

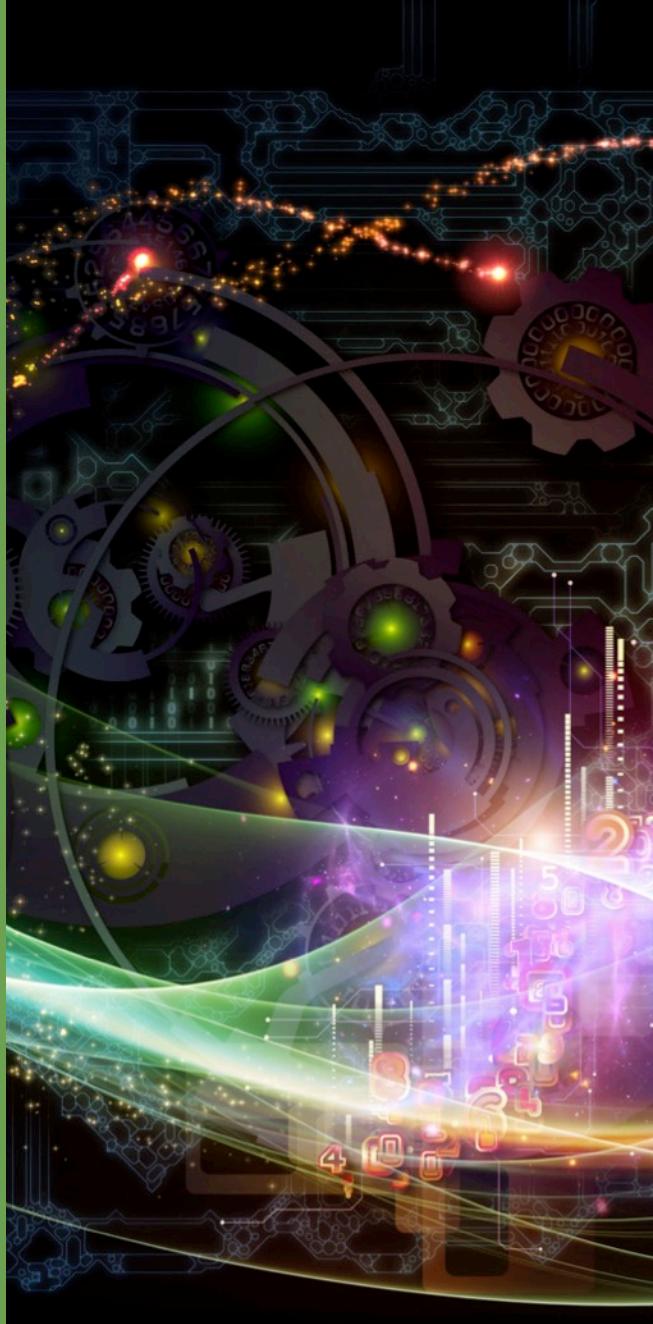
Software Solutions Symposium 2017

March 20–23, 2017

Software Sustainment: Continuous Engineering to Deliver Warfighter Capability

Michael H. McLendon (SEI)
John Stankowski (OSD)
Dr. Forrest Shull (SEI)
Stephany Bellomo (SEI)

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213



Software Engineering Institute

| Carnegie Mellon University

Software Sustainment: Continuous
Engineering to Deliver Warfighter Capability
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for
public release and unlimited distribution. Please see Copyright
notice for non-US Government use and distribution.

Disclaimer

Copyright 2017 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM-0004593

SW Sustainment Headlines!!!

