

Process Improvement and CMMI®

- Developing Complex Systems- Using CMMI® to Achieve Effective Systems and Software Engineering Integration

8th Annual CMMI Technology Conference and User Group
November 17-20, 2008
Hyatt Regency Tech Center
Denver, Colorado

Theme: Investigation, Measures, and Lessons Learned About the Relationship Between CMMI® Process Capability and Project or Program Performance.

Dr. Kenneth E. Nidiffer
Director of Strategic Plans for
Government Programs
nidiffer@sei.cmu.edu
703.908.1117



Software Engineering Institute

Carnegie Mellon

© 2008 Carnegie Mellon University

The Software Engineering Institute - Improving the Practice of Engineering: Create, Apply and Amplify

Federally Funded Research and Development Center

Created in 1984

Sponsored by the U.S. Department of Defense

Locations in Pittsburgh, PA; Washington, DC;
Frankfurt, Germany

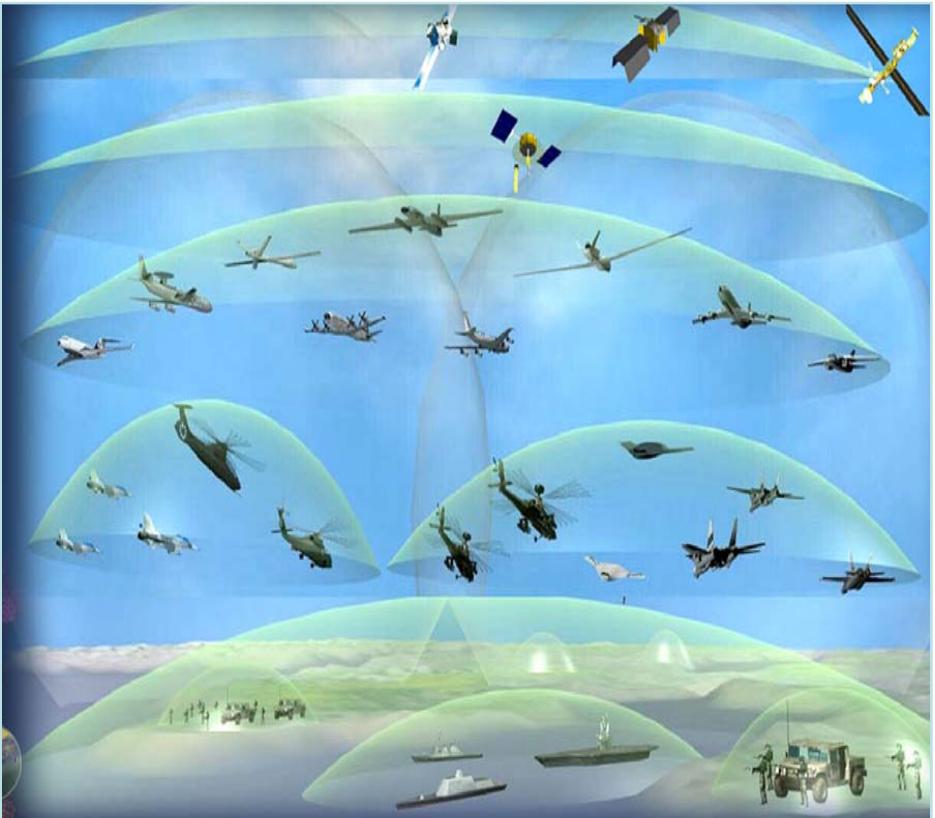
Operated by Carnegie Mellon University



Overview

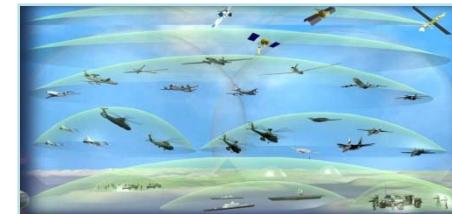


- **Integration Trends**
 - Development
 - Mission
 - Technology
 - Engineering
 - Risk
- **CMMI Benefits**
- **Ten Future Trends**
- **Wrap-up**



Development Complexity

Need for Space, Air, Ground, Water, Underwater Software-Intensive Systems to be Integrated



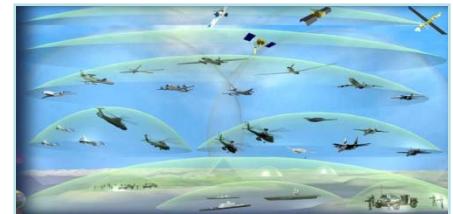
- Several million SLOC programs; “Hybrid” systems combining legacy re-use, COTS, new development
- Multi-contractor teams using different processes; dispersed engineering, development & operational locations
- New technologies create opportunities/challenges; products change/evolve, corporations mutate
- Business/operational needs change - often faster than full system capability can be implemented
- Skillset Shortfalls; Cost and schedule constraints
- Demands for increased integration, interoperability, system of system capabilities
- Enterprise perspectives/requirements; sustainment concerns



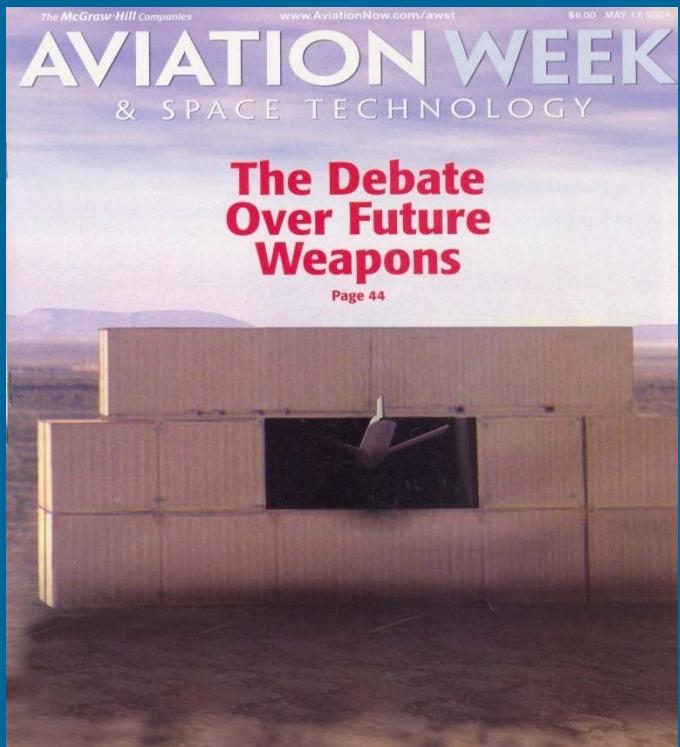
Development Complexity of Software-Intensive Systems is Increasing



