Introduction to

Software Product Line Adoption

Linda Northrop: Director, Product Line Systems Program
lmn@sei.cmu.edu

Larry Jones: Product Lines Systems Program
lgj@sei.cmu.edu

Software Engineering Institute

Sponsored by the U.S. Department of Defense
© 2005 by Carnegie Mellon University
<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEP 2005</td>
<td></td>
<td>00-00-2005 to 00-00-2005</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to Software Product Line Adoption</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5a. CONTRACT NUMBER</th>
<th>5b. GRANT NUMBER</th>
<th>5c. PROGRAM ELEMENT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5d. PROJECT NUMBER</th>
<th>5e. TASK NUMBER</th>
<th>5f. WORK UNIT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, 15213</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. PERFORMING ORGANIZATION REPORT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10. SPONSOR/MONITOR’S ACRONYM(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>11. SPONSOR/MONITOR’S REPORT NUMBER(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>12. DISTRIBUTION/AVAILABILITY STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release; distribution unlimited</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14. ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15. SUBJECT TERMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT unclassified</td>
</tr>
<tr>
<td>b. ABSTRACT unclassified</td>
</tr>
<tr>
<td>c. THIS PAGE unclassified</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17. LIMITATION OF ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same as Report (SAR)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>18. NUMBER OF PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>107</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
Tutorial Objectives

This tutorial will acquaint participants with
• issues surrounding software product line adoption
• a phased, pattern-based adoption approach
• adoption planning artifacts
• explicit linkage of software product line adoption with other improvement efforts
Tutorial Outline

About Software Product Line Adoption
  • Background
  • Benefits
  • Barriers
  • Risks
  • Plans
  • Technology Change

Phased Product Line Adoption: a Roadmap
Phased Technology Adoption
Using the Adoption Roadmap
Example Product Line Plans
Connections to Other Improvement Initiatives
Conclusion
Product Line Adoption

**Product line adoption** involves moving from some form of developing software-intensive systems with a single-system mentality to developing them as 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.
The Adoption Endgame

Effectively achieve an operational product line.
• have
  - a core asset base
  - supportive processes and organizational structures
• develop products from that asset base in a way that achieves business goals
• improve and extend the software product line adoption effort as long as it makes sense
Essential Product Line Activities

Each of these is essential, as is the blending of all three.
Barriers to Product Line Adoption

Cost, cost, and cost....
You have to invest to eventually save.
Barriers to Product Line Adoption

Time, time, and time
More Barriers

Lack of knowledge
Need for organizational change
Cultural resistance
Lack of sufficient management support
Lack of necessary talent
Incompatible development processes
Globalization of workforce
Stove-piped mentality
No clear path to follow
Others??????
What is the SEI Framework for Software Product Line Practice℠?

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 4.2 –
http://www.sei.cmu.edu/productlines/framework.html
### SEI Framework

#### Essential Activities

<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>Data Collection, Metrics, and Tracking</td>
<td>Customer Interface Management</td>
</tr>
<tr>
<td>Component Development</td>
<td>Make/Buy/Mine/Commission Analysis</td>
<td>Developing an Acquisition Strategy</td>
</tr>
<tr>
<td>COTS Utilization</td>
<td>Process Definition</td>
<td>Funding</td>
</tr>
<tr>
<td>Mining Existing Assets</td>
<td>Scoping</td>
<td>Launching and Institutionalizing</td>
</tr>
<tr>
<td>Requirements Engineering</td>
<td>Technical Planning</td>
<td>Market Analysis</td>
</tr>
<tr>
<td>Software System Integration</td>
<td>Technical Risk Management</td>
<td>Operations</td>
</tr>
<tr>
<td>Testing</td>
<td>Tool Support</td>
<td>Organizational Planning</td>
</tr>
<tr>
<td>Understanding Relevant Domains</td>
<td></td>
<td>Organizational Risk Management</td>
</tr>
</tbody>
</table>

### Practice Areas
“Launching and Institutionalizing” Practice Area  -  1

The “Launching and Institutionalizing” practice area is about making the change to a product line approach.

It is about moving from a given level of product line sophistication to a higher level.

It is this practice area that describes the act of product line adoption and involves judicious and timely application of product line practices.
All organizations launch and institutionalize change.

Product line adoption is such a change.
  • Technology change experts have models and practices to assist in ensuring successful change.
  • These have to be adapted for software product line adoption.
  • What you need to do is launch and institutionalize practices in each of the 29 practice areas.
  • How you go about doing that depends on specific organizational context and the change models and practices you use.