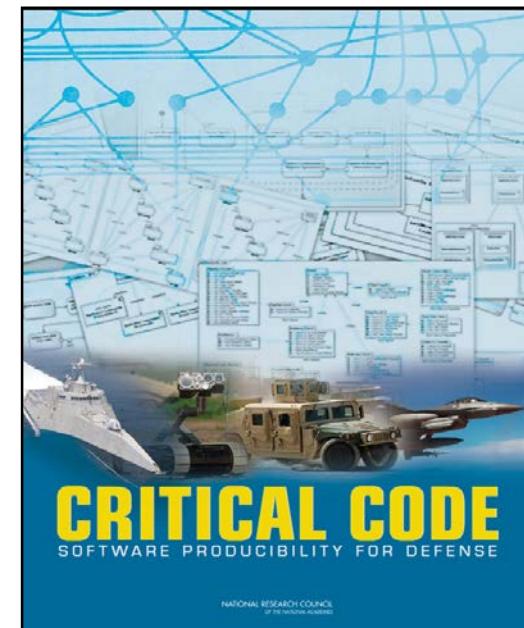
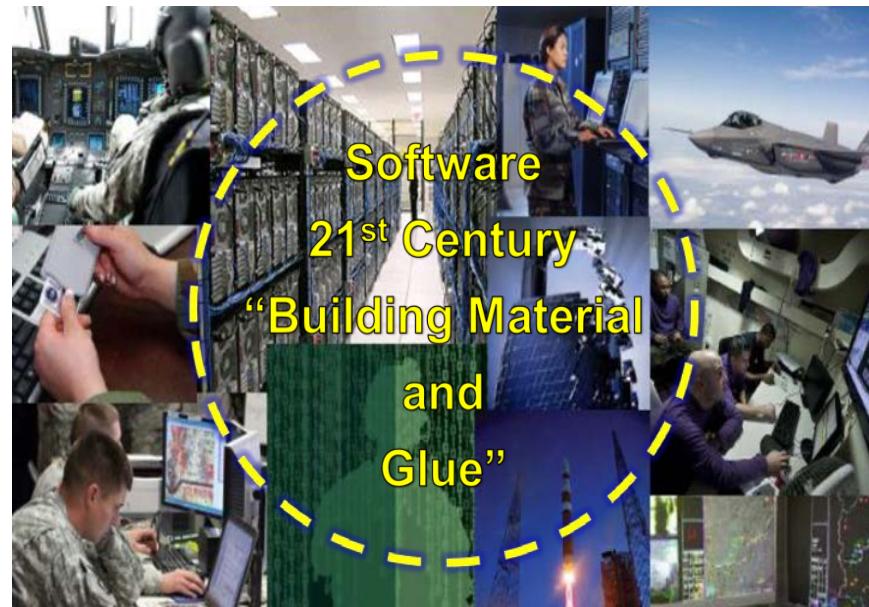


Goal - Sustained Military Advantage



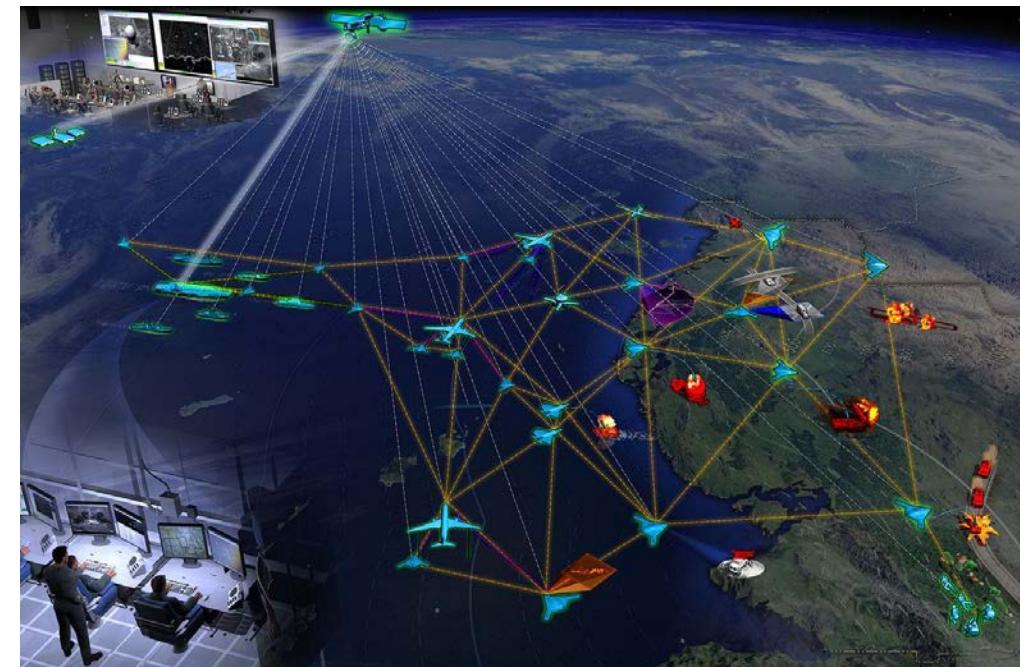
Software and DoD

- Warfighter capability embodied in **software**...a unique source of strategic, military advantage
- DoD's ability to produce and evolve **software** at core of achieving and, sustaining mission superiority, agility, more timely and better decisions
- **Software**...essential to vast range of military system capabilities, operations...deepening and broadening role [NRC Critical Code, 2010]
- Many aspects of **cyber security (defensive and offensive)** closely linked with software capability
- Creates imperatives for leading and managing software-intensive systems



