Army
Software Product Line Workshop

Linda Northrop
Director
Research, Technology, and System Solutions Program

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
**Army Software Product Line Workshop**

*Presented at the SEI Army Software Product Line Workshop, 12 Feb 2009, Orlando, FL.*
Software Engineering Institute (SEI)

Department of Defense R&D Laboratory (FFRDC)

Created in 1984

Under contract to Carnegie Mellon University

Offices in Pittsburgh, PA; Washington, DC; and Frankfurt, Germany

SEI Mission: advance software and related disciplines to ensure the
development and operation of systems with predictable and improved
cost, schedule, and quality.
SEI Technical Programs

Networked Systems Survivability (CERT)
- Secure Software and Systems
- Cyberthreat and Vulnerability Analysis
- Enterprise Workforce Development
- Forensics

Software Engineering Process Management (SEPM)
- Capability Maturity Model Integration (CMMI)
- Team Software Process (TSP)
- Software Engineering Measurement and Analysis (SEMA)

Research, Technology, and System Solutions (RTSS)
- Architecture-Centric Engineering
- Product Line Practice
- System of Systems Practice
- System of Systems Software Assurance
- Ultra-Large-Scale (ULS) System Perspective

Acquisition Support (ASP)

Independent Research and Development (IR&D)
SEI Technical Programs

Networked Systems Survivability (CERT)
• Secure Software and Systems
• Cyberthreat and Vulnerability Analysis
• Enterprise Workforce Development
• Forensics

Software Engineering Process Management (SEPM)
• Capability Maturity Model Integration (CMMI)
• Team Software Process (TSP)
• Software Engineering Measurement and Analysis (SEMA)

Research, Technology, and System Solutions (RTSS)
• Architecture-Centric Engineering
• Product Line Practice
• System of Systems Practice
• System of Systems Software Assurance
• Ultra-Large-Scale (ULS) System Perspective

Acquisition Support (ASP)

Independent Research and Development (IR&D)
Mission of the SEI Research, Technology, and System Solutions Program

The **Research, Technology, and System Solutions Program** enables

- cost effective
- development, evolution, and recomposition of
- predictably high-quality systems
- at all scales

With regard to its **software product line effort**, it aims to

- make product line development and acquisition a low-risk, high-return proposition for all organizations.
Some of the Organizations Using RTSS Technology

- ABB
- Honeywell
- U.S. Air Force
- EADS
- Cummins
- JAXA
- Intuit
- GM
- Raytheon
- Boeing
- United States Navy
- U.S. Army
- Airbus
- Bosch
- FedEx
- Toyota
- Thales
- Lockheed Martin
- General Dynamics
- Unisys
- NCR
- NASA
- Rockwell Collins
- Motorola
Summary of SEI Contributions

Models and Guidance
- A Framework for Software Product Line Practice℠
- Software Product Line Acquisition: A Companion to A Framework for Software Product Line Practice
- Product line practice patterns
- Product line adoption roadmap
- Pedagogical product line

Methods and Technology
- product line analysis
- architecture definition, documentation, evaluation (ATAM®), and recovery
- mining assets
- production planning
- Structured Intuitive Model for Product Line Economics (SIMPLE)
- Product Line Technical Probe℠ (PLTP℠)
- Product Line Quick Look (PLQL)
- Interactive workshops in product line measurement, variability management, product line management
- Prediction-enabled component technology

Book
Software Product Lines: Practices and Patterns

Curriculum and Certificate Programs
- Five courses and three certificate programs
- Product Line Executive Seminar

Conferences and Workshops
- SPLC 1, SPLC2, SPLC 2004; SPLC 2006; Workshops 1997 - 2005; Army Product Line Workshop 2007

Technical Reports, publications, and Web site

Software Engineering Institute
DoD Product Line Workshops

Hands-on meetings to

• identify industry-wide best practices in software product lines
• share DoD software product line practices, experiences, and issues
• discuss ways in which the current gap between commercial best practice and DoD practice can be bridged
• gather material for and review the DoD Acquisition Companion to the SEI Framework for Software Product Line Practice (Software Product Line Acquisition: A Companion to A Framework for Software Product Line Practice)
Today’s Workshop Is Funded by ASSIP

The goal of the United States Army Strategic Software Improvement Program (ASSIP) is to dramatically improve the acquisition of software-intensive systems.

ASSIP has funded the delivery of courses from the SEI Software Product Line Curriculum at Army locations and presentations on software product lines at PEO sites.

