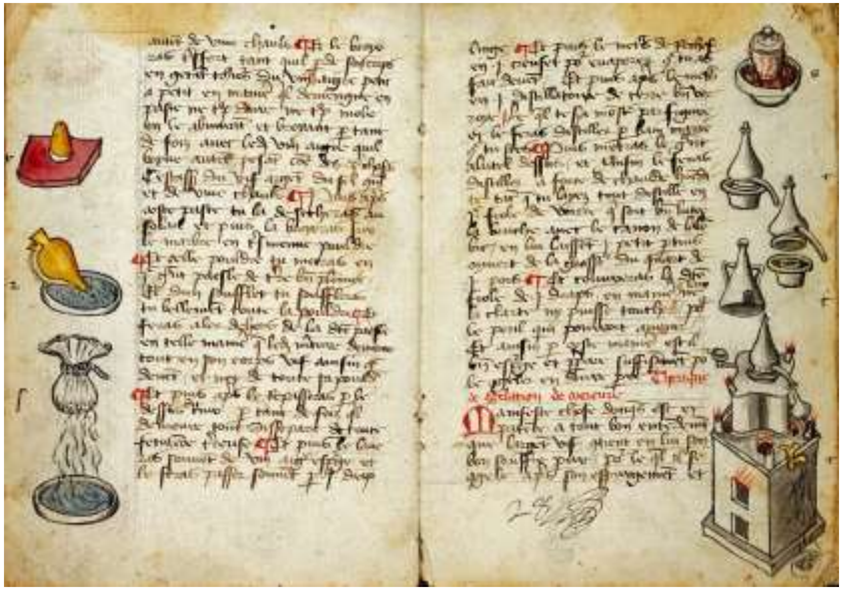
The First Law of Systems Science: Conservation of Complexity

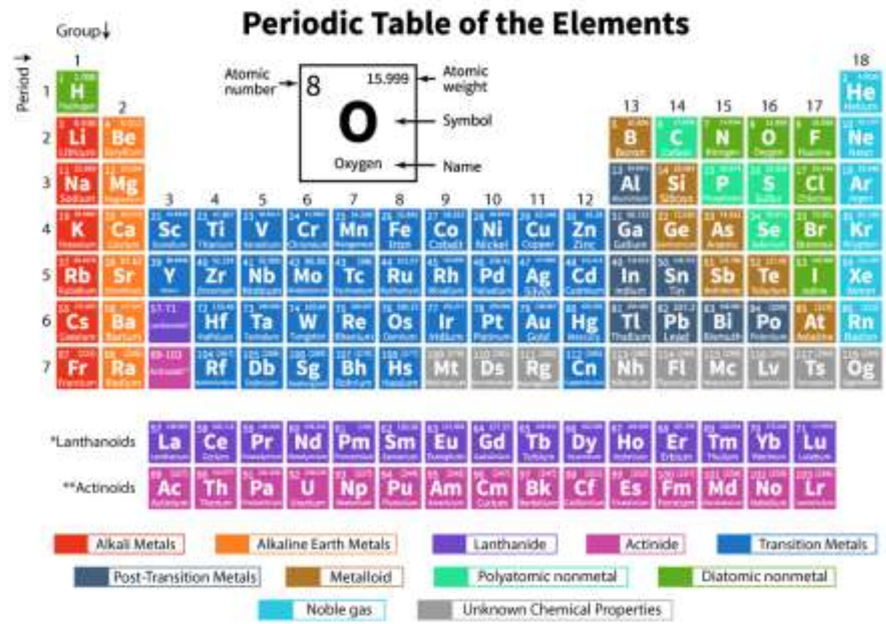
INCOSE International Workshop 2023

Prof. Olivier de Weck
Massachusetts Institute of Technology
deweck@mit.edu

From Alchemy to Chemistry

IDSS
MIT INSTITUTE FOR DATA, SYSTEMS, AND SOCIETY

Book on Alchemy (recipes) – 1600s

Islamic and European alchemists developed a basic set of laboratory techniques, theories, and terms, some of which are still in use today. However, they did not understand the underlying building blocks of matter, still relying on the 4 elements of Greek philosophy.



Periodic Table of Elements – 1800s

In 1817, German physicist Johann Wolfgang Döbereiner began to formulate one of the earliest attempts to classify the elements. In **1829**, he found that he could form some of the elements into groups of three, with the members of each group having related properties. It took 100+ years to fill the table

Alchemy – Chemistry – Chemical Engineering → **300+ Years**

Audience Survey

Go to **www.menti.com** and use the code **2170 5205**

Instructions

Go to

www.menti.com

Enter the code

2170 5205

Or use QR code

Where are we on our Systems Engineering (SE) journey?

- We are in a transition phase between practice (with plenty of heuristics and data) and the beginnings of a deeper theory

- What are the laws that can accurately predict the behavior of complex systems under a set of given assumptions ?

- In order for any “laws” to be accepted as true, there needs to be a set of experiments and data to validate (or falsify) them

Systems Engineering in 2023 is where Chemical Engineering was in 1823 !

Fundamental Laws in Science

- First Law of Thermodynamics
  - Conservation of Energy
  - Rudolf Clausius 1850