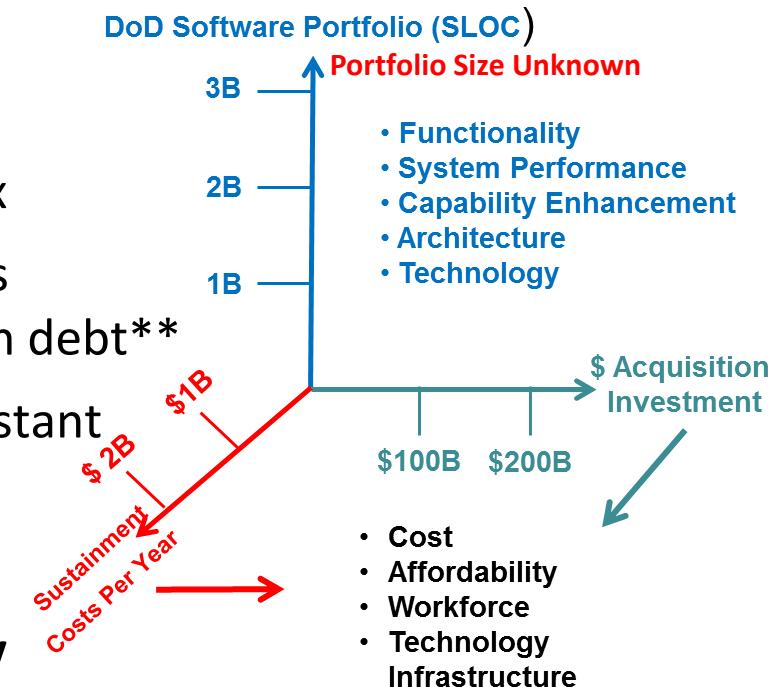
Software Challenges for Leadership

- Acquire & sustain software-driven systems operating interdependently with high assurance in a complex net-centric, cyber environment to continuously achieve mission success over the life cycle
 - Affordably with velocity
 - Spectrum of systems (Bus IT... C4ISR...Platforms...nodes in a larger scale SoS ecosystem
 - All systems are SW or controlled by SW
- Create technical & program mgmt infrastructure capabilities to acquire and continually engineer systems for warfighter competitiveness
- Plan & execute in a dynamic policy, governance, advancing technology, & \$\$\$ constrained environment for life cycle program success



Software Sustainment – Bottom Line

- Our paradigms about sustainment formed from decades of HW experience
- SW – now foundation building material in the engineering of systems; the defensive/offensive weaponry of cyber security – different nature than HW...
- **Not a physics of failure domain**...demand/\$ do not scale by optempo, force structure size...
- About continuous software engineering; not mx
- Increasing, unrecognized bow wave of demands (composition, complexity, characteristics,* tech debt**)
- Complex technical infrastructure*** needs constant refresh , \$; opportunities to achieve > value
- Nest in unaffordable O&S trends
- **Enterprise challenge; needs enterprise strategy**



* Software demographics (size, language, architecture, etc.)

** Technical debt - a metaphor referring to the eventual consequences of inadequate architecture and software development

*** Workforce, knowledge, practice, tools, systems integration lab/test environment

Contact Info

Michael H. McLendon

Associate Director

Software Solutions Division

mmclendon@sei.cmu.edu

703-247-1389 (Arlington)

U.S. Mail

Carnegie Mellon University
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Customer Relations

