



CarnegieMellon
Software Engineering Institute

Software Product Lines: Capitalizing on Your Process Improvement Investment

Lawrence G. Jones

Linda M. Northrop



So You've Invested in Process Improvement..

Process improvement has documented benefits.

But, there is more to software development than process.

A focus on process alone does not achieve the maximum possible organizational benefit.

A complementary focus on product is necessary.

If you develop multiple similar systems, software product lines can multiply your process improvement benefits.



Software Product Lines: Documented Benefits

Improved productivity
by as much as 10x

Decreased time to market (to field, to launch...)
by as much as an order of magnitude

Decreased cost
by as much as 60%

Decreased labor needs
by as much as 10X fewer software developers

Increased quality
by as much as 10X fewer defects



How Does Process Improvement Relate to Software Product Lines?

Process discipline is required to succeed with software product lines.

An organization's process improvement efforts poise it to succeed with software product lines.

Questions to ask

- What CMM maturity level do I have to have to be successful with product lines?
- Does my process improvement prowess guarantee my success with software product lines?



Today's Presentation

Product Line Context

The Product Line Practice Framework

CMMI Models

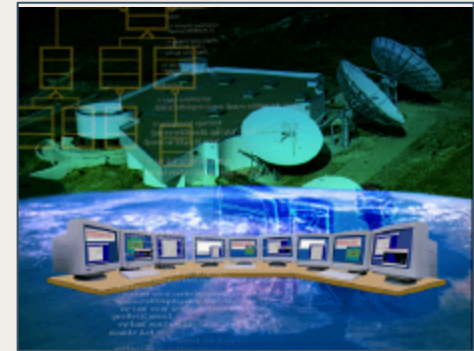
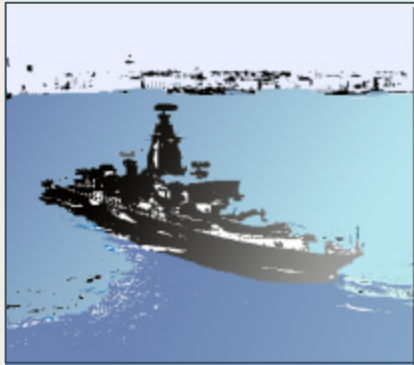
Some Framework - CMMI Relationships

Conclusions

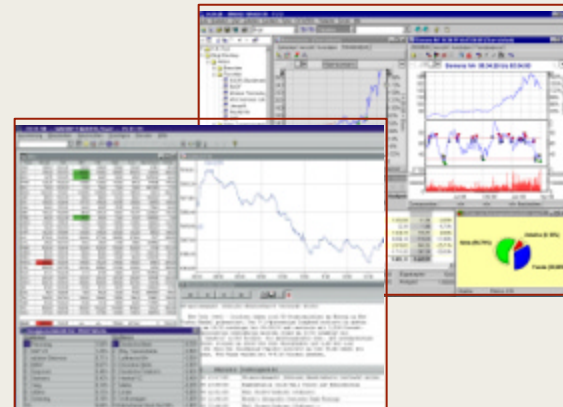


Carnegie Mellon
Software Engineering Institute

A Proven Solution



Software Product Lines





What is a Software Product Line?

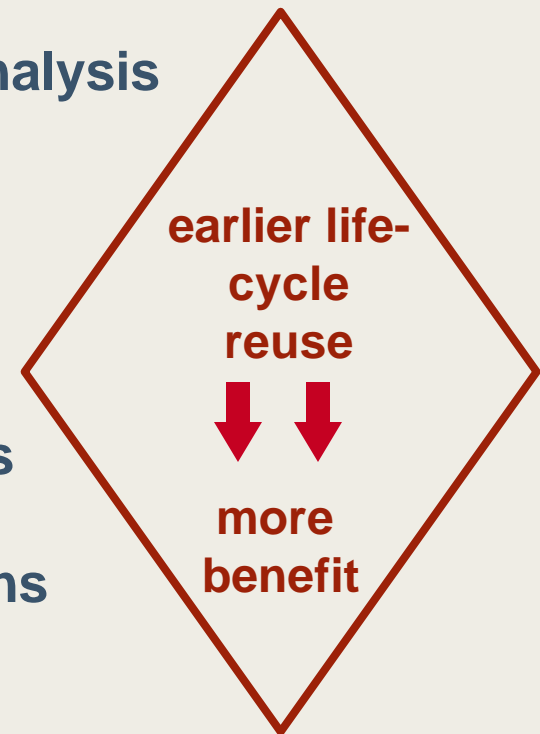
A software product line is a **set** of software-intensive systems sharing a **common, managed set of features** that satisfy the specific needs of a **particular market segment** or mission and that are **developed from a common set of core assets** in a **prescribed way**.