Adoption plans are an important output of this practice area. They specify the specific approach an organization takes in launching its product line effort.
Technology Change Essentials

“Managing Technological Change”
Carnegie Mellon University
Software Engineering Institute

© 2005 by Carnegie Mellon University
Product Line Adoption Plans

In order to launch a product line, an organization needs to plan its attack.

In any organization, there may be a hierarchical set of goals, strategies, and plans.

Organizations usually decide to adopt a product line approach as a strategy to achieve specific business goals. Product line adoption may in fact be a strategy in a business plan.

Adopting a software product line then becomes the goal of a product line adoption plan, which describes how the necessary product line practices are to be rolled out across the organization.
A Hierarchy of Plans

Business Plan

Business goals

Strategy: Adopt a Product Line Approach

Product Line Adoption Plan

Action Plan

Action Plan

Action Plan
Factors Influencing Adoption

Organizational Context

product line readiness
barriers
enablers
unique characteristics
culture
other ongoing activities
Factors Influencing Adoption

Organizational Context
- product line readiness
- barriers
- enablers
- unique characteristics
- culture
- other ongoing activities

Product Line Adoption Plan

Adoption Support
- The Framework
- product line approaches
- product line adoption roadmap
- change management mechanisms
- change models
- planning process

Product Line Action Plans
Tutorial Outline

About Software Product Line Adoption
  • Background
  • Benefits
  • Barriers
  • Risks
  • Plans
  • Technology Change

Phased Product Line Adoption: a Roadmap
Phased Technology Adoption
Using the Adoption Roadmap
Example Product Line Plans
Connections to Other Improvement Initiatives
Conclusion
Patterns

Patterns are a way of expressing common context and problem-solution pairs.

Patterns have been found to be useful in building architecture, economics, software architecture, software design, software implementation, process improvement, and other areas.

Patterns help effect a divide-and-conquer approach.

We have defined *software product line practice patterns*, which will assist in planning and effecting product line adoption.
Software Product Line Practice Pattern

Pattern

- **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)
Factory Pattern - 1

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

Context:
An organization is considering (or fielding) a product line.

Problem:
To map the entire product line effort
Factory Pattern - 2

Static:
The *Factory* pattern consists of the following subpatterns:

<table>
<thead>
<tr>
<th>Subpattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What to Build</td>
<td>yields the set of products to be included in the product line along with an associated business case</td>
</tr>
<tr>
<td>Each Asset</td>
<td>provides individual core assets and their attached processes</td>
</tr>
<tr>
<td>Product Parts</td>
<td>supplies the core assets from which products will be built</td>
</tr>
<tr>
<td>Assembly Line</td>
<td>provides the production infrastructure</td>
</tr>
<tr>
<td>Product Builder</td>
<td>yields the individual products in the product line</td>
</tr>
<tr>
<td>Cold Start</td>
<td>prepares the organization for its first product line operation</td>
</tr>
<tr>
<td>In Motion</td>
<td>keeps the product line organization running</td>
</tr>
<tr>
<td>Monitor</td>
<td>keeps watch on the organization and responds with any needed changes</td>
</tr>
</tbody>
</table>
Factory Pattern - 3

Dynamic Structure
A Variant for Adoption

The **Factory** pattern is already a roadmap for the entire product line organization:
- a top-down view of the product line organization
- a blueprint for a divide-and-conquer strategy

Organizations that lack the ability to define and follow processes, even lightweight or agile ones, need to address that deficiency early in their adoption path.

Even though the “Process Definition” practice area is part of the Assembly Line pattern, it is called out separately in a variant on the **Factory** pattern.

The variant is called the **Adoption Factory** pattern.
Adoption Factory Pattern - 1

Dynamic Structure
Adoption Factory Pattern - 2

The Adoption Factory provides the necessary abstraction of the major product line activities involved and their dependencies.

Owing to the highly iterative nature of product line adoption and operations, the arrows should never be interpreted as suggesting strictly linear dependencies.

The Adoption Factory lays out the technology change that needs to occur in moving to a software product line approach. It does NOT provide change management mechanisms.
Useful Views

When using the Adoption Factory pattern to plan, analyze, and implement an organization’s specific product line adoption activities, it is useful to portray the roadmap from the following six different views:

1. Adoption Phases
2. Focus Areas
3. Phases and Focus Areas
4. Practice Areas
5. Outputs
6. Roles
Phases and Focus Area View