Email: info@sei.cmu.edu
Telephone: +1 412-268-5800

Web

www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm



Software Solutions Symposium 2017

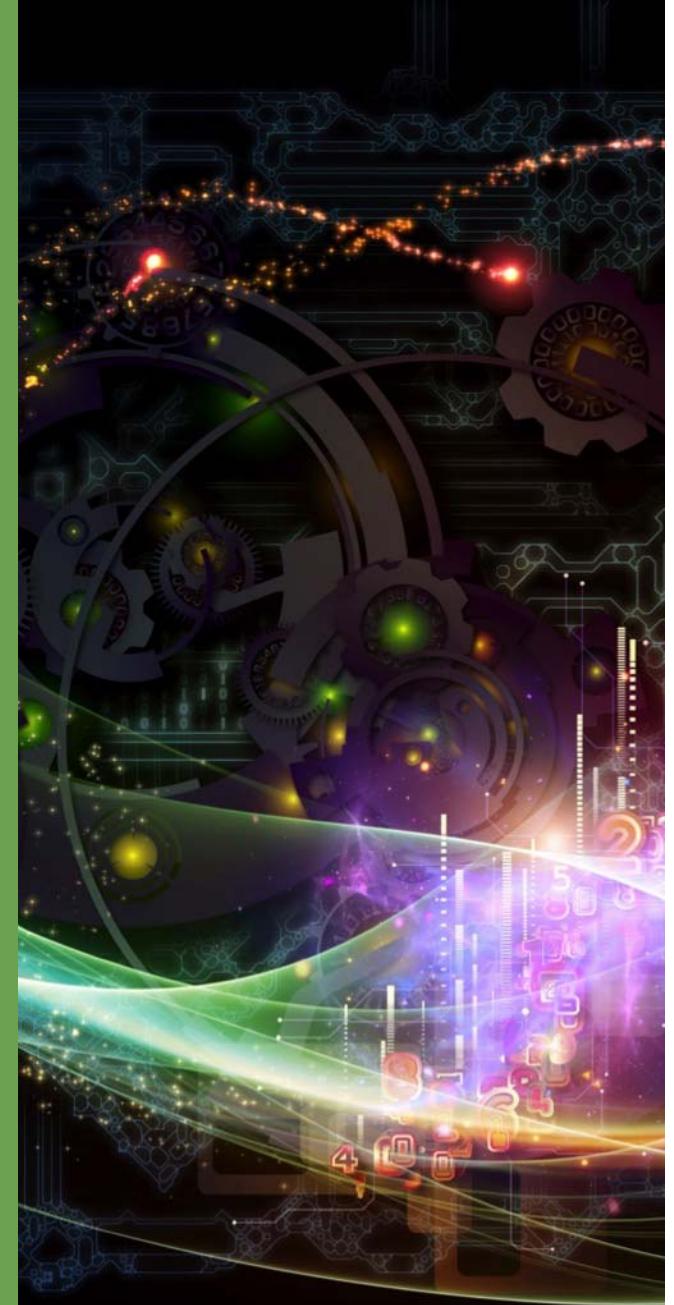
March 20–23, 2017

Software Maintenance / Sustainment: Supporting the Warfighter—A Matter of National Security

John Stankowski

Office of the Deputy Assistant Secretary of
Defense for Maintenance Policy and Programs

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213



Software Maintenance / Sustainment

Supporting the Warfighter— A Matter of National Security



March 2017

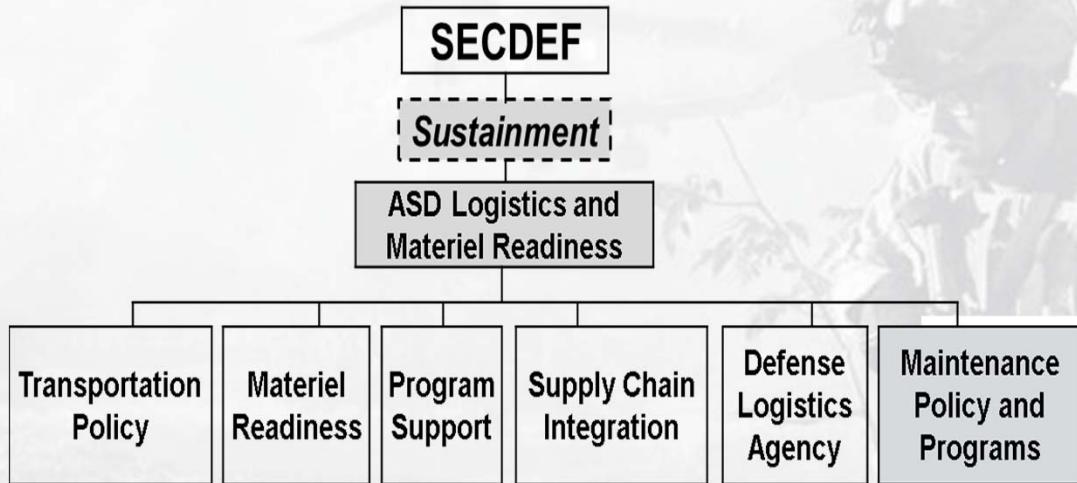
John Stankowski
Director for Maintenance Programs and Resources
Office of the Deputy Assistant Secretary of Defense
for Maintenance Policy and Programs

Agenda

- Organization and Focus
- Context
- Software Maintenance / Sustainment & Title 10
- Recent Activities
- Path Forward



Organization and Focus: Maintenance Policy and Programs



Key Responsibilities

- Life Cycle Sustainment
- Materiel Readiness
- Requirements and Resources
- Technology Integration
- Enterprise Optimization

Enterprise Lens:

- Operationally Ready Systems Today ...and Tomorrow
- Innovative, Diverse, and Agile Workforce
- Balanced Enterprise Capabilities & Capacity
- Evolve Capabilities for the Future Fight—Sustainment Technology Program