How Do Product Lines Help?

Product lines amortize the investment in these and other *core assets*:

- requirements and requirements analysis
- domain model
- software architecture and design
- performance engineering
- documentation
- test plans, test cases, and data
- people: their knowledge and skills
- processes, methods, and tools
- budgets, schedules, and work plans
- Software components

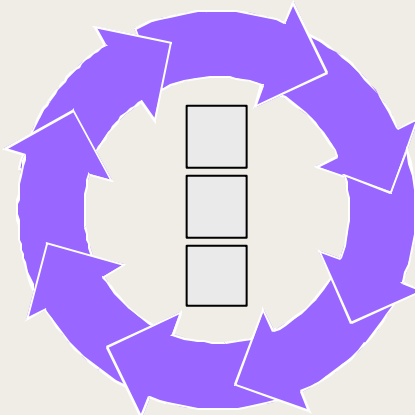


Software product lines epitomize strategic reuse.

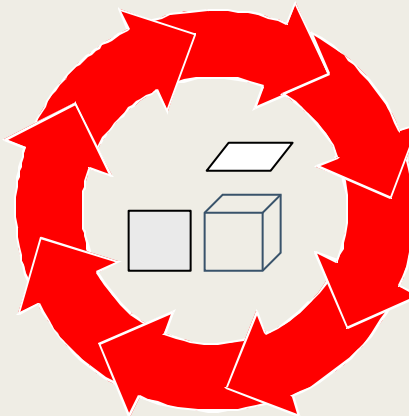


The Key Concepts

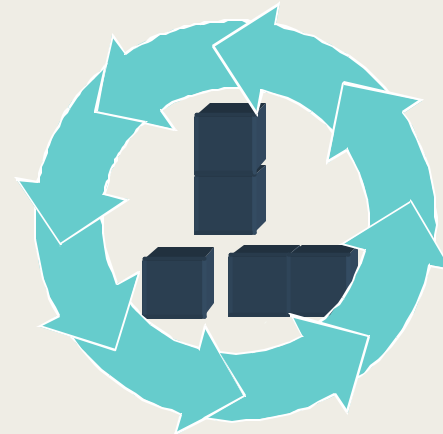
Use of a
common
asset base



in production



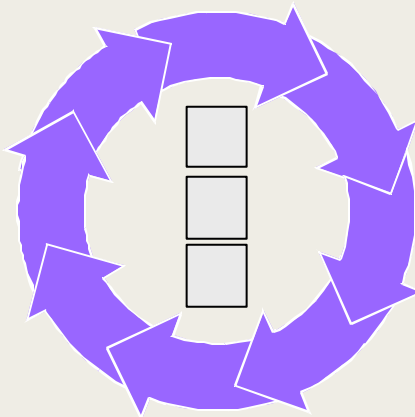
of a related
set of products





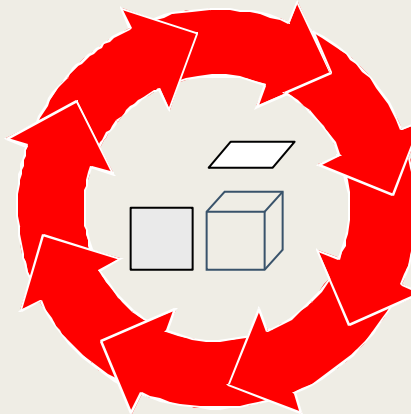
The Key Concepts

Use of a
common
asset base



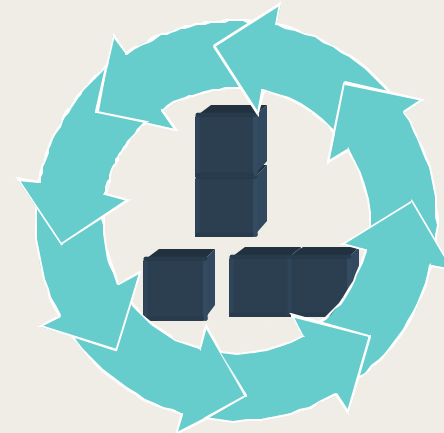
Architecture

in production



Production Plan

of a related
set of products



Scope Definition
Business Case



What's Different About Reuse with Software Product Lines?

