Software Engineering Institute | Carnegie Mellon

Adopting Software Product Lines

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Module 1: Course Introduction

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM20-0359

Introductions

Instructor Introduction

Participant Introductions

- name
- company/position
- background
- software product line activities and interests

Course Objectives

Adopting Software Product Lines is a two-day course designed to

- provide a practical introduction to software product line adoption
- introduce participants to a pattern-based product line adoption roadmap
- introduce participants to a phased approach for handling product line adoption as a technology change
- provide guidelines and artifacts for product line adoption planning
- explore linkage of software product line adoption with other on-going activities

Course Entry Criteria

This course is designed for

- software engineers and technical managers interested in moving to a software product line approach
- anyone who is responsible for creating and/or managing a software product line adoption effort

Participants must have

- experience in designing and developing software-intensive systems
- familiarity with modern software engineering concepts
- completed the Software Product Lines course

Course Outcomes

After the Adopting Software Product Lines course, participants should

- understand the issues surrounding software product line adoption

and be able to

- apply a phased, pattern-based approach to handling product line adoption
- overcome technical and organizational adoption barriers using practical methods
- successfully plan for product line adoption
- build and execute product line adoption plans tailored to a specific organization
- link product line adoption with other ongoing improvement initiatives and software engineering approaches

Course Structure

Course Introduction

Adoption Fundamentals

Using Adoption Models

Adoption Planning

Exploiting Other Current Activities

Wrap-Up

Course Strategy

Lectures and discussion will be used to introduce concepts.

Exercises will be used to reinforce concepts and develop skills.

Course Materials

Course description

Course outline

Printed copy of course slides

Technical reports and technical notes

- *Software Product Line Adoption Roadmap*
- *Software Process Improvement and Product Line Practice: CMMI and the Framework for Software Product Line Practice*
- *Software Process Improvement and Product Line Practice: Building on Your Process Improvement Infrastructure*
- *Product Line Adoption in a CMMI Environment*

Supplemental materials

- exercise handouts

Course Agenda - 1

Day One:

Course Introduction

Part 1: Adoption Fundamentals

- Software Product Line Adoption: What and Why?
- Adoption Factory Pattern

Part 2: Using Adoption Models

- Change Models and Mechanisms
- The IDEAL Model

Course Agenda - 2

Day Two:

Part 2: Using Adoption Models (cont.)

- Using Adoption Factory with IDEAL

Part 3: Adoption Planning

- Plans and Planning
- Planning Example

Part 4: Exploiting Other Current Activities

- CMMI and Process Improvement
- Architecture-Centric Development and Hardware Platform Engineering

Wrap-Up

Rules of Engagement

We will be very busy over the next two days. To complete everything and get the most from the course, we will need to follow some rules of engagement:

- Your participation is essential.
- Feel free to ask questions at any time.
- Discussion is good, but we might need to cut some discussions short in the interest of time.
- Please try to limit side discussions during the lectures.
- Please turn off cell phones, laptops, Blackberries, etc.
- Let's try to start on time.

Course Completion Criteria

To receive a certificate for completion of this course participants must

- actively participate in classroom discussions and exercises during both days
- not miss any classroom time

Expectation Setting

What do you want to get out of this course?

Do your expectations match course content?

Logistics

Breakfast, lunch, and breaks

Restrooms

Security

Smoking

Questions So Far?

Software Engineering Institute | Carnegie Mellon

Adopting Software Product Lines

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Part 1: Adoption Fundamentals
Module 2: Software Product Line Adoption: What and Why?

Session Outcomes

After this session participants should

- be reacquainted with software product line fundamentals
- understand the concepts underlying software product line adoption
- know the costs, benefits, barriers, and risks to product line adoption
- know what successful software product line adoption “looks like”
- appreciate what is involved in product line adoption planning

Session Topics

Software product line adoption

Benefits, barriers, risks

Existing support

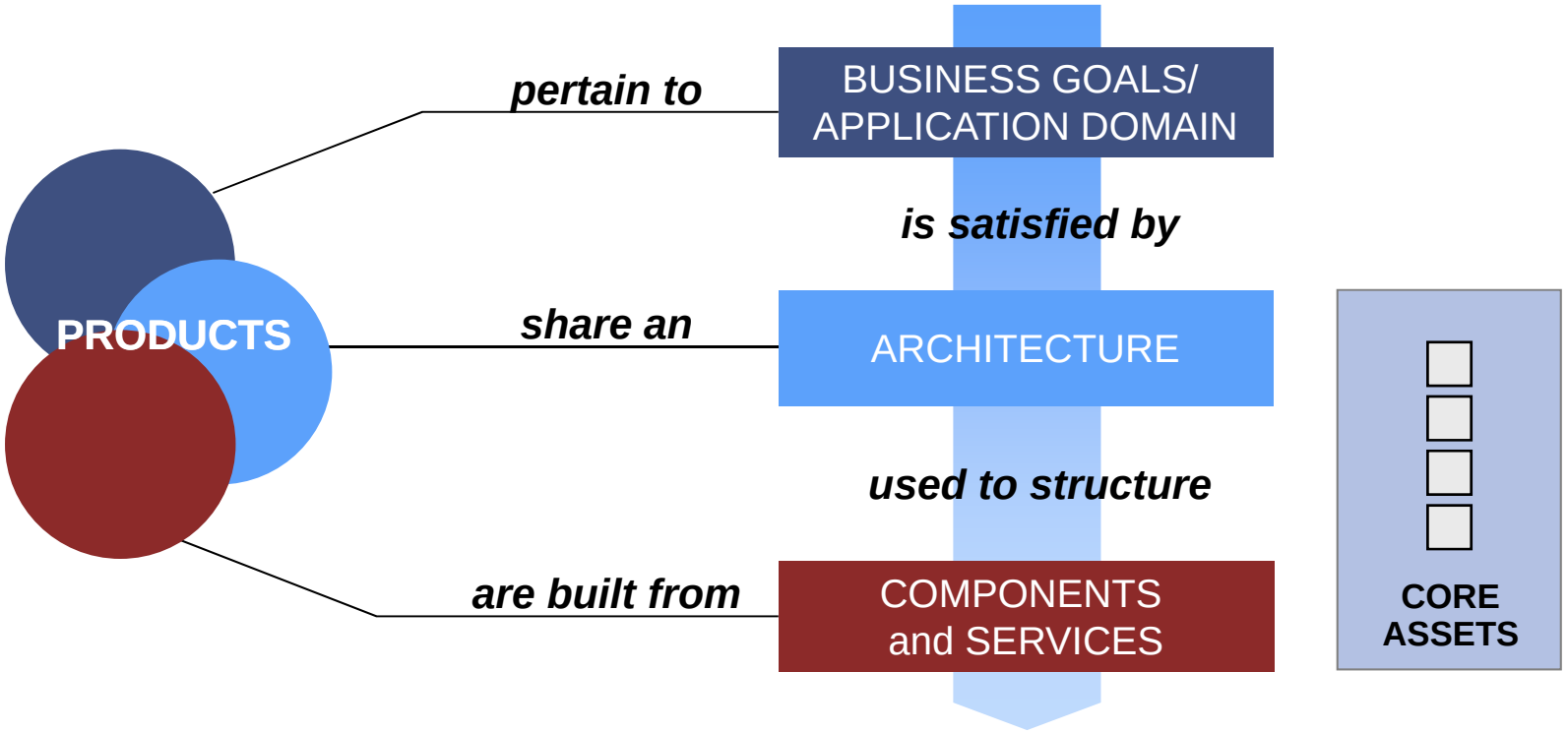
Further needs

Summary

Software Product Lines - 1

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.**

Software Product Lines

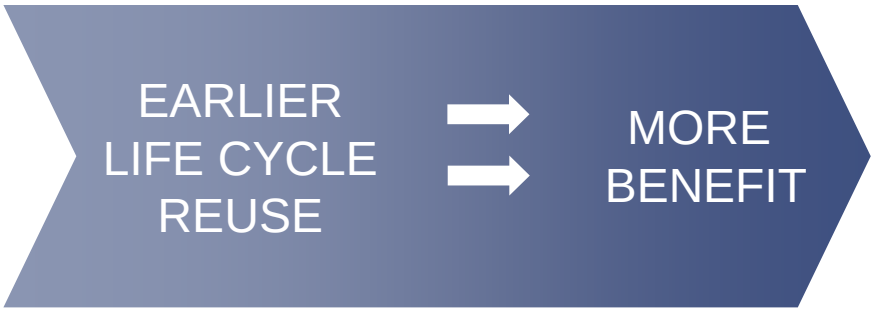


Product lines
- take economic advantage of commonality
- bound variation

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 and services



PRODUCT LINES = STRATEGIC REUSE

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.

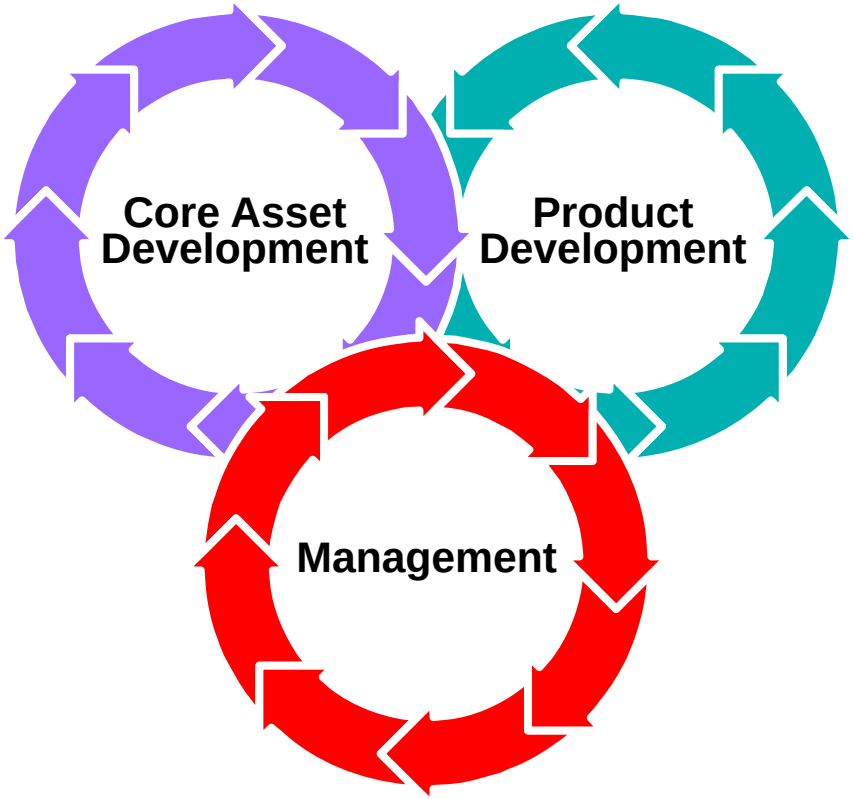
The Adoption Endgame

To have an **operational software product line.**

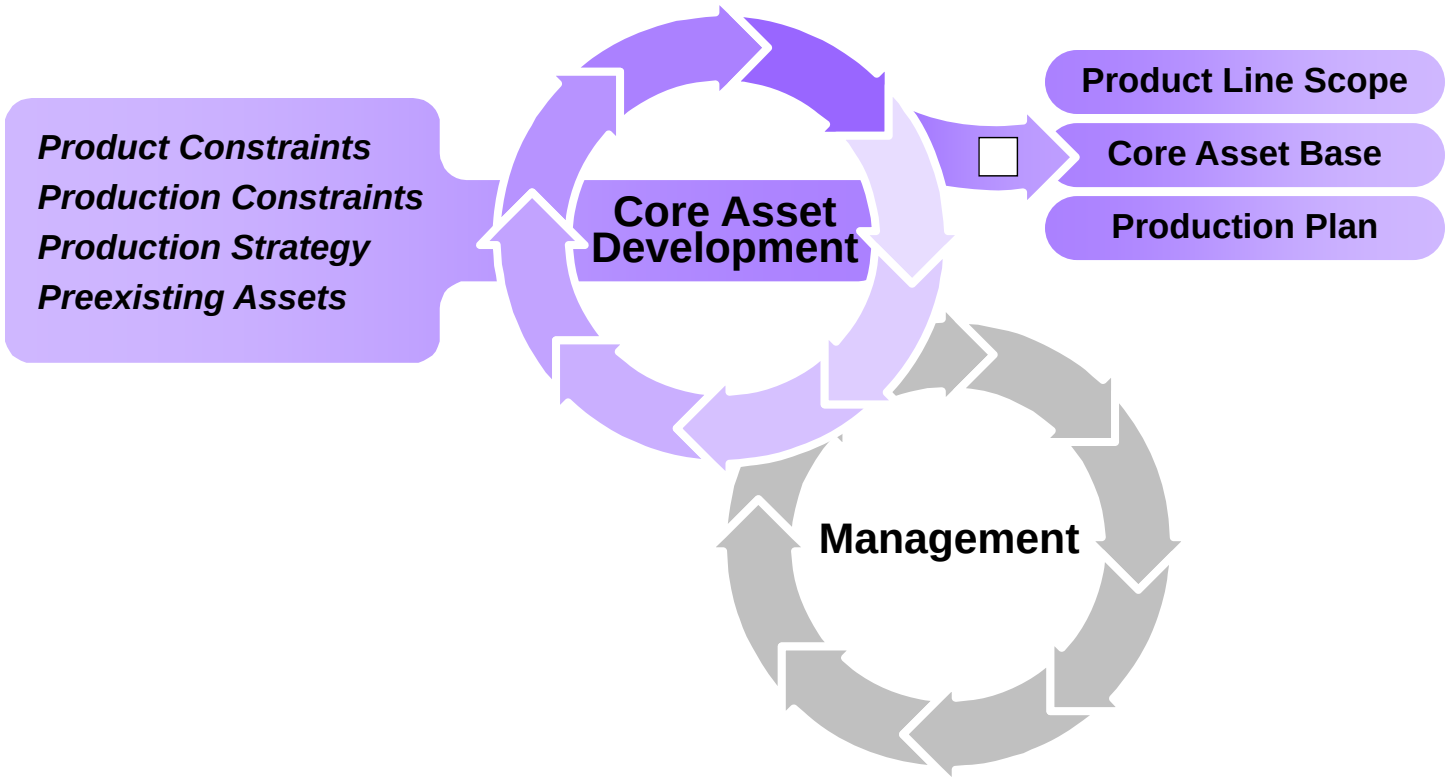
To do that, an organization must

- have
  - a core asset base
  - supportive processes and organizational structures
- develop products from that asset base in a way that achieves business goals
- prepare the organization to institutionalize product line practices

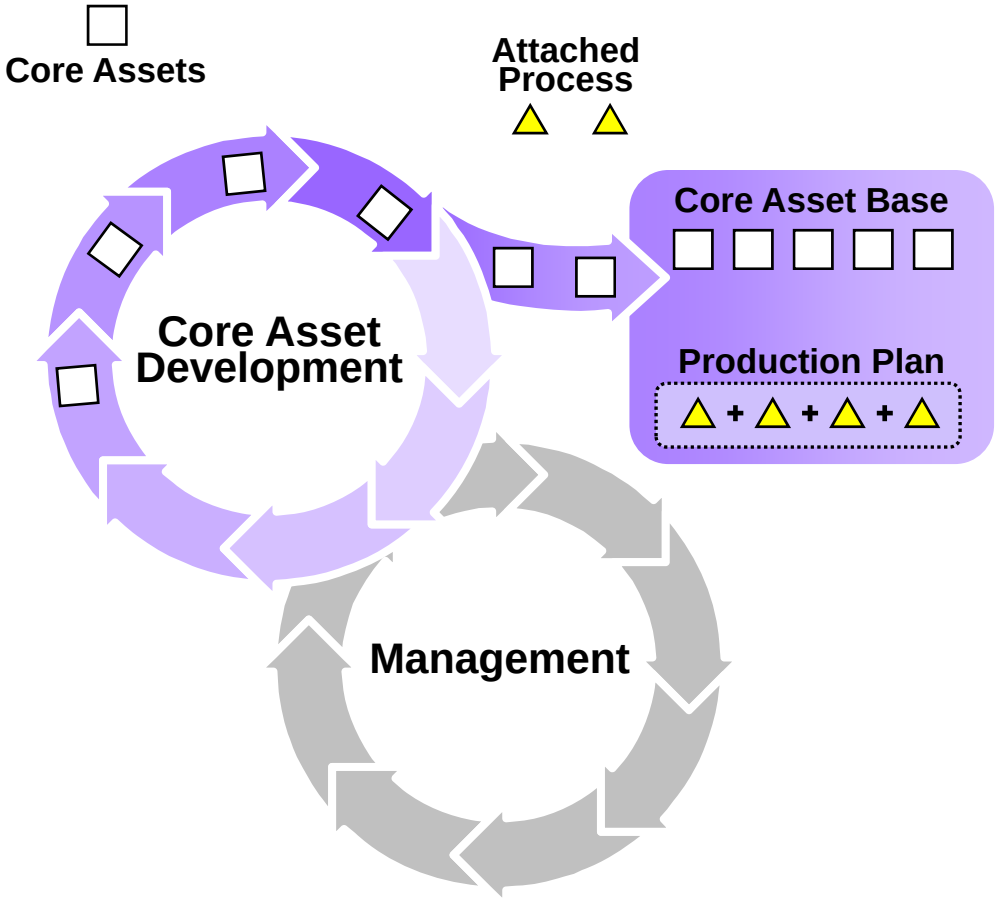
The Three Essential Activities



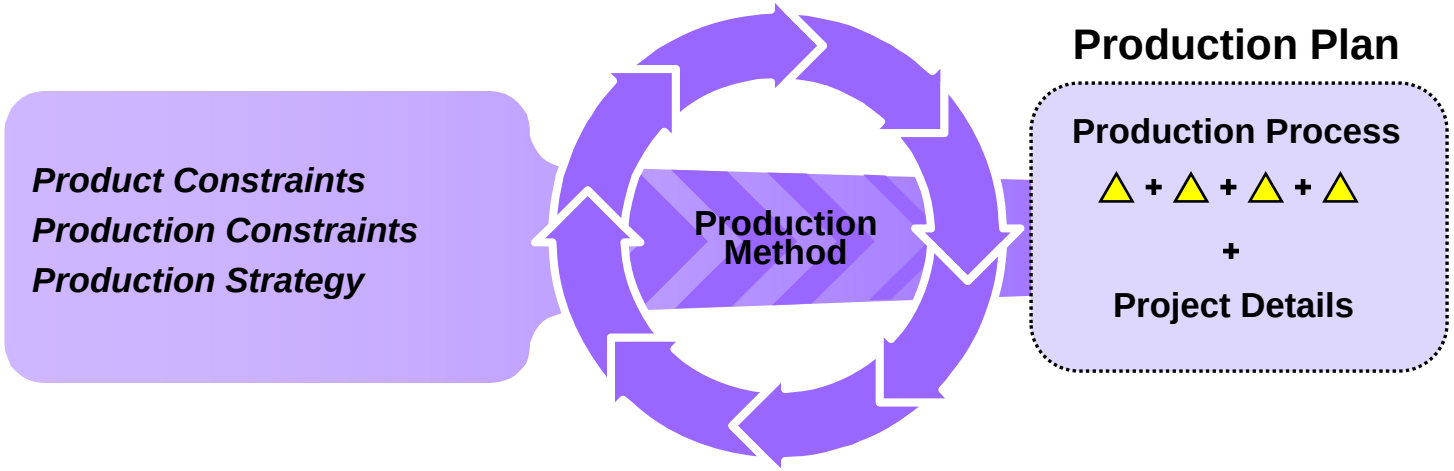
Core Asset Development



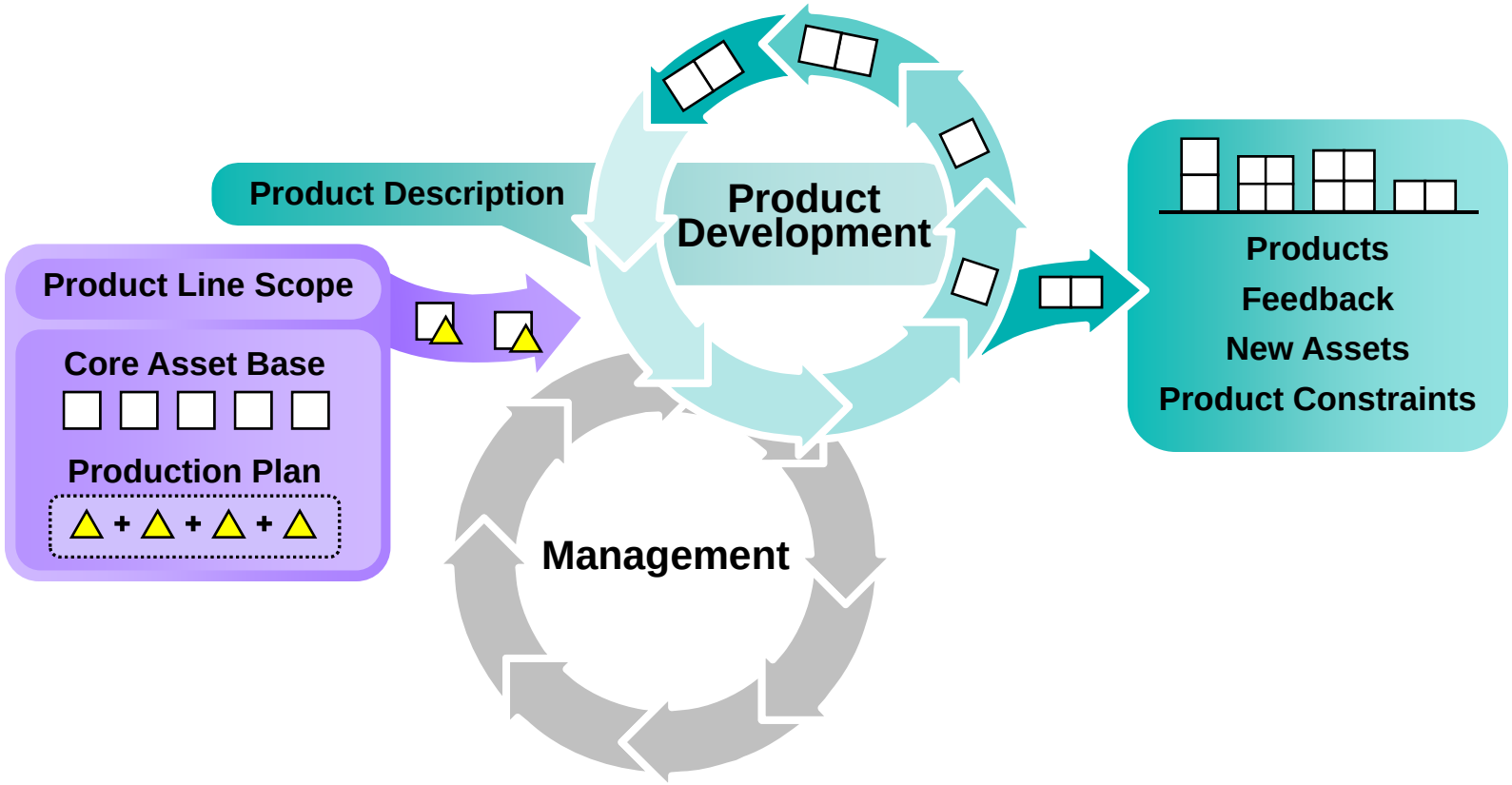
Attached Processes



Product Line Production Plan



Product Development



Different Approaches - 1

Proactive: Develop the core assets first.

- Develop the scope first and use it as a “mission” statement.
- Products come to market quickly with minimum code writing.
- Requires upfront investment and predictive knowledge

Reactive: Start with one or more products.

- From them, generate the product line core assets and then future products; the scope evolves more dramatically.
- Much lower cost of entry
- The architecture and other core assets must be robust, extensible, and appropriate to future product line needs.

Different Approaches - 2

Incremental: In either a reactive or proactive approach, it is possible to develop the core asset base in stages, while planning from the beginning to develop a product line.

- Develop part of the core asset base, including the architecture and some of the components.
- Develop one or more products.
- Develop part of the rest of the core asset base.
- Develop more products.
- Evolve more of the core asset base.
- …

Session Topics

Software product line adoption

Benefits, barriers, risks

Existing support

Further needs

Summary

Real World Motivation

Organizations use product line practices to:

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

Value Proposition for Software Product Lines

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

- increased quality—by as much as 10x
- decreased cost—by as much as 60%
- decreased labor needs—by as much as 87%
- decreased time to market (to field, to launch...)—by as much as 98%
- ability to move into new markets—in months, not years

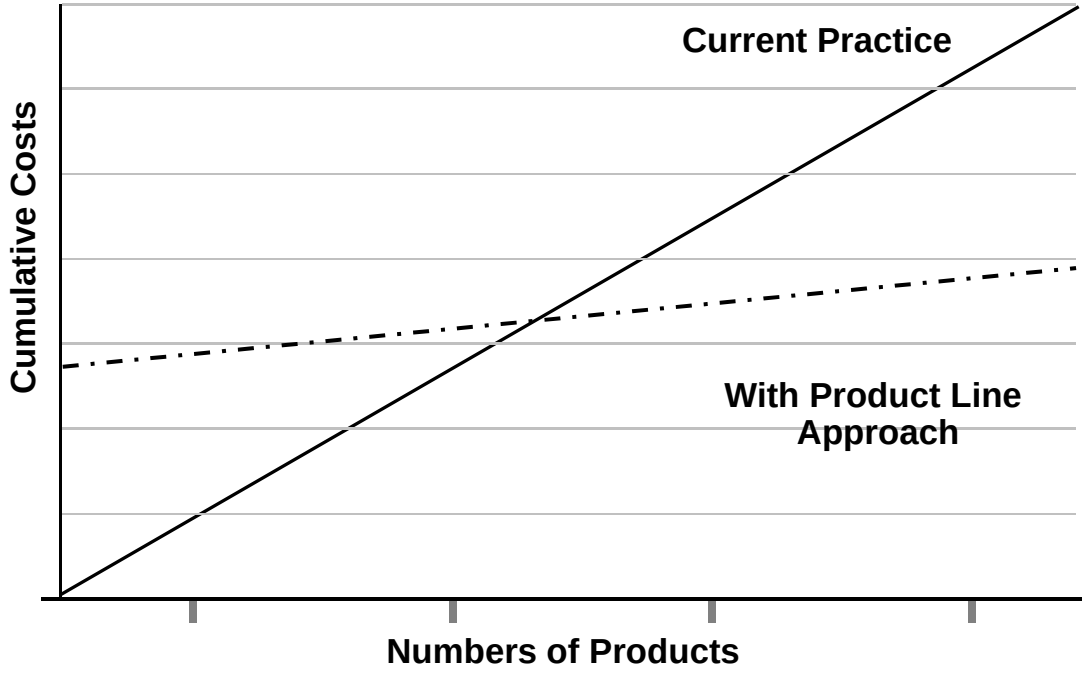
Barriers To Product Line Adoption

Costs Of A Software Product Line

Core Assets	Costs
Architecture	Must support variation inherent in the product line
Software Components	Must be designed to be general without a loss of performance; must build in support for variation points
Test Plans, Test Cases, Test Data	Must consider variation points and multiple instances of the product line
Business Case and Market Analysis	Must address a family of software products, not just one product
Project Plans	Must be generic or be made extensible to accommodate product variations
Tools and Processes	Must be more robust
People, Skills, Training	Must involve training and expertise centered around the assets and procedures associated with the product line

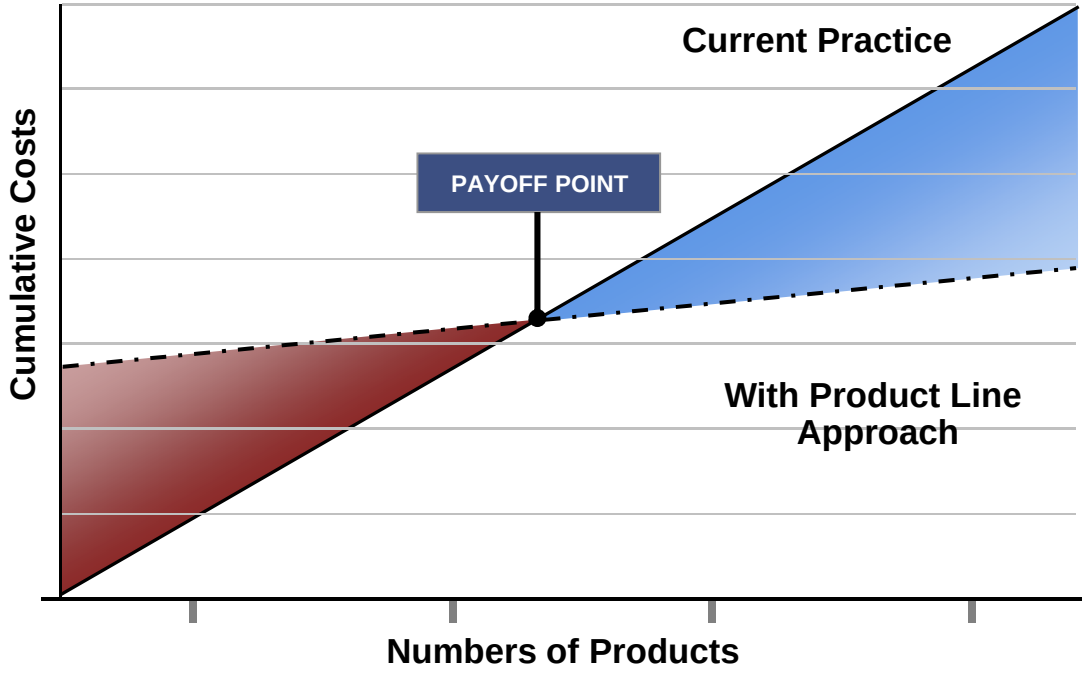
In addition, the costs associated with organizational change.

Economics Of Product Lines



Weiss. D.M. & and Lai, C.T.R..
Software Product-Line Engineering: A Family-Based Software Development Process
Reading, MA: Addison-Wesley, 1999.

Economics Of Product Lines



Weiss. D.M. & and Lai, C.T.R..
Software Product-Line Engineering: A Family-Based Software Development Process
Reading, MA: Addison-Wesley, 1999.

Barriers To Product Line Adoption

Time Needed For Product Line Adoption

Time is needed to

- launch the product line effort
  - educate
  - address cultural barriers
- define supportive processes and organizational structures
- develop a core asset base
- lead the organization to an operational product line
- continue to do business

An organization can't go out of business while adopting a product line approach.

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

Change management models are useful.

A product line adoption roadmap is helpful.

Big, Obvious Risks

An organization that **cannot overcome its barriers** to product line adoption is unlikely to succeed.

An organization that **does not know what is necessary** to succeed with software product lines is unlikely to succeed. (At least in a cost effective, timely way.)

An organization that **does not know how** to go about product line adoption is unlikely to succeed. (without pain and suffering)

> *"If we do not succeed, we run the risk of failure."*
>
> *(in)famous American VP*

Session Topics

Software product line adoption

Benefits, barriers, risks

Existing support

Further needs

Summary

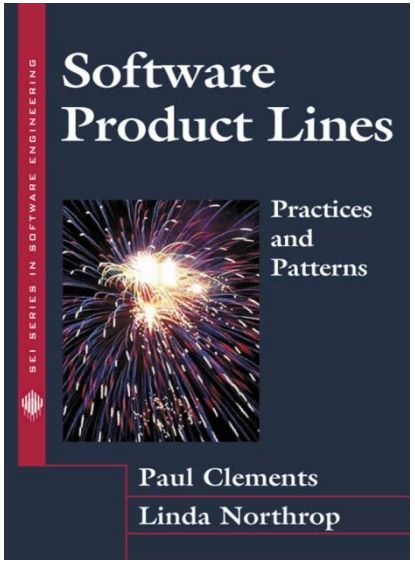
The SEI Framework For Software Product Line Practice[SM]

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 5.0 –
http://www.sei.cmu.edu/productlines/frame_report/index.html

Version 4.0 –
in Software Product Lines: Practices and Patterns



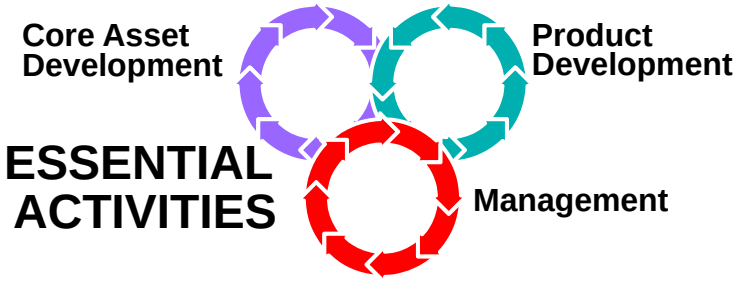
The Goals of the Framework

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.

Framework Version 5.0



PRACTICE AREAS		
Software Engineering	Technical Management	Organizational Management
Architecture Definition	Configuration Management	Building a Business Case
Architecture Evaluation	Make/Buy/Mine/Commission Analysis	Customer Interface Management
Component Development	Measurement and Tracking	Developing an Acquisition Strategy
Mining Existing Assets	Process Discipline	Funding
Requirements Engineering	Scoping	Launching and Institutionalizing
Software System Integration	Technical Planning	Market Analysis
Testing	Technical Risk Management	Operations
Understanding Relevant Domains	Tool Support	Organizational Planning
Using Externally Available Software		Organizational Risk Management
		Structuring the Organization
		Technology Forecasting
		Training

"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.

