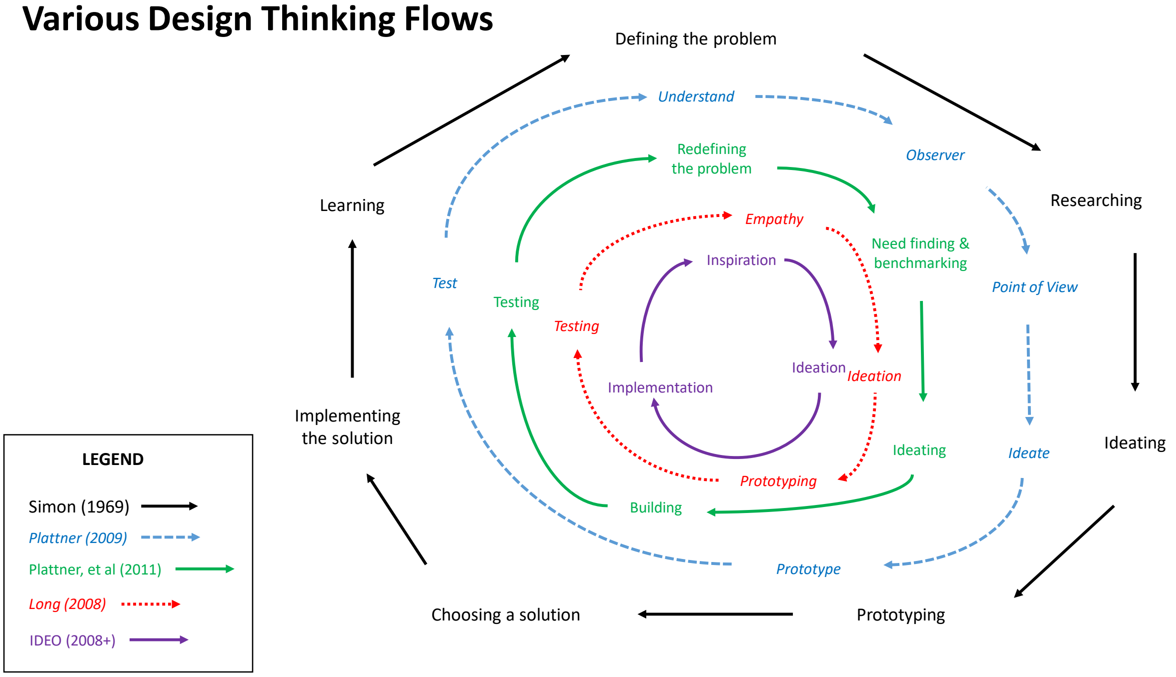
Design Thinking authors present their methodologies as a flow, but take pains to emphasize the cyclical, continuous learning and continuous improvement aspects of their processes. Discoveries from experimentation are expected to occur frequently; when they occur feedback to earlier process steps triggers another design iteration that incorporates the lessons learned.

In 1969, Herbert Simon proposed a seven stage process for product design that closed the loop:

- Defining the problem
- Researching
- Ideating
- Prototyping
- Choosing a solution
- Implementing the solution
- Learning.

As shown in the figure below, later depictions have tended toward simplifying the process, while still retaining the fundamental principle of cycles of learning [2, 3, 4, 5]. This figure also highlights the fact that there is no Design Thinking process decomposition or flow that is universally-accepted as orthodoxy. Many other variants exist; apologies to the reader if your preferred Design Thinking flow has been omitted from this brief treatise!

Various Design Thinking Flows



A few Design Thinking principles may be gleaned from these process flow examples:

- Design Thinking places high value on understanding the problem, not just intellectually or as explicit formal models, but also by gaining deep empathy for the end user and their experience to the point that this empathy inspires the designer to create life-changing solutions.
- Ideation, the synthesis and conceptualization of novel solutions, is an essential element of the process. Redefining the problem to open up more solution options is also a recognized innovation technique.
- Prototyping of solutions, i.e. making them tangible to facilitate user design feedback, is the preferred method of evaluating solution ideas.
- The testing of prototype solutions is focused more on continuous learning, rather than the formal verification of satisfaction of requirements that leads to final user acceptance.

At this point, the experienced engineer who has mastered contemporary systems engineering disciplines may be tempted to say that Design Thinking is nothing new; not much more than marketing hype:

- I care about the end user (indeed, all stakeholders) and seek to understand their use cases at the depth needed to meet their needs and values and even delight them by creating additional value.
- Creating innovative solutions to problems is what I do daily; I have a toolkit of ideation techniques that I use as appropriate.
- Prototyping? I frequently create sketches, mock-ups, models, simulations and partial solutions so that I can gain user feedback and assess the potential performance of a solution concept.
- Of course, we evaluate the prototypes, but someone has to eventually make the decision that a design build is ready to deploy to the end users.

In order to discern the difference between contemporary SE principles and practices and those promoted as Design Thinking, we will have to look deeper. To do so, let's compare the precepts of Design Thinking with the fifteen principles that Project Performance International promotes as the foundation for the effective engineering of systems.

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

In Design Thinking, designers seek to understand user needs at a deep and even emotional level, sufficient to create empathy for what the users feel about the product (not just think about the product or do with it). A diverse range of techniques is recommended, with a strong preference for direct user interaction over modeling-at-a-distance. However, the emphasis on the user is primarily qualitative; there is little discussion in Design Thinking literature on quantifying requirements, MOEs and goals. Indeed, some authors assert that Design Thinking is entirely needs-driven, not requirements-driven. These authors make no accommodation for formally capturing a set of system requirements that define the characteristics of an acceptable solution or for defining a value model against which value delivery can be maximized in conducting trade-offs.

Design Thinking focuses on the human-facing aspects of the problem; system-to-system or subsystem-to-subsystem interactions that enable the user experience are not explicitly addressed, nor are the interests of other stakeholders.

Design Thinking quotes:

"By learning to observe human behaviors and needs in the context of real life, DT participants discover human-centered questions and problems worth trying to solve. Better yet, it does so within a remarkably empathetic process that puts the experience of human beings at the center of the equation." [4]

"A user-centric culture is the core of any of the design thinking models and should be maintained in every stage in the design process. The user-centric approach ensures that the final product or design has one ultimate goal; that is to say, the goal of solving the user problem in a desirable and usable way." [6]

"Designers are required to conduct deep user research methods such as the Journey Map in order to collect the information that contributes to understanding the user needs" [6]

"Empathy means that designers should put themselves in the consumer shoes. They are not only required to think about the consumer needs, but also their feeling about the product or competitive products. Collecting information about the consumers' feelings about the product contribute to building the so-called emotional design; the design that address users' emotions in addition to their physical needs." [6]

During the design process, storytelling is used to understand the underlying problems that consumers face when using a product or a service, and then use this knowledge in reflective practice to formulate the solution which is tested by the consumers. … In the problem framing stage, the storytelling can be used in this explorative initiative. It helps the design team to engage with the user to define the problem using a qualitative data from the consumers. [7]

"Problem exploration effort often reveals underlying and unarticulated customer needs. At times, it becomes evident that the original customer problem was a symptom of a bigger problem. If the engineering design focused on what the customer originally articulated (a symptom), the larger problem (and the real need) would remain only partially addressed, if at all. For this reason and others, design thinking is inherently need-driven and **not requirements-driven**" [8]

"Numerous facets of the design thinking process facilitate the problem re-definition process. … taking a holistic perspective; embracing diversity of thought through an interdisciplinary approach; using qualitative research techniques; having a user-centered approach; and using sacrificial concepts. Over-arching is the intrinsically iterative nature of the entire design thinking process where exploration and failure are embraced as necessary aspects of learning and improving the designer's understanding of the problem to be solved. … This iterative problem definition approach… enables cognitive biases and ingrained assumptions of the design team to be challenged and updated." [8]

"The design research conducted in this study primarily used qualitative methods such as observation, ethnography, and interviewing, the latter of which formed the focus of the efforts. … To engineers trained in quantitative methods, qualitative methods can seem to contradict intuition. Qualitative methods can provide a useful augmentation to quantitative studies to more richly illuminate processes, cultures, relationships, and motivations that impact a system's design. As engineering design is a social, organizational, cultural, political, and a mechanical activity, only a diversity of research methods can help tap latent and unarticulated customer needs and enable improvements beyond current ideas and inherent biases." [8]

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

Design Thinking is generally devoid of the concept of requirements. In Design Thinking, future user needs are considered in current design iterations. No effort is made to limit the focus to feasible requirements or solutions; Design Thinking uses far-fetched ideas (aka sacrificial concepts) to elicit feedback and create learning.