Business dimension

Iteration

Architecture focus

Pre-planning

Process and product connection



Software Product Lines Involve

strategic
reuse

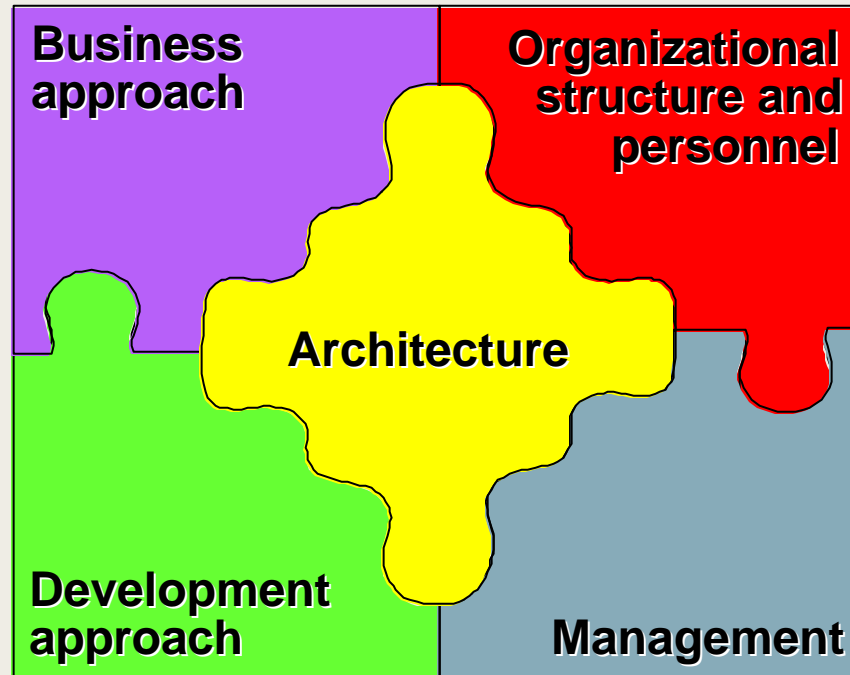


business strategy
and
technical strategy

employed to achieve explicit business goals



Necessary Changes



The architecture is the foundation of everything.



Product Line Practice

Contexts for product lines **vary** widely

- nature of products
- nature of market or mission
- business goals
- organizational infrastructure
- workforce distribution
- process discipline
- artifact maturity

But there are **universal essential elements and practices.**



CarnegieMellon
Software Engineering Institute

Today's Presentation

Product Line Context

The Product Line Practice Framework

CMMI Models

Some Framework - CMMI Relationships

Conclusions



Carnegie Mellon
Software Engineering Institute

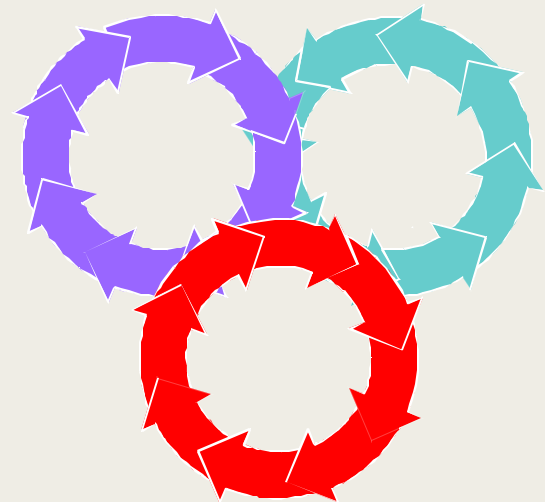
SEI Product Line Practice Framework

Conceptual framework

Describes product line essential activities

Describes essential and proven product line practices in the areas of

- software engineering
- technical management
- organizational management





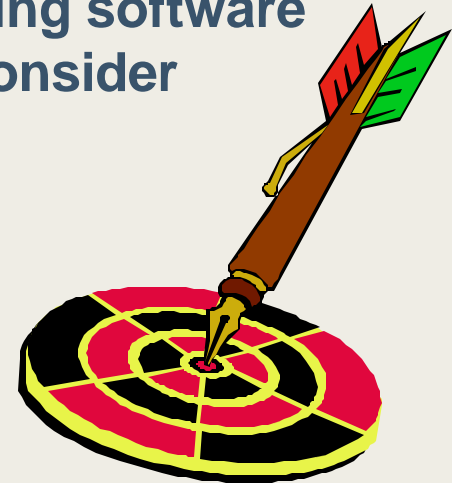
Framework Goals

Identify the foundational concepts underlying software product lines and the essential issues to consider before fielding a product line.

