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
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
Need for Mission Integration

Less a Matter of Hitting a Window

And More a Matter of The Right Window - Right Now

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

<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

<table>
<thead>
<tr>
<th>Decade</th>
<th>Plane</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1960’s</td>
<td>F-4A</td>
<td>1000</td>
</tr>
<tr>
<td>1970’s</td>
<td>F-15A</td>
<td>50K</td>
</tr>
<tr>
<td>1980’s</td>
<td>F-16C</td>
<td>300K</td>
</tr>
<tr>
<td>1990’s</td>
<td>F-22</td>
<td>1.7M</td>
</tr>
<tr>
<td>2000+</td>
<td>F-35</td>
<td>&gt;6M</td>
</tr>
</tbody>
</table>
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

Integrated Circuit Complexity

Source: Intel
Facilitating Integration: Increased Technological Rate of Adoption

<table>
<thead>
<tr>
<th>Invention</th>
<th>Year of Invention</th>
<th>Percentage of Ownership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electricity</td>
<td>1873</td>
<td>100</td>
</tr>
<tr>
<td>Telephone</td>
<td>1876</td>
<td>100</td>
</tr>
<tr>
<td>Automobile</td>
<td>1886</td>
<td>100</td>
</tr>
<tr>
<td>Radio</td>
<td>1905</td>
<td>90</td>
</tr>
<tr>
<td>VCR</td>
<td>1952</td>
<td>80</td>
</tr>
<tr>
<td>Microwave</td>
<td>1953</td>
<td>70</td>
</tr>
<tr>
<td>Television</td>
<td>1926</td>
<td>60</td>
</tr>
<tr>
<td>Cell Phone</td>
<td>1983</td>
<td>50</td>
</tr>
<tr>
<td>PC</td>
<td>1975</td>
<td>40</td>
</tr>
<tr>
<td>Internet</td>
<td>1975</td>
<td>30</td>
</tr>
</tbody>
</table>

Source: Rich Kaplan, Microsoft

Automobile = 56 years
Telephone = 36 years
Television = 26 years
Cell phone = 14 years

No. of Years Since Invention vs. Percentage of Ownership
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

<table>
<thead>
<tr>
<th>Platform</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Emphasis</td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td>Objectives/Capabilities</td>
</tr>
<tr>
<td>Acquisition Model</td>
<td></td>
</tr>
<tr>
<td>Dominant Prime</td>
<td>Strategic Teaming</td>
</tr>
<tr>
<td>Program Execution</td>
<td></td>
</tr>
<tr>
<td>&quot;Boxes&quot;</td>
<td>&quot;Layers &amp; Stacks&quot;</td>
</tr>
<tr>
<td>Integration Challenge</td>
<td></td>
</tr>
<tr>
<td>Proprietary</td>
<td>Plug &amp; Play</td>
</tr>
<tr>
<td>Architectures and Standards</td>
<td></td>
</tr>
</tbody>
</table>

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, Causal Analysis and Resolution</td>
</tr>
<tr>
<td>4 Quantitatively Managed</td>
<td>Quantitative Management</td>
<td>Organizational Process Performance, Quantitative Project Management</td>
</tr>
<tr>
<td>2 Managed</td>
<td>Basic Project Management</td>
<td>Requirements Management, Project Planning, Project Monitoring and Control, Supplier Agreement Management, Measurement and Analysis, Process and Product Quality Assurance, 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 (CLC)
- Business Objective - Voice of Business
- Voice of Customer
- Special Cause of Variation
- Voice of Process, Common Cause of Variation, Current Voice of Business
- Time

Data

Business Objective - Voice of Business

Voice of Customer

Special Cause of Variation

Voice of Process, Common Cause of Variation, Current Voice of Business

Time
Focus on Business Objectives

![Diagram showing maturity levels and subprocesses]

- **Requirements Management**
- **Project Planning**
- **Project Monitoring and Control**
- **Supplier Agreement Management**
- **Measurement and Analysis**
- **Process and Product Quality Assurance**
- **Configuration Management**

- **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**

- **Organizational Process Performance**
- **Quantitative Project Management**

- **Organizational Innovation & Deployment**
- **Causal Analysis and Resolution**

**Target Profile For Maturity Level 2**

**Target Profile For Maturity Level 3**

**Target Profile For Maturity Level 4**

**Target Profile For Maturity Level 5**
CMMI® Provides a Framework for Software and System Engineering to Become More Integrated

Systems Engineering (SE)

SW Systems Engineering

- System Analysis
- System Design
- Software (SW) Requirements Analysis
- Architectural SW Design
- Detailed SW Design
- Code and Unit Test
- SW Subsystem Testing
- SW Integration Testing
- SW System Testing
- System Integrated Testing
- System Testing
Prior to Product Integration – Left Side of Vee Chart

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

SG 1: Prepare for Product Integration
Preparation for product integration is conducted.

SG 2: Ensure Interface Compatibility
The product component interfaces, both internal and external, are compatible.

SG 3: Assemble Product Components and Deliver the Product
Verified product components are assembled and the integrated, verified, and validated product is delivered.

Source: SEI CMMI® Training Material
Product Integration Goals

Prepare for Product Integration

Ensure Interface Compatibility

Assemblies

Sub-assemblies

Assemble Product Components and Deliver the Product

Source: SEI CMMI® Training Material
Integration Management By Business Objectives

OPERATIONAL

- Mission
- Function
- Products
- System

Traceability

SYSTEM of SYSTEMS

Enterprise Perspective

- User Requirements
- Policy & Direction

Configuration Control Authority

Existing Configuration Systems Integration Management

Incemental Implementation

- System A
- System B
- System N

Policy & Standards Compliance

DoD Service Command

Software Engineering Institute | Carnegie Mellon

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
Northrop Grumman Unveils New Modeling and Simulation Research Center

New Aviation Ship Integration Center, a state-of-the-art research facility established in partnership with the U.S. Navy to conduct modeling, simulation, research, development and in-depth analysis for CVN 21-class aircraft carriers and other aviation-capable ships.

OSD (AT&L) Policy – Prototyping and Competition, 2007
Higher-Maturity Approaches to Process Improvement Are Important and Synergistic Trends

Data-Driven (e.g., Six Sigma, Lean)

- Determine what your processes can do (Voice of Process)
  - Statistical Process Control
- Clarify what your customer wants (Voice of Customer)
  - Critical to Quality (CTQs)
- Identify and prioritize improvement opportunities
  - Causal analysis of data
- Determine where your customers/competitors are going (Voice of Business)
  - Design for Six Sigma

Model-Driven (e.g., CMM, CMMI)

- Determine the industry best practice
  - Benchmarking, models
- Compare your current practices to the model
  - Appraisal, education
- Identify and prioritize improvement opportunities
  - Implementation
  - Institutionalization
- Look for ways to optimize the processes

CMMI and Six Sigma, Siviy, et al, 2007, Addison Wesley
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,