Design Thinking quotes:

"Products should be designed for the future as well as the present as consumers are expected to use the product for years after buying it. Designing with the future technology in mind helps designers to address not only today's problems but also the future problems that may face consumers." [6]

"The accuracy or even feasibility of the concepts are not important at this stage, as most will never survive beyond a description, a sketch, and perhaps a crude physical prototype. A good sacrificial concept draws from one or more key elements of a design challenge and offers creative solutions that may be considered far-fetched or even impractical; designers are given a reprieve from technical criticism here. Each addresses one or more user needs based on preliminary hypotheses about what the user may consider to be useful or desirable." [8]

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

Design thinking makes no explicit distinction between requirements and goals, focusing on user needs. Because the focus is on deep and continuous learning about the users' needs, there is no point where an explicit set of requirements is formalized as the "must-haves" for a particular product release, against which a set of goals that represent additional stakeholder value may be considered.

Consequently, Design Thinking makes no mention of formal requirements analysis techniques that translate user needs into a set of system requirements, or that distinguish a system requirements specification from a stakeholder value model that can be used to inform maximization of value delivery.

Design Thinking quote:

"Problem exploration effort often reveals underlying and unarticulated customer needs. At times, it becomes evident that the original customer problem was a symptom of a bigger problem. If the engineering design focused on what the customer originally articulated (a symptom), the larger problem (and the real need) would remain only partially addressed, if at all. For this reason and others, design thinking is inherently need-driven and **not requirements-driven**" [8]

PPI Principle #4: Define system requirements, measures of effectiveness, goals and solutions having regard to the whole of the (remaining) life cycle of the system of interest.

Design Thinking's focus on continuous learning about user needs includes full lifecycle considerations, though these considerations don't appear to be a significant point of emphasis, nor are specific techniques recommended for gathering or differentiating the needs of test, manufacturing, deployment, support or disposal personnel from end users in the operational phase of the product life cycle.

Design Thinking quote:

"Designers should also think about how their products will be used during the consumer life in the context of other items and circumstances." [6]

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

Design Thinking processes are represented as continuous learning and improvement feedback loops; the need for distinction between a formal problem and solution statement is not highlighted, though may be inferred in some variants of the methodology.

For example, the Stanford University D.School's Design Thinking process translates the problem understanding into a point-of-view, a micro-theory used to develop solution concepts. This appears to be an extension of the problem definition, rather than an explicit solution physical architecture.

Beyond the initial conceptualization of solutions, Design Thinking blends the problem - solution models at each iteration in the form of sacrificial prototypes.

Design Thinking focuses on the user-facing problem description/solution and has no unique techniques for design problems that don't directly and holistically interact with users, i.e. Design Thinking isn't truly recursive in working at all levels of system context/definition. There is no consideration of how successively lower levels of subsystems will interact to produce the desired user experience.

Design Thinking quotes:

"The design thinking process model, however, seems to be only applicable to the entire problem; not to specific sub-problems" [5]

The acquired knowledge is then condensed into a sort of micro-theory about the problem or the user needs, the 'point of view' (POV) that is afterwards used to develop solution concepts in the 'ideation' step" [5]

"Design thinking suggests several sophisticated methods for synthesizing insights from the user research … Among these frameworks are 'Personas', '2-Axis Mappings', 'User Journeys', or 'Causal Maps'. They help to align the researched information in a qualitative way, in order to condense them into a so-called 'Point of View'—a kind of micro theory about the user needs, which determines the further direction of the process." [5]

"Stanford D.School design thinking process … includes the first three stages of Empathize, Define, and Ideate. Those three stages focus on investigating and defining the problem that needs to be addressed and how it will be solved in the final product or service." [6]

"Design is represented in a complete picture of the product or the situation as our consumers see the design as a whole rather than its parts. Therefore, designers are required to think about their design as a whole product rather than individual components." [6]

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

Design Thinking baselines the problem definition only after extensive primary and secondary research. There is no equivalent concept of a formal design description baseline. Based on feedback from users, Design Thinking suffers from a lack of Configuration Management (CM) tools, with proponents preferring manual methods to data capture, synthesis, but suffering potential loss of baseline integrity and control for large projects and distributed teams.

Design Thinking quotes:

"The problem is not defined until an extensive phase of user and secondary research has been conducted, and the ideas are then generated during the process." [5]

"This (DT) approach allowed for the possibility of redesigning the original problem statement to better meet the needs of the users." [8]

"Readers familiar with the design thinking process recognize the use of posters and sticky notes used to capture the data synthesis, analysis and subsequent ideation. These tools were effective for several goals: working quickly; working democratically; enabling quieter team members to share their thoughts and ideas; visualizing a very large number of ideas and notes simultaneously; and reorganizing, filtering, and updating analysis. However, the team was geographically dispersed and not able to keep the information gathered in one location for continuous and visual reference. Effectively maintaining the data on the posters and sticky notes long term was challenging." [8]

PPI Principle #7: Identify and develop descriptions of solutions alternatives (designs) that are both **feasible (i.e., can meet requirements) and potentially are the most effective. Put aside from further consideration, as potential solutions, all other alternatives (unless the assessment of that potential solution changes). Note: MOEs could include development cost, time to market or other measures unrelated to system capabilities.**

Design Thinking focuses on the user desirability of products, consistent with business viability and technology feasibility. Design Thinking focuses on innovation and leverages many classical ideation techniques to generate solutions.

No effort is made to limit Design Thinking's focus to feasible solutions; the use of far-fetched ideas as sacrificial prototypes to elicit feedback and create learning is recommended.

Beyond user feedback, there is no specific approach for parallel evaluation of multiple feasible solution concepts.

Design Thinking quotes:

"Ideation: The goal is not to find a perfect solution at this point. Instead, DT participants seek novel perspectives with a bias toward innovation. … DT relies on a creative process based on "building up" ideas (rather than the typical analytical process that looks to "break down" ideas)" [4]

"Design thinking makes extensive use of classical ideation techniques, borrowed from other creative disciplines, to generate ideas" [5]

"The accuracy or even feasibility of the concepts are not important at this stage, as most will never survive beyond a description, a sketch, and perhaps a crude physical prototype. A good sacrificial concept draws from one or more key elements of a design challenge and offers creative solutions that may be considered far-fetched or even impractical; designers are given a reprieve from technical criticism here. Each addresses one or more user needs based on preliminary hypotheses about what the user may consider to be useful or desirable." [8]

PPI Principle #8: Develop solution descriptions for enabling systems concurrently and in balance with the solution description for the system of interest. Note: an "enabling system" is a system which enables some phase of the life cycle of the system of interest. The internal design of an enabling system must be related to the internal design of the system of interest.

Design Thinking doesn't emphasize the concurrent engineering of enabling systems, e.g. manufacturing or support, though such systems could use Design Thinking to improve their usability.

