A Complementary Approach to Enterprise Systems Engineering

B. E. White, Ph.D.

The MITRE Corporation

26 October 2005

National Defense Industrial Association

8th Annual

Systems Engineering Conference

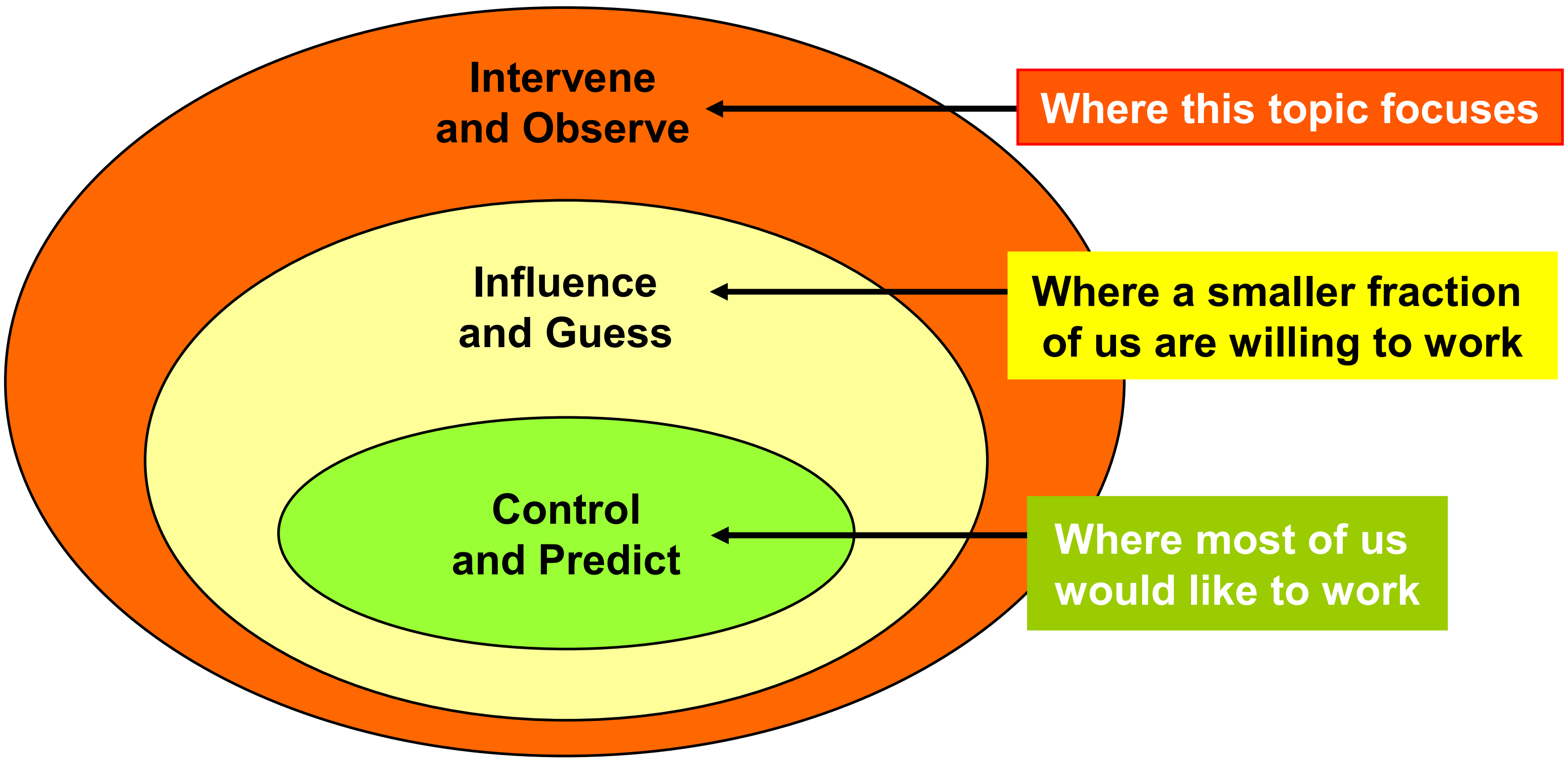
October 24-27, 2005

Hyatt Regency Islandia, San Diego California

Outline of Talk

- Purpose
- Definition of systems engineering terms
  - Traditional Systems Engineering (TSE)
  - Enterprise Systems Engineering (ESE)
  - Complex-System Engineering (CSE)
- Characterizing enterprise environments
- A regimen for CSE
  - Explanation of activities
  - Preliminary evaluations
- Summary

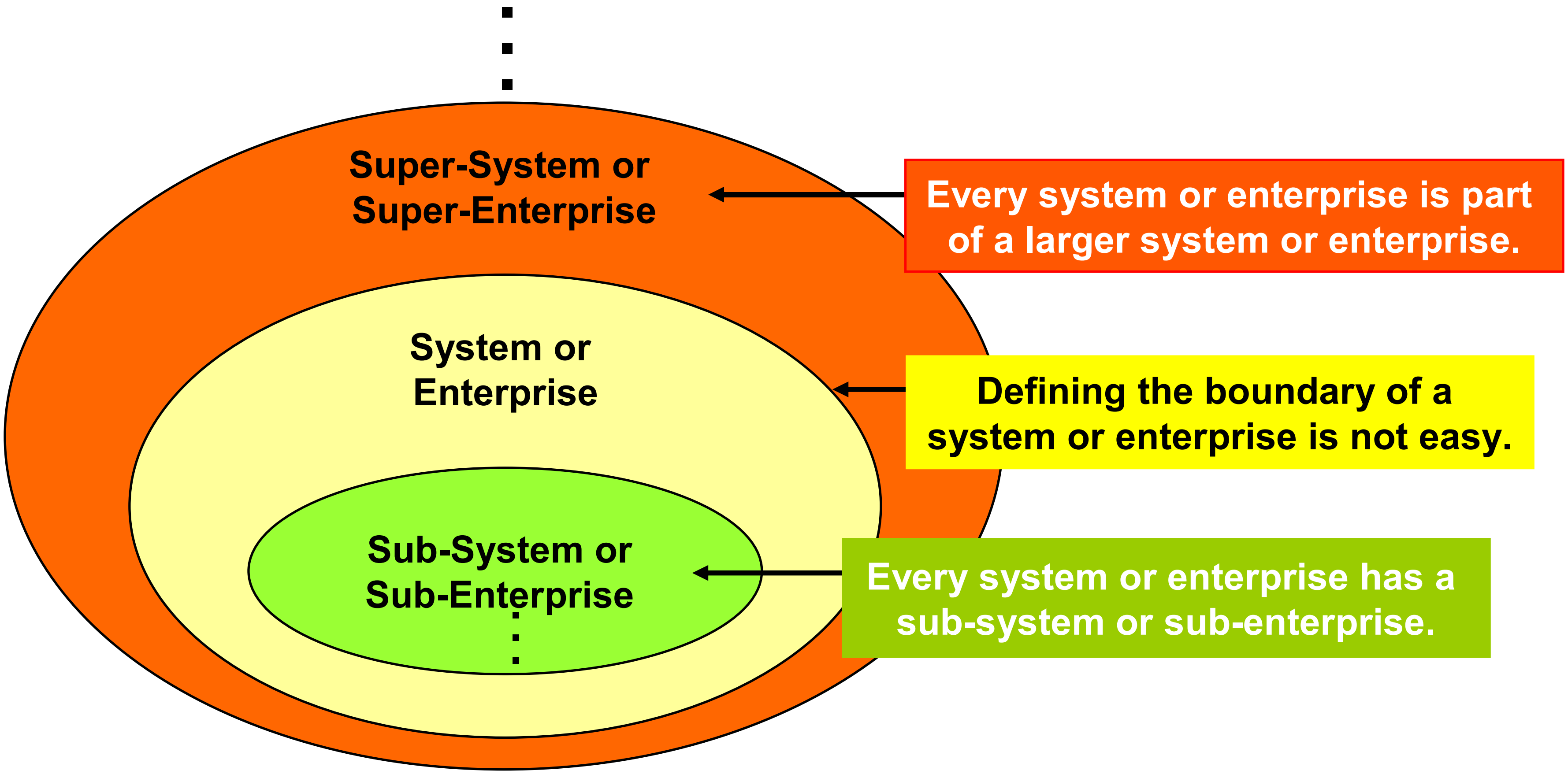
Context of This Talk



After [Gharajedaghi, 1999, p. 31]

Systems and Enterprises Are Nested – and Changing Their Boundaries Can Be Illuminating

See Notes Page

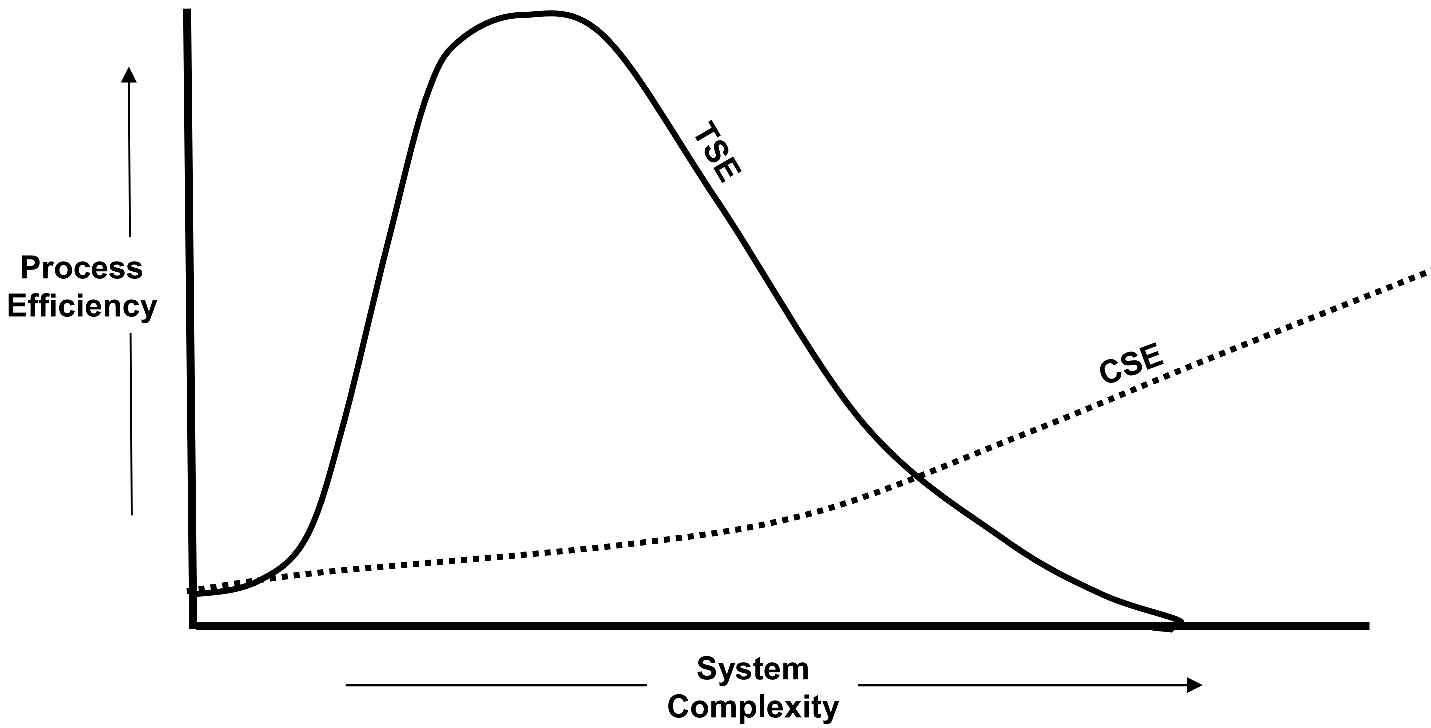


Some feel that no matter at what scale one is, in this nested structure, the same known SE techniques can be applied to effect good results.

Others say, no, depending on the scale in question, some radically different SE techniques may be needed.

Notional View of Applicability of TSE and CSE

Just as some believe that traditional system engineering can be successfully applied to every system, there will be those who believe that complex-system engineering is appropriate for every system.



Source: Mike Kuras

Motivation

- Of course, there is a continuum in thinking about this.
  - There's a whole spectrum of individuals between those taking a traditionalist view and those searching for new ways of systems thinking.
- We think it is important to offer a different mindset (the regimen) to
  - "Capture the imagination" of those open to it
  - Provide "food for thought" for those wedded to more conventional views.
- During the following it may help to become a little more humble
  - Reverse (or suspend) the assumption* that one can always pre-specify, predict, and control system or enterprise behavior and performance
  - Broaden your definition of systems engineering to include the management of "complex" environments that include people, organizations, etc.

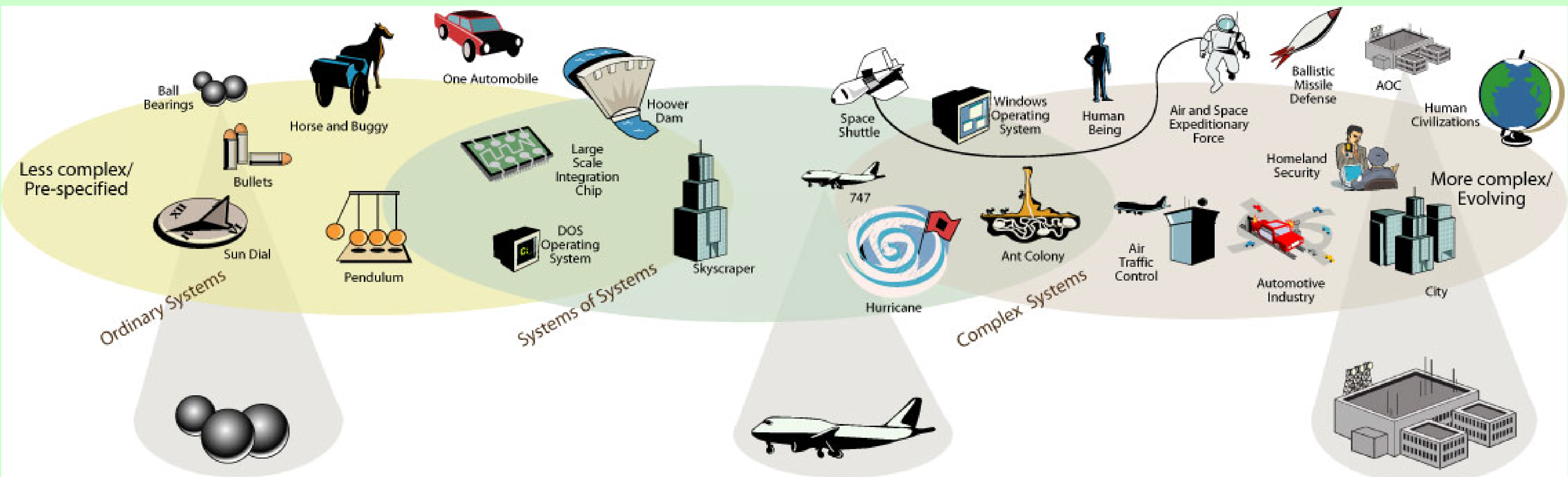
* [Johansson, 2004, pp. 53-57]

See Notes Page

A Spectrum of Systems

System: An instance of a set of degrees of freedom* having relationships with one another sufficiently cohesive to distinguish the system from its environment.**

*Normally grouped into subsets or elements **This cohesion is also called system identity



Less complex
Pre-specified

More complex
Evolving

[Kuras and White, INCOSE, 2005]

Distinguishing Attributes of Two Classes of Systems

Complex-systems	Non-complex systems
Unique	Identical and reproducible
Development and operation concurrent and continuous	Development and operations are separate and distinct
Emergence: development and operation at multiple scales	One predominant scale amenable to reductionist analysis and synthesis
Stochastic, unpredictable	Predictable at its predominant scale
Always open	Treatable as closed or with completely specified inputs
Learning and memory of prior history alters behavior	Repeatable transients
Requires both cooperation and competition to function effectively	Competition (for resources), friction and so forth reduce effectiveness

[Kuras, 2005]

Distinguishing Attributes of Two Classes of Systems (Concluded)

Complex-systems	Non-complex systems
Robust and broadly inefficient	Can be optimized and made efficient
Ambiguous and shifting boundaries	Well-defined, distinct boundaries at its predominant scale
Explores and tests *new* possibilities	Development progressively removes *unwanted* possibilities
Self-integrating and re-integrating	Integrated by external agents in one or more configurations
Dominated by transient and short-range relationships	Dominated by uniform and permanent relationships
Can exhibit relational networks at $O(n)$, $O(n^2)$, and $O(\sim 2^n)$	Can exhibit relational networks at $O(n)$ and $O(n^2)$
Hierarchies are partial and transient	Hierarchies are important, extensive, and durable

Assertion: Complex-systems can only be engineered by intervention, not by specification and then development.

[Kuras, 2005]

See Notes Page

Complex-Systems and CSE vs. Non-Complex Systems and TSE

- Complex-systems evolve naturally
  - Non-complex systems do not.
- Many organizations are complex-system enterprises. (see next chart)
- CSE creates/shapes environmental conditions which focus and accelerate actions of people/organizations.
- CSE is complementary to TSE.
- TSE is applicable to some of the parts of an enterprise.
  - TSE techniques should still be applied when appropriate.
  - TSE is not to be abandoned.

[Kuras and White, MIT, 2005]

See Notes Page

Enterprises

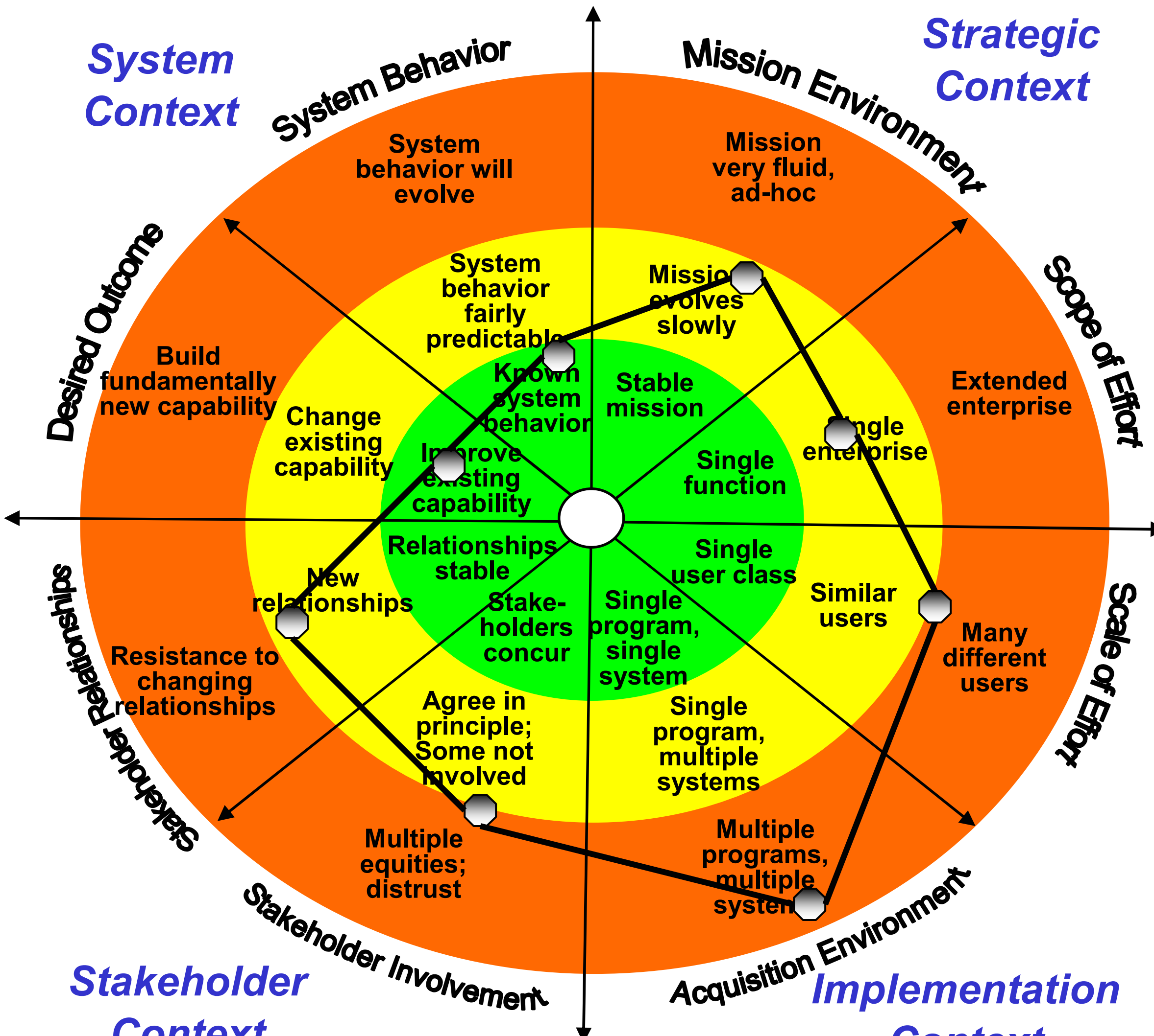
- **Enterprises are complex-systems functioning at multiple scales.**
  - **Scale: Combination of {field of view, resolution} plus {organizational, process, technical} aspects**
  - **Often "emergence" occurs & "patterns" appear when changing scales.**
- **Enterprises are characterized by *homeostatic** environments.**
- **Enterprise evolution is driven primarily by people/organizations acting autonomously but collectively.**
- **It is important and useful to characterize the enterprise's operational and developmental environment.**

---

* [Yates, 2002]

[Kuras and White, MIT, 2005]

ESE Environment Characterization Template



- **Typical program domain**
  - *Traditional systems engineering*
  - *Chief Engineer inside the program; reports to program manager*
- **Transitional domain**
  - *Systems engineering across boundaries*
  - *Work across system/program boundaries*
  - *Influence vs authority*
- **Messy frontier**
  - *Political engineering (power, control…)*
  - *High risk, potentially high reward*
  - *Foster cooperative behavior*

Source: Renee Stevens

MITRE

Regimen for CSE

- A regimen (not recipe) for CSE
  - Developed by SEPO's Mike Kuras
  - In paper presented at INCOSE's 2005 Symposium [Kuras-White, 2005]
- 8 CSE activities are advocated
  - Emphasize the Developmental Environment.
  - Shape Development During Operations.
  - Identify Outcome Spaces.
  - Establish Rewards (and Penalties).
  - Judge Actual Results.
  - Apply Developmental Stimulants.
  - Characterize Continuously.
  - Enforce Safety Regulations.
- The above activities are <u>not</u> independent of one another.

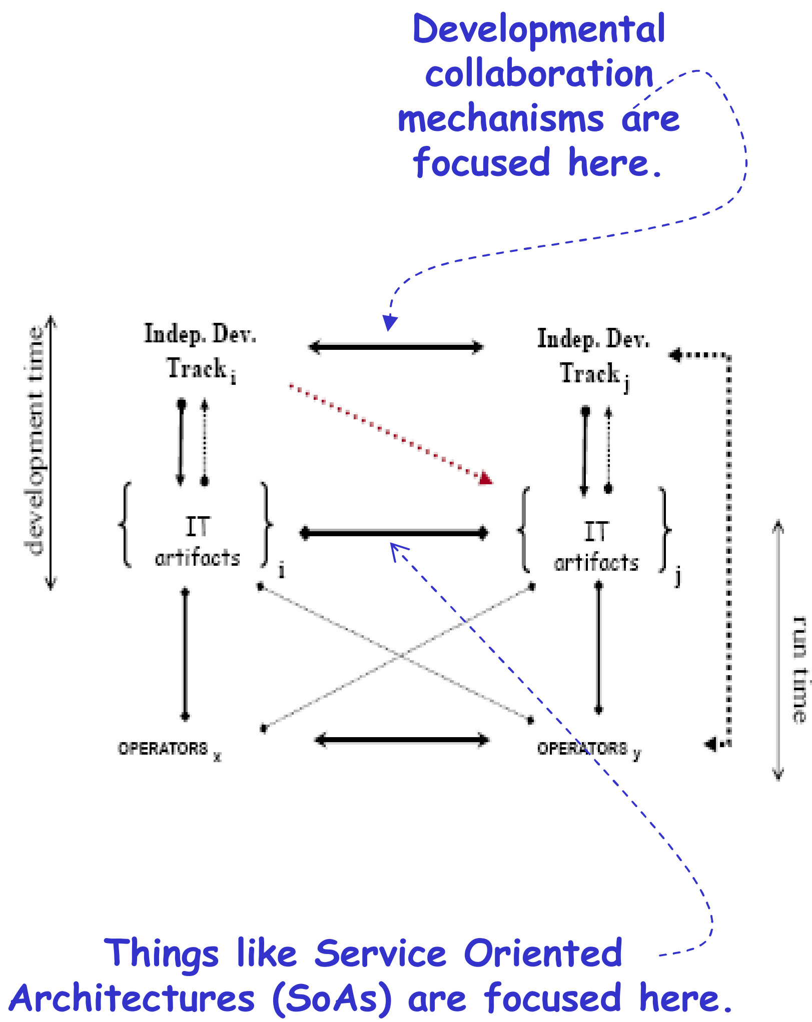
Emphasize the Developmental Environment

- Define, augment, and shape enterprise environment to be
  - Conducive to change/evolution
  - Supportive of both cooperation and competition.
- Don't try to "build" the complex-system; it builds itself.
  - Heed "the gardener" (not "the watchmaker") metaphor.
    - If it doesn't rain…
    - If rabbits are eating the plants…
  - Understand "the shopping mall" metaphor.
- Methods for engineering environments are inherently open ended, e.g.,
  - Modulate the flux of developers, e.g.,
    - Establish stipends for participation
    - Ensure unfettered information exchange
  - Manage towards stability in the face of changes like people joining or leaving the environment.
  - Divert funds from contract awards to performance rewards.
  - Use both *in situ* environments and partially artificial extensions.

See Notes Page

Shape Development During Operations

- Development and operation overlap and occur simultaneously in a complex system. The life cycle is not development *and then* operations.
- Engineering should be applied to operations as well as to development.
- Interoperability at different scales requires different mechanisms.
- Provide mechanisms for developmental collaboration across the enterprise.
- Examples
  - Involve operators in development (JEFX, JWID, ADOCS, etc.)
  - Involve developers in operations (Joint STARS in '91)



See Notes Page

Identify Outcome Spaces

- Identify and formulate broad Outcome Spaces that appeal to many enterprise participants, not narrow and specific outcomes.
- Focus and shape evolution while focusing on goals; do not try to pre-specify an end-state.
  - Operational Outcome Spaces do not always directly inform development.
  - Developmental Outcome Spaces do not directly determine operations.
  - If specific desired outcomes can be achieved directly by individual entities, then encourage competition.
  - If collective action is required to achieve outcomes, then encourage cooperation.
- Examples of "good" Outcome Spaces
  - U.S. Army's "Own the Night"
    - Not: Detailed specifications for night-vision goggles
  - The "X-Prize"
    - Take a passenger into space, return to earth, and then repeat within a week with the same method.
  - 2005 DARPA "Grand Challenge"
    - Advance technologies that will save the lives of our uniformed men and women on the battlefield.
  - Neutralize hostile cruise and ballistic missile threats to the U.S.
    - Destroy any/all incoming cruise and ballistic missiles before impact.