In addition, the ASSIP has funded the Army Senior Leader Program, which has involved tutorials on software architecture and software product lines among other topics.

ASSIP is funding this workshop to bring together those in the Army community who are using or trying to use product line practices.
Workshop Goals

Share Army and DoD product line practices, experience and issues, from both development and acquisition viewpoints
Examine barriers and enablers to much broader adoption of software product line practices within the Army
Determine the steps needed to make software product line practices more beneficial and relevant to Army programs
Discuss ways in which the Army’s Strategic Software Improvement Program (ASSIP) can be of assistance
Agenda

0800 – 0830    Introductions
0830 – 0915    Welcome and background: Linda Northrop, SEI
0915 – 1000    A Proactive Product Line Acquisition Approach, John Bergey, SEI
1000 – 1015    BREAK
1015 – 1045    An Approach to Product Line Acquisition Planning, Larry Jones, SEI

DoD software product line experience presentations
   1045 – 1130    Paul Jensen    Overwatch, Textron Systems
   1130 – 1215    Brian Kemper    PEO STRI
1215 – 1300    LUNCH

DoD software product line experience presentations continued
   1300 – 1345    Don Snelgrove    BAE
   1345 – 1430    Ed Dunn    NUWC
1430 – 1445    BREAK
1445 – 1600    Discussion: Product line acquisition support -needs and priorities.
1600 – 1630    Workshop Wrap-up
What Is A Software Product Line?

A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way.

- a new application of a proven concept
- an innovative, growing concept in software engineering
The Key Concepts

Use of a core asset base in production of a related set of products
The Key Concepts

Use of a core asset base in production of a related set of products

Architecture, Production Plan, Scope Definition Business Case
Widespread Use of Software Product Lines

Successful software product lines have been built for families of among other things

- mobile phones
- command and control ship systems
- satellite ground station systems
- avionics systems
- command and control/situation awareness systems
- pagers
- engine control systems
- mass storage devices
- billing systems
- web-based retail systems
- printers
- consumer electronic products
- acquisition management enterprise systems
- financial and tax systems
- medical devices
- farm fish management software
Specific Examples - 1

- **akvasmart**: Feed control and farm management software
- **Bold Stroke Avionics**: Gas turbines, train control, semantic graphics framework
- **Asea Brown Boveri**: Computer printer servers, storage servers, network camera and scanner servers
- **E-Com Technology Ltd.**: Medical imaging workstations
- **Boeing**: Customized solutions for transportation industries
- **Dialnet**: Internet payment gateway infrastructure products
- **E-Com Technology Ltd.**: AXE family of telecommunications switches
- **Ericsson**: Elevator control systems
- **Ericsson**: Software for engines, transmissions and controllers
- **Lucent Technologies**: RAID controller firmware for disk storage units
- **5ESS telecommunications switch**: Mobile phones, mobile browsers, telecom products for public, private and cellular networks
- **Software Engineering Institute**: Interferometer product line
- **LSI Logic**: Interferometer product line
- **NASA**: Interferometer product line

Linda Northrop  © 2009 Carnegie Mellon University
Specific Examples - 2

**PHILIPS**
High-end televisions, PKI telecommunications switching system, diagnostic imaging equipment

**RICOH**
Office appliances

**BOSCH**
Automotive gasoline systems

**Rockwell Collins**
Commercial flight control system avionics, Common Army Avionics System (CAAS), U.S. Army helicopters

**RaloN**
Revenue acquisition management systems

**SIEMENS**
Software for viewing and quantifying radiological images

**symbian**
EPOC operating system

**TELVENT**
Industrial supervisory control and business process management systems

**testo**
Climate and flue gas measurement devices

**NAVSEA**
Warfare Centers

**Command and control simulator for Army fire support**

**ALLTEL**
Support software

**FIDELITY NATIONAL FINANCIAL**

**MOTOROLA**
Pagers product line

**Software Engineering Institute | Carnegie Mellon**

Linda Northrop
© 2009 Carnegie Mellon University
Real World Motivation

Organizations use product line practices to:

- achieve large scale productivity gains
- improve time to market
- maintain market presence
- sustain unprecedented growth
- achieve greater market agility
- compensate for an inability to hire
- enable mass customization
- get control of diverse product configurations
- improve product quality
- increase customer satisfaction
- increase predictability of cost, schedule, and quality
Software Product Lines Value Proposition

The systematic use of software product line practices results in significant organizational benefits including

- increased quality
  - by as much as 10x

- decreased cost
  - by as much as 60%

- decreased labor needs
  - by as much as 87%

- decreased time to market (to field, to launch...)
  - by as much as 98%

- ability to move into new markets
  - in months, not years
The Value of Options

A software product line approach provides options to future market opportunities.

• The exact opportunities and their certainty are impossible to predict.
• Organizations need a way to conduct product experiments in low-cost, low-risk ways.
• Software product lines permit those kind of experiments through predefined variation points that can be exercised to meet new needs.

Options to future mission needs are important to the DoD.
Necessary Changes

The product line architecture is central to success.
The SEI Framework for Software Product Line Practice is a conceptual framework that describes the essential activities and twenty-nine practice areas necessary for successful software product lines.

The Framework, originally conceived in 1998, is evolving based on the experience and information provided by the community.

Version 4.0 – in Software Product Lines: Practices and Patterns

Version 5.0 –
http://www.sei.cmu.edu/productlines/framework.html
Three Essential Activities

All three activities are interrelated and highly iterative.

There is no “first” activity.

- In some contexts, existing products are mined for core assets.
- In others, core assets may be developed or procured for future use.

There is a strong feedback loop between the core assets and the products.

Strong management at multiple levels is needed throughout. Management oversees core asset and product development. Management orchestrates all activities and processes needed to make the three essential activities work together.
Driving the Essential Activities

Supporting the essential activities are essential practices that fall into practice areas. A **practice area** is a body of work or a collection of activities that an organization must master to successfully carry out the essential work of a product line.

Three Categories Of Practice Areas

The practice areas represent common activities in software development that are adapted to the needs of a product line approach.
## PRACTICE AREAS

<table>
<thead>
<tr>
<th>Software Engineering</th>
<th>Technical Management</th>
<th>Organizational Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture Definition</td>
<td>Configuration Management</td>
<td>Building a Business Case</td>
</tr>
<tr>
<td>Architecture Evaluation</td>
<td>Make/Buy/Mine/Commission Analysis</td>
<td>Customer Interface Management</td>
</tr>
<tr>
<td>Component Development</td>
<td>Measurement and Tracking</td>
<td>Developing an Acquisition Strategy</td>
</tr>
<tr>
<td>Mining Existing Assets</td>
<td>Process Discipline</td>
<td>Funding</td>
</tr>
<tr>
<td>Requirements Engineering</td>
<td>Scoping</td>
<td>Launching and Institutionalizing</td>
</tr>
<tr>
<td>Software System Integration</td>
<td>Technical Planning</td>
<td>Market Analysis</td>
</tr>
<tr>
<td>Testing</td>
<td>Technical Risk Management</td>
<td>Operations</td>
</tr>
<tr>
<td>Understanding Relevant Domains</td>
<td>Tool Support</td>
<td>Organizational Planning</td>
</tr>
<tr>
<td>Using Externally Available Software</td>
<td>Key: New Name and Substantial Change Substantial Change</td>
<td>Organizational Risk Management</td>
</tr>
</tbody>
</table>

**Key:**
- New Name and Substantial Change
- Substantial Change

---

**Framework Version 5.0**

**ESSENTIAL ACTIVITIES**

- Core Asset Development
- Product Development
- Management
Dilemma: How Do You Apply The 29 Practice Areas?

PRACTICE AREAS

| Software Engineering | Technical Management | Organizational Management |

GUIDANCE

Case Studies

Patterns

Probe

Curriculum
Case Studies

CelsiusTech – CMU/SEI-96-TR-016
http://www.sei.cmu.edu/publications/documents/01.reports/96.tr.016.html

Cummins, Inc. Software Product Lines: Practices and Patterns

Market Maker Software Product Lines: Practices and Patterns

NRO/Raytheon – CMU/SEI-2001-TR-030
http://www.sei.cmu.edu/publications/documents/01.reports/02tr030.html

NUWC – CMU/SEI-2002-TN-018
http://www.sei.cmu.edu/publications/documents/02.reports/02tn018.html

Salion, Inc. – CMU/SEI-2002-TR-038
http://www.sei.cmu.edu/publications/documents/02.reports/02tr038.html

http://www.sei.cmu.edu/publications/documents/05.reports/05tr019.html
Help To Make It Happen

**PRACTICE AREAS**

<table>
<thead>
<tr>
<th>Software Engineering</th>
<th>Technical Management</th>
<th>Organizational Management</th>
</tr>
</thead>
</table>

**GUIDANCE**

- Case Studies
- Patterns
- Probe
- Curriculum

*Software Engineering Institute | Carnegie Mellon*

Linda Northrop
© 2009 Carnegie Mellon University
Software Product Line Practice Patterns

- **Context**: Organizational Situation
- **Problem**: What part of a product line effort needs to be accomplished
- **Solution**: Grouping of practice areas, Relations among these practice areas (and/or groups if there is more than one)
# Current Set Of Patterns

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Variants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assembly Line</td>
<td></td>
</tr>
<tr>
<td>Cold Start</td>
<td>Warm Start</td>
</tr>
<tr>
<td>Curriculum</td>
<td></td>
</tr>
<tr>
<td>Each Asset</td>
<td>Each Asset Apprentice</td>
</tr>
<tr>
<td></td>
<td>Evolve Each Asset</td>
</tr>
<tr>
<td>Essentials Coverage</td>
<td></td>
</tr>
<tr>
<td>Factory</td>
<td>Adoption Factory</td>
</tr>
<tr>
<td>In Motion</td>
<td></td>
</tr>
<tr>
<td>Monitor</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>Process Improvement</td>
</tr>
<tr>
<td>Product Builder</td>
<td>Product Gen</td>
</tr>
<tr>
<td>Product Parts</td>
<td>Green Field</td>
</tr>
<tr>
<td></td>
<td>Barren Field</td>
</tr>
<tr>
<td></td>
<td>Plowed Field</td>
</tr>
<tr>
<td>What to Build</td>
<td>Analysis</td>
</tr>
<tr>
<td></td>
<td>Forced March</td>
</tr>
</tbody>
</table>
Help To Make It Happen

Core Asset Development  Product Development  Management

ESSENTIAL ACTIVITIES

PRACTICE AREAS

| Software Engineering | Technical Management | Organizational Management |

GUIDANCE

Case Studies  Patterns  Probe  Curriculum
What Is An SEI Product Line Technical Probe (PLTP)?

The SEI PLTP is a method for examining an organization’s readiness to adopt or ability to succeed with a software product line approach.

- It is a diagnostic tool based on the SEI Framework for Software Product Line Practice.
- The 29 practice areas are the basis of data collection and analysis.
Help To Make It Happen

PRACTICE AREAS

<table>
<thead>
<tr>
<th>Software Engineering</th>
<th>Technical Management</th>
<th>Organizational Management</th>
</tr>
</thead>
</table>

GUIDANCE

- Case Studies
- Patterns
- Probe
- Curriculum
The SEI Software Product Line Curriculum

### Three Certificate Programs

<table>
<thead>
<tr>
<th>Course</th>
<th>Professional</th>
<th>Leader</th>
<th>Team Member</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Product Lines</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Adopting Software Product Lines</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Developing Software Product Lines</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PLTP Team Training</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>PLTP Leader Training</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLTP Lead Observation</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

✓: course required to receive certificate
Adding An Adoption Roadmap

PRACTICE AREAS

| Software Engineering | Technical Management | Organizational Management |

GUIDANCE

- Case Studies
- Patterns
- Probe
- Curriculum

ADOPTION FACTORY

Software Engineering Institute | Carnegie Mellon

Linda Northrop
© 2009 Carnegie Mellon University
The Product Line Adoption Endgame

To have an operational software product line.

To do that, an organization must

• have
  – a core asset base
  – supportive processes and organizational structures

• develop products from that asset base in a way that achieves business goals

• prepare itself to institutionalize product line practices
The SEI Adoption Factory Pattern

- Phases:
  - Establish Context
  - Establish Production Capability
  - Operate Product Line