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

Design Thinking doesn't focus on logical solution descriptions; it jumps to a prototype used to gain direct user feedback. But nothing prevents a visual or sacrificial "idea" prototype being a functional model and/or state model of a proposed solution concept. Design Thinking authors speak of an artificial world that may include logical solution descriptions associated with the physical elements.

Design Thinking quotes:

"What designers especially know about is the "artificial world"—the human-made world of artifacts. What they especially know how to do is the proposing of additions to and changes to the artificial world. Their knowledge, skills, and values lie in the techniques of the artificial. … So design knowledge is of and about the artificial world and how to contribute to the creation and maintenance of that world." [1]

"The selected idea is then visualized or built ('prototype') in order to test it and gather feedback from prospective users ('test')" [5]

PPI Principle #10: Be prepared to iterate in design to drive up overall effectiveness, but not at the expense of the requirements.

In Design Thinking, requirements are never fixed/frozen; every product is a prototype and an opportunity for learning and improvement. Iteration is a point of emphasis. Sacrificial concepts are iterated rapidly until a few evolve into feasible design concepts. Iteration may limit the solution scope to that which is feasible for a subset of the full user/stakeholder population originally intended.

Design Thinking quotes:

"Prototyping. Once participants identify a wide range of possible solutions, the next step is to rapidly mock up examples. To DT advocates, the idea is to help make an idea real, tangible, and accessible. Ultimately, DT has a natural bias toward action. The best way to approach this—as many designers will tell you—is to use a rapid prototyping process fueled by an attitude of "fail and fail fast," something ideally suited for learning in a complex and often messy 21st century world." [4]

"Prototyping is considered one of the most important stages in all the design thinking processes. This stage helps designers to visualize the ideas and build a better understanding of how this design might solve the user problem." [6]

"In the prototyping stage, storytelling plays an even more critical role as the consumers can use the solution and provide feedback to the design team. The team will be able to observe how the consumers use the prototype solution and build a realistic understanding of the suggested solutions."[7]

"Finally, the team benefitted from using 'sacrificial prototyping,' which is a method of rapidly prototyping draft concepts and ideas with the intent of enabling potential users to provide significant feedback early in the design process. This contrasts the more common approach of using expensive prototypes that focus on demonstrating technical feasibility. … The iterative process of feedback and refinement gradually advances the sacrificial concepts to what can be considered preliminary design concepts and then to final design concepts that resemble more traditional engineering design concepts." [8]

"It was not possible for the team to synthesize a design concept that would satisfy all … users, or even all the users that were interviewed. A smaller and manageable subset of the users was needed to make forward progress. … The users were selected because synthesis of the design research identified them as having the largest impact, being the newest field of research, being a most difficult problem, and possessing a need to stay ahead of the technology 'curve' to be competitive [8]

"The restless reinvention tends to build an iterative process. The process is based on prototyping solutions for old problems with new ways based on actions. Keeping in mind that nothing is perfect, this turns every product into a prototype that is a case for iteration and development." [9]

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

Design Thinking offers no decision-making or trade-off methods for effectiveness evaluation or risk/opportunity assessment beyond qualitative user feedback concerning prototypes. Qualitative methods that produce face-to-face user feedback are emphasized over quantitative analysis techniques.

Design Thinking quotes:

"The storytelling can be used in this part (Solution Framing) of the design process to build a two [way] communication with the consumer to validate the efficiency of the solution" [7]

"Qualitative methods can provide a useful augmentation to quantitative studies to more richly illuminate processes, cultures, relationships, and motivations that impact a system's design. As engineering design is a social, organizational, cultural, political, and a mechanical activity, only a diversity of research methods can help tap latent and unarticulated customer needs and enable improvements beyond current ideas and inherent biases." [8]

"During the study, a profuse amount (more than could be used) of invaluable information was obtained using qualitative methods. The inductive and highly interactive approach to data analysis provided vital insights to the team on the user needs, stakeholder concerns, organizational constraints and opportunities, and potential technical solutions. The data synthesis and analysis process took extensive face-to-face time. " [8]

PPI Principle #12: Subject to level of risk, independently verify work products (is the job being done right?)

Design Thinking accomplishes some level of independent verification by direct user testing of prototypes. Without a formal requirements baseline, these activities may be primarily validation of the prototype features against user needs. There is no mention of verification of other work products including subsystems.

Design Thinking quotes:

"Testing - Fail early to succeed sooner" … focus on extensive user testing in order to improve their respective concepts [5]

"design thinking process tends to ensure this approach (toward achieving customer satisfaction) is achieved through number of methods such as user testing, validation, and multiple iterations between different stages" [6]

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

Design Thinking seeks user feedback early to confirm that the right product/solution is being built. With the emphasis on rapid iteration with direct user engagement in prototypes, validation is critical to success. Building the wrong product is a cardinal sin.

Design Thinking quotes:

"Testing. DT deeply values testing all assumptions. Solutions need to work. And better yet, solutions need to work in the real world and have an observable positive impact on the human experience." [4]

"Prototypes only needed to be good enough to suggest possibilities and engage audiences." [4]

"gather user feedback in early stages of the process, in order not to waste lots of resources by building something that nobody wants" [5]

PPI Principle #14: The act of managing is needed to plan and implement the effective and efficient transformation of requirements and goals into solutions.

Design Thinking requires the act of managing as with any other type of project, but its iterative nature, less-tangible work products and increased level of multi-disciplinary collaboration may demand different management methods than other design approaches.

Design Thinking quotes:

"The holistic and inherently iterative approach of design thinking means progress is measured more in increased clarity and understanding of the problem and its potential solutions. While critical, these "products" were less tangible and measurable than traditional intermediary engineering deliverables that often include more measureable constructs such as early hardware components or versions of software. Managers may not understand or be able to effectively use traditional methods to evaluate, and thus reward, the quality or progress of this non-traditional approach." [8]

"The design thinking process draws a design team to work in a more collective or interdisciplinary manner. … In design thinking, the research, ideation, and synthesis steps all require the team to interact in an interdisciplinary manner where existing and new ideas are owned and iterated upon by the team." [8]

"More familiar with working in a divide-and-conquer style multidisciplinary approach, the interdisciplinary and constantly interactive nature of the data synthesis process required all team members to work differently" [8]

PPI Principle #15: Decide early on development strategy, between waterfall, incremental, evolutionary and spiral, based on ability to define good, stable requirements up front; risk due to technology; risk due to complexity; and other sources and levels of risk and opportunity.

Design Thinking almost exclusively uses an evolutionary product development strategy in which the learning from each cycle/iteration feeds the next cycle. Risk is mitigated by getting prototype solutions in front of end users early and often during development.

Design Thinking quotes:

"The process essentially comes down to a continuously evolving feedback loop with four elements: empathy, ideation, prototyping, and testing." [4]

"Step-wise and predictable linear processes are not always better." [8]

Final Thoughts

Effective engineering of systems, whether complex or simple or addressing simple, complex or "wicked" problems, requires adapting engineering processes, methods and tools to the job at hand. There is no one-size-fits-all set of engineering practices that is "right" for every project. Such tailoring should be driven by principles that have been proven, across diverse situations, to lead to the success of design projects.

Design Thinking provides value to the process of engineering solutions by its emphasis on techniques that create a deep understanding of the problem, empathy with users and frequent feedback from users to validate the designer's understanding of the problem and how well the evolving solution may satisfy the users' needs. The value provided by Design Thinking would appear to be greatest in a consumer product development context, and least for embedded systems.

Design Thinking covers much less scope that contemporary systems engineering practices, being relatively silent on anything but the user-facing aspects of design. Its dependence on qualitative user feedback for effectiveness evaluation and decision-making, verification and validation leaves significant gaps when designing systems for which performance is the primary differentiator or when designing subsystems that don't interact with human users. In addition, the concurrent engineering of enabling systems appears to be missing from Design Thinking processes.