Establish Context

Establish Production Capability

Operate Product Line

Product

What to Build

Product Parts

Product Builder

Process

Process Definition

Assembly Line

Organization

Cold Start

In Motion

Monitor

Adoption Factory Pattern

Informs
## Associated Practice Areas

<table>
<thead>
<tr>
<th>Establish Context</th>
<th>Establish Production Capability</th>
<th>Operate Product Line</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product</strong></td>
<td><strong>Process</strong></td>
<td><strong>Organization</strong></td>
</tr>
<tr>
<td>Marketing Analysis</td>
<td>Requirements Engineering</td>
<td>Requirements Engineering</td>
</tr>
<tr>
<td>Understanding Relevant Domains</td>
<td>Architecture Definition</td>
<td>Architecture Definition</td>
</tr>
<tr>
<td>Technology Forecasting</td>
<td>Architecture Evaluation</td>
<td>Architecture Evaluation</td>
</tr>
<tr>
<td>Building a Business Case</td>
<td>Mining Existing Assets</td>
<td>Mining Existing Assets</td>
</tr>
<tr>
<td>Scoping</td>
<td>Component Development</td>
<td>Component Development</td>
</tr>
<tr>
<td></td>
<td>COTS Utilization</td>
<td>COTS Utilization</td>
</tr>
<tr>
<td></td>
<td>Software System Integration</td>
<td>Software System Integration</td>
</tr>
<tr>
<td></td>
<td>Testing</td>
<td>Testing</td>
</tr>
</tbody>
</table>

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

**Launched and Institutionalizing**
- Launching and Institutionalizing Funding
- Launching and Institutionalizing Operations
- Organizational Planning
- Customer Interface Management
- Organizational Risk Management
- Developing an Acquisition Strategy
- Training

**Data Collection, Metrics and Tracking**
- Technical Risk Management
- Organizational Risk Management
- Customer Interface Management
- Organizational Planning

**Structure**
- Structuring the Organization
- Operations
- Organizational Planning
- Customer Interface Management
- Organizational Risk Management
- Developing an Acquisition Strategy
- Training
Outputs View

Another useful and more detailed perspective of the Phases and Focus Areas view can be obtained by listing the outputs typically generated in each of the nine cells.

The information in this view can serve as a handy checklist for representative output from each phase.

For details see page 17 of *Software Product Line Adoption Roadmap*
CMU/SEI-2004-TR-022
## Outputs View - 2

<table>
<thead>
<tr>
<th>Product Outputs</th>
<th>Establish Context Phase</th>
<th>Establish Production Capability Phase</th>
<th>Operate Product Line Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• marketing description</td>
<td>• product line requirements</td>
<td>• product requirements</td>
</tr>
<tr>
<td></td>
<td>• domain model</td>
<td>• product line architecture</td>
<td>• product architecture</td>
</tr>
<tr>
<td></td>
<td>• technology survey</td>
<td>• documentation</td>
<td>documentation</td>
</tr>
<tr>
<td></td>
<td>• economic model</td>
<td>• product line architecture</td>
<td>• product architecture</td>
</tr>
<tr>
<td></td>
<td>• business use cases</td>
<td>• evaluation report</td>
<td>evaluation report</td>
</tr>
<tr>
<td></td>
<td>• cost/benefit model</td>
<td>• asset inventory</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• business case</td>
<td>• mining plan and process</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• scope definition</td>
<td>• mined assets</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• commercial off-the-shelf (COTS) criteria</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• COTS assets</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• core components</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• product line test strategy, test cases, test architecture, test scripts, and test plan</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• attached processes</td>
<td></td>
</tr>
</tbody>
</table>

Operate Product Line Phase

- product requirements
- product architecture documentation
- product architecture evaluation report
- product specific components (mined, COTS, or built new)
- product test strategy, test cases, test architecture, test plan
# Outputs View - 3

<table>
<thead>
<tr>
<th>Process outputs</th>
<th>Establish Context Phase</th>
<th>Establish Production Capability Phase</th>
<th>Operate Product Line Phase</th>
</tr>
</thead>
</table>
| defined processes for | • requirements engineering  
| | • project management  
| | • software configuration management  
| | • development  
| | • testing  
| | • risk management  
| | • architecture conformance  
| | • configuration management process for product lines  
| | • tool support list  
| | • development tool set  
| | • production tool set  
| | • measurement plan  
| | • core asset metrics  
| | • core asset work plans  
| | • production plan |
### Outputs View - 3

