Systems Engineering in Search of The Elusive Optimum

Professor Derek K Hitchins

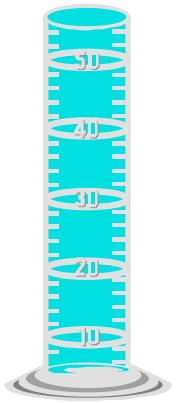
At a recent SE Conference

- Concern that systems engineering seems to reappear and disappear periodically on about a 25 year cycle—coincidentally, perhaps, once every generation…
- Many papers, although interesting, did not seem to be about systems engineering *per se*—whatever that is…?
  - work already done presented as though originally founded in systems engineering?
  - illustrating formalized processes which were claimed as successful instances of SE
- Perhaps these two phenomena not entirely disconnected?

Questions

Questions:—

1. Why is systems engineering seen by successive generations as exciting, important, essential, only to be supplanted after a very few years?



- what are it's supposed advantages?
- why do they seem so important, yet so elusive?
- what goes wrong?
- is the cycle inevitable, or preventable?
- is there a way of achieving and maintaining the ultimate advantage offered by some proponents of SE?

2. What is the essence of SE—is there a "litmus test" to see if SE is/has been really employed?

Basics

First, what is a system?

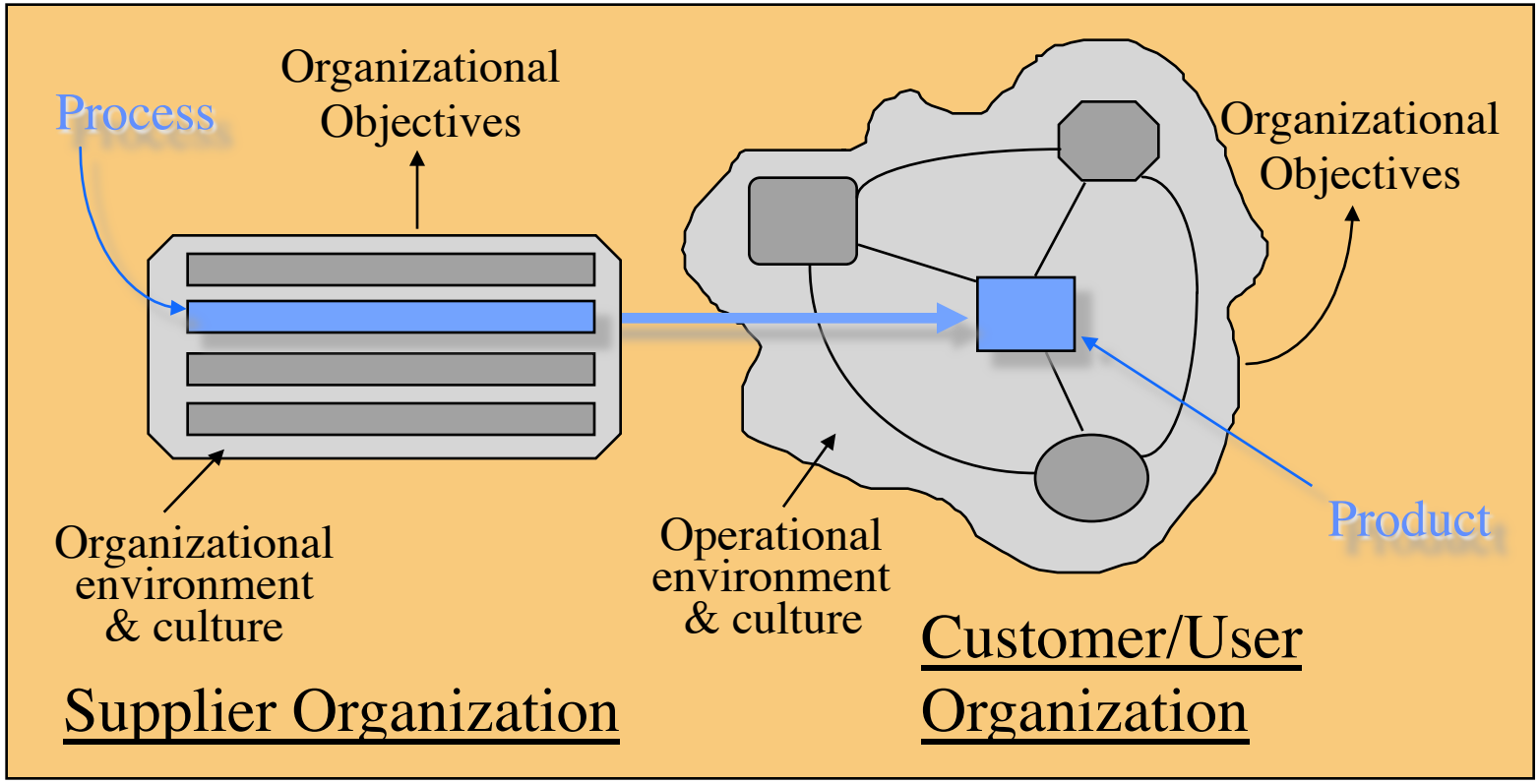
- "…an open set of complementary, interacting parts with properties, capabilities and behaviours (PCBs) emerging both from the parts *and* from their interactions
- From the definition, any system could clearly be made up from different kinds of parts, arranged and interconnected in different ways. Some parts and some configurations may produce overall PCBs which were more valued, i.e. optimal
- Fundamentally, systems engineering is about synthesizing optimal solutions to problems

Promise of SE

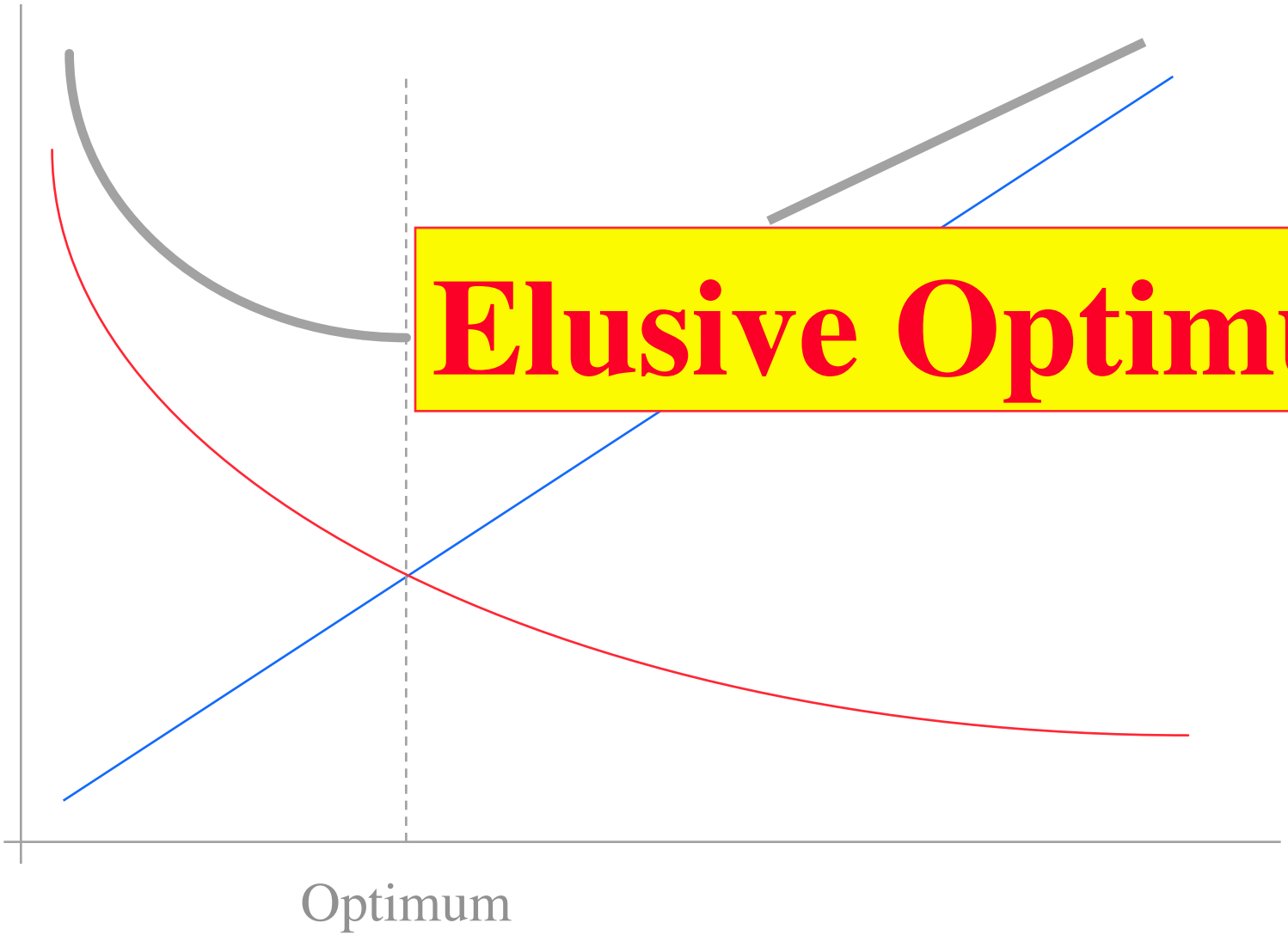
- Anyone can create a new system. SE creates *optimal* systems —to meet any need or opportunity, where:—
  - "system" could be process or a product
  - "optimum" defined by context
    - e.g. best balance of efficiency, effectiveness and quality
    - e.g. shortest time to market consistent with cost and quality
    - e.g. meeting customer's and users' needs at minimum cost
  - "Optimum" = max/min for some ratio, e.g. value for money, kills per loss, profit per air mile, cost-exchange ratio, etc.