Clearly, the Design Thinking community has done an outstanding job of marketing its innovation capabilities to a wide audience, from a platform of clear value of Design Thinking concepts, mostly in an assumed context of consumer product development. Despite some obvious gaps in Design Thinking, the contemporary systems engineering community could consider carefully the reasons why its disciplines have been bypassed in some companies as the primary engines of innovation and adapt its practices accordingly. The Design Thinking emphasis on empathy, feeling and emotion of users in particular warrants attention for some applications of systems engineering, especially development of consumer products.

References:

[1] Cross, N., (1982), "'Designerly Ways of Knowing: Design Discipline versus Design Science". Design Studies, Vol. 3 No. 4 pp. 221-227.

[2] Brown, T., (2008), "Design Thinking". Harvard Business Review, June, 2008, pp. 84-92.

[3] Greene, M., Gonzalez, R., Papalambros, P., and McGowan, A. (2017), "Design Thinking vs. Systems Thinking for Engineering Design: What's the Difference?" 21st International Conference on Engineering Design, Vancouver, B.C., Canada, August 21-25, 2017.

[4] Long, C. (2012), "Teach Your Students to Fail Better with Design Thinking." Learning & Leading with Technology, Vol. 9, No. 5, pp. 16-20.

[5] Mueller, R., and Thoring, K. (2012), "Design Thinking vs Lean Startup: A Comparison of Two User-Driven Innovation Strategies." International Design Management Research Conference, Boston, MA, USA, August 8-9, 2012.

[6] Elmansy, R., "How Design Thinking Approaches Problems." https://www.designorate.com/how-design-thinking-approaches-problems/

[7] Elmansy, R., "The Role of Storytelling in the Design Process." https://www.designorate.com/the-role-of-storytelling-in-the-design-process/

[8] McGowan, AM, Bakula, C., and Castner, R. (2017), "Lessons Learned from Applying Design Thinking in a NASA Rapid Design Study in Aeronautics." Proceedings of AIAA SciTech 2017, Grapevine, FL, Jan 9-13.

[9] Elmansy, R., "IBM Design Thinking Model: A Shift Toward Big Enterprises." https://www.designorate.com/ibm-design-thinking/

UPCOMING SCHEDULED TRAINING VIA PPI LIVE-ONLINE™

For more information on content, costs, and delivery, please visit the PPI Live-Online™ website

Course Details	Course Dates
Systems Engineering North America UTC -4:00 (EDT 8:00) PPI Live-Online	25 Oct - 29 Oct 2021
Systems Engineering South America UTC -3:00 (BRT 9:00) PPI Live-Online *(Only available in South America)*	25 Oct - 29 Oct 2021
Systems Engineering Europe UTC +1:00 (CET 9:00) PPI Live-Online	01 Nov - 05 Nov 2021
Systems Engineering United Kingdom UTC +0:00 (GMT 8:00) PPI Live-Online	01 Nov - 05 Nov 2021
Requirements Analysis and Specification Writing North America UTC -4:00 (EDT 8:00) PPI Live-Online	04 Oct - 08 Oct 2021
Requirements Analysis and Specification Writing South America UTC -3:00 (BRT 9:00) PPI Live-Online *(Only available in South America)*	04 Oct - 08 Oct 2021
Requirements Analysis and Specification Writing Turkey UTC +3:00 (TRT 8:00) PPI Live-Online	04 Oct - 08 Oct 2021
Systems Engineering Management Europe UTC +2:00 (CEST 9:00) PPI Live-Online	18 Oct - 22 Oct 2021
Systems Engineering Management United Kingdom UTC +1:00 (BST 8:00) PPI Live-Online	18 Oct - 22 Oct 2021
Systems Engineering Management South Africa UTC +2:00 (SAST 9:00) PPI Live-Online *(Only available in South Africa)*	18 Oct - 22 Oct 2021
Architectural Design Asia UTC +8:00 (SGT 5:00) PPI Live-Online	25 Oct - 29 Oct 2021
Architectural Design Oceania UTC +11:00 (AEDT 8:00) PPI Live-Online	25 Oct - 29 Oct 2021
Architectural Design North America UTC -5:00 (EST 8:00) PPI Live-Online	06 Dec - 10 Dec 2021
Requirements, OCD & CONOPS in Military Capability Development Turkey UTC +3:00 (TRT 8:00) PPI Live-Online	08 Nov - 12 Nov 2021
Requirements, OCD & CONOPS in Military Capability Development Saudi Arabia UTC +3:00 (AST 8:00) PPI Live-Online	08 Nov - 12 Nov 2021
Requirements, OCD & CONOPS in Military Capability Development North America UTC -5:00 (EST 8:00) PPI Live-Online	15 Nov - 19 Nov 2021

A Comparison of the Usability of Five MBSE Language/Tool Combinations

By Wioletta Kowalczyk (BSc BS MS, PPI SyEN Contributor)

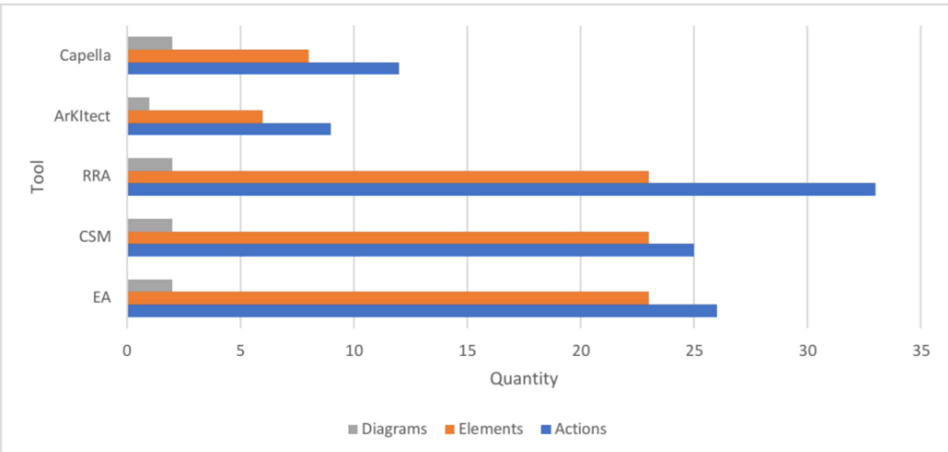
Engineers and other problem solvers use logical modeling, often functional modeling, extensively in their work. The form of such modeling may be as basic as a MS Project Gantt Chart, or as sophisticated as an executable behavior model of a complex, technology-intensive system. Many different tools are in use – software and cloud services. Tools may use different languages, sometimes proprietory (specific to the tool), sometimes public domain like SysML 1.x – so, what is the ease of use these proprietary Model Based Systems Engineering (MBSE) languages versus the well known SysML 1.x MBSE language?

Ease of use may be quantified by using a measure such as the number of steps or actions it takes to decompose a function in an item-flow decomposition, the number of information elements involved, and the number of diagrams. These terms are defined below. In this article, the parent function has two sub-functions. We have a function say F1, two sub-functions (F1.1 and F1.2) and 3 item flows (F1_in, F1.1_out/F1.2_in, and F1_out) - 6 elements in the resulting model.

This article based on [2] provides a short comparison of the non-SysML tools: ArKItect and Capella versus SysML1.5 [1] tools: Enterprise Architect (EA), Cameos System Modeler (CSM) and Rational Rhapsody Architect (RRA).

In order to to understand the comparison, the following key definitions are presented:

- **action:** (not necessarily atomic, i.e. could involve multiple user steps) user action resulting in the creation, modification or suppression of an element or a diagram.
- **element:** (model) element (including link) that is created, modified or deleted by a user action in a diagram.
- **diagram:** (model) view that displays a set of elements and that is created, modified or deleted by a user action.