$$\Delta U = Q - W.$$

Change In internal state

External stimulus

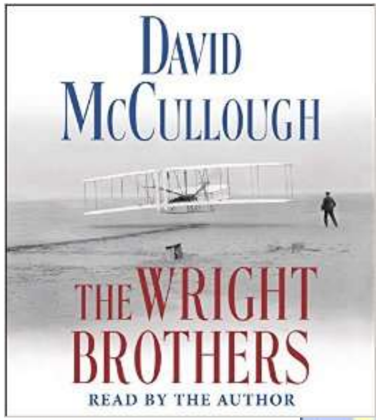
System observable response

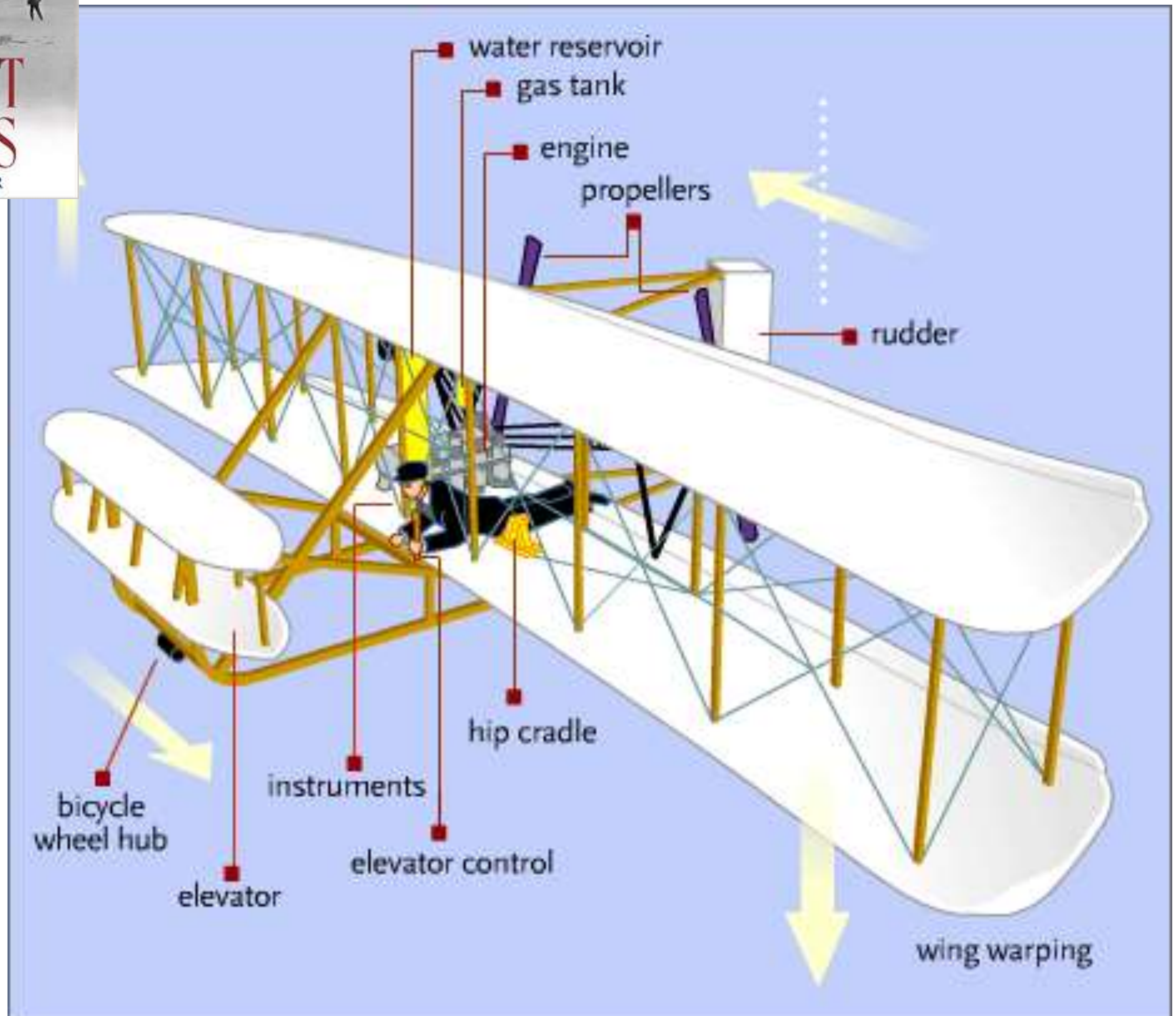
- Second Law of Classical Mechanics
  - Conservation of Angular Momentum
  - Leonhard Euler 1736

$$\dot{\underline{H}} = \underline{T} - \underline{\omega} \times \left[\underline{\underline{I}}\,\underline{\omega} \right]$$

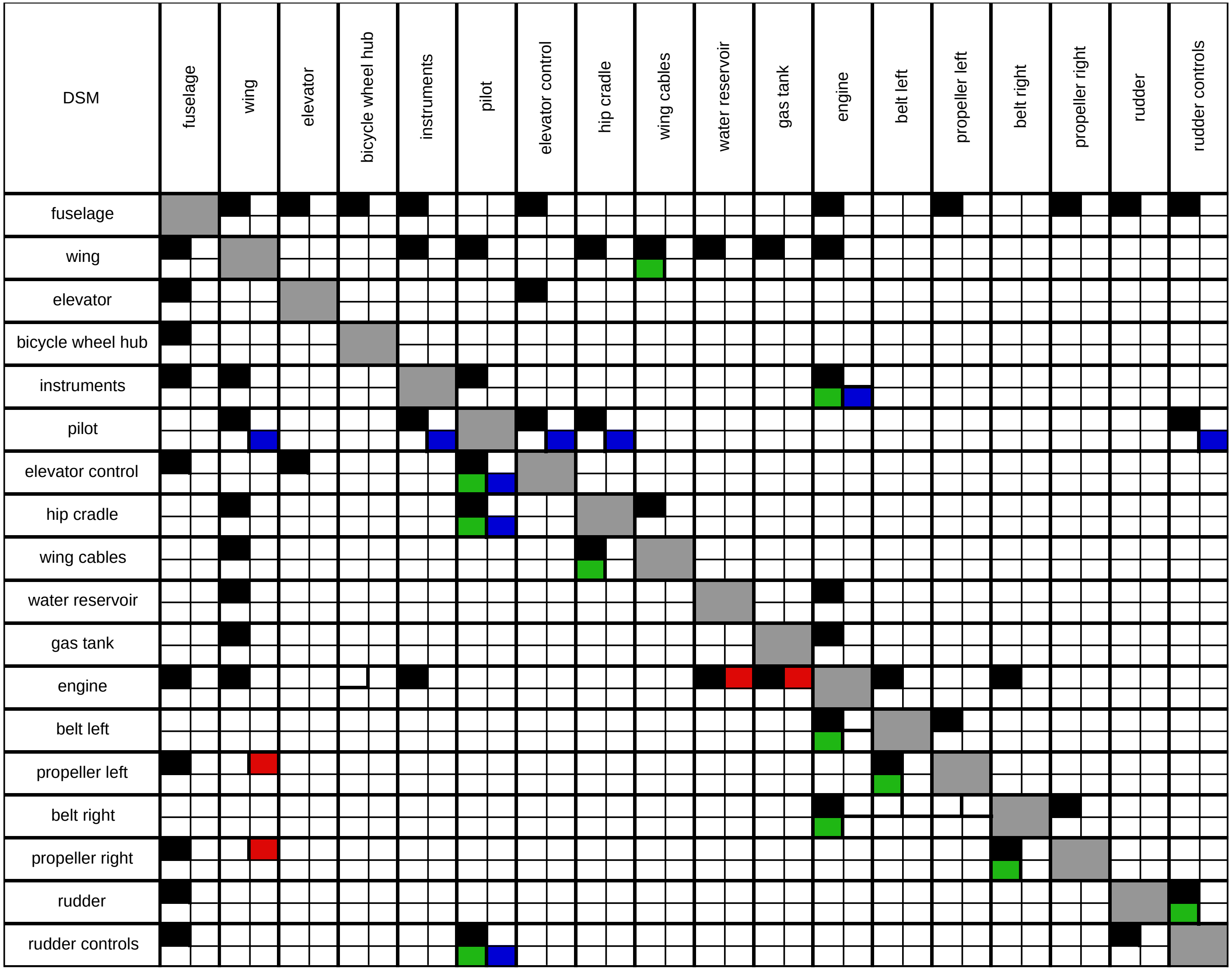
What is the conserved quantity in Systems Science (and therefore Systems Engineering)?