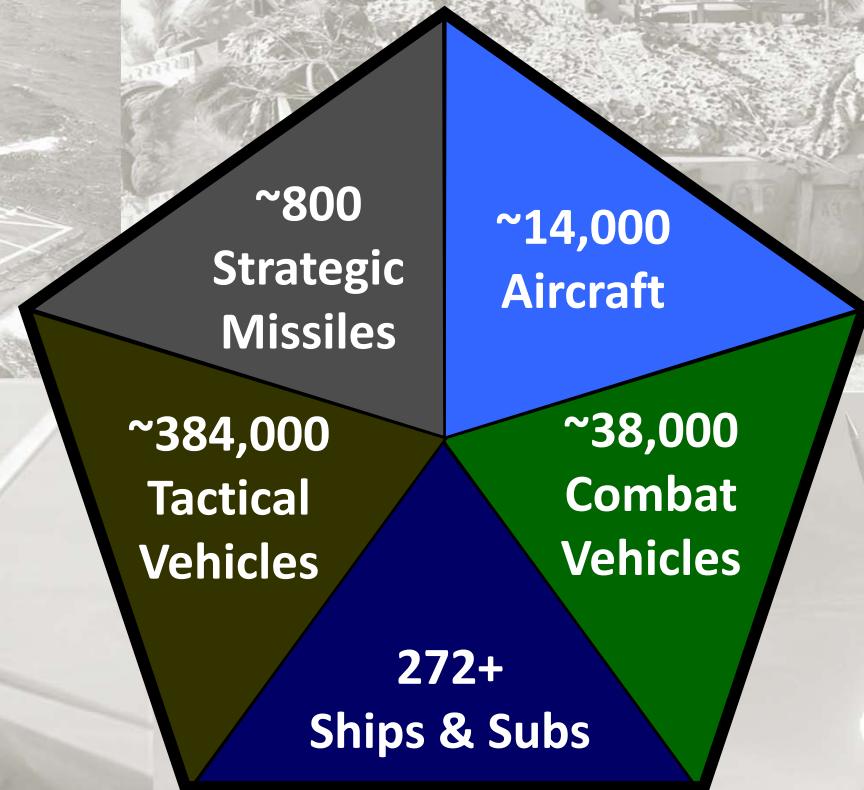


Context: DoD Software Maintenance / Sustainment

Equipment, Platform and Weapon System Inventory Valued Over \$350B

Annual Funding: ~\$5.6B (PB61) For Software Maintenance / Sustainment

Workforce: More than ~15,000 Government Personnel (DMDC) And Thousands In Private Sector Companies



Software is Big Business in DoD



Context: Major DoD Software Maintenance / Sustainment Activities by Location and Service





Context: Software in DoD

- ***Software integral to the operation and functionality of our equipment, platforms and weapon systems — approaching 100%***
- ***Software maintenance / sustainment is a critical capability directly supporting Joint Force readiness***
- ***Software maintenance / sustainment is a lot different from hardware maintenance, however...***

It's about more than money, it's about National Security 7



Software Maintenance / Sustainment and Title 10

- **Title 10 United States Code § 2460, Definition of depot-level maintenance and repair**

The term includes “*all aspects of software maintenance classified by the Department of Defense as of July 1, 1995, as depot-level maintenance and repair*”

- **Title 10 United States Code § 2464, Core logistics capabilities**

“It is essential for the national defense that the Department of Defense maintain a core logistics capability that is Government-owned and Government-operated (including Government personnel and Government-owned and Government-operated equipment and facilities) to ensure a ready and controlled source of technical competence and resources necessary to ensure effective and timely response to a mobilization, national defense contingency situations, and other emergency requirements.”

Enterprise Lens:

- Operationally Ready Systems Today ...and Tomorrow
- Innovative, Diverse, and Agile Workforce
- Balanced Enterprise Capabilities & Capacity



Recent Activities

- **Software Maintenance / Sustainment Panels at DoD Maintenance Symposium since 2012**
- **UAS Software Maintenance / Sustainment Working Group**
- **On-going MPP-sponsored study of the DoD software maintenance and sustainment enterprise being conducted by SEI**
- **DAU Software Sustainment Curriculum Working Group**

Path Forward

- **Maintain and strengthen software focus**
 - Strategic Communication and Stakeholder Buy-In
 - Recruitment and Retention
- **Address policy issues proactively**
 - Definition
 - Core Calculation
- **Identify opportunities for enterprise optimization**
 - Licenses and Technical Data Packages
 - Training Resources