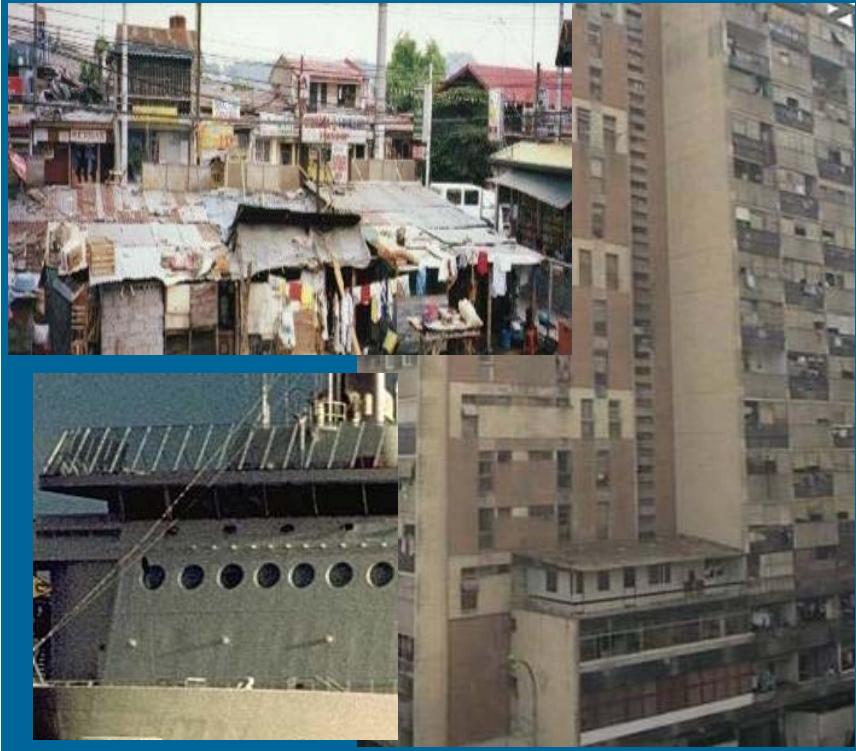
Need for Mission Integration



Less a Matter of Hitting
a Window



And More a Matter of
The Right Window - Right
Now



Software Engineering Institute

Carnegie Mellon

Developing Complex Systems – Using CMMI to Achieve
Effective Systems and Software Engineering Integration
Dr. Kenneth E. Nidiffer
© 2008 Carnegie Mellon University

Software Engineering Trends That Impact Systems Engineering



Traditional

- Standalone systems
- Mostly source code
- Requirements-driven
- Control over evolution
- Focus on software
- Stable requirements
- Premium on cost
- Staffing workable

Future

- Everything connected-maybe
- Mostly COTS components
- Requirements are emergent
- No control over COTS evolution
- Focus on systems and software
- Rapid change
- Premium on value, speed, quality
- Scarcity of critical talent

Emerging Dynamics of Bringing Systems and Software Engineering in Continued Partnership

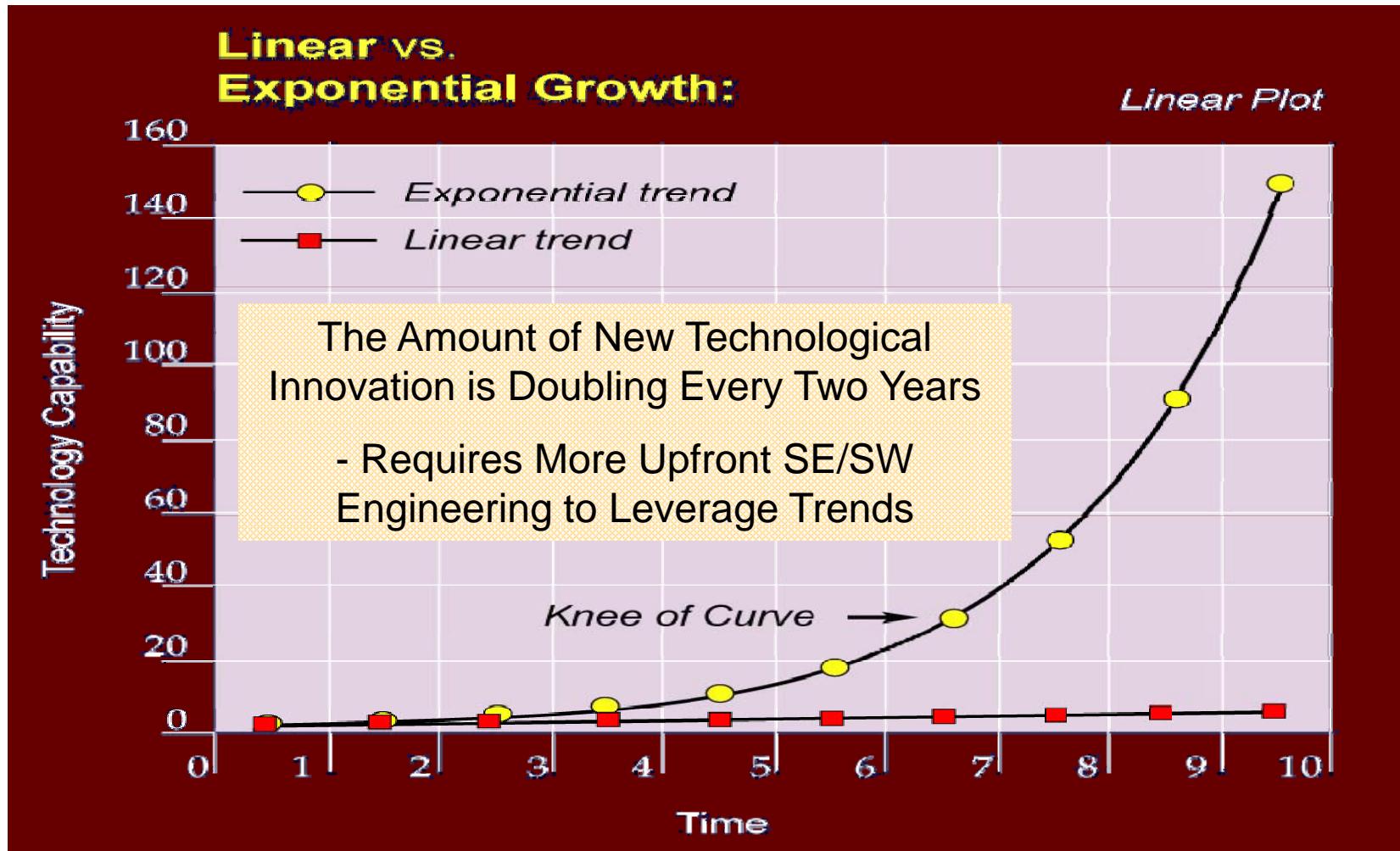


Software Engineering Institute

Carnegie Mellon

Developing Complex Systems – Using CMMI to Achieve Effective Systems and Software Engineering Integration
Dr. Kenneth E. Nidiffer
© 2008 Carnegie Mellon University

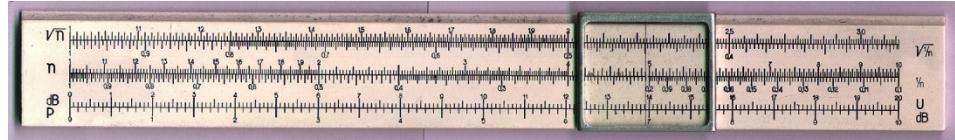
The Acceleration of Innovation in the 21st Century: - Facilitating Our Ability to Integrate



Facilitating Integration: Augustine's Law - Growth of Software is an Order of Magnitude Every 10 Years



In The Beginning



1960's



F-4A
1000
LOC



1970's



F-15A
50,000
LOC



1980's



F-16C
300K
LOC



1990's



F-22
1.7M
LOC



2000+



F-35
>6M
LOC



Software Engineering Institute

Carnegie Mellon

Developing Complex Systems – Using CMMI to Achieve Effective Systems and Software Engineering Integration
Dr. Kenneth E. Nidiffer
© 2008 Carnegie Mellon University