See Notes Page

Establish Rewards (and Penalties)

- It is assumed that each autonomous agent of an enterprise
  - Makes decisions and takes actions to achieve what they perceive as desired outcomes
  - Is motivated by externally applied rewards and penalties
- These actions determine enterprise change/evolution.
- Rewards should link specific populations of operators and/or developers to Outcome Spaces.
  - Create financial and other types of incentive opportunities for groups of independent contractors, not for individual programs.
- Rewards
  - Influence, but do not specify, decision making outcomes
  - Can accelerate enterprise change/evolution
- Achievement Rewards are not contract awards.
  - Typically contracts are awarded before outcomes are achieved.
  - Rewards are for performance and not the plausibility of promises.
- Example of a "good" Reward
  - $10 million and a plaque for the X-prize

See Notes Page

Judge Actual Results

- Judging is the explicit assignment of Rewards to appropriate autonomous agents for actual outcomes achieved.
- The Judging activity of the CSE regimen
  - Ties Rewards to actual outcomes
  - Provides opportunities to “weed the garden”
  - Completes Outcome Space–to–Rewards–to–autonomous agents linkage
  - Is tightly coupled to Development Environment and Rewards
- When change occurs in an enterprise, the acceptability of the change needs to be determined.
  - For example, change should not inhibit future change and should not prevent the enterprise from continuing to operate successfully.
  - A “healthy” enterprise does not become less “complex” as it evolves.
- Rewards for positive change should be allocated to those responsible for its achievement.
- Rewards modulate resource flows from the environment to the enterprise.
- Examples
  - X-Prize
  - DARPA Grand Challenges

Apply Developmental Stimulants

- Accelerate desired outcomes by stimulating autonomous agents to interact appropriately.
  - "Stir the pot" and/or "change the rules".
  - This is the most significant factor in accelerating enterprise evolution.
- Outside agents may be able to facilitate the necessary interactions, so inject additional autonomous agents as facilitators and brokers.
  - Example: MITRE as facilitator of "Cursor on Target (CoT)".
- Autonomous agents should be making "informed" decisions.
  - Endeavor to increase the frequency, intensity, and persistence of autonomous agent interactions.
- Developmental Stimulants are not outcomes.
  - They encourage autonomous agents to create outcomes for which they are mutually and not individually accountable.
- Pay for collective results; for example
  - Modify DD-250 Form to Reward a working, integrated system.
  - No autonomous agent (contractor team) gets paid for delivering a component system that is not successfully integrated.

See Notes Page

Characterize Continuously

- Capture and publish current "features" of the enterprise and its environment that seem to matter (e.g., Outcome Spaces and actual outcomes achieved, Rewards, and Judging results).
  - Help autonomous agents to "think globally but to act locally".
  - Focus on "now" and do not try to pre-specify the distant future.
  - Continuously refine these features to gain consistency in agent actions.
  - Ensure that accurate evaluation criteria and metrics are developed and publicized for refined levels of the features.
  - Avoid too much detail (refinement) because metrics and efforts may become localized and not support overall enterprise performance improvement.
  - Balance the continuing characterization of existing features with initiating the characterization of new features.
- Analogical examples
  - The daily stock market report
  - Highway traffic reports
  - Best/most recent Time Critical Targeting (TCT) times

Enforce Safety Regulations

- Safety Regulations focus on ensuring the continuous operation of the complex-system or enterprise – not on what it does or does not do.
  - Formulate and enforce rules that keep the enterprise functioning.
  - Develop and monitor measures of
    - "Fitness"
    - Measures of the rate of change
- Guard against complex-system failure modes: stagnation, disintegration, or collapse.
  - Absence of change may signal the potential death of the enterprise.
  - Ensure change can occur without destabilizing or destroying the enterprise.
- Examples
  - Criteria for vetting or training new autonomous agents as well as "weeding out" dysfunctional ones
  - Enforcing contractual obligations among autonomous agents
  - Managed redundancy/retirement
    - Microsoft's File Manager and Explorer
  - MIT Lincoln Laboratory's "off-line, in-line, on-line"

See Notes Page

In Summary, Who Does All This?

- People have asked
  - Who is responsible for making all this happen?!
  - Who actually "engineers the environment" of the enterprise to accelerate its evolution?
- These are good questions beyond the present scope.
- The CSE regimen is akin to enterprise "governance".
- This role of exercising the regimen can be taken by people with respect, authority, power, and "purposeful cohesion".
- It seems likely that this "governing body" would be external to the enterprise.

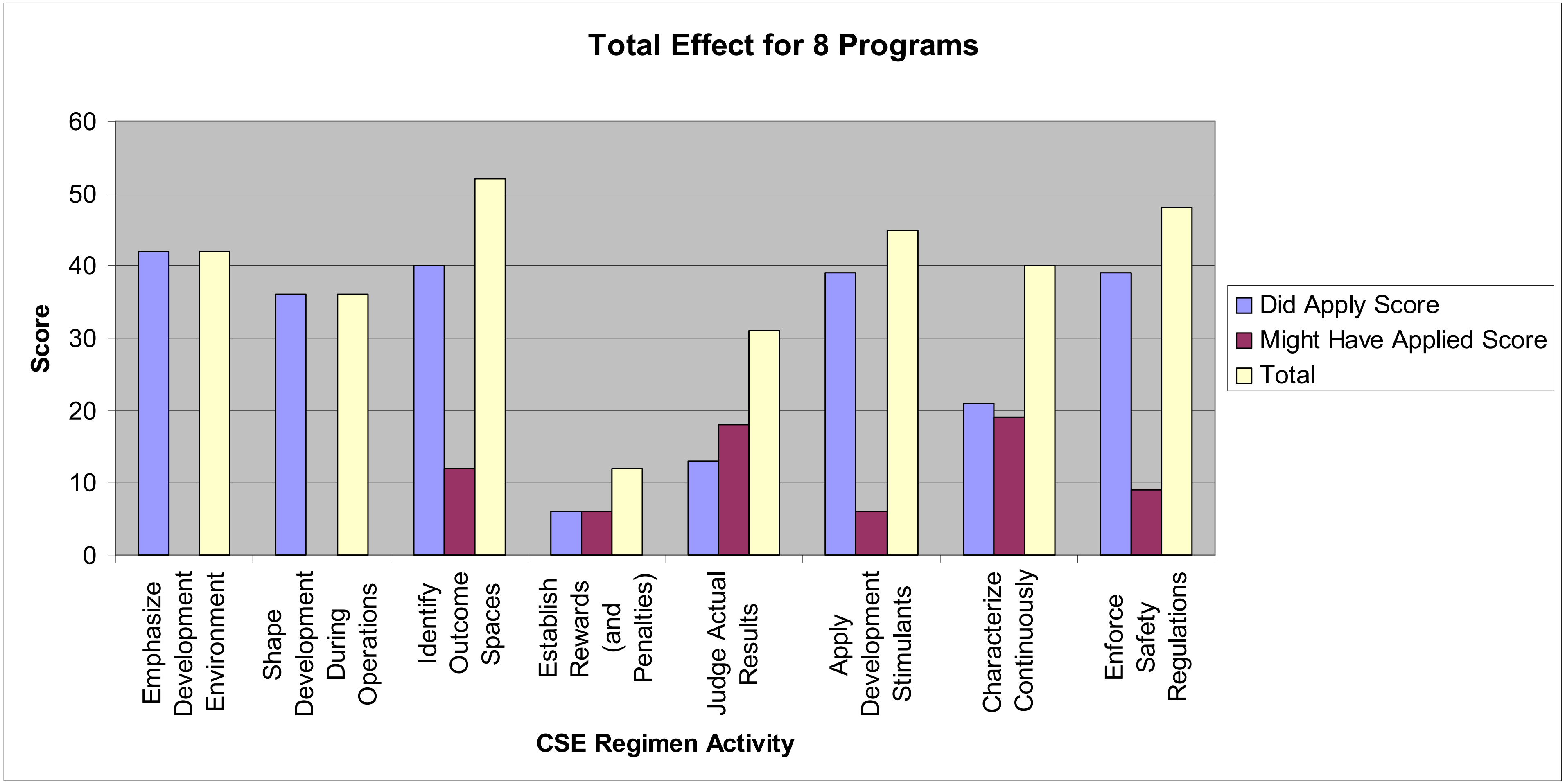
MITRE-Only 18-Feb CSE Workshop

- Purpose: Determine to what extent the CSE regimen applied to programs
- Methodology
  - Program experts provided basic information in advance
    - Program profile: program name, objective, sponsor, funding, years involved, type and number of contractors, etc.
    - Ratings on positive/negative impact of each regimen activity
  - Two hours were spent explaining/discussing the regimen.
  - Each expert briefed their program for about 30 minutes, focusing on "stories" about selected regimen activities.
  - The wrap-up discussion summarized overall impressions about applicability of regimen to programs.
  - Each expert revisited and revised their pre-meeting ratings afterwards based on what they learned during the meeting.
- Conclusions
  - The regimen applied (or could have applied) to most programs.
  - With few exceptions, the regimen had a positive impact.

MITRE Programs Involved

- Department of Defense Intelligence Information System
- National Airspace System Communications Modernization
- Air Operations Center Weapons System
- Americas Shield Initiative
- United States Visitor and Immigrant Status Indicator Technology
- Net Centric Enterprise Services
- Theater Battle Management Core System