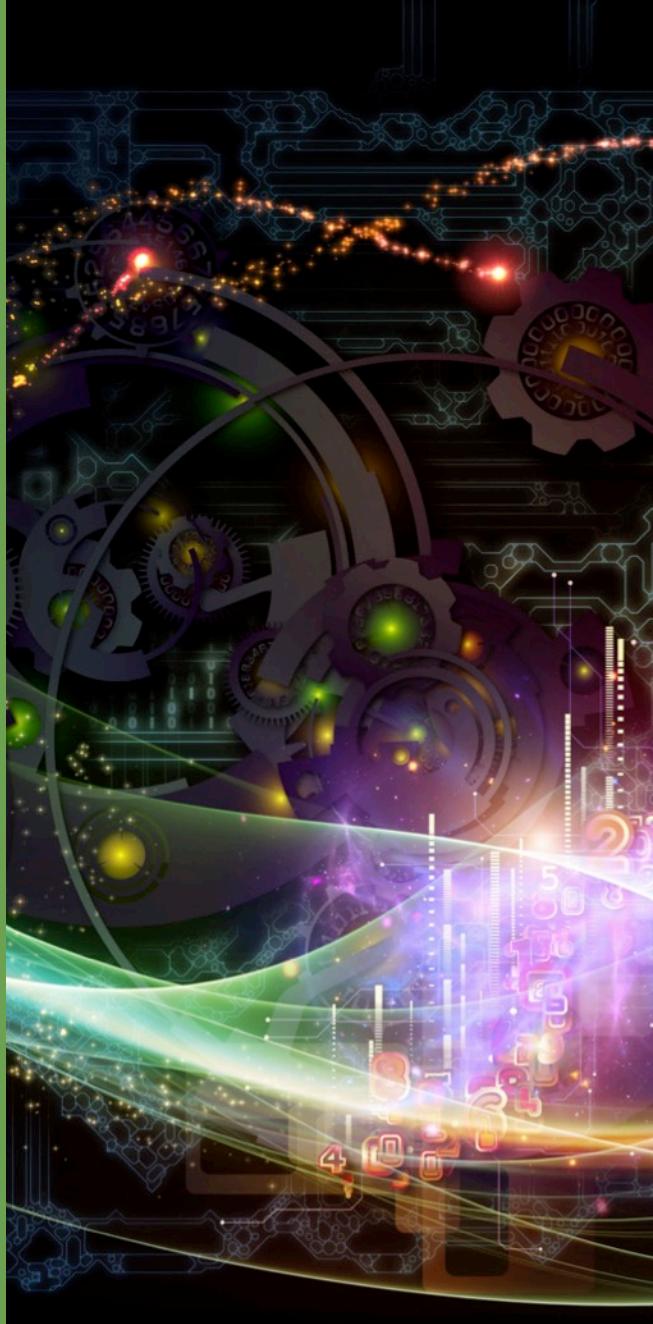
Software Solutions Symposium 2017

March 20–23, 2017

Sustainment and Technical Debt Discussion Topics

Stephany Bellomo

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213



Software Engineering Institute

| Carnegie Mellon University

Case Study of Government Modernization Project
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been
approved for public release and unlimited distribution.

Document Markings

Copyright 2017 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

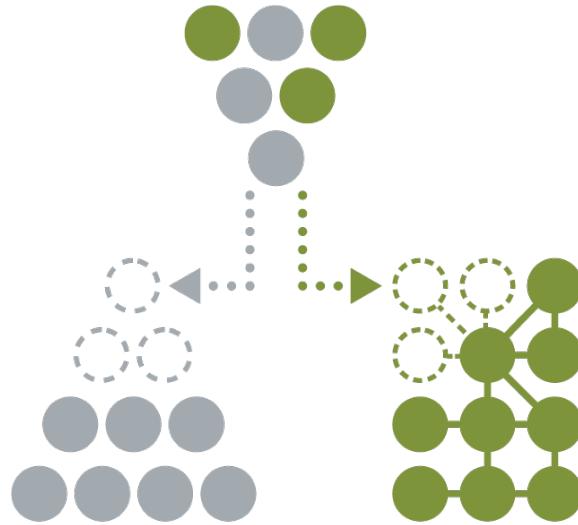
[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM-0004596



High-level Definition of Technical Debt



In software-intensive systems, technical debt is the collection of **design or implementation constructs** that are **expedient in the short term**, but set up a technical context that can make future changes more costly or impossible.

“Managing Technical Debt in Software Engineering,” Dagstuhl Reports, Vol. 6, Issue 4. <http://www.dagstuhl.de/16162>.

Symptoms and Challenges

Common Symptoms

- Slowing velocity for business capability
- Inability to make changes in the codebase (paralysis)
- Increasing cost and effort to maintain systems

State of the practice

- Kitchen sink syndrome – All my problems are technical debt
- Tactical analysis – Fix easy stuff, arbitrary use of complexity metrics, one-off defect reduction sprints, etc.

A more **Strategic Approach** is needed to
effectively manage Technical Debt



Refined Definition of Technical Debt

- A necessary step in forwarding the state of the practice is clarity around what is, and is not, technical debt
 - Analysis of examples yields refined definition below...

Technical debt examples*

- “There were two highly coupled modules that should have been designed separately from the beginning”
- “A simple API call turned into a nightmare...”



Refined Definition - Technical debt is a software design issue that:

- Exists in an **executable system artifact** (e.g., code)
- Is traced to **several locations** in the system
- Has a **quantifiable** effect on attributes of interest to developers

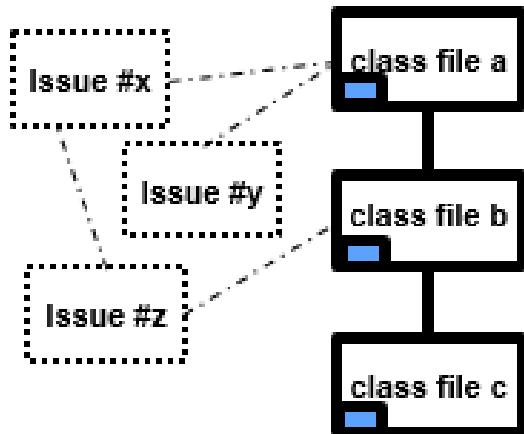
**From surveys, interviews, and issue tracker analysis*

Technical Debt Detection and Measurement

- There is no simple measure for technical debt, however, several artifacts produce useful evidence
- **Vision:** Overlay architectural information from multiple sources to detect and measure high risk areas

Multi-Artifact Analysis

Example



- **Source 1: Issue Tracker** - All these files have integer overflows that cause crashes
- **Source 2: Static Analysis** - One of the files participates in an architecture violation (cross-module cycles, improper hierarchy)
- **Source 3: Bug churn** - Developers create a patch every place they see the similar integer overflow issue.