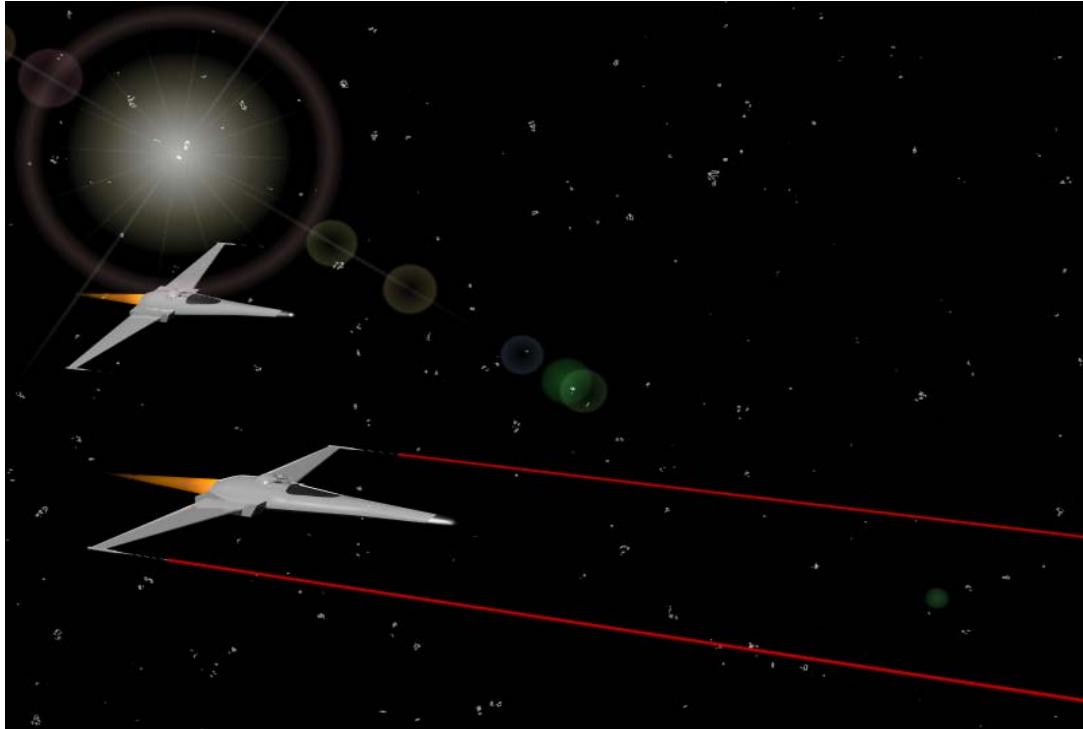
Facilitating Integration: Given Augustine's Law Holds



2080?



F-50 - 4.7B Lines of Code



Need for increased functionality will be a forcing function to bring the fields of software and systems engineering closer together

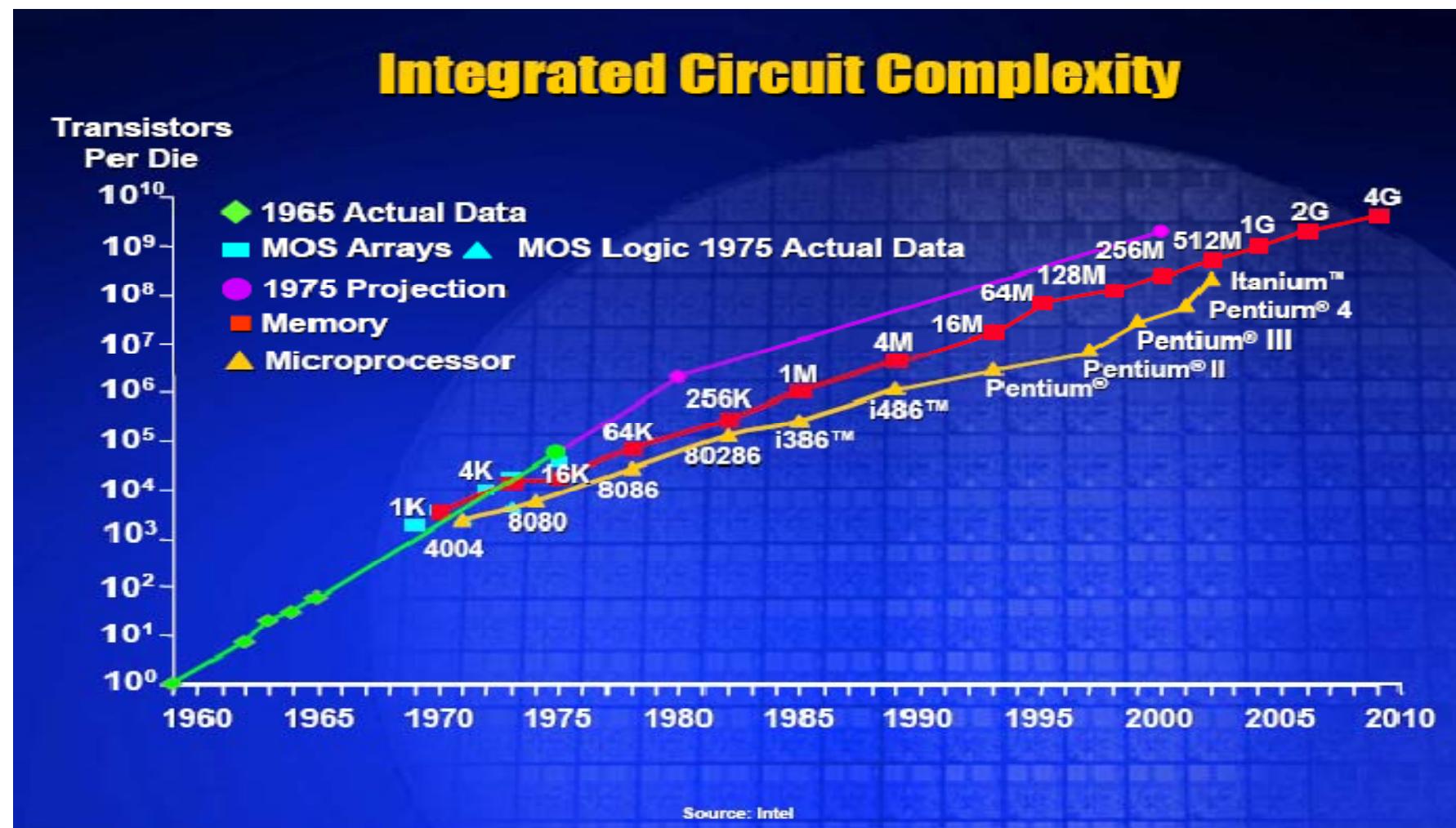
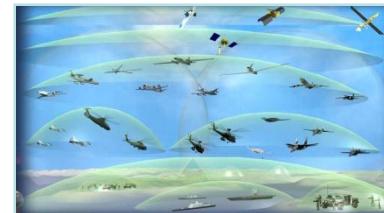