<table>
<thead>
<tr>
<th>Organization outputs</th>
<th>Establish Context Phase</th>
<th>Establish Production Capability Phase</th>
<th>Operate Product Line Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• adoption plan</td>
<td>• progress reports</td>
<td>• organizational metrics</td>
</tr>
<tr>
<td></td>
<td>• funding model</td>
<td>• risks and mitigation strategies</td>
<td>• cost/pricing model</td>
</tr>
<tr>
<td></td>
<td>• organization chart</td>
<td></td>
<td>• product release strategy</td>
</tr>
<tr>
<td></td>
<td>• product line concept</td>
<td></td>
<td>• trouble reports</td>
</tr>
<tr>
<td></td>
<td>of operations (CONOPS)</td>
<td></td>
<td>• customer feedback</td>
</tr>
<tr>
<td></td>
<td>• marketing plan</td>
<td></td>
<td>• upgraded plans</td>
</tr>
<tr>
<td></td>
<td>• product proposals</td>
<td></td>
<td>• improvement suggestions</td>
</tr>
<tr>
<td></td>
<td>• acquisition strategy</td>
<td></td>
<td>• risks and mitigation</td>
</tr>
<tr>
<td></td>
<td>• organization risk</td>
<td></td>
<td>strategies</td>
</tr>
<tr>
<td></td>
<td>management plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>or process</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• training plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• product line training</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Roles View

Another instructive view depicts the type of people who need to be involved in the product line adoption effort.

The Roles View lists the typical roles associated with each cell of the Phases and Focus Areas view.

This view can be used for identifying staffing needs and making assignments.

Some roles may appear in multiple phases, but the tasks those roles perform will vary with the phase.

See page 19 of
Software Product Line Adoption Roadmap
CMU/SEI-2004-TR-022
## Roles View - 2

<table>
<thead>
<tr>
<th>Product-related roles</th>
<th>Establish Context Phase</th>
<th>Establish Production Capability Phase</th>
<th>Operate Product Line Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• marketer</td>
<td>• core asset developer:</td>
<td>• product developer:</td>
</tr>
<tr>
<td></td>
<td>• market analyst</td>
<td>• requirements engineer</td>
<td>• requirements engineer</td>
</tr>
<tr>
<td></td>
<td>• domain expert</td>
<td>• architect</td>
<td>• architect</td>
</tr>
<tr>
<td></td>
<td>• product manager</td>
<td>• architecture evaluator</td>
<td>• architecture evaluator</td>
</tr>
<tr>
<td></td>
<td>• senior manager</td>
<td>• component developer</td>
<td>• component developer</td>
</tr>
<tr>
<td></td>
<td>• technology scout</td>
<td>• tester</td>
<td>• tester</td>
</tr>
<tr>
<td></td>
<td>• architect</td>
<td>• software integrator</td>
<td>• software integrator</td>
</tr>
</tbody>
</table>
## Roles - 3

<table>
<thead>
<tr>
<th>Process-related roles</th>
<th>Establish Context Phase</th>
<th>Establish Production Capability Phase</th>
<th>Operate Product Line Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• technical manager</td>
<td>• technical manager</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• process owner</td>
<td>• process owner</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• process group member</td>
<td>• process group member</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• technical support</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• tool specialist</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• measurement specialist</td>
<td></td>
</tr>
</tbody>
</table>
## Roles - 4

<table>
<thead>
<tr>
<th>Organization-related roles</th>
<th>Establish Context Phase</th>
<th>Establish Production Capability Phase</th>
<th>Operate Product Line Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>product line manager</td>
<td>product line manager</td>
<td>product line manager</td>
<td>product line manager</td>
</tr>
<tr>
<td>software manager</td>
<td>software manager</td>
<td>software manager</td>
<td>product manager</td>
</tr>
<tr>
<td>business unit or organization manager</td>
<td>business unit or organization manager</td>
<td>financial manager</td>
<td>product manager</td>
</tr>
<tr>
<td>product manager</td>
<td>product manager</td>
<td>financial manager</td>
<td>business unit or organization manager</td>
</tr>
<tr>
<td>acquisition expert</td>
<td>acquisition expert</td>
<td>training developer</td>
<td>business unit or organization manager</td>
</tr>
<tr>
<td>financial manager</td>
<td>financial manager</td>
<td>trainer</td>
<td>financial manager</td>
</tr>
<tr>
<td>human resource manager</td>
<td>human resource manager</td>
<td></td>
<td>customer field representative</td>
</tr>
<tr>
<td>training planner</td>
<td>training planner</td>
<td></td>
<td></td>
</tr>
<tr>
<td>training developer</td>
<td>training developer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trainer</td>
<td>trainer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Pattern and Practice Area Sequencing

Guidelines:
1. Use the Adoption Factory pattern and its associated views as an overall layout of what needs to be accomplished.
2. Plan to master the practice areas in a continuous way that begins at the phase where each practice area first appears.
3. Use organization-specific information to focus first on those practice areas that have the most immediate impact.
4. Look across the phase horizon, and, where it makes sense, begin early to prepare for those activities presenting the greatest challenge.
5. During the adoption process, iterate back and address practice areas that were initially covered lightly.
Exercise

How would you use the Adoption Factory Pattern to assist the software development manager in this scenario?