Identify practice areas that an organization creating or acquiring software product lines must master.

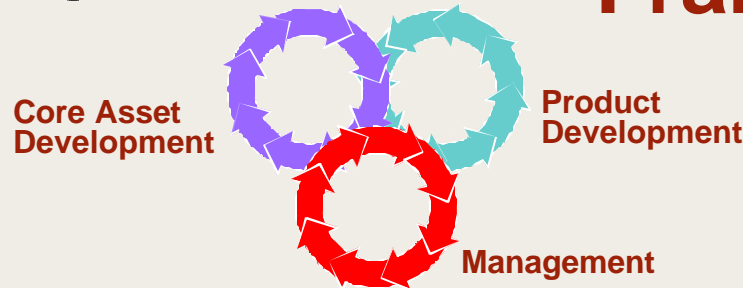
Define practices in each practice area where current knowledge is sufficient to do so.

Provide guidance to an organization about how to move to a product line approach for software.





Framework



Essential Activities

<p>Architecture Definition Architecture Evaluation Component Development COTS Utilization Mining Existing Assets Requirements Engineering Software System Integration Testing Understanding Relevant Domains</p> <p>Software Engineering</p>	<p>Configuration Management Data Collection, Metrics, and Tracking Make/Buy/Mine/Commission Analysis Process Definition Scoping Technical Planning Technical Risk Management Tool Support</p> <p>Technical Management</p>	<p>Building a Business Case Customer Interface Management Implementing an Acquisition Strategy Funding Launching and Institutionalizing Market Analysis Operations Organizational Planning Organizational Risk Management Structuring the Organization Technology Forecasting Training</p> <p>Organizational Management</p>
---	--	--

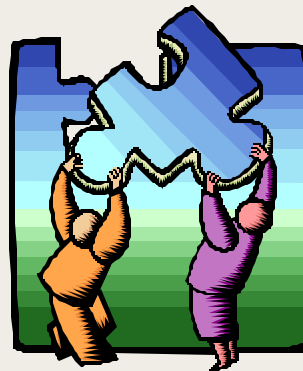
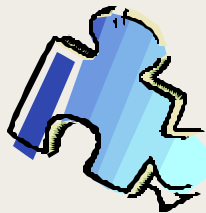
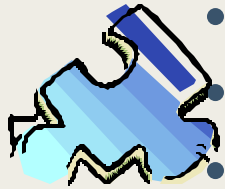
Practice Areas



Practice Area Descriptions

For individual practice areas the framework has

- Introductory description
- Aspects peculiar to product lines
- Application to core asset development
- Application to product development
- Specific practices
- Practice risks
- References





CarnegieMellon
Software Engineering Institute

Today's Presentation

Product Line Context

The Product Line Practice Framework

CMMI Models

Some Framework - CMMI Relationships

Conclusions



What is a CMMI Model?

A CMMI model contains the essential elements of effective processes

- for one or more disciplines
- structured using one of two representation schemes

Currently, there are four models:

- CMMI-SE/SW (System Engineering/Software Engineering)
- CMMI-SE/SW/IPPD
 - (adds Integrated Product and Process Development)
- CMMI/SE/SW/IPPD/SS
 - (adds Supplier Sourcing)
- CMMI-SW (removes the SE amplifications)

For each model, there are two representations published as separate documents:

- staged
- continuous



CMMI-SE/SW/IPPD/SS Process Areas (Staged)

Level	Process Areas
5 Optimizing	Organizational Innovation and Deployment Causal Analysis and Resolution
4 Quantitatively Managed	Organizational Process Performance Quantitative Project Management
3 Defined	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organizational Process Definition Organizational Training Integrated Project Management (for IPPD) Risk Management Integrated Teaming Integrated Supplier Management Decision Analysis and Resolution Organizational Environment for Integration
2 Managed	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management Measurement and Analysis Process and Product Quality Assurance Configuration Management
1 Initial	



CMMI SE/SW Continuous Representation

The Process Areas are identical.

Unlike the staged representation, the continuous representation does not specify an explicit implementation order for Process Areas.