Software Engineering Institute

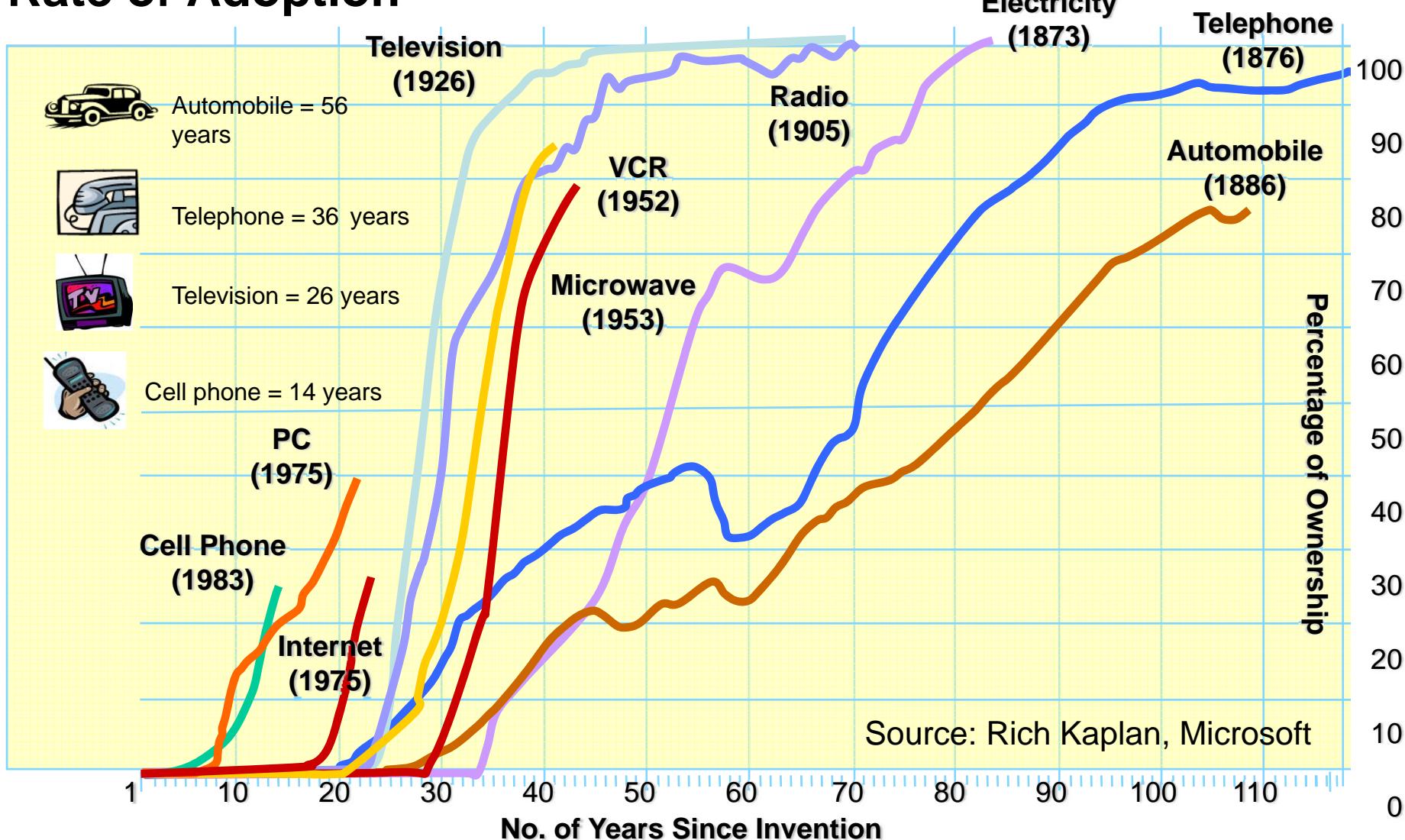
Carnegie Mellon

Developing Complex Systems – Using CMMI to Achieve Effective Systems and Software Engineering Integration
Dr. Kenneth E. Nidiffer
© 2008 Carnegie Mellon University

Facilitating Integration: Moore's Law - The Number of Transistors That Can be Placed on an Integrated Circuit is Doubling Approximately Every Two Years



Facilitating Integration: Increased Technological Rate of Adoption



Management Integration: Life of a Program Manager in a System of Systems Operation...



Relationship Between Integration Complexity and Acquisition Success Improving and More Improvements are on the Way But



Software is Growing in Complexity

- 80% of some weapon system functionality is dependent upon software
- Consequences of software failure can be catastrophic

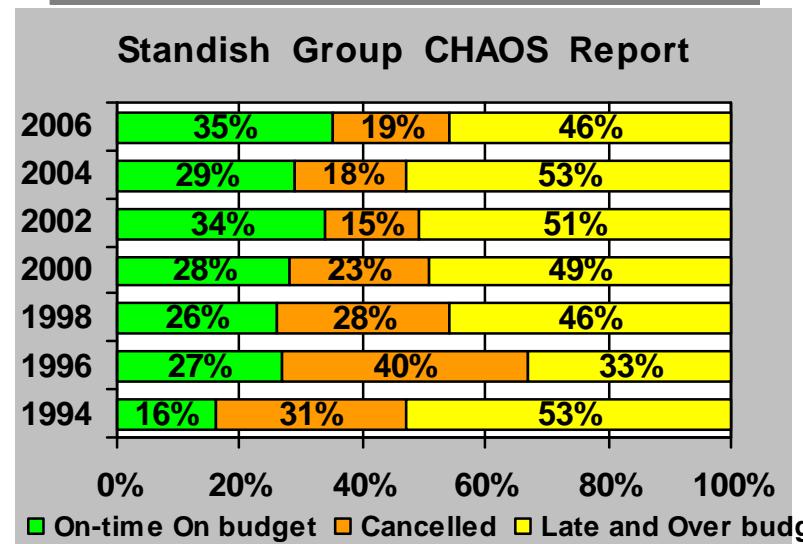
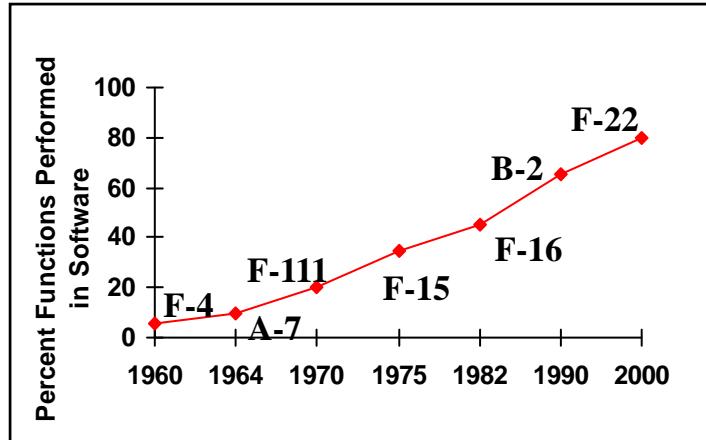
Software Acquisition is Difficult

- 46% are over-budget (by an average of 47%) or late (by an average of 72%)
- “Successful projects” have 68% of specified features

Software is Pervasive

- IT Systems, C4ISR, Weapons, etc

On-going Changes to the Acquisition Process Targeted at Correcting this Issue



Integration Challenges: Some Drivers That Increase the Risk of Engineering Software-Intensive Systems



Need Exists to Address Both Sides, and Do So with Compressed Delivery Schedules via Improvements in Systems/Software Engineering

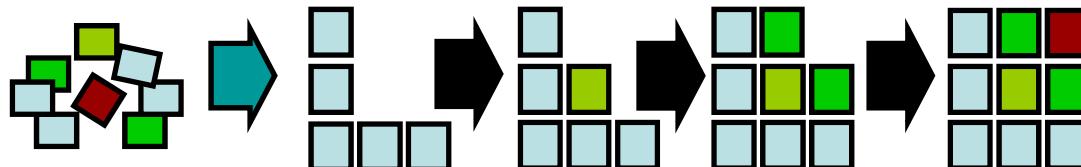


CMMI® Product Integration (PI)



Purpose

Assemble the product from the product components, ensure that the product, as integrated, functions properly, and deliver the product.