Scenario:
The software development manager of a robot manufacturer has launched an initial product line effort for the software in its line of warehouse robots. He started by defining a software architecture for the entire family of robots. The architects are struggling with the amount of variability they have to contend with, and the developers are not used to following the dictates of an architecture, much less a common one. He is wondering if there would have been a better way to begin product line adoption and would like some guidance as to how organizations should proceed, what activities he might have missed, what midcourse corrections he can take, and who he should involve.
Tutorial Outline

About Software Product Line Adoption
Phased Product Line Adoption: a Roadmap

Phased Technology Adoption
  • Entry Criteria
  • Initiating
  • Diagnosing
  • Establishing (Planning)
  • Acting
  • Learning

Using the Adoption Roadmap
Example Product Line Plans
Connections to Other Improvement Initiatives
Conclusion
Some Phased Approaches to Change

A simple gap analysis approach:
  • Determine where you are.
  • Determine where you want to be.
  • Analyze the gap between.
  • Make a plan to overcome the gap.
  • Execute the plan.
  • Learn lessons and do it again.

A popular approach: Plan Do Check Act.

A more formal approach: IDEAL model (process improvement)

Other approaches
  • Win-Win Spiral (software development)
  • Six Sigma (process improvement)
IDEAL: An Improvement Approach

The IDEALSTM Model

**Initiating**
- Stimulus for improvement
- Set context and establish sponsorship
- Establish improvement infrastructure

**Diagnosing**
- Appraise and characterize current practice
- Develop recommendations and document phase results
- Set priorities and strategies

**Learning**
- Revise organizational approach
- Document and analyze lessons

**Acting**
- Define processes and measures
- Plan and execute pilots
- Plan, execute, and track installation
- Establish process action teams
- Plan actions

[McFeeley 96]

SM IDEAL is a service mark of Carnegie Mellon University.
Using IDEAL for Product Lines

Tailor the detailed activities to fit the product line approach.
  • The IDEAL model was defined with process improvement in mind.
  • The IDEAL model must be “informed” by good product line guidance.

IDEAL can be a useful guide for the “Launching and Institutionalization” practice area.

Understand that there are special entry criteria for product line adoption.
  • Product line adoption is not as “universally” applicable as process improvement.
Entry Criteria for Product Lines

Is there an overall fit for a software product line approach?

- Are there multiple systems with sufficient commonality?
- Does the organization have articulated goals it is trying to achieve with a software product line approach?
- Do the benefits of successful product lines match the goals of the organization?
- Is there sufficient support within the organization to launch a software product line adoption effort?
Once a product line approach has been deemed appropriate to pursue further

- establish sponsorship
- promote management and staff awareness
- obtain staffing and resource commitments
  - this includes the infrastructure to oversee the product line adoption, e.g., product line manager and staff
- set product line adoption goals
Diagnosing: Checking Product Line Conditions - 1

Diagnostics you might perform

• Evaluate the business and technical viability of the product line opportunity.
• Examine the product line context.
  - market
  - organization
  - business unit
  - individuals
• Identify organizational strengths and weaknesses related to change implementation.
Diagnosing: Checking Product Line Conditions - 2

• Analyze the organization against the 29 practice areas from the Framework.
  - Are the right set of practices in place for single system development?
  - Is there knowledge about how to transform these practices into product line practices?
  - Is there knowledge about how to invent or choose new product line-specific practices?
  - Is there sufficient discipline to adhere to product line processes and practices?

Some diagnostics include recommendations.
Product Line Diagnostic Instruments

SEI Product Line Quick Look\textsuperscript{SM} (PLQL\textsuperscript{SM})

SEI Product Line Technical Probe\textsuperscript{SM} (PLTP\textsuperscript{SM})

Bosch Product Line Potential Analysis*

European Union ITEA (Information Technology for European Advancement) BAPO (Business, Architecture, Process, Organization) evaluation*

Others?

* Software Product Lines: Third International Software Product Line Conference, Boston, MA, August, 2004
What Is The SEI Product Line Technical Probe (PLTP)?

A method for examining an organization’s readiness to adopt or ability to succeed with a software product line approach

• diagnostic tool based on the SEI Framework for Software Product Line Practice
• Practice areas are the basis of data collection and analysis.