It is observed that the various SysML modeling tools in their basic configurations needed similar numbers of required actions, elements and diagrams to model the decomposition of a function. This similarity appears to be mostly due to the reliance of the tool on the SysML 1.x set of standards.

The non-SysML modeling tools are more efficient in modeling the decomposition of a function regarding the number of required elements (and therefore the number of actions), because they handle the instantiation and inheritance mechanisms between the elements in the meta-model, instead of the user needing to deal with them explicitly.

Another factor in the ease of use of different tools and languages is the user's familiarity with the language. Nor is ease of use the only Measure of Effectiveness (MOE) of a language. Other MOEs include logical intergity, expressiveness , intuitiveness, and absence of redundancy in means of expression, i.e, avoidance of having multiple ways of expressing the same information. [3]

[1] ISO/IEC 19514:2017: "Information technology — Object management group systems modeling language (OMG SysML)

[2] Regis Casteran, https://medium.com/seatwork/functions-in-systems-model-1f96f96a818

[3] Robert Halligan, PPI private communication

"

As to methods, there may be a million and then some, but principles are few. Those who grasp principles can successfully select their own methods.

HARRINGTON EMERSON

An Overview of the Requirements Interchange Format (ReqIF)

By Alwyn Smit (PPI Principal Consultant and Course Presenter)

The automotive industry introduced requirements management around 1999. Not surprisingly, with this established discipline in place, manufacturers and suppliers strive for collaborative requirements management where requirements management does not stop at company boundaries. However, two companies in the manufacturing industry can rarely work on the same requirements repository for technical and organizational reasons, and sometimes do not work with the same requirements authoring tools. Therefore, a generic, non-proprietary format for requirements information was needed to satisfy the urgent industry need for exchanging requirement information between different companies without losing the advantage of requirements management at the organizations' boundaries.

The Requirements Interchange Format (ReqIF) defines such an open, non-proprietary exchange format. Companies can exchange requirement information by transferring XML documents that comply with the ReqIF format.

Some of the benefits of using the ReqIF standard are:

- The benefits of applying requirements management methods across company boundaries improve the collaboration between partner companies.
- The partner companies do not have to use the same requirements authoring tool, and suppliers do not need to have multiple requirements authoring tools to fulfil the needs of their customers with regards to compatibility.
- Requirement information can be exchanged within a company, even if different authors use different tools to author requirements.

The ReqIF standard essentially defines a standardized format to generate a requirements document with a hierarchical structure that uses formatted text (including references to binary files) to express: uniquely identified requirements together with their associated attributes, established relationships between requirements, groups of relations, and user access control. These also happen to be the underlying features of most requirements authoring tools.

SYSTEMS ENGINEERING RESOURCES

IIBA's KnowledgeHub

The International Institute of Business Analysis (IIBA) has introduced for IIBA members the IIBA's KnowledgeHub, a new online source actionable, how-to business analysis content, knowledge, and tools. The KnowledgeHub provides members with online, searchable access to the BABOK® Guide and BA community-driven content, such as relevant "how do I" scenarios, templates, videos, checklists, and infographics that can be applied to business analysis practices. Scenarios topics range from capturing key requirements, to understanding business processes, to identifying needs, and describing user system interactions.
More information

Featured Organization: Digital Metrology Standards Consortium (DMSC™)

The Digital Metrology Standards Consortium (DMSC™ Inc.) aims to identify needed standards in the field of digital metrology, and to promote, foster, and encourage the development and interoperability of these standards, along with related and supporting standards that will benefit industry as a whole. Metrology is the scientific study of measurement. Metrology fosters a common understanding of units of measurement. The Dimensional Measuring Interface Standard (DMIS) and the Quality Information Framework (QIF) are two such standards for which the DSMC has responsibility to continue development, maintenance and support, as well as coordination and harmonization with other related standards efforts. The DMSC™ is an ANSI Accredited Standards Developing Organization, as well as an A-Liaison to ISO.

QIF, especially relevant to systems engineering tools information exchange, is a Unified XML Framework standard for CAD quality measurement systems. QIF was embraced by the International Standards Organization (ISO) and released as ISO 23952:2020 in August 2020.

The DSMC invites participation within the consortium of other standards groups and activities that seek to resolve the technology and other issues of automated (digital) metrology. Membership of the DMSC is open to companies and other organizations with an interest in the field.

For more information on the DMSC see: https://qifstandards.org/about-dmsc/

The INCOSE International Symposium IS2021 – Something for Everyone!

The International Council on Systems Engineering (INCOSE) has published a Book of Abstracts of Plenaries, Papers, Key Reserve Papers, Presentations, Panels, Tutorials and Invited Content associated with its 30th International Symposium, conducted online over 17-22 July, 2021. Just the list of titles, reproduced below, provides a snapshot of the scope of systems engineering as perceived by the INCOSE community, as well as a path to topics of individual interest.

The full Book of Abstracts is downloadable at https://www.incose.org/symp2021/symposium/event-schedule.

Keynote - Plenaries

K1: Countering Digital Authoritarianism

K2: The role of architecture in achieving Society

K3: How systems engineering made solar cars a reality

Presidents Panel: Accelerating through Adversity – Back to the Future!

Papers

002: Assessing a supplier to the offshore oil and gas industry following a worldwide pandemic

004: Developing domain-specific AI-based tools to boost cross-enterprise knowledge reuse and improve …

006: Enterprise Architecture Process Guide for the Unified Architecture Framework (UAF)

007: Aspect-Oriented Architecting Using Architecture Frameworks

008: You Don't Save Money by Doing Less Testing – You Save Money by Doing More of the Right Testing

009: Why Systems Engineers May Have an Edge When It Comes to Personal Resilience

011: I nnovative Approaches to Superset Asset Templates using Feature-Based Product Line Engineering

013: A Guide for Systems Engineers to Finding Your Role in 21st-Century Software-Dominant Organization

016: Evaluation of Requirements Management Processes Utilizing System Modeling Language (SysML)

018: Experience in Designing for Cyber Resiliency in Embedded DoD Systems

020: Formulas and Guidelines for Deriving Functional System Requirements from a Systems Engineering …

021: How Missile Engineering is Taking Product Line Engineering to the Extreme at Raytheon

024: Employing a Model Based Conceptual Design Approach to Design for Resilience

027: Putting the Social in Systems Engineering: An Overview and Conceptual Development

028: The value of trade-off studies for student projects

029: Analyzing Standard Operating Procedures Using Model-based System Engineering Diagrams

030: The risk maturity model: a new tool for improved risk management and feedback

031: Feature-based Product Line Engineering: An Essential Ingredient in Agile Acquisition

032: Social Science Solutions for the Systems Engineer: What's Needed

034: Challenges in Detecting Emergent Behavior in System Testing

037: Unlocking the power of big data within the early design phase of the new product development …

038: Product portfolio mapping used to structure a mature sub-system with large variation - A case study

040: Conceptual modeling of energy storage systems

041: Predicting failure events from crowd-derived inputs: schedule slips and missed requirements

042: From UAF to SysML: Transitioning from System of Systems to Systems Architecture

043: Agility in the Future of Systems Engineering (FuSE) - A Roadmap of Foundational Concepts

045: Security as a Functional Requirement in the Future of Systems Engineering

046: Network Rails Systems Integration for Delivery (SI4D) Framework

047: Insights for Systems Security Engineering from Multilayer Network Models

049: Security in the Future of Systems Engineering (FuSE), a Roadmap of Foundation Concepts

051: Developing a Model Based Systems Engineering Architecture for Defense Wearable Technology

052: Applying Systems Engineering framework for architecting a Smart Parking System within a Smart

053: Integrating Safety Analysis into Model-Based Systems Engineering for Aircraft Systems: A Literature …

054: A value-driven, integrated approach to Model-Based Product Line Engineering

055: Dealing with COVID-19 Pandemic in Complex Societal System for Resilience Study: A Systems Approach