"Launching and Institutionalizing" Practice Area - 2

All organizations launch and institutionalize change.

Product line adoption is such a change.

- Change experts have models and practices to assist in ensuring successful change.
- These have to be adapted for software product line adoption.
  - You need to launch and institutionalize practices in each of the 29 practice areas.
  - How you go about doing that depends on specific organizational context as well as the change models and practices you use.

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

This Course

The Adopting Software Product Lines course focuses on the "Launching" part of the "Launching and Institutionalizing" practice area.

Successful adoption sets the stage for institutionalization.

Product Line Adoption Plans

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

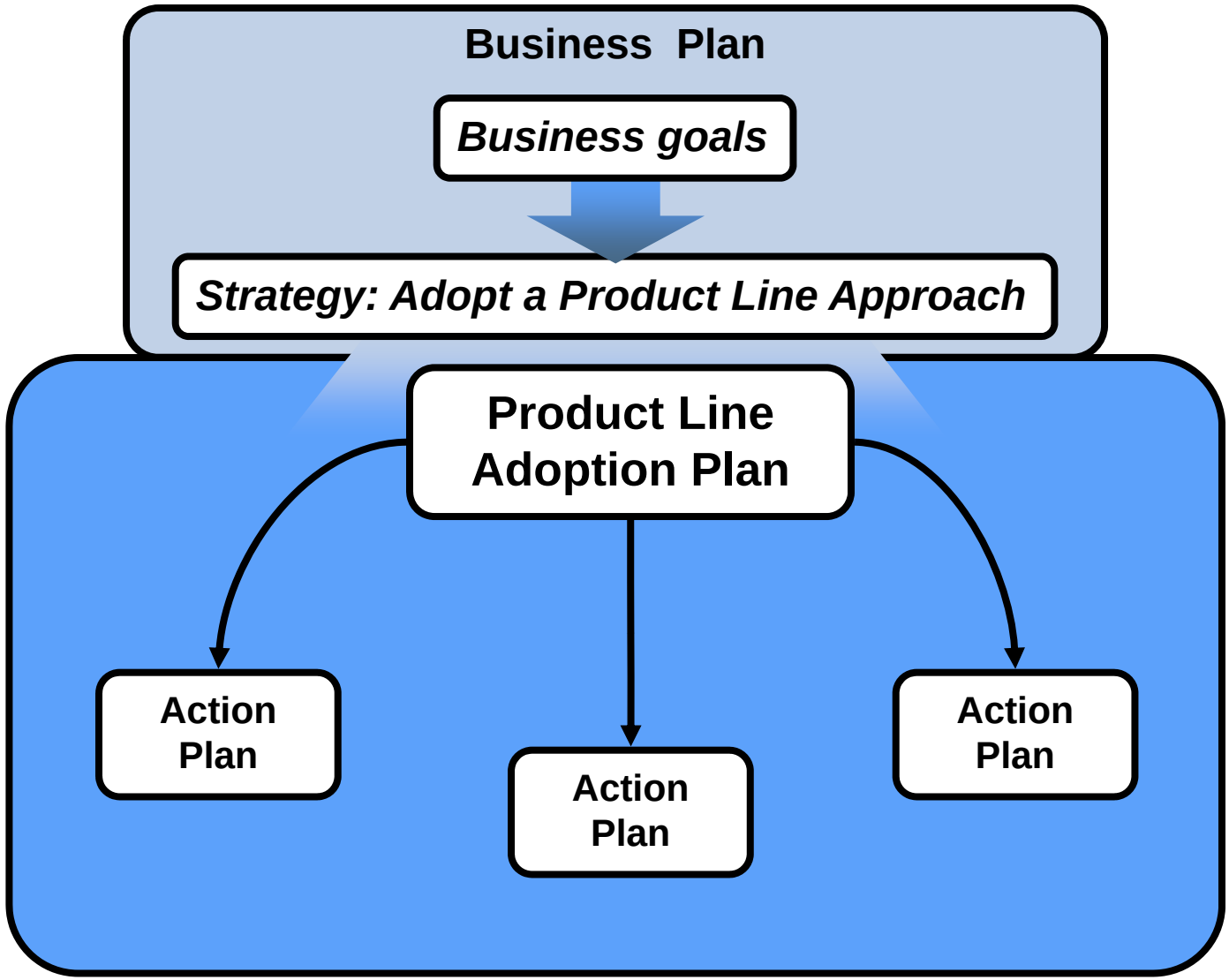
In any organization, there may be a hierarchical set of goals, strategies, and plans.

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 is the purpose 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



Discussion

What are some unique characteristics that distinguish product line adoption from typical technology-focused change?

Session Topics

Software product line adoption

Benefits, barriers, risks

Existing support

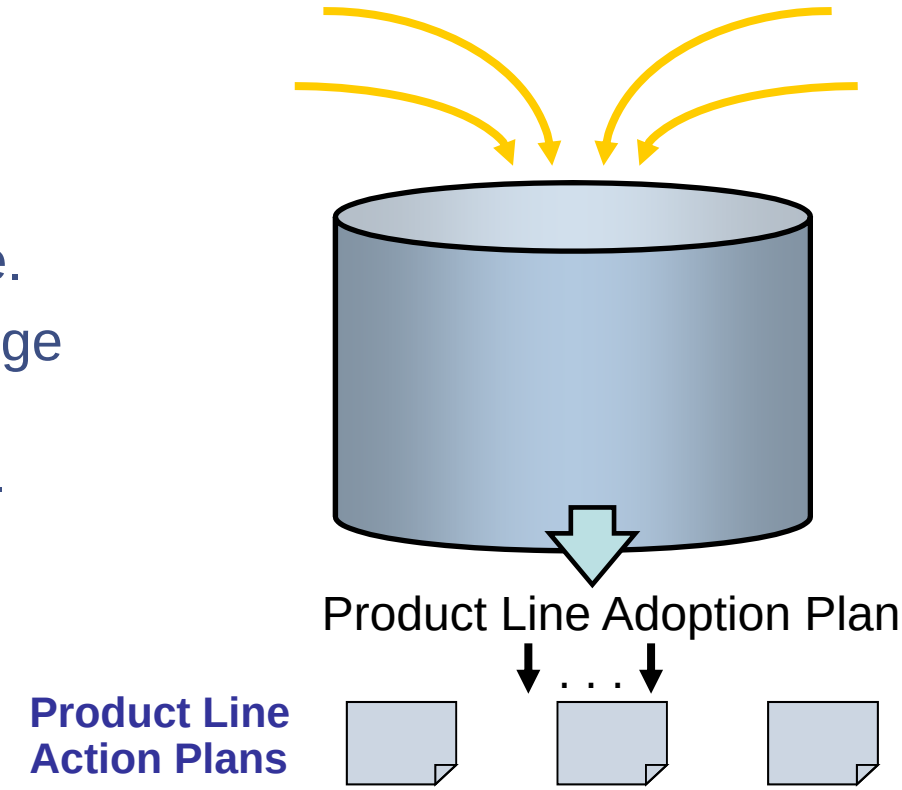
Further needs

Summary

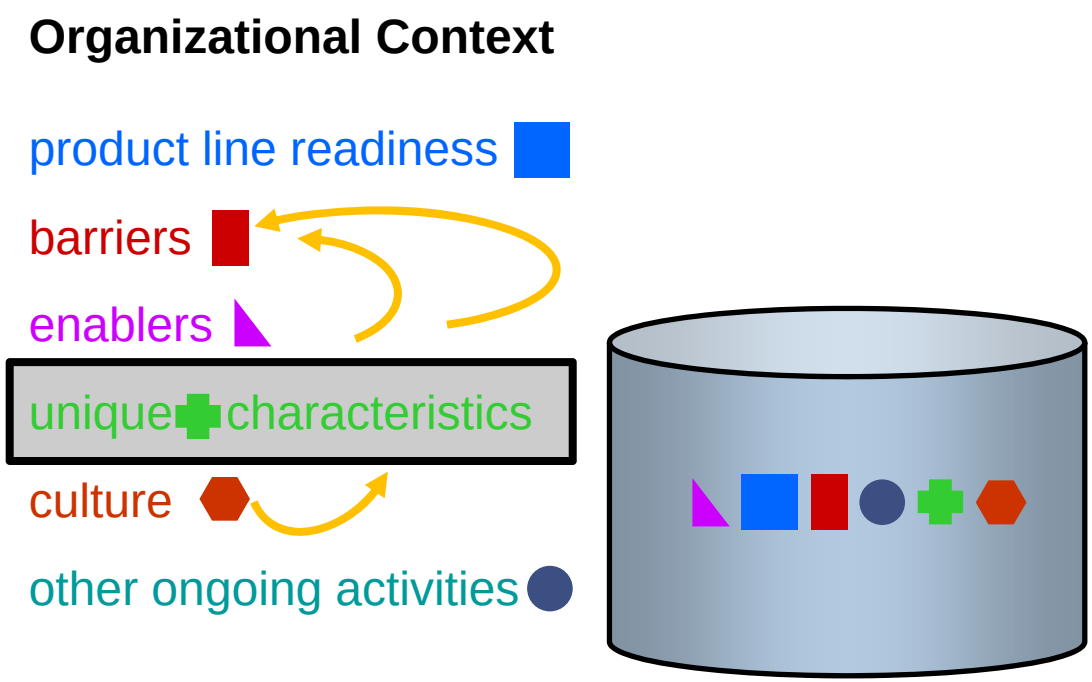
Product Line Adoption Planning

There are many factors that must inform product line adoption and hence a Product Line Adoption Plan.

- Some relate to product line-specific needs and guidance.
- Some relate to generic change models and practices.
- Some relate to organization-specific needs and culture.



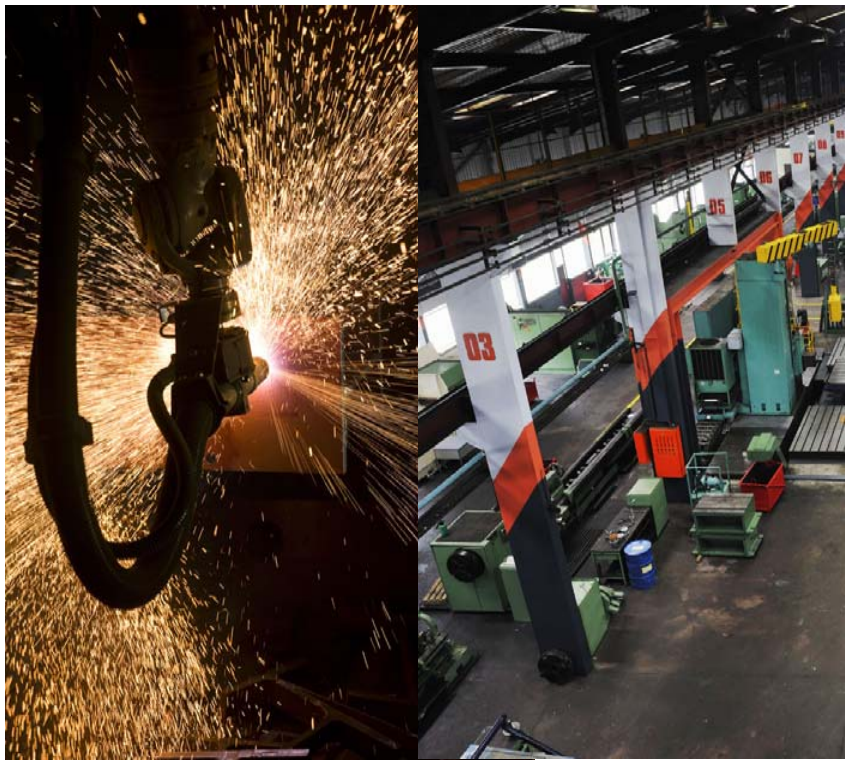
Factors Influencing Adoption



Engineering-Based Organizations

Organizations where

- The products in the product line have both hardware and software components
- The individuals involved in the product line have an engineering background



Typical Engineering-Based Organization

You are here

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

Developer-Focused Organizations

Organizations

- That produce software applications and enterprise systems.
- Have typically not grown out of a hardware or engineering culture, but rather an IT culture.

Typical Engineering-Based Organization

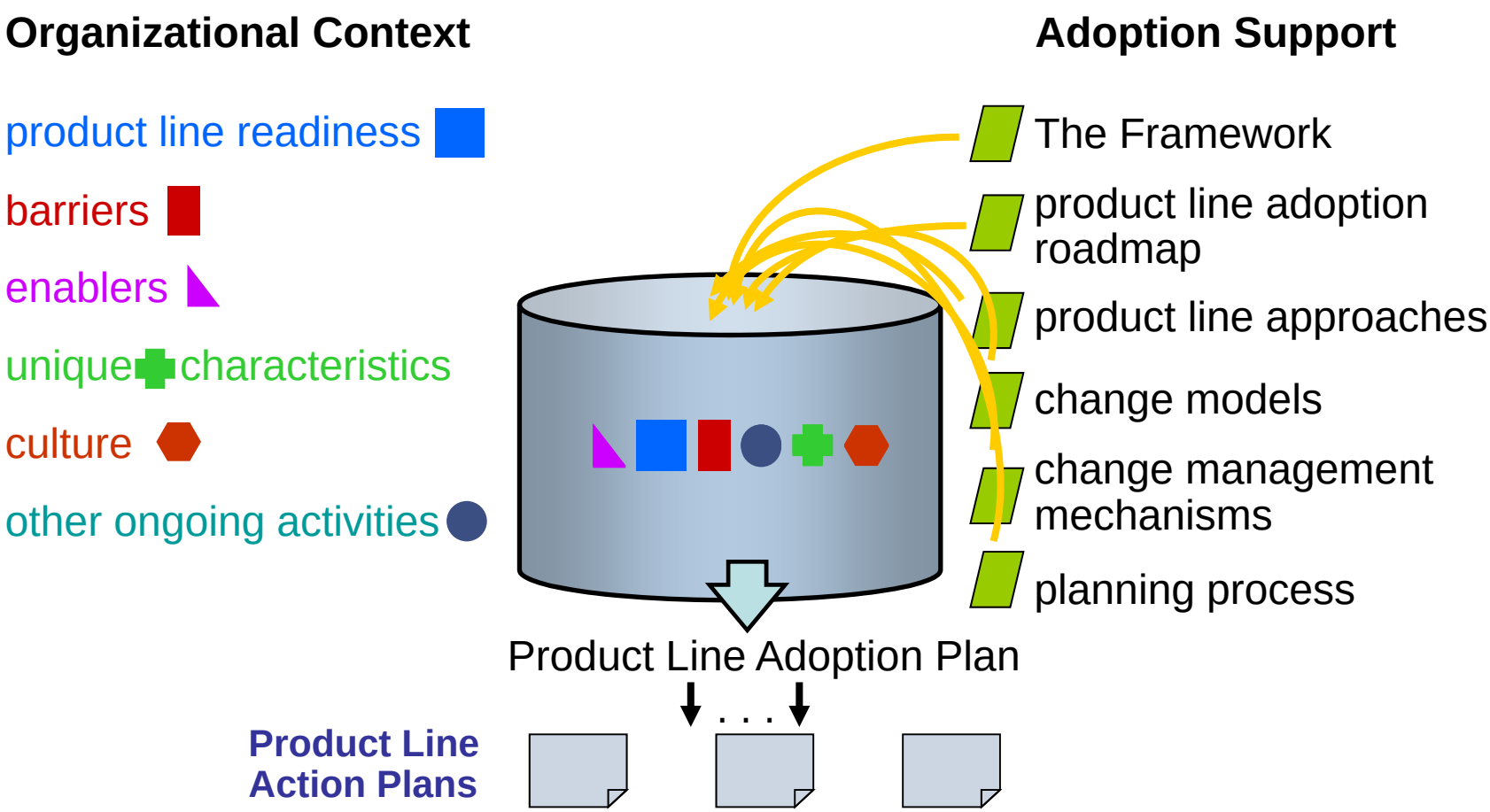
You are here

You are here

You are here

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

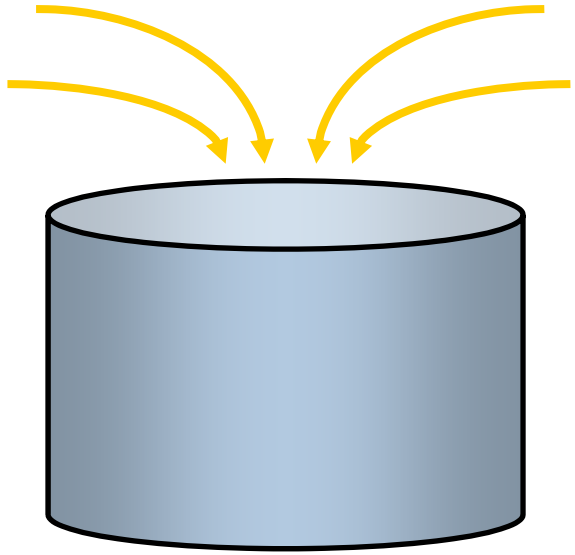
Factors Influencing Adoption



About This Course

In this course we will carefully examine the needed generic guidance related to both product lines and organizational change and the facets of an organization's context.

We will learn what goes in the "kettle," what comes out, and the processes involved.



Session Topics

Software product line adoption

Benefits, barriers, risks

Existing support

Further needs

Summary

Summary

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.

An organization's Product Line Adoption Plan should specify its product line adoption approach.

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

- generic guidance
  - for product lines
  - for organizational change
- organizational context

In this course we provide more specific guidance about just how the "Launching" part of the "Launching and Institutionalizing" practice area is achieved by individual organizations.

Software Engineering Institute | Carnegie Mellon

Adopting Software Product Lines

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Part 1: Adoption Fundamentals
Module 3: Adoption Factory Pattern

Session Outcomes

After this session participants should

- know how the Factory pattern and its constituent subpatterns describe the entire product line effort
- be familiar with the Adoption Factory pattern
- be familiar with six different ways to view the Adoption Factory pattern

Session Topics

Review of Product Line Practice Patterns

The Adoption Factory Pattern

Useful Views of the Adoption Factory Pattern

Exercise

Summary

Patterns

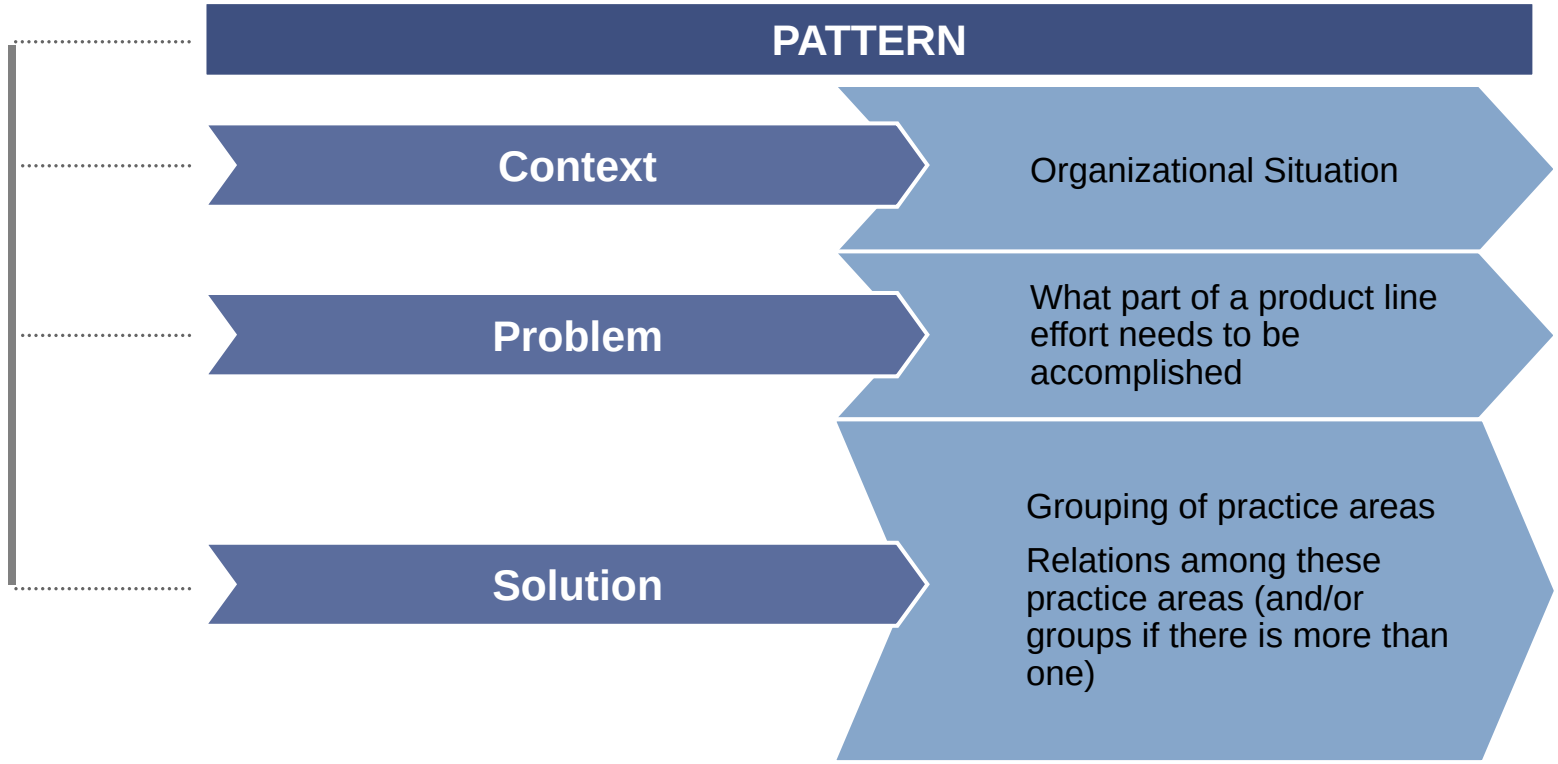
Patterns are a way of expressing common context and problem-solution pairs.

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 Patterns



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

Template for Software Product Line Practice Patterns

Name: a unique and intuitive pattern name and a short summary of the pattern

Example: one or more scenarios to help illustrate the context and the problem

Context: the organizational situations in which the pattern may apply

Problem: what part of a product line effort needs to be accomplished

Solution: the basis for the practice area pattern grouping underlying the pattern

Static: lists the practice areas in each group

Dynamics: a table, diagram(s), or possibly scenario(s) describing the relations among the practice areas in each group and/or among the groups if there is more than one

Application: any suggested guidelines for applying the pattern

Variants: a brief description of known variants or specializations of the pattern

Consequences: the benefits and any known limitations of the pattern

A Word About Pattern Dynamics

Software product line activities are iterative.

- Any relation that shows a dynamic interaction between two practice areas has to be iterative.
- The relations will vary with the pattern.
- Whatever the relation, it is always iterative.
- The arrow shown below denotes the shifting of active emphasis but NOT sequential activity.

Current Set of Patterns

Pattern	Variants
Assembly Line	
Cold Start	Warm Start
Curriculum	
Each Asset	Each Asset Apprentice Evolve Each Asset
Essentials Coverage	
Factory	Adoption Factory
In Motion	
Monitor	
Process	Process Improvement
Product Builder	Product Gen
Product Parts	Green Field Barren Field Plowed Field
What to Build	Analysis Forced March

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.

Factory Pattern - 2

Problem:

To map the entire product line effort

Solution:

Fielding a product line involves

- deciding what and justifying products to include in the product line
- preparing the organization for a product line approach
- designing and providing the core assets that will be used to construct the products in the product line
- building and using the production infrastructure (necessary plans, processes, and tools)
- building products from the core assets in a prescribed way
- monitoring the product line effort, keeping a pulse on the adoption activities and the product line operations, and applying course corrections as necessary to keep the organization on course

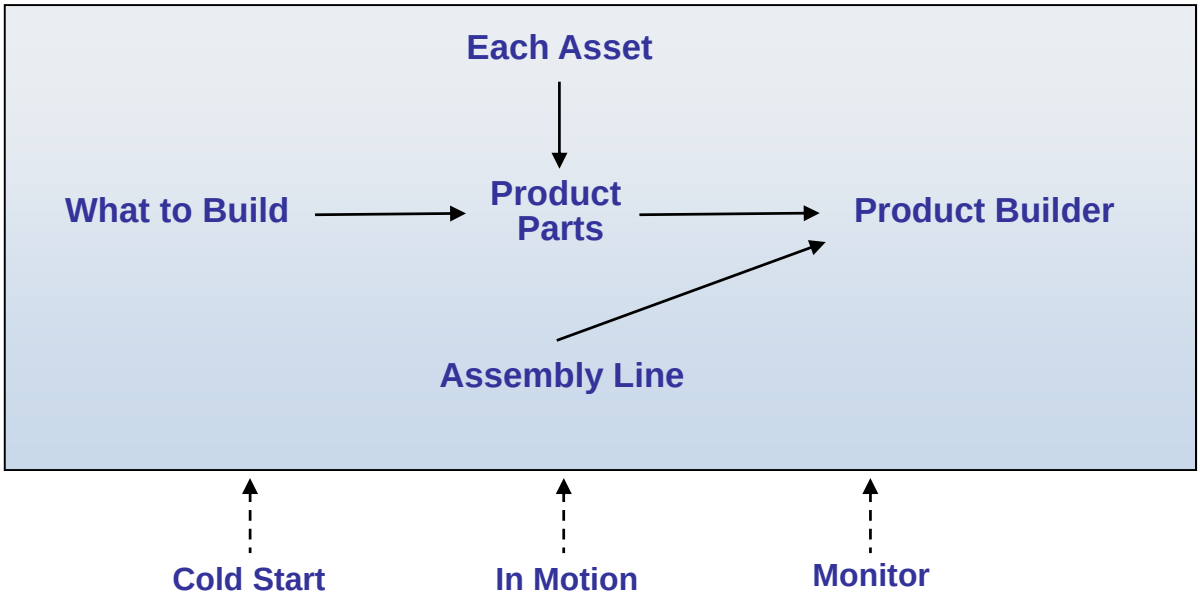
Factory Pattern - 3

Static:

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

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

Factory Pattern - 4



Informs and information flow

Supports

Factory Pattern - 5

Application:

The ***Factory*** pattern offers an abstraction of the entire product line organization – a high-level view and a blueprint for a “divide and conquer” strategy. It can be applied as a roadmap for the entire product line organization.

Factory Pattern - 6

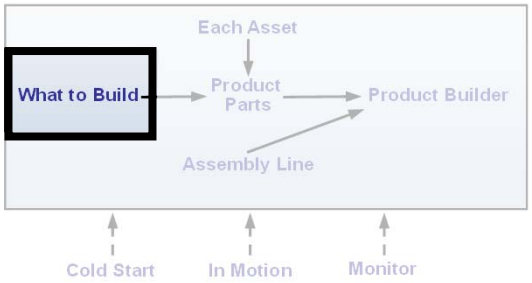
Variants:

Adoption Factory

Consequences:

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

What to Build Pattern - 1



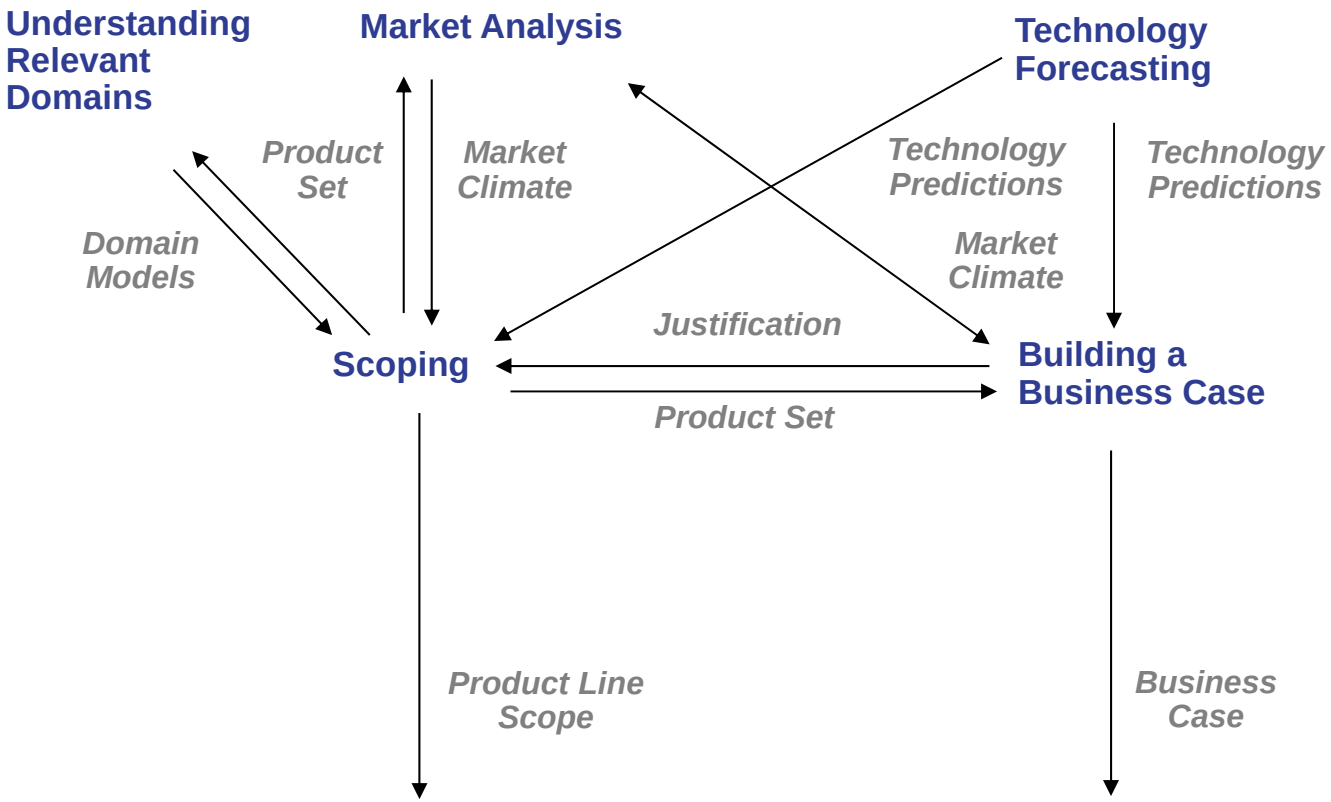
Name:

The ***What to Build*** pattern helps an organization determine what products ought to be in its software product line – what products to build.

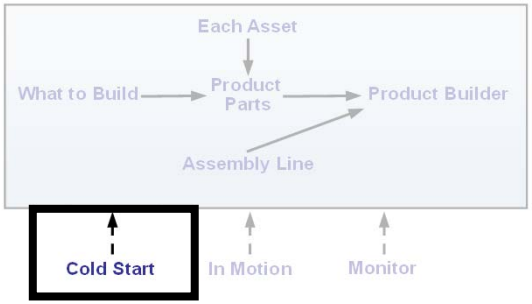
Context:

An organization has decided to field a software product line and knows the general product area for the set of products.

What To Build Pattern - 2



Cold Start Pattern - 1



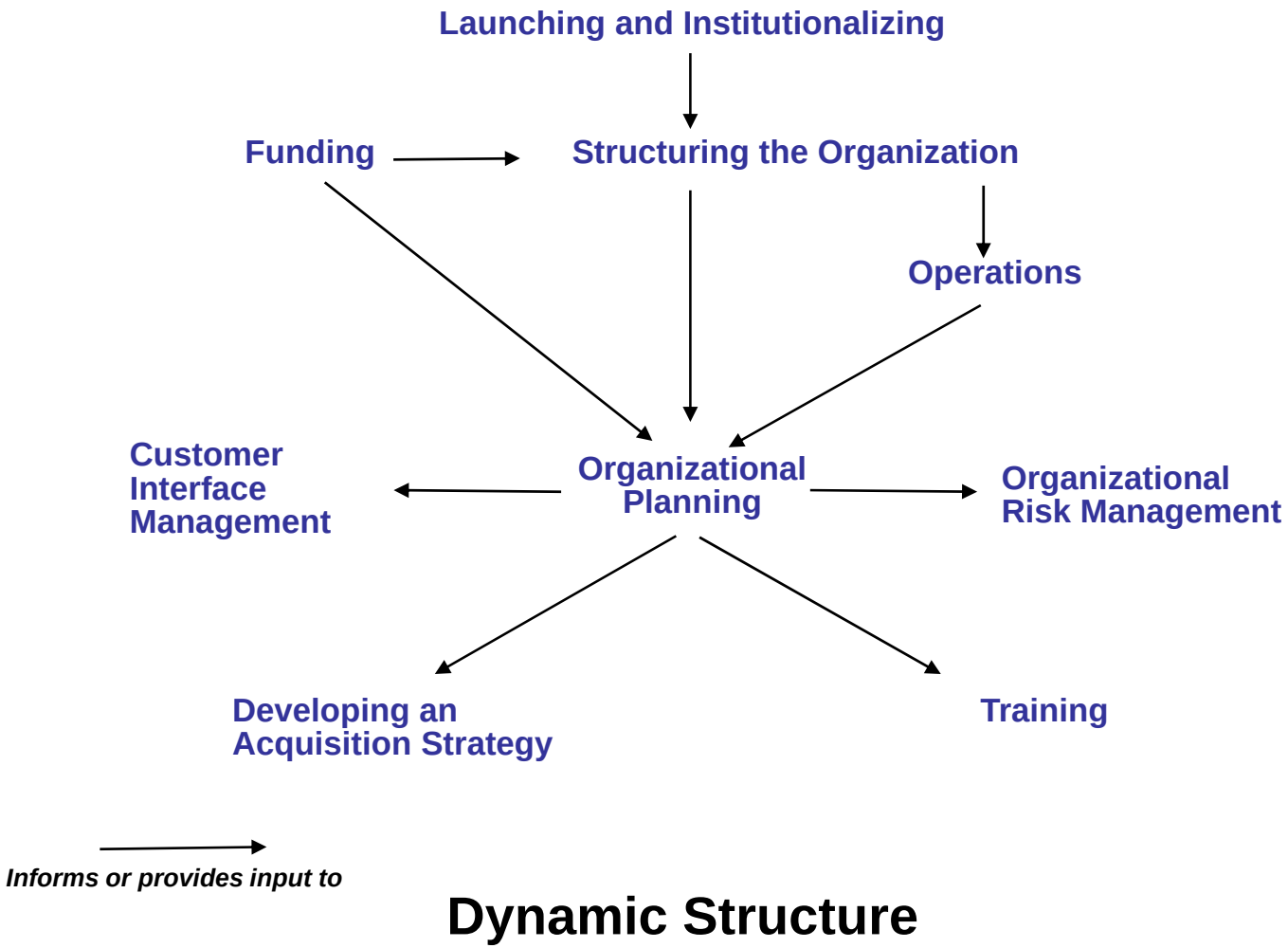
Name:

The ***Cold Start*** pattern should be used when an organization is launching a software product line for the first time.