Outcome is a set of findings that portray organizational
• strengths
• challenges
with regard to a product line approach
Establishing: Planning the Product Line Adoption

While considering organizational and technical context:

• Choose an appropriate product line approach.
• Set priorities.
• Develop an overall product line adoption plan.
• Develop lower level action plans to
  - improve organizational capabilities
  - specify how a pilot will be implemented
  - implement one or more practices
  - address change issues
Using Pilots

Pilot projects can be an important way to reduce risk, learn more, and build advocacy. A pilot may be implemented as a complete iteration of the IDEAL model.

The criteria for choosing a pilot include
• scope: The pilot should be done in a relatively short time frame with reasonable resources.
• importance and visibility: The organization should care whether the pilot succeeds. But the pilot should not be so important that its failure would be disastrous.
• probability of success: The effort should have a reasonable chance to succeed.
• choice of participants: Participants in the pilot should be advocates (or at least be open-minded).
Acting: Following the Plans

Form appropriate working groups to implement the plans. Perform the activities in the plans. Track the progress against the plans. Take corrective action as necessary. Change the plans as necessary. Manage risks associated with the plan.

See any number of guides to project management

- Program Management Institute Body of Knowledge*
- CMMI Project Management and Control process area.

Learning: Tuning and Improvement - 1

Consolidate data and lessons learned.

Measure results against established goals.

Modify products, processes, and organizational structures to reflect lessons learned and to take advantage of potential optimizations.
Tutorial Outline

About Software Product Line Adoption
Phased Product Line Adoption: a Roadmap
Phased Technology Adoption
**Using the Adoption Roadmap**
Example Product Line Plans
Connections to Other Improvement Initiatives
Conclusion
Adoption Factory and Change Models

The *Adoption Factory* pattern is a generic roadmap for product line adoption. It lays out the technology change that needs to occur in moving to a software product line approach.

- Adoption Factory lacks change management mechanisms and guidance.

A change model is useful for generic guidance about organizational change.

A change model and the *Adoption Factory* pattern can be coupled in a complementary way to guide product line adoption.

In particular, the IDEAL Model is a general model for guiding change.

- IDEAL lacks specific information about the change taking place.
- In particular, IDEAL lacks any product line-specific guidance.

To be used successfully both need to be informed by relevant organization-specific information.
Using IDEAL and Adoption Factory

The IDEAL model lays out a phased approach for the change; that is, the product line adoption or any part of that adoption process.

The Adoption Factory pattern chunks and orders the changes to occur in the actual product line adoption.
Adoption Factory and IDEAL Phases - 1

**Initiating:**
You can use the *Adoption Factory* pattern as an easily understood adoption vocabulary that can be shared across an organization and marks organizational progress. You can use the completion of phases or focus areas as product line adoption goals. You can use the associated roles to guide staffing and management.
Diagnosing:
You can use the *Adoption Factory* pattern to gauge where in the move to product lines your organization is and benchmark your activities by measuring yourself against the practice areas in that phase of Adoption Factory.
Establishing:
You can use the incremental nature of the Adoption Factory pattern to structure a Product Line Adoption Plan.
You can use the subpatterns and their associated practice areas as the basis of subservient action plans.
Adoption Factory and IDEAL Phases - 4

Acting:
You would follow the plans that are based on the Adoption Factory pattern.
You would apply the practice areas in the “Organization” focus area to steer and manage the activities.
Adoption Factory and IDEAL Phases - 5

Learning:

You can

- collect data and lessons learned in each phase of the Adoption Factory pattern as specified by the “Data Collection, Metrics, and Tracking” practice area
- analyze results against established goals
- iterate through the pattern phases and focus on different practice areas, modify products, processes, and organizational structures to reflect lessons learned and to take advantage of potential optimizations
Tutorial Outline

About Software Product Line Adoption
Phased Product Line Adoption: a Roadmap
Phased Technology Adoption
Using the Adoption Roadmap
Example Product Line Plans
Connections to Other Improvement Initiatives
Conclusion
## Product Line Adoption Plans

The Adoption Factory is not a product line adoption plan but it supports the development of product line adoption plans.

<table>
<thead>
<tr>
<th>Type of plan</th>
<th>Plan characteristics</th>
<th>Connection to Adoption Factory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Plan</td>
<td>• lays out overall company strategies to achieve business goals</td>
<td>• It’s a prerequisite for using the Adoption Factory pattern.</td>
</tr>
<tr>
<td></td>
<td>• might specify adopting a software product line for a particular vertical segment of business</td>
<td>• Its goals wills serve as inputs to the product line business case.</td>
</tr>
<tr>
<td>Product Line</td>
<td>• describes how product line practices will be rolled out across the organization</td>
<td>• The pattern is used as an overall plan structure.</td>
</tr>
<tr>
<td>Adoption Plan</td>
<td></td>
<td>• Phases and focus areas become natural milestones.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The pattern is customized to fit organization-specific contexts, strengths, needs and challenges.</td>
</tr>
<tr>
<td>Product Line</td>
<td>• addresses a specific portion of a product line adoption plan</td>
<td>• It maps to a particular phase, focus area, subpattern, or practice area in the pattern.</td>
</tr>
<tr>
<td>Action Plan</td>
<td></td>
<td>• Practice Areas, Roles and Outputs views provide details for it.</td>
</tr>
</tbody>
</table>
Plans, Adoption Factory, and IDEAL - 2

