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

Dr. Kenneth E. Nidiffer
Director of Strategic Plans for
Government Programs
nidiffer@sei.cmu.edu
703.908.1117
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 Trends That Impact Systems Engineering

<table>
<thead>
<tr>
<th>Traditional</th>
<th>Future</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Standalone systems</td>
<td>• Everything connected-maybe</td>
</tr>
<tr>
<td>• Mostly source code</td>
<td>• Mostly COTS components</td>
</tr>
<tr>
<td>• Requirements-driven</td>
<td>• Requirements are emergent</td>
</tr>
<tr>
<td>• Control over evolution</td>
<td>• No control over COTS evolution</td>
</tr>
<tr>
<td>• Focus on software</td>
<td>• Focus on systems and software</td>
</tr>
<tr>
<td>• Stable requirements</td>
<td>• Rapid change</td>
</tr>
<tr>
<td>• Premium on cost</td>
<td>• Premium on value, speed, quality</td>
</tr>
<tr>
<td>• Staffing workable</td>
<td>• Scarcity of critical talent</td>
</tr>
</tbody>
</table>

**Emerging Dynamics of Bringing Systems and Software Engineering in Continued Partnership**
The Acceleration of Innovation in the 21st Century:
- Facilitating Our Ability to Integrate

The Amount of New Technological Innovation is Doubling Every Two Years
- Requires More Upfront SE/SW Engineering to Leverage Trends
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

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
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

Source: Rich Kaplan, Microsoft

Automobile = 56 years
Telephone = 36 years
Television = 26 years
Cell phone = 14 years
PC = 38 years
Internet = 38 years
VCR = 47 years
Microwave = 47 years
Radio = 72 years
Electricity = 139 years
Telephone = 139 years

Percentage of Ownership
No. of Years Since Invention

Source: Rich Kaplan, Microsoft
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

Platform: Customer Emphasis

Requirements: Acquisition Model

Dominant Prime: Program Execution

“Boxes”: Integration Challenge

Proprietary: Architectures and Standards

Enterprise: Objectives/Capabilities

Strategic Teaming: “Layers & Stacks”

Plug & Play

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

Continuous (More PA Focused)

... for a single process area or a set of process areas

Staged (More Business Focused)

... for a specified set of process areas across an organization

Source: SEI CMMI® Training Material
### Staged Representation: PAs by Maturity Level

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

Source: SEI CMMI® Training Material
Run Chart - Definitions

- Upper Control Limit (ULC)
- Lower Control Limit (LCL)
- Business Objective - Voice of Business
- Voice of Process, Common Cause of Variation, Current Voice of Business
- Voice of Customer
- Special Cause of Variation
- Data
- Time→

Definitions:
- Upper Control Limit (ULC)
- Lower Control Limit (LCL)
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

System Level

Subsystem Level

Component Level

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

SYSTEM of SYSTEMS

Enterprise

Enterprise Managers

User Requirements

Policy & Direction

Configuration Control Authority

Existing Configuration

Systems Integration Management

Incremental Implementation

DoD

Service

Command

Policy & Standards Compliance

Mission

Function

Products

System

Traceability

Enterprise Perspective

SYSTEM of SYSTEMS

Policy & Standards

Compliance

Mission

Function

Products

System

Traceability

DoD

Service

Command

Enterprise Perspective
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

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

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

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


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


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


