



A Calendar for Systems Engineering

*Dr Yogananda Jeppu
Distinguished Lecturer, IEEE 2024-2026
13 May 2026*

Copyright Notice

A Calendar for Systems Engineering by Yogananda Jeppu is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.

You are free:

to Share — to copy, distribute and transmit the work

to Remix — to adapt the work

Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Noncommercial — You may not use this work for commercial purposes.

Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

For details please visit the website.



Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nc-sa/3.0/>

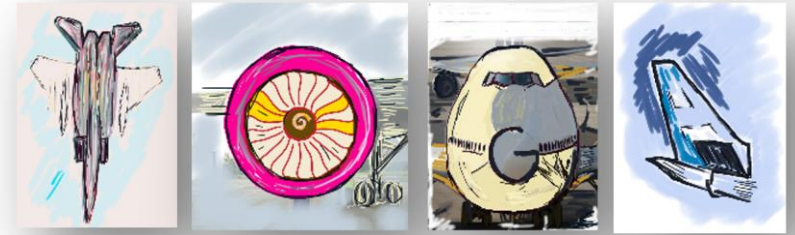
Disclaimer

- The views, methods are my experiences working in the field for some time. What I have experienced may not be applicable to your application. Use this presentation as a knowledge gain but USE YOUR JUDGMENT!
- I have used pictures and materials available in Google. I have not collected the references every time. In case anyone feels that I have used their published material and not referenced it here please feel free to mail me. I will put in the reference.
- **The firm/company I work for does not endorse these views which are mine.**

Presenter

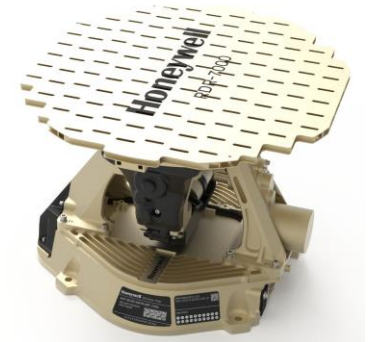


About me



- I am Yogananda Jeppu. I am a Distinguished Lecturer of IEEE Systems Council
- I have a PhD in safety critical control system testing using models. I have 38 years experience in control system design, 6 Degree Of Freedom simulation, Verification and Validation, System Testing, MBSE, and Formal methods.
- I have worked on the Indian Light Combat Aircraft (LCA) fly by wire control system and the Indian SARAS aircraft autopilot. I have worked on model based commercial aircraft fly by wire flight control law programs of Boeing, Airbus, Gulfstream and Comac C919.
- I am an adjunct faculty at DIAT Pune and MIT Manipal.
- I am on the divisional board of the National Aerospace Laboratories.

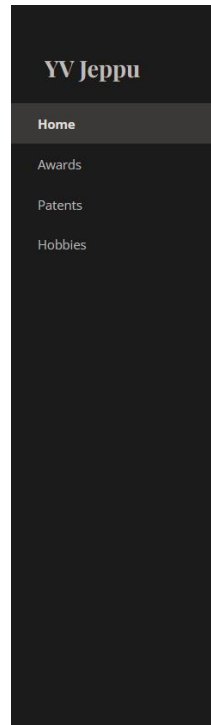
Some of my projects



Visit my website <https://sites.google.com/view/yvjeppu/> for links to these projects

Check out my personal website

You can find some good Simulink models and tutorials on SE



Yogananda Jeppu



About me

Hi,

I am Yogananda Jeppu. I have 36 years experience in control system design, 6DOF simulation, Model Based Verification and Validation, System Testing. I have worked on [Akash Missile System](#), the [Indian Light Combat Aircraft \(LCA\)](#) control system and the [Indian SARAS aircraft](#). I have worked on model based commercial aircraft flight control law programs of [Boeing 747-8](#), [Airbus A 350](#), [Gulfstream](#) and [Comac C919](#).

Currently I am a [Technical Fellow](#) at Boeing, looking at [Model Based System Engineering](#) and [Model Based Safety](#).

My earlier projects from Honeywell are [Honeywell Anthem](#) and the [RDR 7000 weather radar system](#).

I have a PhD in safety critical control system testing from [IIT Bombay](#), India, a MTech from [DIAT Pune](#), India, and BE from [Manipal Institute of Technology, Manipal, India](#).

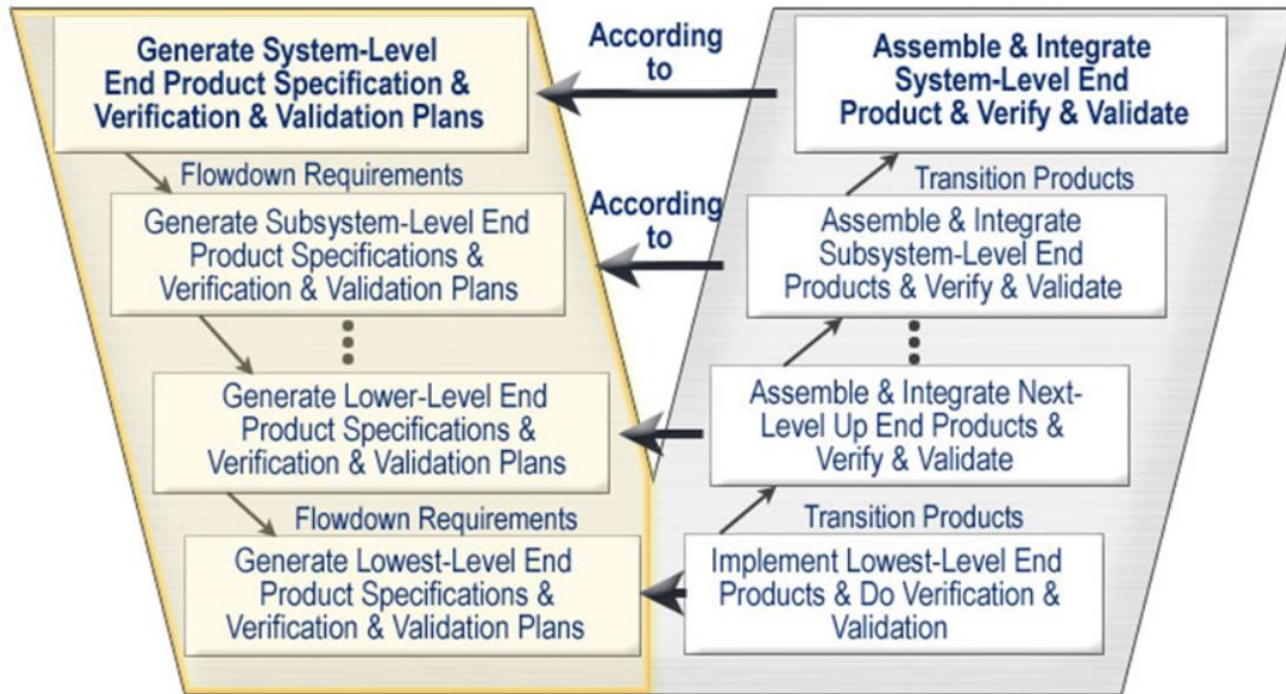
Email me : yvjeppu@gmail.com

[My LinkedIn Profile](#) provides a detailed resume



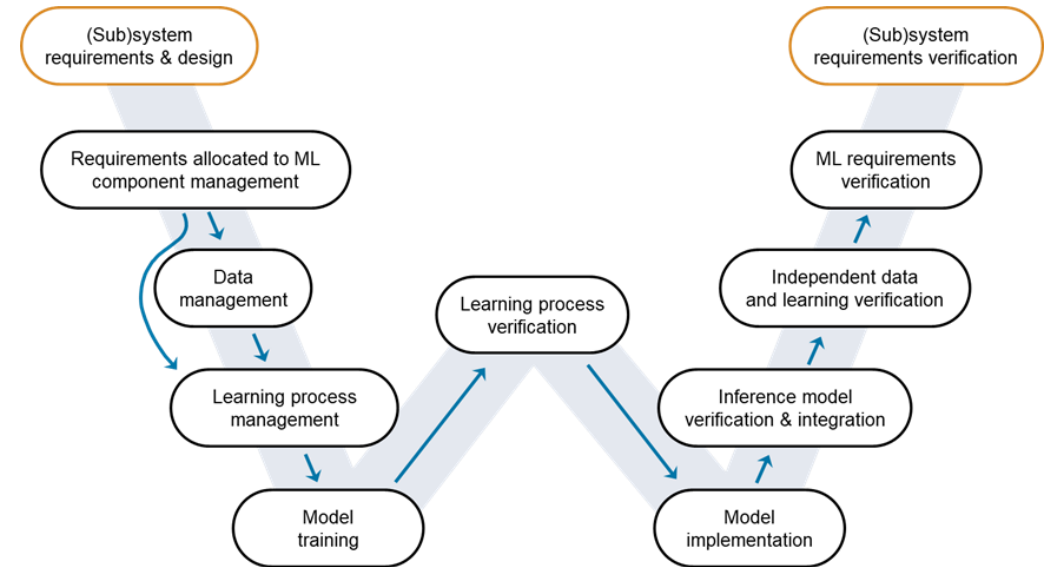
Systems Lifecycle

The 'V' – A Macro View



https://sebokwiki.org/wiki/Vee_Life_Cycle_Model

<https://blogs.mathworks.com/deep-learning/2023/07/11/the-road-to-ai-certification-the-importance-of-verification-and-validation-in-ai/>



We have got tired of seeing this shape so I am giving a twist to the tail here!

The Twelve Months

By George Ellis

**Snowy, Flowy, Blowy,
Showery, Flowery, Bowery,
Hoppy, Croppy, Droppy,
Breezy, Sneezzy, Freezy.**



The Twelve Months



***Bounding, Operating, Breaking,
Modelling, Requiring, Proofing,
Certifying, Architecting, Optimizing,
Testing, Deploying, Sustaining!***

This will be our story today

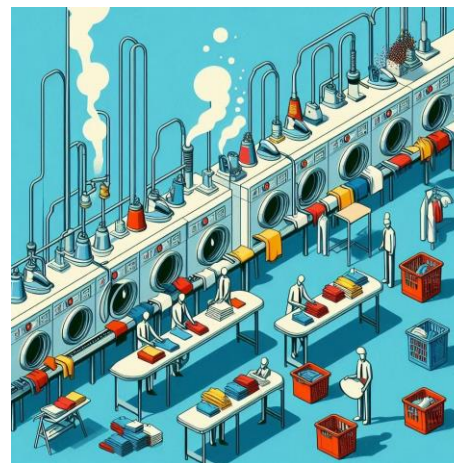
In the depth of winter, I finally learned that within me there lay an invincible summer.

JANUARY

2026
CALENDAR

SUN	MON	TUE	WED	THU	FRI	SAT
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Define your boundary



Left is
definitely
not equal
to Right

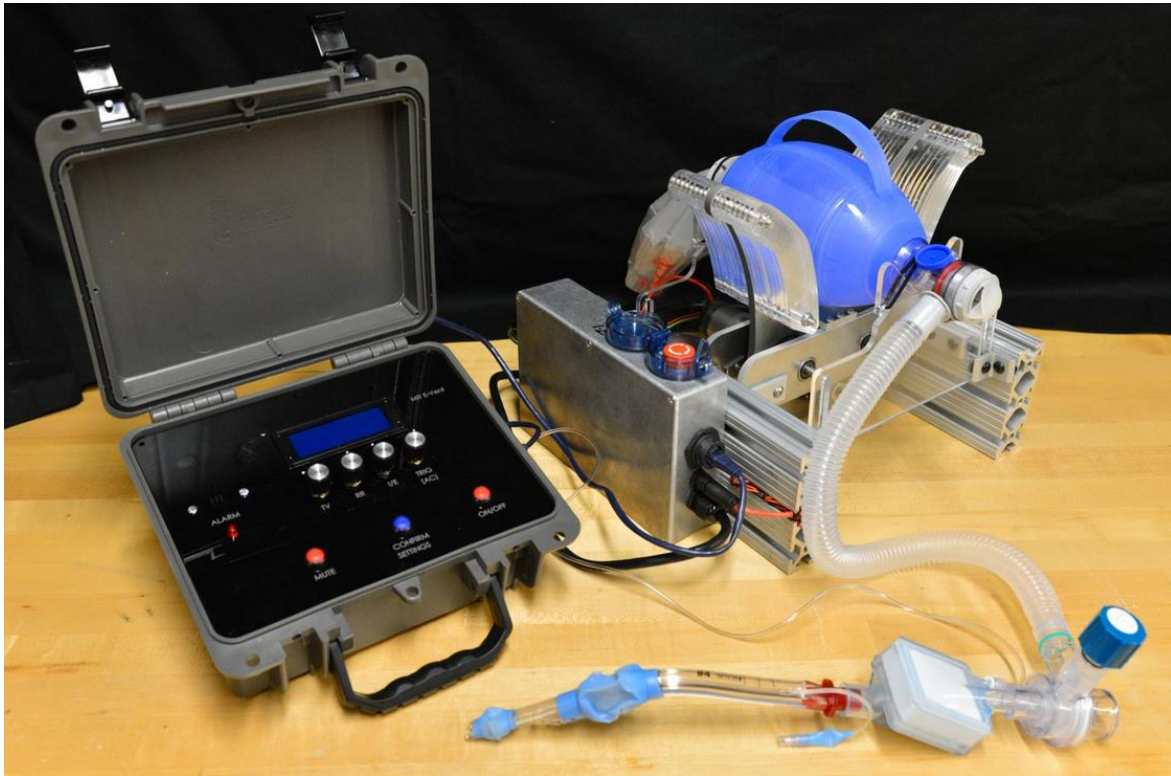


Define your boundary

- **Developmental control:** does the system developer have control over the entity's development? Can the developer influence the requirements of the entity, or are requirements defined outside of the developer's sphere of influence? Is funding part of the developer's budget, or is it controlled by another organization?
- **Operational control:** once fielded, will the entity be under the operational control of the organization that controls the system? Will the tasks and missions performed by the entity be directed by the owner of the system? Will another organization have operational control at times?
- **Functional allocation:** in the functional definition of the system, is the systems engineer "allowed" to allocate functions to the entity?
- **Unity of purpose:** is the entity dedicated to the system's success? Once fielded, can the entity be removed without objection by another entity?