Goals, Objectives & Targets (GOTs)

- Systems engineering seeks to create:—
  - a **process** which is optimized to meet the GOTs of the Supplier
  - …*and* which creates…
  - a **product,** optimized to meet the GOTs of the Customer and End-User



- Goal? Repeat business, which furthers ***Supplier's*** GOTs


Elusive Optimum
Optimum

Achieving the Optimum System

- "Optimum" demands balancing process…
- System elements arranged for overall performance, but…
- …elements mutually interact—changing any element or relationship affects other elements
- So, balancing can be tricky—
  - hence arcane company secrets
- Achieving "optimum" needs *whole system* in balance
- Excluded elements unable to contribute/adapt to optimizing process
- Must **include non-adaptable items** so that others may adapt around them to find best *practical* optimum

Context Shifts

- "Optimum" defined by context, interacting systems, environment and fore-knowledge.
- Tempo of change increasing in recent decades
  - improved communications, transport, information handling
  - reducing commercial product lifecycles
  - increasingly non-linear dynamic world
    - edge of chaos, self-organized criticality, auto-complexity generation
  - shortening "horizon of knowability"
- If context, interacting systems, environment and knowledge change, then "optimum" configuration must change, too
- Increasing tempo demands a fresh look at how we design and create future systems—a new systems engineering

"Optimum" Dynamics

- "Optimum" cannot be some ***fixed*** balance point
  - Optimum path to any goal is **not** fixed—it changes as the environment changes, as new factors emerge
  - Properties, capabilities, behaviours and arrangement of components for (e.g.) Optimum Performance are determined by context/environment —as this changes, so does optimum point
- Optimum system performance depends on other, interacting systems staying the same—increasingly, they do not! As they change, so does the optimum performance point:—
  - change continuous, inevitable—2nd Law of Thermodynamics

Significance of "Optimum"

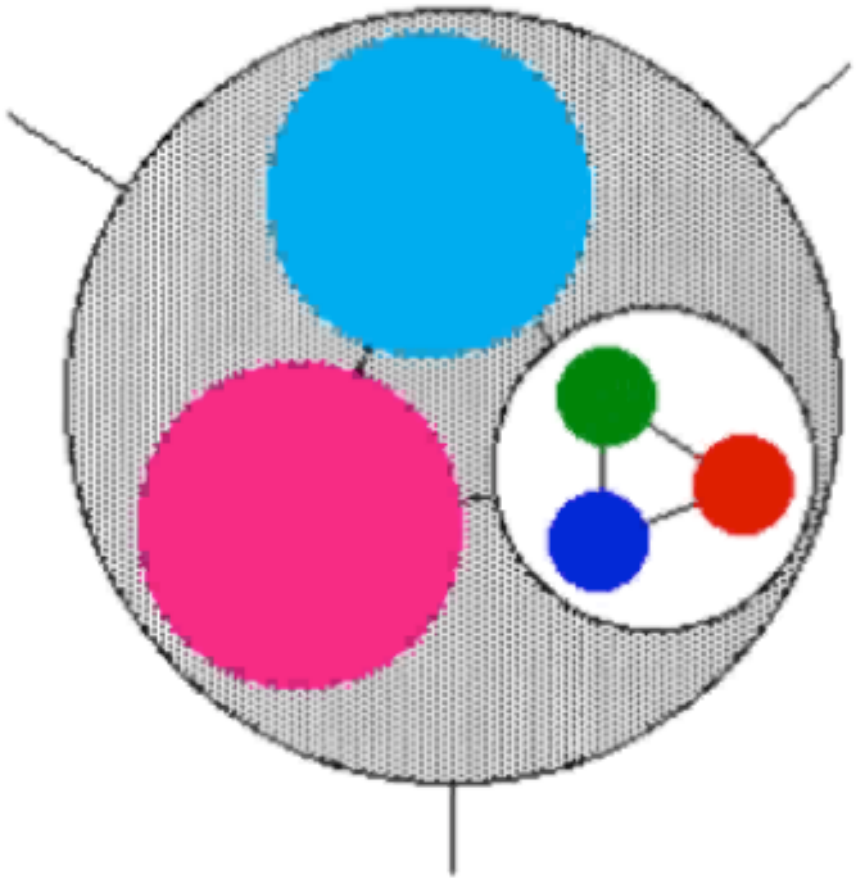
…change is continuous and inevitable…

- It is this observation, more than any other, which distinguishes systems engineering from other disciplines
- Unlike their technology, "optimal" systems are forever in dynamic state of change—
  - induced by mutual interactions with other, changing systems
- Challenge of systems engineering—

create effective, enduring systems
which pursue and maintain optimality
through change.

Dynamic Optimization

- Figure represents a typical nesting view of systems within systems within systems…
- Suppose right hand White system is to be optimized. Then…
  - Adjust Red subsystem
  - Affects Blue and Green
  - Affects White as intended…
  - White affects Magenta & Cyan…
  - Which rebound on White in turn…
  - affecting Blue, Green and Red
- Optimizing requires this dynamic view of the process



Looking Outwards

- Because systems exist within systems exist within systems…
- …what is true for subsystem is true for whole system:—
  - whole system must be balanced with all other, interacting systems in their mutual operating environment
  - implies that system-to-be-created must be viewed as though it already existed and was operating
  - only in this way can the whole system be created to complement interacting whole systems yet achieve its purpose
- For this reason, systems engineers concern themselves with looking "outwards" into the future environment
  - often involves some form of modelling
  - always involves imagination

Systems Engineering Litmus Test A

Regardless of system type…

- First and foremost—look ***outwards***
  - into environment,
  - into other systems
- Identify what effect your potential system will have on other (future) systems in the (future) environment
- Identify/anticipate response of those future systems
- Hence establish requisite properties, capabilities and behaviours of your dynamic, interacting system
- Conceptually creates (future) order—compatible, balanced, enduring effective system

Systems Engineering Litmus Test A

Regardless of system type…

Systems Engineering Litmus Test A

"*Systems Engineering looks* outwards *first to identify the* Dynamic Problem Space"

"LOOK OUTWARDS *FIRST*…"

Systems Engineering Litmus Test B

"The whole system—product and/or process—
is* continually re-balanced *to optimize
properties, capabilities and behaviours
throughout its lifecycle"

"RE-OPTIMIZE THROUGH LIFE…"

Systems Engineering Litmus Test C

"The whole system—process and/or product—is balanced by adjusting **internal** *parts/relationships to optimize* **whole-system** *properties, capabilities and behaviours"*

