Adopting Software Product Lines: Getting Leverage from Your Process Improvement

Larry Jones
Linda Northrop
Software Engineering Institute

Sponsored by the U.S. Department of Defense
© 2005 by Carnegie Mellon University
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
Today’s Presentation

Product Line Context

The Product Line Practice Framework
Phased Product Line Adoption
Some Framework - CMMI Relationships
Process Infrastructure Support for Product Line Adoption
Conclusion
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 test data
- people: their knowledge and skills
- processes, methods, and tools
- budgets, schedules, and work plans
- components

*product lines = strategic reuse*
What’s Different About Reuse with Software Product Lines?

Business dimension

Iteration

Architecture focus

Preplanning

Process and product connection
Today’s Presentation

Product Line Context

The Product Line Practice Framework

Phased Product Line Adoption

Some Framework - CMMI Relationships

Process Infrastructure Support for Product Line Adoption

Conclusion
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
The Goals of the Framework

The goals of the Framework are to

- Identify the foundational concepts underlying software product lines and the essential activities to consider before developing a product line.
- Identify practice areas that an organization developing 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.

The Framework is not a maturity model or a process guide.
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

*Examples of actual practice – different concept from CMMI
Today's Presentation

Product Line Context

The Product Line Practice Framework

Phased Product Line Adoption

Some Framework - CMMI Relationships

Process Infrastructure Support for Product Line Adoption

Conclusion
Dilemma: How Do You Apply the 29 Practice Areas?

Organizations still have to figure out how to put the practice areas into play.

Twenty-nine is a "big" number.

A divide and conquer approach is helpful. But an organization has to determine
  • which practice areas to concentrate on first
  • how to assign responsibility for each practice area
  • how to best "chunk" the practice areas
  • how to feed results from practice areas to each other
How Do Product Line Practice Patterns Help?

Product line practice patterns
• address recurring product line problems
• codify existing, well-proven software product line experience
• identify and specify abstractions that are broader in scope than single practice areas
• provide an additional common vocabulary for understanding product lines
• are a means of documenting new product line efforts
• help manage complexity
• can be combined to build complex product line solutions

Currently the SEI has defined 12 patterns and 11 variants.
Factory Pattern

The Factory pattern is a composite pattern that describes the entire product line organization.

![Diagram of Factory Pattern]

Dynamic Structure

© 2006 by Carnegie Mellon University
Adoption Factory Pattern

*Adoption Factory* pattern is a variant of the Factory Pattern.

**Adoption phases**
- Establish the Context
- Develop the Production Capability
- Operate the Product Line

**Focus areas**
- product
- process
- organization

© 2006 by Carnegie Mellon University
Using the Adoption Factory Pattern

To use the Adoption Factory pattern as a roadmap

- Elaborate the practice areas associated with its subpatterns.
- Plan to master these practice areas in a continuous way that begins at the phase where they first appear.
Today’s Presentation

Product Line Context

The Product Line Practice Framework

Phased Product Line Adoption

Some Framework - CMMI Relationships

Process Infrastructure Support for Product Line Adoption

Conclusion
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.

Some questions are

- How much and what kind of process discipline is necessary?
- How much of a basis do CMMI processes provide?
### CMMI-SE/SW/IPPD/SS Process Areas (Staged)

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

© 2006 by Carnegie Mellon University
## CMMI - Framework Comparisons - 1

<table>
<thead>
<tr>
<th>Area of Comparison</th>
<th>CMMI</th>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus</td>
<td>generic process improvement</td>
<td>prescriptive for a specific approach</td>
</tr>
<tr>
<td>Coverage</td>
<td>Process Management</td>
<td>Software Engineering</td>
</tr>
<tr>
<td></td>
<td>Project Management</td>
<td>Technical Management</td>
</tr>
<tr>
<td></td>
<td>Engineering</td>
<td>Organizational Management</td>
</tr>
<tr>
<td>Foundational unit</td>
<td>Process Area</td>
<td>Practice Area</td>
</tr>
<tr>
<td>Diagnostic</td>
<td>Appraisal</td>
<td>Probe</td>
</tr>
</tbody>
</table>

© 2006 by Carnegie Mellon University
## CMMI - Framework Comparisons - 2

<table>
<thead>
<tr>
<th>Area of Comparison</th>
<th>CMMI</th>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains &quot;How To&quot;</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>De facto standard</td>
<td>Yes (SW-CMM)</td>
<td>No (but growing)</td>
</tr>
<tr>
<td>Maturity Levels</td>
<td>Yes (staged)</td>
<td>No</td>
</tr>
<tr>
<td>Capability Levels</td>
<td>Yes (continuous)</td>
<td>No</td>
</tr>
</tbody>
</table>

© 2005 by Carnegie Mellon University
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