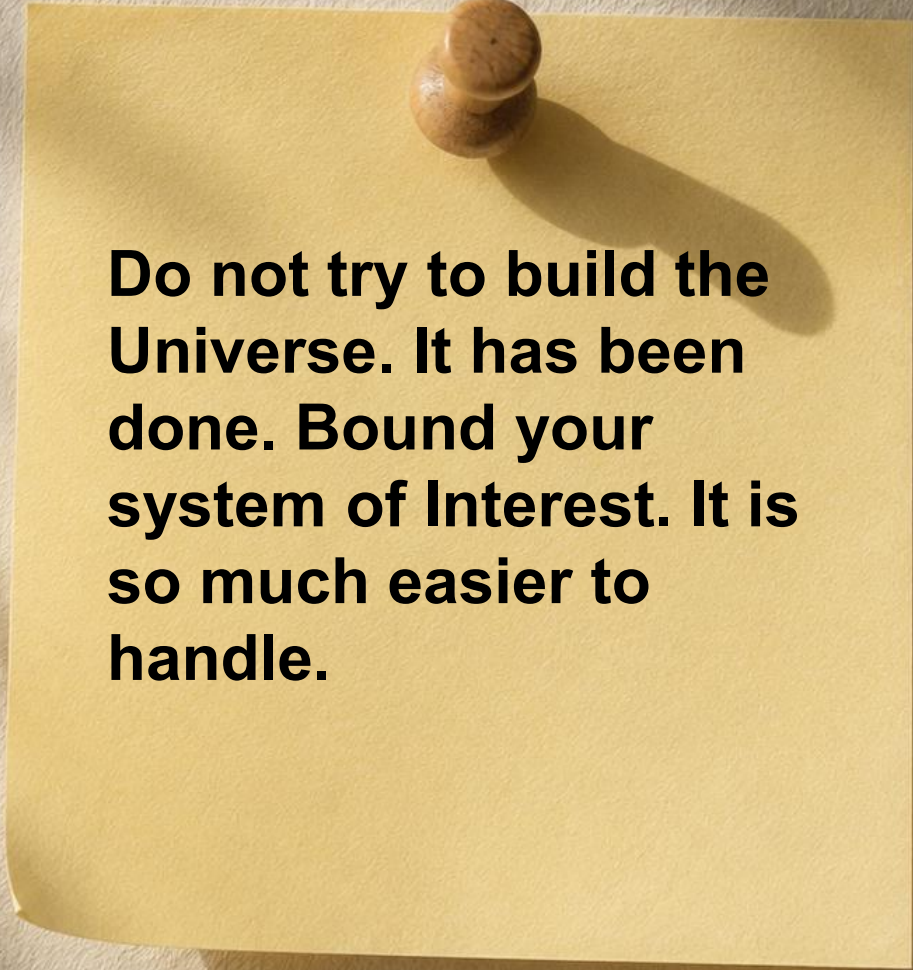
Boundary

- <https://emergency-vent.mit.edu/>



The MIT Ventilator system has this as the boundary. Outside of this will be the oxygen cylinders, pipes in hospital, delivery of oxygen, emergency generators etc.

Bounding



Do not try to build the Universe. It has been done. Bound your system of Interest. It is so much easier to handle.

Though February is short, it is filled with lots of love and sweet surprises.

FEBRUARY

2026 CALENDAR

SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

Define your ConOPs

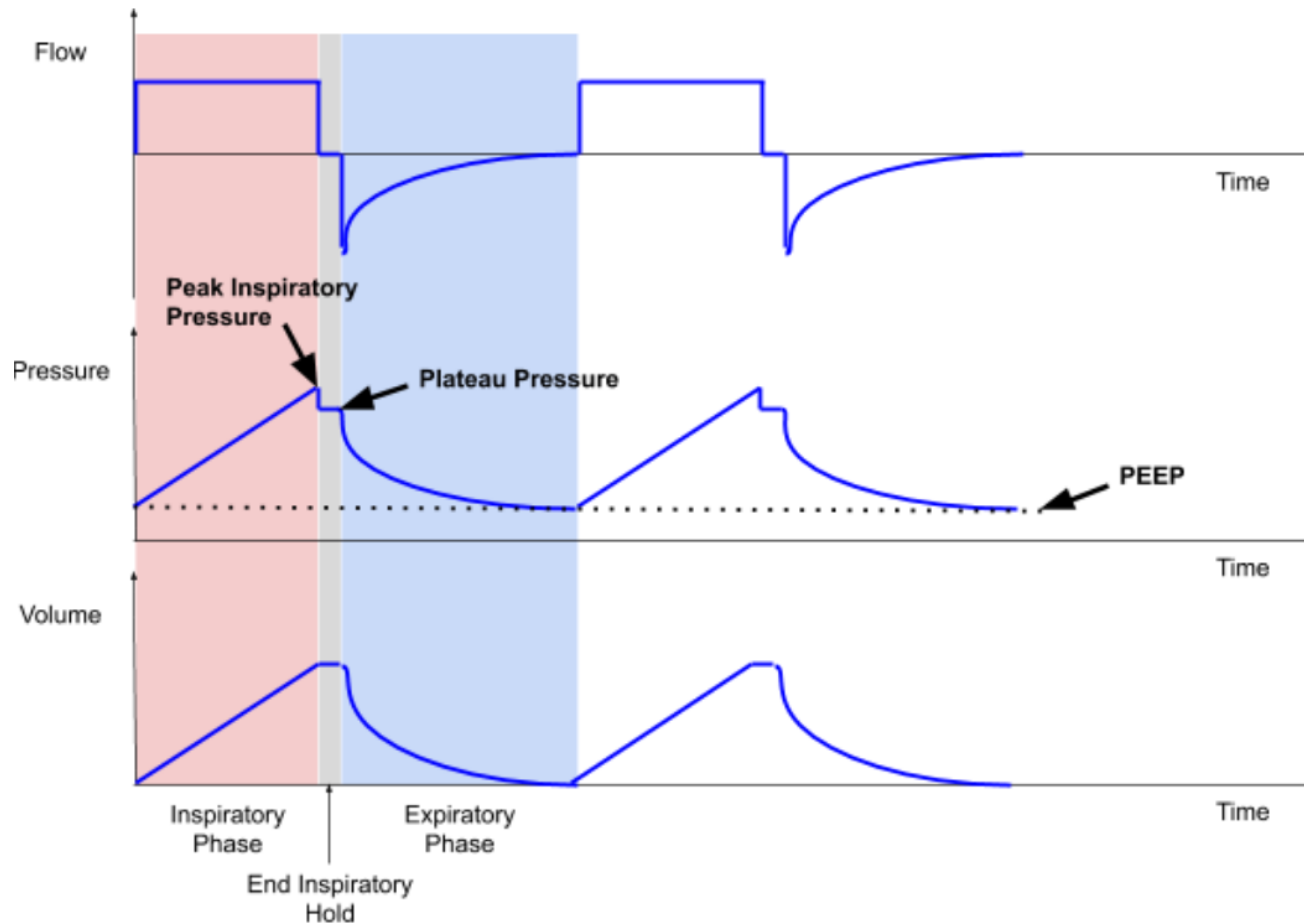


Using coffee machines today can be daunting to many. I see this in the airport lounges.

ConOps

- The concept of operations (**ConOps**) establishes the foundation for initial design specifications according to the long-term vision. Throughout the systems life cycle, the operation of the system requires the systems engineer to be an active participant in reviews, change management and integrated master schedule activities to ensure the system operations continue to meet the evolving needs of stakeholders, and are consistent with the architecture through the eventual decommissioning or disposal of the system. – SEBoK
- - [https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_\(SEBoK\)](https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK))
- **ConOps is how a human or other systems interact with the system in your boundary.**

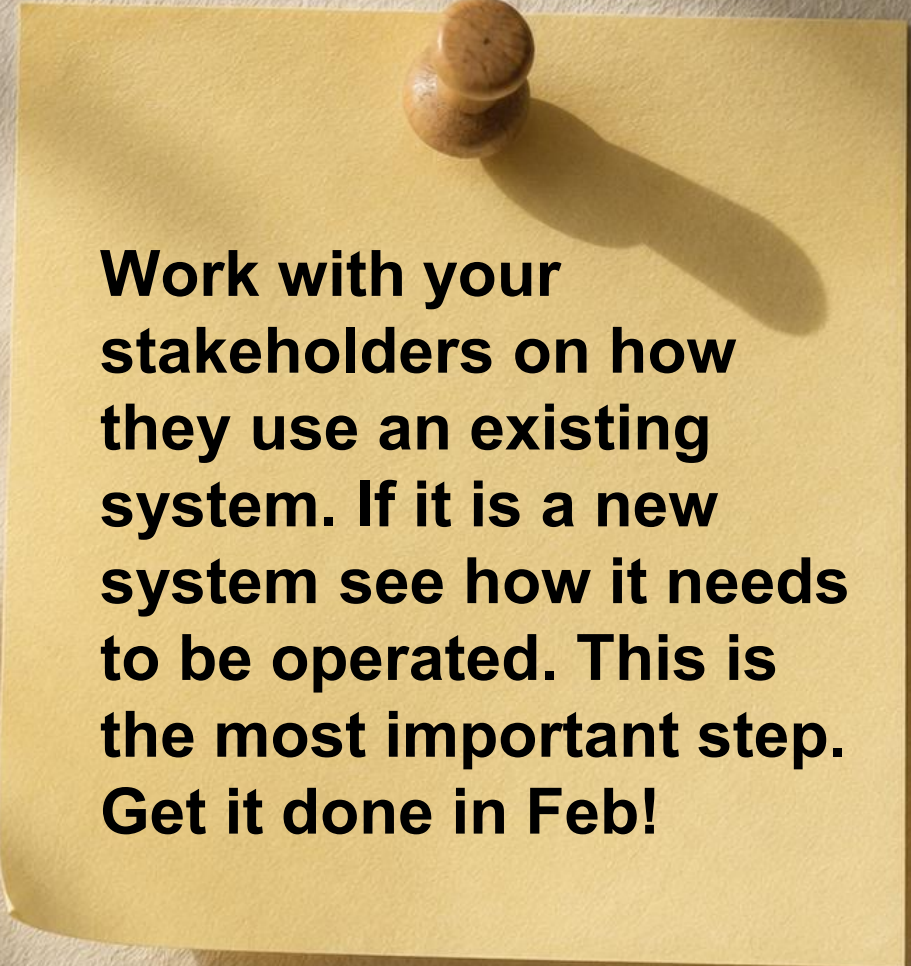
Ventilator



In ConOps we need to specify how the **STAKEHOLDERS** interact with the system in your boundary and **HOW** it interacts with other systems over **TIME**.

This has to be detailed up front.

Operating



Work with your stakeholders on how they use an existing system. If it is a new system see how it needs to be operated. This is the most important step. Get it done in Feb!

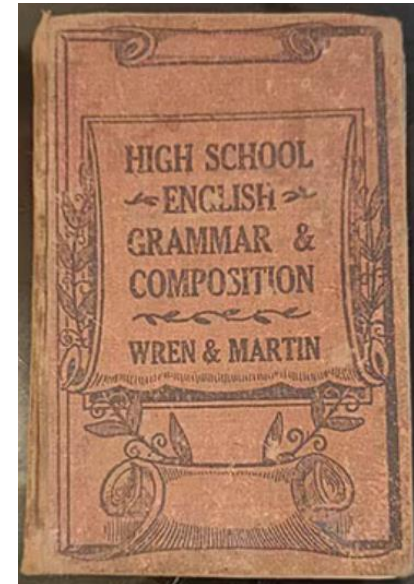
Spring is nature's way of saying, 'Let's party!'

MARCH

2026
CALENDAR

SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Define your Nouns and Verbs



I have found systems are always about nouns and verbs. I wonder is a grammar book will help us 😊

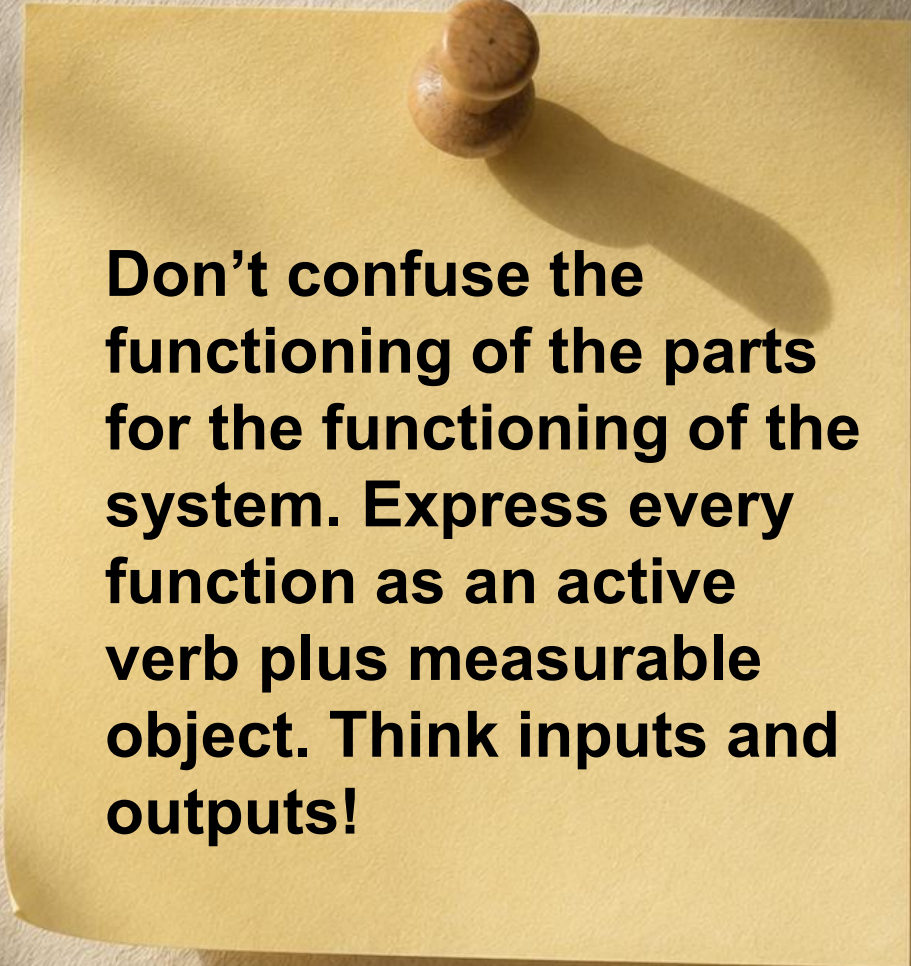
Functional breakdown

- This is kind of easy. What are the verbs “ings” required to make the ConOps happen? A very simple question.
- Make a list of all these “ings”.
- **WHAT** is the system supposed to do is the question you need to ask? **WHAT** is the **core transformation** the **bounded system** is providing. What Nouns are the Verbs consuming/transforming?
- Aircraft brake management system – decelerates the aircraft on ground. So “**aircraft decelerating**” is a primary function of the system. **Remember, all you need to write is what the function does, not how it does it. Note – I am not saying braking here.**

Functional breakdown

Function (verb phrase)	Consumes (Inputs)	Produces (Outputs)
Delivering ventilation	Pressurized air/oxygen mixture, control settings (RR, TV, I:E, PEEP, mode)	Inspiratory airflow into patient lungs
Triggering breath	Patient effort signal (pressure/flow drop), trigger sensitivity setting	Initiation of mechanical inspiration
Maintaining PEEP	Expiratory valve control, PEEP setting	Stable positive pressure at end expiration
Blending gases	Oxygen supply, air supply, FiO ₂ setpoint	Mixed gas stream at desired oxygen concentration
Controlling flow and pressure	Valve control signals, target pressure/volume/time profile	Delivered flow profile and pressure waveform
Monitoring airway pressure	Airway pressure sensor data	Continuous pressure measurements
Limiting airway pressure	Real-time pressure data	Relief action (valve actuation, cutoff, or alarm) to cap pressure at ≤40 cm H ₂ O

Breaking



Don't confuse the functioning of the parts for the functioning of the system. Express every function as an active verb plus measurable object. Think inputs and outputs!