The Wright Flyer (1903)





Structural DSM of Wright Flyer



Legend	
■ (black)	Physical connection
■ (red)	Mass flow
■ (green)	Energy flow
■ (blue)	Information flow

DSM 18x18

Connections

62 Physical

4 Mass Flow

11 Energy Flow

9 Info Flow

Total: 86

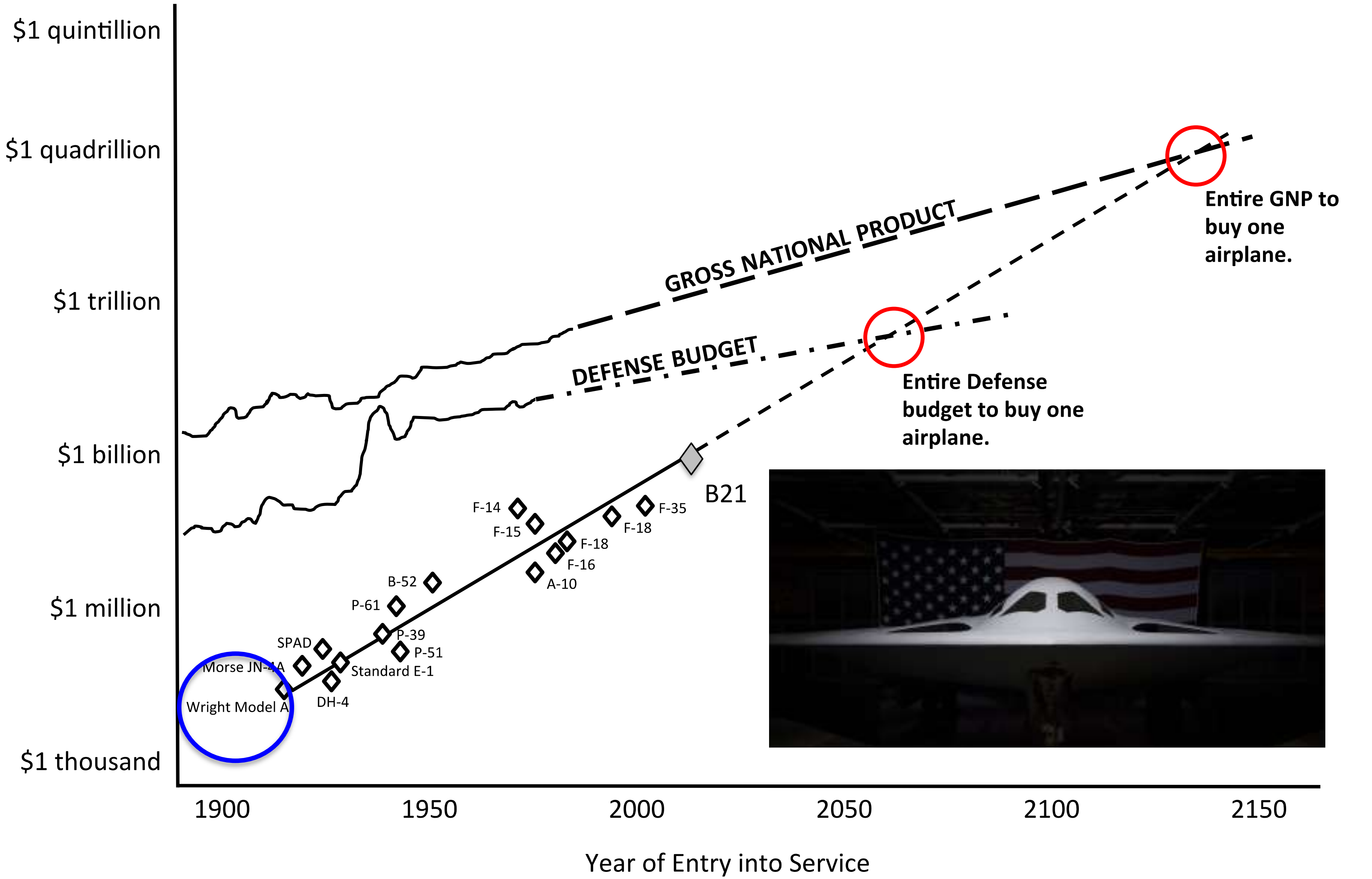
NZF = 86/1,224

= **7% density**

<k>=~5

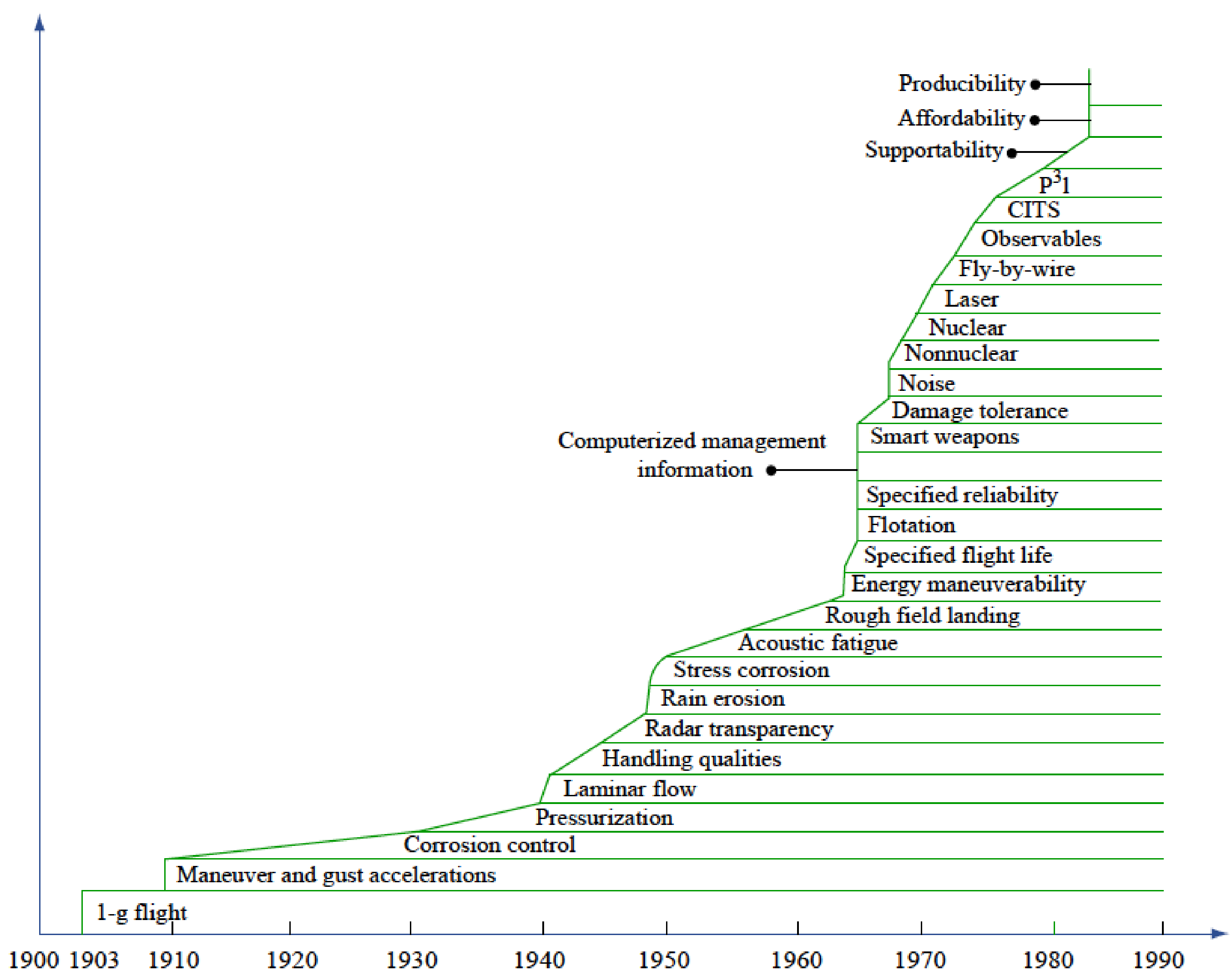
Design Structure Matrix (DSM) – captures structure of elements of form

Augustine's 16th Law



Norm Augustine, *Augustine's Laws*, 6th Edition, AIAA Press, 1997.