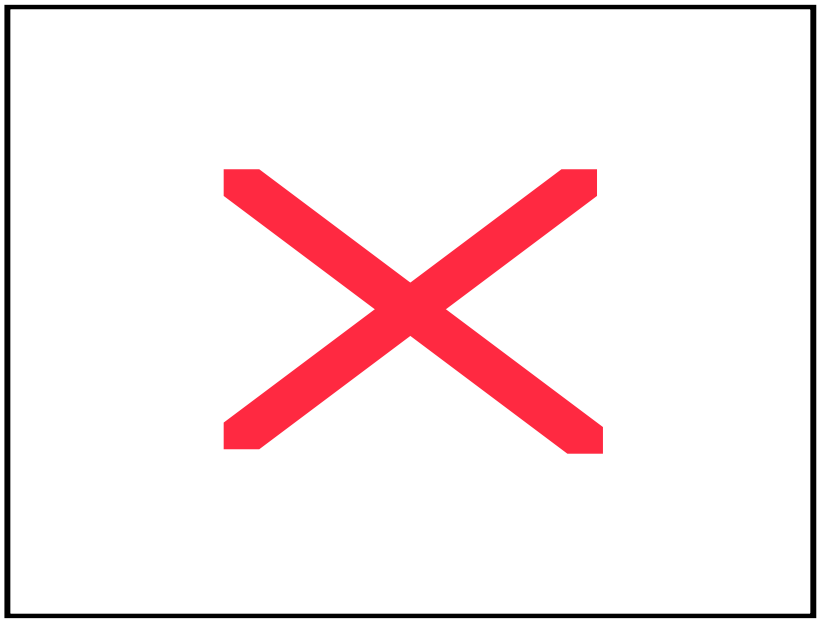
"ADJUST PARTS TO OPTIMIZE WHOLE..."

Systems Engineering Litmus Test D

"Properties, capabilities and behaviours, are optimized* while *the whole system* is interacting *with other systems in its operating environment"

– Often, this necessitates modelling

"OPTIMIZE *DYNAMIC* PERFORMANCE..."



Applying Litmus Tests to Contemporary Practices

Item	Candidate	Litmus Tests *A*	*B*	*C*	*D*	Comment
1	Classic decomposition, progressive build-up "V"	N	N	N	N	Generally fails all Litmus Tests. Systematic, but not *systems*.
2	Quality Function Deployment	N	Y	N	N	Balances contributions during specification. Static only. Necessary, but not sufficient
3	Classic systems engineering. Continually proposing variety of solutions, trading-off to find best fit…	Y	Y	Y	Y	Can be hit-or-miss. Needs many solutions to ensure finding optimum. *D* presumes system proving in dynamic simulation
4	Concurrent Engineering, UK-style[1]—integrated product design teams	Y	N	Y	N	Intended to reduce rework by anticipating errors and omissions. Static, one-off. *Creates* organizational entropy
5	Concurrent Engineering, UK-style—telescoped, overlapped processes/activities	N	Y	Y	N	Intended to minimize time to market. *Creates* organizational entropy. Fragile, high-risk.
6	Centralized software development, intensive software systems[2]	N	N	N	N	Increases both organizational and process entropy. Not systems engineering.

Optimum whole comprised of sub-optimal parts…

Choosing elements/subsystems which are themselves optimized does NOT result in an optimum overall system

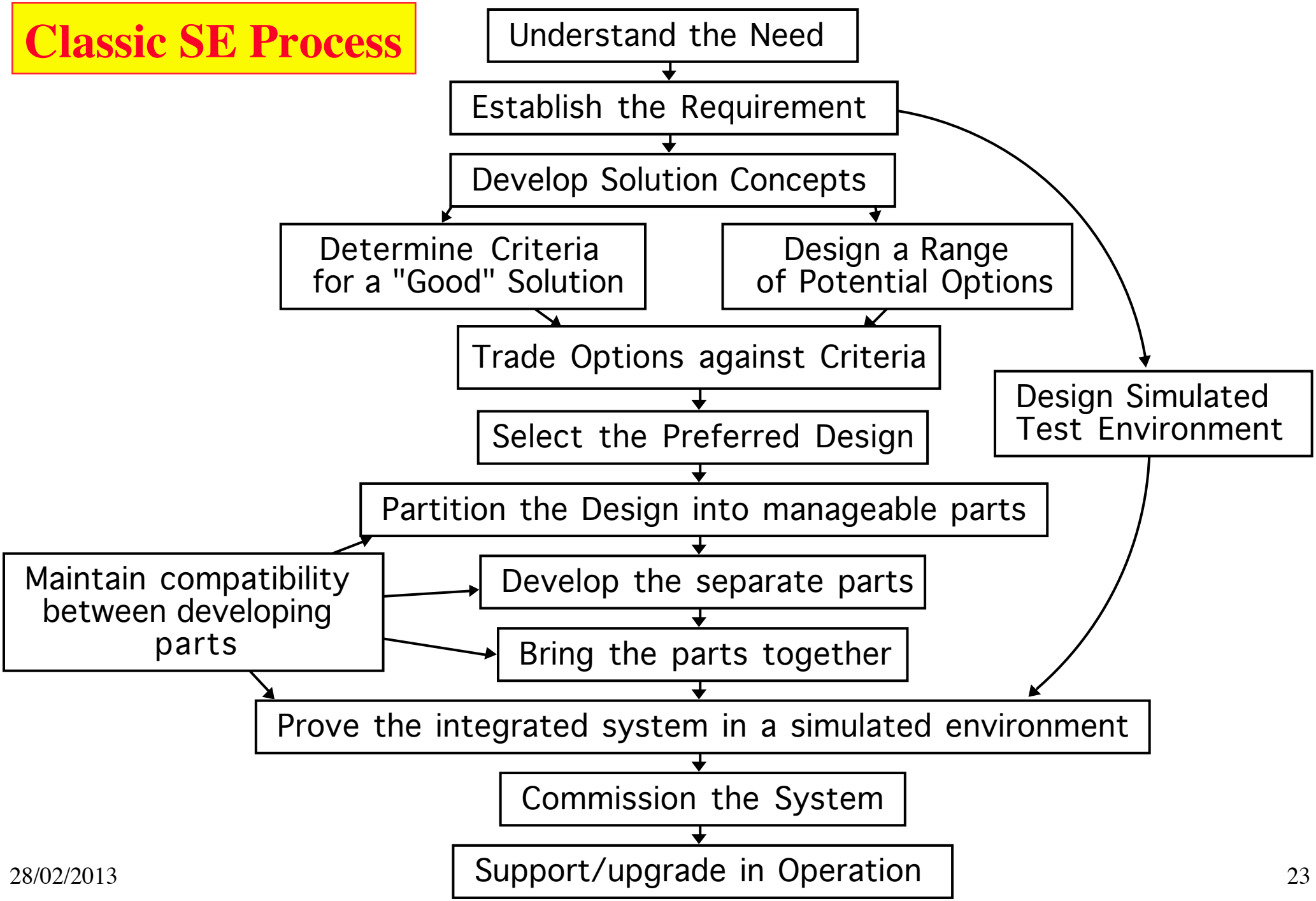
- collection of soloists does NOT make a good choir/orchestra
- F1 car made from the best parts of Williams, Ferrari, Maclaren,Beneton, etc., would NOT make the best F1 car
- well-designed and tested artificial heart may NOT suit the transplant patient

- Reason? Each element/subsystem has either been:—
  - designed/adjusted to optimize some other overall system, or…
  - optimized in its own right and its own context

Systems Engineering Process Steps

- Explore/bound the dynamic problem space
- Synthesize a dynamic solution
- Develop viable solution concepts
- Choose the optimum solution concept
- Design the dynamic system solution
- Select, connect and configure parts to meet the design
- Test and tune dynamic system solution in representative dynamic problem space
  - adjust/modify parts and interactions to realize/optimize PCBs of whole
- Fit dynamic solution into dynamic problem space
- Resolve the dynamic problem

Classic SE Process



Optimum
and
Entropy

Entropy—the silver bullet!

- "Optimum" associated with minimizing configuration entropy
  - degree of disorder in the pattern
- This notion requires proof
- If true, then "optimum" = simple, linear/untangled

Can it really be that easy?

Entropy—the silver bullet!

- Minimum entropy condition simply recognized by inspection—
  - human cortex *evolved* able to reduce (perceived) entropy
  - humans see patterns in clouds, embers, tea-leaves, etc.,
  - sense of musical rhythm and cadence…
- So, for some, yes! it really can be that simple!