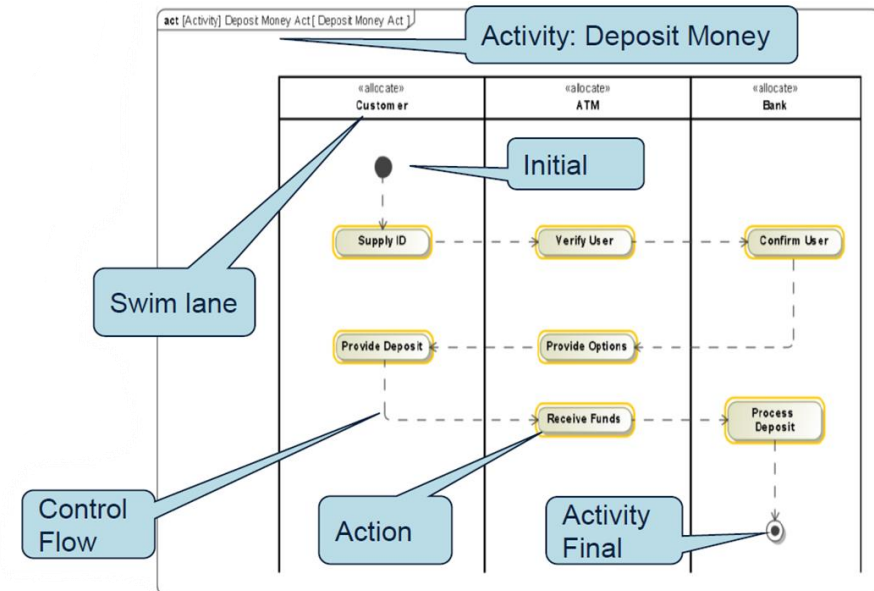
April hath put a spirit of youth in everything

APRIL

2026
CALENDAR

SUN	MON	TUE	WED	THU	FRI	SAT
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Define your Models



OPM – Object Process Model



US 20070050180A1

(19) **United States**

(12) **Patent Application Publication**
Dori

(10) **Pub. No.: US 2007/0050180 A1**

(43) **Pub. Date: Mar. 1, 2007**

(54) **MODELING SYSTEM**

(76) Inventor: **Dov Dori**, Kfar-Vradim (IL)

Correspondence Address:
PEARL COHEN ZEDEK, LLP
PEARL COHEN ZEDEK LATZER, LLP
1500 BROADWAY 12TH FLOOR
NEW YORK, NY 10036 (US)

(21) Appl. No.: **11/509,599**

(22) Filed: **Aug. 25, 2006**

Related U.S. Application Data

(63) Continuation of application No. 09/808,781, filed on Mar. 15, 2001, now Pat. No. 7,099,809.

(60) Provisional application No. 60/201,860, filed on May 4, 2000. Provisional application No. 60/201,860, filed on Jul. 5, 2000.

Public

(51) **Int. Cl.**
G06F 17/10

(52) **U.S. Cl.**

(57)

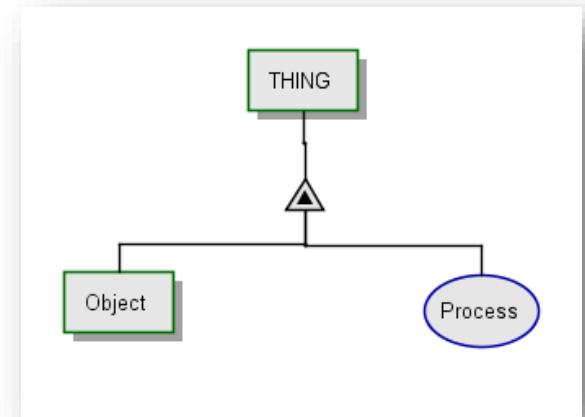
Disclosed herein are descriptions of a graphing technique that models objects and relationships between entities. The technique includes automatic code generation and language translation.

The screenshot shows the ISO website interface. At the top, there is a navigation bar with the ISO logo and the text "International Organization for Standardization" and "When the world agrees". Below this, there are several tabs: "Standards", "All about ISO", "Taking part", and "Store". The "Store" tab is selected. A search bar is visible on the right. Below the navigation bar, there is a breadcrumb trail: "Home > Store > Standards catalogue > Browse by ICS > 25 > 25.040 > 25.040.01 > ISO/PAS 19450:2015". The main content area displays the title "ISO/PAS 19450:2015" with a "Preview" button. Below the title, the text reads "Automation systems and integration -- Object-Process Methodology".

https://en.wikipedia.org/wiki/Object_Process_Methodology

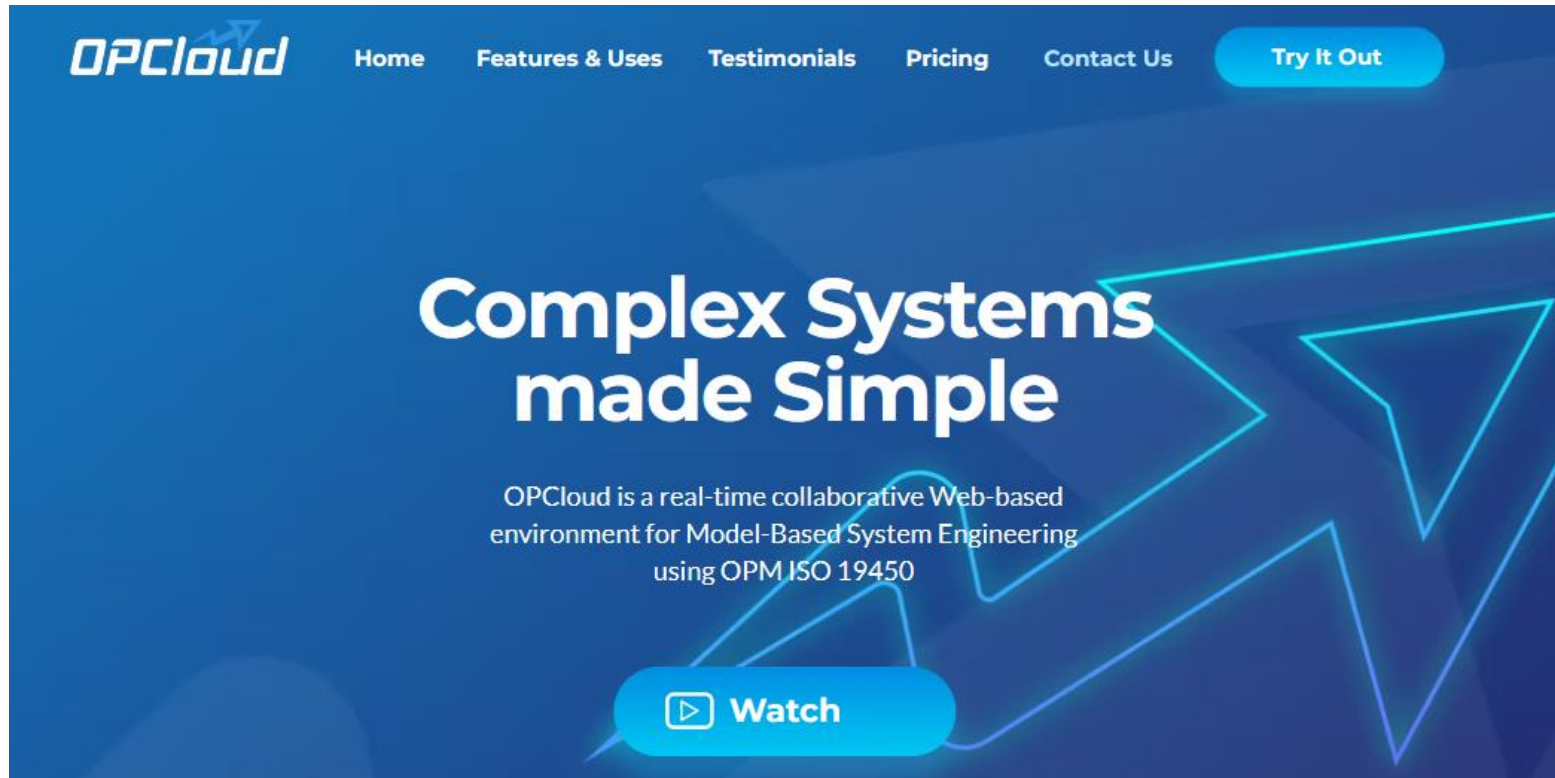
OPM – Object Process Model

- The basic element of the OPM is the existence of the THING. The thing is something that exists and it does something.
- The existence of the THING is the Object (physical or mental)
- The “does something” of the THING is the Process (transforms Object
- There is much material on the internet including a MBSE course using OPM.
- At this time tool is not very important. It is important to know the modelling language and its grammar and what you are talking about using this modelling language.



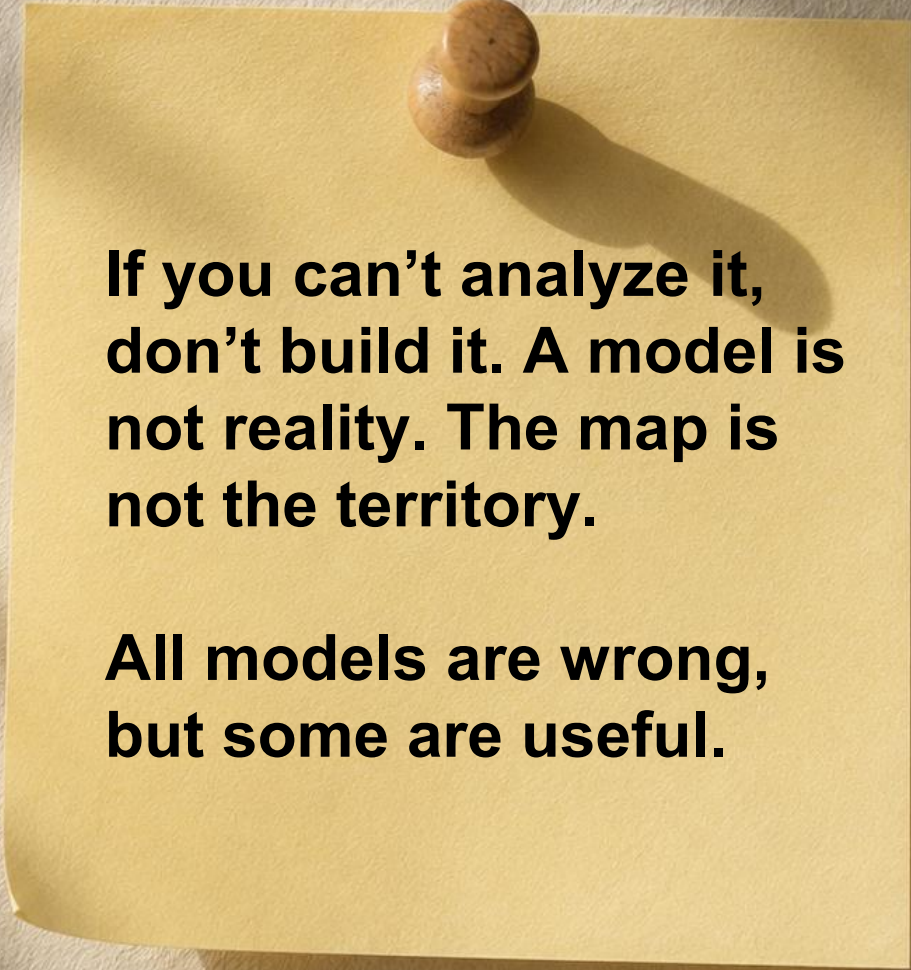
THING is physical.
THING exhibits Object, as well as Process.
Object is physical.

OPM – Object Process Model



<https://www.opcloud.tech/>

Modelling



**If you can't analyze it,
don't build it. A model is
not reality. The map is
not the territory.**

**All models are wrong,
but some are useful.**

The world's favorite season is the spring. All things seem possible in May.

MAY

2026
CALENDAR

SUN	MON	TUE	WED	THU	FRI	SAT
31					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Define your WHATs

What I want to do or build	What is the boundary	What are the functions	What are the interactions
What are the hazards	What are mitigations for the hazards	What are the requirements	What are the constraints
What is done by software	What is done by hardware	What is done by others	What is the design for functions
What is to be done to implement	What is to be done to validate	What is to be done to certify	What is to be done to deploy

I call this my tapestry of WHATs.