056: From Brownfield to Greenfield Development – Understanding and Managing the Transition

057: Application of natural language processing for systematic requirement management in model-based systems engineering

058: Conceptual Modelling of Seasonal Energy Storage Technologies for Residential Heating in a Dutch

062: A Framework for Identifying and Managing New Operational Requirements during Naval Vessel B

064: How can simplified requirements affect project efficiency – A case study in oil and gas

065: Application of T-shaped engineering skills in complex multidisciplinary projects

067: Idea Development Method, Applying Systems Design Thinking in a Very Small Entity

074: Enhancing Enterprise Architecture with Resilience Perspective

075: Application of A3 Architecture Overviews in Subsea Front-End Engineering Studies: A Case Study

076: Developing a Topic Network of Published Systems Engineering Research

078: A Method to Visualize the Relationship between Regulations and Architectural Constraints

079: Opportunities and Challenges of Sociotechnical Systems Engineering

080: A Metrics Framework to Facilitate Integration of Disaggregated Software Development

081: A Concept for a Digital Thread based on the Connection of System Models and Specific Models

085: Using Models and Simulation for Concept Analysis of Electric Roads

088: Requirement Patterns in the Construction Industry

089: Solar Energy Investment Framework for Real Estate in Norway – a Case Study in Systems Engineering

094: The Systems Engineering Conundrum: Where is the Engineering?

095: Demonstrating the Value of Systems Engineering as the Professional Standard of Care

098: STPA-Sec Analysis for the DevSecOps Reference Design

099: Verification and Validation of SysML Models

101: Ontology-Based search engine for simulation models from their related system function

102: Resilience Requirements Patterns

103: The Benefits of Enhanced Contact Tracing and Quarantine to Resume and Maintain College-Ca ...

105: Systems Thinking in Socially Engaged Design Settings: What Can We Learn?

106: Systems Thinking: A Critical Skill for Systems Engineers

109: Implementation of tailored requirements engineering and management principles in a supplier to ...

110: Framework for Formal Verification of Machine Learning Based Complex System-of-System ...

112: Integrated Security Views in UAF

114: Investigation of Remote Work for Aerospace Systems Engineers

115: An Elaboration of Service Views within the UAF

116: Model-Based Systems Product Line Engineering of Physical Protection Systems

117: The Evolution of HELIX: A Competency Model for Complex Problem Solving

119: Technical Leadership of Virtual and Remotely Distributed Teams

123: Return on Investment in Model-Based Systems Engineering Software Tools

124: Measuring performance and identifying metrics of machine protection systems for particle accele

128: An Assessment of the Adequacy of Common Definitions of the Concept of System

130: Is CAD A Good Paradigm for MBSE?

131: Towards a Software Defined Truck

132: A Systems Engineering Approach to the Design and Education of a Robotic Baby

137: UAF (Unified Architecture Framework) Based MBSE (UBM) Method to build a System of Systems

Key Reserve Papers

KRP005: Overview of the Revised Standard on Architecture Description – ISO/IEC 42010

KRP010: Organizational Redesign Through Digital Transformation: A Case Study in the Life ...

KRP017: Architecture Literacy

KRP025: An Agile Systems Engineering Analysis of a University CubeSat Project Organization

KRP048: OMG RAAML standard for model-based Fault Tree Analysis

KRP063: Architecture Analysis Methods

KRP083: Broadening the Definition of Breadth in Systems Engineering

KRP091: MBSE Enabled Trade-Off Analyses

KRP096: Using Digital Viewpoint Concept Model for Defining Digital Engineering Information

KRP104: A State-of-Practice Survey of the Automotive and Space Industry Product Develop ...

KRP108: Automated trade study analysis based on dynamic requirements verification in the ...

KRP121: Workforce and Evaluation and Training for Digital Engineering in the US Department of …

KRP125: Conceptualizing the Lessons Learned Process in Delivery Projects: Detecting …

KRP136: Model of Models Methodology: Reuse Your Architectural Data

Presentations

Pr01: Using Heuristics to Refine the System Physical Architecture

Pr02: Utilizing a Human Readiness Level (HRL) Scale to Promote Effective System Integration

Pr03: How do we know that we know? - A Model-Based-Knowledge-Management Concept Support …

Pr07: Systems Engineering Professional Certification Standard

Pr08: How to get the most out of your Systems Engineering consultants

Pr09: Delighting your client as a Systems Engineering consultant

Pr11: Safety Engineering of Semi-Autonomous Cars

Pr12: Towards an Integrated Approach of Systems Behavior Modeling and Specification

Pr13: 6 Vs and 3 Ts of Systems Engineering

Pr14: Conflict is your friend- Managing healthy conflict in the systems engineering workplace

Pr15: Making Your Case- Negotiation and persuasion for the systems engineer

Pr16: Providing truth, trust and traceability to MBSE

Pr17: Economic Analysis of Unmanned Aerial Vehicle (UAV) Platform Options

Pr18: Am I doing the right job and am I doing the job right?

Pr20: System Holarchy Structures for Sustainable Development Goals

Pr21: Integrating MBSE and Product Lifecycle Management

Pr22: System of Systems Modeling to empower decision makers in drone based services

Pr23: A Systems Theory Approach to Building Management

Pr24: Designing Systems by Drawing Pictures and Telling Stories

Pr25: Systems Engineering – A Matter of Perspectives

Pr26: Practical demonstration of a highly functional system-centric digital thread

Pr28: MBSE Components in the Supply Chain, Spring 2021 Student Capstone Project

Pr29: Ushering in a New Era for Feature-based Product Line Engineering with the ISO/IEC 26580

Pr30: Defining a Measurement Framework for Digital Engineering

Pr31: From Systems to Silicon: MBSE-Enabled Digital Electronics Verification

Panels

P1: The MBSE Futurist's Dilemma: Diffusing systems engineering practices in an AI dominated era

P2: Systems Engineering at the Hello – Frameworks for Applying Systems Engineering in Early Stage

P3: A Framework for Understanding Systems Engineering Principles and Heuristics

P4: Human-AI Teaming: A Human Systems Integration Perspective

P7: Solving the Digital Engineering Information Exchange Challenge

P8: Heuristics for Systems Engineering: Useful or Dangerous? Outdated or Enduring

P9: Investigating transdisciplinary systems approaches for health care access

Tutorials

T07: Overview of the INCOSE SE Handbook Version 4.0

T15: Systems Security Engineering: A Loss-Driven Focus

T19: Applied Systems Theory to Enhance Systems Engineering Practice for Complex Systems

T20: Handling Organizational Complexity

T21: From Operational Concept Development to Systems Architecture Definition with SysML and MBSE

T23: Leadership Skills for Systems Engineers

T24: Modeling and Analysis of Standard Operating Procedures

T25: Introduction to Model Simulation and Engineering Analysis with SysML

T26: Artificial Intelligence for Systems Engineers: Going Deep With Machine Learning and Deep Neur …

Invited Content

IC01: Viewing Grand Challenges as a System

IC03: Using Systems Thinking to Add Value in these Uncertain Times

IC04: Spectacular Views of the City

IC04: The next Systems Challenge: Developing resilient, effective, inclusive, sustainable so…

ICT01: Panel: To "Vee" or not to "Vee"

ICT02: DE meets SE: Building a Joint Culture

ICT03: S.O.S. for FSS: The need for Systems of Systems (SoS) Thinking per Financial ..p. 147

ICT04: Leading the Way to Diversity, Equity, and Inclusion in Systems Engineering

ICT05: Panel: The Journey from SysML 1.7 to 2.0

NASA Video Series on Systems Engineering

NASA has available online a useful video series on aspects of systems engineering. The individual video titles are:

1. Systems Engineering and the Project Life Cycle
2. The Central Elements of Project Management
3. How to Use Requirements
4. How to Conduct a Review
5. Systems Design vs. Engineering Design
6. The Central Role of the Systems Hierarchy
7. The Riskiest Part of Systems Design - Interfaces and Integration
8. Concept of Operations
9. Using Trade Studies to Make Systems Decisions
10. Integrating Reliability, Safety, and the Other Specialties
11. Verification - How to be Sure You Got What You Needed.