- Free choice of implementation order is implied, *but* PA interrelationships restrict complete freedom.

Experienced implementers often take advantage of the strengths of both representations, e.g.,

- Use staged ordering as a “first cut” prioritization.
- Vary the basic implementation ordering based on business needs or “where it hurts most.”



CarnegieMellon
Software Engineering Institute

Today's Presentation

Product Line Context

The Product Line Practice Framework

CMMI Models

Some Framework - CMMI Relationships

Conclusions



Process Discipline Provides a Foundation for Product Line Practice

Product line practice involves strategic reuse.

A strategic effort requires more coordination, discipline, and commonality of approach than a more independent effort.

An organization with a culture of process discipline is better poised for product line success.

The question again is, “How much process discipline?”



CMMI - Framework Comparisons - 1

<u>Area of Comparison</u>	<u>CMMI</u>	<u>Framework</u>
Focus	generic process improvement	prescriptive for a specific approach
Coverage	Process Management Project Management Engineering Support	Software Engineering Technical Management Organizational Management
Foundational unit	Process Area	Practice Area
Diagnostic	Appraisal	Probe



CMMI - Framework Comparisons - 2

<u>Area of Comparison</u>	<u>CMMI</u>	<u>Framework</u>
Contains “How To”	No	Yes
De facto standard	Yes (SW-CMM)	No (but growing)
Maturity Levels	Yes (staged)	No
Capability Levels	Yes (continuous)	No



Process Areas (CMMI) and Practice Areas (Framework)

The most appropriate units for detailed comparison

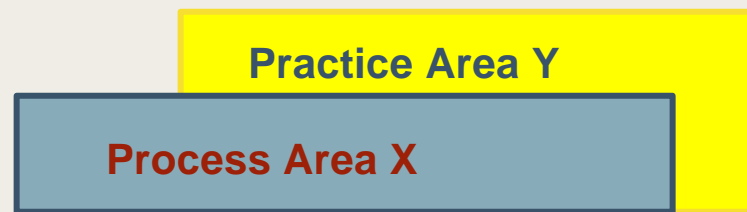
- **CMMI Process Areas**
 - Describe where an organization should have *processes*
 - 25 within CMMI-SE/SW/IPPD/SS Model
- **Framework Practice Areas**
 - Describe where an organization should have *expertise* (sometimes this includes processes)
 - 29 within the Framework



Process Areas and Practice Areas

Certain CMMI Process Areas provide a process-oriented foundation for certain other Framework Practice Areas.

This foundation may be stronger



or weaker



In no case is the process area coverage a direct substitute for the practice area coverage.

More is always required for product lines.



Process Areas that Provide a Stronger Foundation for Practice Areas

CMMI Process Areas

Configuration Management
Requirements Management
Project Planning
Organizational Training
* Measurement and Analysis
* Risk Management
* Decision Analysis & Resolution
* Technical Solution

** denotes Process Areas not found in (Software) CMM V1.1*

Framework Practice Areas

Configuration Management
Configuration Management
Technical Planning
Training
Data Collection, Metrics, and Tracking
Technical Risk Management
Make/Buy/ Mine/Commission Analysis
Make/Buy/ Mine/Commission Analysis



Process Areas that Provide a Weaker Foundation for Practice Areas - 1

CMMI Process Areas

Organizational Process Definition

Supplier Agreement Management

Project Monitoring and Control

Project Planning

* Requirements Development

* Risk Management

* Technical Solution

* Product Integration

* Verification

* Validation

Framework Practice Areas

Process Definition

Acquisition Strategy, COTS Utilization,
Make/Buy/Mine/Commission Analysis

Data Collection, Metrics, and Tracking

Organizational Planning

Requirements Engineering

Organizational Risk Management

Arch Defn, Comp Dev, COTS Util

Software System Integration

Testing, Architecture Evaluation

Testing



Process Areas that Provide a Weaker Foundation for Practice Areas - 2

CMMI Process Areas

- * Integrated Proj Mgt (IPPD)
- * Org Environment for Integration
- * Integrated Teaming
- * Organizational Innovation and Deployment
- * Integrated Supplier Management

Framework Practice Areas

- Data Collection, Metrics & Tracking
- Customer Interface Management
- Structuring the Organization
- Customer Interface Management, Structuring the Organization
- Technology Forecasting
- COTS Utilization, Developing an Acquisition Strategy, Make/Buy/Mine/Commission Analysis