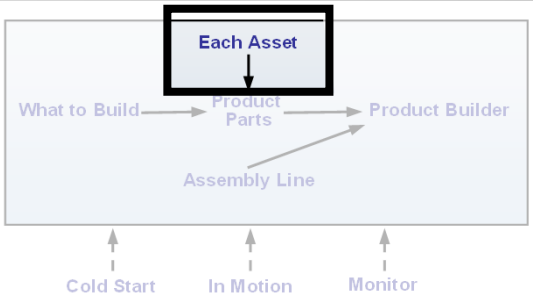
Context:

An organization is launching its first software product line.

Cold Start Pattern - 2



Each Asset Pattern - 1



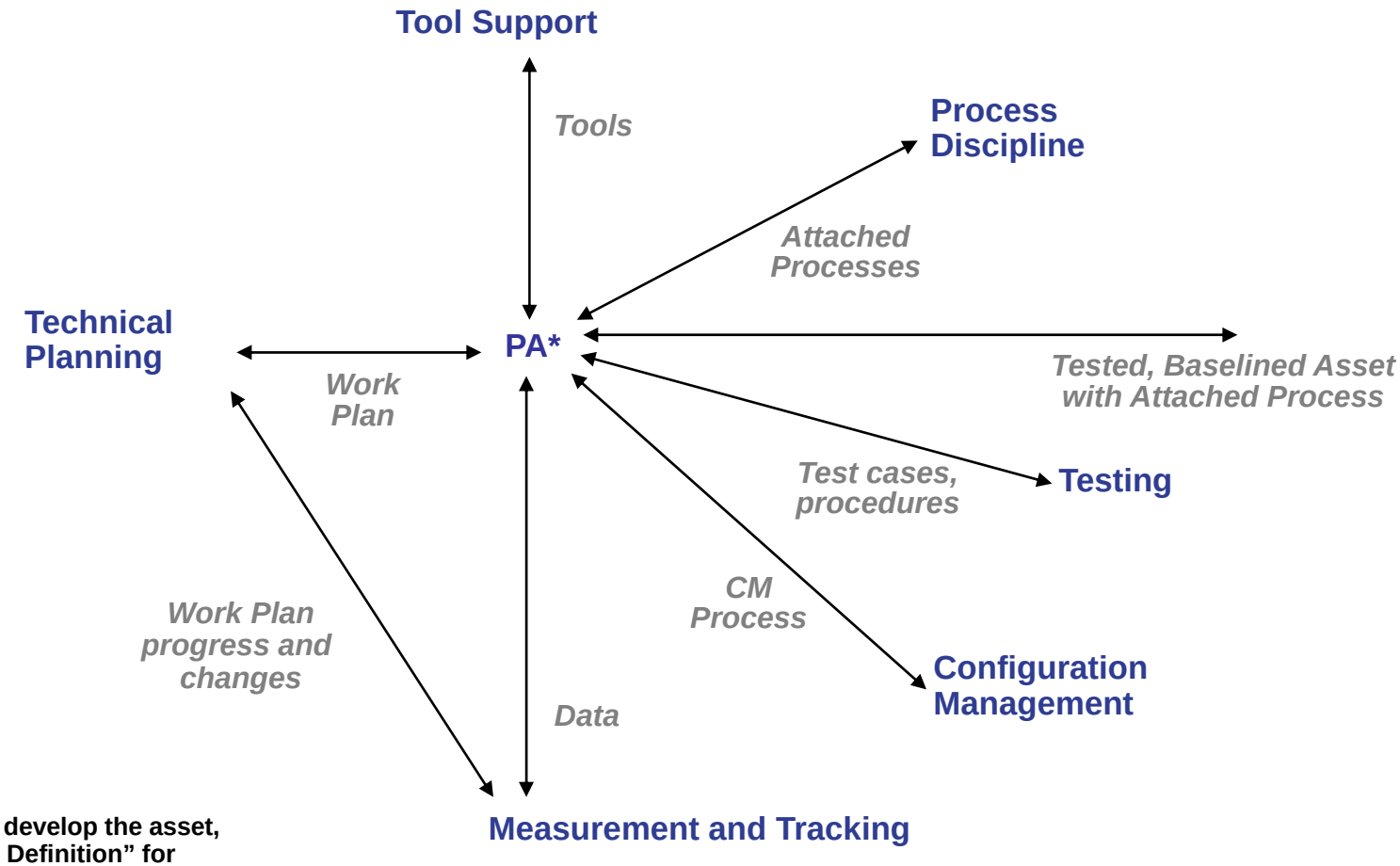
Name:

The ***Each Asset*** pattern should be used whenever any asset in the core asset base is being developed.

Context:

The pattern user knows the asset to be developed, has the specifications or other necessary information for the asset, and knows who will complete the task. The person(s) to complete the task is knowledgeable in that area.

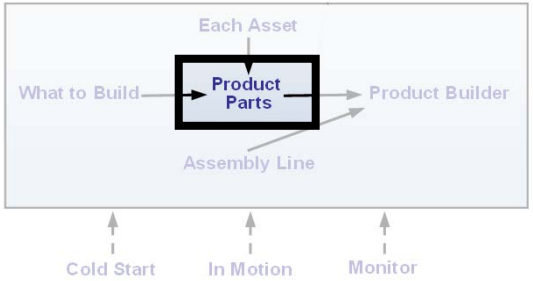
Each Asset Pattern - 2



PA* = practices to develop the asset, e.g., "Architecture Definition" for the architecture, "Requirements Engineering" for requirements, etc.

Dynamic Structure

Product Parts Pattern - 1



Name:

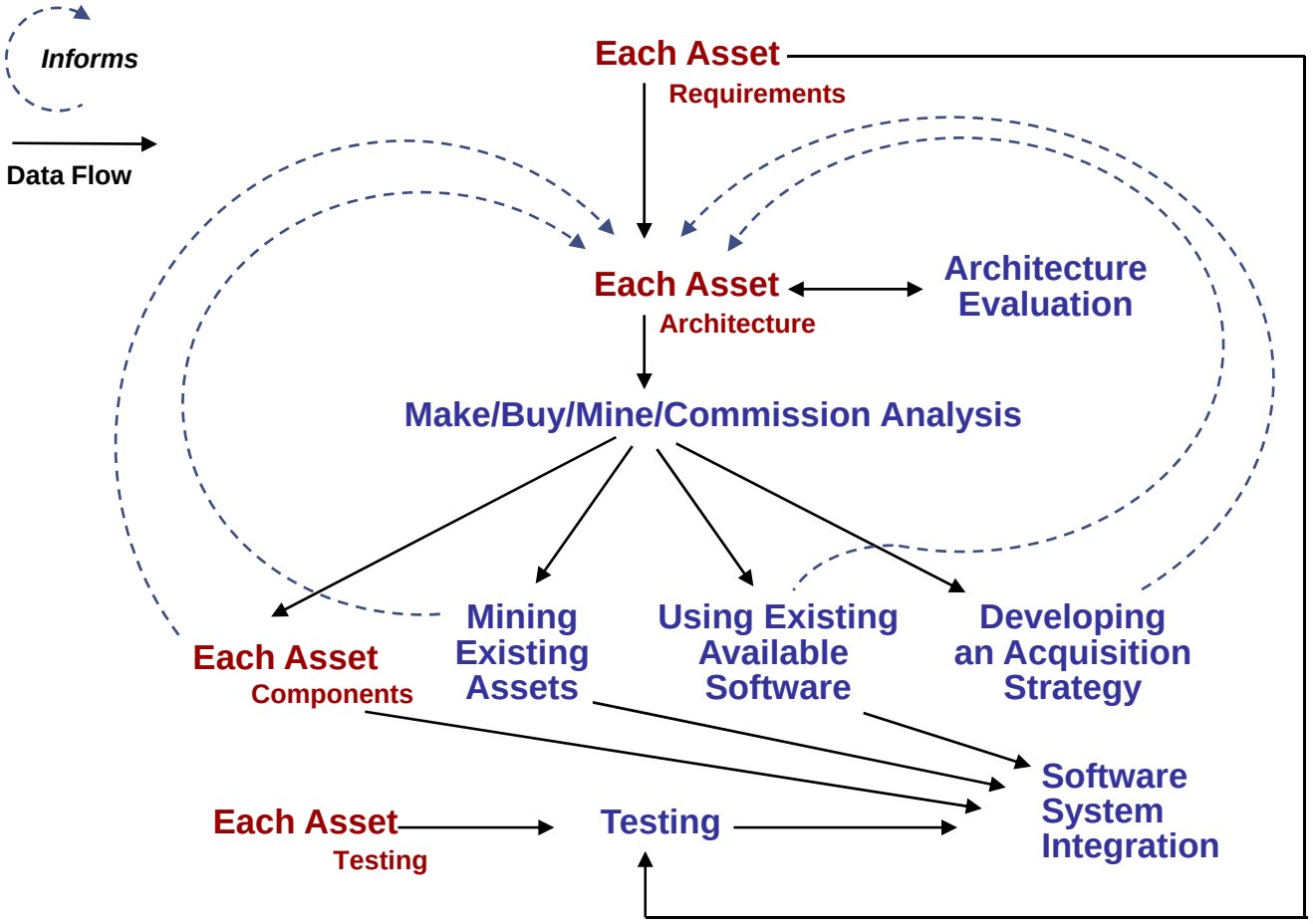
The ***Product Parts*** pattern is a composite pattern. It consists of practice areas and other patterns that should be used to provide the core assets that will be part of the products in the product line.

Context:

An organization knows what products are to be included in the product line and has designated knowledgeable individuals or groups to develop* the core assets.

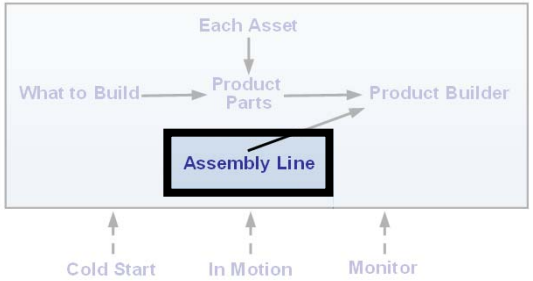
* *build, buy, mine, or contract*

Product Parts Pattern - 2



Dynamic Structure

Assembly Line Pattern - 1



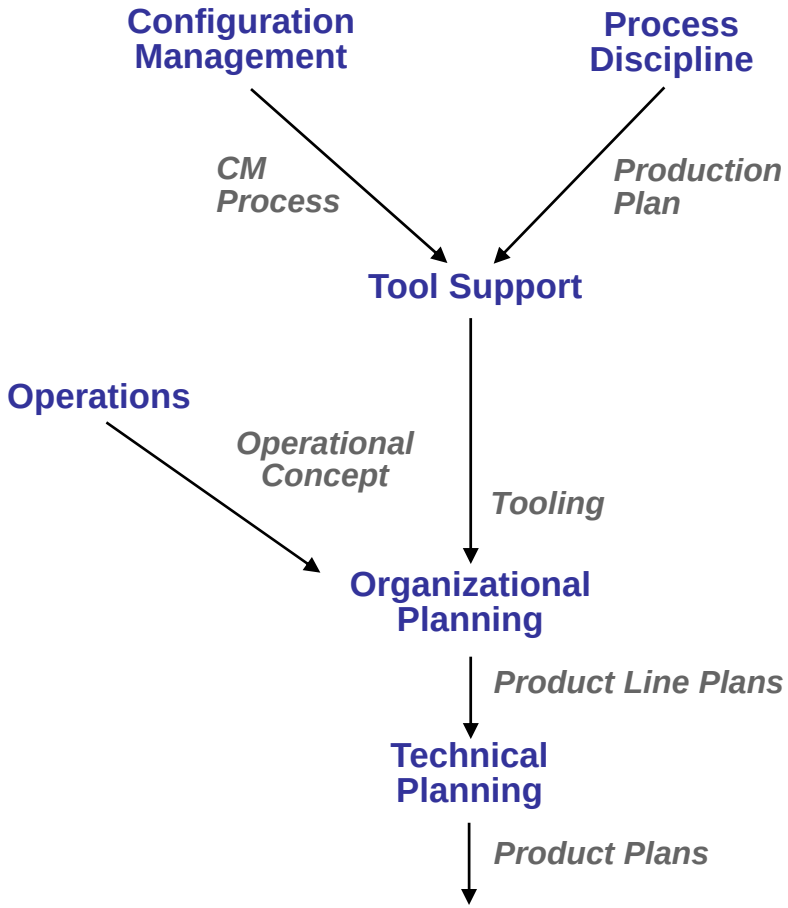
Name:

The ***Assembly Line*** pattern should be used to set up and run the production capability of a software product line.

Context:

An organization has made a decision to launch a product line effort.

Assembly Line Pattern - 2



Dynamic Structure

Process Pattern - 1

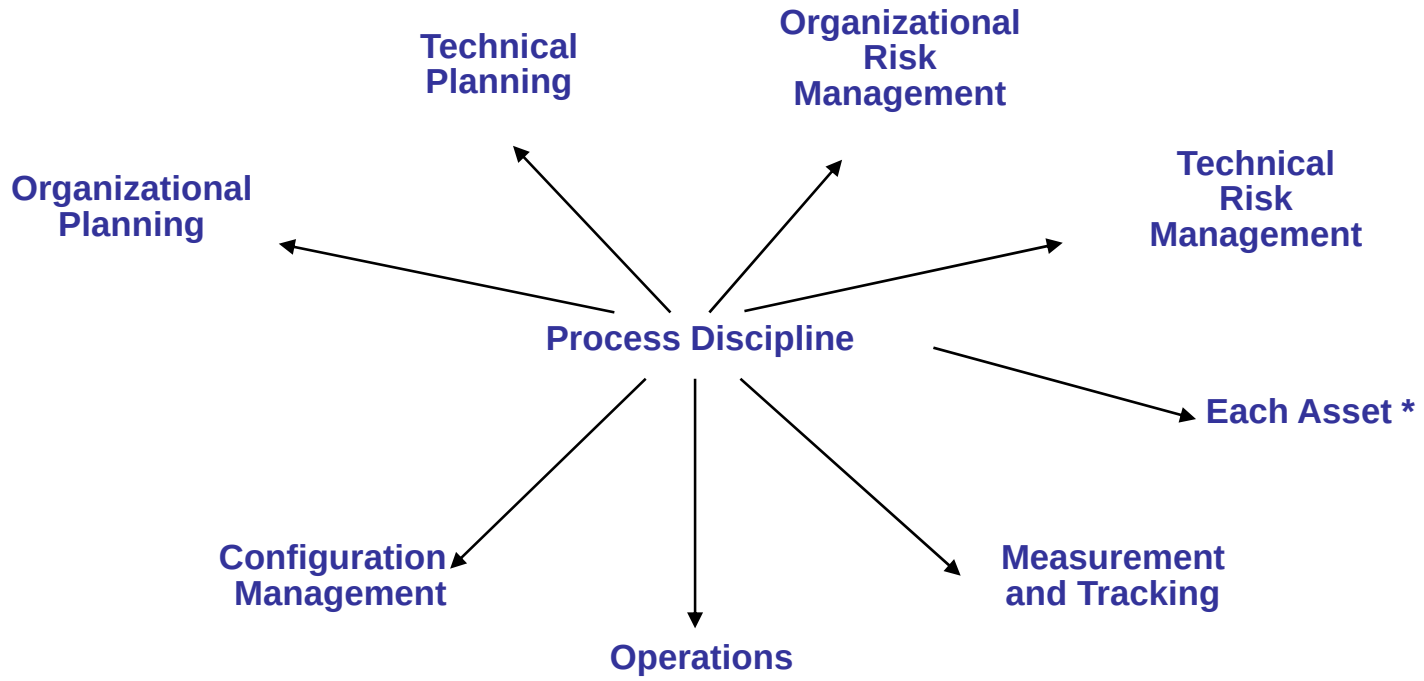
Name:

The ***Process*** pattern should be used to support all the product line activities that require processes.

Context:

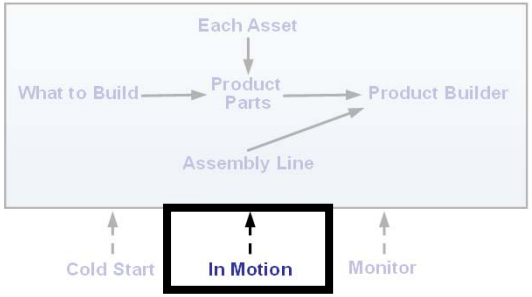
An organization has made a decision to launch a product line effort.

Process Pattern - 2



Dynamic Structure

In Motion Pattern - 1



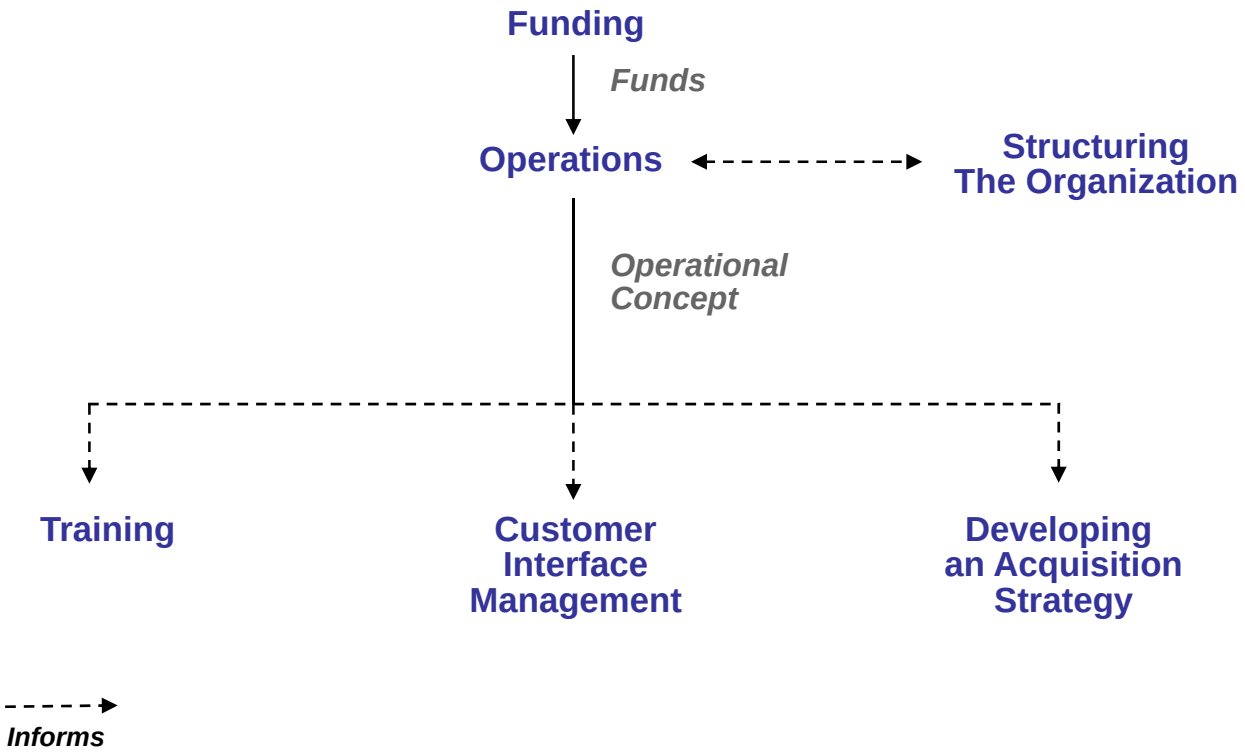
Name:

The ***In Motion*** pattern should be used to keep a product line effort in motion after it has been launched.

Context:

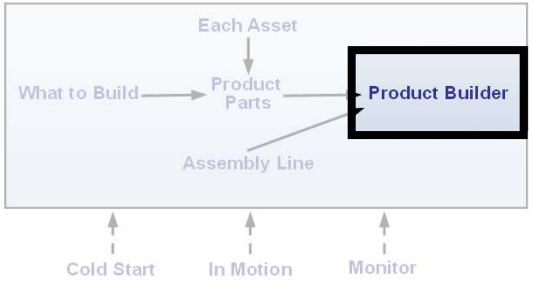
An organization has launched a product line effort.

In Motion Pattern - 2



Dynamic Structure

Product Builder Pattern - 1



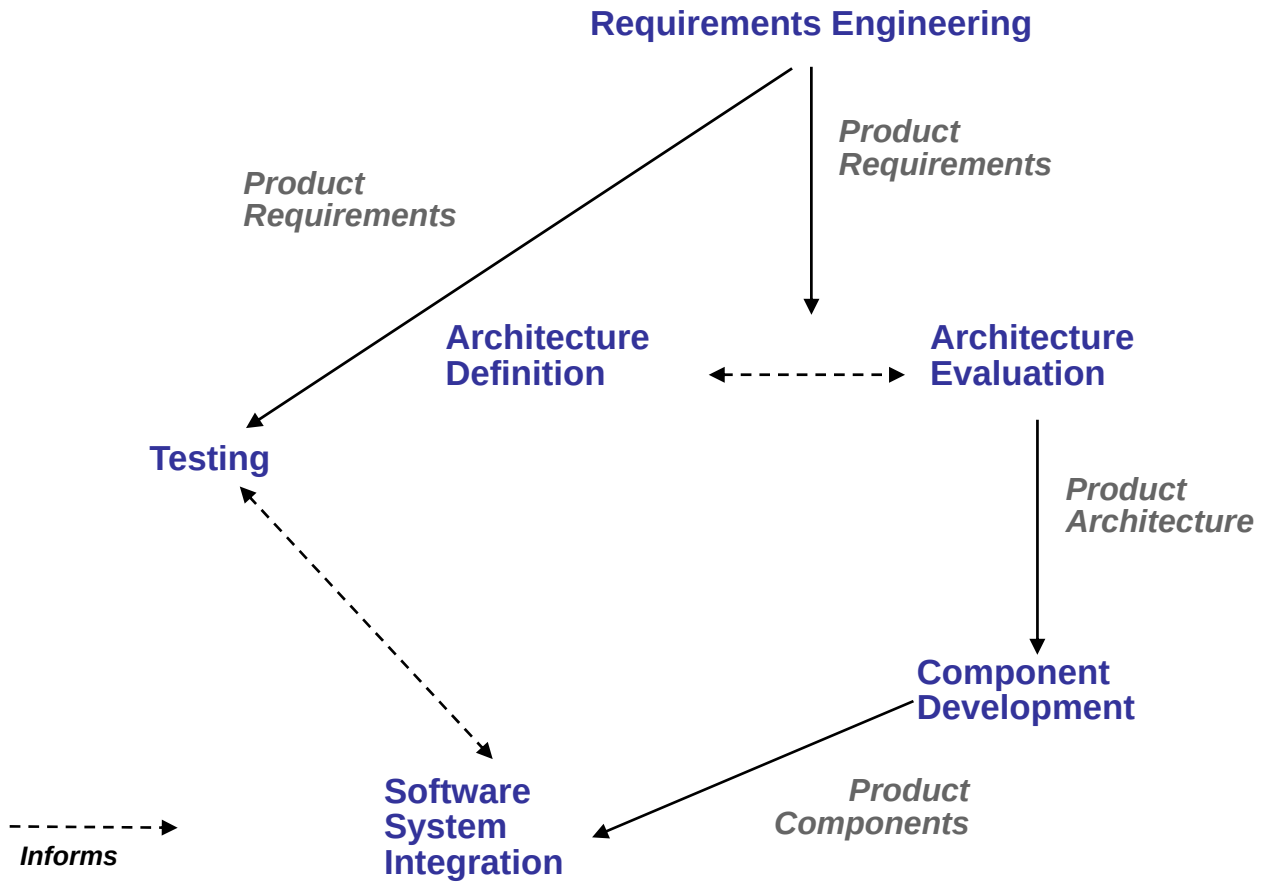
Name:

The ***Product Builder*** pattern should be used whenever any product in the product line is being developed.

Context:

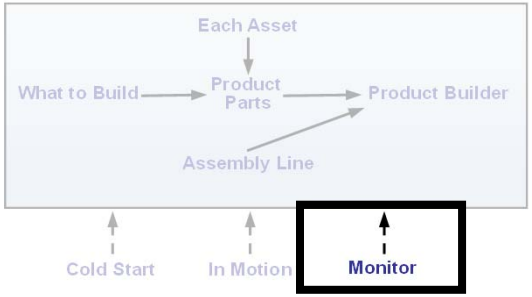
An organization has already established the production plan, the production capability, and the core asset base and has designated knowledgeable individuals or groups to develop a product that has been determined to be in the product line.

Product Builder Pattern - 2



Dynamic Structure

Monitor Pattern - 1



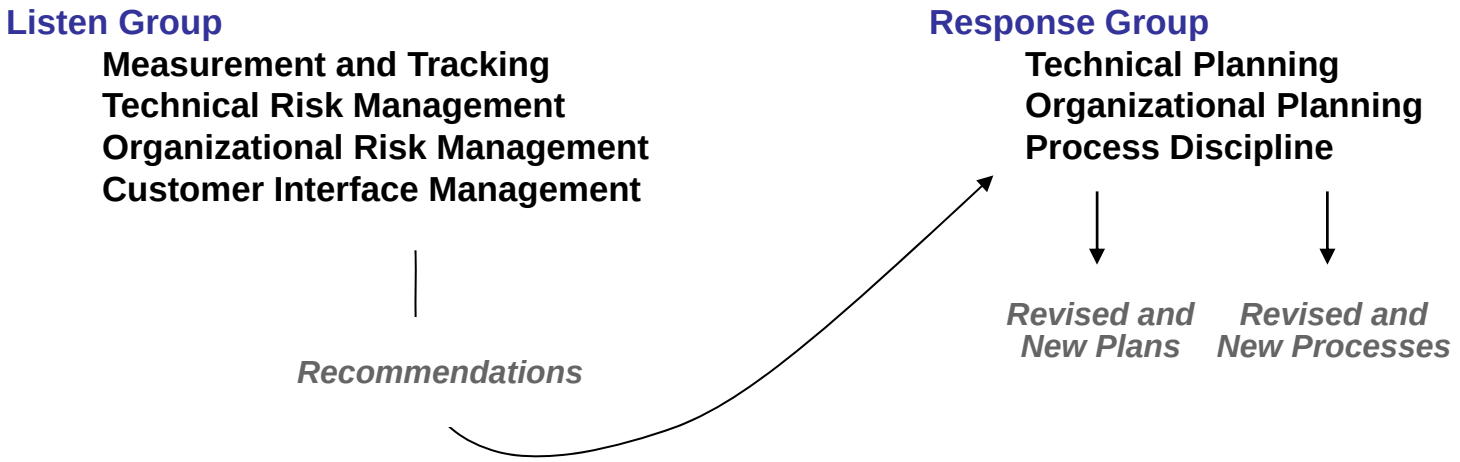
Name:

The ***Monitor*** pattern should be used to monitor an operational product line and apply course corrections to keep activities on track.

Context:

An organization has an operational product line.

Monitor Pattern - 2



Dynamic Structure

Using Patterns to Support Adoption

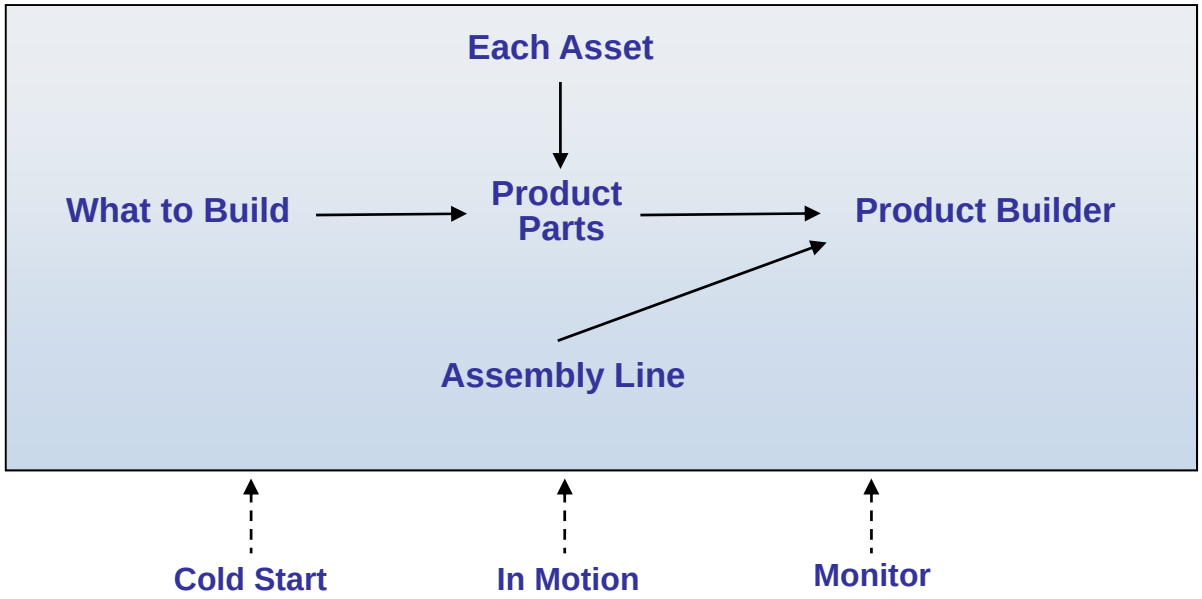
The “Launching and Institutionalizing” practice area lays out what needs to occur when an organization adopts a product line approach.

An organization that does not know how to go about product line adoption is unlikely to succeed (without pain and suffering).

A generic roadmap to product line adoption would be useful.

The ***Factory*** pattern can serve as the basis for such a roadmap.

Factory Pattern Revisited



Informs and information flow

Supports

Dynamic Structure

Session Topics

Review of Product Line Practice Patterns

The Adoption Factory Pattern

Useful Views of the Adoption Factory Pattern

Exercise

Summary

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 Discipline” 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

Name:

The ***Adoption Factory*** pattern is a composite pattern that describes the milestones in any product line adoption effort and their dependencies.

Context:

An organization is considering or fielding a product line for the first time.

Adoption Factory Pattern - 2

Solution:

A product line adoption roadmap must include the following seven major activities:

- deciding and justifying what products to include in the product line
- defining, documenting, and following processes for software development and management
- preparing the organization for a product line approach
- designing and providing the core assets that will be used to construct the products in the product line
- building and using the production infrastructure (necessary plans, processes, and tools)
- building products from the core assets in a prescribed way
- monitoring the product line effort, keeping a pulse on the adoption activities and the product line operations, and applying course corrections as necessary to keep the organization on course

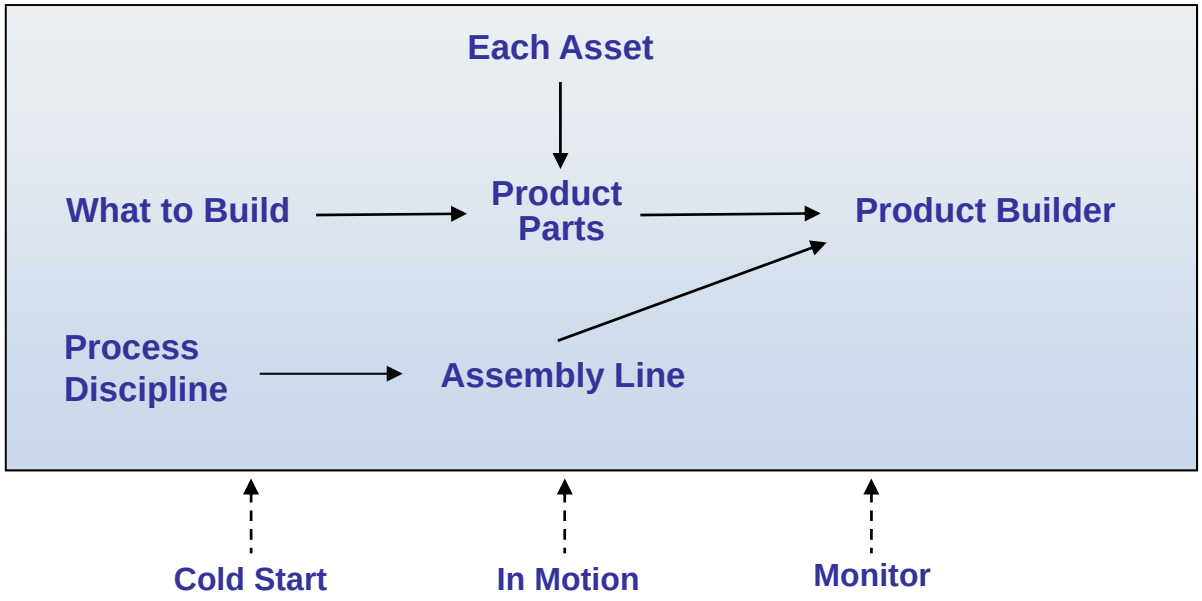
Adoption Factory Pattern - 3

Static:

The ***Adoption Factory*** pattern consists of the “Process Discipline” practice area and the following subpatterns:

- Assembly Line
- Cold Start
- Each Asset
- In Motion
- Monitor
- Product Builder
- Product Parts
- What to Build

Adoption Factory Pattern - 4



Informs and information flow

Supports

Dynamic Structure

Adoption Factory Pattern - 5

Application:

The ***Adoption Factory*** pattern can be used as a generic product line adoption roadmap for any organization attempting a product line approach for the first time. It can serve as the basis for a phased software product line or product line adoption plan.