Practical Suggestions for Sustainment

- Manage technical debt strategically by:
 - Involving sustainment groups early and continuously
 - Asking hard questions regularly
 - What are areas of pain? How to measure? When to pay down?
 - Collaborating on funding strategy for ongoing technical debt management (e.g., 10% of budget)
- Encourage development practices such as:
 - Create ***technical debt issue*** records and revisit regularly (manage as you would defects)
 - Focus static analysis on design-related violations
 - Monitor defects and change tracking logs for risk areas

Software Solutions Symposium 2017

March 20–23, 2017

The DoD Software Sustainment Ecosystem

Forrest Shull

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213



Software Engineering Institute

| Carnegie Mellon University

The DoD Software Sustainment Ecosystem
© 2017 Carnegie Mellon University

[Distribution Statement A. This material has been approved for
public release and unlimited distribution.]

Copyright 2017 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM-0004594



Software Sustainment - Definition

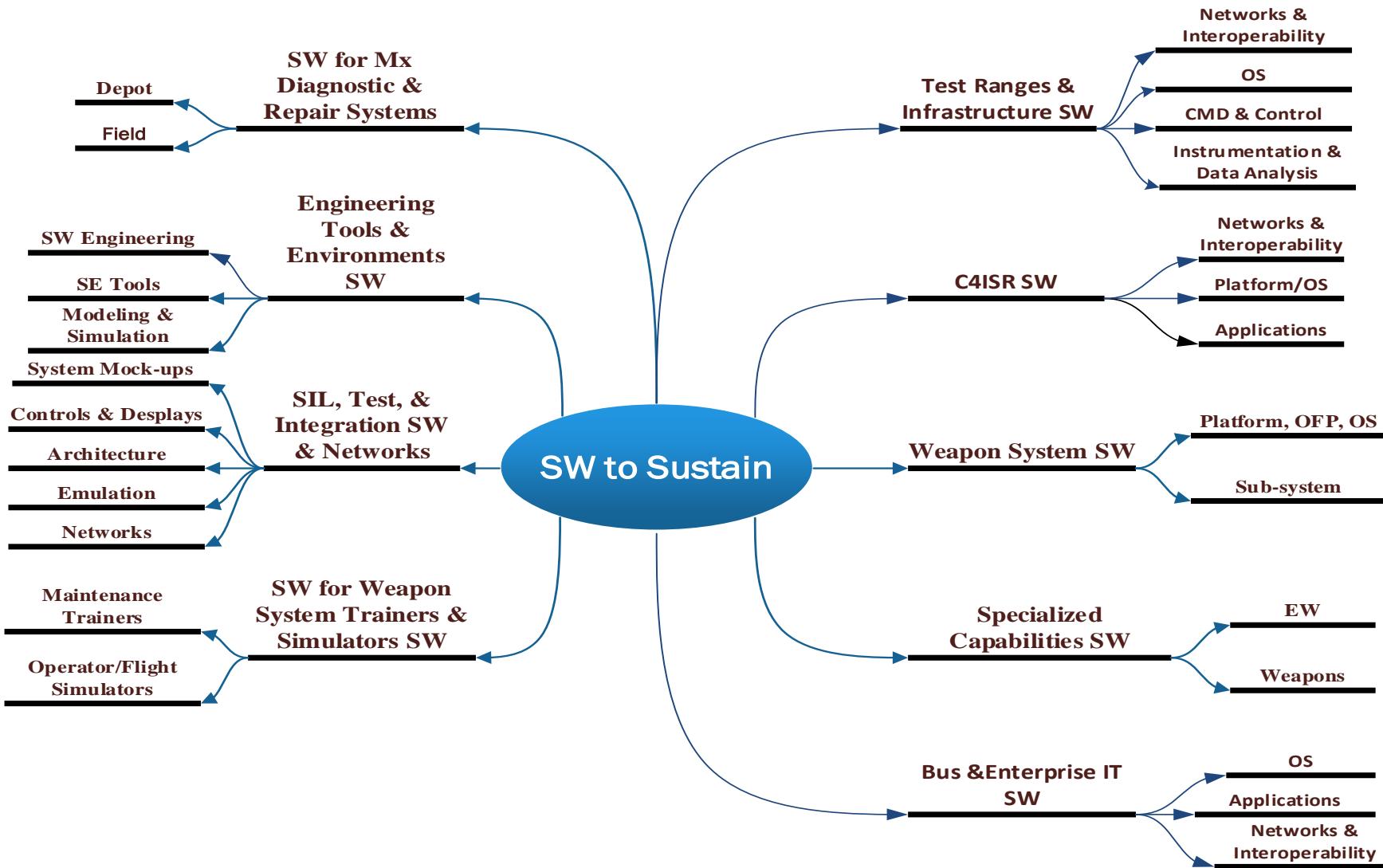
Software sustainment:

Orchestrating the processes, practices, technical resources, information, and workforce competencies for systems and software engineering,

to enable systems to continue mission operations and also to be enhanced to meet evolving threat and capability needs.