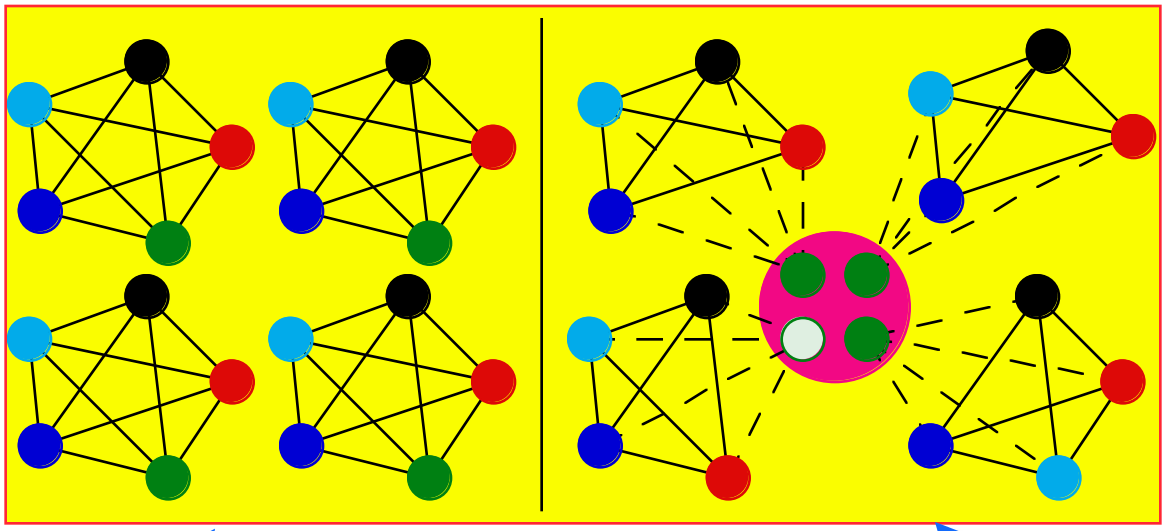
About Entropy…

- "The measure of a system's energy that is *unavailable for work!*"
- **High entropy**—then internal energy remains locked inside, unable to escape
- **Low entropy**—then most of energy can be directed towards useful work
- Hence "optimum" is associated with low (configuration) entropy, minimum disorder, explaining why…
- …Systems Engineering seeks to reduce entropy in Process *and* Product

Corollary:—no entropy reduction, no SE

Entropy & Economy of Scale

- Independent
- Concentrated / motivated small teams

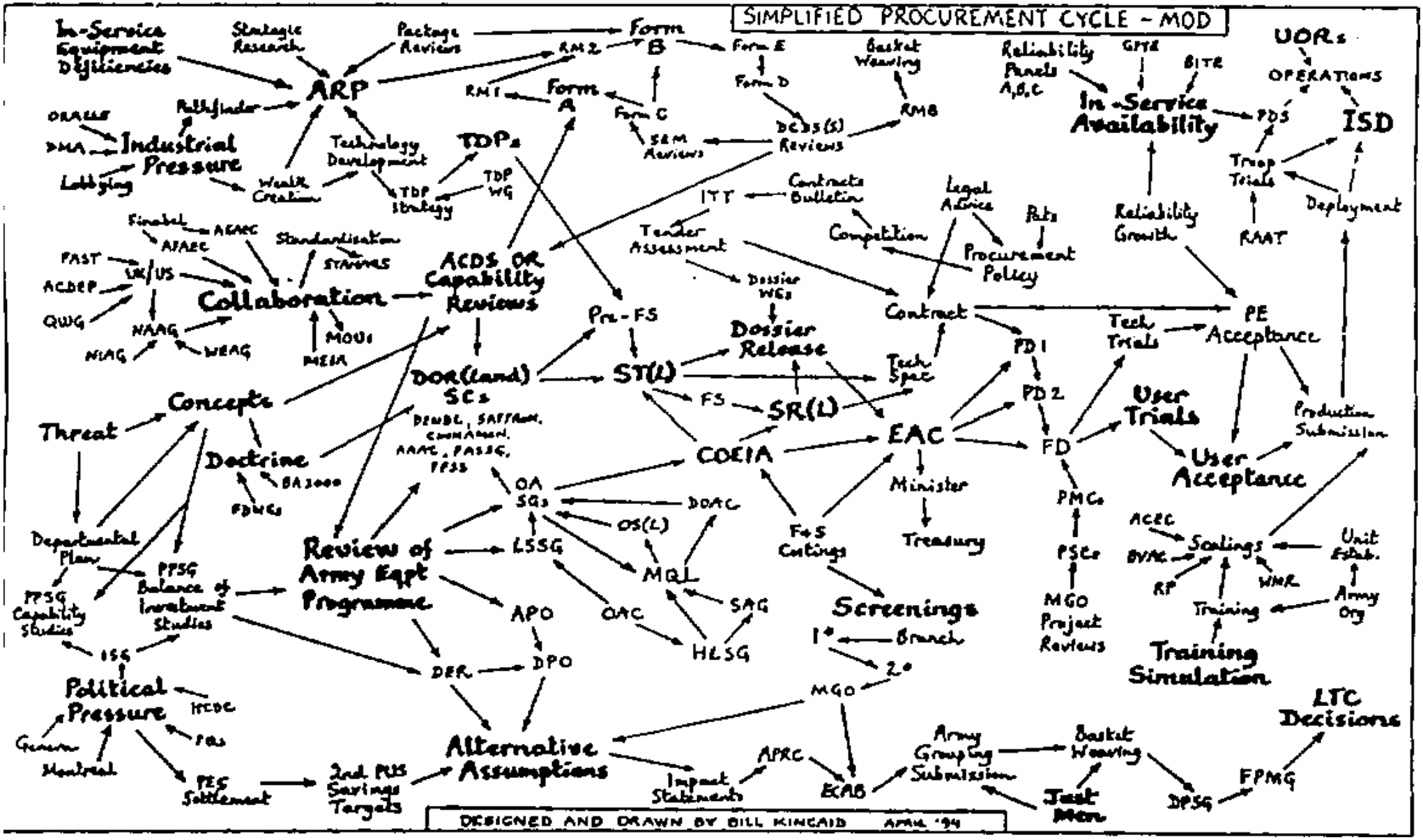


- Economy of Scale
- Control of standards
- Impersonal large teams

- Total 4 x 31 + 15 = 139 different arrangements for four independent teams

- Total = 2^{19} - 1 = 524,287 different ways
- high *potential* entropy, need additional management & control

High-Entropy Organization



Mathematics

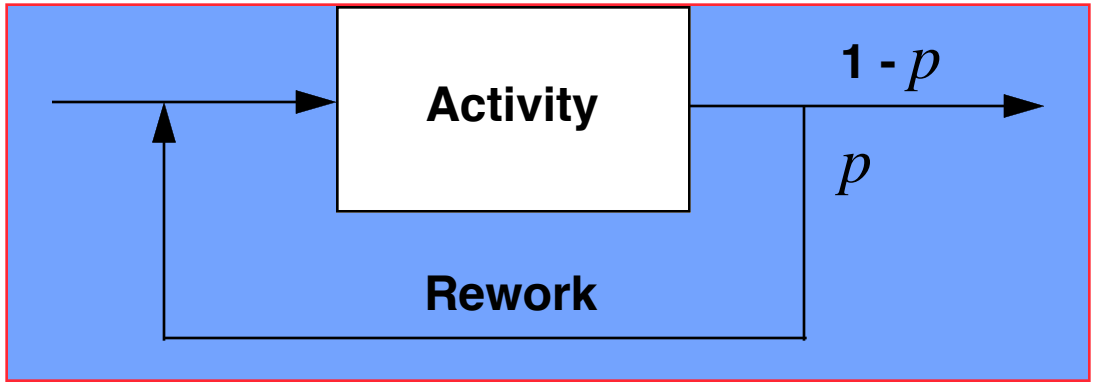
- Didn't need the mathematics—result was obvious by comparing two diagrams
- Generally, do not need mathematics to find optimum
- Human eye-brain combination *superb* at identifying best proportions, balance points, patterns…
  - pyramids, gothic cathedrals, bridges…built on instinctive architect/ engineer judgement
  - human cortex minimizes perceived entropy—powerful machine
- Engineers deal with static world of enduring optimality. Distrust instinct—prefer calculation.
- Systems engineering deals with dynamic world of shifting optimality—experience-based-instinct invaluable