Requirements

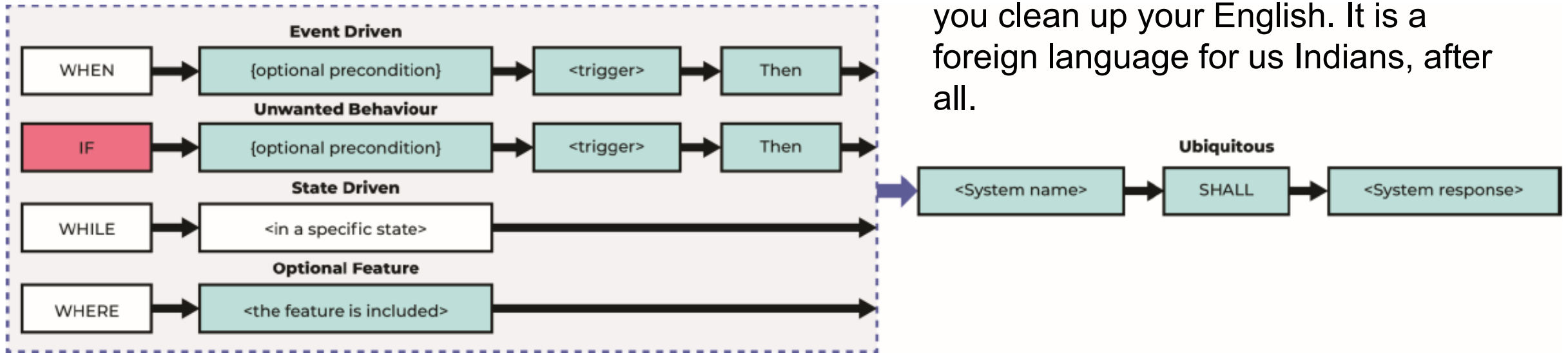
- A set of stakeholder requirements are clarified and translated from statements of need into engineering-oriented language in order to enable proper architecture definition, design, and verification activities that are needed as the basis for system requirements analysis.
- The system requirements are based around identification and synthesis of the functions required of any solution system associated with performance and other quality measures and provide the basis for the assessment of candidate solutions and verification of the completed system. The system requirements are expressed in technical language that is useful for architecture and design: unambiguous, consistent, coherent, exhaustive, and verifiable.
- **Make a list of all verbs associated with the system. Tell WHAT they do. Simple, isn't it? Do not shy away from using LLMs.**

EARS

- Easy Approach to Requirements Syntax

- <http://www.alistairmavin.com/ears/>

LLMs understand EARS. Try and see. But a caution - you should know what the requirement is. Use LLMs to help you clean up your English. It is a foreign language for us Indians, after all.

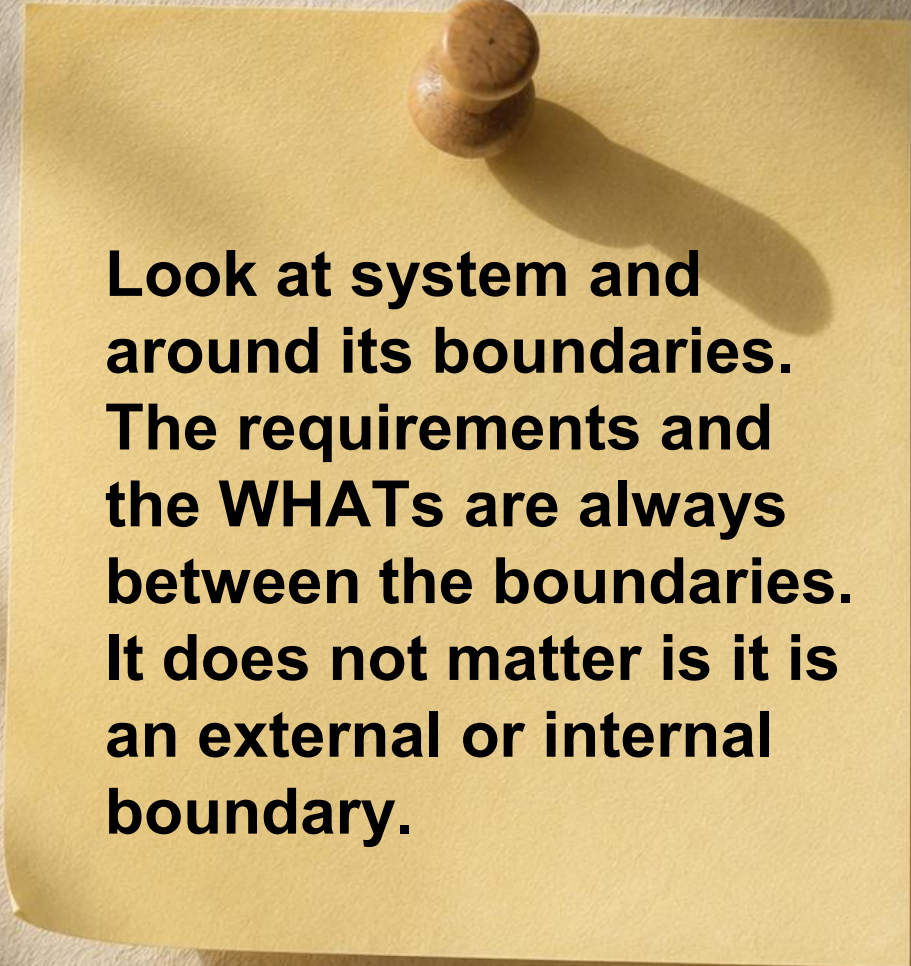


MIT Ventilator requirements

Thank U ChatGPT!

- **Respiratory Rate (RR)**
 - The system shall allow the respiratory rate to be set between 6 and 40 breaths per minute.
 - While in *Assist Control mode*, the system shall allow respiratory rates of 6–9 breaths per minute.
- **Tidal Volume (TV)**
 - The system shall allow the tidal volume to be set between 200 and 800 mL, based on patient weight.
- **I/E Ratio**
 - The system shall allow the inspiratory-expiratory ratio to be adjusted between 1:1 and 1:4.
 - The system shall default the inspiratory-expiratory ratio to 1:2 at startup.
- **Assist Control Trigger Sensitivity**
 - When a patient attempts to inspire, the system shall detect a pressure drop between 2 and 7 cm H₂O relative to PEEP and trigger a ventilator breath.

Requiring



Look at system and around its boundaries. The requirements and the WHATs are always between the boundaries. It does not matter is it is an external or internal boundary.

In early June the world of leaf and blade and flowers explodes.

JUNE

2026 CALENDAR

SUN	MON	TUE	WED	THU	FRI	SAT
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Define your Safety



Safety

- Safety - is the state of being "safe", the condition of being protected from harm or other non-desirable outcomes. Safety can also refer to the control of recognized hazards in order to achieve an acceptable level of risk.
- Safety is a perception!



System Safety

- In the most general sense, safety is freedom from harm. As an engineering discipline, system safety is concerned with minimizing hazards that can result in a mishap with an expected severity and with a predicted probability. These events can occur in elements of life-critical systems as well as other system elements.
- System Safety is the application of engineering and management principles, criteria, and techniques to achieve acceptable risk, within the constraints of operational effectiveness and suitability, time, and cost, throughout all phases of the system life cycle. - MIL-STD-882E
- **There is lot of work to be done here. Do NOT ignore this step!**

Hazards and risks

- <https://emergency-vent.mit.edu/resources/safety-first/>

Risk

Mitigation

Patient circuit disconnect	Pressure sensor identifies loss of pressure; Alarm Convert to manual bagging; Correct disconnect
Bag rupture	Pressure sensor identifies loss of pressure; Alarm Convert to manual bagging; Replace bag
Oxygen leak	This risk is native to the manual resuscitation bag and is addressed by the bag instructions. The bag is exposed and any leaked oxygen will dissipate
EMI emission motor	This is a low risk, should it be an issue a metal faraday cage can be placed around the motor (Conduct formal testing if required) Board traces have been minimized and grounded
EMI interference with controller	There are no high frequency signals susceptible to interference; DC simple analog and digital channels Controller has been used in an OR environment and no interference noticed

Systems have a structure

- If we control the hazards, we can make the system safer. Controlling the hazard is done by set of function that can do this. Monitoring, Tripping, Regulating ... are some of the functions that can help safety.
- Functions interact with other functions. You soon realize that your system functions are hierarchal. Monitoring indicates to the Regulating to do something. Regulating finds that it cannot regulate so it asks Tripping to cut off and make the system safe. **Understand this hierarchy in systems.** (Note the verbs again here.)
- Systems have **feedback** and **delays** and **gains**. These, cause the **negative emergent behavior** that makes the system unsafe. **UNDERSTAND** this in context of your system in your defined boundary. Make your system safe!

System Theoretic Process Analysis

Partnership for Systems Approaches to Safety and Security (PSASS)



[What's new?](#) [Home](#) [Research](#) [Materials](#) [People](#) [STAMP Workshops](#) [Contact](#) [STAMP Tools](#)

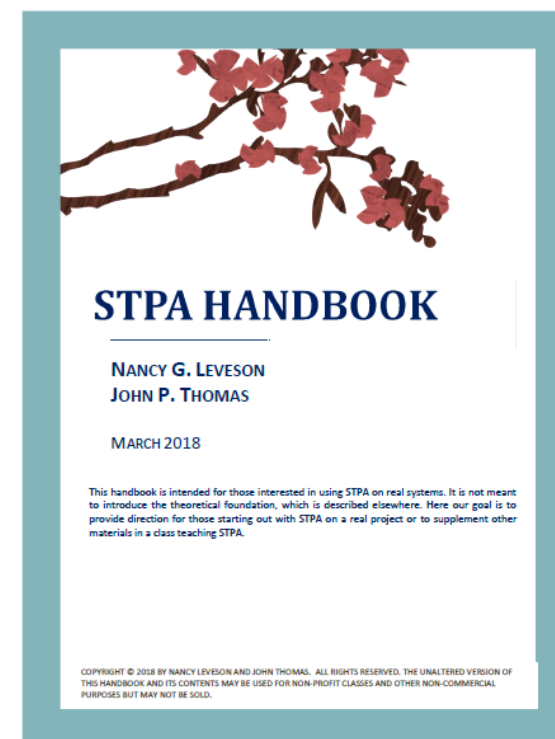
Home

STPA Handbook is now available

- [Download English version](#)
- [Download Japanese version](#) (translated by JAXA)

Site Map

- [What's new?](#)
- [Home](#)
- [Research](#)
- [Materials \(Handbook, Papers, Etc.\)](#)
- [People](#)
 - [PSASS Structure](#)



<http://psas.scripts.mit.edu/home/>

STPA tool



Products Solutions Academia Support **Community** Events

Get MATLAB



File Exchange

Search bar with 'File Exchange' dropdown and search icon

MATLAB Central ▾ | Files | Authors | My File Exchange | Contribute | About

Trial software



STPA Safety Analysis Tool in Simulink

version 1.0.2 (49.7 KB) by [Aditya Jeppu](#)

This is a STPA (Systems Theoretic Process Analysis) tool using Simulink and Matlab.

★★★★★ 1 Rating

12 Downloads ⓘ

Updated 02 Dec 2019

[View License](#)

+ Follow

Download

Overview

Functions

Models

STPA (http://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf) is a safety analysis method developed at MIT, Massachusetts. I have made a simple Simulink library and provide two Matlab files to analyze the model. The details of the method are available in the handbook. The library has blocks for controller, actuator, process, losses, hazards and control actions. The user can make a hierarchical control structure (HCS) using these building blocks. The Matlab files read the HCS and generate a report that can be used for further analysis. This is a preliminary work and suggestions are welcome for updates. I have named it MOOSE : Matlab Tool for STPA Evaluation.

Requires

[MATLAB](#)

[Simulink](#)

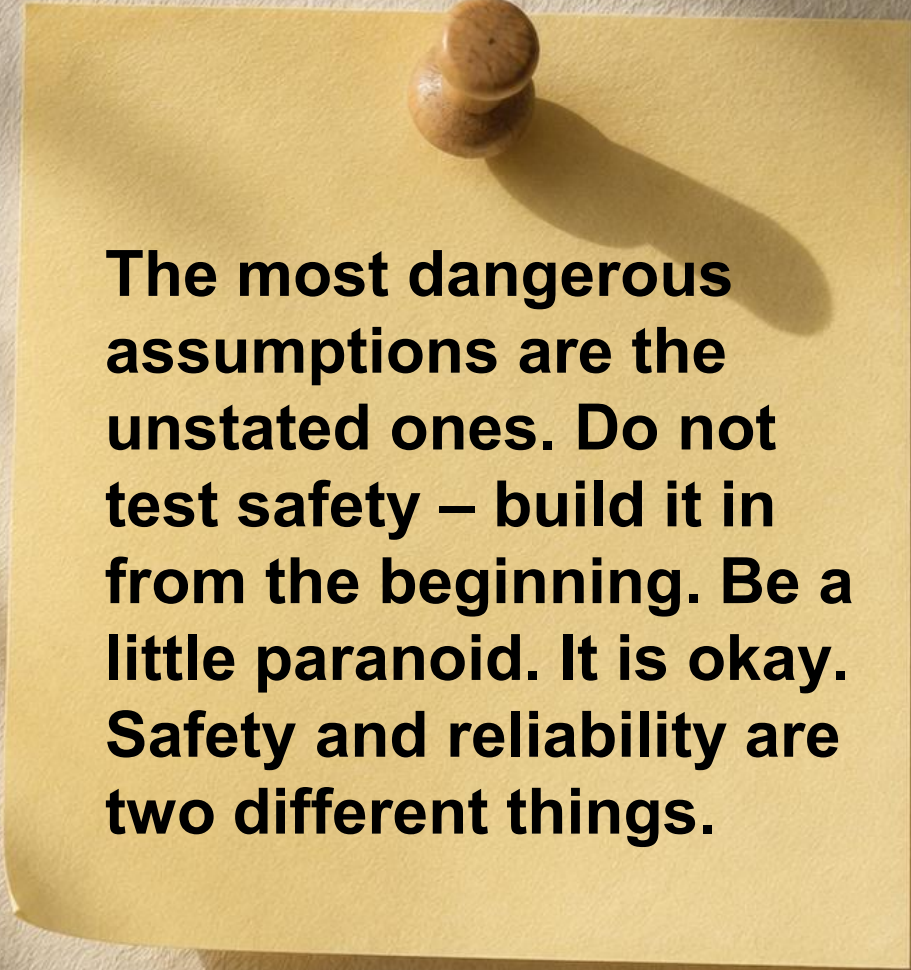
MATLAB Release Compatibility

Created with R2019a

Aditya Jeppu (2019). STPA Safety Analysis Tool in Simulink

(<https://www.mathworks.com/matlabcentral/fileexchange/73434-stpa-safety-analysis-tool-in-simulink>), MATLAB Central File Exchange. Retrieved December 17, 2019.