Variants:

none to date but variations based on variants of the subpatterns it contains are certainly possible

Adoption Factory Pattern - 6

Consequences:

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.

Session Topics

Review of Product Line Practice Patterns

The Adoption Factory Pattern

Useful Views of the Adoption Factory Pattern

Exercise

Summary

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:

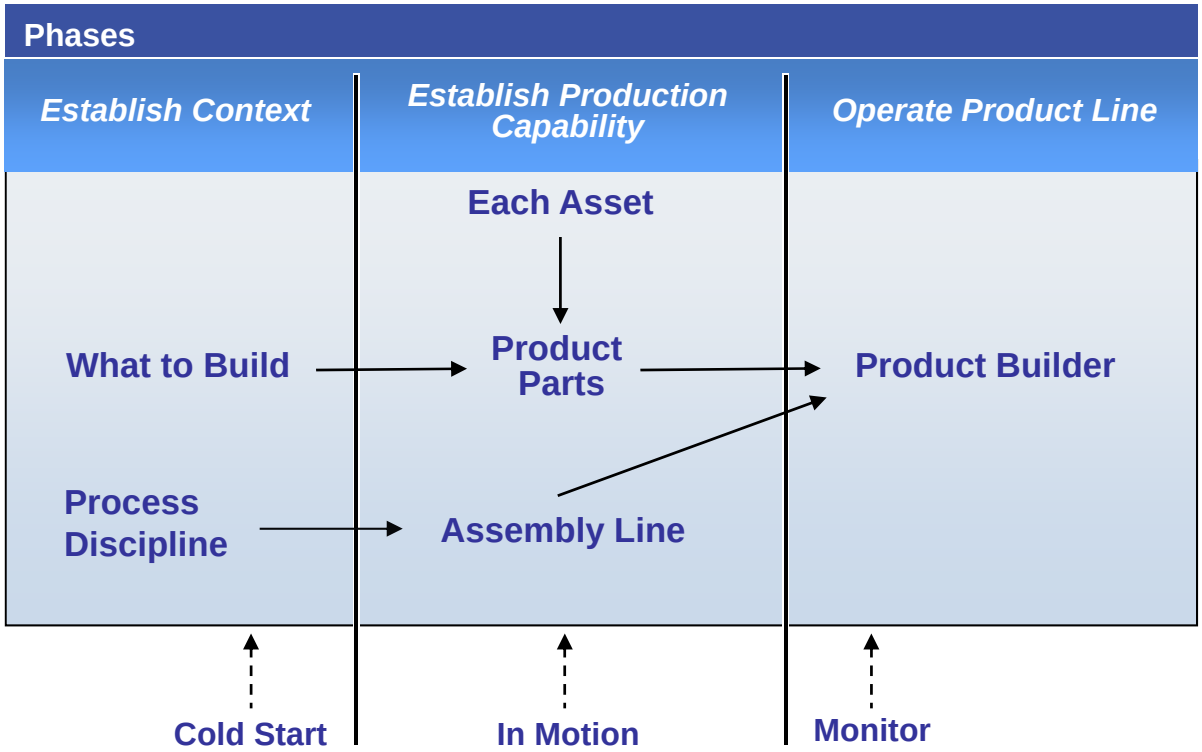
1. Adoption Phases
2. Focus Areas
3. Phases and Focus Areas
4. Practice Areas
5. Outputs
6. Roles

Adoption Phases View - 1

We can view the dynamic structure of the ***Adoption Factory*** pattern as three columns that designate the temporal phases of product line adoption as follows:

- **Establish Context:** involves paving the way for the product line adoption
- **Establish Production Capability:** involves developing the core asset base and the production infrastructure, and effectively managing those efforts at project and cross-project levels.
- **Operate Product Line:** involves using the core asset base to efficiently build products and effectively monitoring and improving the product line operation.

Adoption Phases View - 2



Informs and information flow

Supports

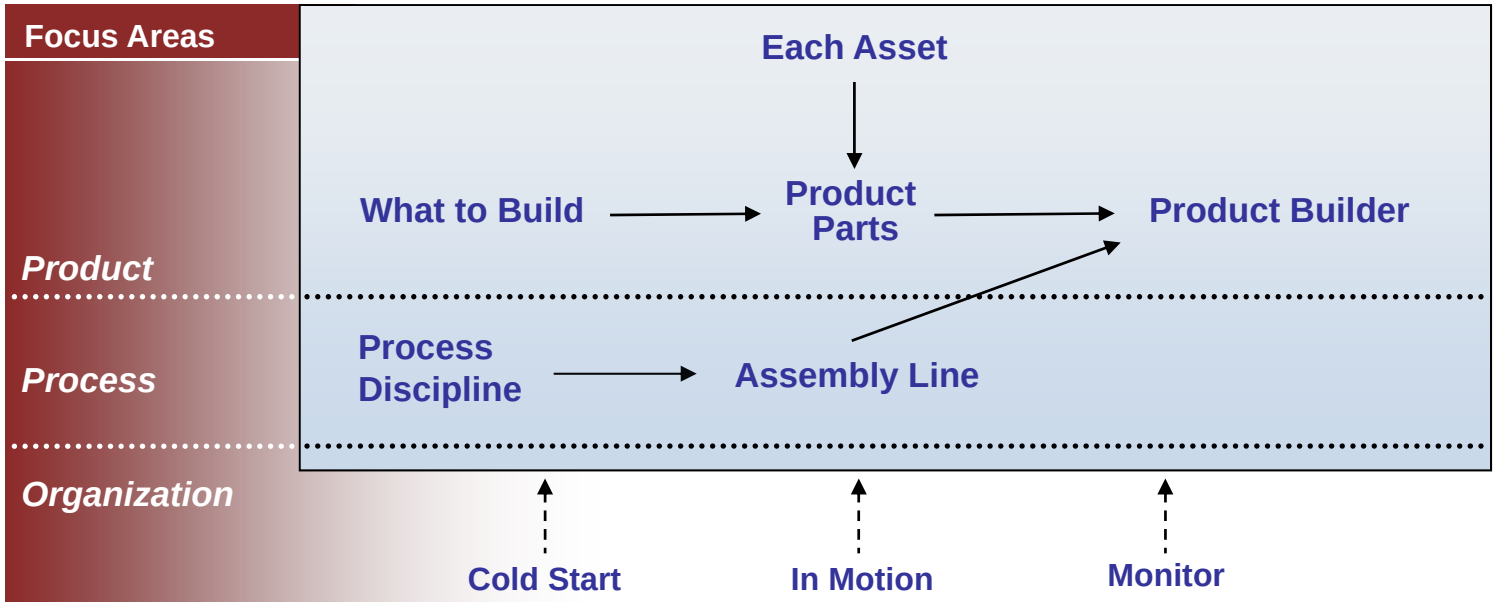
Adoption Factory Pattern

Focus Areas View - 1

We can view the dynamic structure of the ***Adoption Factory*** pattern as three rows, each one corresponding to a focus area for certain patterns and practice areas.

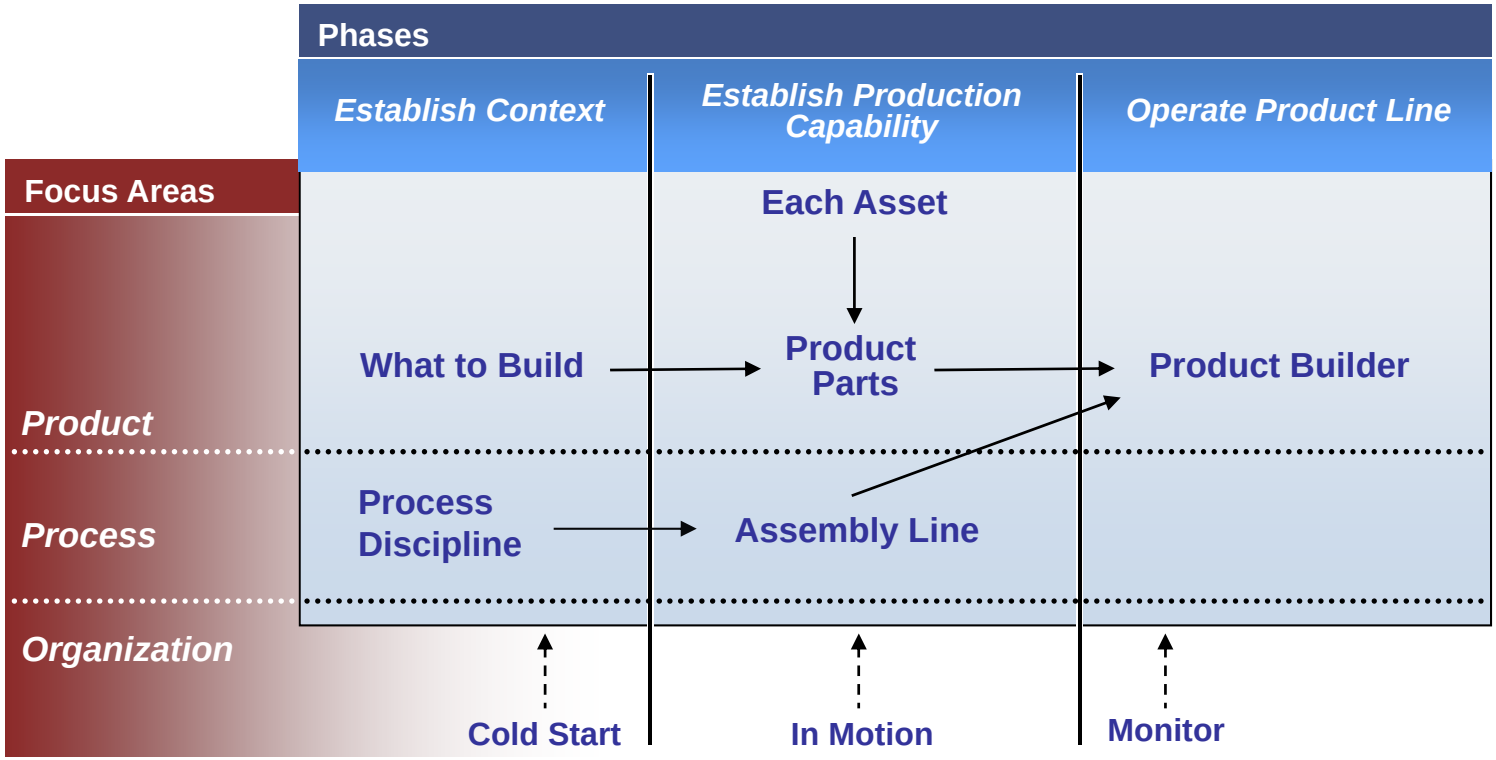
- **product:** involves those activities for defining and developing products and their common assets.
- **process:** involves the underlying processes and production infrastructure necessary to adopt a product line approach.
- **organization:** involves the management practices and activities necessary to adopt a product line approach and operate a software product line.

Focus Areas View - 2



Adoption Factory Pattern

Phases and Focus Areas View



Informs and information flow

Supports

Adoption Factory Pattern

Practice Areas View - 1

Detailed product line adoption planning requires the detail beneath the subpatterns of the ***Adoption Factory*** pattern.

That detail is provided by the practice areas associated with the subpatterns.

The Practice Areas view shows the ***Adoption Factory*** pattern and its constituent practice areas elaborated in a view that also shows the focus areas and adoption phases.

Some practice areas appear in multiple phases. The actual practices will vary dependent on the phase and the overall objective of the associated pattern.

Associated Practice Areas

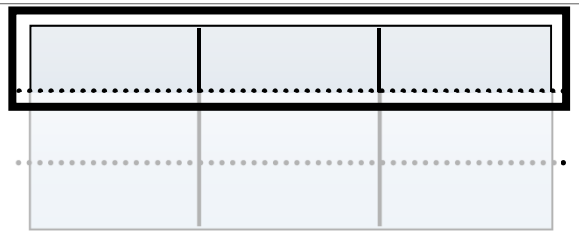
	Establish Context	Establish Production Capability	Operate Product Line
Product	• Market Analysis • Understanding Relevant Domains • Technology Forecasting • Building a Business Case • Scoping	• Requirements Engineering • Architecture Definition • Architecture Evaluation • Mining Existing Assets • Component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

Outputs View - 1

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.

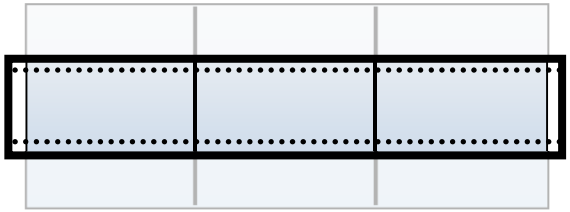
The information in this view can serve as a handy checklist for representative output from each phase.

Outputs View - 2



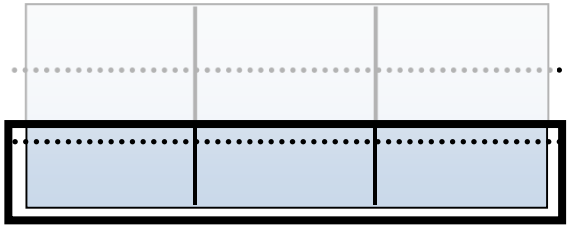
	Establish Context Phase	Establish Production Capability Phase	Operate Product Line Phase
Product outputs	• marketing description • domain model • technology survey • economic model • business use cases • cost/benefit model • business case • scope definition	• product line requirements • product line architecture documentation • product line architecture evaluation report • asset inventory • mining plan and process • mined assets • externally available software (EAS) criteria • EAS assets • core components • product line test strategy, • test cases, test architecture, test scripts, and test plan • attached processes • *products*	• product requirements • product architecture documentation • product architecture evaluation report • product specific components (mined, EAS, or built new) • product test strategy, test cases, test architecture, test plan • **products**

Outputs View - 3



	Establish Context Phase	Establish Production Capability Phase	Operate Product Line Phase
Process outputs	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 - 4



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

Roles View - 1

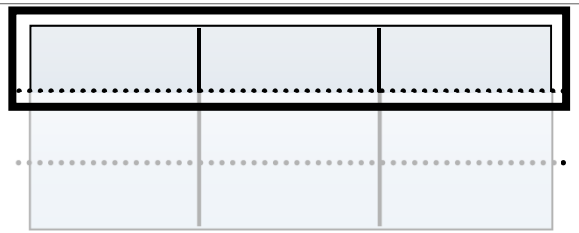
Another instructive view depicts the type of people who need to be involved in the product line adoption effort.

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.

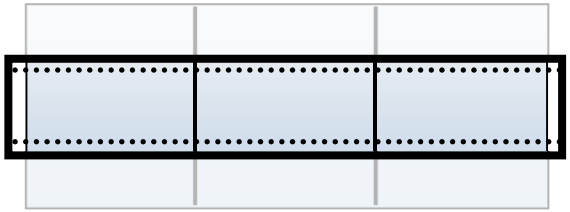
Some roles may appear in multiple phases, but the tasks those roles perform will vary with the phase.

Roles View - 2



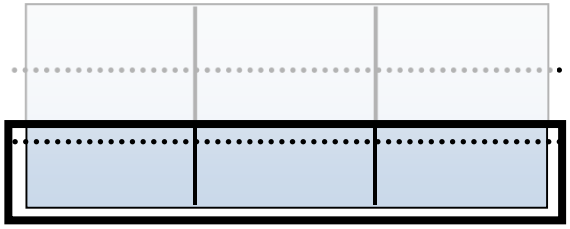
	Establish Context Phase	Establish Production Capability Phase	Operate Product Line Phase
Product-related roles	• marketer • market analyst • domain expert • product manager • senior manager • technology scout • architect	core asset developer: • requirements engineer • architect • architecture evaluator • component developer • tester • software integrator	product developer: • requirements engineer • architect • architecture evaluator • component developer • tester • software integrator

Roles - 3



	Establish Context Phase	Establish Production Capability Phase	Operate Product Line Phase
Process-related roles	• technical manager • process owner • process group member	• technical manager • process owner • process group member • technical support • tool specialist • measurement specialist	

Roles - 4



	Establish Context Phase	Establish Production Capability Phase	Operate Product Line Phase
Organization-related roles	• product line manager • software manager • business unit or organization manager • product manager • acquisition expert • financial manager • human resource manager • training planner • training developer • trainer	• product line manager • software manager • business unit or organization manager • financial manager • training developer • trainer	• product line manager • product manager • business unit or organization manager • customer field representative • salesperson

Session Topics

Review of Product Line Practice Patterns

The Adoption Factory Pattern

Useful Views of the Adoption Factory Pattern

Exercise

Summary

Exercise

See the exercise handout.

Session Topics

Review of Product Line Practice Patterns

The Adoption Factory Pattern

Useful Views of the Adoption Factory Pattern

Exercise

Summary

Summary

Product line practice patterns are helpful in planning and implementing a product line effort.

- They provide a useful abstraction to conquer the 29 practice areas.

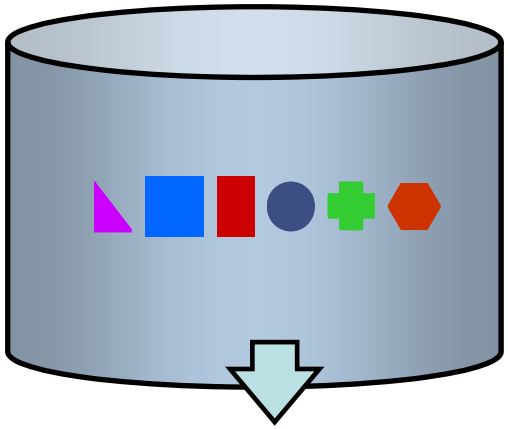
The Adoption Factory pattern, which is a variant of the Factory pattern, is a generic, phased product line adoption roadmap.

The Adoption Factory pattern has six views that provide milestones, dependencies, and underlying detail.

Factors Influencing Adoption

Organizational Context

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



Product Line Adoption Plan

. . .

Product Line Action Plans

Adoption Support

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

Software Engineering Institute | Carnegie Mellon

Adopting Software Product Lines

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Part 2: Using Adoption Models
Module 4: Change Models and Mechanisms

Session Outcomes

After this session participants should

- understand some of the operative forces in change
- be familiar with some basic concepts and change models
- be able to anticipate ways people resist change in software product line adoption
- know some techniques to mitigate resistance in software product line adoption and to facilitate the associated change

Session Topics

Some basic concepts of change

Some change models

Some critical factors involved in organizational change

Exercise

Summary

About This Session

> "Nothing endures but change."
> *Heraclitus*

Product line adoption is a change that impacts all aspects of the way an organization does business.

There are technical changes and there are non-technical changes.

Change experts have models and practices to assist in ensuring successful change.

In this session we show sample ideas from change experts that are useful in product line adoption.

Truth in advertising:

- This is not a change management course.
- Our sampling is necessarily incomplete.
- These ideas were developed independently.
  - Terminology and models are not unified.

Change Isn't Easy

Change should be undertaken to help an organization meet its business goals.

Because meaningful change is difficult, profound, and risky, it should not be left to chance.

Change should be deliberately planned and executed like a well-managed project.

The more you understand the forces involved in change, the better you are able to control change.

"There is nothing more difficult to plan, more doubtful of success, nor more dangerous to manage than the creation of a new order of things."

Niccolo Machiavelli

Roles For Implementing Change

Champions

- Advocate the technology and/or innovation and influence the organization to support its implementation

Sponsors

- Authorize and/or reinforce the change

Change Agents

- Implement the change effort as agents of the sponsors

Targets

- Adopt the change by collaborating on planning and enacting new practices and behaviors

NOTE: Roles often overlap and change with time and circumstance, for example, a target may become a new sponsor.

Roles for Implementing Change: Discussion

In your organization, who plays the following roles in product line adoption?

- Champion
- Sponsor
- Change agent
- Target

Session Topics

Some basic concepts of change

Some change models

Some critical factors involved in organizational change

Exercise

Summary

Some Change Models

Satir – Weinberg

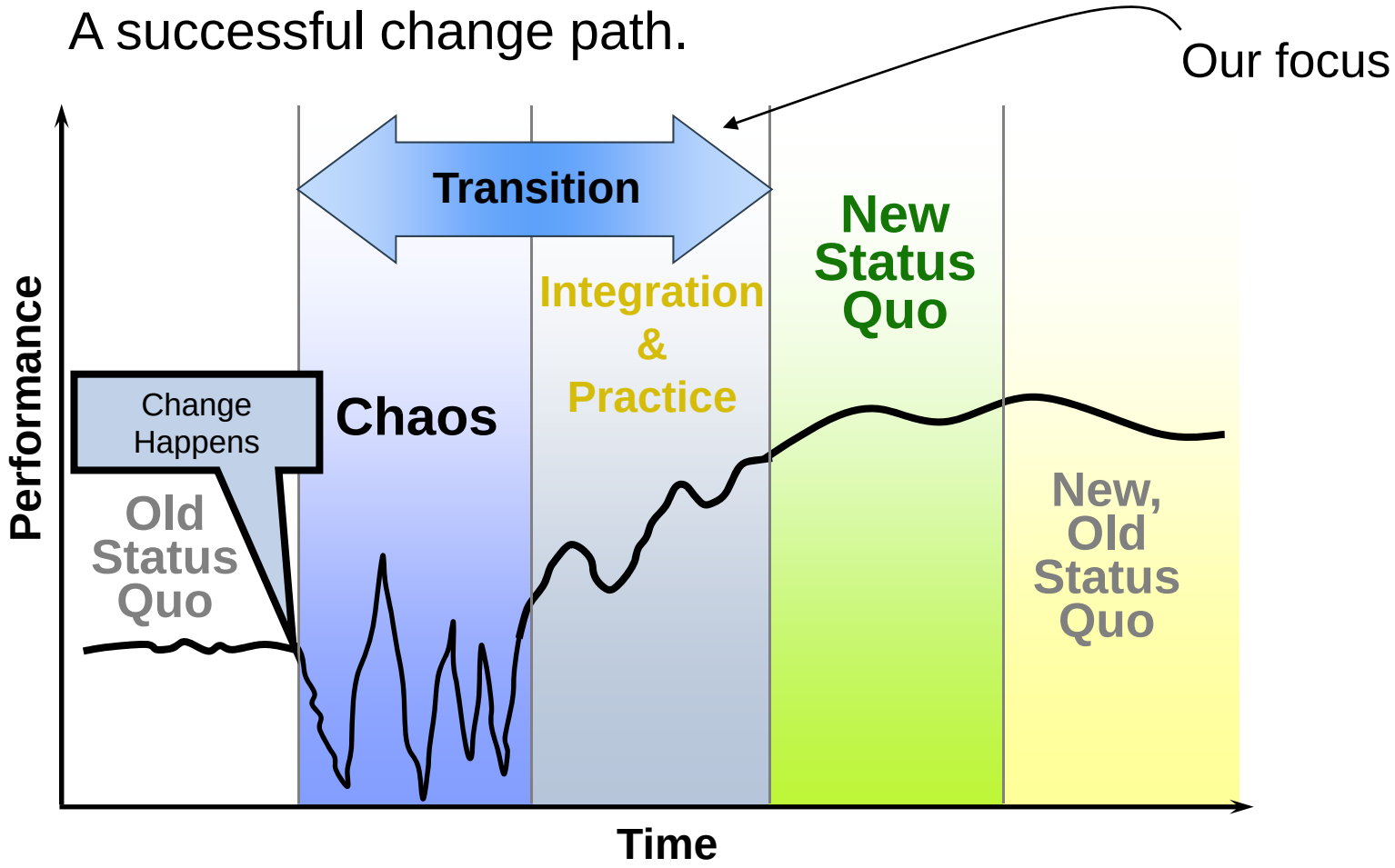
Patterson – Connor

Moore

"All models are wrong, but some are useful."

George Box
(Quality and Statistics Engineer)

Satir - Weinberg Change Model



For a detailed discussion of Virginia Satir's model, see G. Weinberg, *Quality Software Management, Vol. 4: Anticipating Change*, ISBN 0-932633-32-3.

The Transition State

The transition state is the state during which an organization changes from one way of operating to another.

Key challenges in change management are to

- minimize the duration of the transition state
- damp the chaotic responses that occur during the transition state

The material in this session is geared to helping you manage the transition state.

Characteristics of the Transition State

High uncertainty, low stability

Poor information: insufficient and often conflicting

- high perceived inconsistency

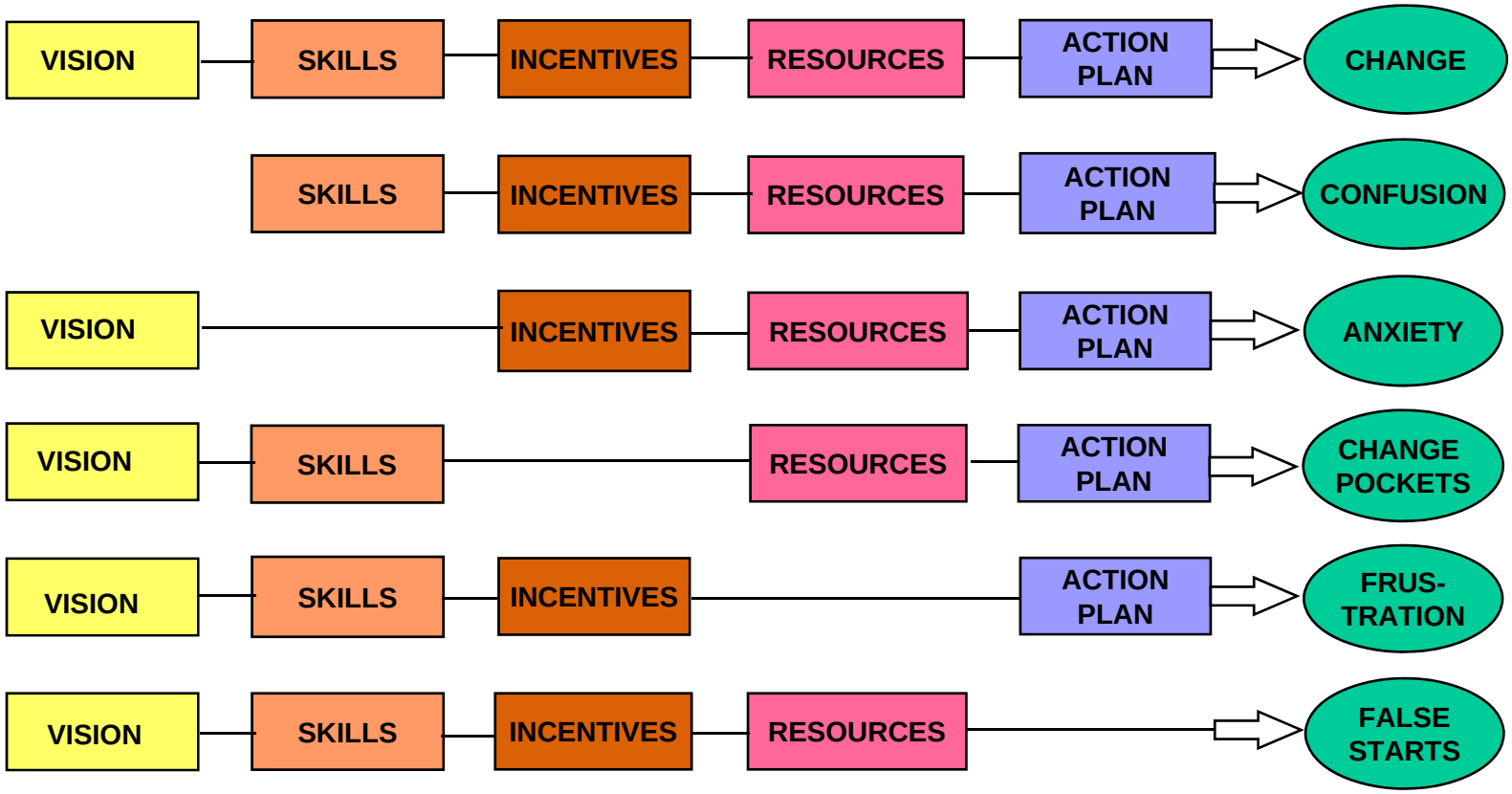
Emotional stress

- high energy, often undirected
- increased conflict, particularly across groups

Yearning for the “good old days”

- previous ways of doing things become revered and clung to more tenaciously

Transition Essentials



"Managing Technological Change"
Carnegie Mellon University
Software Engineering Institute

What People Also Need During Transition

Information

- what? why? when? how? – frequently updated

Inclusion

- involve stakeholders, buy into the vision

Support

- communications, resources, reassurance

Freedom

- to contribute
- from blame

What Organizations Also Need During Transition

Sponsorship and champions

Structures for managing the transition

Effective, two-way communication mechanisms

- once is not enough
- transmission does not guarantee reception (or understanding)

Leverage points for getting started

Islands of stability

- clarity about not only what is changing, but what isn't changing

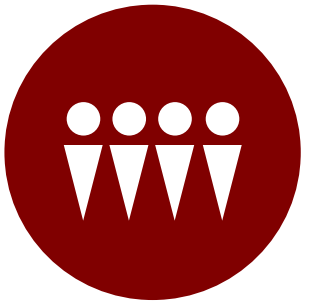
Attention to organizational dynamics

Example Change Structures - 1

Teams may be formed and given responsibilities and roles for managing, facilitating, and implementing a change effort.

Steering Group to guide and enable

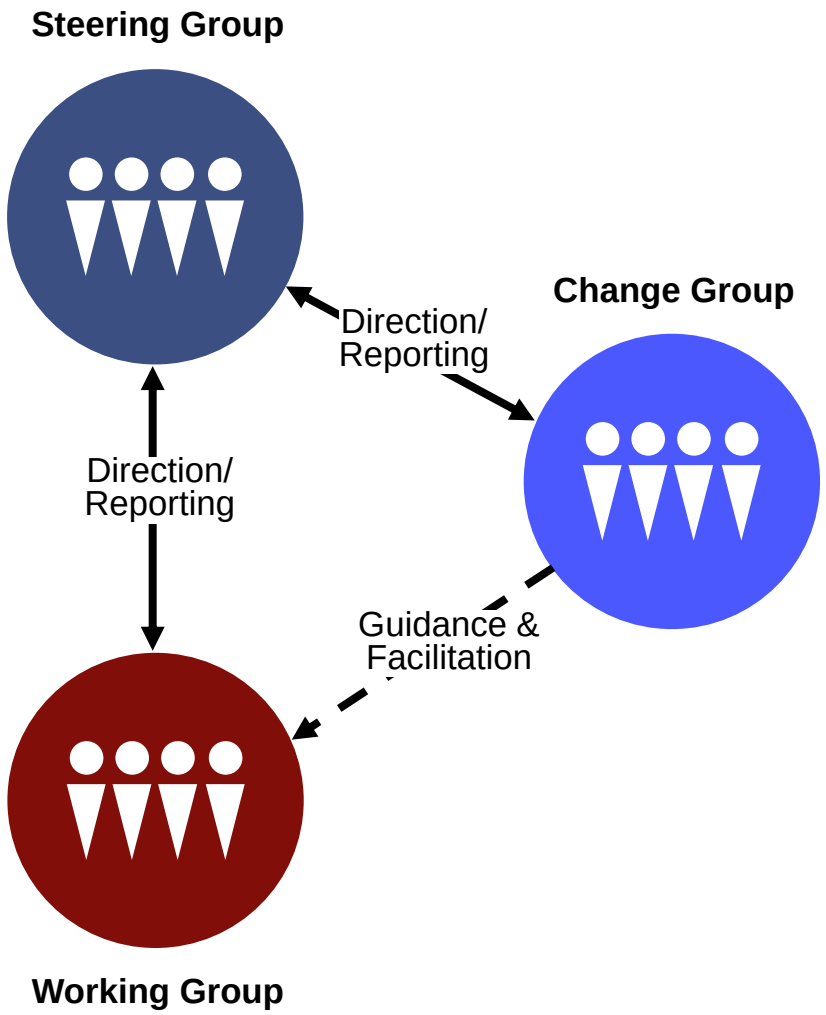
Change Group to facilitate



Working Groups to do detailed work

Example Change Structures - 2

It is useful to establish these types of change structures and mechanisms early.



Working Groups are not formed until later; they actually do much of the implementation.