Finding the
ptimum Entropy

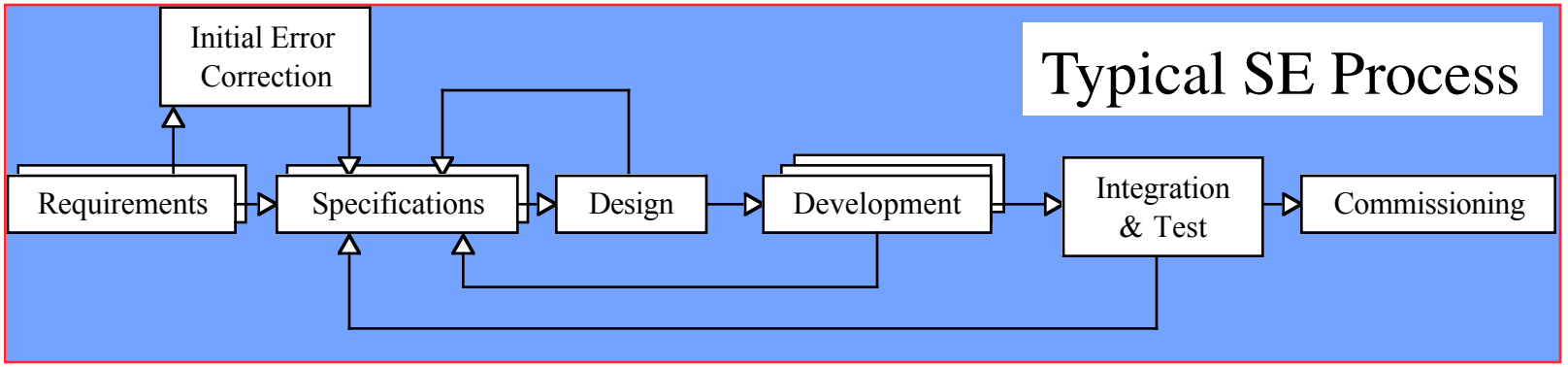
What does Entropy Look Like?

- Many parts in whole
- Turbulent flow in *process*
- Tangled-up parts in *product*

Turbulence in Process



- p = 1 – (Direct Path Time/Total Time)
  - where Direct Path Time is the time without rework
    Total Time is the time with rework

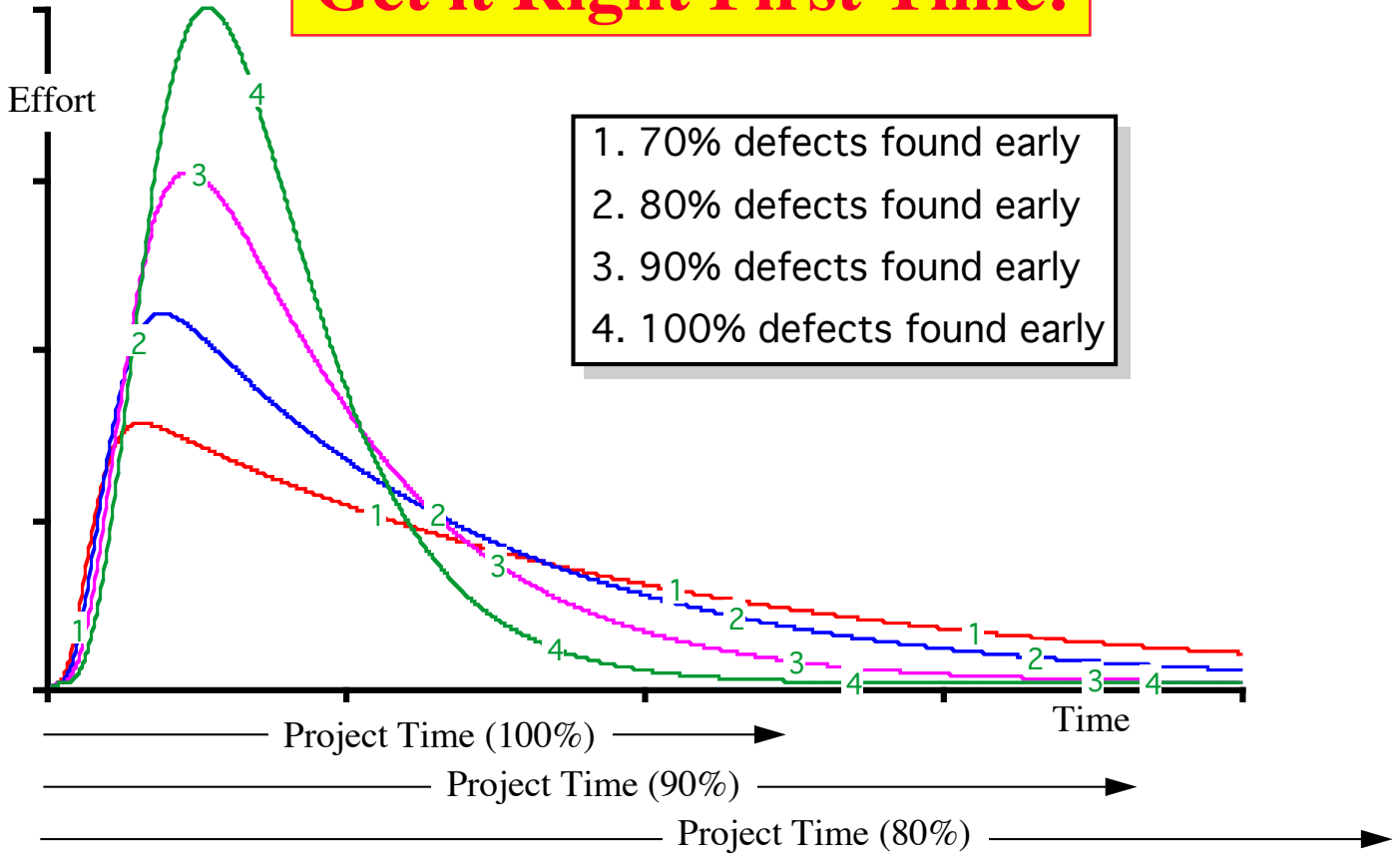


Process turbulence loops caused by errors, poor strategy, supervision delays, specification "drift", inexperience, inadequate training, rework…

Finding the Optimum

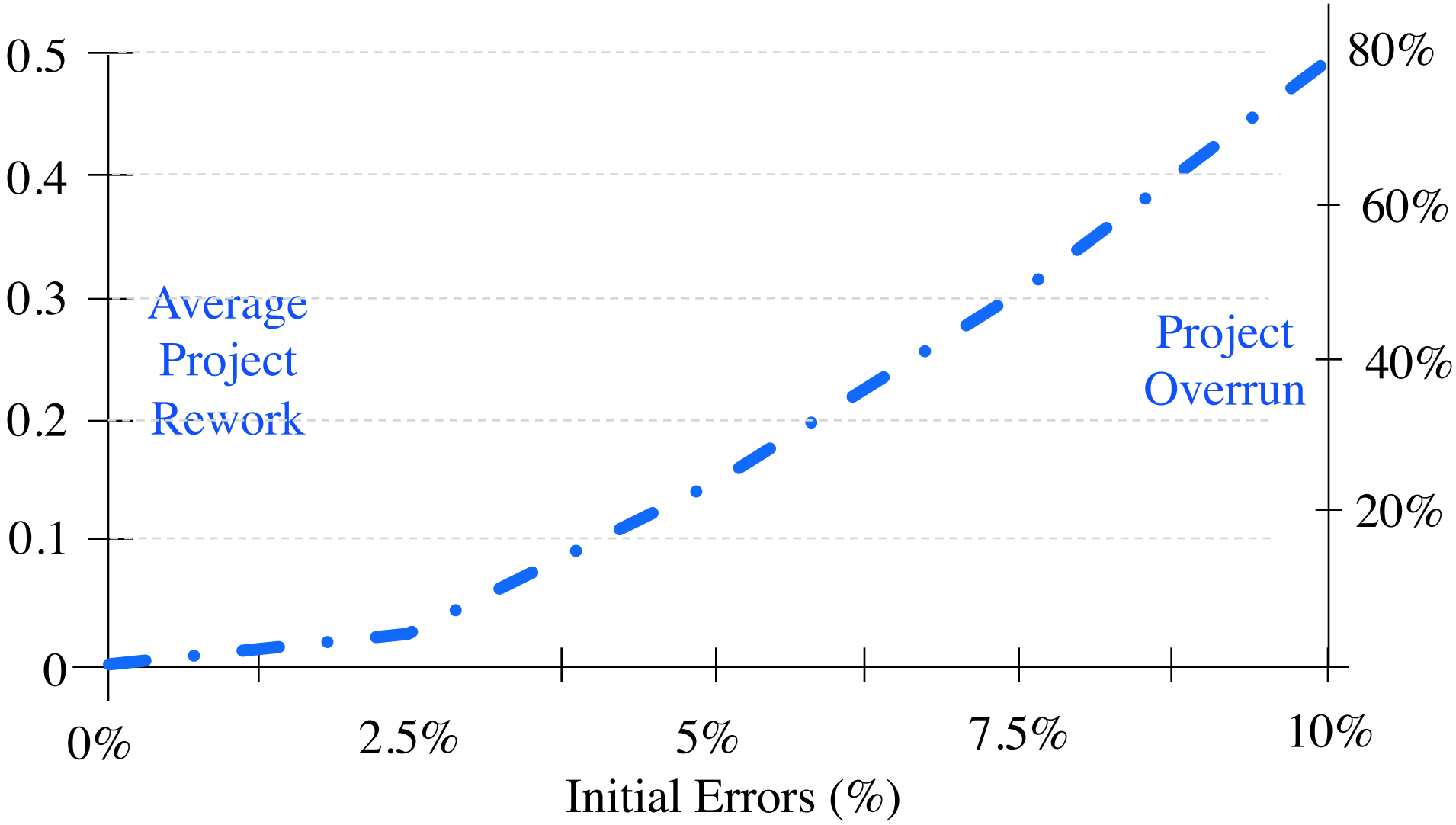
- In general:—
  - measure vital parameter(s) of whole system while it interacts with other systems in operational environment
  - vary internal parts, observe effect on whole
  - change each internal part in turn to slightly improve whole—cumulative selection
  - continue until no more improvement possible
  - test to detect/avoid local optima
- Best done using model

Get it Right First Time!



- 2.6 units of time to complete the 100% line (line 4)
- 3.3 units to complete the 90% IEE line (line 3)—10% errors left in…
  Mean project rework turbulence, p = (1 - 2.6 / 3.3) = 0.30.
- 30% rework *throughout the project*. caused by 10% uncorrected errors in the specifications (Optimum = zero errors!!)

Initial Errors Vs. Project Duration*



* Derived from Dynamic SE Cost Model

Optima & Symmetry

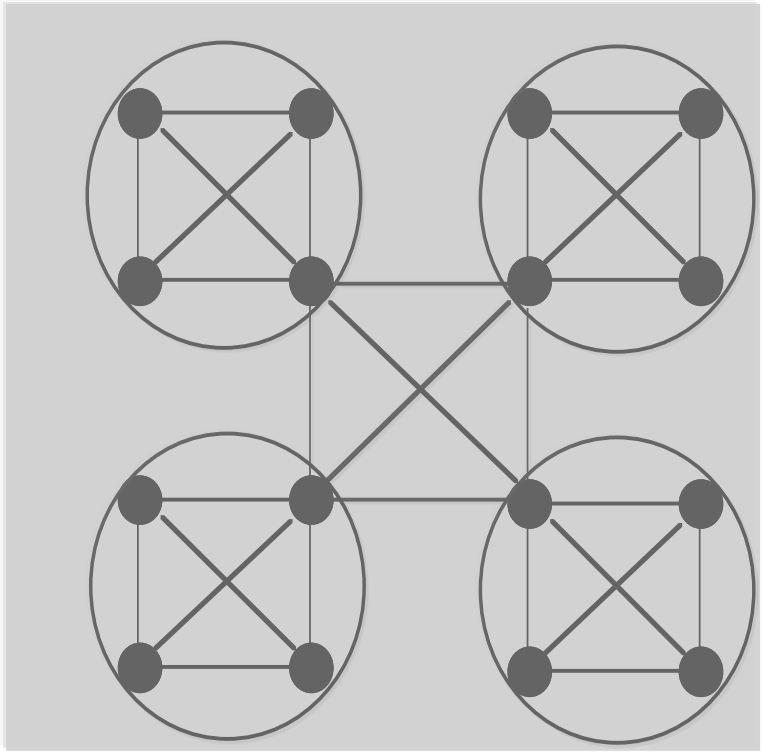


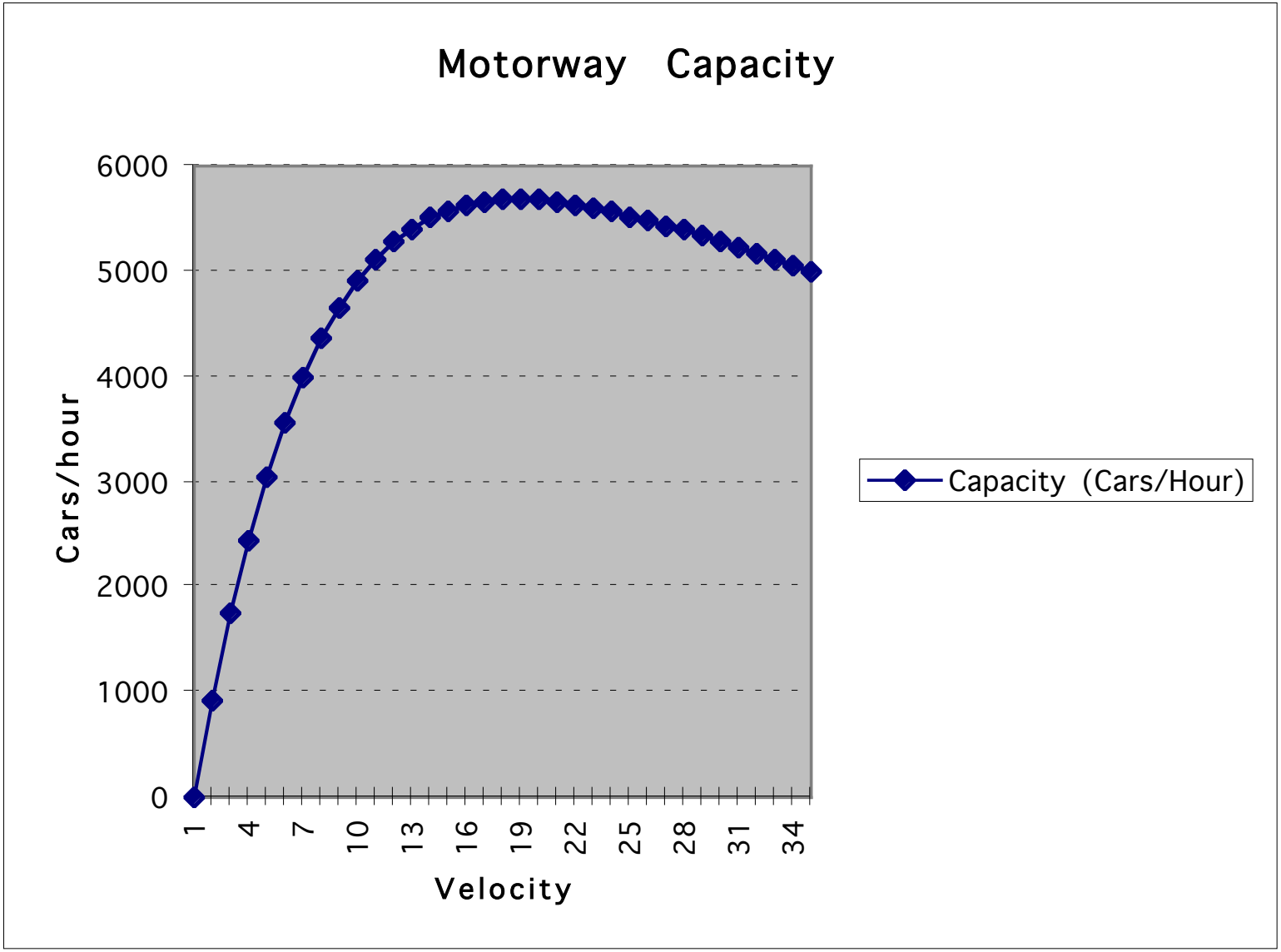
Diagram can be interpreted as a typical hierarchy with the centre entity removed, or as a flat structure with "nearest neighbour" connections—occurs widely in social activities

- 16 entities in a single cluster, fully connected = 240 connections
- 16 entities in 2 clusters of 8 employs 114 connections
- 16 entities in 2 clusters of 6 and 2 clusters of 2 employs 76 connections
- 16 entities in 8 clusters of 2 employs 72 connections
- 16 entities in 4 clusters of 4 employs 60 connections

Symmetry→simplicity→reduced connectivity→reduced entropy→reduced energy

N.B. Human eye-brain combination superlative at recognizing symmetry—no maths!