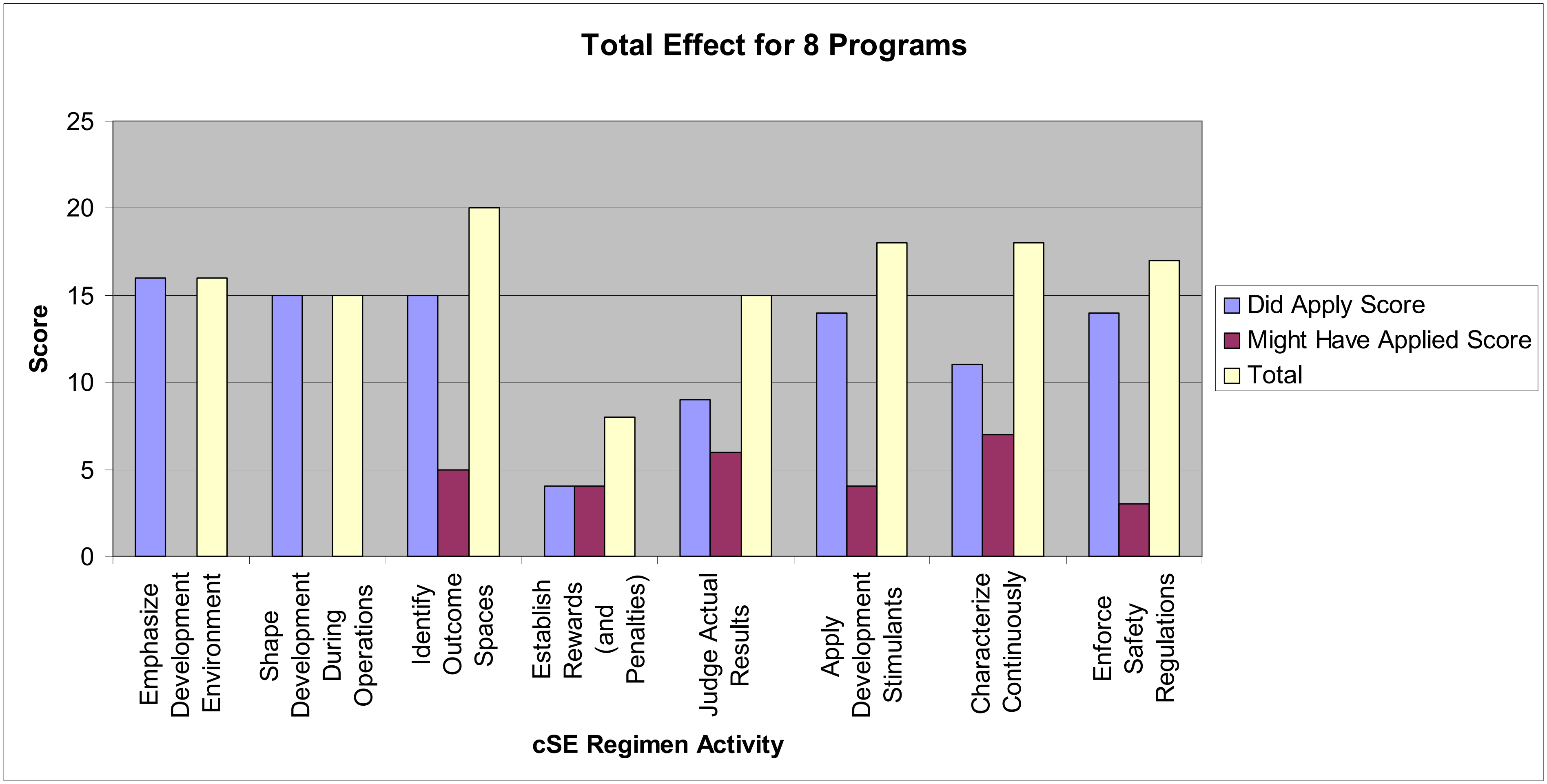
Numerical Results



<u>**Note on Individual Program *Impact Value* (whether positive or negative)**</u>

Major = 9
Significant = 3
Minor = 1

Numerical Results (Concluded)



<u>Note on Individual Program *Impact Value* (whether positive or negative)</u>

Major = 3
Significant = 2
Minor = 1

Summary

- A distinct mind-set for approaching CSE has been offered.
  - Concentrate on engineering the whole enterprise environment.
  - Continue to apply traditional SE techniques to individual systems.
- Terminology related to traditional and enterprise SE was gathered.
  - Definitions were crafted in an attempt to foster better understanding.
- A template for characterizing ESE environments was suggested.
- A CSE regimen for intervening in enterprise environments to achieve better outcomes was introduced.
  - Further work is needed to improve and validate the regimen.

References

Ackoff, Russell L., 1981, Creating the Corporate Future: Plan or be Planned For, John Wiley & Sons, Inc.

Axelrod, Robert, and Michael D. Cohen, 2000, Harnessing Complexity – Organizational Implications of a Scientific Frontier, Basic Books (Perseus Book Group), New York

Barabasi, Albert-Laszlo, 2003, Linked: How Everything Is Connected to Everything Else and What It Means, Penguin Group

Barnett, Thomas P. M., 2004, The Pentagon's New Map – War and Peace in the Twenty-First Century, G. P. Putnam's Sons

Bar-Yam, Yaneer, 2005, Making Things Work: Solving Complex Problems in a Complex World, NECSI Knowledge Press, 1st edition

Boardman, John, 24 March 2005, "Systems Thinking, System of Systems, and the Extended Enterprise," Keynote speech, Conference on Systems Engineering Research, Stevens Institute of Technology, Hoboken, NJ http://www.stevens-tech.edu/cser/

Brown, John Seely, and Paul Duquid, 2000, The Social Life of Information, Harvard Business School Press, Boston, MA

Gharajedaghi, Jamshid, 1999, Systems Thinking – Managing Chaos and Complexity: A Platform for Designing Business Architecture, Butterworth Heinemann, Boston, MA

Hughes, Thomas P., 1999, Rescuing Prometheus, Pantheon/ Random House, New York, ISBN 0-679-41151-8. Review: http://cstv.uwaterloo.ca/reviews/hughes.html

Johansson, Frans, 2004, The Medici Effect – Breakthrough Insights at the Intersection of Ideas, Concepts, and Cultures, Harvard Business School Press, Boston, MA

Kuras, M. L., 12 January 2005, "Introduction to Complex-system Engineering," for the Air Force (AF) Scientific Advisory Board (SAB); Presentation to Skip Saunders' subcommittee on System of Systems (SoS) Study

References (Continued)

Kuras, M. L., and B. E. White, 28 March 2005, “A Regimen for Complex-System Engineering (cSE),” Presentation to Graduate Class in Enterprise Architecture @ MIT – Engineering Systems Division (ESD)

Kuras, M. L., and B. E. White, 11 July 2005, “Engineering Enterprises Using Complex-System Engineering,” INCOSE 2005 Symposium, 10-15 July 2005, Rochester Riverside Convention Center, Rochester, NY

Magee, C. L., and O. L. de Weck, 20 June – 24 June 2004, “Complex System Classification,” Fourteenth Annual International Symposium of the International Council On Systems Engineering (INCOSE)
http://sepo1.mitre.org:8080/lcse/jsp/reference/Magee_and_deWeck_complex_system_classification.pdf

Malone, Thomas W., 2004, The Future of Work: How the New Order of Business Will Shape Your Organization, Your Management Style, and Your Life, Harvard Business School Press, Boston, Massachusetts
http://sepo1.mitre.org:8080/lcse/jsp/reference/the_future_of_work.doc

Moffat, James, September 2003, “Complexity Theory and Network Centric Warfare,” DoD Command and Control Research Program, CCRP Publication Series,
www.dodccrp.org
http://sepo1.mitre.org:8080/lcse/jsp/reference/Hill_Moffat_Complexity.pdf

Norman, D. O., and M. L. Kuras, January 2004, “Engineering Complex Systems, Chapter α,” The MITRE Corporation,
http://www.mitre.org/work/tech_papers/tech_papers_04/norman_engineering/index.html

Richardson, Kurt A., Jeffrey A. Goldstein, Peter M. Allen, and David Snowden (Editors), 2005, Emergence, Complexity and Organization, E:CO ANNUAL, Volume 6, ISBN: 0-9766814-0-4, ISCE Publishing, Mansfield, MA 02048

Senge, Peter M., 1990, The Fifth Discipline: The Art and Practice of the Learning Organization, Doubleday/Currency, New York, IBSN 0-385-26094-6

Stevens, Richard, Peter Brook, Ken Jackson, and Stuart Arnold, 1998, Systems Engineering: Coping with Complexity, Prentice-Hall

References (Concluded)

Sussman, Joseph M., 29-30 May 2002, "Collected Views on Complexity in Systems," Working Paper Series, ESD-WP-2003-01.06, ESD Internal Symposium, Massachusetts Institute of Technology, Cambridge, MA
http://sepo1.mitre.org:8080/lcse/jsp/reference/collected_views.pdf

Yates, F. Eugene, M.D. (Synopsis of talk), 11 November 2002, "From Homeostasis to Homeodynamics – Energy, Action, Stabilty [*sic*], Senescence,"
http://www.google.com/search?hl=en&lr=&q=Yates%2C+UCLA%2C+Homeostasis%2C+Homeodynamics&btnG=Search

Weinberg, Gerald M., 1975, An Introduction to General Systems Thinking, Wiley Series on Systems Engineering & Analysis, John Wiley & Sons Inc.