Source: SEI CMMI® Training Material

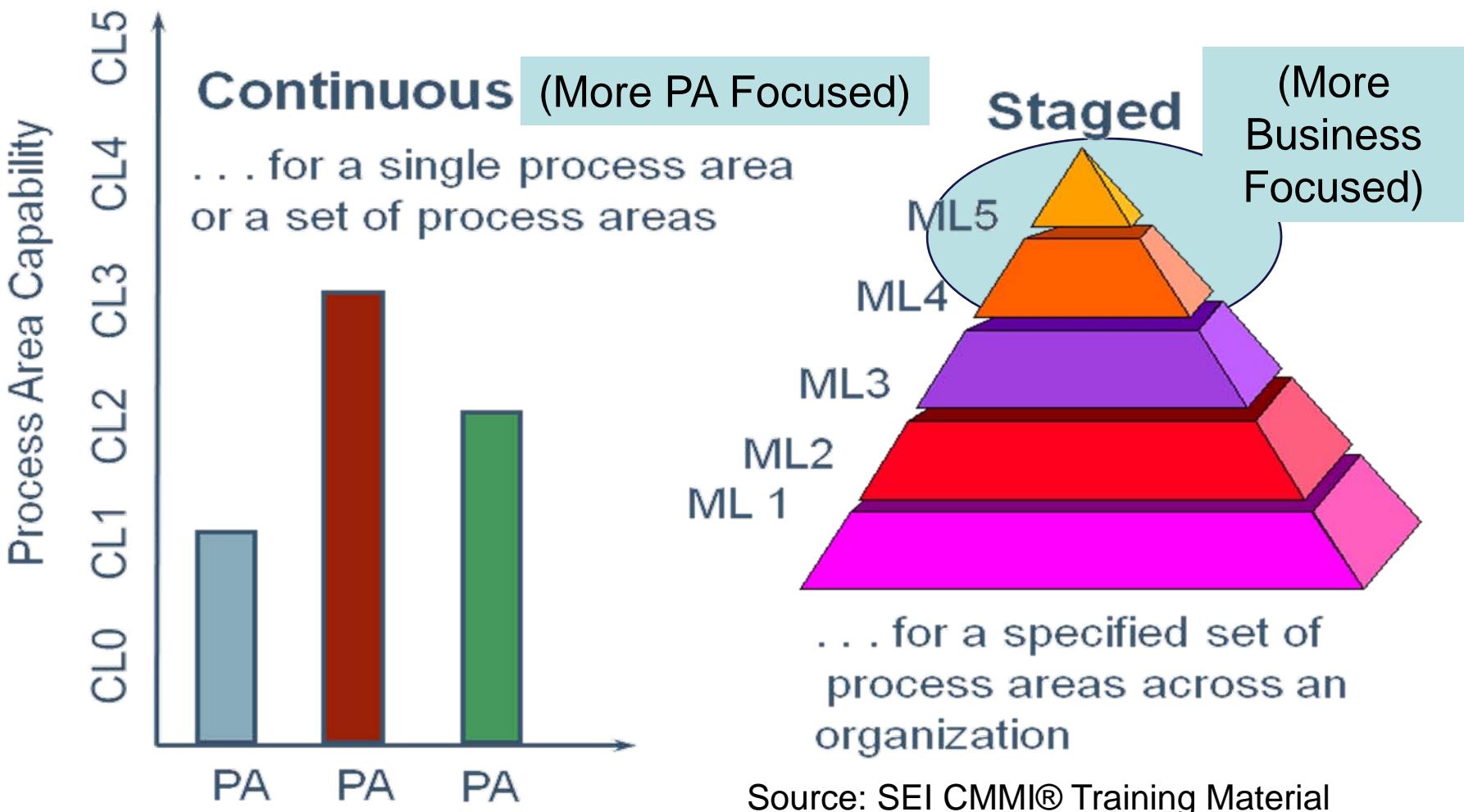


Software Engineering Institute

Carnegie Mellon

Developing Complex Systems – Using CMMI to Achieve
Effective Systems and Software Engineering Integration
Dr. Kenneth E. Nidiffer
© 2008 Carnegie Mellon University

Two Representations – Focus at Higher Maturity May Be Different Depending on Representation



Staged Representation: PAs by Maturity Level



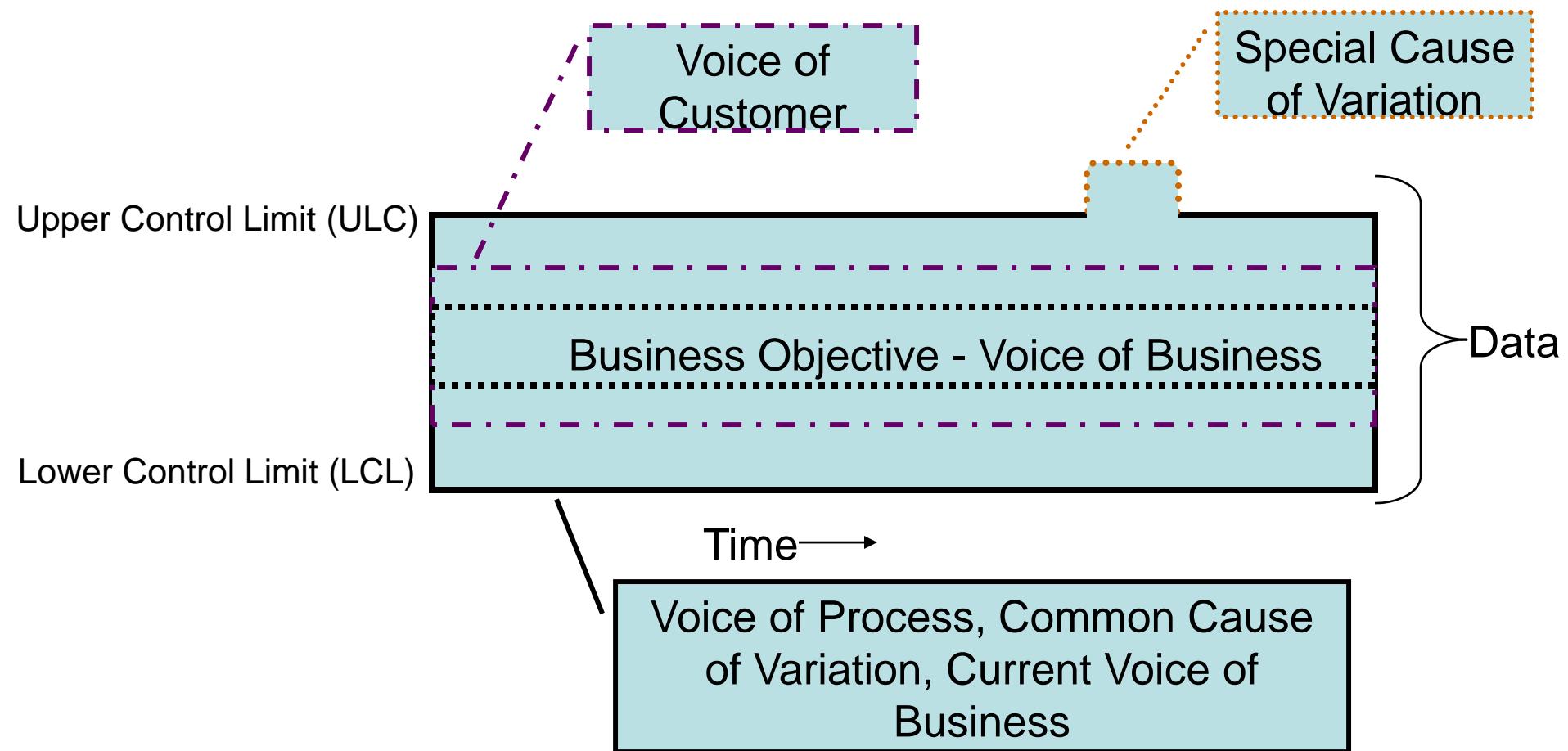
Level	Focus	Process Areas	
5 Optimizing	<i>Continuous Process Improvement</i>	Organizational Innovation and Deployment Causal Analysis and Resolution	
4 Quantitatively Managed	<i>Quantitative Management</i>	Organizational Process Performance Quantitative Project Management	
3 Defined	<i>Process Standardization</i>	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organizational Process Definition +IPPD Organizational Training Integrated Project Management +IPPD Risk Management Decision Analysis and Resolution	
2 Managed	<i>Basic Project Management</i>	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management Measurement and Analysis Process and Product Quality Assurance Configuration Management	
1 Initial			