Optima—Maximum Motorway Capacity

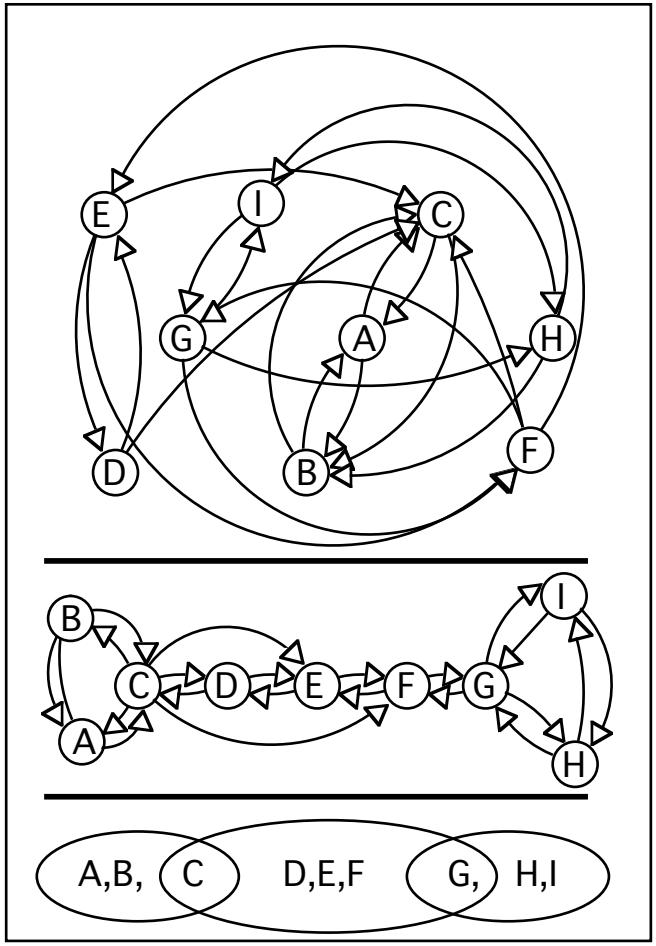


Untangling Connections

Tangled set of open, interacting systems

Untangled set

Perceived set of systems



E		1				1		1
	I		1		1			
		C		1			1	
	1		G		1			1
		1		A			1	
	1		1		H			
1		1				D		
		1		1			B	
1		1	1					F

A	1	1						
1	B	1						
1	1	C						
		1	D	1				
		1	1	E	1			
		1		1	F	1		
					1	G	1	1
						1	H	1
						1	1	I

Matrix scored by interface x distance from entity

Upper matrix scores 86.

Lower matrix scores 28

N.B. Note symmetry about leading diagonal in lower N^2 chart

Program uses genetic algorithm

Entropy Index

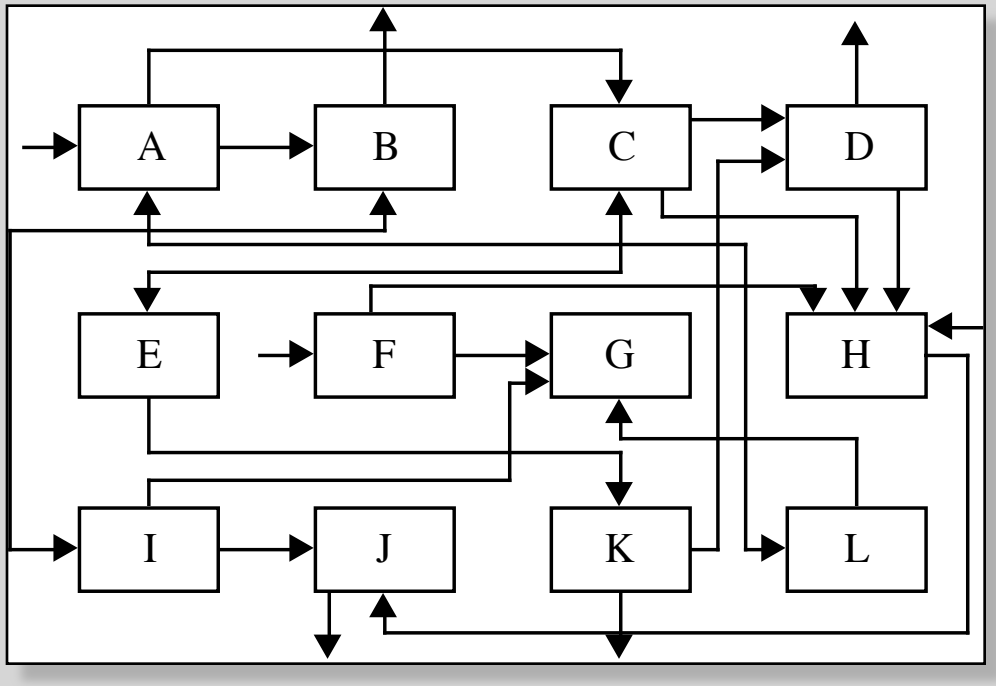
Before = 86/240 = 0.358

After = 28/240 = 0.1167

Practical Untangling

- A major airline's base storage depot has 12 groups of spares and consumables
  - Arranged in large rectangular hangar.
  - Items "picked" from groups as needed
- Individual "picks" visit one, two or more groups in sequence, according to the "pick-list".
- Pattern of "pick-movements" around the hangar uneven, some types occurring more frequently than others
- Unnecessary travelling around hangar, taking time and effort, potentially causing a safety hazard
- Rectangular room— only suitable space available. Can anything be done to improve the situation?

Pick-Group Example—before

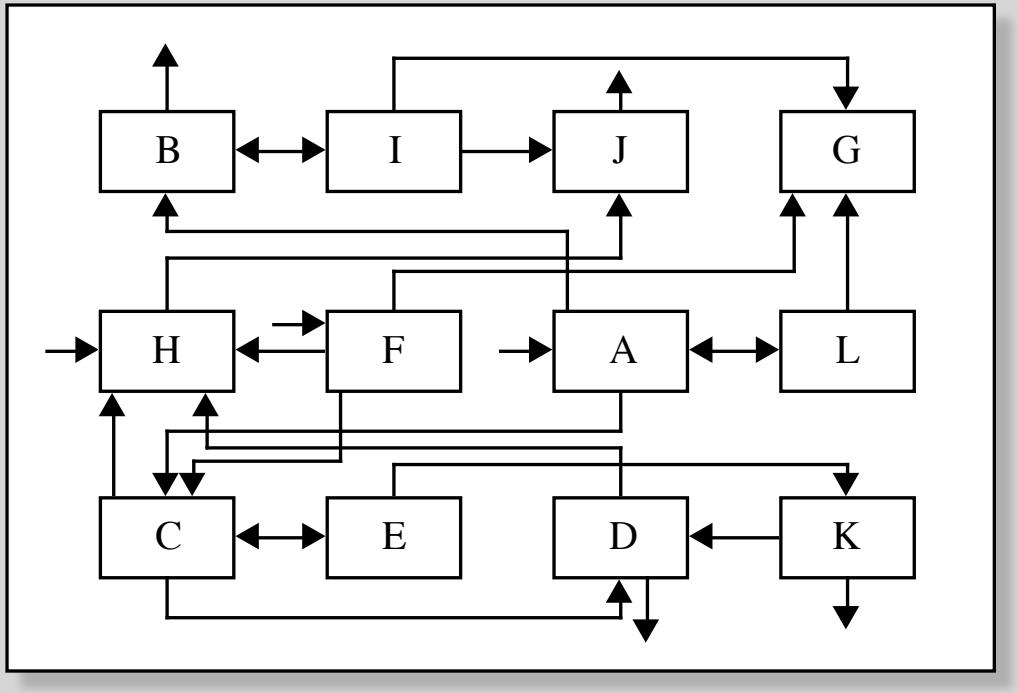


- Figure shows rectangular room with 12 pick-groups, A—L, and arrows showing principal movement paths