Functional Requirements Explosion in Aviation



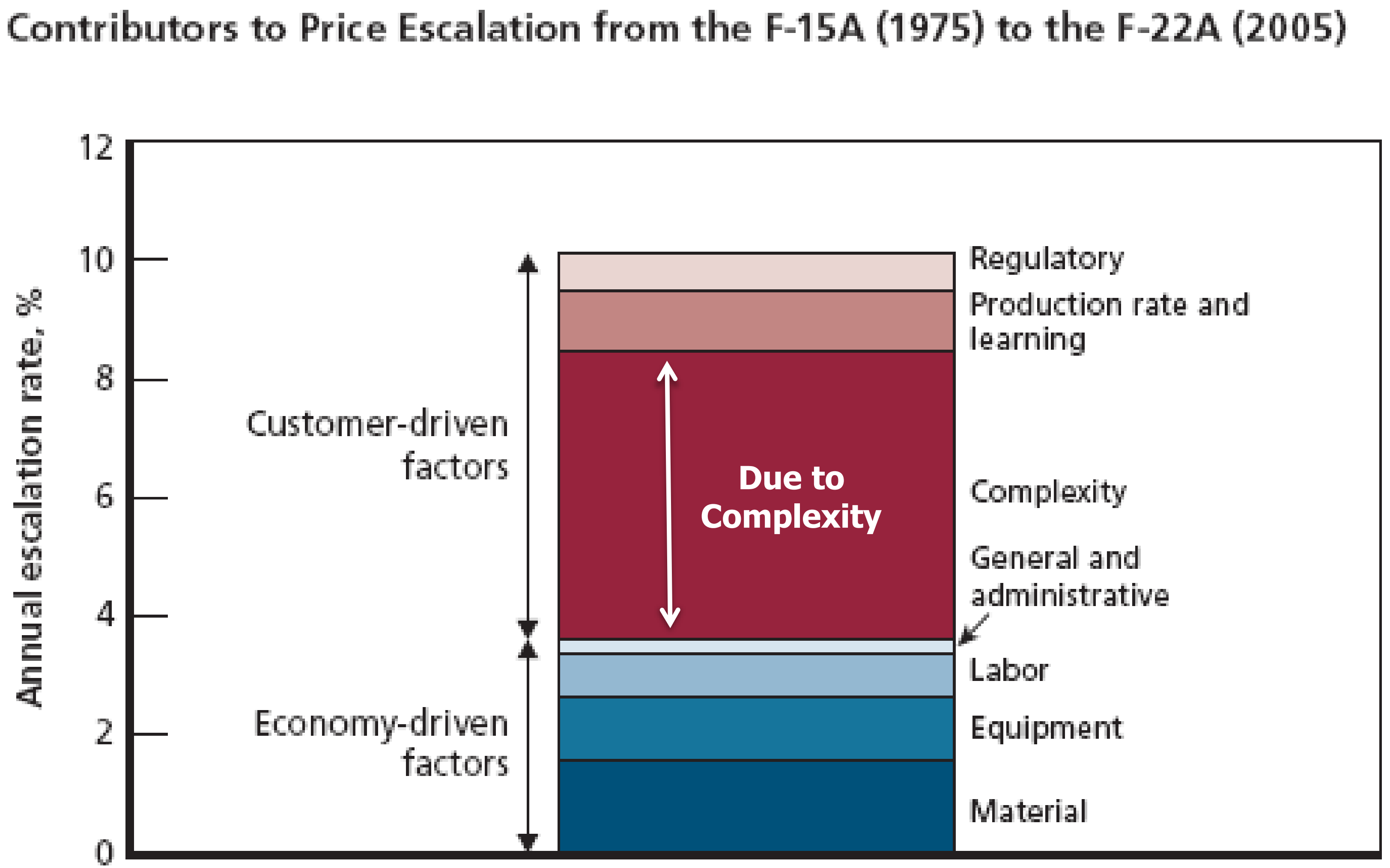
Design requirements growth for aerospace vehicles.

Image by MIT OpenCourseWare.

F-35 JSF

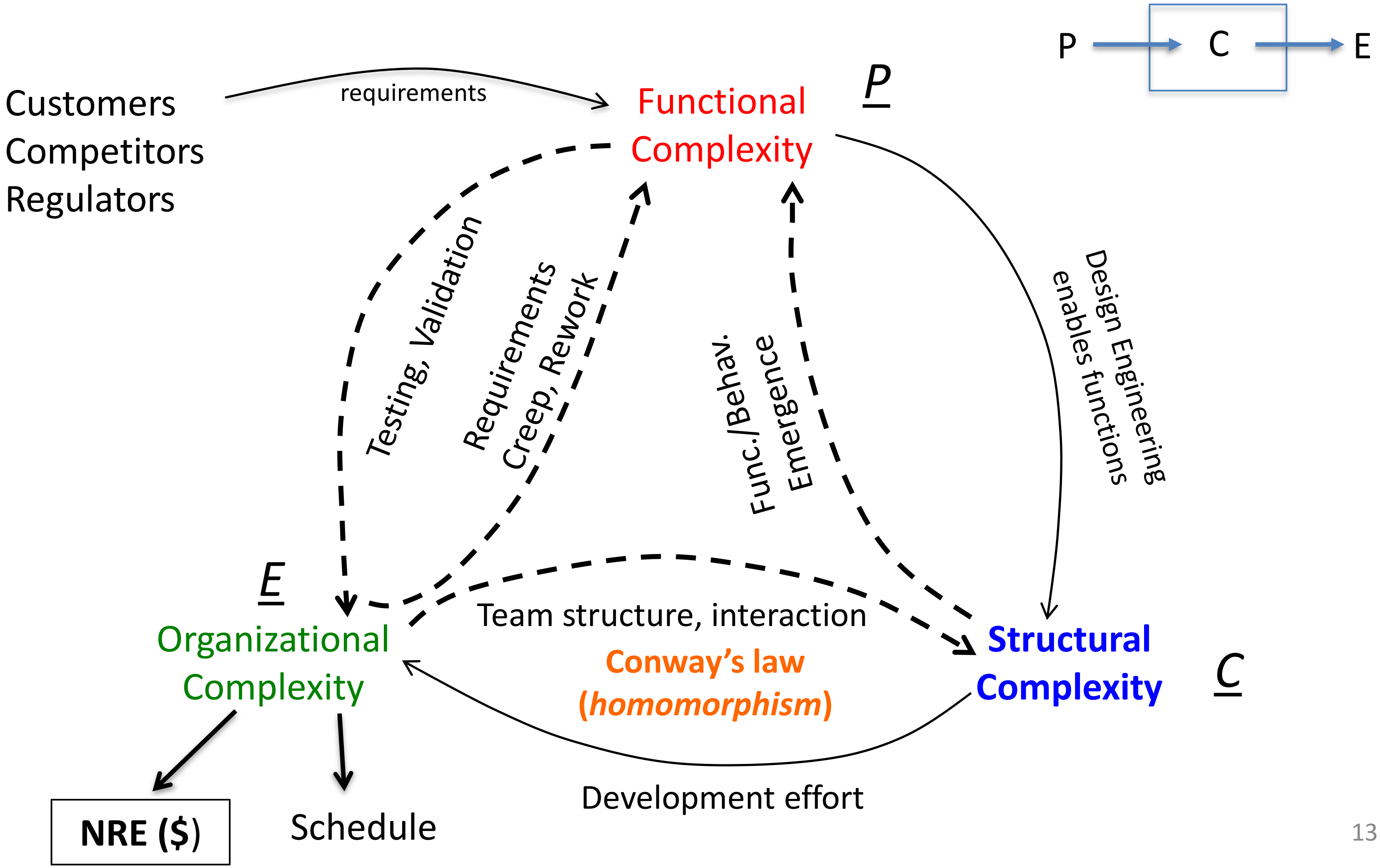
Active Matrix Liquid Crystal Display image display
Sensor fusion
Binocular 40 degree by degree field-of-view
Integrated day and night camera
Ejection Safe to 600 knots equivalent air speed

What is driving this escalation of cost?



Source: DARPA TTO (2008)

Three Dimensions of Complexity



NRE Cost – Non-Recurrent Engineering Cost

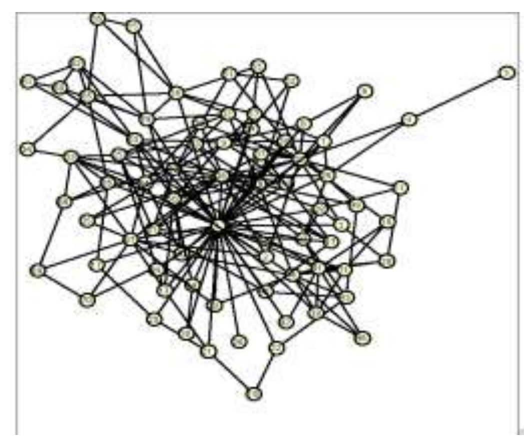
The Structural Complexity Metric

Structural Complexity, $C = C_1 + C_2 . C_3$

Complexity due to system topology (a scaling factor) typically > 1

Complexity due to components alone (number and heterogeneity of components)

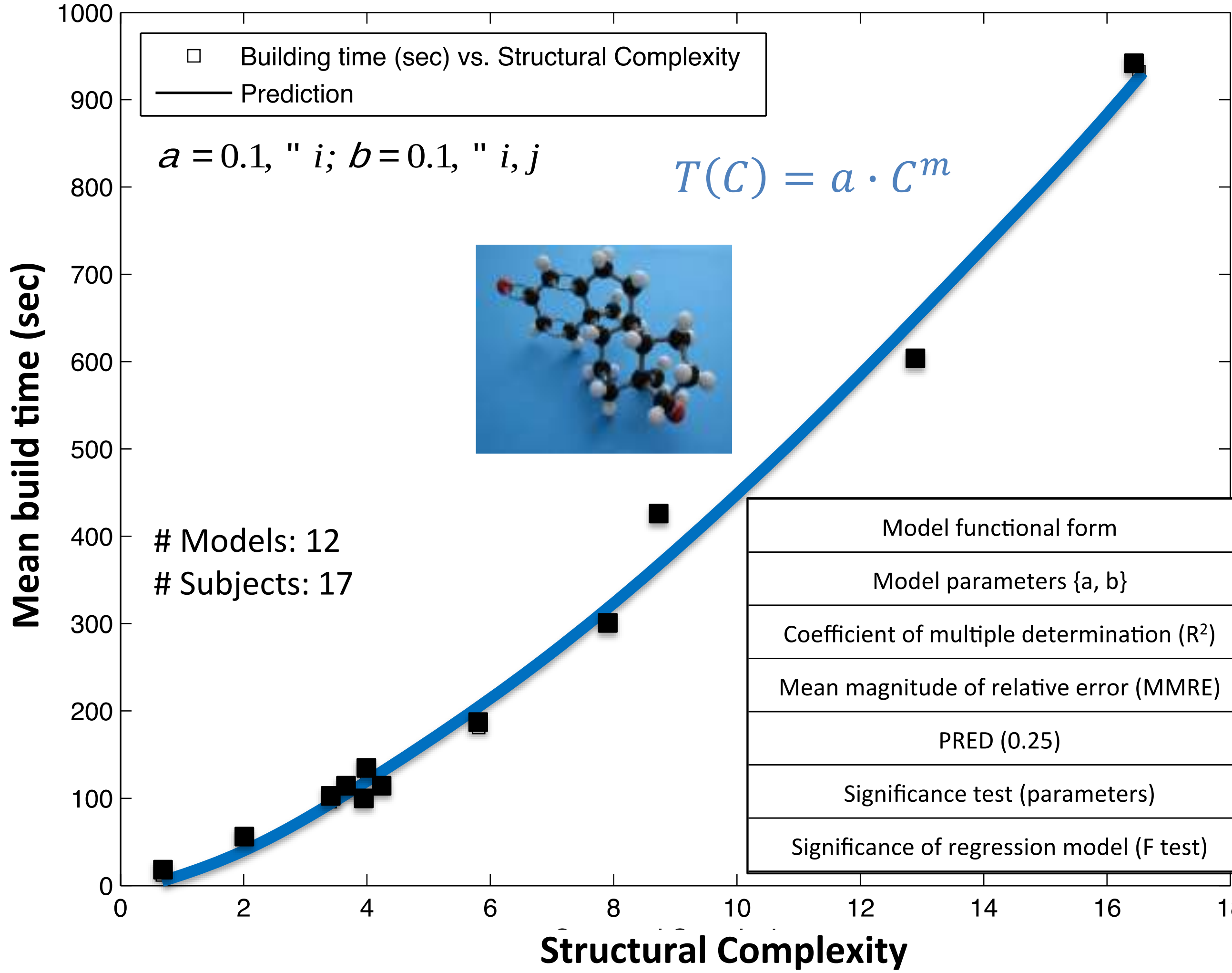
Complexity due to pair-wise component interactions (number and heterogeneity of interactions)



Sinha, Kaushik, and Olivier L. de Weck. "Empirical validation of structural complexity metric and complexity management for engineering systems." *Systems Engineering* 19, no. 3 (2016): 193-206.

Experiment: We slow down w/complexity

IDSS
MIT INSTITUTE FOR DATA, SYSTEMS, AND SOCIETY



$T(C) = a \cdot C^m$

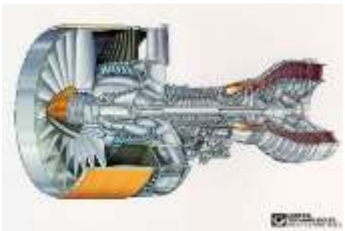
Exponent *m ~ 1.5*

Model functional form	$Y = aX^b$
Model parameters {a, b}	{ 14.68, 1.4775}
Coefficient of multiple determination (R^2)	0.992
Mean magnitude of relative error (MMRE)	0.107
PRED (0.25)	0.9167
Significance test (parameters)	t_a = 28.2, t_b = 30.67 (>t_0= 2.131)
Significance of regression model (F test)	f = 124 > $f_{0.05,1,10}$ = 4.54

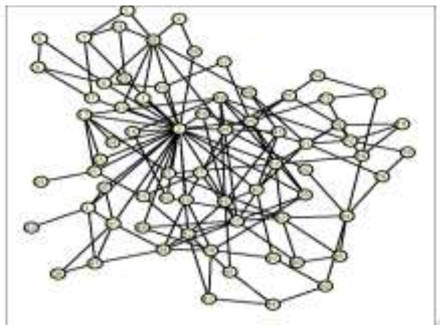
Structural Complexity, $C = O(n^{1.08})$ ¬ *mild super-linearity*

Average build time, $t = O(C^{1.48})$ ¬ *strong super-linearity*

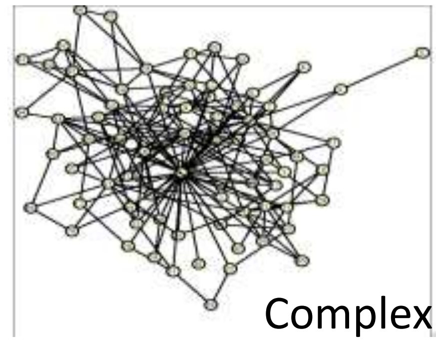
Empirical Data: Complexity Increase of Engines



Old



Complexity = 351



New

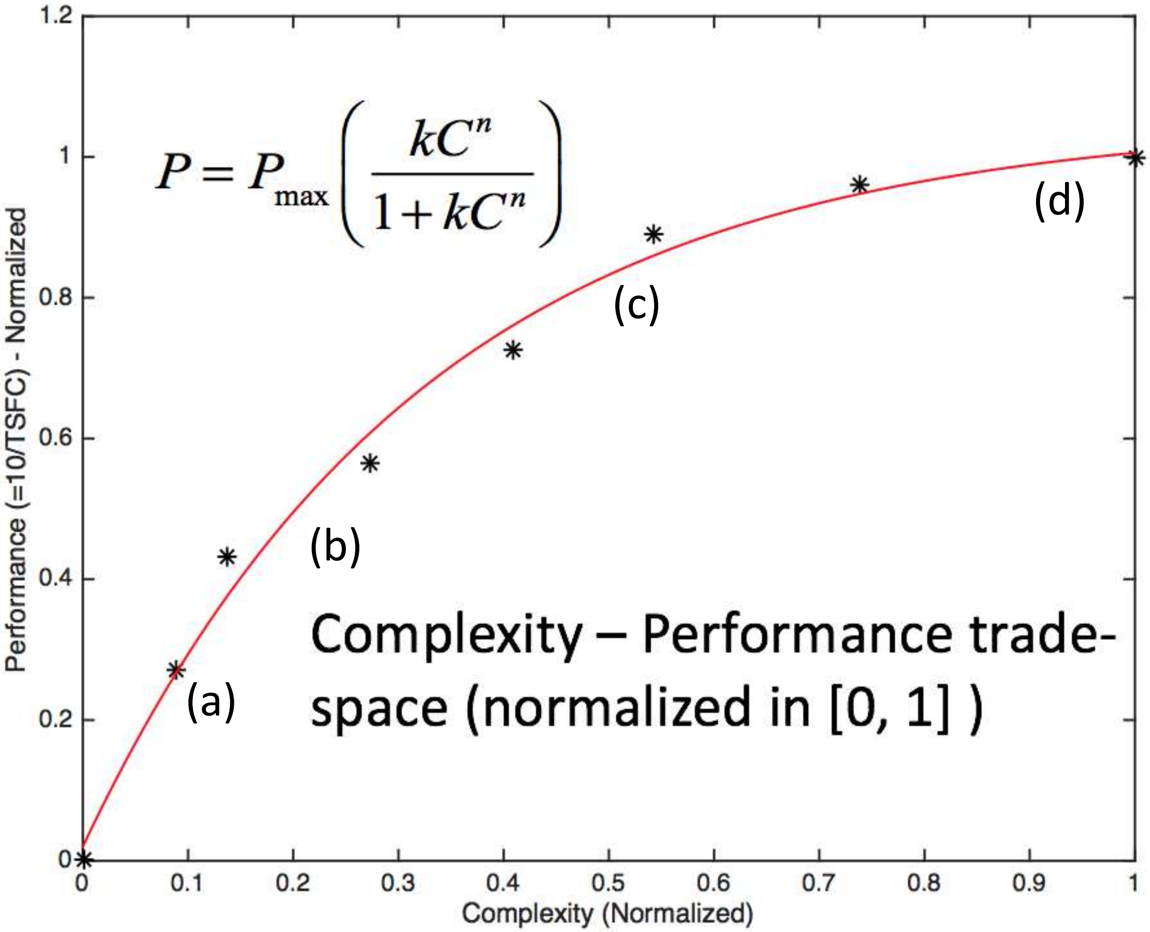
Complexity = 499

Complexity increase +42%

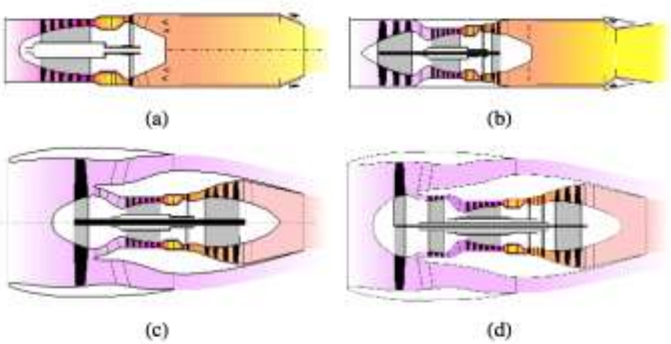
	C_1		C_2		C_3		C		C/C_{ML}		C_{new}/C_{old}
	Old	**New**	**Old**	**New**	**Old**	**New**	**Old**	**New**	**Old**	**New**	
Most Likely	161	188	126	184	1.51	1.69	351	499	1	1	1.42
Mean	179	244	141	240.4	1.51	1.69	392	650.3	1.12	1.30	1.65
Median	178	242	139	238.9	1.51	1.69	388	646.8	1.10	1.29	1.66
70 percentile	181	247.9	145	246.2	1.51	1.69	399.6	663.94	1.14	1.33	1.66

Trend towards more distributed architecture with higher structural complexity and significantly higher development cost*. Similar trend was observed in Printing Systems.

Diminishing Returns with Complexity



Left: Diminishing returns of normalized TSFC performance for air-breathing aircraft engines versus complexity, Bottom: evolution from turbojet to geared high BPR turbofans



IDSS
MIT INSTITUTE FOR DATA, SYSTEMS, AND SOCIETY

Theoretical Foundations

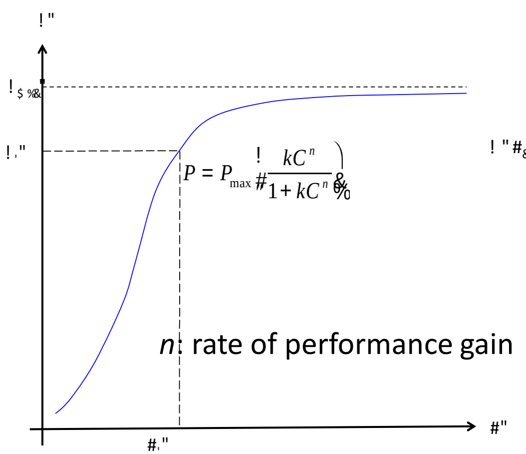
"TO" state:

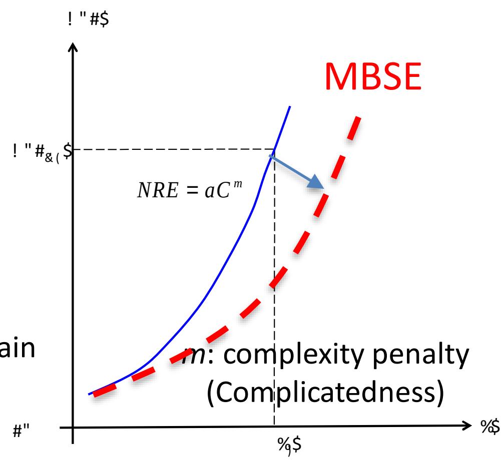
"The systems engineering foundations have a stronger **scientific and mathematical grounding** based on **advanced practices, heuristics, systems observable phenomena, and formal ontologies.** The foundations are shared across application domains, and provide additional rationale for selecting and adapting practices to **maximize value** for the particular application."

https://violin-strawberry-9kms.squarespace.com/theoretical-foundations

Complexity and Value Maximization

Complexity budget C* is the level of complexity that maximizes system Value !



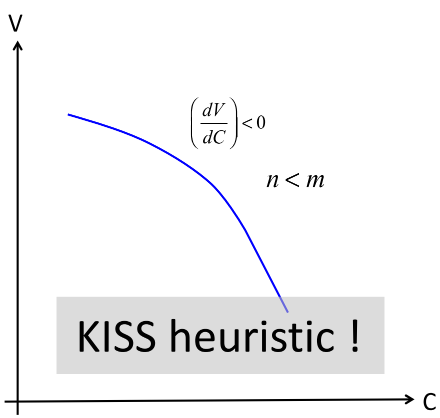


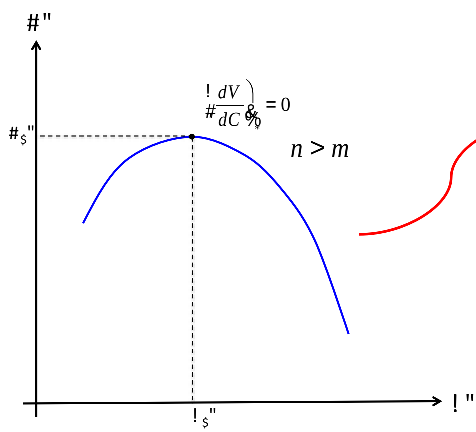
$$P = P_{\max}\left(\frac{kC^n}{1+kC^n}\right)$$

$$NRE = aC^m$$

$$V = \frac{P}{NRE} = P_{\max}\left(\frac{k}{a}\right)\left[\frac{C^{(n-m)}}{1+kC^n}\right] = S\left[\frac{C^{(n-m)}}{1+kC^n}\right]$$

Value V is the ratio of Performance P over non-recurring Effort E → what is V*?

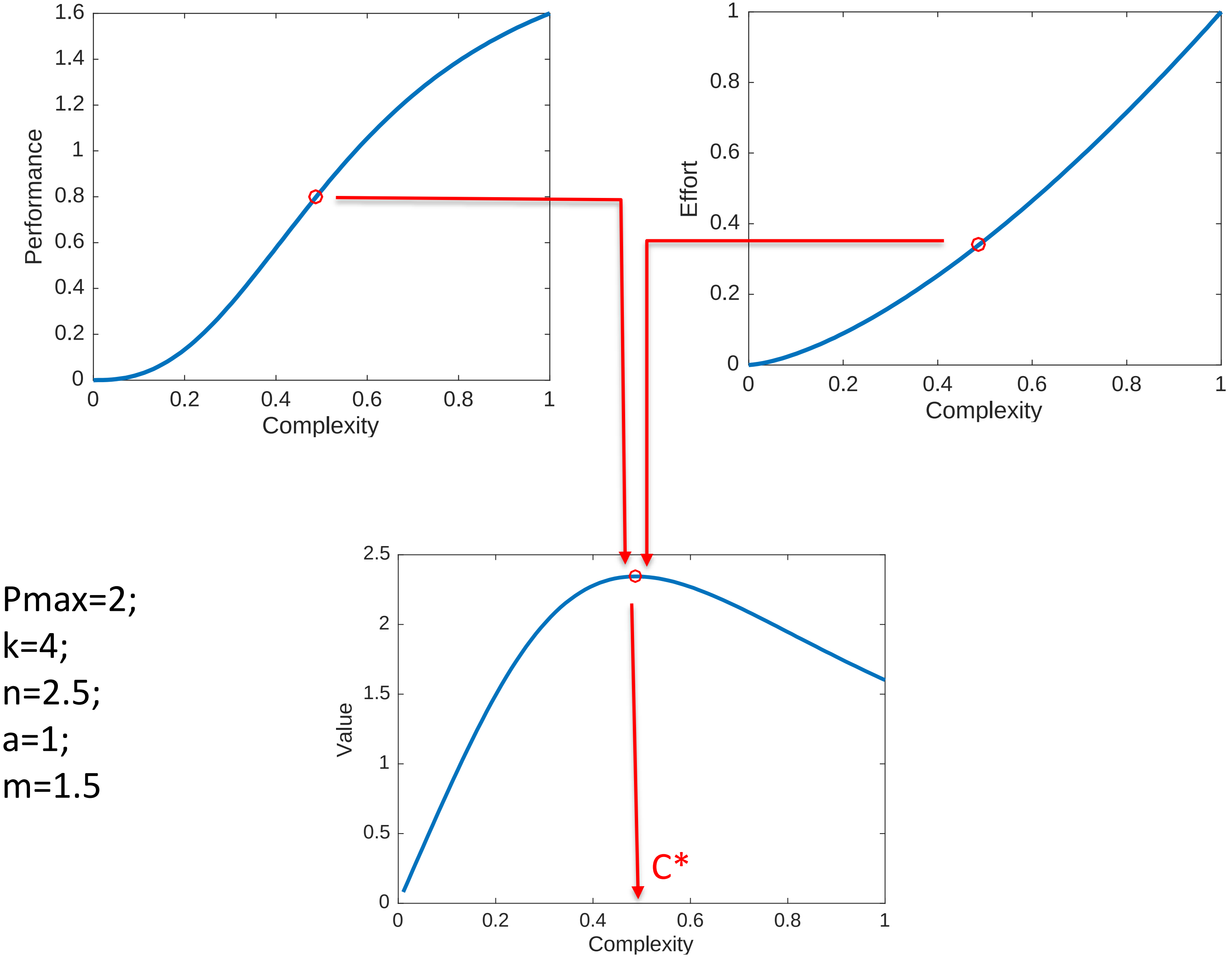




$$C_*^n = \frac{\left(\frac{n}{m}\right) - 1}{k}; \; P_* = P_{\max}\left(1 - \frac{m}{n}\right)$$