<table>
<thead>
<tr>
<th>CMMI Process Areas</th>
<th>Framework Practice Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Management</td>
<td>Configuration Management</td>
</tr>
<tr>
<td>Requirements Management</td>
<td>Configuration Management</td>
</tr>
<tr>
<td>Project Planning</td>
<td>Technical Planning</td>
</tr>
<tr>
<td>Organizational Training</td>
<td>Training</td>
</tr>
<tr>
<td>Measurement and Analysis</td>
<td>Data Collection, Metrics, and Tracking</td>
</tr>
<tr>
<td>Risk Management</td>
<td>Technical Risk Management</td>
</tr>
<tr>
<td>Decision Analysis &amp; Resolution</td>
<td>Make/Buy/ Mine/Commission Analysis</td>
</tr>
<tr>
<td>Technical Solution</td>
<td>Make/Buy/ Mine/Commission Analysis</td>
</tr>
</tbody>
</table>

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

© 2006 by Carnegie Mellon University
**Process Areas that Provide a Weaker Foundation for Practice Areas - 1**

<table>
<thead>
<tr>
<th>CMMI Process Areas</th>
<th>Framework Practice Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational Process Definition</td>
<td>Process Definition</td>
</tr>
<tr>
<td>Supplier Agreement Management</td>
<td>Acquisition Strategy, COTS Utilization, Make/Buy/Mine/Commission Analysis</td>
</tr>
<tr>
<td>Project Monitoring and Control</td>
<td>Data Collection, Metrics, and Tracking</td>
</tr>
<tr>
<td>Project Planning</td>
<td>Organizational Planning</td>
</tr>
<tr>
<td>* Requirements Development</td>
<td>Requirements Engineering</td>
</tr>
<tr>
<td>* Risk Management</td>
<td>Organizational Risk Management</td>
</tr>
<tr>
<td>* Technical Solution</td>
<td>Arch Defn, Comp Dev, COTS Util</td>
</tr>
<tr>
<td>* Product Integration</td>
<td>Software System Integration</td>
</tr>
<tr>
<td>* Verification</td>
<td>Testing, Architecture Evaluation</td>
</tr>
<tr>
<td>* Validation</td>
<td>Testing</td>
</tr>
</tbody>
</table>

© 2006 by Carnegie Mellon University
### 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

© 2006 by Carnegie Mellon University
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

© 2006 by Carnegie Mellon University
### CMMI Support for Adoption Factory

#### Establish Context
- Marketing Analysis
- Understanding Relevant Domains
- Technology Forecasting
- Building a Business Case
- Scoping

#### Establish Production Capability
- Requirements Engineering
- Architecture Definition
- Architecture Evaluation
- Mining Existing Assets
- Component Development
- COTS Utilization
- Software System Integration
- Testing

#### Operate Product Line
- Requirements Engineering
- Architecture Definition
- Architecture Evaluation
- Mining Existing Assets
- Component Development
- COTS Utilization
- Software System Integration
- Testing

#### Process Definition
- Make/Buy/Mine/Commission
- Configuration Management
- Tool Support
- Data Collection, Metrics, Tracking
- Technical Planning
- Technical Risk Management

#### Organization
- Launching and Institutionalizing Funding
- Structuring the Organization Operations
- Organizational Planning
- Customer Interface Management
- Organizational Risk Management
- Developing an Acquisition Strategy
- Training

#### Previous Next

Exit Slide Show
But There’s More …

Even if you have mature CMMI processes in place, as we have seen, product line processes always have special aspects, many with process implications.

These special aspects are found in the Framework for each practice area

- Aspects Peculiar to Product Lines
- Application to Core Asset Development
- Application to Product Development
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
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.
Which CMMI Model Representation Supports Software Product Lines?

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
Leveraging CMMI Process Areas to Software Product Lines

If you are in the early stages of CMMI adoption and are still choosing processes to implement, 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).
Today’s Presentation

Product Line Context

The Product Line Practice Framework

Phased Product Line Adoption

Some Framework - CMMI Relationships

Process Infrastructure Support for Product Line Adoption

Conclusion
Process Improvement Infrastructure

A typical process improvement infrastructure includes
- organizational elements for oversight & implementation
  - MSG, (SE)PG, PAT
- generic process assets (process asset library)
- training infrastructure
- other change management assets
  - change & resistance management, sponsorship, teaming
- ... many other things are possible

An existing process improvement infrastructure might be augmented (or copied) to provide support for software product line adoption.

Controlled adaptation and reuse of these infrastructure assets is absolutely consistent with the notion of a product line core asset base.
Today’s Presentation

Product Line Context

The Product Line Practice Framework

Phased Product Line Adoption

Some Framework - CMMI Relationships

Process Infrastructure Support for Product Line Adoption

Conclusion
Conclusions

A process improvement program provides a basis for success with product lines.

You can get leverage from
  • CMMI models (both representations)
    - but additional practices are necessary for product lines
  • your process improvement infrastructure

Software product line practice built on a foundation of process discipline yields significant paybacks.
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 4.2 can be found at

Acronyms

CMM<sup>SM</sup>  Capability Maturity Model
CMMI<sup>SM</sup>  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

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

**Linda Northrop**
Director
Product Line Systems Program
Telephone: 412-268-7638
Email: lmn@sei.cmu.edu

**Larry Jones**
Product Line Systems Program
Telephone: 719-548-4744
Email: lji@sei.cmu.edu

**U.S. mail:**
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

**World Wide Web:**
http://www.sei.cmu.edu/ata
http://www.sei.cmu.edu/plp

**SEI Fax:** 412-268-5758

© 2006 by Carnegie Mellon University