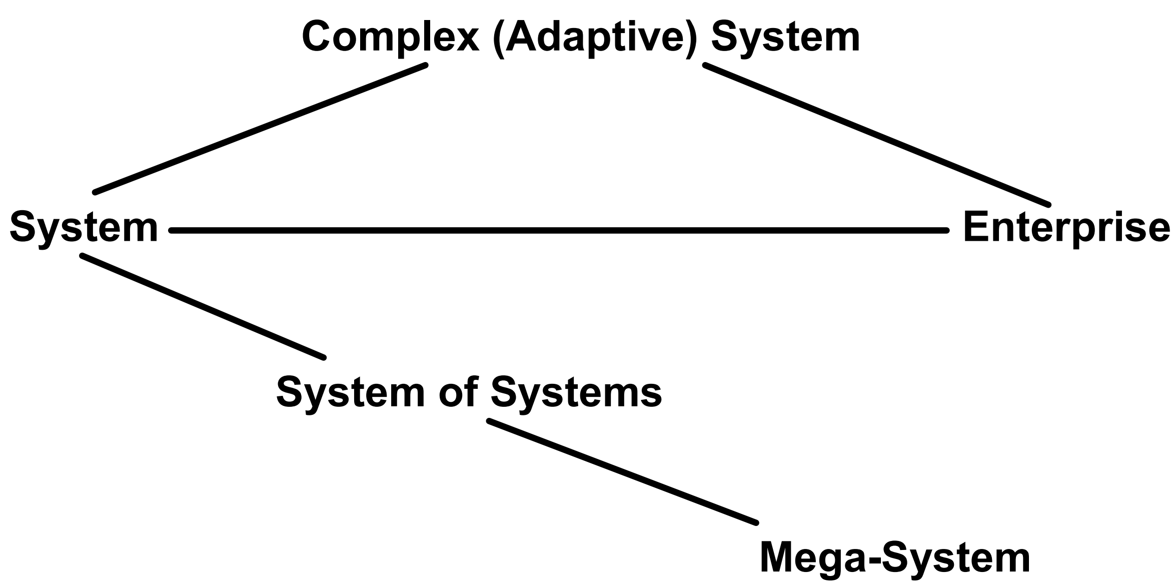
White, B. E., 14 June 2005, "Engineering Enterprises Using Complex-System Engineering (CSE)," Presentation to 1st Annual System of Systems (SoS) Engineering Conference, 13-14 June 2005, Frank J. Pasquerilla Conference Center, Johnstown, PA

White, B. E., 5 October 2005, "An Approach to Engineering Complex Systems," Presentation to New England Chapter of INCOSE, Best Western TLC Hotel, Waltham, MA

Backup Charts

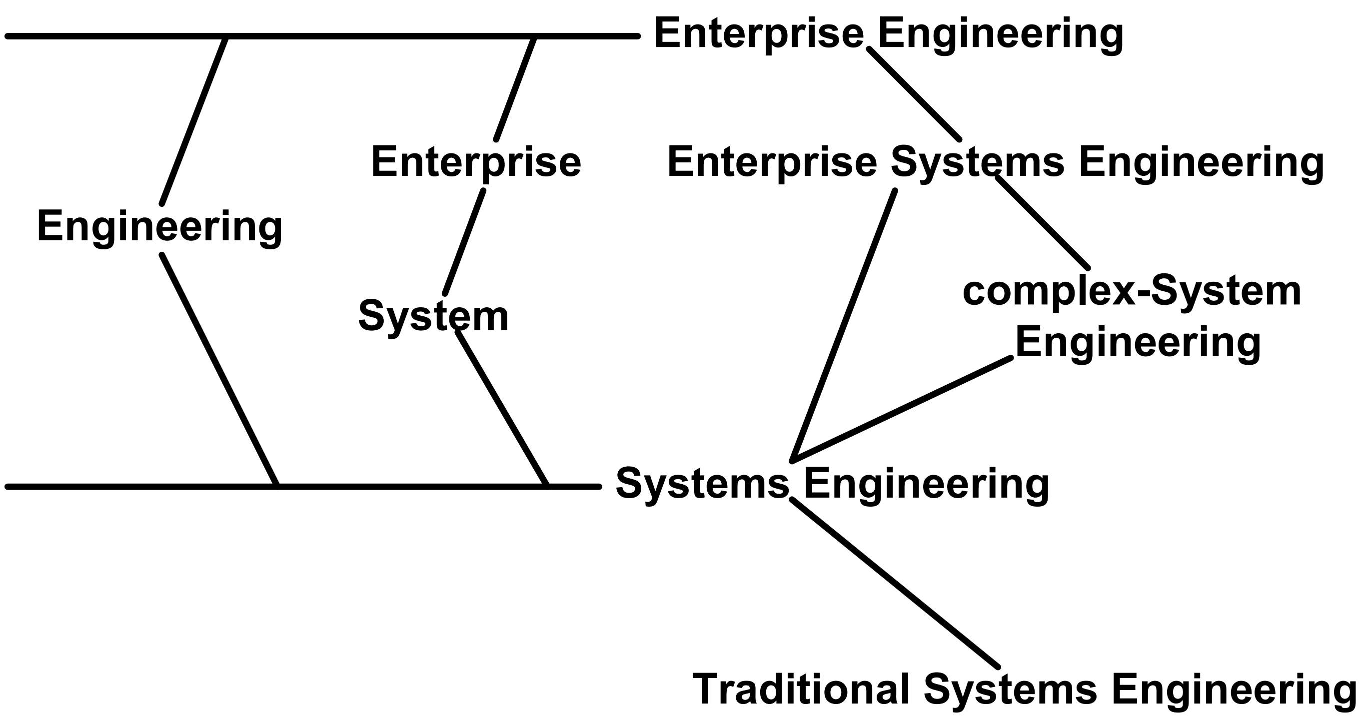
Definitions

System Definitions Diagram



Definitions (Continued)

Engineering Definitions Diagram



Definitions (Continued)

System: **An interacting mix of elements forming a whole greater than the sum of its parts.**

Features: These elements may include people, cultures, organizations, policies, services, techniques, technologies, information/data, facilities, products, procedures, processes, and other human-made or natural) entities. The whole is sufficiently cohesive to have an identity distinct from its environment.

Note: In this definition a system does not necessarily have to be fully understood, have a defined goal/objective, or have to be designed or orchestrated to perform an activity.

System of Systems (SoS): **A collection of systems that functions to achieve a purpose not generally achievable by the individual systems acting independently.**

Features: Each system can operate independently and is managed primarily to accomplish its own separate purpose. A SoS can be geographically distributed, and can exhibit evolutionary development and/or emergent behavior.

Complex System: **An open system with continually cooperating and competing elements.**

Features: This type system continually evolves, changing its behavior in response to itself and its external environment (often in unexpected ways). Changes between states of order and chaotic flux are possible. The relationships of the elements are imperfectly known, and are difficult to describe, understand, predict, manage, control, design, and/or change.

Notes: Here "open" means free, unobstructed by artificial means, and with unlimited participation by independent agents and interactions with the system's environment. Also, a complex system that is entirely natural is not an enterprise (see below).

Enterprise: **A complex system exhibiting a relatively stable equilibrium among many interdependent component systems in a shared human endeavor.**

Features: An enterprise may be embedded in a more inclusive complex system. External dependencies may impose environmental, political, legal, operational, economic, legacy, technical, and other constraints.

Notes: According to this definition, an enterprise need not include an agreed-to or defined scope/mission and/or set of goals/objectives. In addition, there is no attempt to include what is necessary to embody a successful enterprise; that is a different topic, i.e., enterprise engineering and enterprise systems engineering (see below).

Definitions (Continued)

Engineering: **Methodically conceiving and implementing solutions to real problems, with something that is meant to work.**

Note: This definition does not imply that the problems are always solved.

Enterprise Engineering: **Application of engineering efforts to the enterprise with emphasis on enhancing capabilities of the whole and understanding the relationships and interactive effects among the components.**

Note: This definition does not necessarily imply that the "best" efforts are applied. (See enterprise systems engineering on next chart.)

Systems Engineering: **An iterative and interdisciplinary management and development process that defines and transforms requirements into an operational system.**

Features: Typically, this process involves environmental, economic, political, and social aspects. Activities include conceiving, researching, architecting, utilizing, designing, developing, fabricating, producing, integrating, testing, deploying, operating, sustaining, and retiring system elements.

Notes: The customer for or user of the system usually states the initial version of the requirements. The systems engineering process is used to help better define and refine these requirements. Further, often the requirements change as further decisions are made as a result of systems engineering. Hence, for conciseness, the use of the single word "defines". This definition does not imply that a successful system is always realized. The word "integrated" is not included in this definition because systems engineering efforts may not be that well integrated.

Definitions (Concluded)

Traditional Systems Engineering (TSE)**: Systems engineering but with limited attention to the non-technical and/or complex system aspects of the system.**

Features: In TSE there is emphasis is on the process of selecting and synthesizing the application of the appropriate scientific and technical knowledge in order to translate system requirements into a system design. Here it is normally assumed and assured that the behavior of the system is completely predictable. Traditional engineering [not just TSE] typically is directed at the removal of unwanted possibilities.

Note: Here it is assumed that TSE is identical to "classical" systems engineering, i.e., customary and accepted methods of doing system engineering.

Enterprise Systems Engineering (ESE)**: A regimen for engineering "successful" enterprises.**