Proofing



The most dangerous assumptions are the unstated ones. Do not test safety – build it in from the beginning. Be a little paranoid. It is okay. Safety and reliability are two different things.

Summer afternoon — summer afternoon.

JULY

2026 CALENDAR

SUN	MON	TUE	WED	THU	FRI	SAT
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

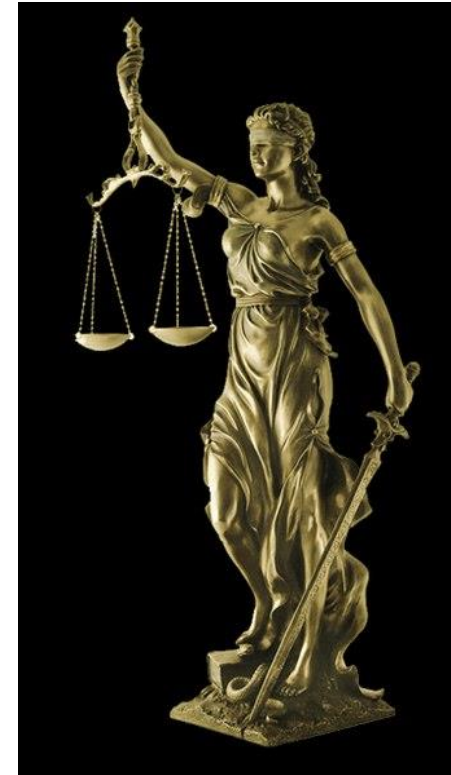
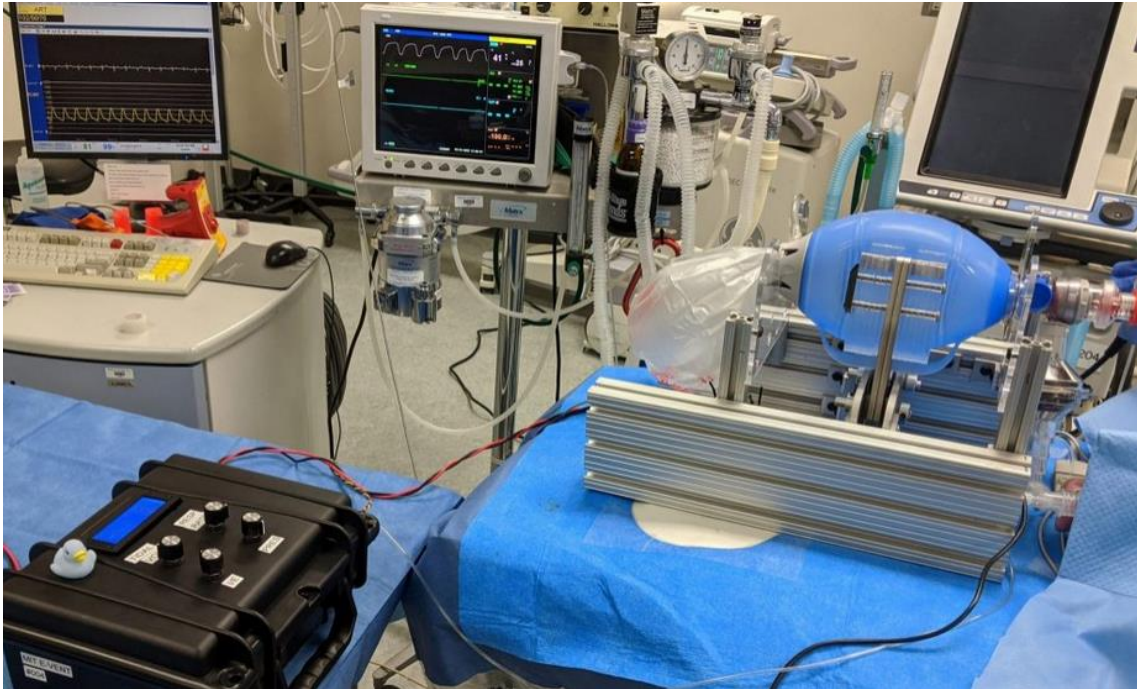
Define your Certification



Let this
not be a
surprise!

The two realms

Engineering



Law

Understand this – You write papers when your product succeeds. Lawyers write papers when it fails.

Standards and Certification

- **A standard is an agreed upon, repeatable way of doing something.** It is a published document that contains a technical specification or other precise criteria designed to be used consistently as a rule, guideline, or definition. Standards help to make life simpler and to increase the reliability and the effectiveness of many goods and services we use. *Standards are created by bringing together the experience and expertise of all interested parties, such as the producers, sellers, buyers, users, and regulators of a particular material, product, process, or service.*

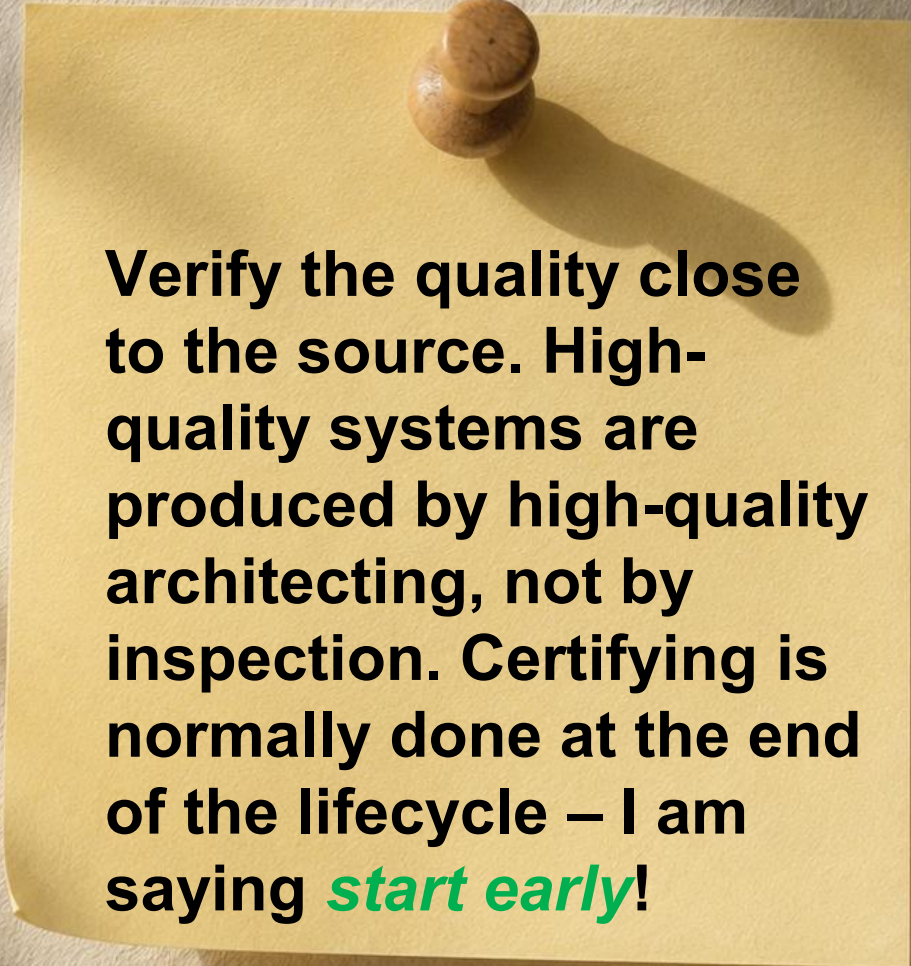
Standards and Certification

- *Certification* refers to the issuing of **written assurance** (the certificate) by an independent external body that it has audited a management system and verified that it conforms to the requirements specified in the standard.
- Typically, other more specific systems engineering standards are not the subject of certification. They are self-imposed in order to improve uniformity of organization and enterprise operations or to improve the quality of products and services. Alternatively, they may be dictated by legislation, policy, or as part of a formal agreement between an acquirer and a supplier.

Medical standards

Standard	Name
ISO 13485	<i>Medical devices — Quality management systems — Requirements for regulatory purposes</i>
ISO 14971	<i>Medical devices — Application of risk management to medical devices</i>
IEC 62304	<i>Medical device software — Software life cycle processes</i>
ISO 62366-1	<i>Medical devices - Part 1: Application of usability engineering to medical devices</i>
ISO 15223-1	<i>Medical devices — Symbols to be used with information to be supplied by the manufacturer — Part 1: General requirements</i>
ISO 14155	<i>Clinical investigation of medical devices for human subjects — Good clinical practice</i>
ISO/TR 24971	<i>Medical devices — Guidance on the application of ISO 14971</i>
ISO 16571	<i>Systems for evacuation of plume generated by medical devices</i>
IEC 80001-1	<i>Safety, effectiveness and security in the implementation and use for connected medical devices or connected health software — Part 1: Application of risk management</i>
IEC/TR 80002-1	<i>Medical device software — Part 1: Guidance on the application of ISO 14971 to medical device software</i>
IEC/TR 80002-2	<i>Medical device software — Part 2: Validation of software for medical device quality systems</i>
IEC/TR 80002-3	<i>Medical device software — Part 3: Process reference model of medical device software life cycle processes (IEC 62304)</i>

Certifying



Verify the quality close to the source. High-quality systems are produced by high-quality architecting, not by inspection. Certifying is normally done at the end of the lifecycle – I am saying *start early!*

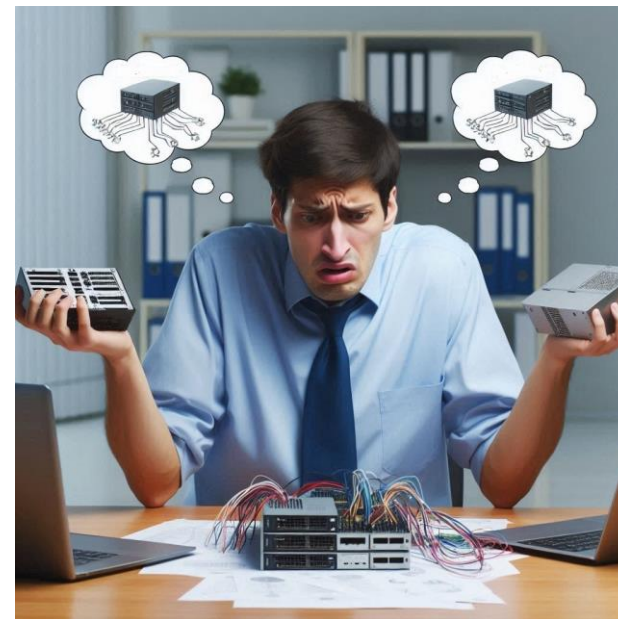
Everything good, everything magical happens between June and August.

AUGUST

2026
CALENDAR

SUN	MON	TUE	WED	THU	FRI	SAT
30	31					1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

Define your Architecture



There are many choices here to satisfy your requirements. This always surprises you.

Architecture Types

- Functional Architecture

- The functional architecture is performed mainly at the start of the systems engineering project effort. It describes the functions, or actions, that must be performed while completing the system's objective or mission. This typically will be **system or solution agnostic**, and will rather focus on **describing the verbs** (e.g. functions) that will describe what needs to be done. These functions may then be organized by mission phase, and also separated by the responsible organization/system/person that will be performing these functions, often called "swimlanes." The linkage between the functions is often what type of information or data is being exchanged between the functions, and is often a noun.

Architecture Types

- Physical Architecture
 - The next phase is the physical architecture, where the actual system components and subsystems are developed. These are typically in a physical block diagram that shows the actual components that make up the system. For this purpose, this is the actual system. The components may also be grouped together that make up the subsystem. The subsystems may also be grouped together to make up the system. The interfaces between the components and subsystems are the physical interfaces, such as wiring, internet connections, Ethernet connections, or wireless connections or energy. **Make a of nouns associated with the verbs.**

Architecture Types

- Allocated Architecture
 - This is where the functions and physical architectures merge, in order to determine if we have all the functions and physical components properly allocated. This is a simple accounting of which functions are performed by which components within the system. An ideal allocation is one function is performed only by one component. However, in complex systems, this may not be the case, as we would have multi-function components, where one component may perform more than one function. **These are like sentences. Associating Nouns and Verbs together.**

Architectural Views

- Operational View
 - This representation is from the users', or operators', perspective. This view would include products that address operational system phases, scenarios, and task flows. Information flow from the users' perspectives might also be addressed. User interfaces would also be described. Example products that might be included in this view would be operational figures or graphics, scenario descriptions (including use cases), task flow diagrams, organization charts, and information flow diagrams.