Change Structure Evolution

In some change initiatives, it makes sense to maintain these change structures indefinitely, for example, for continuous process improvement.

In product line adoption, it may make sense to include these change structures during launching.

As the organization institutionalizes a product line approach, many of these activities become routine duties within the product line organization and the change structures disappear.

Change Structure Evolution - Discussion

What practice areas relate to change structure evolution?

What product line practice patterns relate?

Using Satir and Weinberg's Model

Understand

- Change cannot just be dumped on the organization
- Successful change involves managing the transition state.
- The people and the organization have needs during the transition state.
  - Addressing these needs will shorten the duration of the transition state and reduce the disruption.

Using Satir - Weinberg's Model - Discussion

How could you utilize the Adoption Factory to help address the following transition needs? (Also what gaps remain outside what the Adoption Factory covers?)

- vision
- skills
- incentives
- resources
- information and communication
- inclusion
- sponsors and champions
- change transition structures

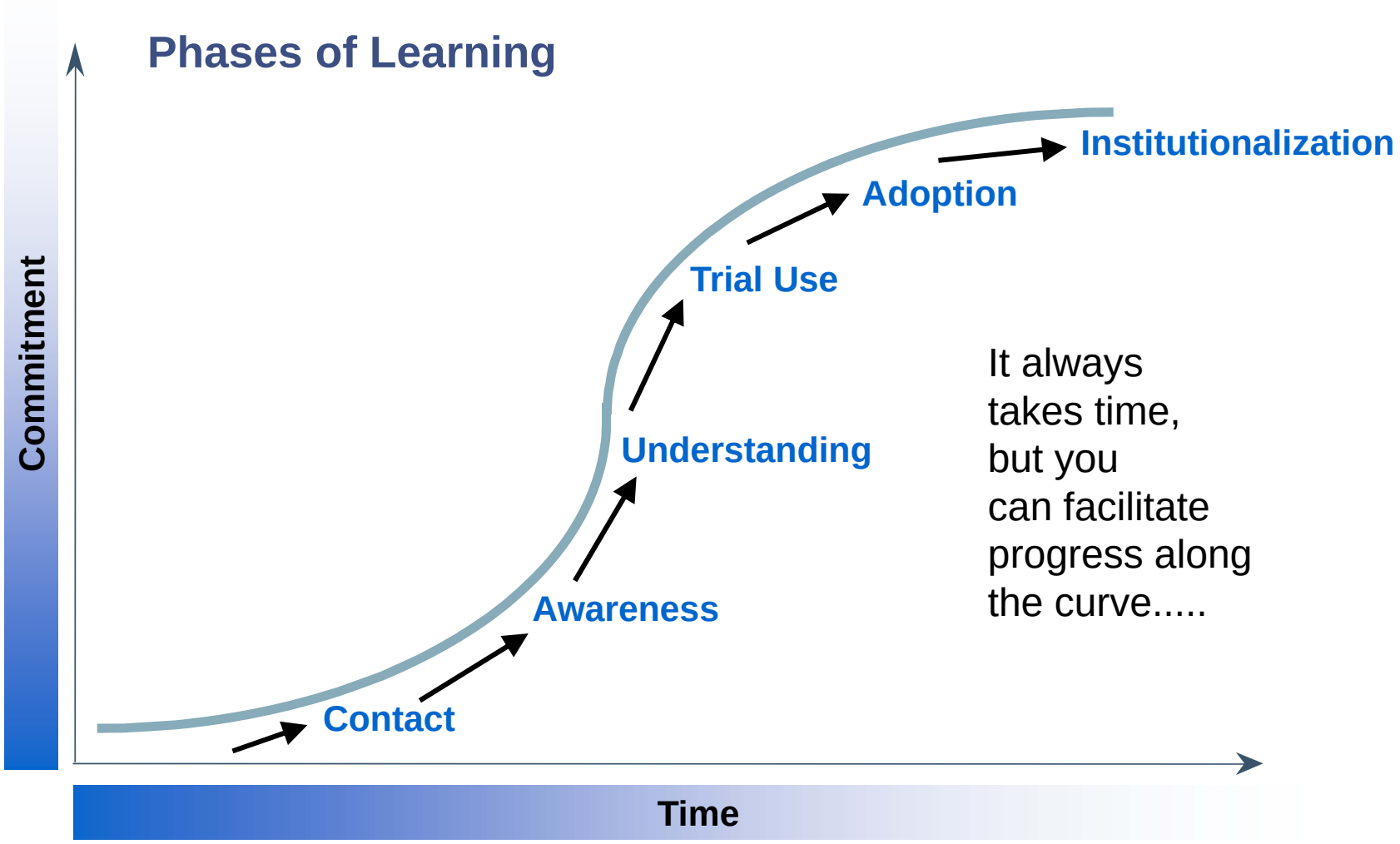
Some Change Models

Satir – Weinberg

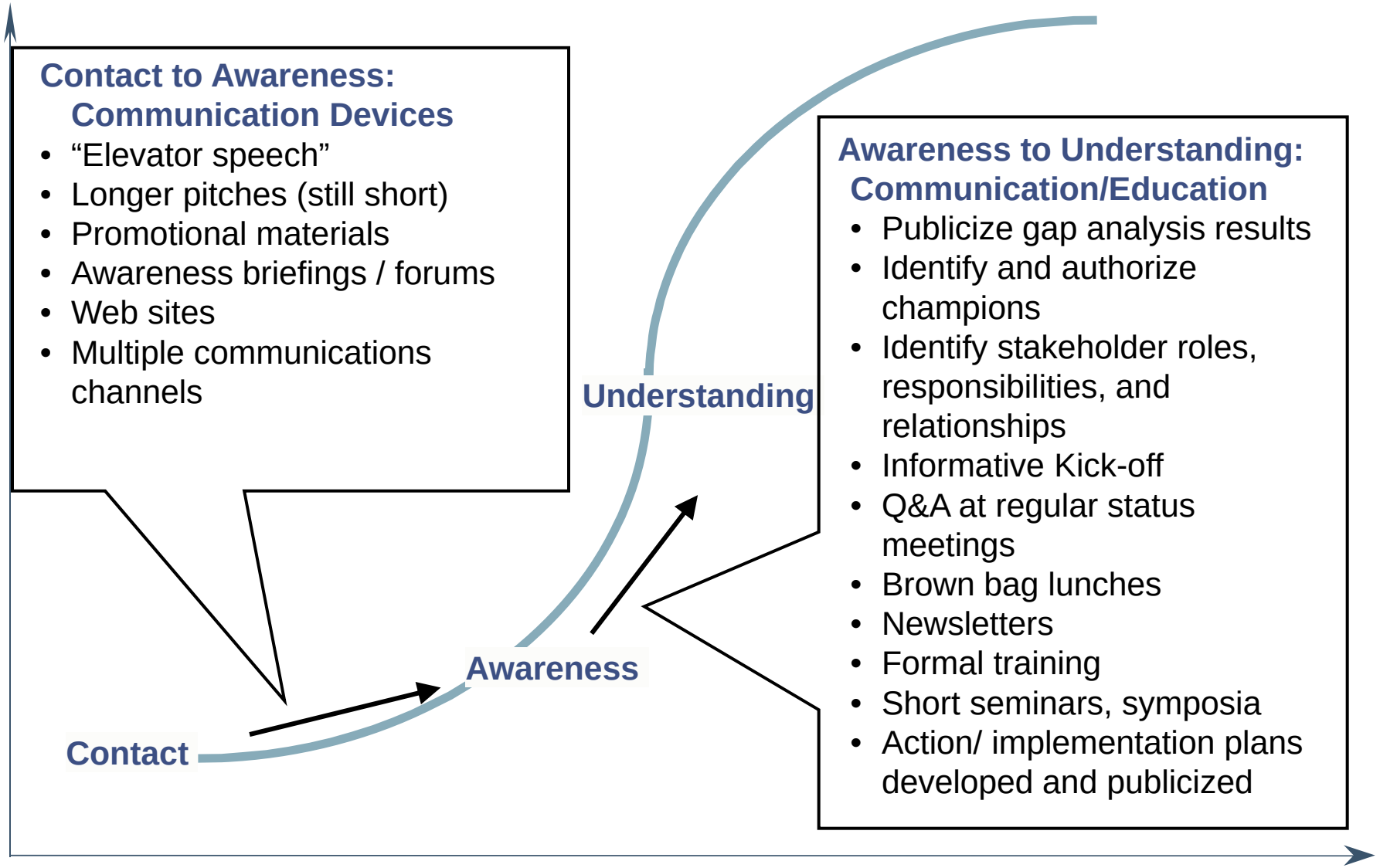
Patterson – Connor

Moore

Patterson - Connor Change Adoption Model



Mechanisms to Facilitate Progress



Adapted from *CMU/SEI-2002-TR-007 (The Road to CMMI …)*

Patterson – Connor Model: Discussion

Contact to Awareness: Communication Devices

- What sort of promotional materials could be used in product line adoption?
- What communication channels would you target?

Awareness to Understanding: Communication/Education

- What practice areas relate to these activities?
- Who are the stakeholders in product line adoption?

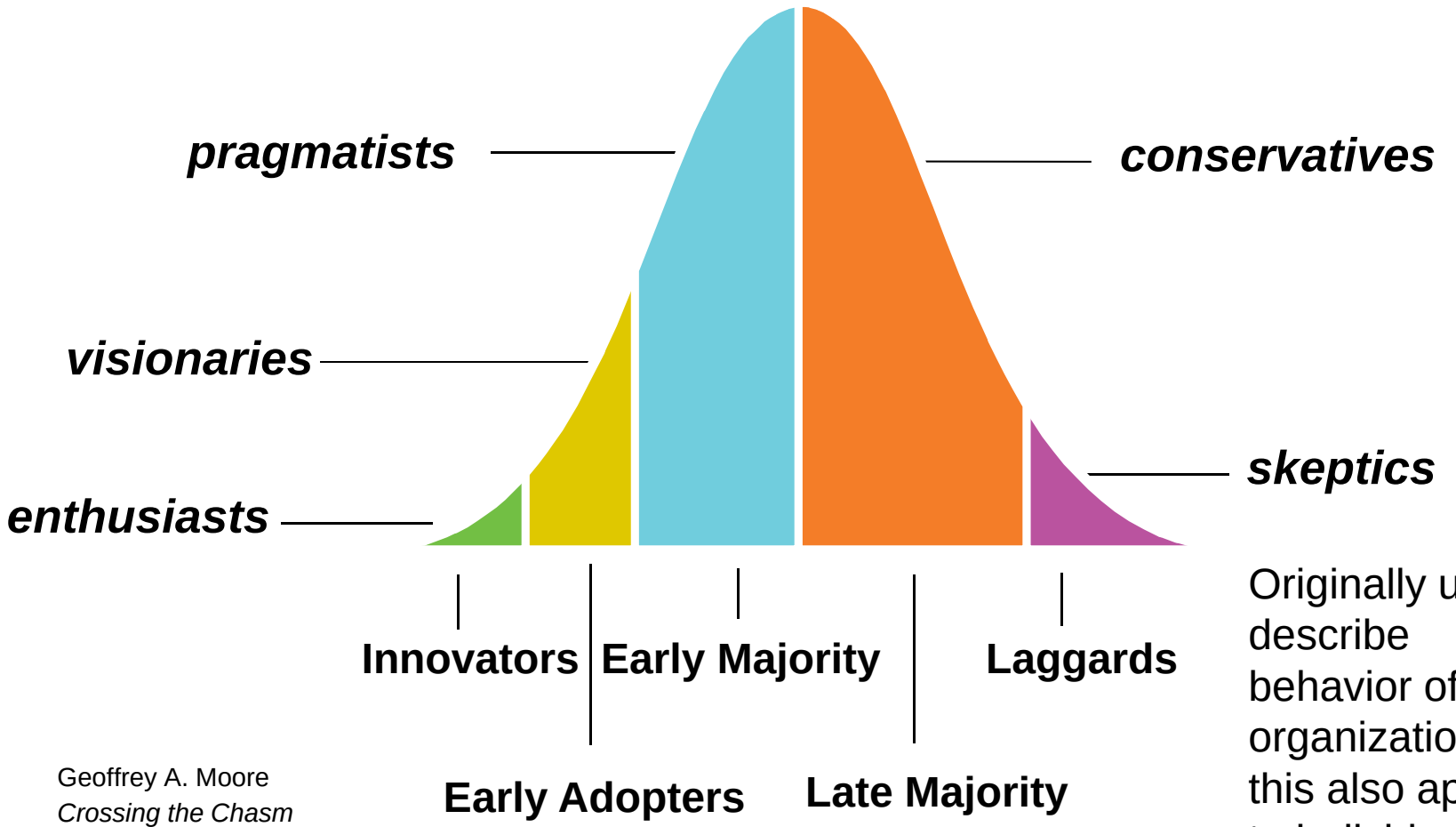
Some Change Models

Satir – Weinberg

Patterson – Connor

Moore

Moore's Technology Adoption Life Cycle



Geoffrey A. Moore
Crossing the Chasm

Originally used to describe behavior of organizations, this also applies to individuals

How Do I Reach These Groups? - 1

Innovators

- Focus on technical goals.
- Put messages where innovators hang out.
- Give them the unabashed truth.
- Avoid image-type embellishments.
- Let them try things out to see how they work.

Early Adopters

- Communicate the vision.
- Focus on business potential (innovation).
- Build strategy around “productizing.”
- Produce concrete deliverables early and often.
- Manage expectations carefully.

How Do I Reach These Groups? - 2

Early Majority

- Show proof of (credible) others' success.
- Focus on predictability and measurability of progress and results (improvement).
- Develop strategies to minimize and manage risk.
- Build relationships of trust based on dependability.
- Be sensitive to cost issues.
- Develop a long-term agenda.
- Be patient.

How Do I Reach These Groups? - 3

Late Majority

- Focus on who else is doing it.
- Emphasize maturity of the technology.
- Develop bundled packages.
- Think through, and present, a “whole solution.”

Laggards

- You won’t reach them.
- Think of ways to be polite yet minimize their disruption.

Moore's Life Cycle: Discussion

Most organizations have employees from a mix of the groups Moore identifies.

Do you see this in your organization?

How will you use Moore's characterization to your advantage in product line adoption?

Successful product line adoption ultimately comes down to changing the behavior of people from which group?

How will you do that?

Session Topics

Some basic concepts of change

Some change models

Some critical factors involved in organizational change

Exercise

Summary

Some Critical Factors in Organizational Change

Prior experiences with change

Sponsorship

Change agent skills

Resistance

Organizational culture

An organization will typically have aspects of these factors that work

- against the change (barriers)
- in favor of the change (enablers)

The challenge: How do you lower the barriers and capitalize on the enablers?

Prior Experiences with Change - 1

What is the organization's experience with on-going and previous change initiatives?

- A previous failure will make things more difficult.
- A previous success may make things easier.

Prior Experiences with Change - 2

How much stress is the organization currently experiencing?

- Are other changes currently in progress?
  - culture change efforts
  - reorganization
  - geographic displacements
  - process improvement or other quality initiatives
  - new software languages or tools
- Have there been changes in jobs and compensation?
- Has there been downsizing?

Hallway Conversation

Sally: *"Have you heard about this product line approach that management is pushing?"*

Fred: *"Don't worry. It'll go the way of all their other crusades – nowhere!"*

Sally: *"Ah, you're probably right. It's the same clowns, just a different circus."*

Sponsorship - 1

Sponsors authorize (executive sponsor) and/or reinforce the change (reinforcing sponsors)

The executive sponsor should

- set the vision
- “own” the change and communicate this ownership to all targets
- build a chain of reinforcing sponsors
- select and empower qualified change agents
- establish an infrastructure to manage the change
- commit resources to the change
- adjust the reward (punishment) system to encourage change
- model any desired changes in behavior
- communicate, communicate, communicate

Sponsorship - 2

The reinforcing sponsors should support the executive sponsor by

- helping the executive sponsor understand the “realities” that will help or hinder the change
- reinforcing the executive sponsor’s message
- modeling any desired changes in behavior
- reinforcing communications up and down the chain

Sponsorship: Discussion

Who is the executive sponsor in product line adoption?

Who is the reinforcing sponsor(s)?

Ways to Fail As a Sponsor

Show inconsistency between words and actions.

- Reward old behaviors.
- Punish new behaviors.

Set unrealistic expectations (There is no silver bullet).

Use a “delegate and forget” style.

Bail out when things don’t go smoothly.

Substitute buzz words for knowledge.

Expect everyone else to change.

In any significant change effort, the sponsors are almost always the ones who will have to change the most.

Sponsors must lead as well as manage.

In the CEO's Office

CEO to Product Line Manager:

"I funded this product line effort SIX months ago. Why don't I see a decrease in product development costs??"

Change Agents

Change agents implement the change effort as agents of the executive sponsor.

The executive sponsor must

- choose the right change agents
- place the change agency in the right spot within the organization
- provide the change agents with resources
- support the change agents in developing their individual change management skills
- be willing to be coached by the change agents in how to be an effective sponsor

Change Agent Skills

Effective change agents have

- a successful history with the organization
- technical credibility with sponsors and targets
- personal credibility with sponsors and targets
- good interpersonal skills
- good communication skills
- good project management skills
- resiliency in the face of difficulty
- the ability to “push back” when necessary
- the ability to keep touch with the line organization
- the personal discipline to model desired behaviors
- a willingness to share credit and celebrate successes
- an understanding of change management principles and techniques
- a working knowledge of the new technology

Product Line Kickoff Meeting

Product Line Manager:

"We are going to change our mode of operation for software development to a product line approach.

I recognize that you are a talented, successful group.

I believe that adopting a software product line approach for our business will make us even more successful.

Our product line adoption path will begin with a product line course, followed by an SEI Product Line Technical Probe, and then planning sessions that will involve many of you.

I expect your full commitment.

We will have ongoing, open review forums to discuss our progress and any issues you have."

Resistance

> "Resistance is something we do when we do not feel safe."
>
> *Henriette Klauaer*

People naturally resist change.

People will resist

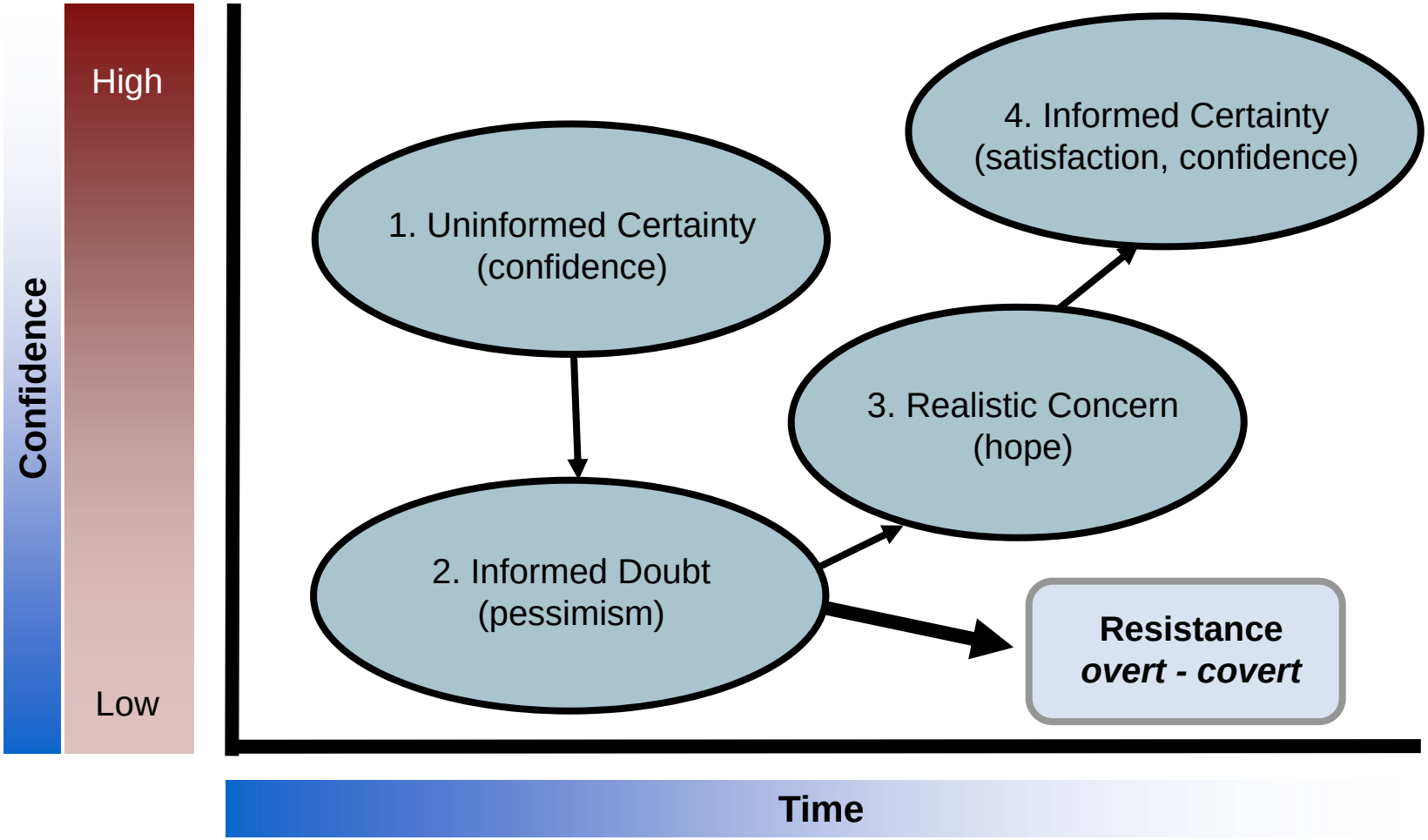
- the consequences of a technology change
- the implementation approach of a technology change
- the people responsible for the implementation
- any change at all

Basic premises

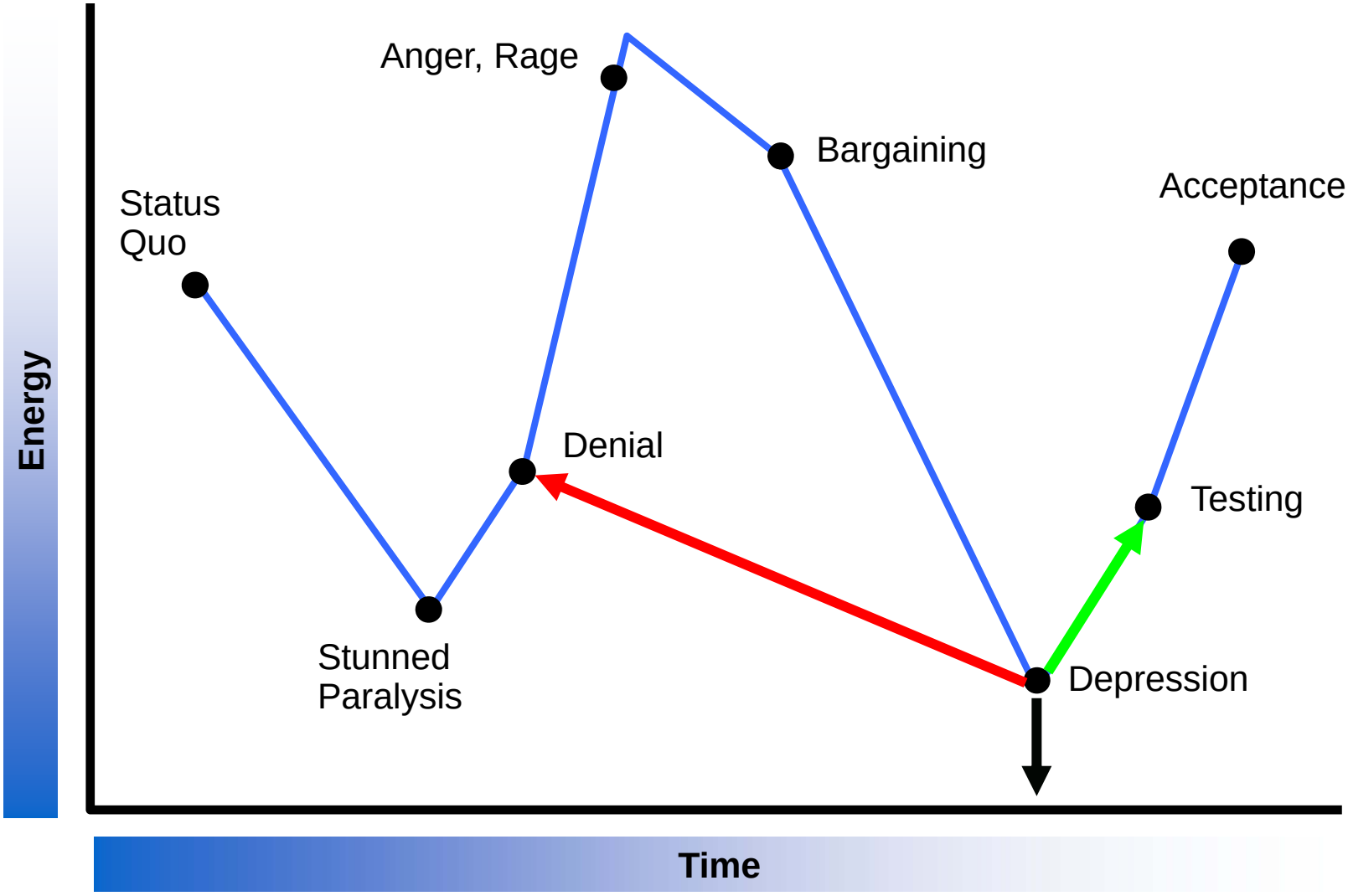
- You cannot suppress resistance.
- If you try to suppress resistance you will only drive it underground.
- Underground resistance manifests itself at the most inopportune times in the worst possible ways.

The change models shown earlier give some insights into resistance.

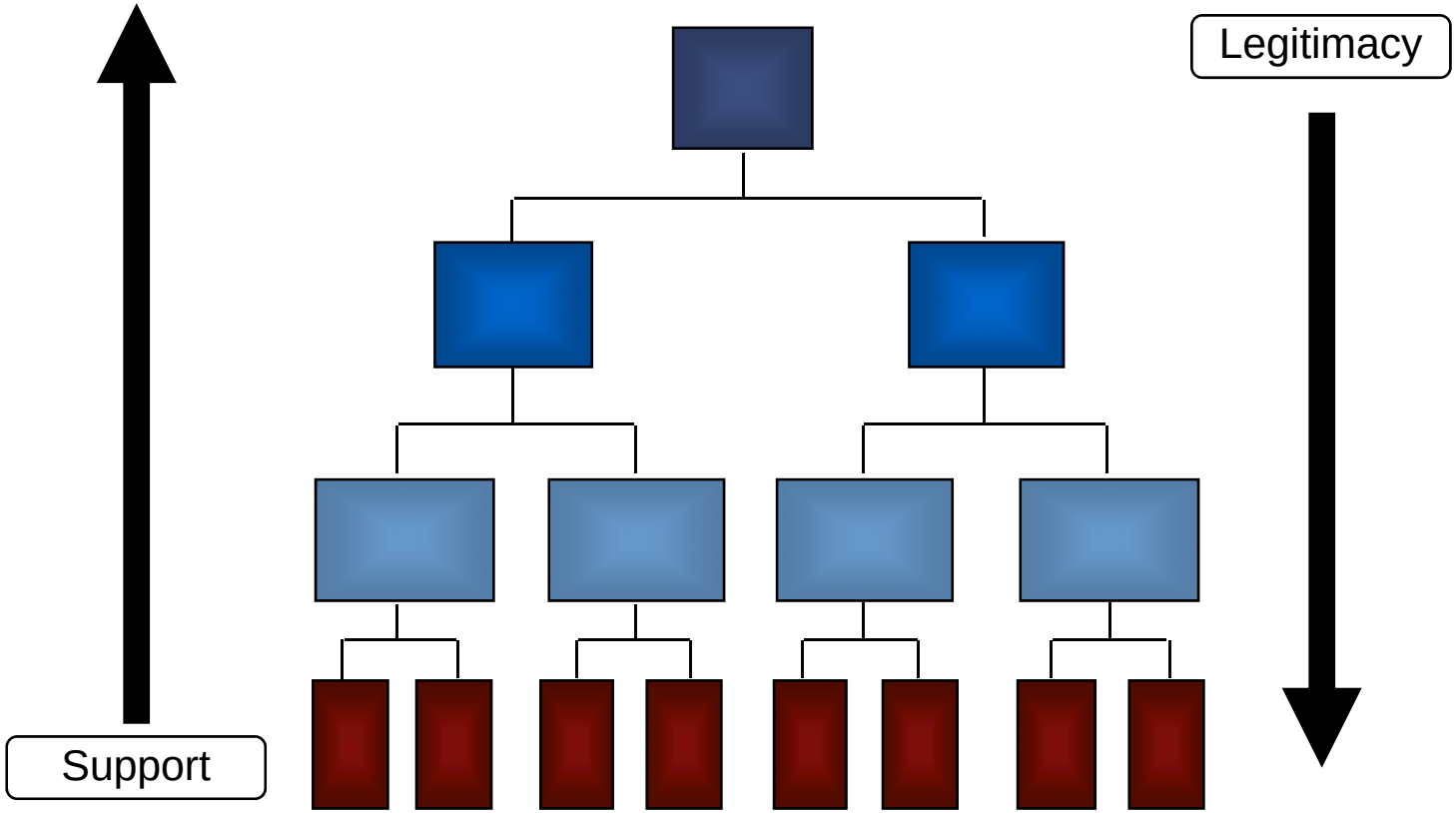
Positive Reaction to Change



Negative Reaction to Change



Building Support for Change



[courtesy J. Maher]

Random Reactions

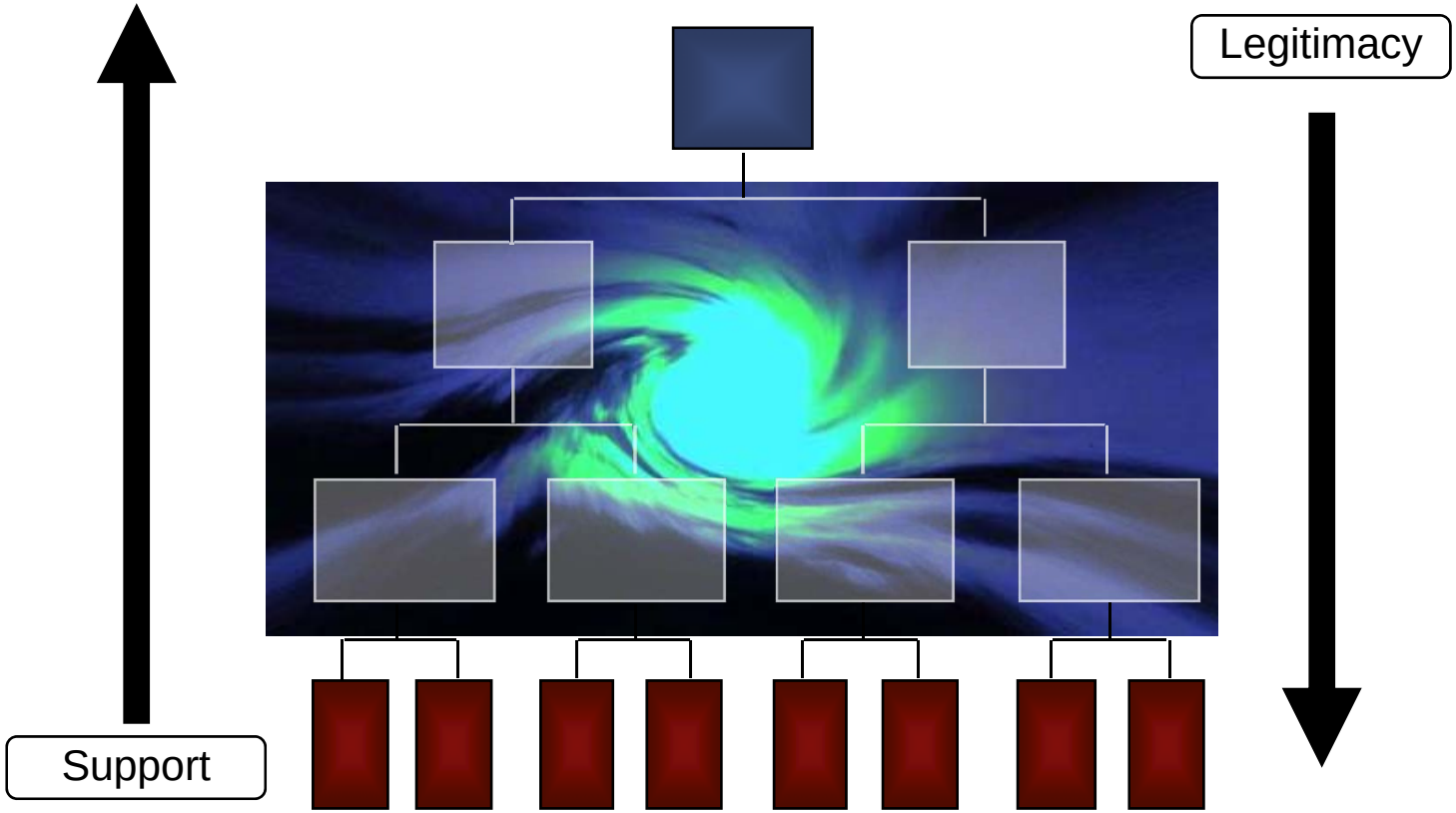
"I'll do this product line thing, but it's for sure I'll never use any core assets Sue developed."

"This product line approach is the most stupid idea management has come up with!"

"Well if this product line approach is so terrific, why aren't we seeing any improvements?"

"I see this product line stuff going away. I am just going to do my job the way I always have."

Middle Management “Black Hole”

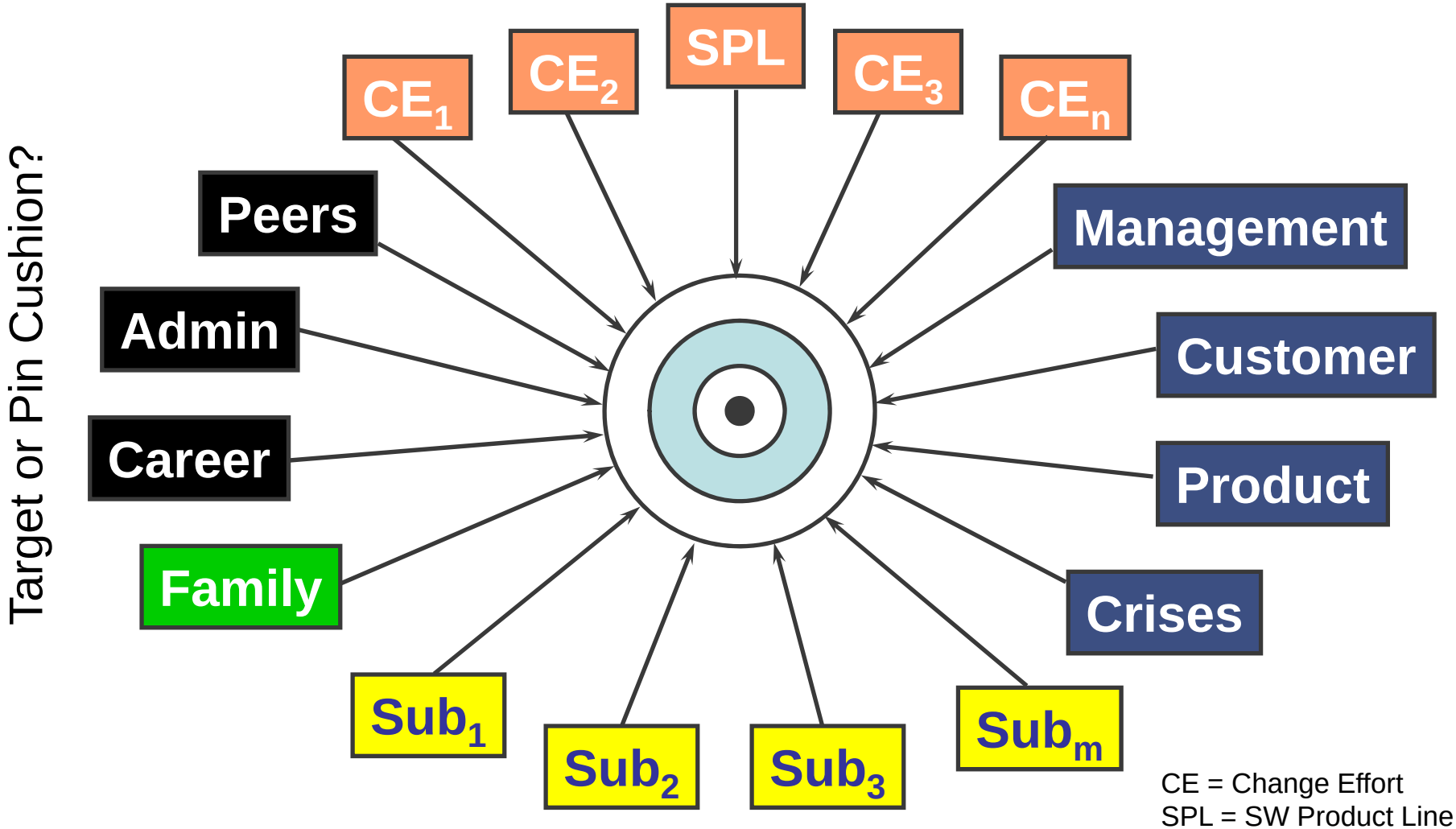


Middle Management Chant

"We will go out of business while we are getting to a product line approach.

I have products to get out the door!"

Middle Management Context



CE = Change Effort
SPL = SW Product Line
Sub = Subordinate

Chart based on one from Chuck Myers

Some Advice for Dealing With Resistance

Involve the targets of change in the change process so that they are "participants" rather than "victims."

Anticipate resistance and plan to address it.

Recognize that some resistance is healthy.

Create an environment where it is safe to surface resistance so that it may be dealt with openly.

Address the needs and problems of middle management.

Understand the roots of the resistance and find product line adoption strategies that will address them.

Organizational Culture

> "A fish discovers water last of all."
>
> *Chinese Proverb*

An operational definition

- Organizational culture is a pattern of shared basic assumptions that a group learned as it solved its problems of external adaptation and internal integration that has worked well enough to be considered valid.
- Therefore, these assumptions are taught to new members as the correct way to perceive, think, and feel in relation to these problems.

Source: Edgar Schein, Organizational Culture and Leadership

Some Elements of Culture

Behavior

- language, customs, traditions, rituals

Norms

- implicit values used to judge fairness and acceptability

Philosophy

- broad ideological principles that guide an organization

Rules

- implicit rules for getting along and advancing

Climate

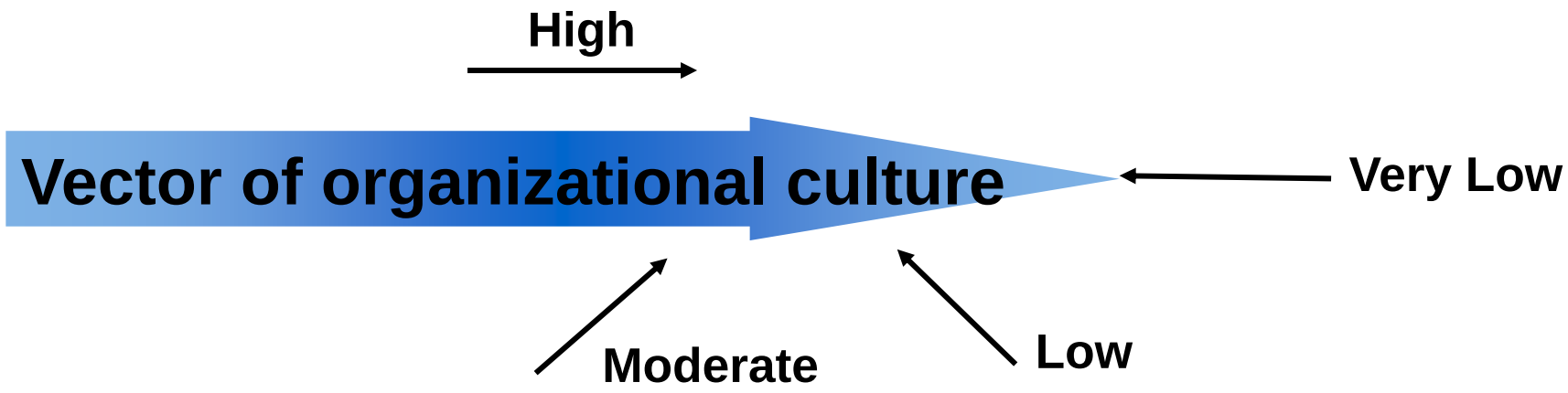
- the overall “feeling” about being a part of the organization

Identity

- ideas, feelings, and images that identify the group leading to a feeling of belonging or a desire to belong

Adapted from Edgar Schein, *Organizational Culture and Leadership*

Probability of Successful Change Related to Alignment with Culture



Key

↗	Vector of change effort
High, Low, etc.	Probability of success

Software Developer Chant

“We take great pride in developing each product tailor-made for every customer from start to finish.

Changing the software constantly to meet every customer desire is a way of life.”

Using Change Management Theory in Product Line Adoption

Understand the models of change and the insights and guidance they provide.

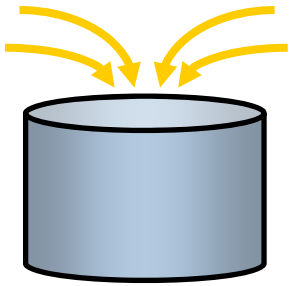
Consider the critical change factors as applied to your organization.

- Identify which are barriers
- Identify which are enablers

Identify specifically who will be affected by what specific changes and the potential impact.

Understand the roots of resistance and select a product line approach that helps address them.

Bring this knowledge to the table when building product line adoption plans.



Session Topics

Some basic concepts of change

Some change models

Some critical factors involved in organizational change

Exercise

Summary

Exercise

Refer to the exercise handout.

Session Topics

Some basic concepts of change

Some change models

Some critical factors involved in organizational change

Exercise

Summary

Summary - 1

Product line adoption involves both technical and non-technical change.

Change is difficult.

Non-technical change are usually more difficult than technical change.

Some critical factors involved in organizational change include

- prior experience with change
- sponsorship
- change agent skills
- resistance
- organizational culture

Summary - 2

There are change models that are useful when planning for and implementing the change associated with product line adoption.

Factors Influencing Adoption

Organizational Context

product line readiness

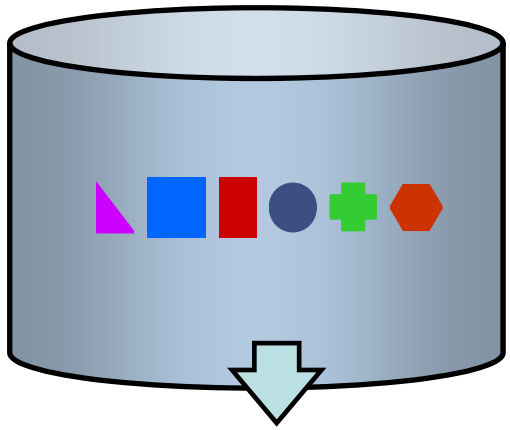
barriers

enablers

unique characteristics

culture

other ongoing activities



Product Line Adoption Plan

. . .

Product Line Action Plans

Adoption Support

The Framework

product line adoption roadmap

product line approaches

change models

change management mechanisms

planning process

Software Engineering Institute | Carnegie Mellon

Adopting Software Product Lines

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Part 2: Using Adoption Models
Module 5: The IDEAL Model

Session Outcomes

After this session participants should

- appreciate the usefulness of a phased change approach
- know the five phases of the IDEAL model
- know the general activities that are involved in each phase of IDEAL and how they relate to product line adoption
- know how to adapt the IDEAL model to a realistic situation
- be familiar with how Six Sigma can apply to product line adoption
- be familiar with how Lean Six Sigma can apply to product line adoption

Session Topics

Phased adoption and an overview of the IDEAL model

Details of the IDEAL model

Exercise

Summary

Why a Phased Approach to Change?

How do you eat a whale sandwich?

- one bite at a time

You don't need to phase in change if

- The scale or impact of the change is very small.
- Business considerations make a "big bang" approach necessary.
- You are certain everything will work the way you want it to the first time.
- You are sure you can identify and mitigate all the risks.
- You can afford to fail.
- You have the time and resources to try again if you fail.
- …

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.

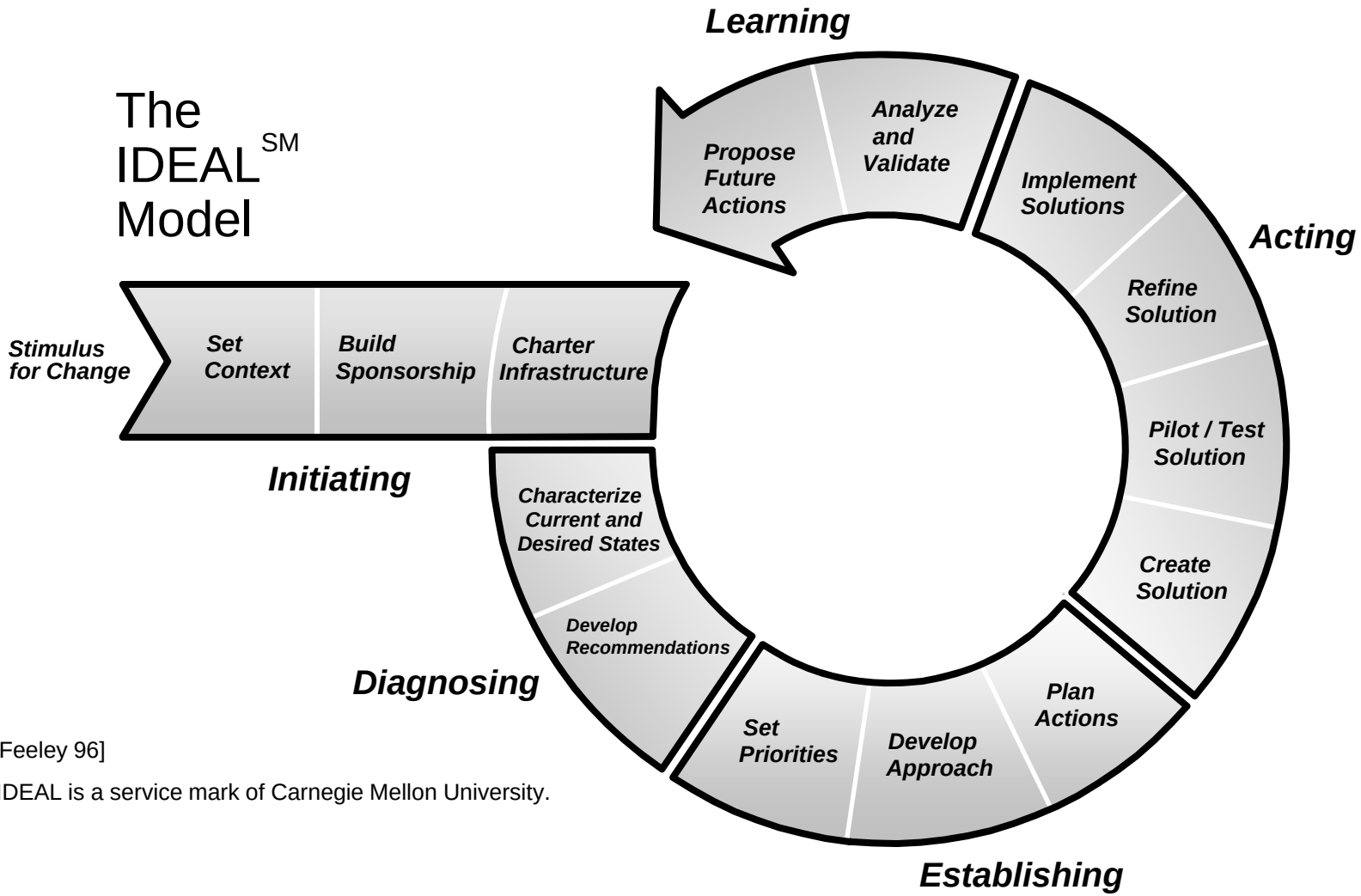
A popular approach: Plan Do Check Act.

A more formal approach: IDEAL model (process improvement)

Other approaches (with process improvement focus)

- Six Sigma
- Lean Six Sigma

IDEAL[SM]: An Improvement Approach



[McFeeley 96]

SM IDEAL is a service mark of Carnegie Mellon University.

"Getting Real" with IDEAL - 1

IDEAL was originally developed as a guide for process improvement and has proven to be a successful change guide.

IDEAL is readily adaptable to other changes.

……

But IDEAL is just a model. Realistic application requires judicious interpretation.

- Entry conditions differ among organizations.
- Iteration is necessary.
  - Mini-iterations within the phases is common.
  - Revisiting and reinforcing the phases is common.
- Each phase will usually require different emphasis depending on the overall iteration.
- Think of the model as an ordered checklist of things to consider.

“Getting Real” with IDEAL - 2

Unless you are already acquainted with the IDEAL model, your first time through will be intuitive (and incomplete).

- Without formal instruction you’ve probably already done an initial pass at some IDEAL activities.
- You will probably need to do some “back filling” to address things that may have initially been overlooked.

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.

A Note on Our Presentation of the IDEAL Model

While we will present each phase of the IDEAL model in a single pass, remember that iteration is inherent in the model.

When developing (and executing) a real plan based on the model, not all the examples given for a particular phase will occur during the same iteration.

Session Topics

Phased adoption and an overview of the IDEAL model

Details of the IDEAL model

Exercise

Summary

Entry Criteria for Product Lines

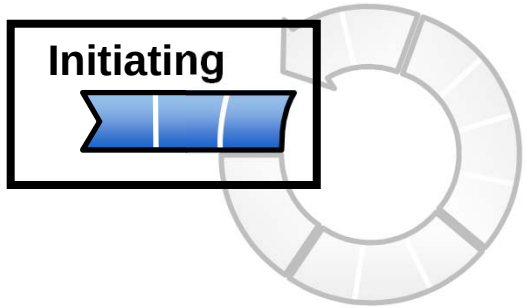
Is there an overall fit for a software product line approach?

- 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?

Initiating: Forming Commitment - 1

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

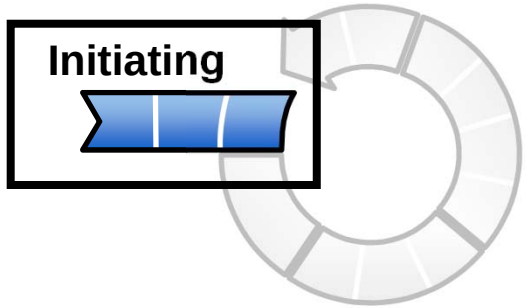


Initiating: Forming Commitment - 2

Management has the job of reinforcing and communicating the foundations established in this phase throughout the other IDEAL phases.

Skipping or neglecting this phase will almost certainly result in a less effective implementation or failure.

If you use early small-scale cycles to “test the water,” be sure to revisit this phase and scale up before proceeding.



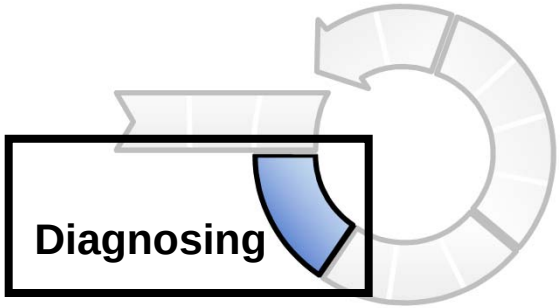
Discussion Question

What ideas from the Change Models session might be applicable in the Initiating Phase?

Diagnosing: Checking Product Line Conditions - 1

Diagnostics you might perform

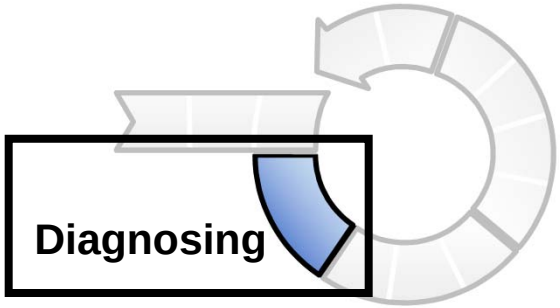
- Evaluate the business and technical viability of the product line opportunity.
- Examine the product line context.
  - market
  - organization
  - business unit
  - Individuals
- Identify organizational strengths and weaknesses related to change implementation.



Diagnosing: Checking Product Line Conditions - 2

- Analyze the organization against the relevant 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.



Diagnostic Instruments

A **good** diagnostic instrument

- is repeatable
- is objective and credible
  - mechanisms to remove bias; separate fact from opinion
  - mechanisms for validation and confirmation
- is comprehensive
  - examines all necessary product line practices
  - examines all levels within the organization
- has reliable and thorough data gathering techniques
  - objective and subjective data
  - artifacts and people
- has reliable data reduction and reporting techniques
  - expert analysis
- produces actionable results
- explores the appropriate level of detail

Product Line Diagnostic Instruments

- SEI Product Line Quick Look℠ (PLQL℠)
- SEI Product Line Technical Probe℠ (PLTP℠)
- Bosch Product Line Potential Analysis*
- European Union ITEA (Information Technology for European Advancement) BAPO (Business, Architecture, Process, Organization) evaluation*
- Ecosystem – Strategy – Architecture – Organizing (ESAO)

- Others?

* *Software Product Lines: Third International Software Product Line Conference, Boston, MA, August, 2004*

SEI Product Line Quick Look (PLQL)

The Product Line Quick Look (PLQL) is a method for gathering initial information relative to an organization's readiness to adopt or ability to succeed with a software product line approach.

The PLQL provides a high-level framing of the organization and its product line effort.

The information gathered can then be used to plan more specific activities needed to progress in product line adoption.

The PLQL is a lightweight, short intervention, and a good first step for an organization wishing to "test the waters."

What Is An SEI Product Line Technical Probe (PLTP)?

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

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

PLTP Outcomes

Set of findings that portray organizational

- strengths
- challenges

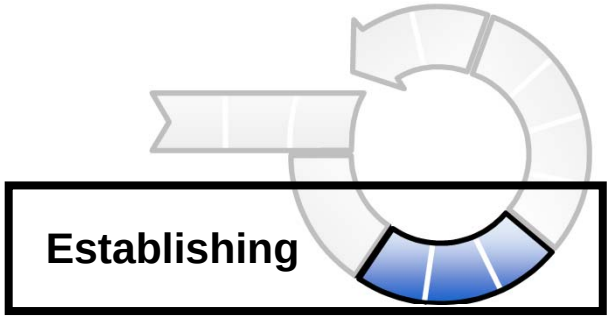
with regard to a product line approach

Findings can be used to develop an adoption plan or action plan with the goal of making the organization more capable of achieving product line success.

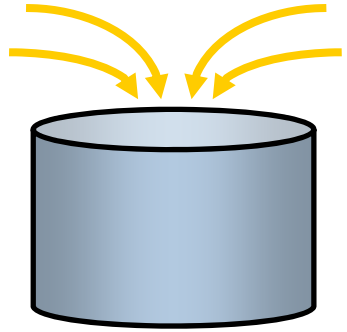
Establishing: Planning the Product Line Adoption

While considering the organizational context and adoption support:

- 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



Considerations



Organizational context

- product line readiness
- findings from a product line diagnostic
- barriers
- enablers
- culture
- unique characteristics
- other ongoing activities

Adoption Support

- Framework for Software Product Line Practice
- Product Line Adoption Roadmap -- product line practice patterns
- Case studies
- Change models and mechanisms
  - general change guidelines
  - specific change considerations

Product Line Plans

Planning is the primary subject of two of the 29 practice areas in the Framework.

- "Technical Planning"
- "Organizational Planning"

Planning is also a key aspect of the "Launching and Institutionalizing" practice area.

While the planning principles are similar for all types of product line plans, the purposes of the plans are different.

The emphasis here is on product line adoption planning.

Plans Supporting Software Product Line Adoption

Business Plan

- A plan that lays out overall company strategies to achieve business goals.
- Such a plan might specify adopting a software product line approach for a particular set of products.

Product Line Adoption Plan

- A plan that describes how product line practices will be rolled out across the organization.

Product Line Action Plan

- A plan to address a specific portion of a product line adoption plan.

Product Line Adoption Plans

Depending on the starting point of the organization, the **Product Line Adoption Plan** may provide for

- education and training
- the definition of processes
- the initiation of practices
- the selection and implementation of pilots
- the engineering of a product line

The Product Line Adoption Plan orchestrates the chosen approach for the product line and the adoption strategy.

Lower level action plans may be developed to augment and provide more detail for needed activities.

- for example, determine production method or develop training

Plans and the Planning Process

Plans are essential to a predictable, manageable, coordinated product line effort.

- Good plans set the stage for a successful product line adoption.
- Poor plans set the stage for an unsuccessful adoption.
- Effective plans are easier to create using an effective planning process.

Plans and planning expertise are also critical when operating the product line.

An effective, repeatable planning process should involve validation, iteration, stakeholder consensus, and adaptation so that the plan fits the plan purpose and organizational environment.

- We will learn such a planning process later in the course.

Using Pilots - 1

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).

Using Pilots - 2

Ways you might use pilots to support product line adoption

- Implement a limited set of products in the envisioned product line.
- Implement a product line approach in a limited portion of the organization.
- Implement a limited set of product line practices
  - across the organization
  - within a limited part of the organization.
- …

Note that if you use pilots as a strategy, this implies additional iterations of the IDEAL cycle for complete product line adoption.

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.

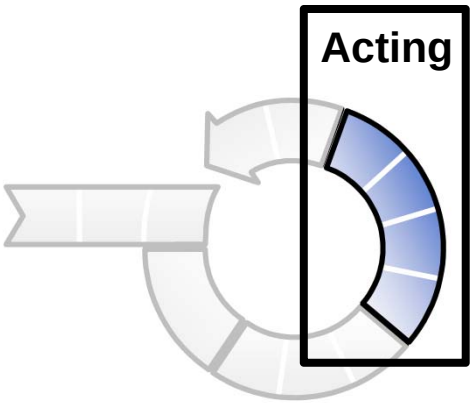
Change the plans as necessary.

Manage risks associated with the plan.

See any number of guides to project management

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

*Project Management Institute. *A Guide to the Project Management Body of Knowledge*. http://www.pmi.org/prod/groups/public/documents/info/pp_pmbok2000welcome.asp

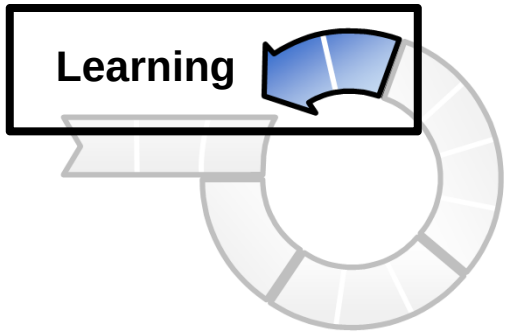


Learning: Tuning and Improvement - 1

Consolidate data and lessons learned.

Measure results against established goals.

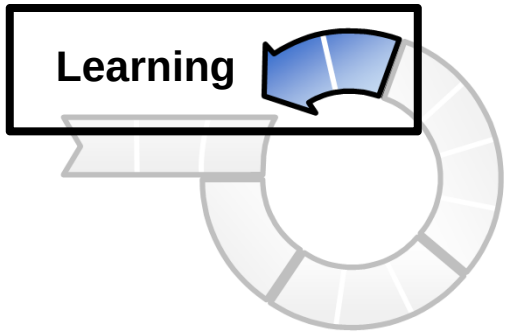
Modify products, processes, and organizational structures to reflect lessons learned and to take advantage of potential optimizations.



Learning: Tuning and Improvement - 2

Consider the Learning Phase early. If you are to learn from the approach, you have to approach an IDEAL iteration as an experiment.

- Design the experiment (What do you want to learn?).
- Design your data collection strategy.
- Collect your data as you go.
- All this has to take place prior to the Learning Phase.



Session Topics

Phased adoption and an overview of the IDEAL model

Details of the IDEAL model

Exercise

Summary

Exercise

See the exercise handout.

Session Topics

Phased adoption and an overview of the IDEAL model

Details of the IDEAL model

Exercise

Summary

What the IDEAL Model Is

The IDEAL model is a general model for guiding change with its roots in software process improvement.

The IDEAL model emphasizes

- how to sequence change activities
- an iterative approach to change
- establishment of appropriate structures and activities to manage change
- mechanisms for learning and improving how change is accomplished
- some activities that are specific to process improvement

What the IDEAL Model Is Not

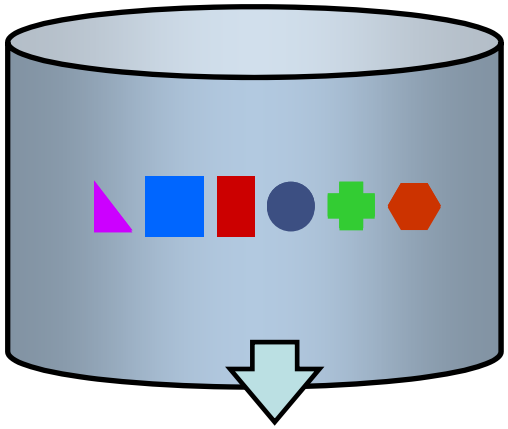
The IDEAL model is **not** specific to software product line adoption. From this perspective, the IDEAL model lacks the following types of information and guidance.

- What are the effective product line practices?
- What are the dependencies among product line practices?
- What roles, skills, and structures are appropriate for launching and institutionalizing a product line?
- What are the important artifacts and deliverables for a product line approach?

Factors Influencing Adoption

Organizational Context

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



Product Line Adoption Plan

. . .

Product Line Action Plans

Adoption Support

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

Software Engineering Institute | Carnegie Mellon

Adopting Software Product Lines

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Part 2: Using Adoption Models
Module 6: Using Adoption Factory with IDEAL

Session Outcomes

After this session participants should

- understand how to use the ***Adoption Factory*** pattern as a generic product line adoption roadmap
- be able to apply the ***Adoption Factory*** pattern with a change model

Session Topics

Adoption Factory use

Practical considerations

Adoption Factory and change models

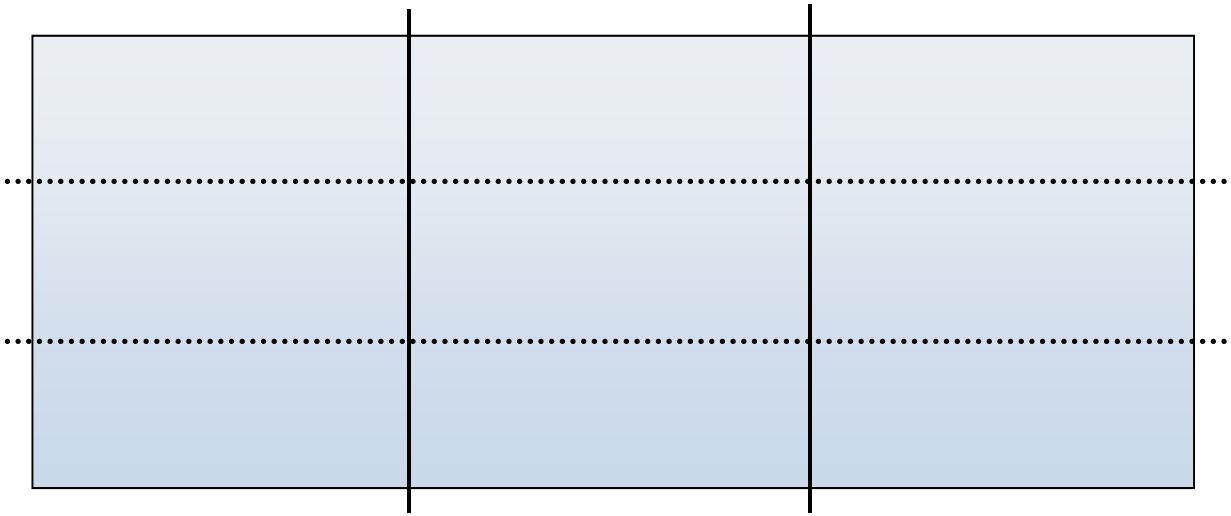
Adoption Factory and product line adoption plans

Exercise

Summary

What the Adoption Factory Pattern Provides - 1

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.



What the Adoption Factory Pattern Provides - 2

When used with its associated views, the ***Adoption Factory*** pattern provides

- a decomposition of the change that needs to occur
- a natural, overall organizing mechanism
- a detailed list of the new or tailored practice areas involved
- guidance as to who needs to be involved
- guidance as to what talent may need to be acquired
- a checklist of typical deliverables that should be generated

What the Adoption Factory Pattern Does Not Provide

The ***Adoption Factory*** pattern does not provide

- an absolute linear sequencing
- organization-specific guidance
- the definition of the practice areas
- specific practices
- prescription for a particular product line approach (proactive, reactive, incremental)
- change management mechanisms
- a product line adoption plan

Session Topics

Adoption Factory use

Practical considerations

- **pattern and practice area sequencing**
- **organization-specific guidance**
- **practice area definition**
- **specific practices**
- **product line approaches**

Adoption Factory and change models

Adoption Factory and product line adoption plans

Exercise

Summary

Practical Considerations

Everything not provided by the *Adoption Factory* pattern still has to be addressed.

Understanding how to address these issues and the support *Adoption Factory* can provide is important.

We will see how *Adoption Factory* relates to the other inputs and outputs of the “kettle.”

Pattern and Practice Area Sequencing - 1

The Phases and Focus Areas View provides an ordering.

- The ordering provided by the phases and focus areas and by the patterns is not strictly linear. There is no cement wall between phases.

The product line practice patterns group practice areas.

- The relationships between practice areas is not strictly linear.

There is **constant iteration** in any product line effort and hence in any product line adoption effort.

Pattern and Practice Area Sequencing - 2

Guidelines:

Use the ***Adoption Factory*** pattern and its associated views as an overall layout of **what** needs to be accomplished.

Plan to master the practice areas in a continuous way that begins at the phase where each practice area first appears.

Use organization-specific information to focus first on those practice areas that have the most immediate impact.

Look across the phase horizon, and, where it makes sense, begin early to prepare for those activities presenting the greatest challenge.

During the adoption process, iterate back and address practice areas that were initially covered lightly.

Pattern and Practice Area Sequencing - 3

Examples:

1. Scoping, which is part of the *What to Build* pattern, is central to the Establish Context Phase.
   - Begin scoping early in the product line effort.
   - Continue scoping throughout the rest of the product line adoption process (and life cycle).

Pattern and Practice Area Sequencing - 4

Examples (continued):

2. Component development, which is part of the *Product Parts* pattern, is central to the Establish Production Capability Phase.

- Delay component development until at least after scoping is complete.
- If an organization dove head first into component development, it should go back and address the practice areas in the What to Build pattern and then to the “Requirements Engineering,” “Architecture Definition,” and “Architecture Evaluation” practice areas, which all inform the “Component Development” practice area in the *Product Parts* pattern.

Pattern and Practice Area Sequencing - 5

Examples (continued):

3. If an organization will rely heavily on legacy assets in its maiden product line effort

- Begin the inventorying practices from the “Mining Existing Assets” practice area in conjunction with scoping during the Establish Context Phase.
- Delay detailed analysis on suitability of assets until after some work on “Architecture Development.”

Organization-Specific Guidance

Appropriate use of the *Adoption Factory* pattern assumes organizational insight regarding

- product line readiness
- related current activities (architecture-centric development, reengineering, process improvement, product alignment, etc.)

This insight should be acquired through a combination of informal and formal mechanisms.

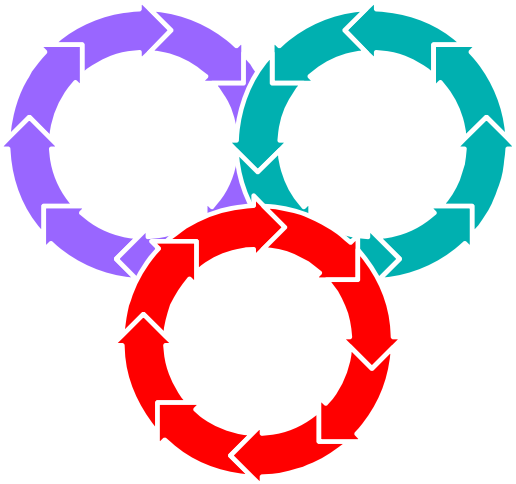
- historical knowledge
- product line diagnostic
- organizational survey

Definition of Practice Areas

The practice areas that constitute the Practice Areas view of the *Adoption Factory* pattern are defined in the Framework for Software Product Line Practice.

Use the Framework as a reference model to learn general information about the practice area as well as

- aspects that are peculiar to product lines
- its application to core asset development
- its application to product development
- example practices
- associated risks
- further reading



Example Practices

Example practices for each practice area

- are described in the Framework
- should be chosen based on organization-specific culture, skills, needs, and fit

Neither the Framework nor the *Adoption Factory* pattern specify a step-by-step methodology.

Any newly adopted practice that gets folded into the product line adoption has its own set of change issues.

- Choose practices based on business goals, product line appropriateness, skills required.
- Keep in mind cultural barriers and enablers.

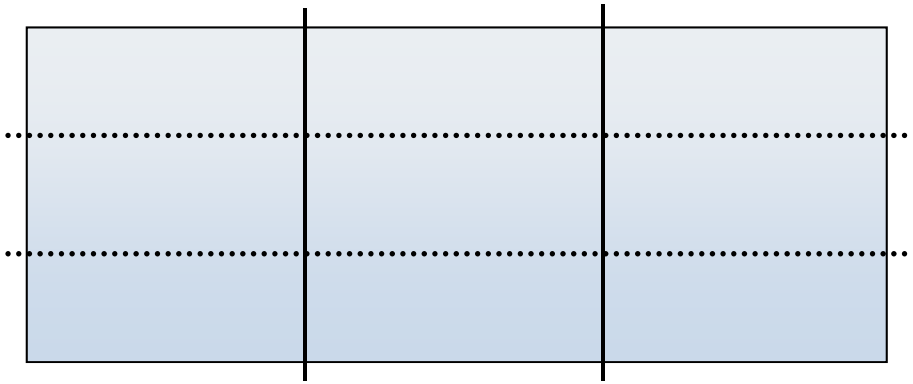
Different Product Line Approaches - 1

The *Adoption Factory* pattern does not provide guidance for selecting a product line approach.

However, it applies regardless of the approach chosen.

Proactive Approach:

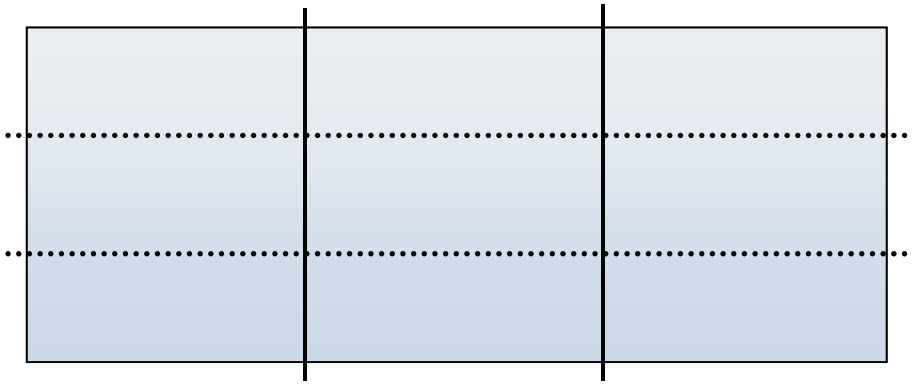
- Mapping to the *Adoption Factory* pattern is direct according to the ordering by phases.



Different Product Line Approaches - 2

Reactive Approach:

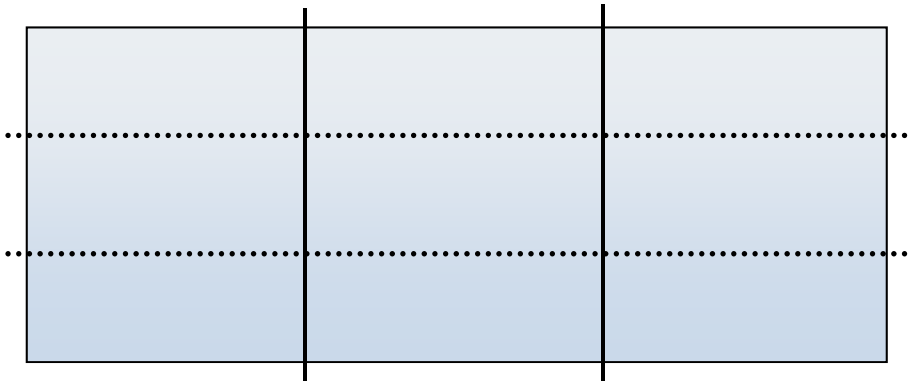
- The Establish Context Phase happens in an informal, micro sense before the first product(s) are built as single systems.
- The Establish Context Phase should be done in a deliberate way once the decision is made to extract or reengineer a core asset base from the product(s).



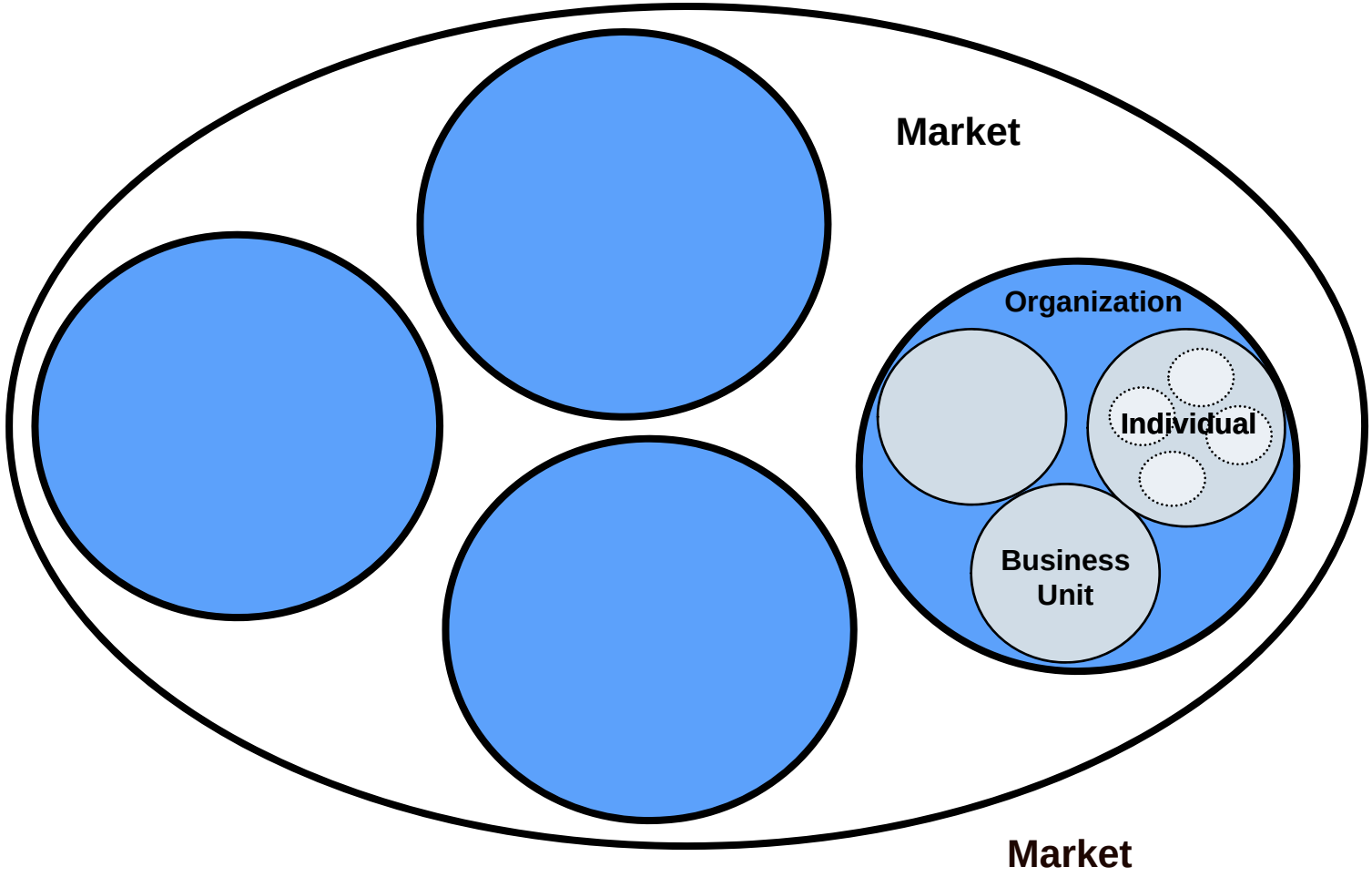
Different Product Line Approaches - 3

Incremental Approach:

- Begin with the Establish Context Phase.
- There is an explicit planned repetition of the phase pair Establish Production Capability and Operate Product Line mapped to the planned incremental strategy.
- Patterns in the Establish Context Phase are revisited lightly before each increment.



Factors to Consider When Choosing a Product Line Approach - 1



Factors to Consider When Choosing a Product Line Approach - 2

Market

- How large is the market volume?
- How mature is the market?
- Do open standards exist for this market?
- How safety-critical is the software in the products that constitute this market?
- Are there clear market leaders?

Factors to Consider When Choosing a Product Line Approach - 3

Organization

- What is the organization's current position in the market?
- How high is the probability that the organization will become a market leader?
- How closely is the organization connected with its customers?
- How much control does the organization have of the product specification?
- How likely is it that the organization will sell the asset base as well as products built from the asset base?
- How high is the motivation to use a product line approach?

Factors to Consider When Choosing a Product Line Approach - 4

Organization (continued)

- How stable is the funding source for the core asset base development and evolution?
- How strong is the high-level management commitment for a product line approach?

Business Unit

- How much experience has the business unit had with products in the targeted product line?
- How many high quality legacy assets does the business unit have for this product line?
- What is the current degree of process discipline within the business unit?
- How much domain-specific talent already exists in the business unit?

Factors to Consider When Choosing a Product Line Approach - 5

Business Unit (continued)

- How keen is the software engineering expertise, in particular the software architecture skills?
- How deeply engrained is the single-system mentality within the business unit?
- How dependent is the business unit on proprietary tools built and used to support a single-system approach?
- How clear is the current business strategy and product strategy?
- How equipped is the business unit to perform activities in the 29 practice areas?
- What is the availability of talent that will be needed to compensate for expertise shortages?

Factors to Consider When Choosing a Product Line Approach - 6

Individual

- Do the individuals have the necessary (additional) capacity to "learn" product line engineering?
- Do the individuals have the necessary expertise?
- If not, how will they become experts?
- How motivated are the individuals?
- Are they willing to learn and apply product line methods?
- Will the individuals benefit from product line engineering?

Guidelines for Using the Factors

Use the questions for each one of the circles of influence to examine the organizational context for product line adoption.

The outer circles provide resources for and put constraints on the inner circles.

The goals of the inner circles depend on a characterization of the outer circles.

Begin with the examination of the market and work inward to the individual.

Choose an adoption approach that best fits the organizational characterization based on the analysis.

Ideas from Buhne, Chastek, Kakola, Northrop, and Thiel. “Exploring the Context of Product Line Adoption.” *Proceedings of the 5th International Workshop on Software Product Family Engineering. Berlin, Germany: Springer-Verlag, 2003.*

Session Topics

Adoption Factory use

Practical considerations

Adoption Factory and change models

Adoption Factory and product line adoption plans

Exercise

Summary

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.

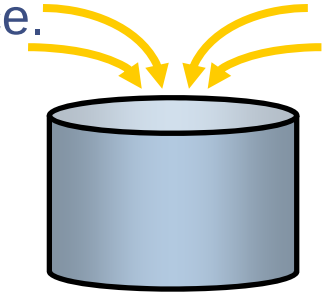
- Adoption Factory lacks change management mechanisms and guidance.

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.

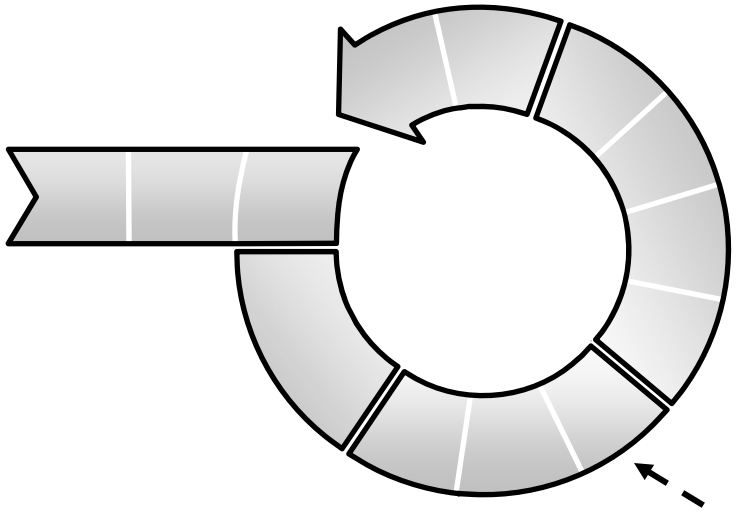
In particular, the **IDEAL Model** is a general model for guiding change.

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



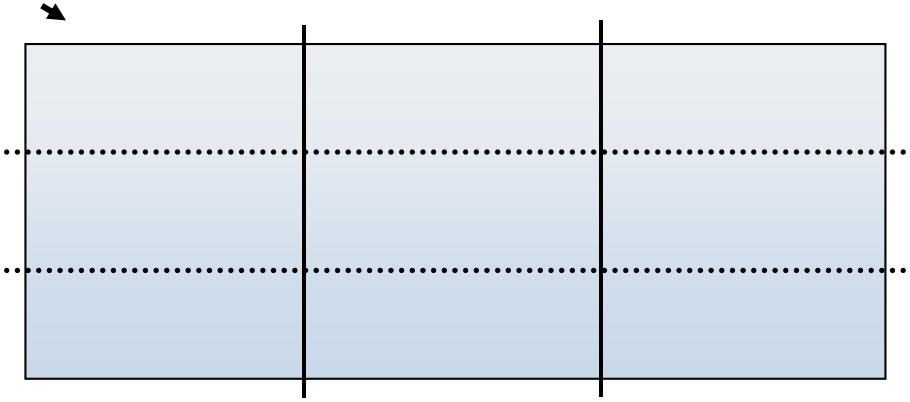
To be used successfully both need to be informed by relevant organization-specific information.

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.



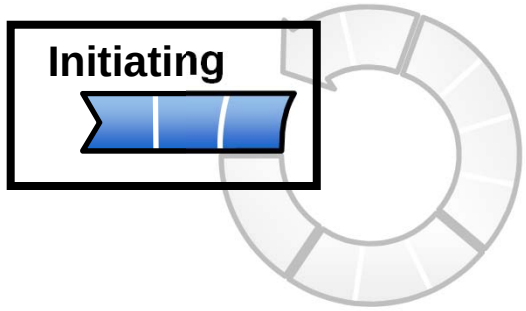
Adoption Factory and IDEAL Phases - 1

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.

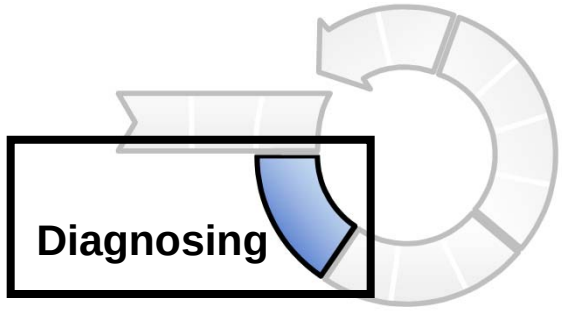
You can use the associated roles to guide staffing and management.



Adoption Factory and IDEAL Phases - 2

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.*

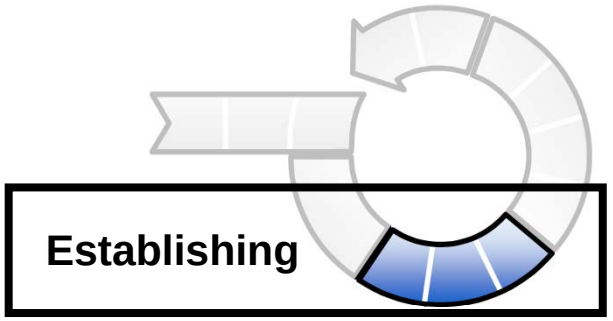


Adoption Factory and IDEAL Phases - 3

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.

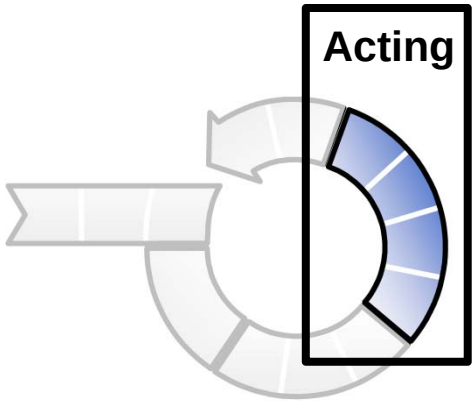


Adoption Factory and IDEAL Phases - 4

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.

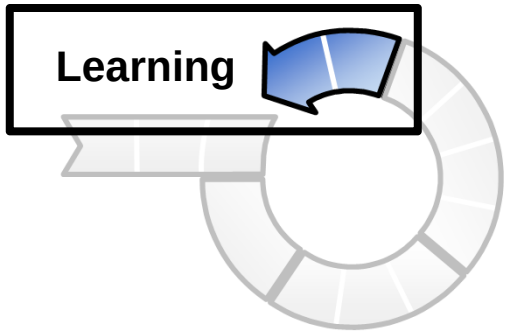


Adoption Factory and IDEAL Phases - 5

Learning:

You can

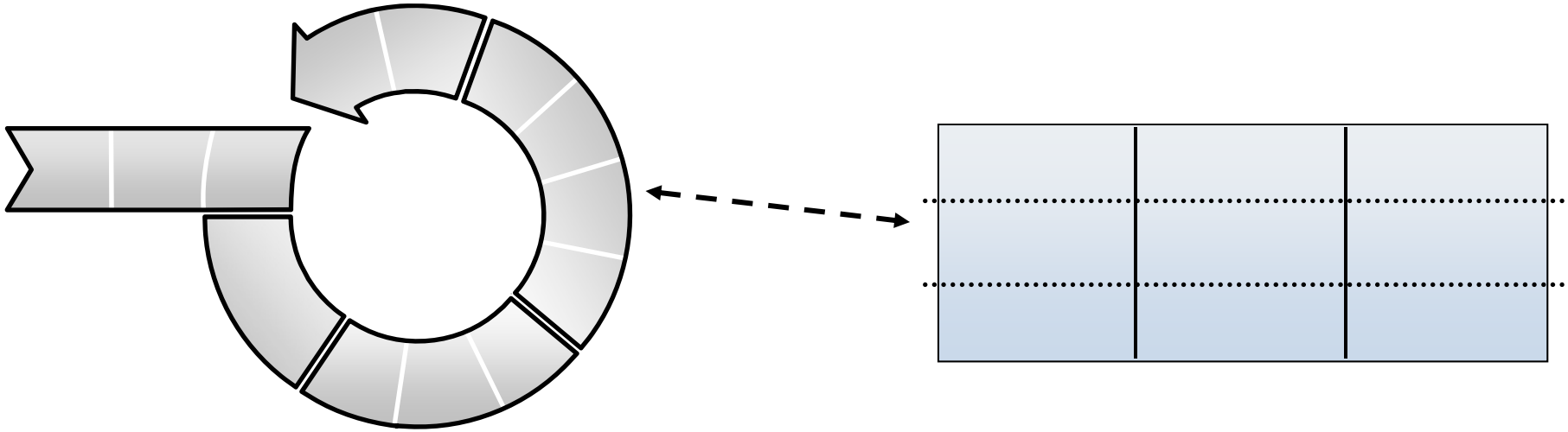
- collect data and lessons learned in each phase of the ***Adoption Factory*** pattern as specified by the “Measurement and Tracking” practice area
- analyze results against established goals
- iterate through the pattern phases and focus on different practice areas, modify products, processes, and organizational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



Practical Dual Use - 1

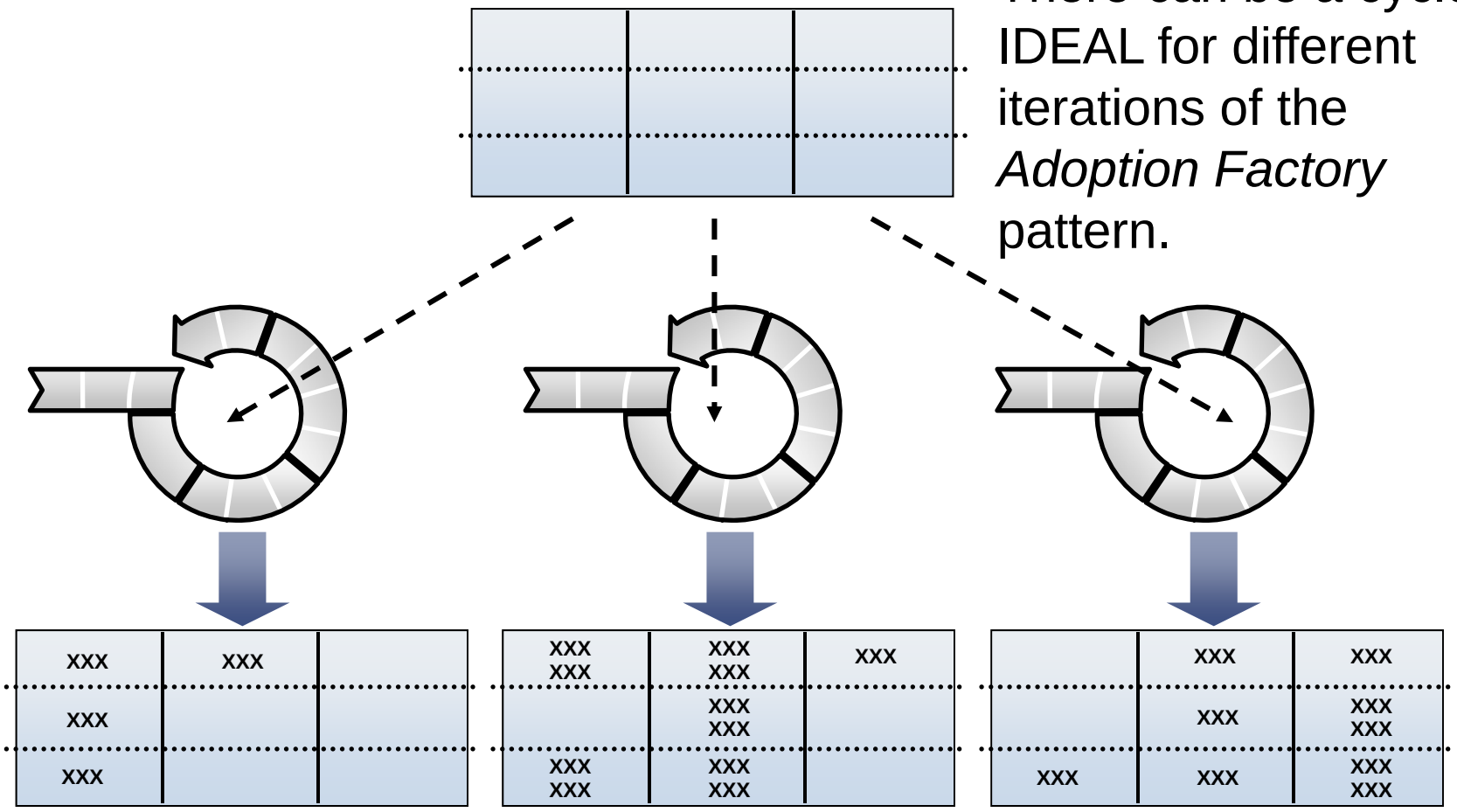
Because both the IDEAL model and the *Adoption Factory* pattern are meant to be applied iteratively based on organization-specific considerations, in practice there will be cross pollination of the two in any number of ways.

- IDEAL and *Adoption Factory* together guide the overall, grand product line adoption plan.



Practical Dual Use - 2

There can be a cycle of IDEAL for different iterations of the *Adoption Factory* pattern.



Practical Dual Use - 3

There are any other number of combinations of the *Adoption Factory* pattern and the IDEAL model.

In combining the two roadmaps always bear in mind that

- IDEAL lays out the phases for change introduction.
- Adoption Factory lays out the changes needed for product line adoption.
- Both need to be informed by organization-specific considerations.

Session Topics

Adoption Factory use

Practical considerations

Adoption Factory and change models

Adoption Factory and product line adoption plans

Exercise

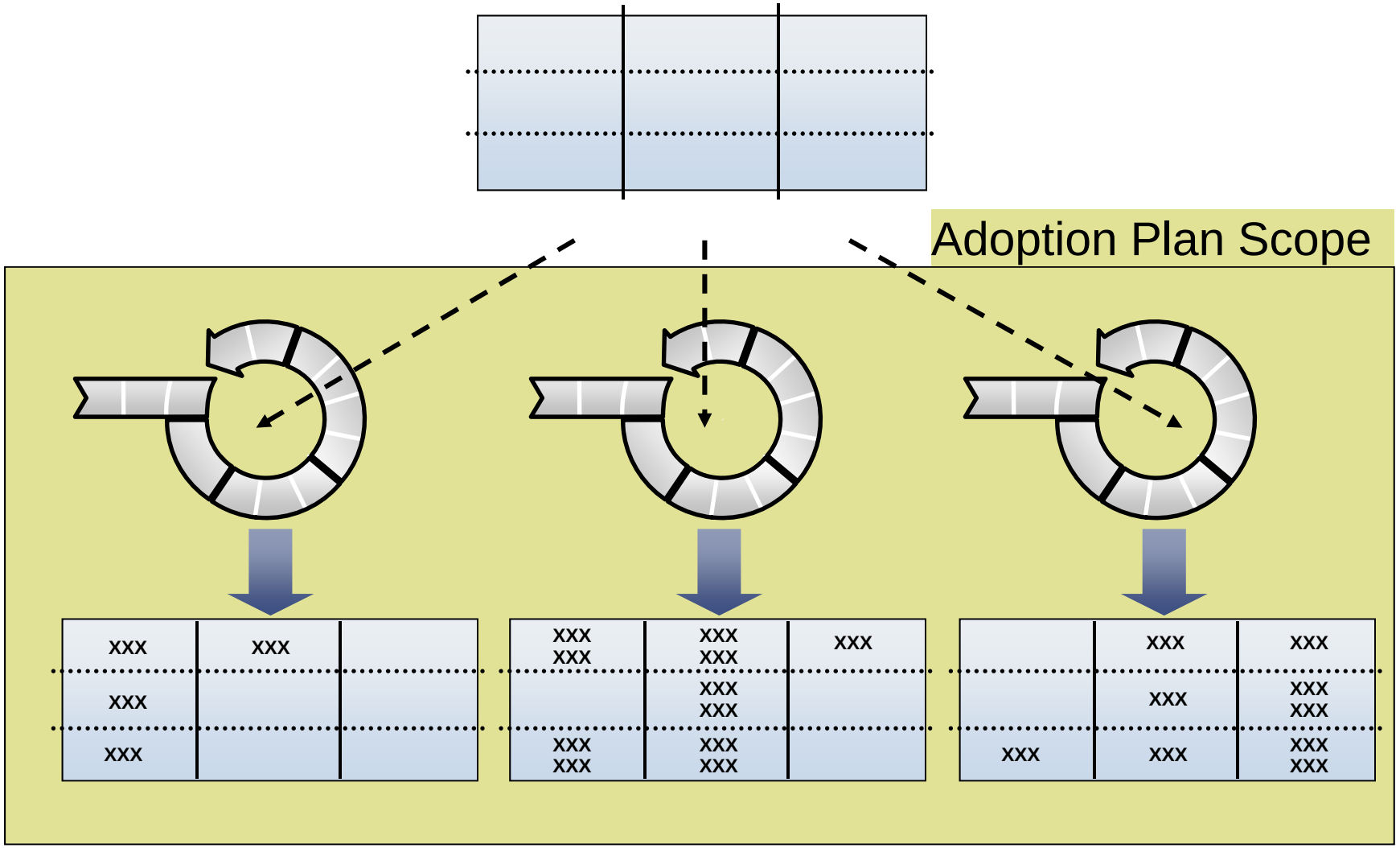
Summary

Product Line Adoption Plans

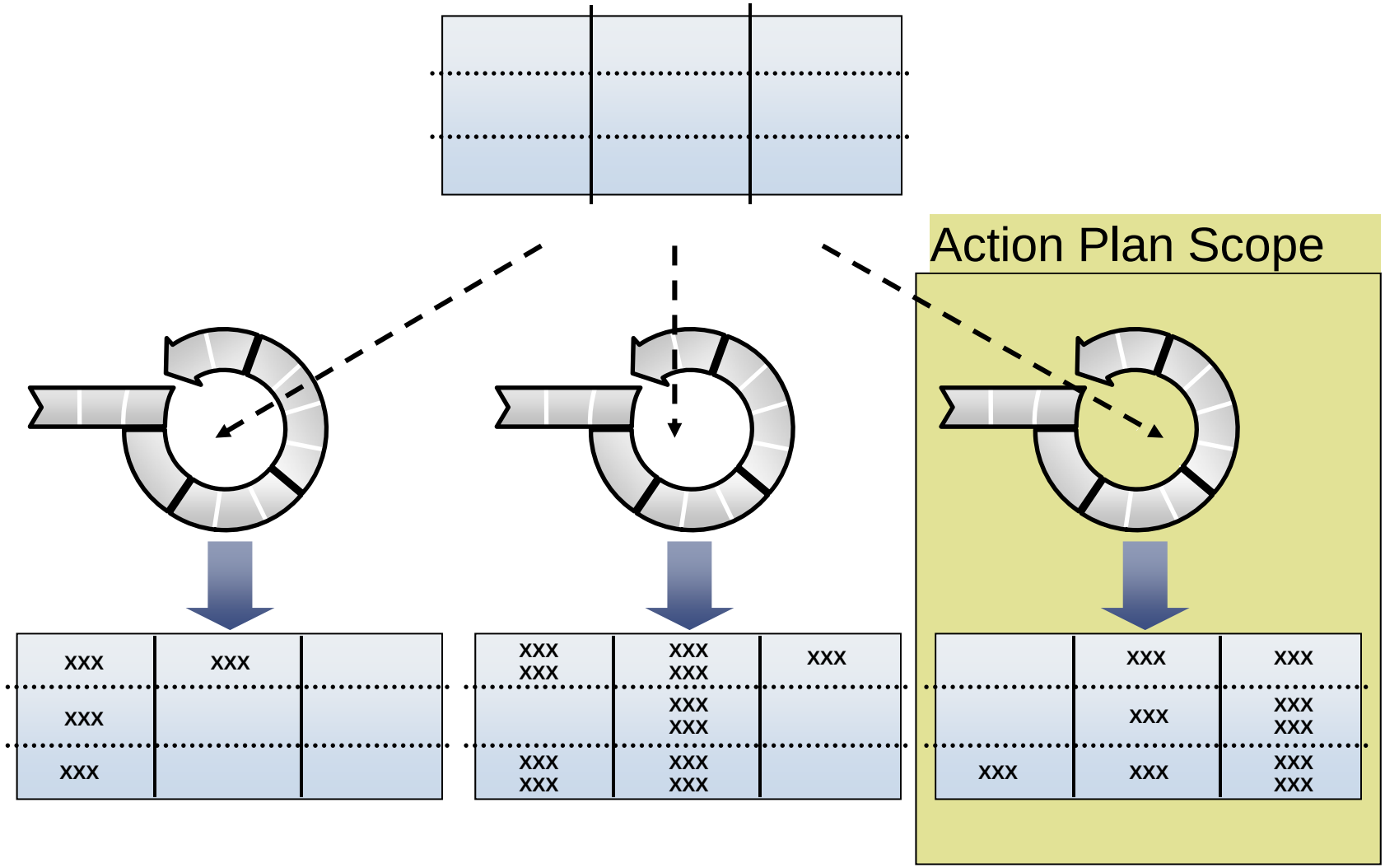
The Adoption Factory is not a product line adoption plan but it supports the development of product line adoption plans.