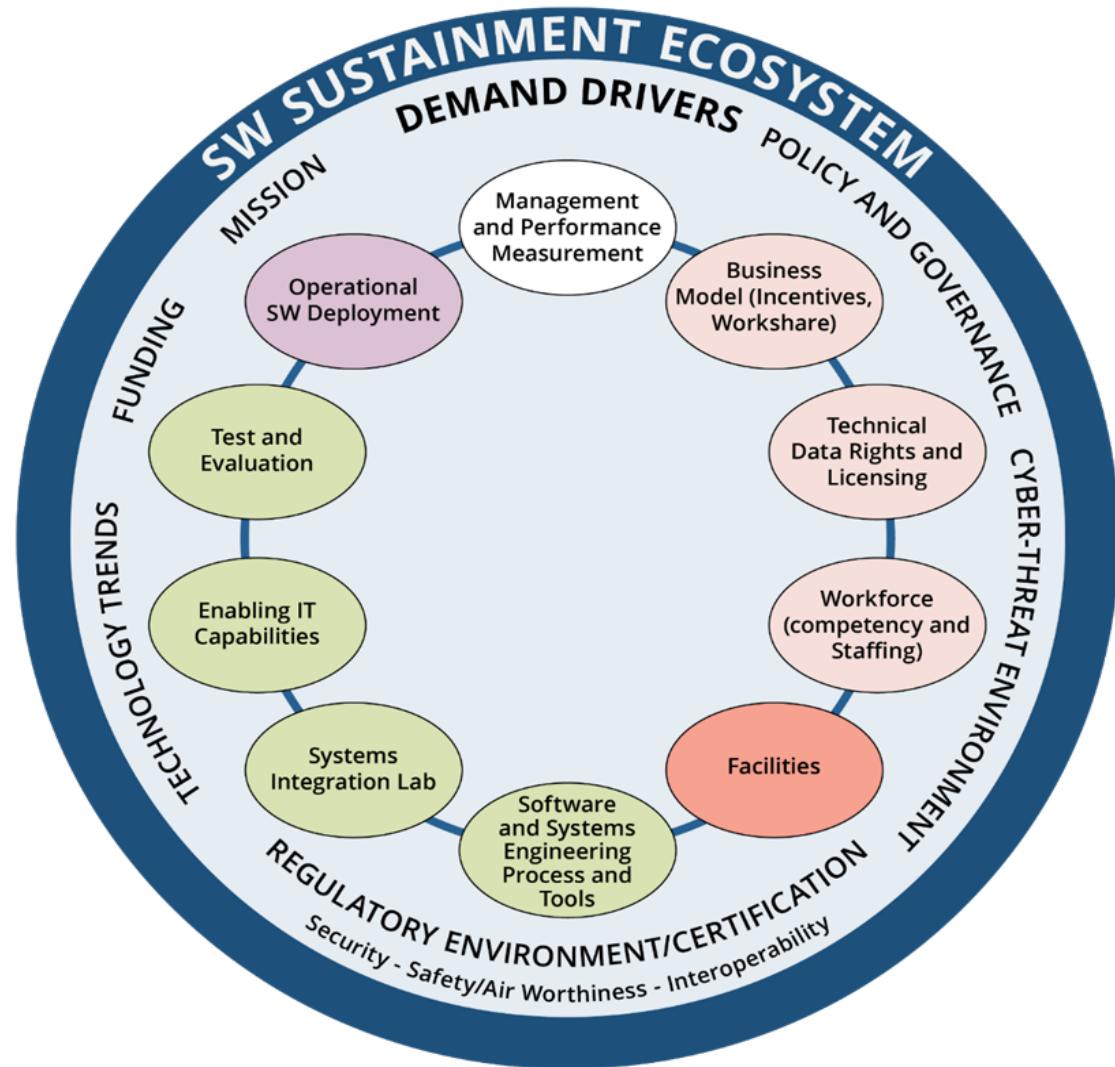


DoD's SW to Sustain is Far-Reaching... But There Is Limited Enterprise Visibility



Ecosystem Overview

- Four **infrastructure** elements: Basic, fundamental resources necessary for the sustainment activities
- Three **knowledge and expertise** elements: Skill sets, the government organic workforce, access to necessary technical information needed to deliver and deploy the capabilities for the warfighter
- Three ungrouped elements:
 - Facilities
 - Operational SW Deployment
 - Mgmt / Performance Measurement



Contact Info

Forrest Shull

Assistant Director of Empirical Research

Software Solutions Division

fjshull@sei.cmu.edu

703-247-1372 (Arlington)



U.S. Mail

Carnegie Mellon University
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Customer Relations

Email: info@sei.cmu.edu
Telephone: +1 412-268-5800

Web

www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm