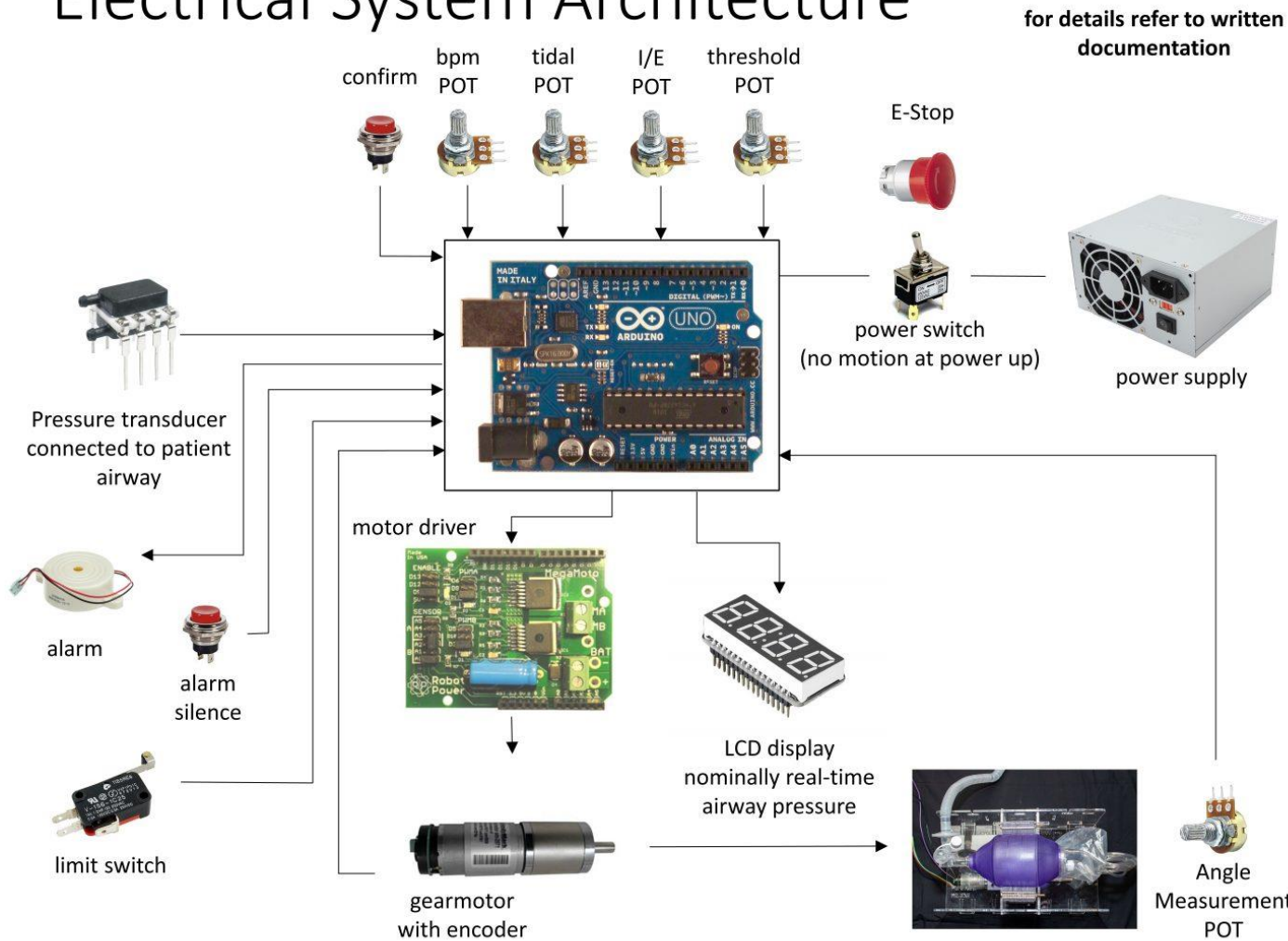
Architectural Views

- Logical View
 - This representation is from the manager's or customer's perspective. The logical view would include products that define the system's boundary with its environment and the functional interfaces with external systems, major system functions and behaviors, data flow, internal and external data sets, internal and external users, and internal functional interfaces. Example products for this view would be functional flow block diagrams, context diagrams.

Architectural Views

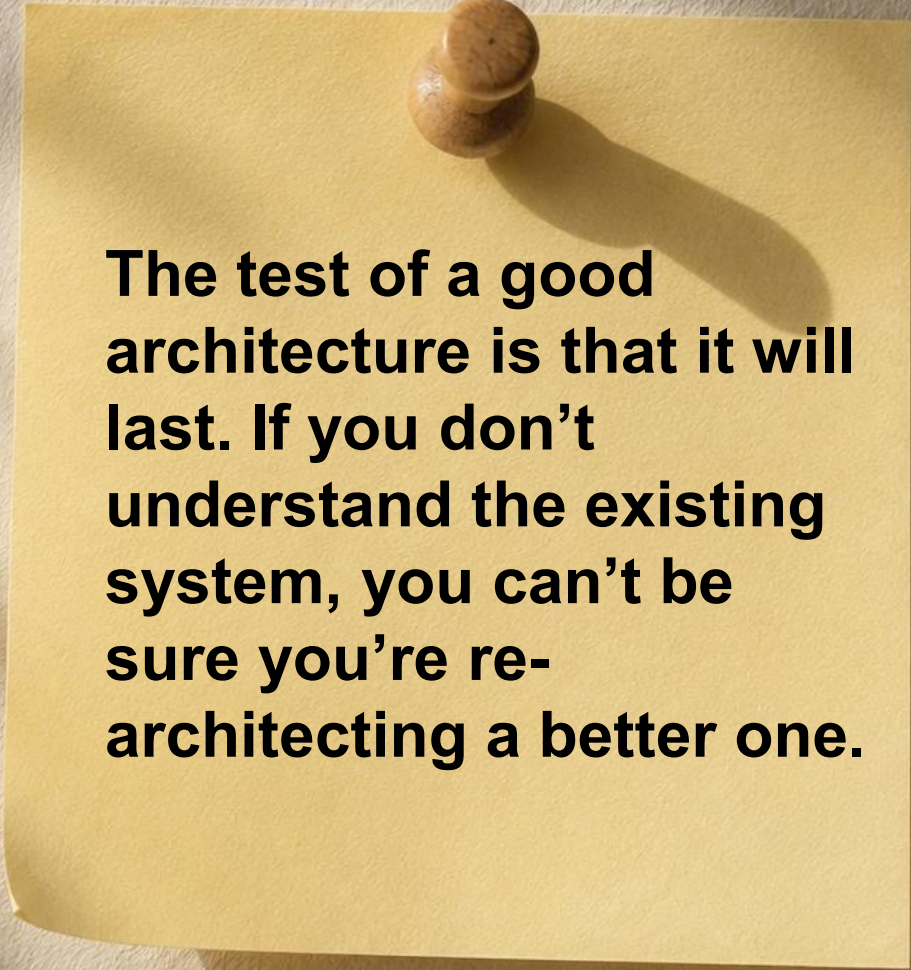
- Physical View
 - This representation is from the designers' perspective. This view would include products that define the physical system boundary, the system's physical components and how they interface and interact together, the internal databases and data structures, the IT infrastructure of the system and the external IT infrastructure with which the system interfaces, and the standards in force in its development. Example products include physical block diagrams down to a fairly high level of detail, database topologies, interface control documents (ICDs), and standards.

Electrical System Architecture



<https://emergency-vent.mit.edu/controls/electrical-hardware/>

Architecting



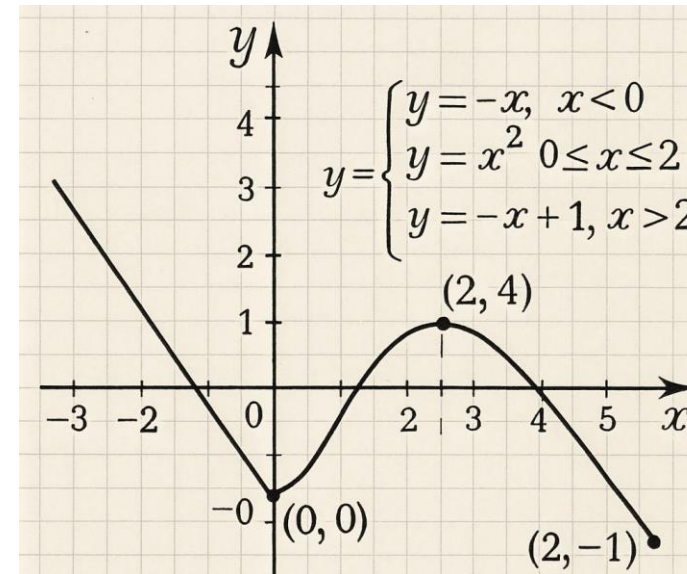
The test of a good architecture is that it will last. If you don't understand the existing system, you can't be sure you're re-architecting a better one.

And then the sun took a step back, the leaves lulled themselves to sleep.

SEPTEMBER 2026 CALENDAR

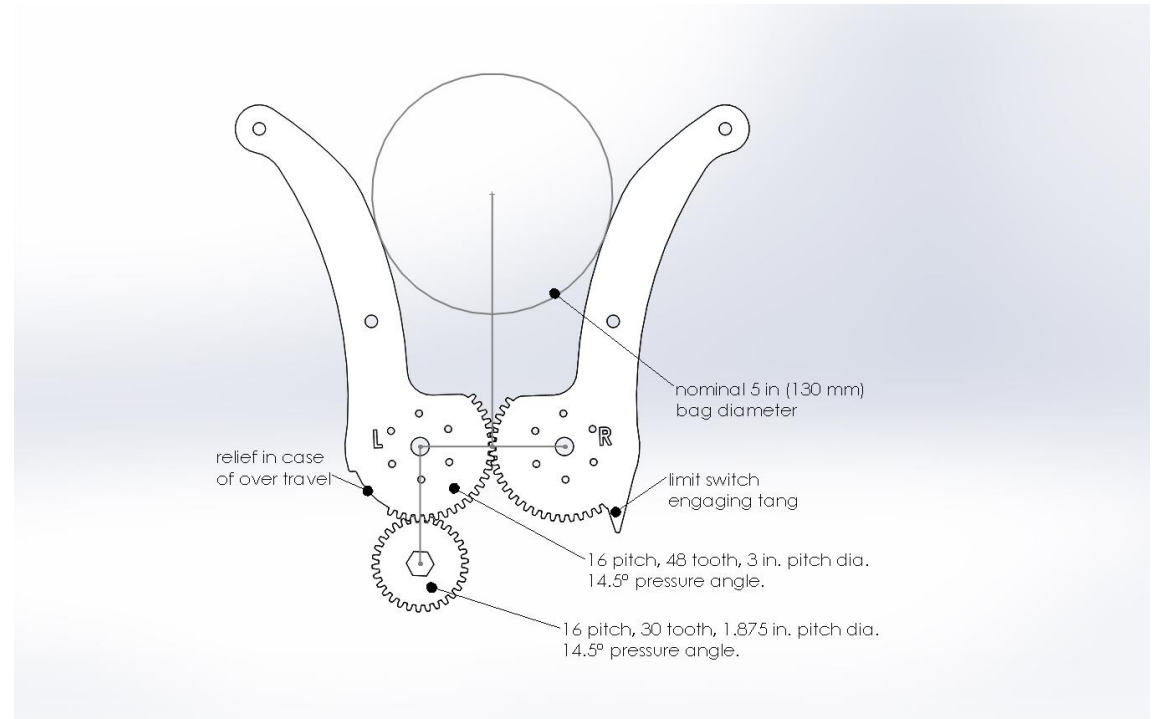
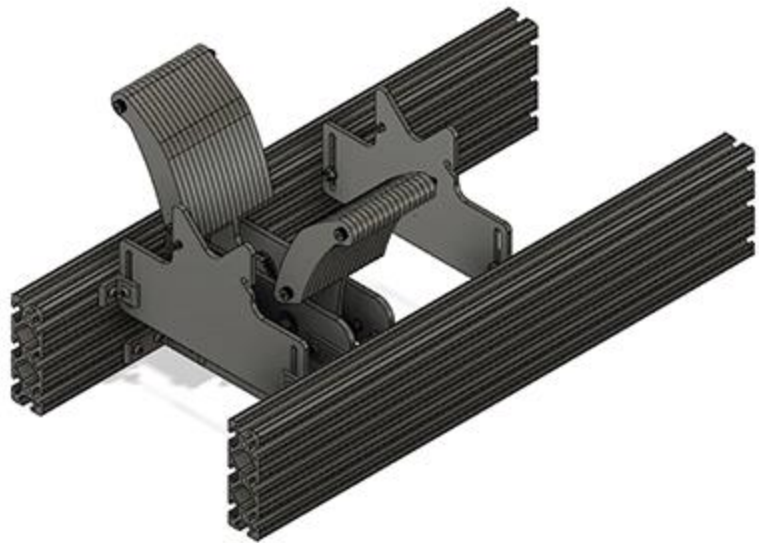
SUN	MON	TUE	WED	THU	FRI	SAT
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Define your Optimized Design



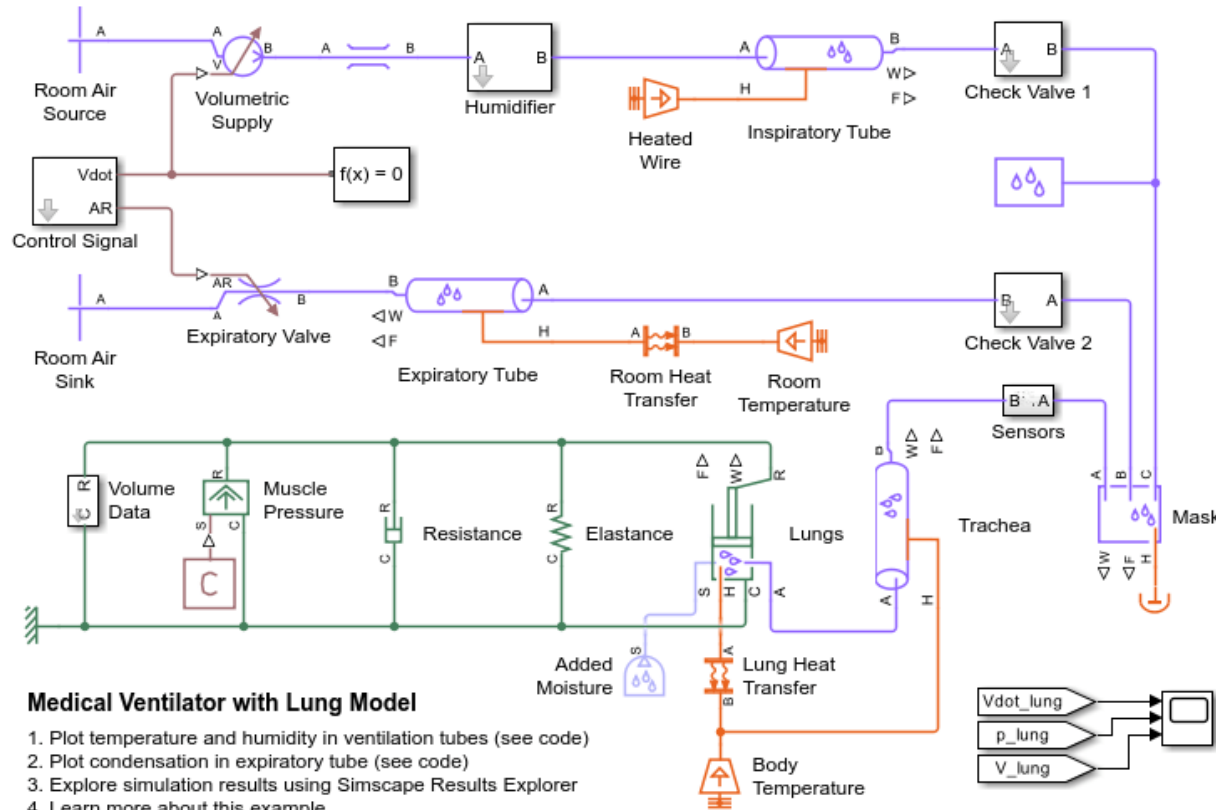
Design of Experiments, Monte Carlo Simulation help a lot here. Use them well.