- Focus Areas:
  - Product
  - Process
  - Organization

- Each Asset:
  - What to Build
  - Product Parts
  - Product Builder

- Process Discipline:
  - Assembly Line

- Informs and information flow:
  - Cold Start
  - In Motion
  - Monitor

- Supports
## Associated Practice Areas

<table>
<thead>
<tr>
<th>Product</th>
<th>Establish Context</th>
<th>Establish Production Capability</th>
<th>Operate Product Line</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Marketing Analysis</td>
<td>• Requirements Engineering</td>
<td>• Requirements Engineering</td>
</tr>
<tr>
<td></td>
<td>• Understanding Relevant Domains</td>
<td>• Architecture Definition</td>
<td>• Architecture Definition</td>
</tr>
<tr>
<td></td>
<td>• Technology Forecasting</td>
<td>• Architecture Evaluation</td>
<td>• Architecture Evaluation</td>
</tr>
<tr>
<td></td>
<td>• Building a Business Case</td>
<td>• Mining Existing Assets</td>
<td>• Mining Existing Assets</td>
</tr>
<tr>
<td></td>
<td>• Scoping</td>
<td>• Component Development</td>
<td>• Component Development</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Using Externally Available Software</td>
<td>• Using Externally Available Software</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Software System Integration</td>
<td>• Software System Integration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Testing</td>
<td>• Testing</td>
</tr>
<tr>
<td>Process</td>
<td>• Process Discipline</td>
<td>• Make/Buy/Mine/Commission</td>
<td>• Measurement and Tracking</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Configuration Management</td>
<td>• Technical Planning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tool Support</td>
<td>• Technical Risk Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Measurement and Tracking</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Technical Planning</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Technical Risk Management</td>
<td></td>
</tr>
<tr>
<td>Organization</td>
<td>• Launching and Institutionalizing</td>
<td>• Launching and Institutionalizing</td>
<td>• Measurement and Tracking</td>
</tr>
<tr>
<td></td>
<td>• Funding</td>
<td>• Funding</td>
<td>• Technical Risk Management</td>
</tr>
<tr>
<td></td>
<td>• Structuring the Organization</td>
<td>• Structuring the Organization</td>
<td>• Organizational Risk Management</td>
</tr>
<tr>
<td></td>
<td>• Operations</td>
<td>• Operations</td>
<td>• Customer Interface Management</td>
</tr>
<tr>
<td></td>
<td>• Organizational Planning</td>
<td>• Organizational Planning</td>
<td>• Organizational Risk Management</td>
</tr>
<tr>
<td></td>
<td>• Customer Interface Management</td>
<td>• Customer Interface Management</td>
<td>• Developing an Acquisition Strategy</td>
</tr>
<tr>
<td></td>
<td>• Organizational Risk Management</td>
<td>• Organizational Risk Management</td>
<td>• Training</td>
</tr>
<tr>
<td></td>
<td>• Developing an Acquisition Strategy</td>
<td>• Developing an Acquisition Strategy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Training</td>
<td>• Training</td>
<td></td>
</tr>
</tbody>
</table>
PLP Products and Services

**Assist others**
- SEI Product Line Technical Probe
- SEI Product Line Quick Look
- Practice-specific workshops
- Planning workshops
- Connecting to other strategies

**Foster widespread awareness**
- Books
- Reports, articles, papers
- Five-course curriculum
- Executive seminar
- Conferences
- Workshops
- Website

**SEI Framework for Software Product Line Practice**

**Enable others**
- Certificate Programs
- Course licensing
- PLTP Leader Certification

**Ensure practicability**
- Methods
- Product Line Practice Patterns
- Case studies
- Adoption Roadmap
- Acquisition Companion
What’s Different About Reuse With Software Product Lines?

- Business dimension
- Iteration
- Architecture focus
- Preplanning
- Process and product connection
## Challenges - Emerging Solutions

### Variation mechanisms and variation management
- AOP/AOSD
- SOA
- End-User Programming

### Automating all or part of the production process
- PACC
- MDA
- DSL
- DDD
- Generative Programming

### Lowering adoption cost
- Agile, Phased Approaches
- Tool Support

### Distributed development and evolution
- Open Source Models
- Collaborative Environments
- Virtual Worlds

### Scaling to systems of systems and ultra-large-scale systems
- Product lines reduce interoperability issues
Product Lines of the Future

Will harness new and emerging technologies

• metadata
• automated derivation
• SOA
• end-user programming
and new forms of collaboration

• cooperative models
• globalization
• virtual worlds
• collaborative environments
to make product lines more doable, pliable, and dynamic.

Tomorrow’s product lines will accrue even greater benefits than those already demonstrated.
Ongoing SEI Product Line Research

Product derivation
  • variation mechanisms
  • production plan definition and implementation
  • product line production including automated derivation

Product line adoption strategies
  • economic models

Adapting product line concepts to exploit new technologies and serve new contexts
  • system of systems
  • service-oriented architectures
  • open source
  • globalization
  • ultra-large scale systems
Contact Information

Linda Northrop
Research, Technology, and System Solutions Program
Telephone: 412-268-7638
Email: lmn@sei.cmu.edu

U.S. Mail:
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-3890

World Wide Web:
http://www.sei.cmu.edu/productlines
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.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation 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.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under