Risk
Rework

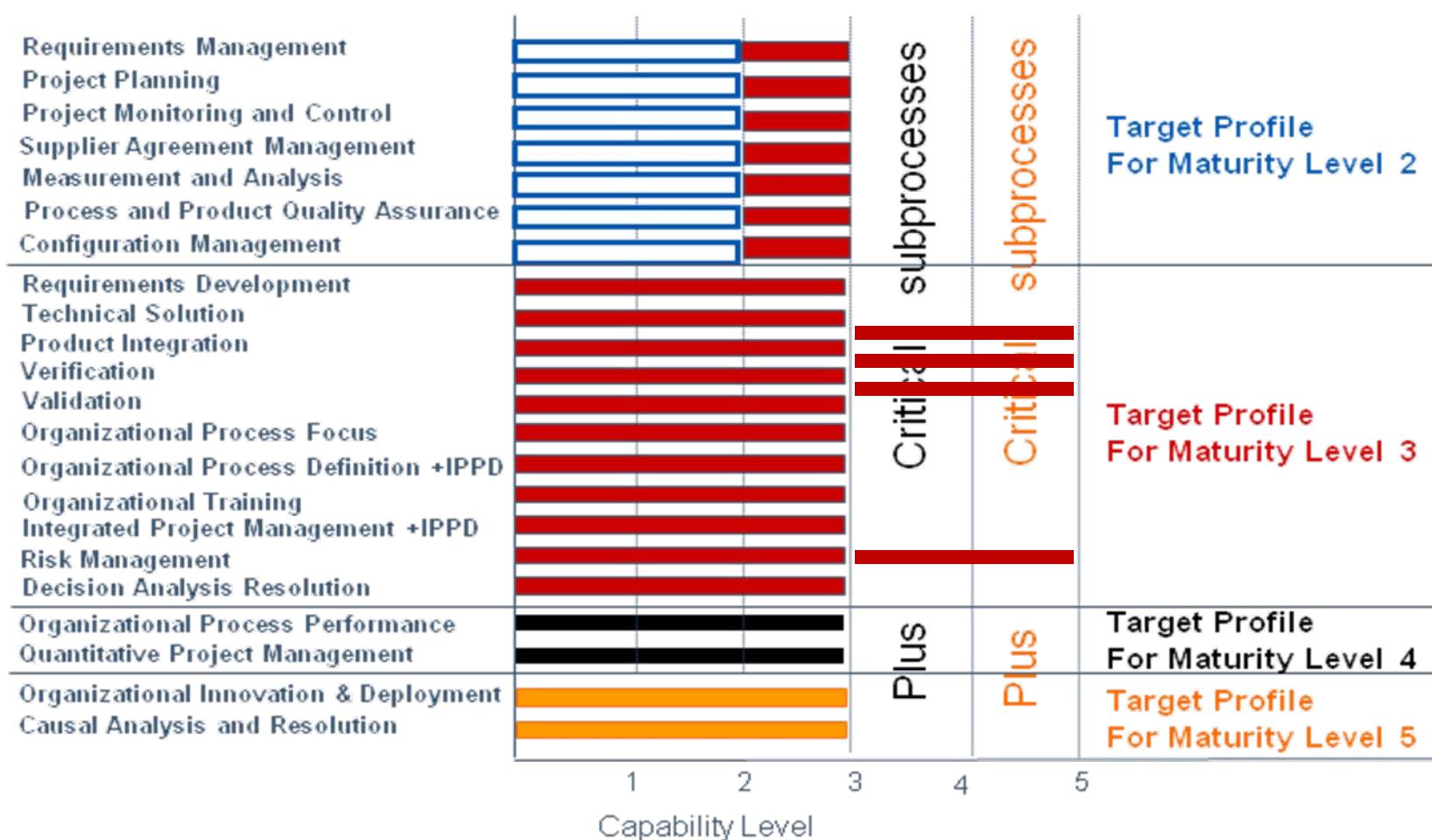
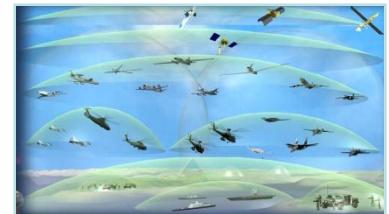
Source: SEI CMMI® Training Material