Action Plan Scope
Tutorial Outline

About Software Product Line Adoption
Phased Technology Adoption
Phased Product Line Adoption: a Roadmap
Using the Adoption Roadmap
Example Product Line Plans

Connections to Other Improvement Initiatives
  • Capability Maturity Model Integration (CMMI)
  • Improvement Infrastructure
  • Architecture-centric Development

Conclusion
Linking Product Line Adoption to Other Improvement Initiatives

Improvement initiatives are all about change.

Successful change has at least two important dimensions.

- the technology change itself

- the “people” and organizational aspects of change
  - The people and organizational aspects are often handled by a supporting improvement infrastructure.

You can build on your existing improvement initiative to gain leverage for software product line adoption.

We will examine linkage to two specific improvement initiatives and consider both dimensions of change for

- Capability Maturity Model Integration (CMMI)
- Architecture-centric development
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 is, “How much process discipline?”

Many organizations use CMMI models as a basis for process improvement.
CMMI - Framework Comparisons

See pages 15-16 of
Software Process Improvement and Product Line Practice: CMMI and the Framework for Software Product Line Practice
CMU/SEI-2002-TN-012
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

For Version 1.1, 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)

<table>
<thead>
<tr>
<th>Level</th>
<th>Process Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Optimizing</td>
<td>Organizational Innovation and Deployment Causal Analysis and Resolution</td>
</tr>
<tr>
<td>4 Quantitatively Managed</td>
<td>Organizational Process Performance Quantitative Project Management</td>
</tr>
<tr>
<td>3 Defined</td>
<td>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</td>
</tr>
<tr>
<td>2 Managed</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>
</tr>
</tbody>
</table>
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 centric on business needs or “where it hurts most.”
## 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>PLQL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLTP</td>
</tr>
</tbody>
</table>
## CMMI - Framework Comparisons - 2

<table>
<thead>
<tr>
<th>Area of Comparison</th>
<th>CMMI</th>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains “How To”</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>
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><strong>CMMI Process Areas</strong></th>
<th><strong>Framework Practice Areas</strong></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
### 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
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

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).

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

About Software Product Line Adoption
Phased Technology Adoption
Phased Product Line Adoption: a Roadmap
Using the Adoption Roadmap
Example Product Line Plans

Connections to Other Improvement Initiatives
• Capability Maturity Model Integration (CMMI)
• Improvement Infrastructure
• Architecture-centric Development

Conclusion
Process Improvement Infrastructure

A typical process improvement infrastructure includes
• organizational elements for oversight and implementation of the improvement effort
• generic process assets
• training infrastructure
• other change management assets
• … 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.
Oversight and Implementation - 1

Typical organizational elements to oversee and implement process improvement

- Management Steering Group
  - a group to oversee the direction and progress of the organization’s process improvement effort (directs the process group)

- Process Group
  - a group to facilitate the definition, maintenance, and improvement of the organization’s processes

- Process Action Team
  - a team chartered to develop and implement specific process improvement activities in accordance with an overall process improvement plan
Oversight and Implementation - 2

Leveraging the process Management Steering Group (MSG)
- Form a Product Line Management Steering Group
- Imitate appropriate structures, roles and procedures
  - set direction and arbitrate conflicting needs
  - support and guide the product line manager and staff
  - provide general support, sponsorship and advocacy
  - coordinate closely with process MSG

Leveraging the Process Group and Process Action teams
- Augment the group/team with product line expertise to facilitate development of processes that support software product line needs.
Generic Process Assets

Such assets are often contained in a process asset library
• a library of information used to make available process assets that may be useful for defining, implementing, and managing processes in the organization
• example contents
  - policies
  - process descriptions
  - procedures
  - plans (e.g., development, quality assurance, testing, piloting, roll-out)
  - process aids (e.g., standards, checklists, templates)
  - lessons-learned reports

These assets can be a basis for product line-specific needs.
Training Infrastructure

This is a special case of CMMI process leverage and improvement infrastructure.

Training is an integral part of any technology change and is crucial for institutionalizing the change.