In contrast to the view reflected in the NASA videos, PPI holds the view that all engineers should be systems engineers in the sense of practicing systems engineering within the scope of their assigned responsibilities, not be a breed apart as represented in the video series. Also, unfortunately, Video 8 claims that a Concept of Operations (CONOPS) and an Operational Concept Description (OCD or OpsCon) are synonyms, whereas this is not so (ref PPI, ISO and INCOSE publications). For an Operational Concept Description, OCD (also called CONUSE, CONEMP, Statement of Operating Intent, Intended Use Description, etc.), the focus is a system-centric description of intended use of the subject of the OCD in terms of users and their relevant characteristics, uses with respect to each user, how the system is to be used for each use, and the expected or interned external conditions during use. A CONOPS for the same system is an operational solution description. Also, the focus of Video 8 is indeed (system) Verification, but verification does not ensure that you got what you needed as claimed in the video, that is validation. The video series nevertheless contains some valuable content, which may be viewed here: https://www.nasa.gov/content/systems-engineering-for-university-level-engineering-projects-and-competitions

---

Systems Engineering and Model-Based Systems Engineering Stakeholder State of the Discipline

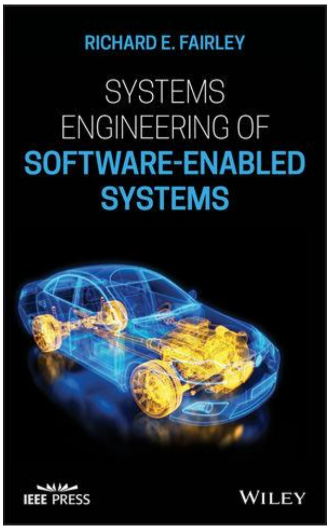
To understand the external state of the Systems Engineering (SE) discipline, the NASA SE Technical Discipline Team (TDT) in 2019 undertook a comprehensive survey across the aerospace industry using over fifty sources. The sources represented academia, USA government agencies, companies that are suppliers to NASA, and SE tool vendors. The survey complements and allows comparison against a similar comprehensive 2017 survey across NASA's internal workforce. This survey has allowed the NASA SE TDT to perform alignment checks on investments it has made in re-tooling SE into the future, such as with MBSE.

The SE TDT's study of the survey results concluded that within the SE discipline, the workforce is healthy and proficient in its technical, systems management, and leadership skills. And that the SE industry is in agreement that the discipline should be focused on innovation. The study also captured the state of MBSE, which is still very much in its infancy. A majority of study sources have infused the methodology into approximately 25% of their organizations, and whilst most agree there are significant benefits to be gained, measuring those benefits can be challenging. Because the study revealed that NASA and its partners are in similar spaces, the aerospace industry has a unique opportunity to partner with its stakeholders in advancing SE.

The following additional information is available here: https://www.nasa.gov/nesc/articles/se-mbse-state-of-the-discipline

- **Study Summary**: Systems Engineering & Model Based Systems Engineering State of the Discipline (PDF)
- **Full Report**: Independent Assessment of Perception from External/non-NASA Systems Engineering (SE) Sources (PDF)
- **Webinar**: Systems Engineering and Model Based Systems Engineering Stakeholder State of the Discipline
- Responses to Q&A.

Book: Systems Engineering of Software-Enabled Systems



Author: Richard E. Fairley
ISBN: 978-1-119-53501-0 June 2019

Wiley-IEEE Press 432 Page

Taken from the Wiley webpage, "*Systems Engineering of Software-Enabled Systems* offers an authoritative review of the most current methods and techniques that can improve the links between systems engineering and software engineering. The author offers an introduction to systems engineering and software engineering and presents the issues caused by the differences between the two during the development process. The book reviews the traditional approaches used by systems engineers and software engineers and explores how they differ.

The book presents an approach to developing software-enabled systems that integrates the incremental approach used by systems engineers and the iterative approach used by software engineers. This unique approach is based on developing system capabilities that will provide the features, behaviors, and quality attributes needed by stakeholders, based on model-based system architecture. In addition, the author covers the management activities that a systems engineer or software engineer must engage in to manage and lead the technical work to be done. This important book:

- Offers an approach to improving the process of working with systems engineers and software engineers
- Contains information on the planning and estimating, measuring and controlling, managing risk, and organizing and leading systems engineering teams
- Includes a discussion of the key points of each chapter and exercises for review
- Suggests numerous references that provide additional readings for development of software-enabled physical systems
- Provides two case studies as running examples throughout the text."

Website: https://sezert.org/en/

This is the home page of the Germany-based SE-ZERT® program. SE-ZERT® is a vocational qualification to the Certified Systems Engineers (GfSE)®“. The program provides the participants opportunities to build and enhance process and content related competencies in the area of systems engineering. The program relies on the standard EN ISO/IEC 17024 for personal certification. Accredited training providers support participants in preparation for the certification, whereby an examination committee by the SE-ZERT® program verifies the participant's knowledge. With a SE-ZERT® certificate, alumni are qualified for industry-independent systems engineering engagements. This vocational qualification is globally available in German and English.The home page provides an overview on the process from looking for a training provider through to achieving the credential as a Certified Systems Engineer (GfSE)®. The process concludes with receipt of a legally protected title with unique identification number awarded by the GfSE e.V.

Swiss Society of Systems Engineering: SSSE Model-based Systems Engineering (MBSE) Knowlegde Exchange

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.

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

US Government Risk Management Framework

The USA National Institute of Standards and Technology (NIST) Risk Management Framework (RMF) is a seven-step process for integrating security, privacy, and supply chain risk management into the system development lifecycle of federal government information systems. Though the RMF is mandatory for federal organizations, it can also be useful for state, local, tribal, and territorial (SLTT) and nongovernmental organizations. However, some of the RMF’s roles and processes are specific to the USA federal government context. See nist.gov/rmf. These aspects of the RMF have been translated into guidance for use by non-federal organizations.

In the USA federal government context, most roles are clearly defined and job titles are prescribed and sector-specific, which may not be the case in USA non-federal, foreign and commercial spaces. Potentially equivalent roles and possible ways to adapt the USA federal processes to other organizations in each of the RMF’s seven steps are described here: https://insights.sei.cmu.edu/blog/translating-the-risk-management-framework-for-nonfederal-organizations by Carnegie Mellon University.

“

Designing a solution doesn’t change the problem! But it may call into question some aspects of the problem definition.

ROBERT HALLIGAN

SYSTEMS ENGINEERING IN SOCIETY

Organization: The Digital Twin Consortium

The Digital Twin Consortium is a global consortium comprising industry, government, and academia. It was founded to accelerate the development, adoption, interoperability, and security of digital twins and enabling technologies.

Digital Twin is a virtual representation of real-world entities and processes, synchronized at a specified frequency and fidelity (DTC). A digital twin is a virtual representation of an object or system that spans its lifecycle, is updated from real-time data, and uses simulation, machine learning and reasoning to help decision-making (IBM). This means creating a highly complex virtual model that is the exact counterpart (or twin) of a physical thing. The 'thing' could be a car, a building, a bridge, or a jet engine. Connected sensors on the physical asset collect data that can be mapped onto the virtual model. Anyone looking at the digital twin can now see crucial information about how the physical thing is doing out there in the real world.