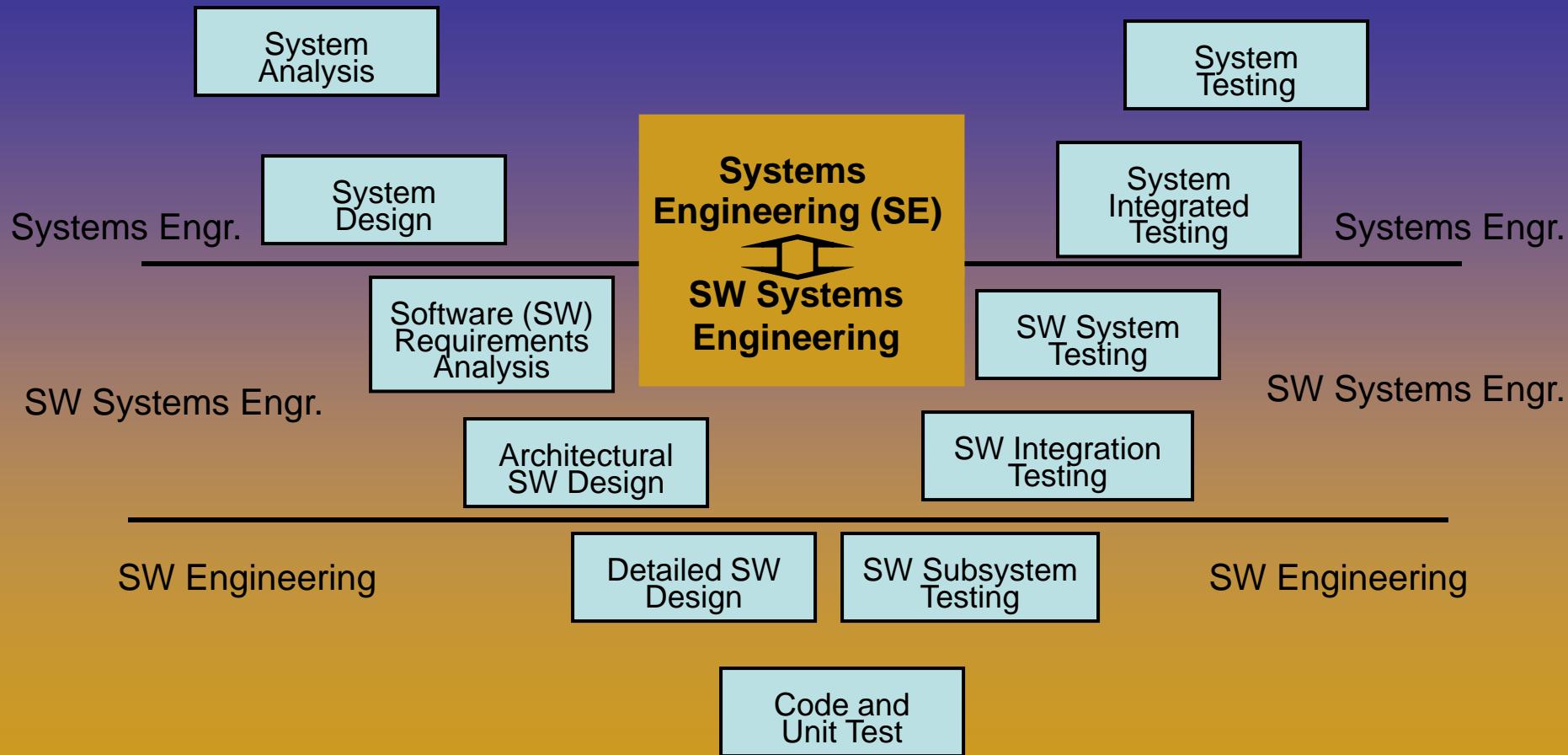
Run Chart - Definitions



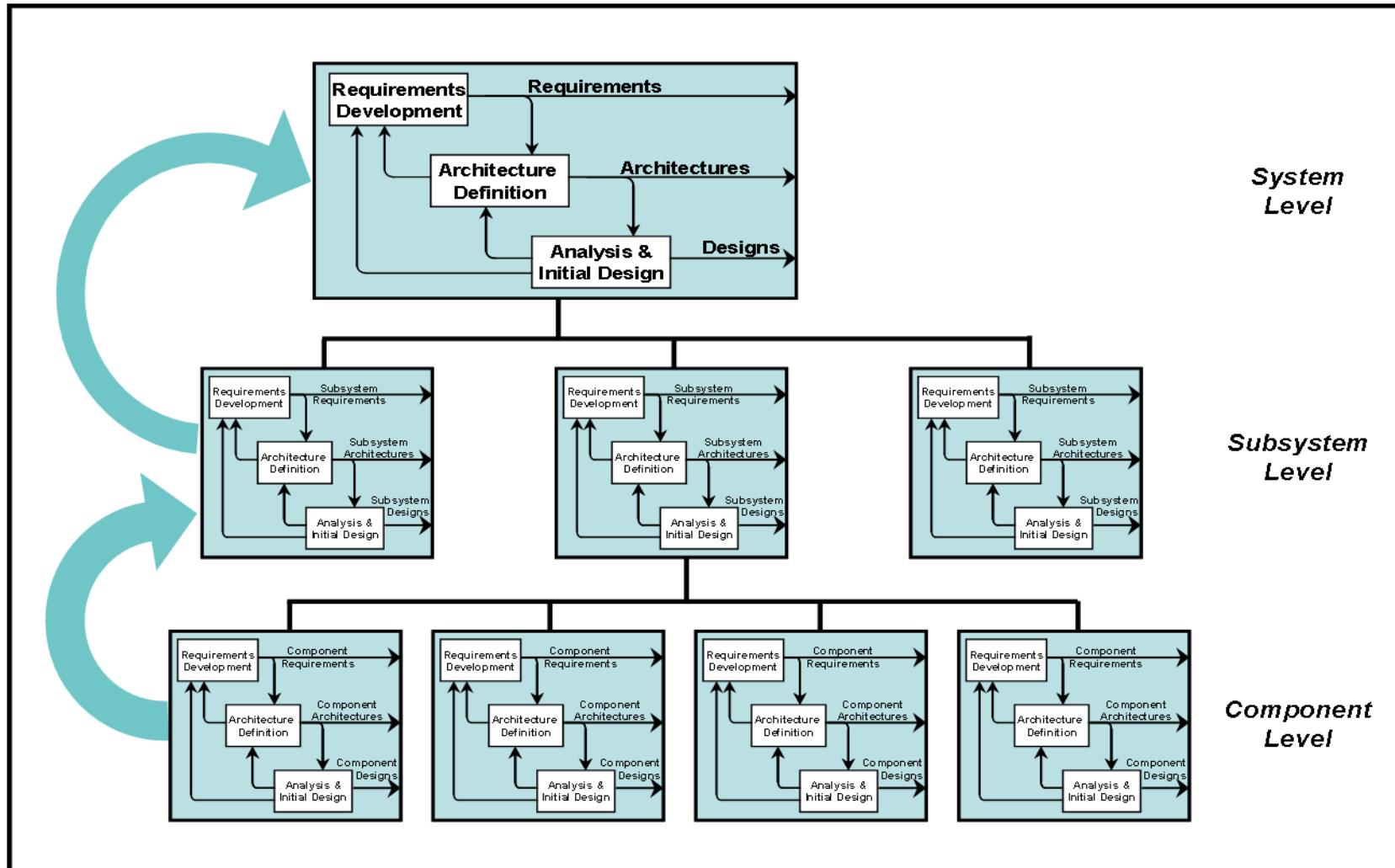
Focus on Business Objectives



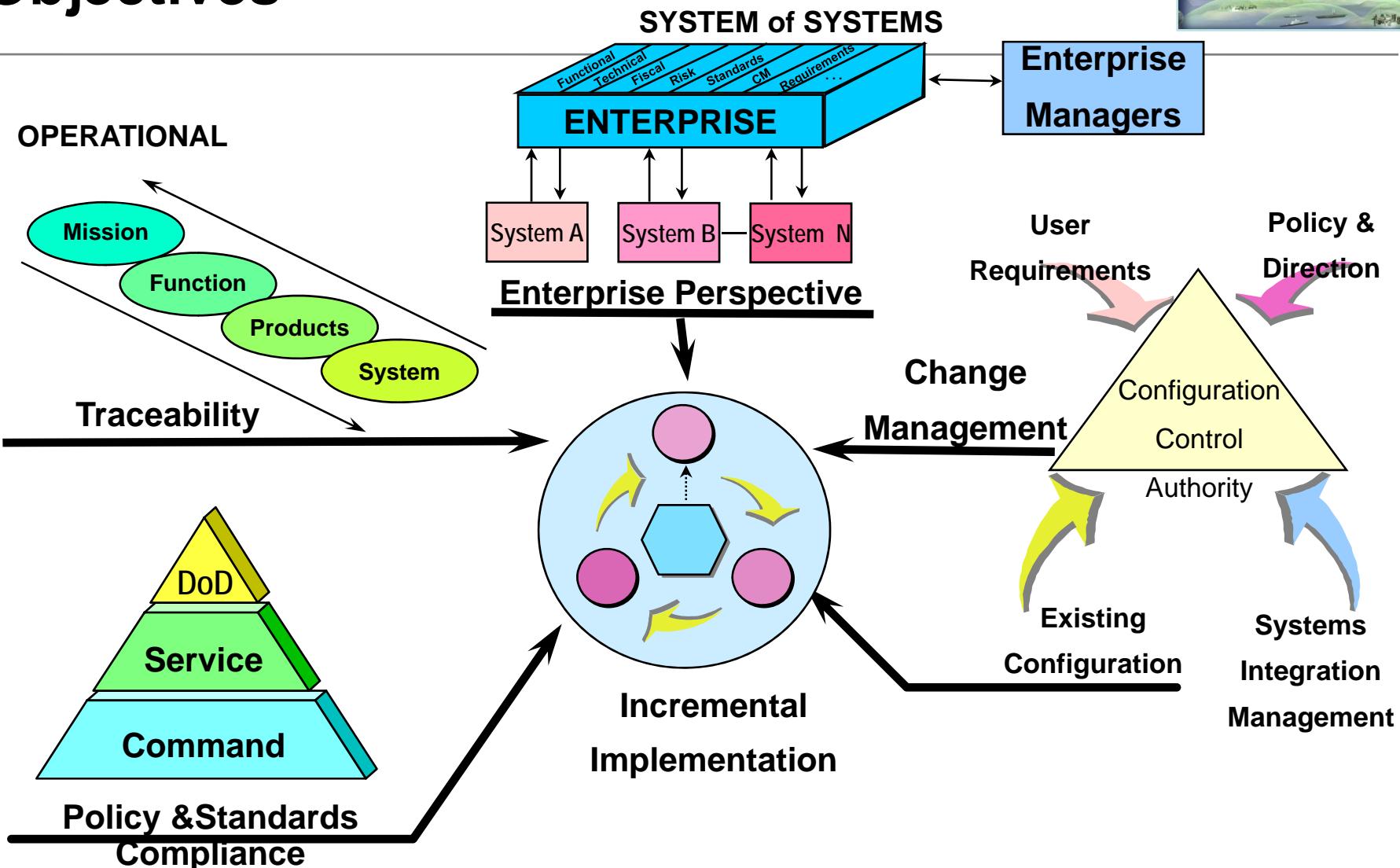
CMMI® Provides a Framework for Software and System Engineering to Become More Integrated