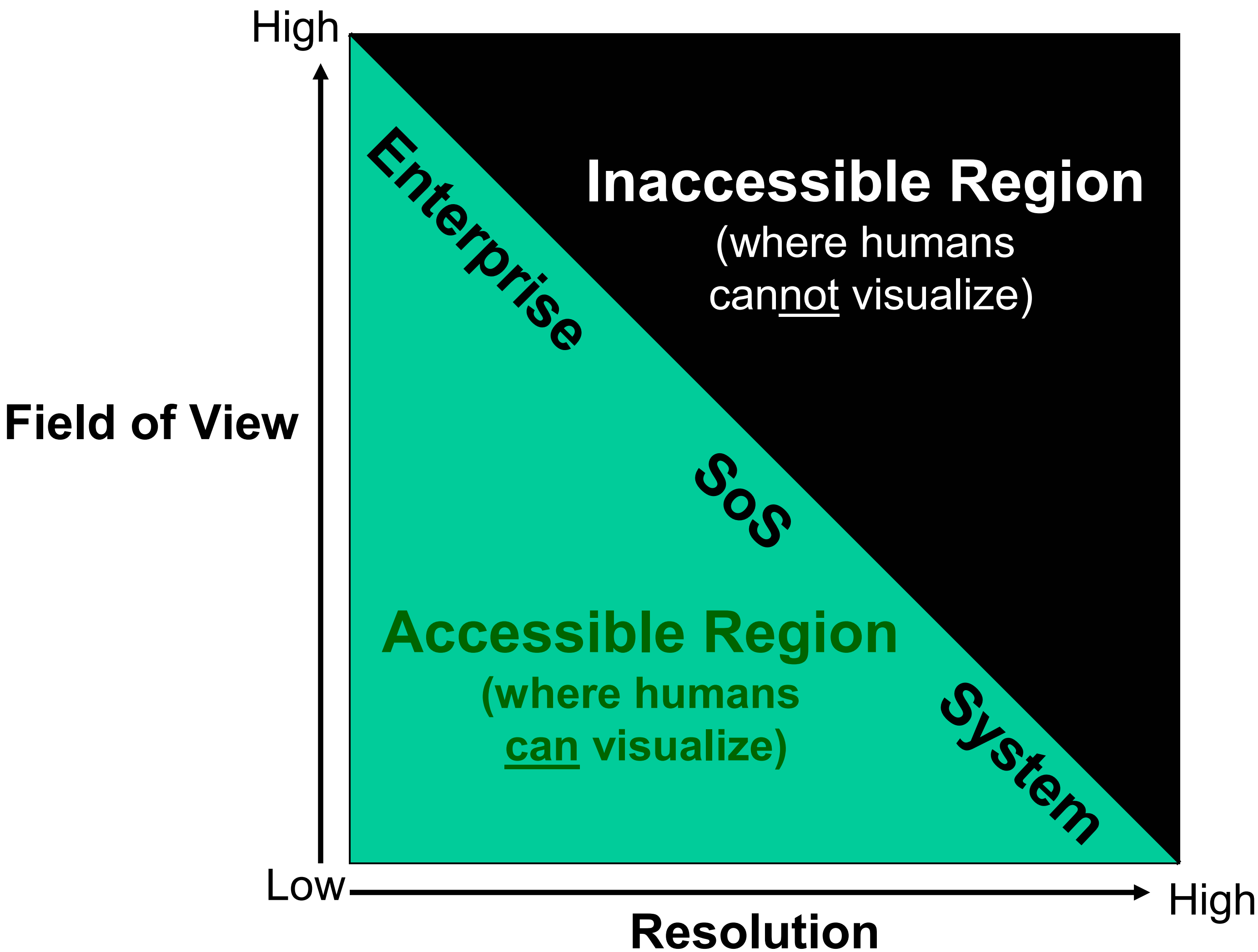
Features: ESE is systems engineering but with emphasis on that body of knowledge, tenets, principles, and precepts, having to do with the analysis, design, implementation, operation, and performance of an enterprise. The enterprise systems engineer concentrates on the whole as distinct from the parts, and its design, application, and interaction with its environment. Some potentially detrimental aspects of TSE are given up, i.e., not applied, in ESE.

Notes: Here "regimen" means a prescribed course of engineering for the promotion of enterprise success.

Complex-System Engineering (CSE)**: ESE but with additional conscious attempts to further open the enterprise to create a less stable equilibrium among many interdependent component systems.**

Features: In CSE, special attention is paid to emergent behavior, especially due to the openness quality, which can either be desirable or undesirable. One tries to instill the deliberate and accelerated management of the natural processes that shape the development of complex systems.

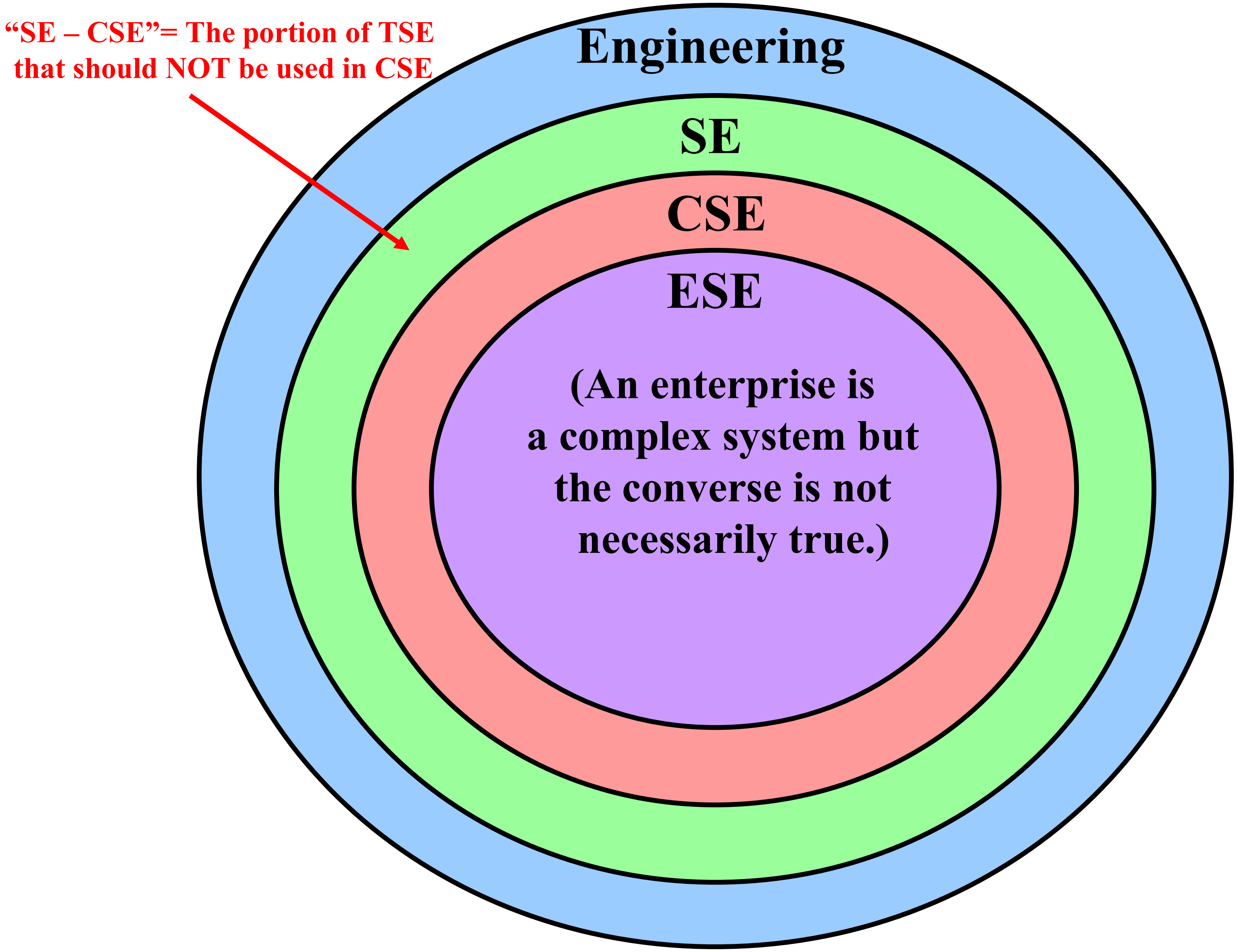
Multiscale View of Complexity



From: Kuras, M. L., and B. E. White, "Engineering Enterprises Using Complex-System Engineering," Paper for INCOSE 10-14 July 2005 Symposium, Rochester, NY