Design



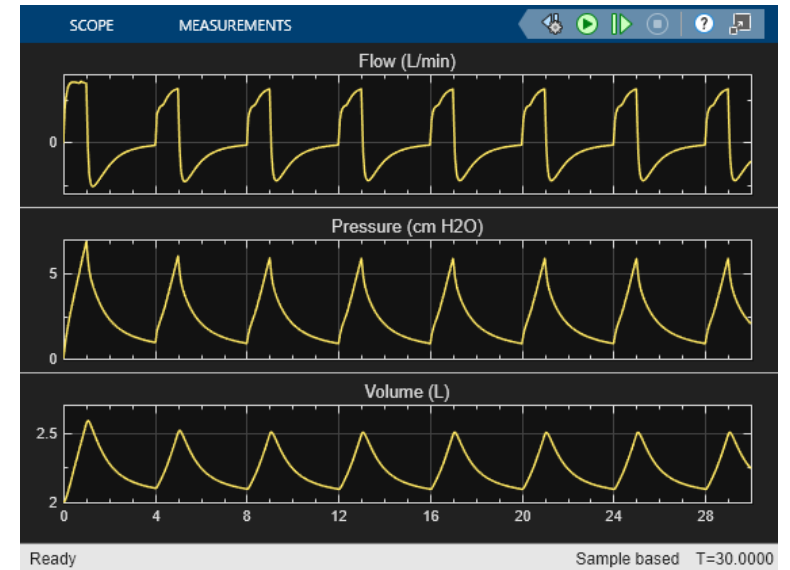
<https://emergency-vent.mit.edu/mechanical/>

Design



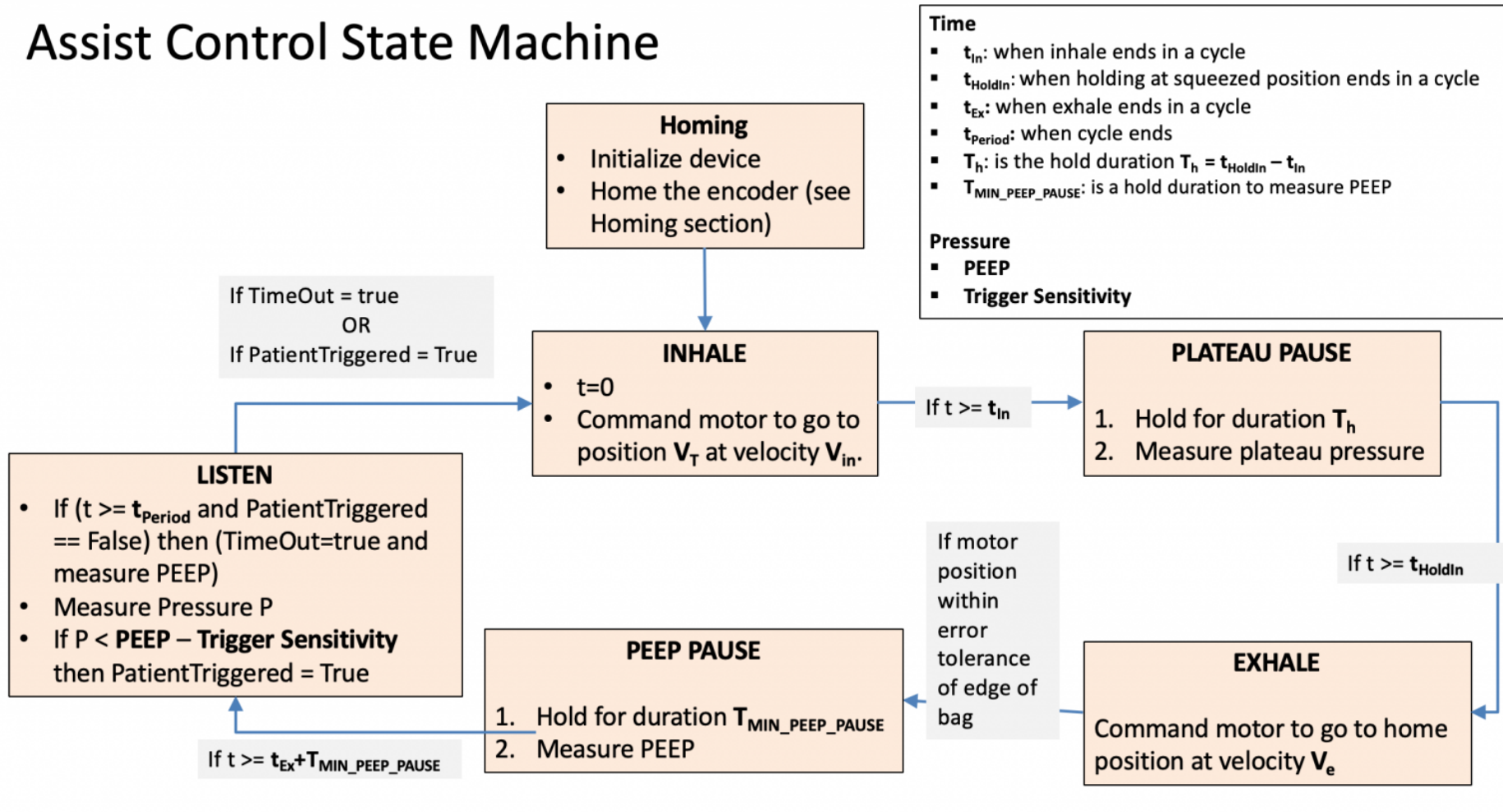
Copyright 2017-2022 The MathWorks, Inc.

<https://www.mathworks.com/help/simscape/ug/medical-ventilator-with-lung-model.html>

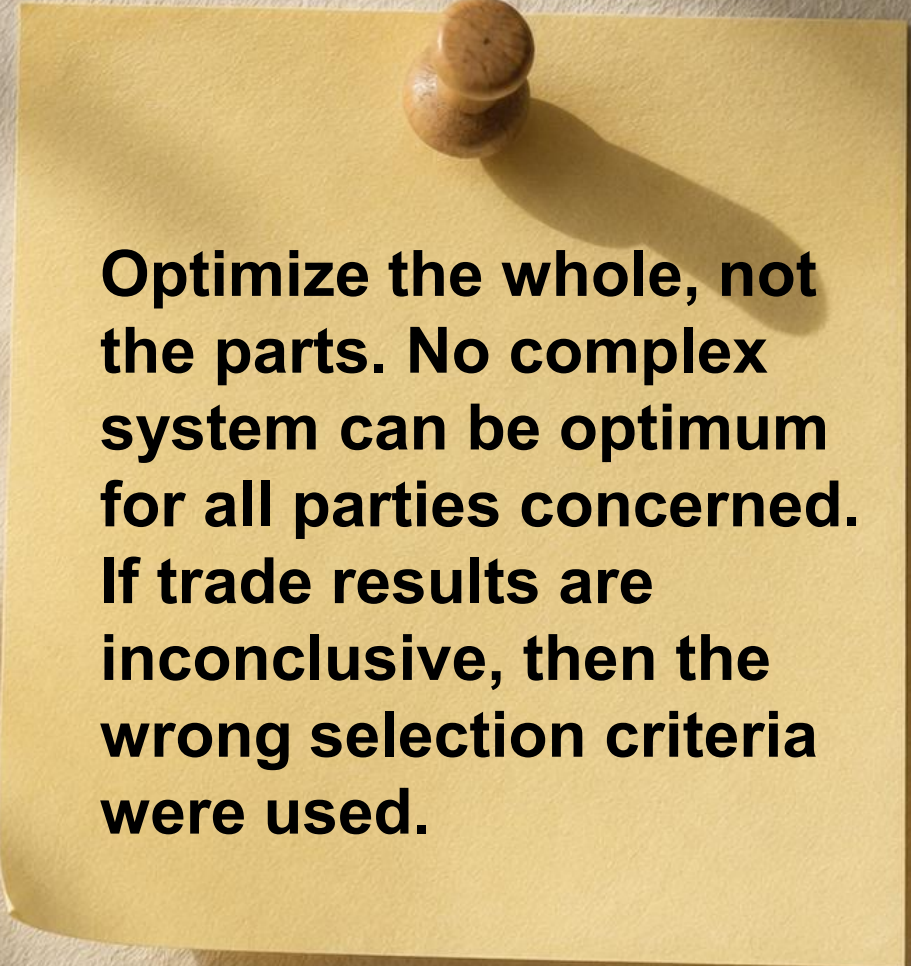


Design

Assist Control State Machine



Optimizing



Optimize the whole, not the parts. No complex system can be optimum for all parties concerned. If trade results are inconclusive, then the wrong selection criteria were used.

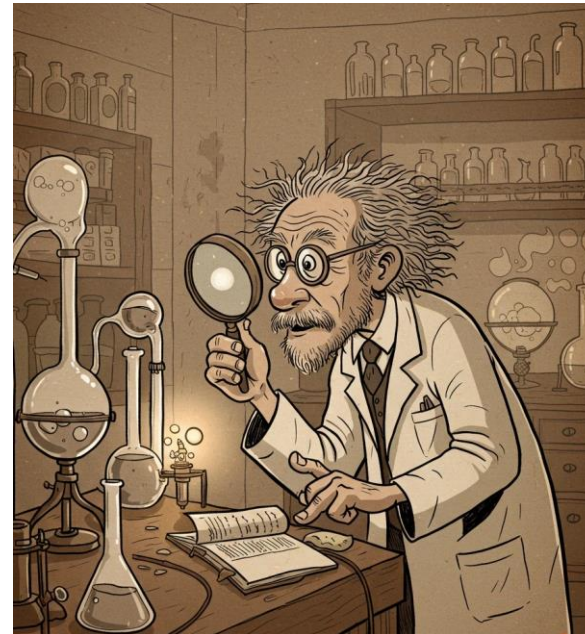
I'm so glad I live in a world where there are Octobers.

OCTOBER

2026
CALENDAR

SUN	MON	TUE	WED	THU	FRI	SAT
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Define your Tests



Testing is very important. Check out my Just Enough on formal methods also in my webpage.

Interface Errors

We always find interface errors during testing. Normally 70% Requirements errors!! Food for thought.

- Construction
- Inadequate functionality
- Location of functionality
- Changes in functionality
- Added functionality
- Misuse of interface
- Misunderstanding of interface
- Data structure alteration
- Inadequate error processing
- Additions to error processing
- Inadequate post-processing
- Inadequate interface support
- Initialization/value errors
- Validation of data constraints
- Timing/performance problems
- Coordination change
- Hardware/software interfaces

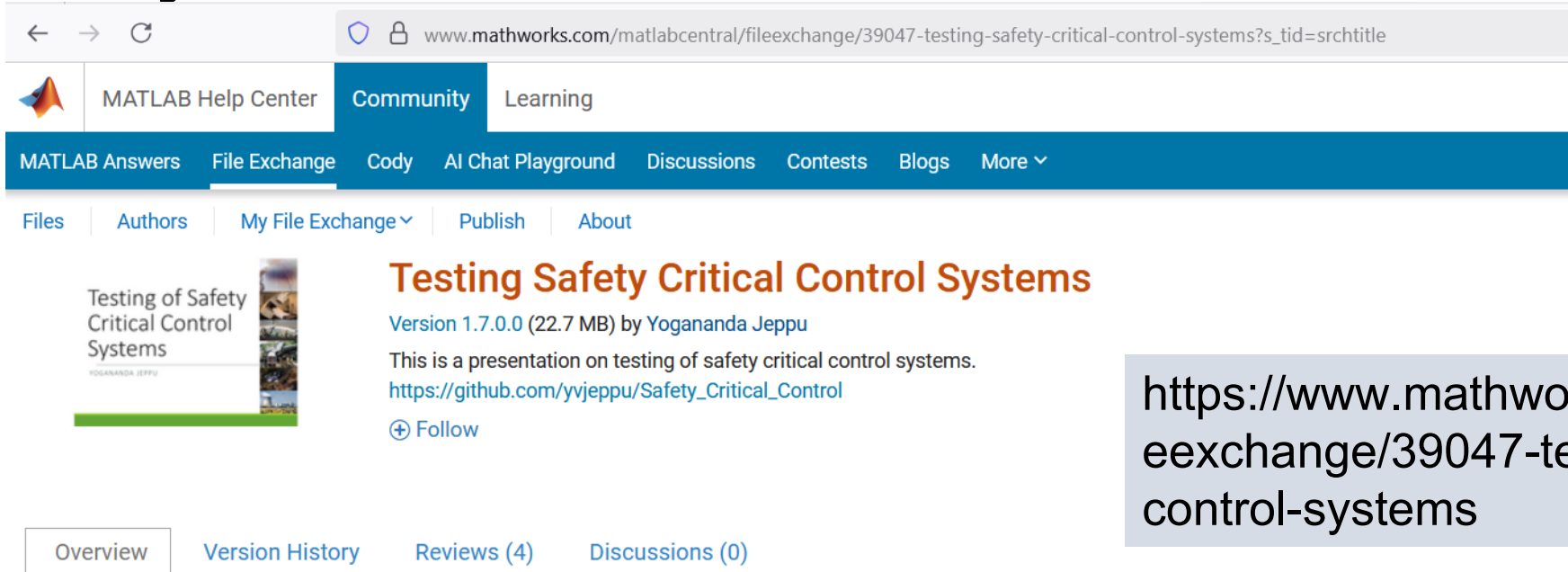
Testing software

- Software has to be tested based on requirements. Both high level requirements and low-level design requirements are tested.
- Code coverage is very important here to understand that there are no code components available that do not lead to any functionality.
- If models are used to generate code it is important to get the model coverage also.
- What do you mean by model coverage? Is it similar to code coverage or do you test the functionality of the model block?