$$NRE_* = a\left[\frac{\left(\frac{n}{m}\right) - 1}{k}\right]^{\frac{m}{n}}; \; V_* = S\left(\frac{m}{n}\right)\left[\frac{\left(\frac{n}{m}\right) - 1}{k}\right]^{\left(1-\frac{m}{n}\right)}$$

Example: Complexity Target to optimize Value



Pmax=2;
k=4;
n=2.5;
a=1;
m=1.5

The First Law of Systems Science and SE: Conservation of Complexity

- First Law of Thermodynamics:
  - Conservation of Energy
  - The change in internal energy ΔU is equal to the heat *Q* added to the system minus the work *W* done by the system.

$$\Delta U = Q - W.$$

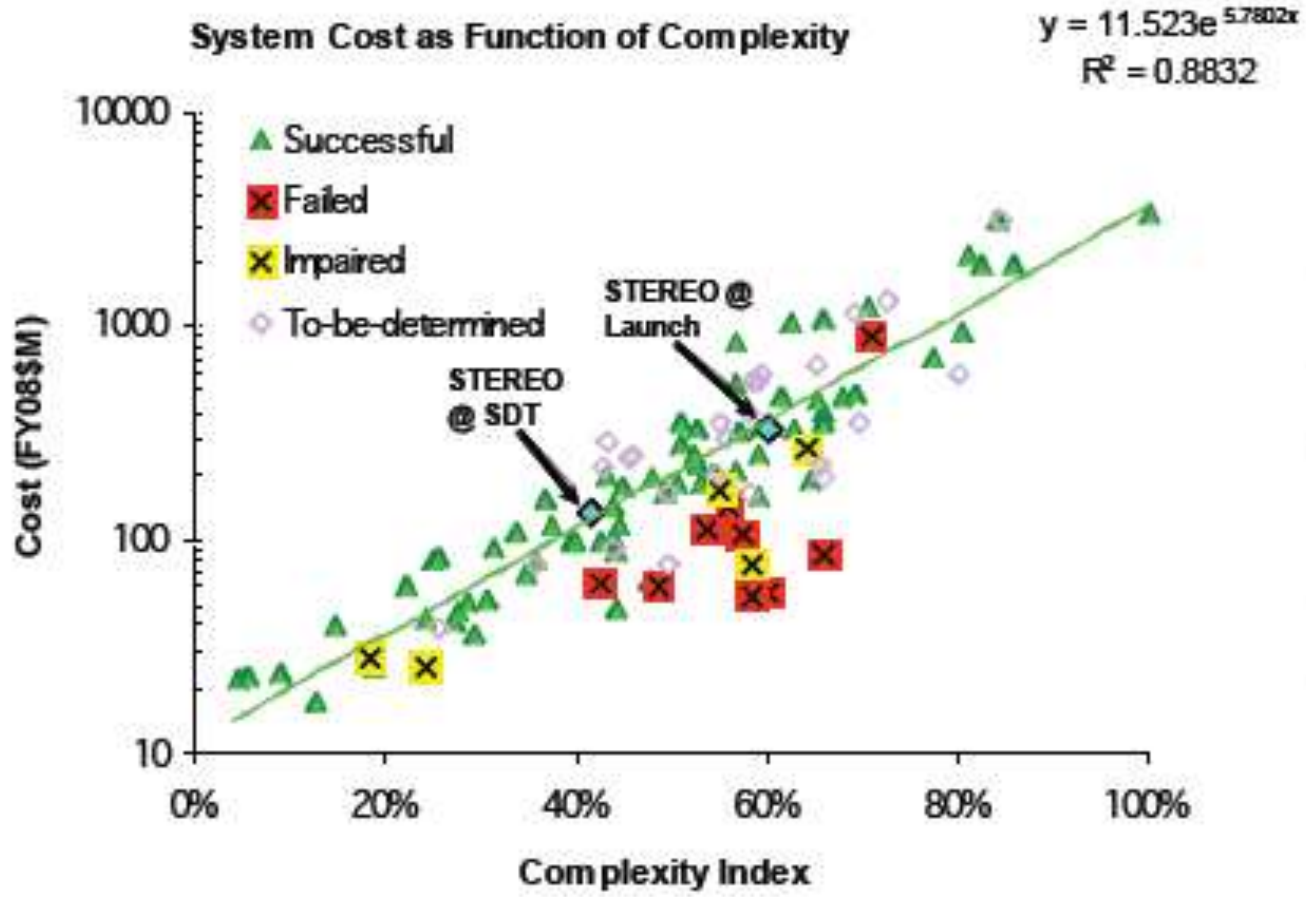
- The First Law of Systems Science and Engineering:
  - **Conservation of Complexity**
  - The change in complexity ΔC of the system is equal to a proportional change in expected performance ΔP minus the change in effort ΔE expended by the enterprise

$$\Delta C = \mu \Delta P - \varepsilon \Delta E$$

$$\varepsilon = -\frac{C^{1-m}}{2am} \qquad \mu = \frac{(1+kC^n)^2}{2PmaxknC^{n-1}(1-kC^n)}$$

Validation of the 1st Law: Successful vs Failed Systems

- **CoBRA (Aerospace Corp., 2008) – Complexity Index based on analysis of historical data.**
- **Projects that were highly complex but tried to cut development cost had high failure rates**



Key Messages

- Complexity *C* of artificial (and natural?) systems has been increasing
- This is driven by customers, competition, and regulation → functional performance P → structural complexity C → organizational effort E
- A rigorous measure of complexity is based on graph energy of DSM
  - C = C1+ C2*C3;
  - C3: Graph Energy is a measure of topological complexity
  - **Explicit complexity-based budgeting with clear targets is needed in SE**
- **First Law of Systems Science and Engineering (according to de Weck-Sinha):**
  - **Conservation of Complexity**
  - **Given a set of functional requirements P, establish minimum needed structural complexity C, and calculate organizational effort E (NRE) to satisfy the first law**
- **Violating the first law can lead to project or system failure !**

IDSS
MIT INSTITUTE FOR DATA, SYSTEMS, AND SOCIETY

INCOSE IW 2023
Future of Systems Engineering (FuSE)

Foundations Stream