Digital twins let us understand the present and predict the future. A digital twin is therefore a tool to help engineers and operators understand not only how products are performing, but how they will perform in the future. Analysis of the data from the connected sensors, combined with other sources of information, allows us to make these predictions.

The Digital Twin Consortium aims to propel the innovation of digital twin technology through consistent architecture and design methods, open-source collaboration, and development of best practices. It is committed to demonstrating the value of digital twin technologies and guiding positive business outcomes for digital twin end users.

The Digital Twin Consortium has three primary objectives:

- Influence the direction of the digital twin industry
- improve interoperability of digital twin technologies, and
- influence the requirements for digital twin standards.

Much of the work of the consortium is carried out through its eight Working Groups:

- Aerospace & Defense
- FinTech
- Healthcare & Life Sciences
- Infrastructure
- Manufacturing
- Natural Resources
- Security & Trustworthiness
- Technology, Teminology & Taxonomy (3T)

The members are committed to using digital twins throughout their global operations and supply chains. They set de facto technical guidelines and taxonomies, publish reference frameworks, develop requirements for new standards, and share use cases to maximize the benefits of digital twins.

The consortium is open to any business, organization, or entity with an interest in digital twins. Digital Twin Consortium is part of the Object Management Group®.

More information: https://www.digitaltwinconsortium.org/index.htm

"

I don't spend my time pontificating about high-concept things; I spend my time solving engineering and manufacturing problems.

ELON MUSK

Systems Engineering at the University of Lagos, Nigeria

A Systems Engineering Department within the University of Lagos was established in 2000 and has performed well ever since. The Department produced the top graduate student in 2016, and systems engineering students led the team that won the Global Management Challenge competition back-to-back in 2018 and 2019. Projects and dissertations have mainly revolved around robotics, biomedical engineering, simulations and modelling.

The program provides students with basic education and skills in analysis, design, monitoring and control of engineering systems. The program stresses the importance of humanistic and societal concerns as they shape the designer's approach to solution of problems confronting modern society. The systems engineering practitioner therefore strives to serve the dual needs of society for the design of reliable and efficient systems, whilst protecting the overall integreity of the host environment. The objective of the programme are:

- To bridge the gap between management/decision science and the engineering profession through the integration of decision science/management with traditional engineering disciplines
- To produce engineers with multidisciplinary skills for today's complex economy
- To impart analytical and cutting-edge computing skills
- To initiate and carry out engineering design
- To engage in management and to pursue research and development.

Undergraduate courses offered in the Department lead to the award of the B.Sc. (Honours) in Systems Engineering. The Department also offers postgraduate programs of M.Sc. and PhD degrees in Systems Engineering.

The Department has produced brilliant minds, many of whom are in industry and academia. The likes of late Prof. Oye-Ibidapo Obe and Late Professor Olunloyo are revered within the Faculty of Engineering community because of their collaborative efforts to establish the Systems Engineering Department.

More information: https://silo.tips/download/department-of-systems-engineering-university-of-lagos

Featured Organization: Digital Metrology Standards Consortium (DMSC™)

The Digital Metrology Standards Consortium (DMSC™ Inc.) aims to identify needed standards in the field of digital metrology, and to promote, foster, and encourage the development and interoperability of these standards, along with related and supporting standards that will benefit industry as a whole. Metrology is the scientific study of measurement. Metrology fosters a common understanding of units of measurement. The Dimensional Measuring Interface Standard (DMIS) and the Quality Information Framework (QIF) are two such standards for which the DSMC has responsibility to continue development, maintenance and support, as well as coordination and harmonization with other related standards efforts. The DMSC™ is an ANSI Accredited Standards Developing Organization, as well as an A-Liaison to ISO.

QIF, especially relevant to systems engineering tools information exchange, is a Unified XML Framework standard for CAD quality measurement systems. QIF was embraced by the International Standards Organization (ISO) and released as ISO 23952:2020 in August 2020.

The DSMC invites participation within the consortium of other standards groups and activities that seek to resolve the technology and other issues of automated (digital) metrology. Membership of the DMSC is open to companies and other organizations with an interest in the field.

For more information on the DMSC see: https://qifstandards.org/about-dmsc/

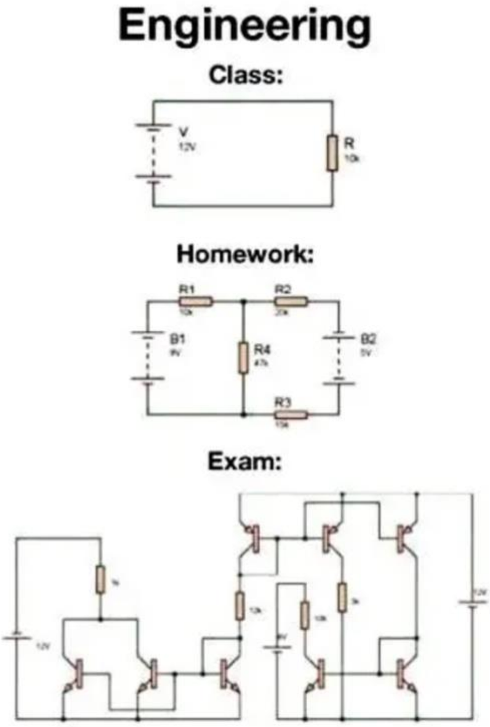
"

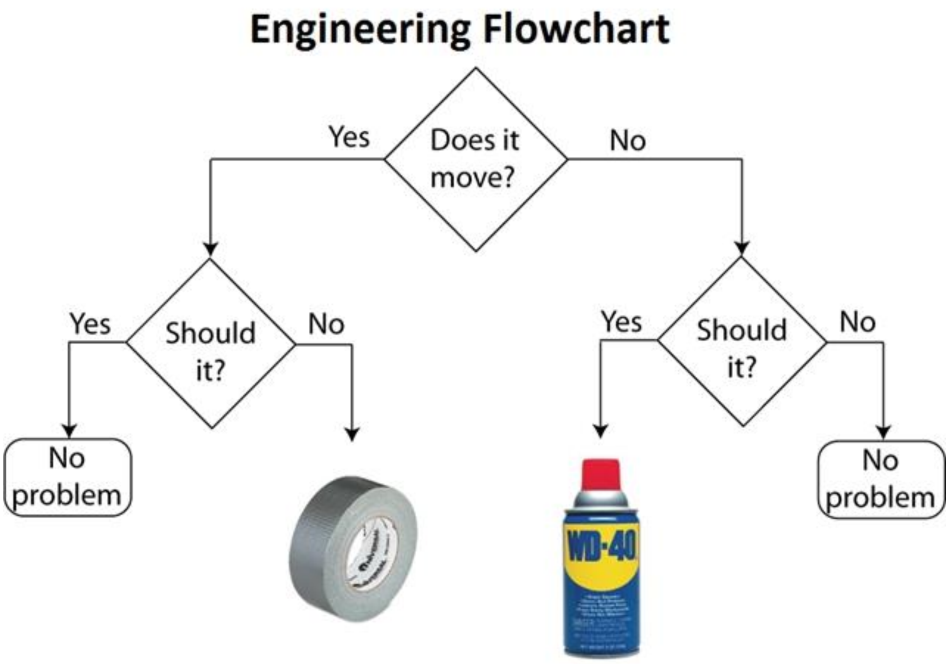
The trick to having good ideas is not to sit around in glorious isolation and try to think big thoughts. The trick is to get more parts on the table.

STEVEN JOHNSON

FINAL THOUGHTS

Hi all! Pippi taking for over from Syenna for this edition! Even crazy converts to systems engineering deserve a break form the hard work of wrapping up these newsjournals. I wanted to share with you some graphics that unite us as engineers! Anyone relate?





The definition of an engineer

Definition of an engineer: somebody who makes precise guesswork based on unreliable data provided by people with questionable knowledge. Never wrong. Likes tables.