An organization that has implemented the CMMI Process Area of *Organizational Training* has an excellent infrastructure to support SPL adoption, including

- processes to determine training needs
- processes to determine level of responsibility for training
- processes to plan and deliver training
- and often a training organization to support all this

This capability can be applied to product line-specific needs.
Other Change Management Assets

Successful process improvement change involves development of change management skills and tools, often in the process group, that don’t necessarily have a process focus. Such assets are useful for software product line adoption.

Examples:
• resistance management
  - ability to analyze change resistance within an organization and ability to plan and execute strategies to overcome resistance
• sponsorship and advocacy development and nurturing
  - building sponsors and champions throughout
• communications strategies
  - up and down the chain
• team creation and performance building
Tutorial Outline

About Software Product Line Adoption
Phased Product Line Adoption: a Roadmap
Phased Technology Adoption
Using the Adoption Roadmap
Example Product Line Plans

Connections to Other Improvement Initiatives
  • Capability Maturity Model Integration (CMMI)
  • Improvement Infrastructure
  • Architecture-centric Development

Conclusion
The software architecture of a software system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.\(^1\)

Architecture is
- the blueprint for a project
- the carrier of most system quality attributes
- a forum for resource tradeoffs
- a contract that allows multi-party development
- an essential part of complex systems

Defining an architecture carries the additional obligations of

• communicating (documenting) it
• evaluating it for fitness of purpose
• assuring conformance to it
Architecture-Centric Development Activities

Architecture-specific activities include the following:
• creating the business case for the system
• understanding the requirements
• creating and/or selecting the architecture
• documenting and communicating the architecture
• analyzing or evaluating the architecture
• implementing the system based on the architecture
• ensuring that the implementation conforms to the architecture

All these activities require a disciplined approach to software development that provides a basis for software product line adoption.
Linkages with the Framework

Direct linkages to the following software engineering practice areas in the Framework include:

• Building a Business Case
• Requirements Engineering
• Architecture Development
• Architecture Evaluation
• Component Development
• Testing

There are also weaker linkages with

• Mining Existing Assets
• COTS Utilization
• Software System Integration
### Influence on 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

- Data Collection, Metrics and Tracking
- Technical Risk Management
- Organizational Risk Management
- Customer Interface Management
- Organizational Planning

*Strong support
*Weak support
Architecture Activities and Product Lines

Of all a product line’s core assets, the product line architecture may well be the most important one for ensuring technical success.

If an organization already uses disciplined practices to develop their single-system software under the aegis of a software architecture, it is well poised to

• define a product line architecture
• follow its dictates in implementing the other core assets and products from those core assets.

As with building on CMMI process improvement, the single-system architecture-centric practices must be adapted to account for product line-unique aspects.
Other Linkages

An organization that has disciplined architecture-centric practices may likely have the following infrastructure that can also be exploited during product line adoption:

- an architecture steering group
- an architecture center of excellence
- architecture documentation standards
- architecture-specific tool support
- architecture training
Tutorial Outline
About Software Product Line Adoption
Phased Technology Adoption
Phased Product Line Adoption
Using the Adoption Roadmap
Example Product Line Plans
Connecting to Other Improvement Initiatives
Conclusion
Product Line Adoption

*Product line adoption* involves moving from some form of developing software-intensive systems with a single-system mentality to developing them as a software product line.
Successful Adoption

The benefits to be accrued by software product lines are proven. The barriers and risks associated with product line adoption are nontrivial.

The barriers can be overcome and the risks mitigated with careful preparation, planning, and execution.

There are two categories of information that must inform product line adoption and a Product Line Adoption Plan:

1. generic guidance
   • for product lines
   • for technology change
2. organizational context
Factors Influencing Adoption

Organizational Context

- product line readiness
- barriers
- enablers
- unique characteristics
- culture
- other ongoing activities
Factors Influencing Adoption

Organizational Context
- product line readiness
- barriers
- enablers
- unique characteristics
- culture
- other ongoing activities

Adoption Support
- The Framework
- product line approaches
- product line adoption roadmap
- change management mechanisms
- change models
- planning process

Product Line Adoption Plan

Product Line Action Plans
We have examined some inputs to the “kettle,” the outputs, and the processes involved.

When planning your product line adoption use the generic guides we have provided and temper them with your own organizational characteristics.

Use a change model and mechanisms that fit your culture and context.

A software product line approach is reuse that pays.

Software product line adoption is worth it and now you have the tools.
Contact Information

Linda Northrop
Director
Product Line Systems Program
Telephone: 412-268-7638
Email: lmn@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

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

Business Development
Product Line Systems Program
Jay Douglass
Telephone: 412-268-6834
Email: jcd@sei.cmu.edu