A	1	1									3
	B							2			
		C	1	1			3				
			D				2				
		3		E						3	
					F	1	1				
						G					
							H		1		
	1					2		I	2		
									J		
			3							K	
1						2					L

- Matrix represents path-lengths between pick-groups A—L. Numbers represent path utilization e.g. 1 = low, 2 = moderate, 3 = heavy
- Travel index $=\Sigma_i$ (Path-length $_i$ * Utilization $_i$) for i = 1 to 12
- Travel index from matrix = 160

Pick-Group Example—after



- Matrix rearranged to reduce overall value of Travel Index from 160 to 56, a real reduction of 65% in the work of travelling between pick-groups
- Some separations increased, e.g. A to B, but overall path-length reduced from 79 to 36, i.e. by 54%
- Matrix score = f(Entropy)

- Figure shows pick-groups rearranged to maintain original movement logic, but reduce overall Travel Index
- Paths form "waterfall"

```
B 2
1 I 2 2
    J
      G
      2 L 1
1       3 A     1
      1     F 1
    1           H
                3 C 1 1
                  3 E   3
                2     D
                      3 K
```

Keeping Systems Engineering Alive…!

- Our cortexes seek to minimize perceived disorder
  - uncomfortable with uncertainty—represents future disorder.
  - dispel perceived uncertainty—produce detailed plans, process models for projects.
  - pretend plans represent future as prescribed and ordained
- Demonstrating true SE affords shortest time to market, optimum quality, efficiency and effectiveness, etc., ***makes no impact***
  - SE seeks optima that are continually changing,
  - continually revises plans
  - must cling to faith in fixed-price contracting, prescribed value-for-money projects with carefully-calculated costs and time-scales?
  - **faith unshaken through countless time scale and budget overruns**.
- Managers obsessed with control?
  - **fear of loss of control overrides all other concerns**
  - so, systems engineering **must be** prescribed, regulated, controlled…
  - to the point where it is no longer systems engineering at all, but some pale, ineffectual systematic rote shadow of the real thing.

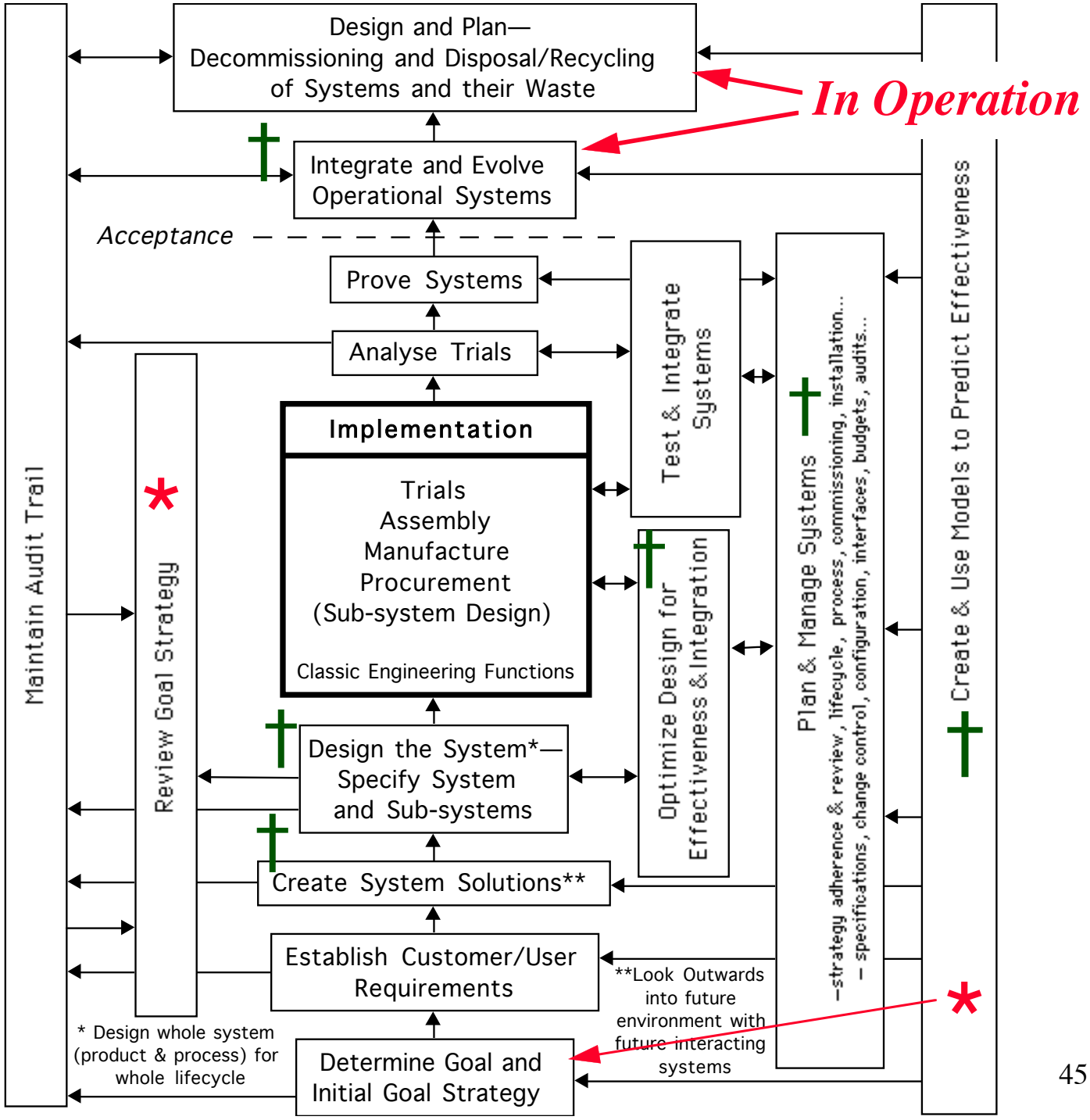
Beating the Reductionists at their Own Game…

- Recognize predominance of reductionism in our culture and education…go with the flow
- establish formal, repeated processes of context-sensitive optimization, not only of product but also of process…
- build repeated optimizing processes into project management practices, into operational use…
- Then…we could, perhaps, get the best of both worlds
  - linear, controlled process beloved of PMs
  - adaptable, flexible process to afford optimum performance

Auto-Adaptive Life Cycle Systems Engineering Process Model

* Continual review of Goal Strategy

† Continual Optimization



Some process model options

- Waterfall
- Sashimi (DeGrace et al, 1990)
- spiral
- evolutionary
- evolutionary acquisition
- regression
- chaos (Raccoon, 1995)
- concurrent
- goal-oriented (Hitchins, 1997)
- etc.,

Bak, P. & Chen, K, (1991), Self-organized Criticality, Scientific American, 264(1)

DeGrace, Peter, & Stahl, Leslie Hulet, (1990), Wicked Problems, Righteous Solutions, Yourdon Press

Hitchins, D.K. (1992) Putting Systems to Work, pp 209-210, John Wiley & Sons, Chichester

Hitchins, D. K., (1997), Systems Engineering the Pyramids, IEE Master Class

Raccoon, L.B.S., (1995), The Chaos Model and the Chaos Life Cycle, Software Engineering Notes, 20(1)

Womack, James P., Jones, Daniel T., & Roos, Daniel (1990), The Machine that Changed the World, Rawson Associates, New York