Testing software

- All the states need to be covered. State transitions are the most dangerous as they define the system behavior through software.
- Simple mistakes in priority can lead to a different behavior.
- I have found cases where we had just a one sample error. We would have given up on it as spurious but this one frame error turned out to be major design flaw.
- **What should be my pass-fail criteria?**
- **Automate your tests as much as possible.**

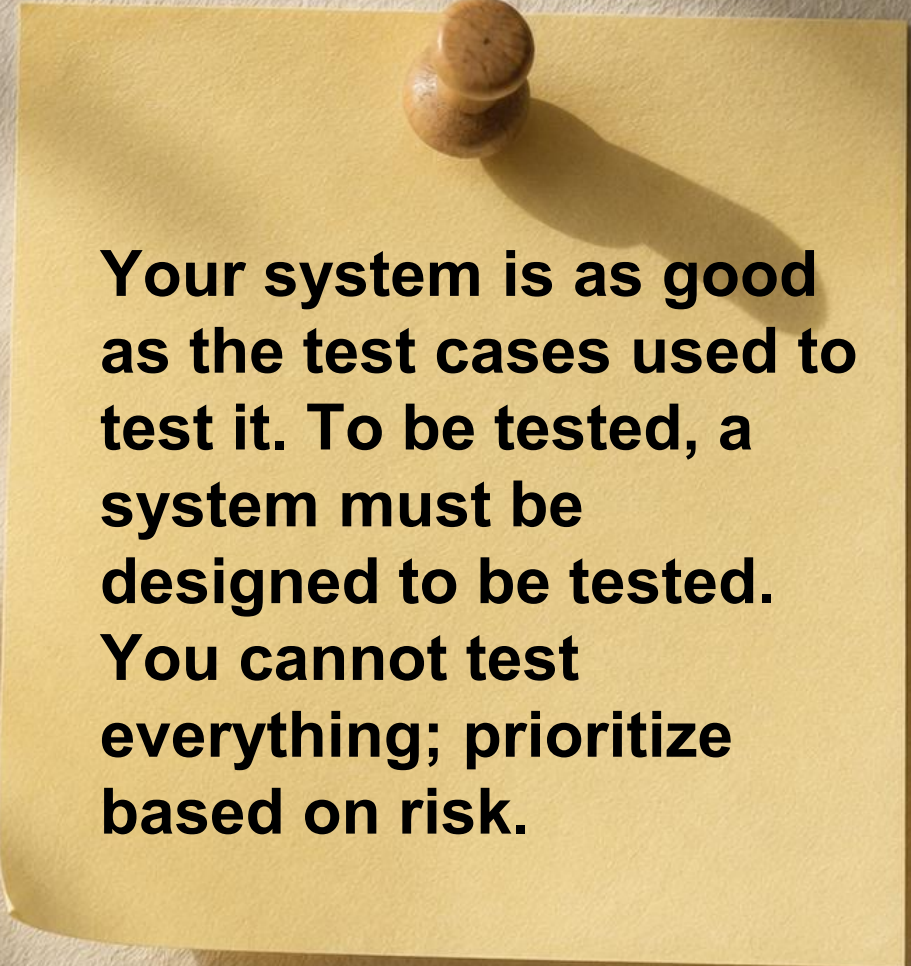
My contributions



The screenshot shows the MATLAB Central File Exchange interface. The browser address bar displays the URL: www.mathworks.com/matlabcentral/fileexchange/39047-testing-safety-critical-control-systems?s_tid=srchtitle. The navigation menu includes 'MATLAB Help Center', 'Community', and 'Learning'. Below this, there are links for 'MATLAB Answers', 'File Exchange', 'Cody', 'AI Chat Playground', 'Discussions', 'Contests', 'Blogs', and 'More'. The main content area features a file card for 'Testing Safety Critical Control Systems' by Yogananda Jeppu. The card includes a thumbnail image, the title, version information (1.7.0.0, 22.7 MB), a description, a GitHub link, and a 'Follow' button. Below the card are tabs for 'Overview', 'Version History', 'Reviews (4)', and 'Discussions (0)'. A note states: 'Editor's Note: This file was selected as MATLAB Central Pick of the Week'. A description below the note reads: 'This has a presentation on testing Safety Critical Control Systems and a brief introduction to Formal Methods'. At the bottom, there is a 'Cite As' section with a citation: 'Yogananda Jeppu (2025). Testing Safety Critical Control Systems (https://github.com/yvjeppu/Safety_Critical_Control), GitHub. Retrieved August 27, 2025.'

<https://www.mathworks.com/matlabcentral/fileexchange/39047-testing-safety-critical-control-systems>

Testing



Your system is as good as the test cases used to test it. To be tested, a system must be designed to be tested. You cannot test everything; prioritize based on risk.

November comes, and November goes.

NOVEMBER

2026
CALENDAR

SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Deploy



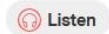
Deployment

All your 10 months of hard work is going to be fructified! Enjoy this moment. Update your resumes with this success story. Get ready of your appraisal.

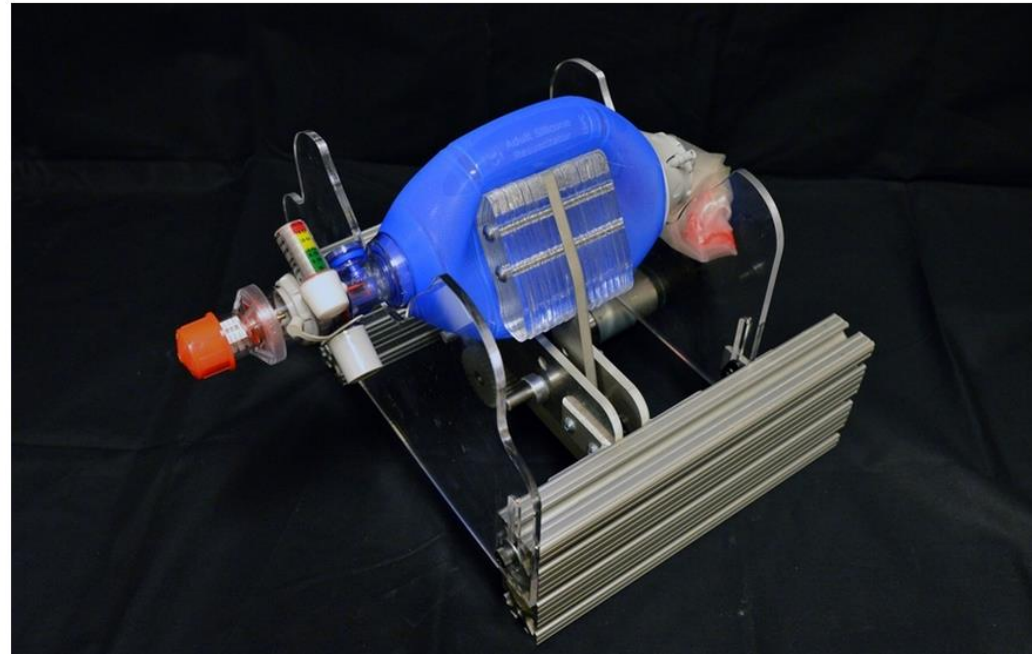


MIT-based team works on rapid deployment of open-source, low-cost ventilator

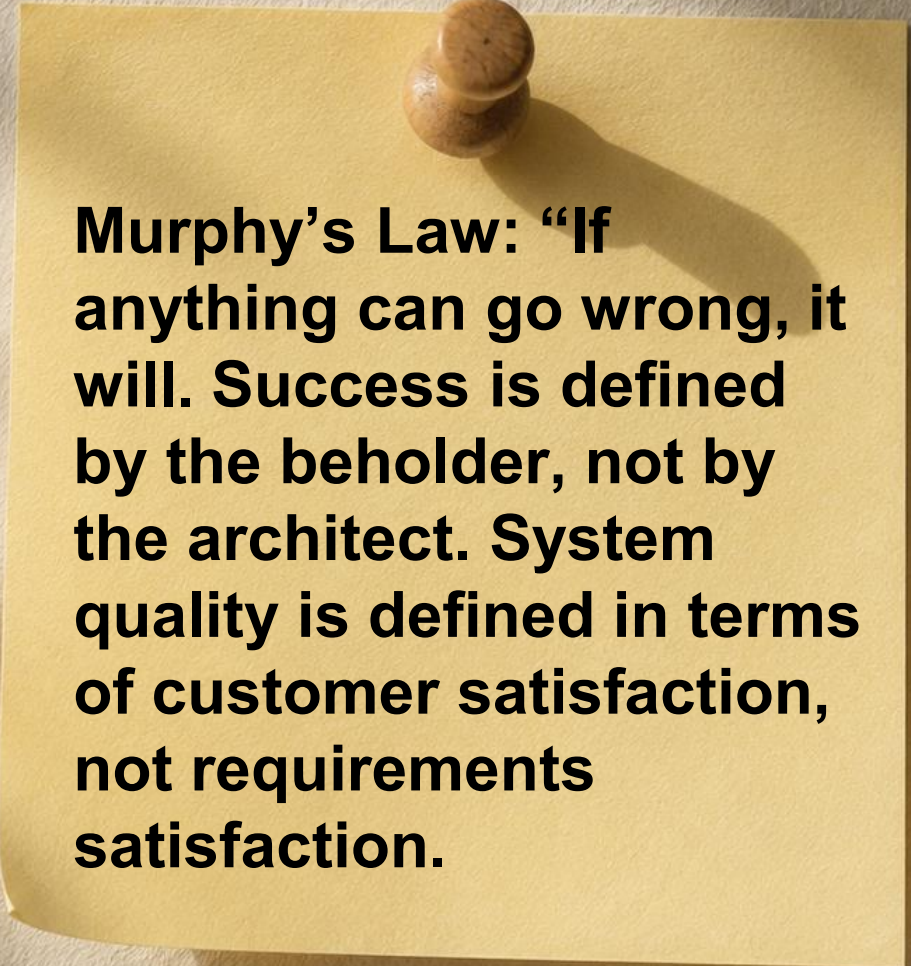
Clinical and design considerations will be published online; goal is to support rapid scale-up of device production to alleviate hospital shortages.



David L. Chandler | MIT News Office
March 26, 2020



Deploying



Murphy's Law: "If anything can go wrong, it will. Success is defined by the beholder, not by the architect. System quality is defined in terms of customer satisfaction, not requirements satisfaction."

Winter is the time for comfort, for good food and warmth.

DECEMBER

2026
CALENDAR

SUN	MON	TUE	WED	THU	FRI	SAT
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Maintain and Sustain



Systems operate for very long durations. Voyager 1 was launched by NASA on September 5, 1977. It experienced revival again in 2025.

Ventilator Management

Tools & Reference > Clinical Procedures

Ventilator Management

Updated: Apr 07, 2020 | Author: Allon Amitai, MD; Chief Editor: Zab Mosenifar, MD, FACP, FCCP [more...](#)



The use of AI in system necessitates data collection and model updates continuously. Plan for it and do it diligently. That is a perpetual calendar activity.

• Introduction to Ventilator Management

Modes of Mechanical Ventilation

Methods of Ventilatory Support

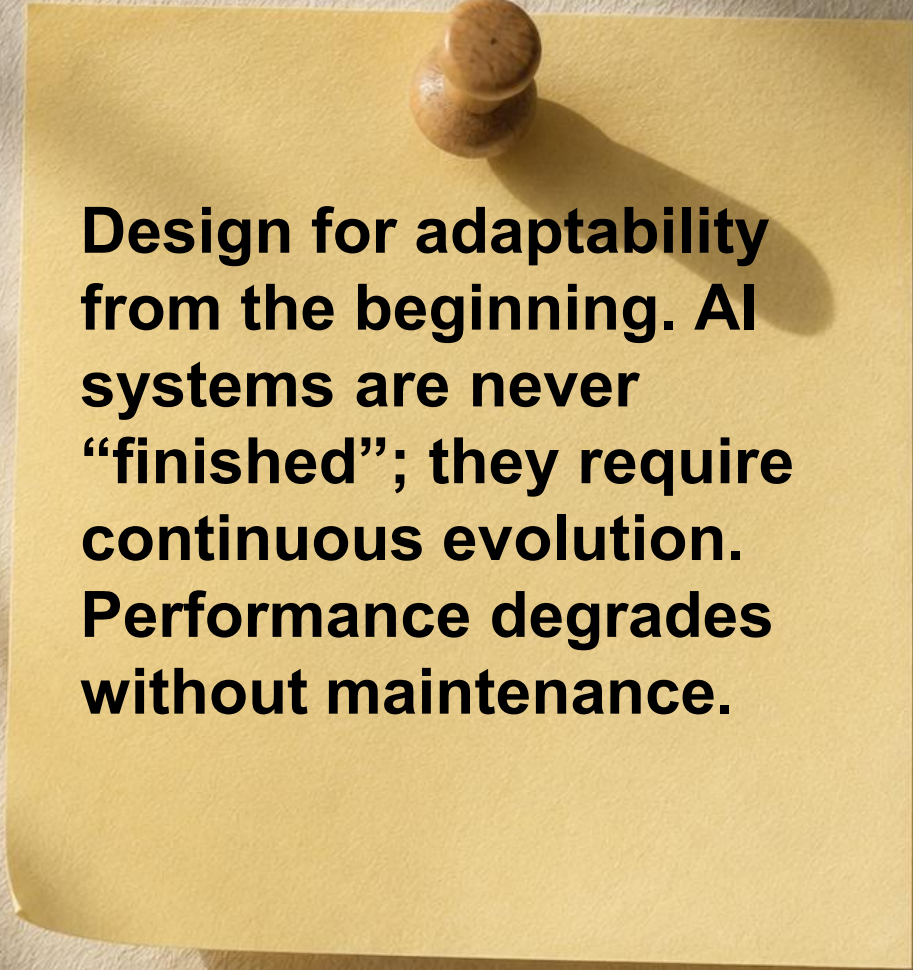
Introduction to Ventilator Management

Intubation, with subsequent mechanical ventilation, is a common life-saving intervention in the emergency department (ED). Given the increasing length of stay of ventilated patients in EDs, it is necessary for emergency practitioners to have a good understanding of techniques to optimize mechanical ventilation and minimize complications.

Many different strategies of positive-pressure ventilation are available; these are based on various permutations of triggered volume-cycled and pressure-cycled

<https://emedicine.medscape.com/article/810126-overview?form=fpf>

Sustaining



Design for adaptability from the beginning. AI systems are never “finished”; they require continuous evolution. Performance degrades without maintenance.

In Summary

- In the field of safety critical system, the medical, aerospace, nuclear fields are not very different, I think.
- The good thing about systems engineering is that it is domain agnostic. **Domain is needed** to understand the behaviour and the specific design and testing later but the process is the **Vee Model**. Repeated **year after year!** It is a **calendar of events** so to say!
- I have done it for 38 years – again and again, year after year.

Thank you

- Contact details
- Yogananda Jeppu
- yvjeppu@gmail.com

Make use of this calendar. Enjoy it.