In the CMMI, but *not* addressed explicitly in Framework

Organizational Process Focus Process and Product Quality Assurance

The following CMMI Process Areas pertain to process evolution from a *qualitative* emphasis to a *quantitative* emphasis and are **purposefully not addressed** in the Framework:

- Organizational Process Performance
- Quantitative Project Management
- Casual Analysis and Resolution



In the Framework, But *Not* Addressed (even weakly) by the CMMI

Software Engineering Practice Areas

- Mining Existing Assets
- Understanding Relevant Domains

Technical Management Practice Areas

- Scoping
- Tool Support

Organizational Management Practice Areas

- Building a Business Case
- Funding
- Launching and Institutionalizing
- Market Analysis
- Operations



A More Detailed Example: Configuration Management - 1

CMMI puts an emphasis on *what* to do.

CMMI Configuration Management Specific Goals

- Baselines are established
- Changes to work products are tracked and controlled
- Integrity of baselines is established and maintained

The generics describe what to do institutionally to support these specific goals, e.g.,

- train people
- assign responsibility
- provide resources



Configuration Management - 2

The Framework adds “*how to*” information to successfully perform configuration management (CM) in a product line context.

CM is more complex for a product line than for a single system. For example

- there must be CM for each version of *each product*
- because of asset sharing, a single unified CM process is needed
- core assets control must account for production by one team and parallel use by several others
- CM tools must be particularly robust

The framework provides further details.



CarnegieMellon
Software Engineering Institute

Today's Presentation

Product Line Context

The Product Line Practice Framework

CMMI Models

Some Framework - CMMI Relationships

Conclusions



Answers- 1

Process discipline provides an important foundation for software product line practice.

It would be *very useful* to be CMMI Level 2 (project focus) in this minimum set of Process Areas

- Requirements Management
- Project Planning
- Configuration Management
- Requirements Development

It would be *even more useful* to be able to standardize these processes across organizational units (Level 3).



Answers- 2

Product line practice is supported by both CMMI model representations.

- continuous (focus on the “minimum” set of Process Areas)
- staged (establish a more solid foundation with a more comprehensive set of Process Areas).

Process maturity is a very helpful foundation. However, success in software product lines requires mastery of many other essential practice areas.

- important technical and technical management practices *plus* product line extensions to CMMI Process Areas
- cross-project strategic business processes not address by CMMI models



Framework Access

Version 4.0 of the Framework can be found in *Software Product Lines: Practices and Patterns* which also contains

- Case studies
- Product line practice patterns
- Description of the Product Line Technical Probe



Version 4.1 will be found at

<http://www.sei.cmu.edu/plp/framework.html>

in early 2003.



CarnegieMellon
Software Engineering Institute

For More Details

Software Process Improvement and Product Line Practice: CMMI and the Framework for Software Product Line Practice

CMU/SEI-2002-TN-012

Available on the SEI web site at

www.sei.cmu.edu/publications/documents/02.reports/02tn012.html



Acronyms

CMMSM	Capability Maturity Model
CMMISM	Capability Maturity Model Integration
CMMI-SE	Capability Maturity Model Integrated for Software Engineering
CMMI/SE/SW	Capability Maturity Model Integrated for Systems Engineering and Software Engineering
CMMI/SE/SW/IPPD	Capability Maturity Model Integrated for Systems Engineering, Software Engineering, and Integrated Product and Process Development
CMMI/SE/SW/IPPD/SS	Capability Maturity Model Integrated for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing

SM Capability Maturity Model Integration and CMMI are service marks of Carnegie Mellon University

Sponsored by the U.S. Department of Defense

© 2002 by Carnegie Mellon University



CarnegieMellon
Software Engineering Institute

For Additional Information on Working with the SEI

Linda Northrop

Director

Product Line Systems Program

Telephone: 412-268-7638

Email: lmn@sei.cmu.edu

Larry Jones

Senior Member of the Technical
Staff

Product Line Systems Program

Telephone: 719-548-4744

Email: lgj@sei.cmu.edu

Tim Denmeade

Business Development

Product Line Systems Program

Telephone: 412-268-8243

Email: td@sei.cmu.edu

US Mail

Software Engineering Institute

Carnegie Mellon University

Pittsburgh, PA 15213-3890

World Wide Web:

<http://www.sei.cmu.edu/ata>

SEI Fax: 412-268-5758