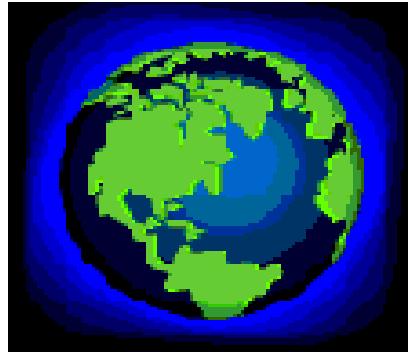
Prior to Product Integration – Left Side of Vee Chart



Integration Management By Business Objectives



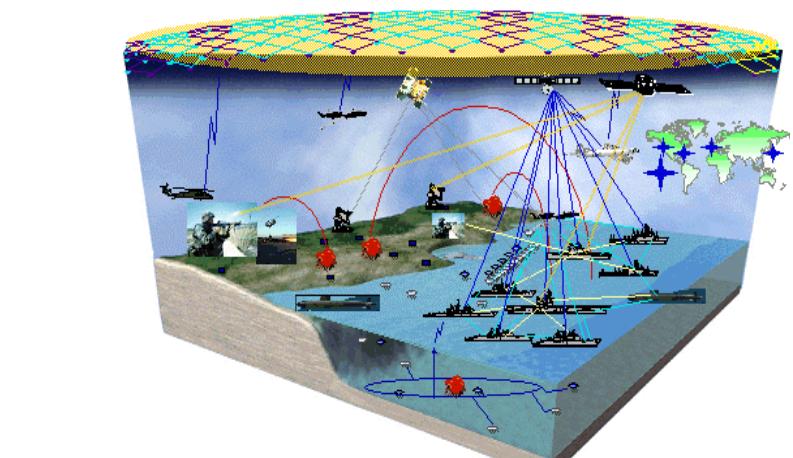
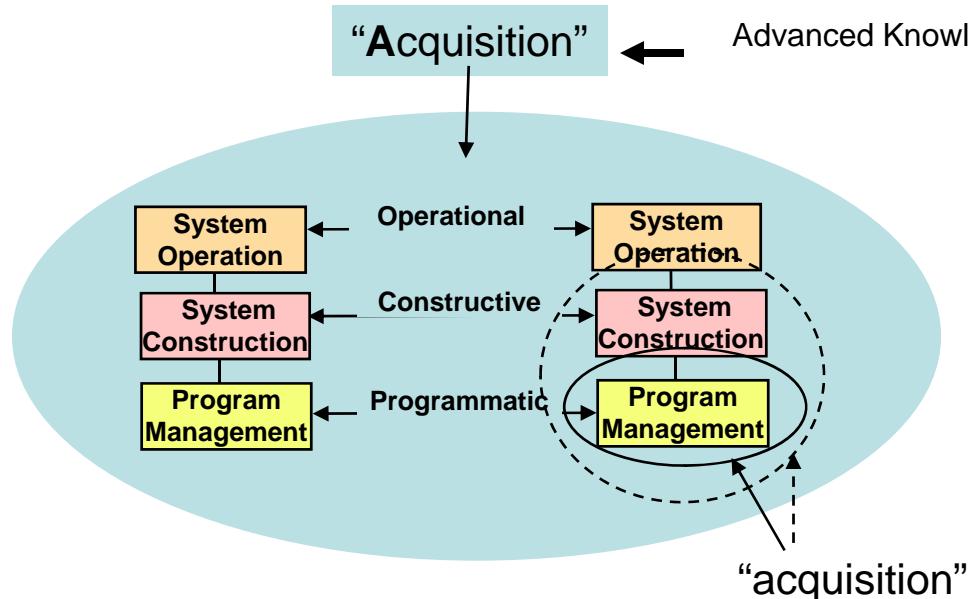
Engineering Integration – Achievement of Flexible Boundary-Crossing Acquisition Structure



2005 study confirmed*:

- In advanced knowledge-based organizations, management's desire for the flow of knowledge is greater than the desire to control boundaries
- Unlike the matrix organization, there is less impact on the dynamics of formal power and control
- **Important to measure the system in terms of user performance**

* Using Communities of Practice to Drive Organizational Performance and Innovation, 2005, APQ study



From "Science and Technology to Support FORCENet," Raytheon TD-06-008. Used by permission.

Ref: Jim Smith, (703) 908-8221, jds@sei.cmu.edu



Software Engineering Institute

Carnegie Mellon

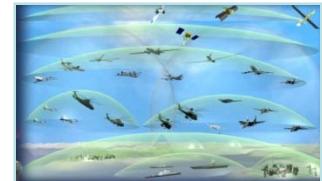
Systems and Software Engineering: Ten Trends



- *Greater integration demands on systems and software engineers will stimulate growth in the field – nationally and internationally*
- *Industry/Gov't will increasingly focus on attracting, training and retaining systems and software engineering talent – short and long run – with emphasis on providing a more integrated work environment (7 by 24, any shore)*
- *Increased reliance on systems and software engineering processes and technologies to effectively manage integration issues*
- *The laws of Augustine's and Moore will continue to hold and will continue to be a forcing function to facilitate the need for integration*



Systems and Software Engineering: Ten Trends



- *Improvements risk-reduction collaboration mechanisms will be significant enablers for increases in systems and software engineering communication and “decision velocity”*
- *Systems and software engineers will continually find way to innovative to reduce integration issues*
- *Increased importance of modeling and simulation*
- *Increased business focus for system and software engineering integration*
- *Shift of systems and software engineering focus from the platform to integrated networks and ground systems*
- *Use of CMMI-Dev will continue to be important!*



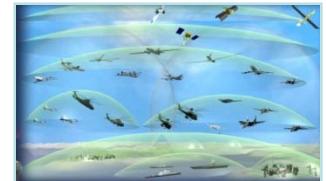


Questions?

THE
KNOWN WORLD
*Beyond here there
be Dragons*



Recommended Readings



Buckman, Robert H. *Building a Knowledge-Driven Organization*. McGraw-Hill, New York, NY, 2004.

Chrissis, M. , Konrad, M., and Shrum, S, CMMI® for Development, Version 1.2, Guidelines for Process Integration and Product Improvement, Fifth Printing, 2007, Addison Wesley

GAO Report: 08-467SP, Defense Acquisitions – Assessment of Selected Weapon Systems, March 2008

Chesbrough, Henry William. *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Harvard Business School Publishing Corporation, Boston, MA 2003.

Drucker, Peter. *Managing in the Next Society*. Truman Talley Books, New York, NY, 2003.

Friedman, Thomas L. “*The World Is Flat*”, Farrar, Straus and Giroux, 2005

Gates, William H. III “*Business @ The Speed of Thought – Using a Digital Nervous System*”, Time Warner Books, 1999

Kurstedt, Harold and Pamela, *Systems and Software Engineering Interfaces, Dealing with the Bumpy Roads*,

Nidiffer, Kenneth E. and Doland, Diana “*Evolving Distributed Project Management*”, special issue IEEE Software, Sept/Oct 2005

Northrop, Linda. *Ultra-Large-Scale Systems – The Software Challenge of the Future*, Software Engineering Institute, June 2006

Rouse, William B. et al, *Understanding R&D Value Creation with Organizational Simulation*, Tennenbaum Institute, H. Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, Oct 2006

Wladawsky-Berger, Irving. “*The Future of IT in an On-Demand World.*” IBM Server Group, Keynote address at OSBC 2005. Archived at <http://www.itconversations.com/shows/detail495.html>

0008 GOES-7 IR 08 30 OCT 91303 120100 01297 09305 04.00



Software Engineering Institute

Carnegie Mellon