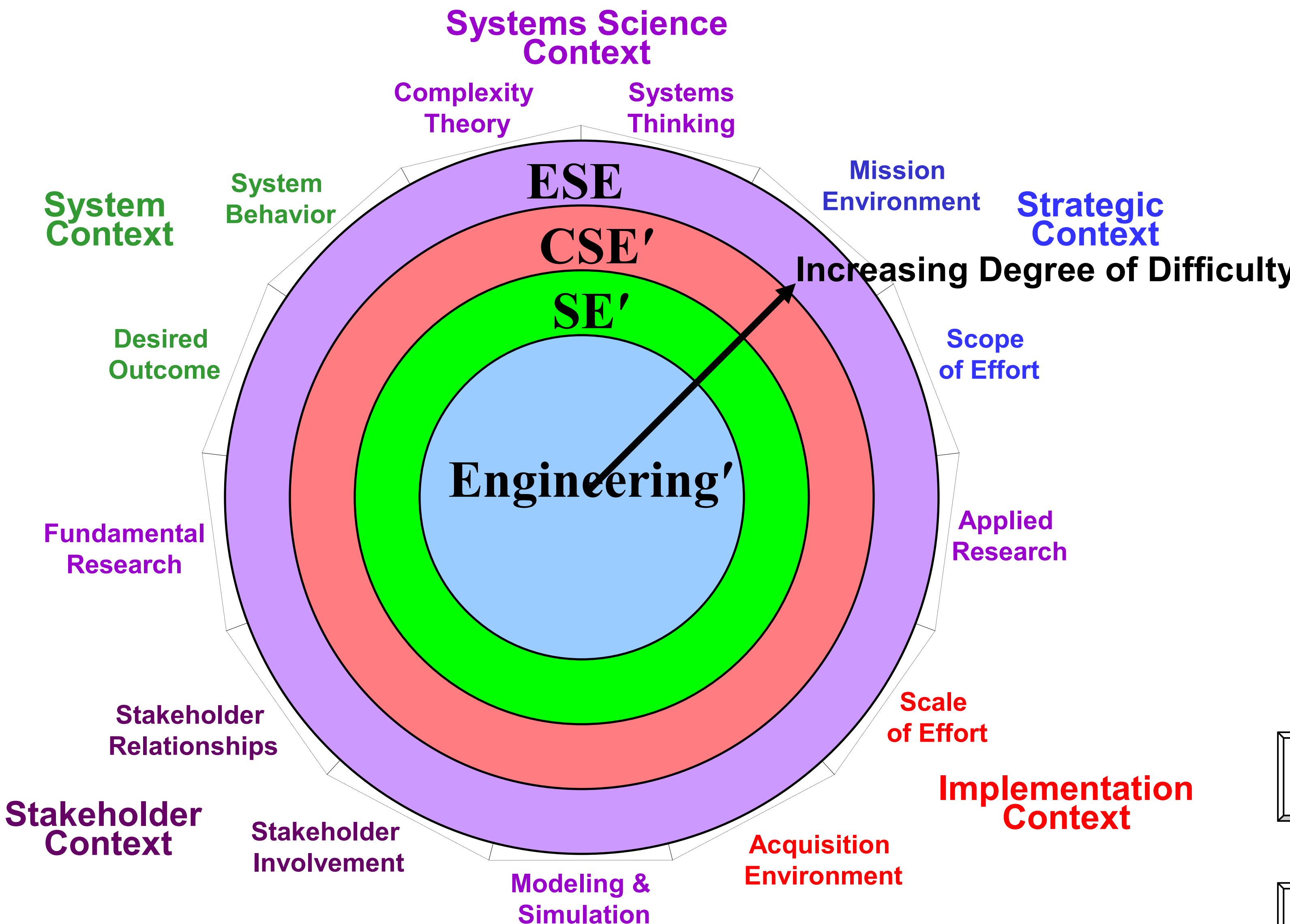
See Notes Page

Set Theory View of Engineering Disciplines



See Notes Page

Degree of Difficulty View of Engineering Disciplines*



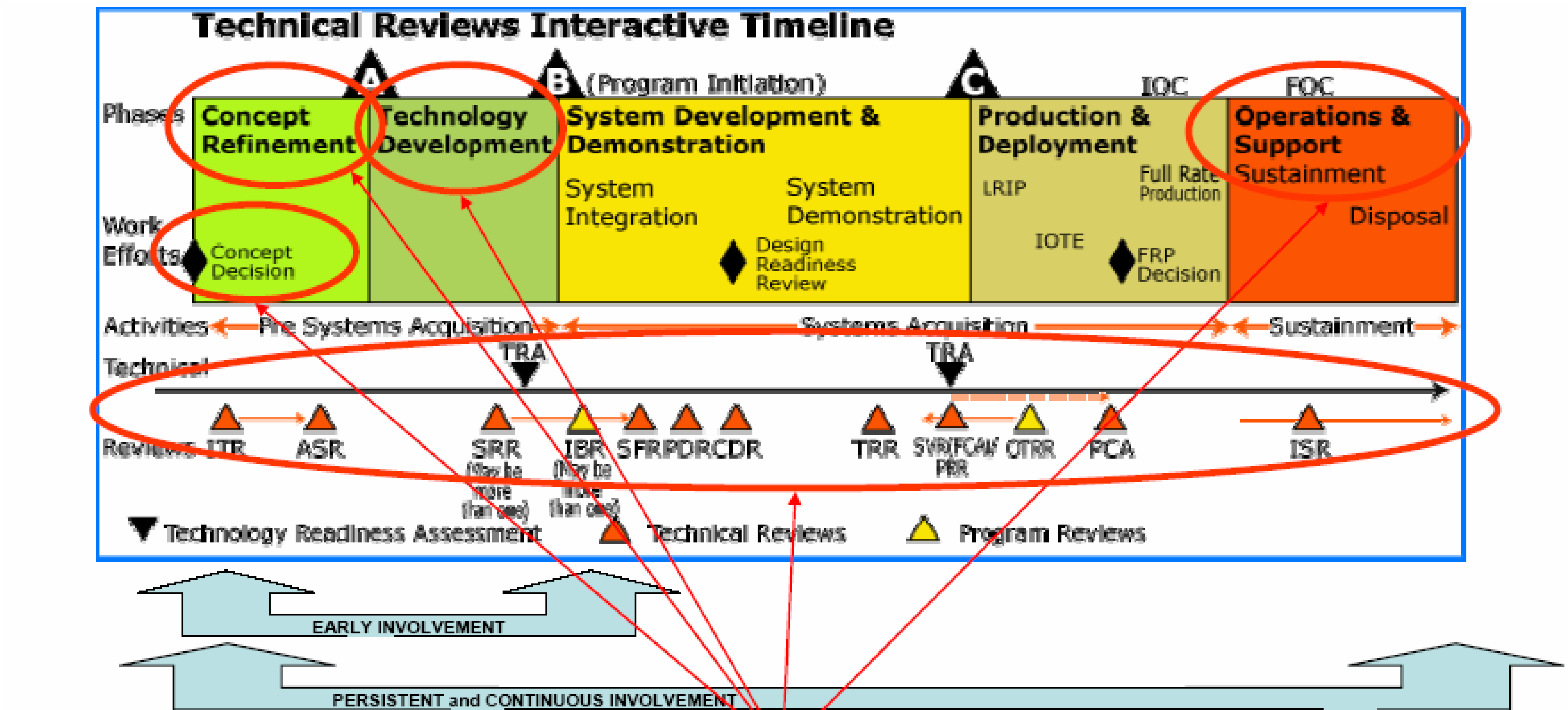
Notes:
Derived from Renee Stevens' template (see Chart 8)
These "rings" should be interpreted as "partitioned" versions of the rings of Chart 32, e.g., Engineering′ above is that portion of The whole Engineering set that is not included in the SE set, etc.

Traditional System Engineering

- Underpinnings of classical linear system analysis
- Hierarchical composition of separately engineered subsystems is common
- Addresses the form, fit, and function of a solution for a problem in two basic steps
  - First, functionality
  - Then, implementation
- Starts with “specifications”
  - Specifications are predictions that are made to come true.
  - Systems are built to “stand alone”.
- Predictions carry a lot of weight.
  - Plans, roadmaps, schedules, etc.
  - Developmental tests are planned independently of implementation.
  - When there is divergence, one tries to restore the validity of the predictions.
- Many detailed tools and procedures
  - Requirements analysis, allocation, and traceability
  - Functional analysis/synthesis, tradeoffs, abstractions, structuring and layering
  - WBSs, PERT and Gantt charts, etc.
  - Developmental processes (waterfall and spiral models, etc.), developmental and quality metrics, configuration control, etc.
  - Modeling/simulation, OSS&E, C4ISPs, ICDs
  - Technology surveys and risk management
  - Unit & integration testing, OT&E, MTPs, etc.
  - System architecting (operational views, employment views, technology views, materiel views, acquisition views, etc.)
- Many techniques are applied and refined successfully at the product level
  - Linearize non-linear problems (externalize memory; employ feedback).
  - More detail is always beneficial.
  - Iterate when possible.
  - Bottom-up and top-down convergence also helps.

Abridged from: Kuras, M. L., 12 Jan 05, “Introduction to Complex-system Engineering,” for the Air Force (AF) Scientific Advisory Board (SAB); Skip Saunders’ subcommittee on System of Systems (SoS) Study.

Driving SE Back Into Programs
[Good Systems Engineering Plans (SEPs) Are Key]



Increased use of disciplined SE, including formal technical reviews, to effectively address all technical issues

17

From: "Driving Systems Engineering into Programs," Mark D. Schaeffer, Principal Deputy Director, Defense Systems, Director, Systems Engineering, Office of the Under Secretary of Defense (AT&L), 23 March 2005, Keynote for CSER, Stevens Institute of Technology

MITRE

Abbreviations and Acronyms

ADOCS = Air Defense Operations Center System
AF = Air Force
AOC = Air Operations Center
ASR = Acquisition Strategy Report
AT&L = Acquisition, Technology, and Logistics
C4ISP = Command, Control, Communications, Computers, and Intelligence Support Plan
CCRP = Command and Control Research Program
CDR = Critical Design Review
CoT = Cursor on Target
CSE = Complex-System Engineering (or cSE)
CSER = Conference on Systems Engineering Research
CTC = Concurrent Technologies Corporation
DARPA = Defense Advanced Research Projects Agency
DoD = Department of Defense
DOS = Disk Operating System
ESD = Engineering Systems Division
ESE = Enterprise Systems Engineering
FOC = Full Operational Capability
FoV = field of view
FRP = Full Rate Production
IBR = Initial Baseline Review
ICD = Interface Control Document
INCOSE = International Council on Systems Engineering
IOC = Interim Operational Capability
IOTE = Initial Operational Test & Evaluation
ISR = Independent Safety Review
IT = information technology
ITR = Independent Technical Review
JEFX = Joint Expeditionary Force Experiment
Joint STARS = Joint Surveillance & Target Attack Radar System
JPDO = Joint Planning and Development Office
JWID = Joint Warfighter Interoperability Demonstration
MIT = Massachusetts Institute of Technology

Abbreviations and Acronyms (Concluded)

MTP = Maintenance Test Plan [or Package]
NDIA = National Defense Industrial Association
NECSI = New England Complex Systems Institute
O = order
OOS&E = Operational Safety, Suitability and Effectiveness
OT&E = Operational Test and Evaluation
OTRR = Operational Test Readiness Review
OUSD = Office of the Under Secretary of Defense
PCA = Physical Configuration Audit
PDR = Preliminary Design Review
SAB = Scientific Advisory Board
SE = systems engineering
SEP = Systems Engineering Plan
SEPO = Systems Engineering Process Office
SFR = System Functional Review
SoA = Service Oriented Architecture
SoS = System of Systems
SoSECE = System of Systems Engineering Center of Excellence
SRR = System Requirements Review
TCT = Time Critical Targeting
TRA = Technical Readiness Assessment
TRR = Technical Readiness Review
TSE = Traditional Systems Engineering (or System)

Speaker Contact Information

Brian E. White, Ph.D.

Telephone: (781) 271-8218

The MITRE Corporation

E-Mail: bewhite@mitre.org