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

Plans, Adoption Factory, and IDEAL - 1



Plans, Adoption Factory, and IDEAL - 2



Session Topics

Adoption Factory use

Practical considerations

Adoption Factory and change models

Adoption Factory and product line adoption plans

Exercise

Summary

Exercise

Refer to the exercise handout.

Session Topics

Adoption Factory use

Practical considerations

Adoption Factory and change models

Adoption Factory and product line adoption plans

Exercise

Summary

Summary

The *Adoption Factory* pattern and its six views provide valuable information regarding what needs to happen in product line adoption as well as the ordering among the activities.

In practical use, the *Adoption Factory* pattern needs to be complemented by organization-specific information, and other product line and change management guidance.

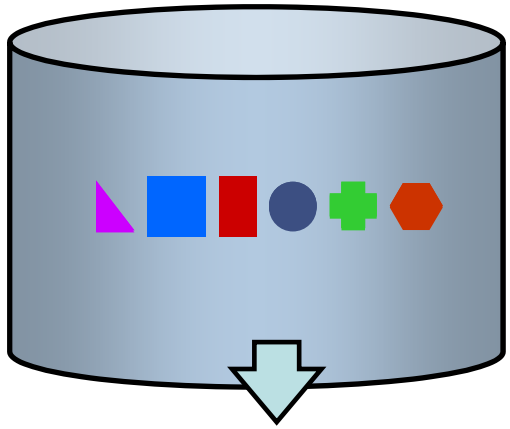
The *Adoption Factory* pattern and change models such as the IDEAL model can be combined in an assortment of ways to support product line adoption.

The *Adoption Factory* pattern is one very important input to product line adoption planning.

Factors Influencing Adoption

Organizational Context

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



Product Line Adoption Plan

. . .

Product Line Action Plans

Adoption Support

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

SEI Use of Adoption Factory

We use the ***Adoption Factory*** pattern routinely in our product line diagnostic instruments

- PLQL
- PLTP

and in our product line planning workshops.

Organizations report the ***Adoption Factory*** pattern

- is extremely useful in understanding product line status, in planning, and in implementing their product line adoption efforts
- provides a useful vocabulary and abstraction
- keeps even geographically distributed organizational counterparts on the same wavelength

Software Engineering Institute | Carnegie Mellon

Adopting Software Product Lines

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Part 3: Adoption Planning
Module 7: Plans and Planning

Session Outcomes

After this session participants should

- understand the types and relationships of plans associated with product line adoption
- understand the typical structure and contents of plans associated with product line adoption
- understand the factors influencing the plans associated with product line adoption
- understand the relationship between plans and a planning process
- understand the principles of a planning process

A Problem

Diagnostic methods often find weaknesses in planning.

- PLTP
- CMMI SCAMPI
- SEI Risk Evaluations

A paradox

- How do you plan for change when you are weak at planning?
  - You don't do it (or don't do it well).
  - You struggle.
  - You get stuck.

Competency at planning is fundamental to successful change.

The solution: Get better at planning.

Planning and Plans

It is useful to distinguish between planning, which is a process, and plans, which are the **artifacts** or **result** of a planning process.

If you have a planning process, it can

- produce repeatable results
- be taught to others
- provide a foundation for a change effort
- be an important reusable asset

We will

- discuss the plans that support product line adoption
- overview a planning process to create these plans
- see an example application of the planning process

Session Topics

Plans

Plan characteristics, relationships, and structure

A planning process

Summary

Product Line Plans - Revisited

Planning is emphasized in the Framework.

- "Technical Planning" practice area for project-level planning
  - core asset development and maintenance plans
  - the project details of production plans (for generating products)
  - product development and maintenance plans
- "Organizational Planning" practice area for strategic-level planning
  - plans for core asset funding, configuration management, training, other cross-project needs
  - **product line adoption plans and action plans**

Planning is also a key aspect of the "Launching and Institutionalizing" practice area.

In the IDEAL Model, planning is in the Establishing Phase.

Plans Supporting Software Product Line Adoption Revisited - 1

Business Plan

- a plan that lays out overall company strategies to achieve business goals
- Such a plan might specify adopting a software product line approach for a particular set of products.

Product Line Adoption Plan

- a plan that describes how product line practices will be rolled out across the organization
- may describe several IDEAL/*Adoption Factory* iterations

Product Line Action Plan

- a plan to address a specific portion of a product line adoption plan, for example, a single IDEAL iteration or a pilot

Plans Supporting Software Product Line Adoption Revisited - 2

A “Get Started” plan

- If you already have an organizational approach for change, such as IDEAL, the first few steps of product line adoption might be part of the Product Line Adoption Plan.
- If you don’t have a standard way to approach change, the first few steps of product line adoption may be considerably less formal or may be intuitive. We will call such steps the “Get Started” plan.
  - The “Get Started” plan should carry you through a diagnostic, after which you should create more formal product line adoption and action plans.

Session Topics

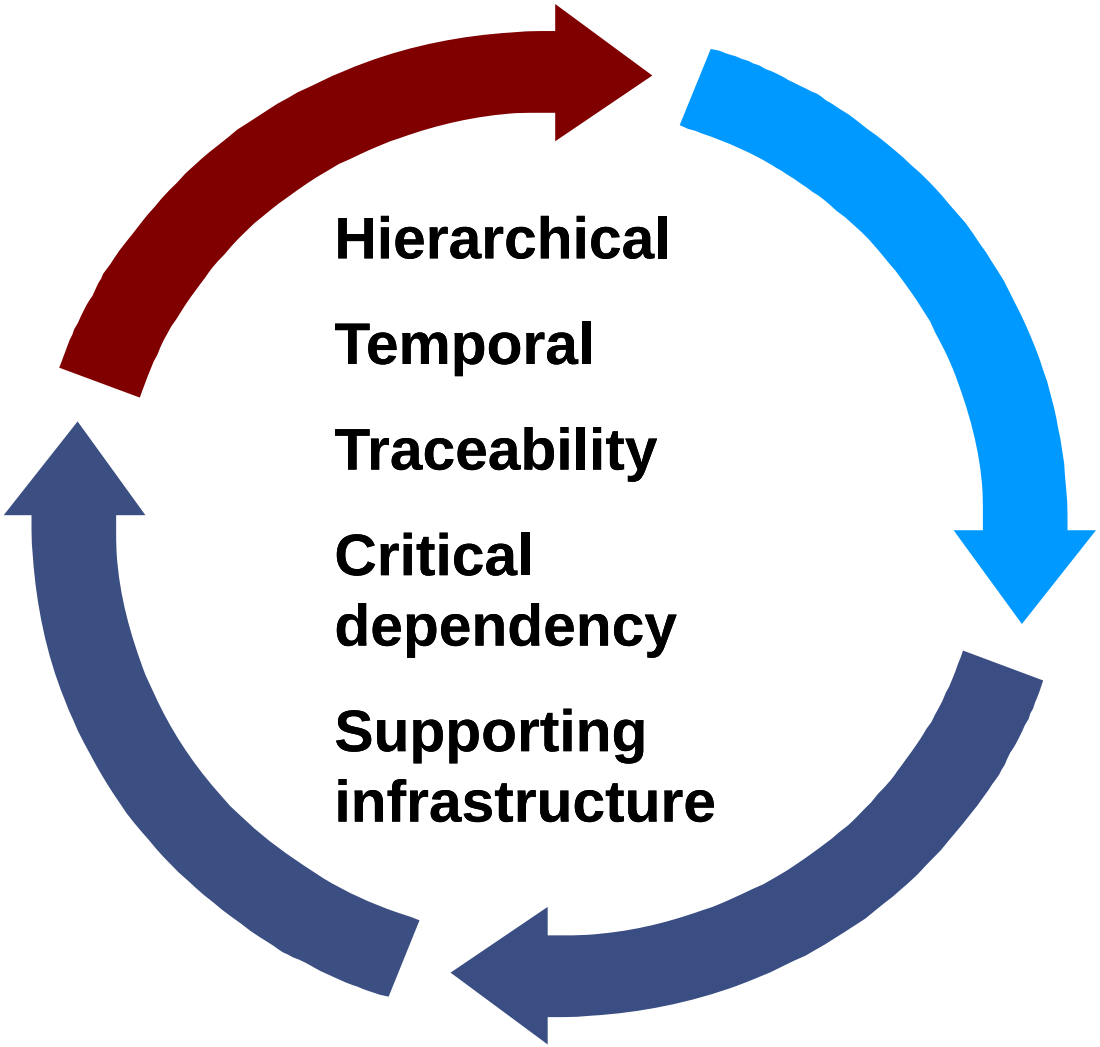
Plans

Plan characteristics, relationships, and structure

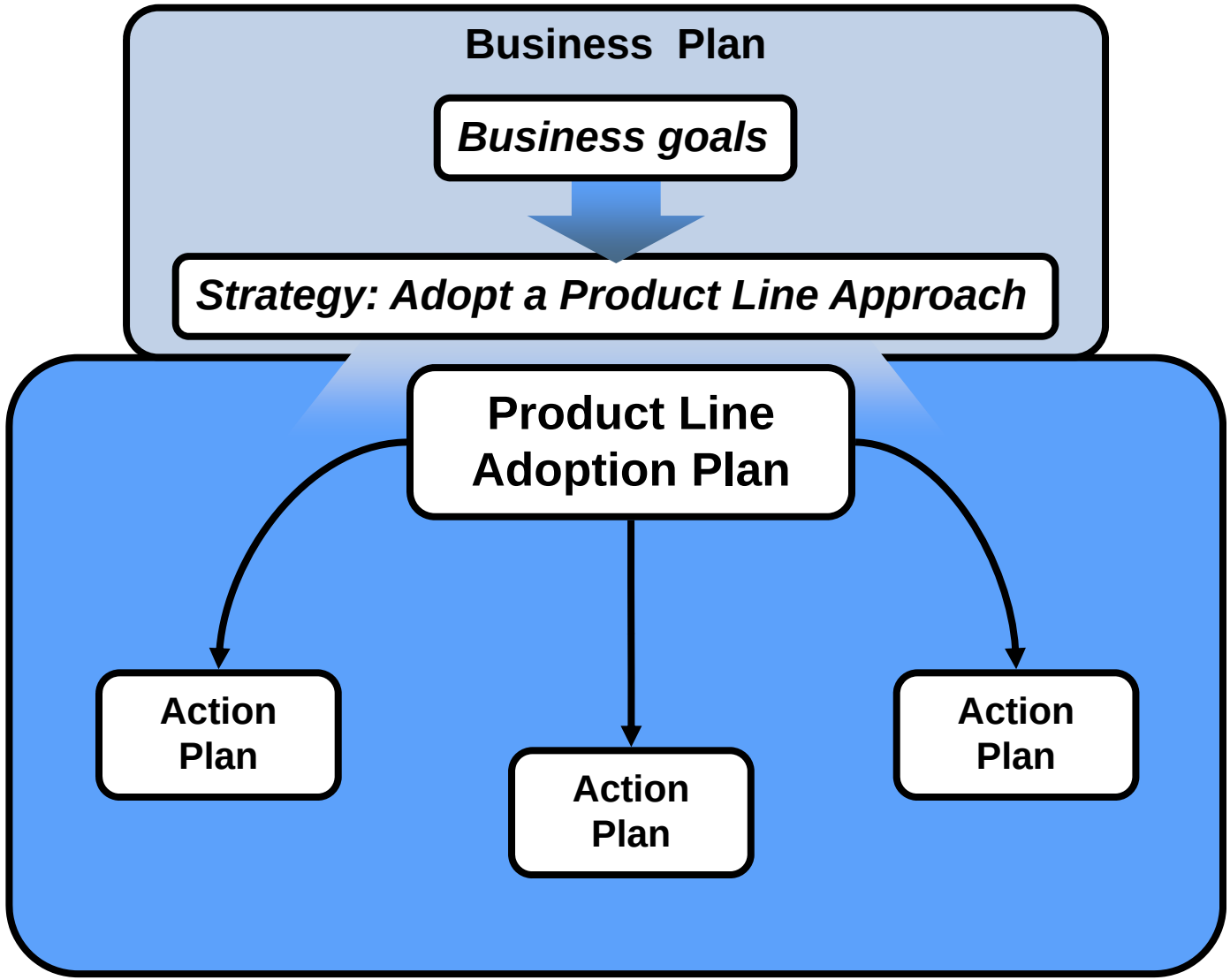
A planning process

Summary

Relationships Among Plans



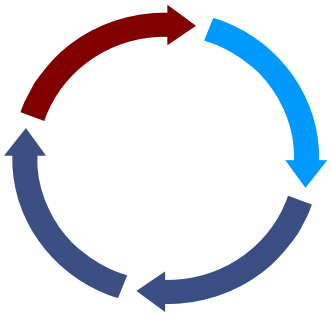
A Hierarchy of Plans



Temporal Relationships Among Plans - 1

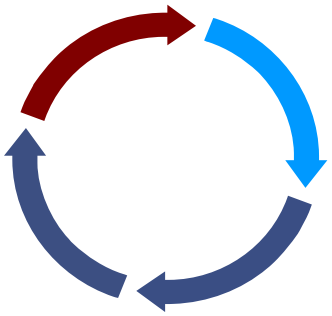
Initiating documents (precede product line plans)

- business plans
- steering group and change group charters
- guiding principles, models, patterns
  - IDEAL
  - *Adoption Factory* pattern
  - the Framework for Software Product Line Practice
- on-going change effort plans
  - process improvement
  - architecture-centric development
  - training
  - etc.



Temporal Relationships Among Plans - 2

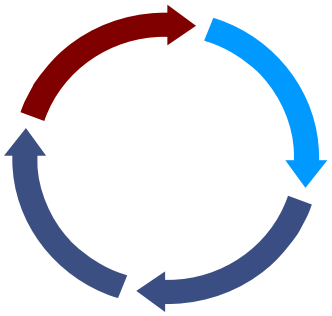
Product Line Adoption and Product Line Action plans



- A “Get Started” plan may precede a more formal Adoption Plan.
- At least a sketch of an Adoption Plan should precede Action Plans
- Refinement and revision of the Adoption Plan may result from lessons learned in action plan execution.
- Some action plans depend on previous action plans.

Temporal Relationships Among Plans - 3

Different strategies for adoption will likely precipitate additional plans



- pilot project plans
  - may have limited organizational scope but include full product line practices
  - may have full organizational scope but limited product line practices
  - may have some combination of the above
- Subsequent roll-out and institutionalization plans must be developed to bring entire organization and all practices up to speed.

Other Plan Relationships

Traceability

- goals and sub-goals both within and across plans

Critical dependency

- schedule
- products assumed from other activities

Supporting infrastructure

- quality improvement culture
- team development
- meeting management
- other environmental interfaces
  - communications
  - finances
  - personnel

Characteristics of Good Plans

Appropriate for purpose

- Usable
- Clear
- Brief
- Easy to update
- Sufficient level of detail
- Internal and external consistency and traceability

Plan Heuristics

Modularize to separate:

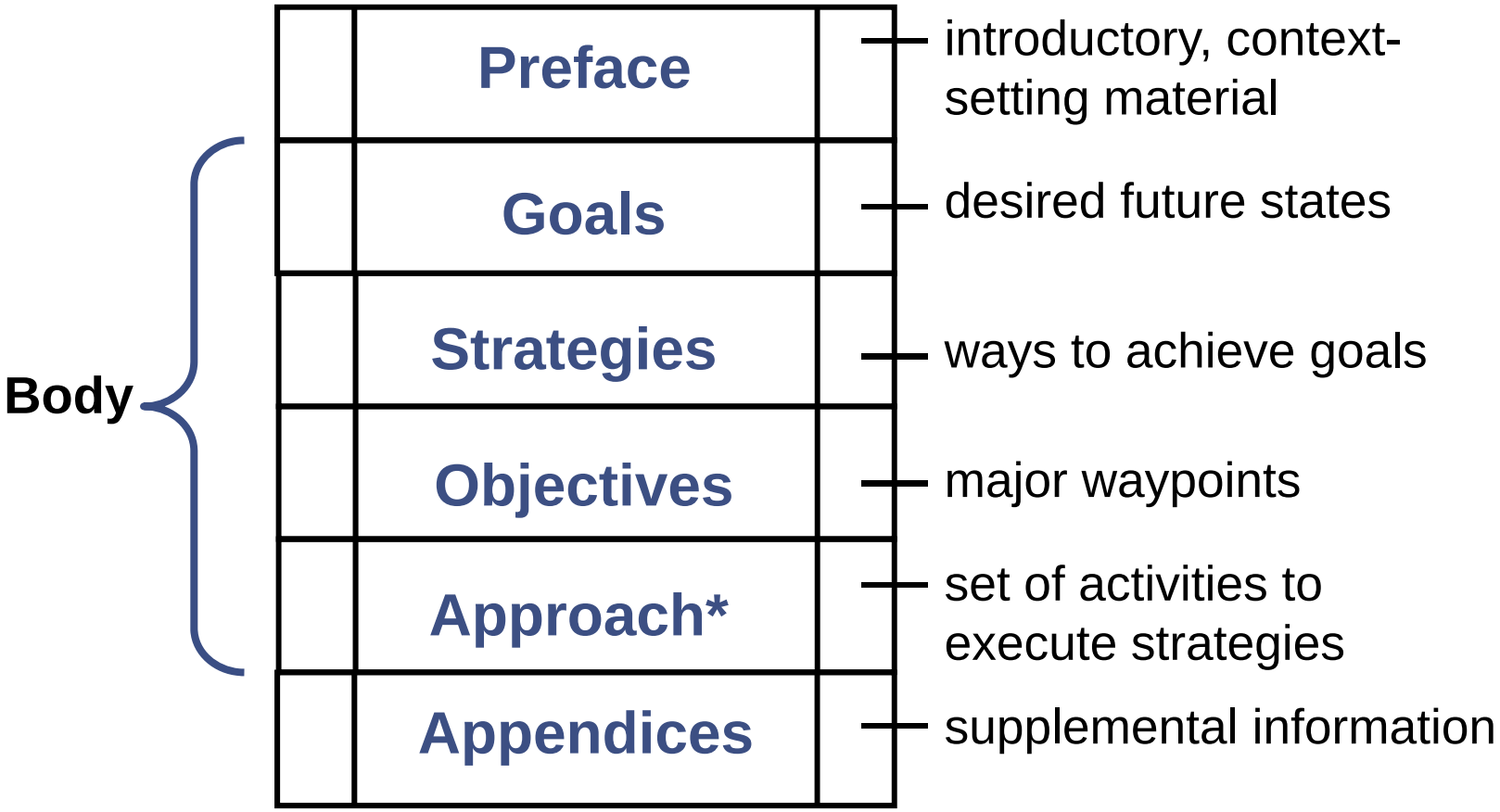
- volatile information from enduring information (best if in different documents or appendices)
- supporting detail from plan tracking detail

A plan reflects a level of understanding at a given point in time. Some details may be deferred or must be expected to be less precise.

Plan contents are often tailored to fit the purpose and environment.

Plans are created using a non-sequential planning process.

A Basic Plan Structure



** Might not be part of an Adoption Plan*

** Typically managed using a project tracking tool*

Session Topics

Plans

Plan characteristics, relationships, and structure

A planning process

Summary

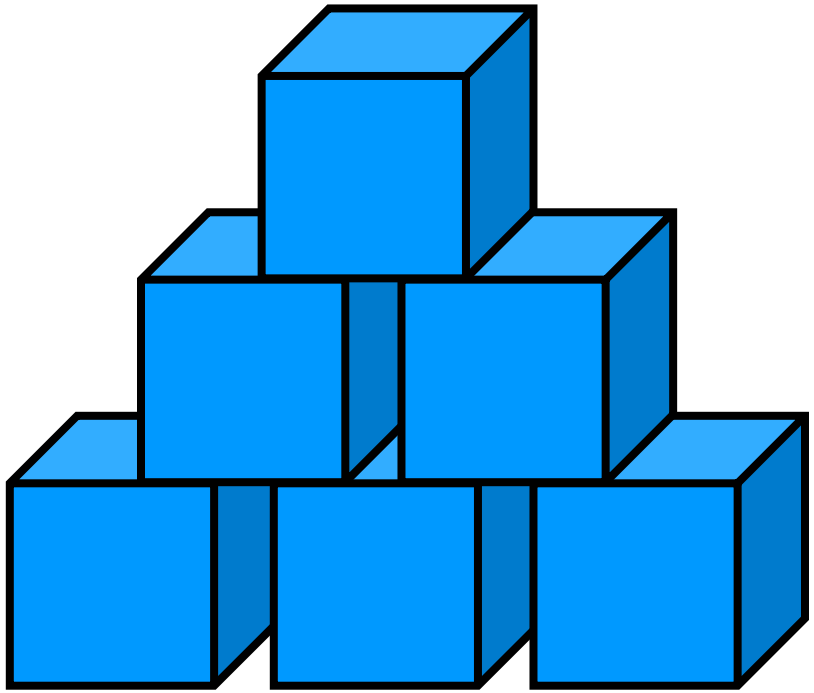
Our Planning Process Principles

Validation

Iteration

Consensus

Adaptation



Guidelines

Definition: A statement of all applicable processes, structures, definitions, and artifacts that provides overall organizational guidance for all strategic initiatives

The guidelines will

- specify overarching processes or fundamental notions
- specify standards or models to be used
- identify supporting organizational structures
- specify standard methods

Criteria:

- Complete
- Applicable
- General

Principle: Validation

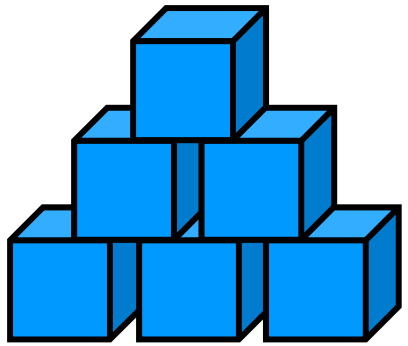
After we draft each work product, we validate it.

We validate each product to determine the following:

- Does it match the criteria for being a “good” work product?
- Is it in consonance with any parent work products?
- Is it appropriate?
- Is it realistic for the organization’s context (known findings, internal knowledge, external expertise, organizational culture)?

We fix any deficiencies exposed during validation.

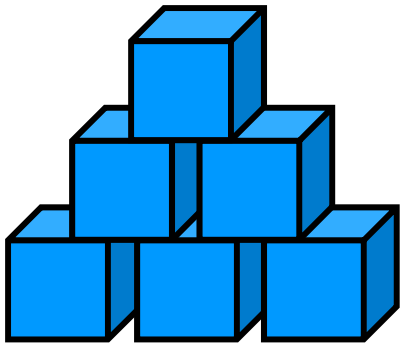
We document validation for future reference.



Principle: Iteration

We recognize the inherent iteration in strategy setting and planning.

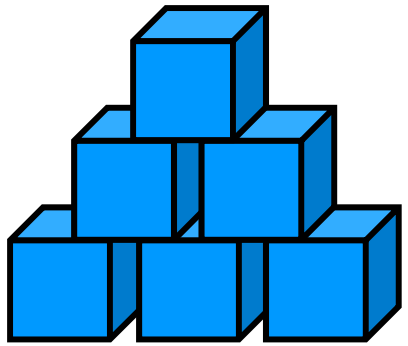
- The strategy setting and planning process is not a sequential process.
- The process is iterative (at best!).
- When to quit iterating is not always evident.



Principle: Consensus

We all must reach consensus on decisions made in our process.

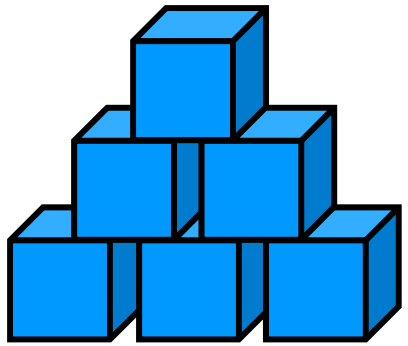
- Stakeholders should be involved during the process.
- The process for building strategies and plans must have consensus-building mechanisms.



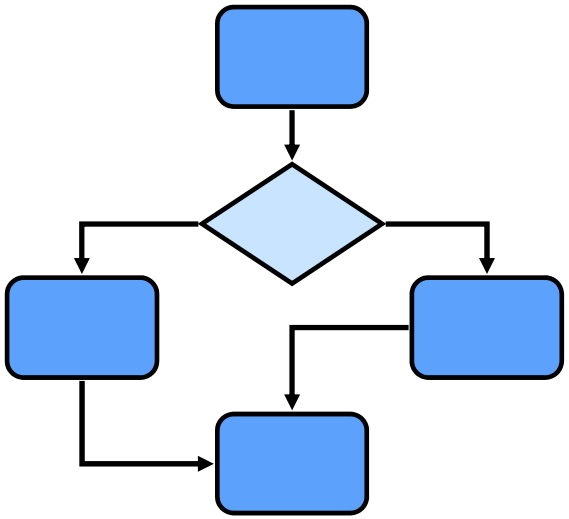
Principle: Adaptation

We must adapt the general strategy setting and planning process to fit the purpose and organization.

- Process steps should match the level of detail required.
- The process should be appropriate to the organizational environment.
- All assumptions necessary to make progress should be documented.



A General Planning Process - 1



1- 3. State and validate

- the goals and rationale for each goal
- the guiding principles and prerequisites
- barriers and enablers

Note: Thus far, the validation is only against the “goodness” criteria for each part of the plan.

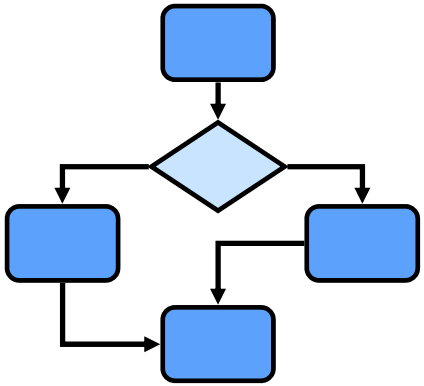
4. Identify the plan owner.

5-6. State the strategies and validate each. Validate the set against the associated goals.

7-8. State the objectives and validate each. Validate the set against the associated goals and strategies.

Note: This may be as far as you go for a *product line adoption plan*.

A General Planning Process - 2



Note: For a *product line action plan* you would also do these steps.

9-10. Define the approach (set of activities) and validate each.

11. Validate the approach against the objectives.

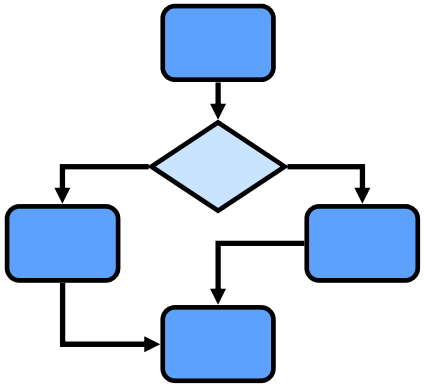
12. Modify the objectives as appropriate.
(Possibly modify the goals and prerequisites.)

13. Diagram the temporal relationships among activities.

14. For each activity, identify and validate activity details.

15. Modify the approach and objectives as appropriate.
(Possibly modify the goals and prerequisites.)

A General Planning Process - 3



Note: Do these steps regardless of the level of the plan.

16. **Assemble the appendix documents.**
17. **Write a guide to the appendices.**
18. **Format the plan for usefulness.**

Some Comments on the General Planning Process - 1

Remember the principle of adaptation.

- Nothing is cast in stone.

For example,

- When developing a hierarchical set of plans, the lower-level plans may inherit goals and objectives, rather than having planning participants develop them.
- If lower-level planning done by a different group reveals the need to modify higher-level plans, changes will have to be negotiated between the groups.

Some Comments on the General Planning Process - 2

Some possible variations

- For a dispirited (and maybe dysfunctional) group it may be good to start out with a “Whine and Cheese” Party
  - Give the group a round robin opportunity to surface their gripes and record them, but with a promise not to raise them again.
- A less drastic version of this is to identify prerequisites for success as an early step.

To keep action plans realistic, generally only plan details out about six months (but update every month).

There are also some product line adoption adaptations to bring to bear on the general process.

Formatting Plans

The outputs of the planning process provide you with the contents of the plan.

You still need to format the contents effectively to make a plan that is practical to guide and track the project.

Remember to separate volatile information from enduring information.

- Frequently, the information in Objectives and Approach is volatile since it contains dates and estimates.
- This information is typically managed using a project tracking tool.

Some Product Line Adaptations - 1

Getting started with product lines is always hard to codify.

- A“Get Started” plan may be intuitive and stand-alone or it may be part of the product line adoption plan.

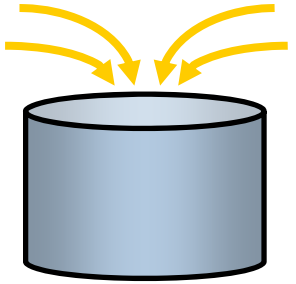
Either way, a sketch of a “Get Started” plan is:

- Perform “sufficient” steps of the IDEAL Initiating Phase to get started. This would include
  - feasibility assessment (product line entry criteria)
  - initial staffing
  - initial training
  - initial sponsorship and champions
  - preparation for a diagnostic
- Conduct a diagnostic.

Some Product Line Adaptations - 2

To proceed to an adoption plan:

- Sketch a high-level product line adoption plan based on the diagnostic results while considering the other product line planning factors. (See the “kettle.”)
- Follow the general planning process as informed by the “kettle.” This should reveal short-term and longer-term issues. This will allow the planners to determine
  - whether there should be an “adoption plan – action plan” hierarchy
  - whether there should be an iterative, pilot-based approach.
- Use these results to revise and refine the product line adoption plan and develop supporting action plans.



Session Topics

Plans

Plan characteristics, relationships, and structure

A planning process

Summary

Summary - 1

There are many types of plans needed to support product line launching and institutionalization.

An adoption plan describes how product line practices will be rolled out across the organization.

- How to get started might be part of the adoption plan or it might precede formal product line adoption planning.

A set of action plans are usually needed to support the adoption plan.

A basic plan structure includes: preface, body, appendices.

Summary - 2

Planning is the foundational step for the implementation of all change activities, including product line adoption.

A structured planning process provides repeatable results.

Our planning process embodies the principles of

- validation
- iteration
- consensus
- adaptation

We have found our planning process is a practical approach to building product line adoption plans.

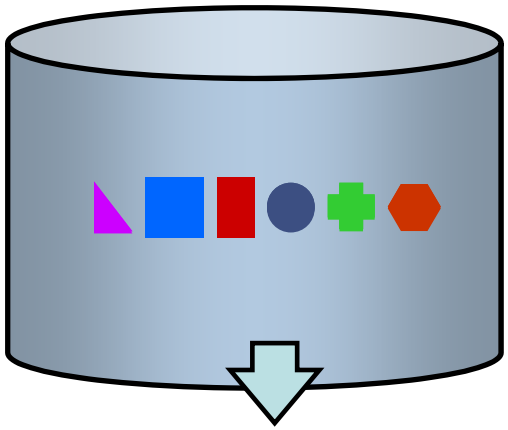
Factors Influencing Adoption

Organizational Context

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

Adoption Support

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



Product Line Adoption Plan

. . .

Product Line Action Plans

Software Engineering Institute | Carnegie Mellon

Adopting Software Product Lines

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Part 3: Adoption Planning
Module 8: Planning Example

Session Outcome

After this session participants should

- be familiar with applying a planning process to product line adoption

Session Topics

Scenario

Goals and Rationale

Barriers and Enablers

Guidelines and Prerequisites

Strategies

Objectives

Approach

Activity Details

Putting it all together

Using the Planning Process

We will go through the planning process demonstrating how to build a skeleton product line action plan based on an example scenario.

While we will gain experience with the planning process, we will defer many details in the interests of getting through the entire planning process.

Scenario - 1

Widget World, a medium-sized company (100 software developers) has determined that adopting a software product line approach is a key strategy to achieve business goals of reducing time to market and reducing field maintenance costs. (Product line criteria were checked and determined to match the company situation.)

- This was made part of the Business Plan.

Key management and technical stakeholders created a “Get Started” plan roughly following the IDEAL model’s initial phases.

The goal of the “Get Started” plan is, “We have a plan to guide us in adopting a software product line approach.”

Scenario - 2

Strategies in the “Get Started” plan include

- Establish the initial **support infrastructure** (product line steering group, product line manager, key technical staff).
- Conduct an SEI **Product Line Technical Probe (PLTP)** to determine the current state of practice.
- Build an initial product line adoption plan and appropriate **action plans** to address the PLTP findings.

We now join Widget World after the PLTP has been conducted.

PLTP Overall Results - 1

Widget World implemented architecture-centric development practices for their principal products, but from an independent, single-system development perspective.

Widget World started a CMMI process improvement program and implemented the following processes at a (staged) Level 2 Maturity Level.

- Requirements Management
- Project Planning
- Project Monitoring and Control
- Configuration Management

PLTP Overall Results - 2

While there are good practices to build on, little has been done to support a software product line approach.

With the exception of key players, the staff is generally uninformed about the product line approach.

There is significant pressure to meet schedule. Sometimes quality and configuration management process adherence is sacrificed.

PLTP Overall Results - 3

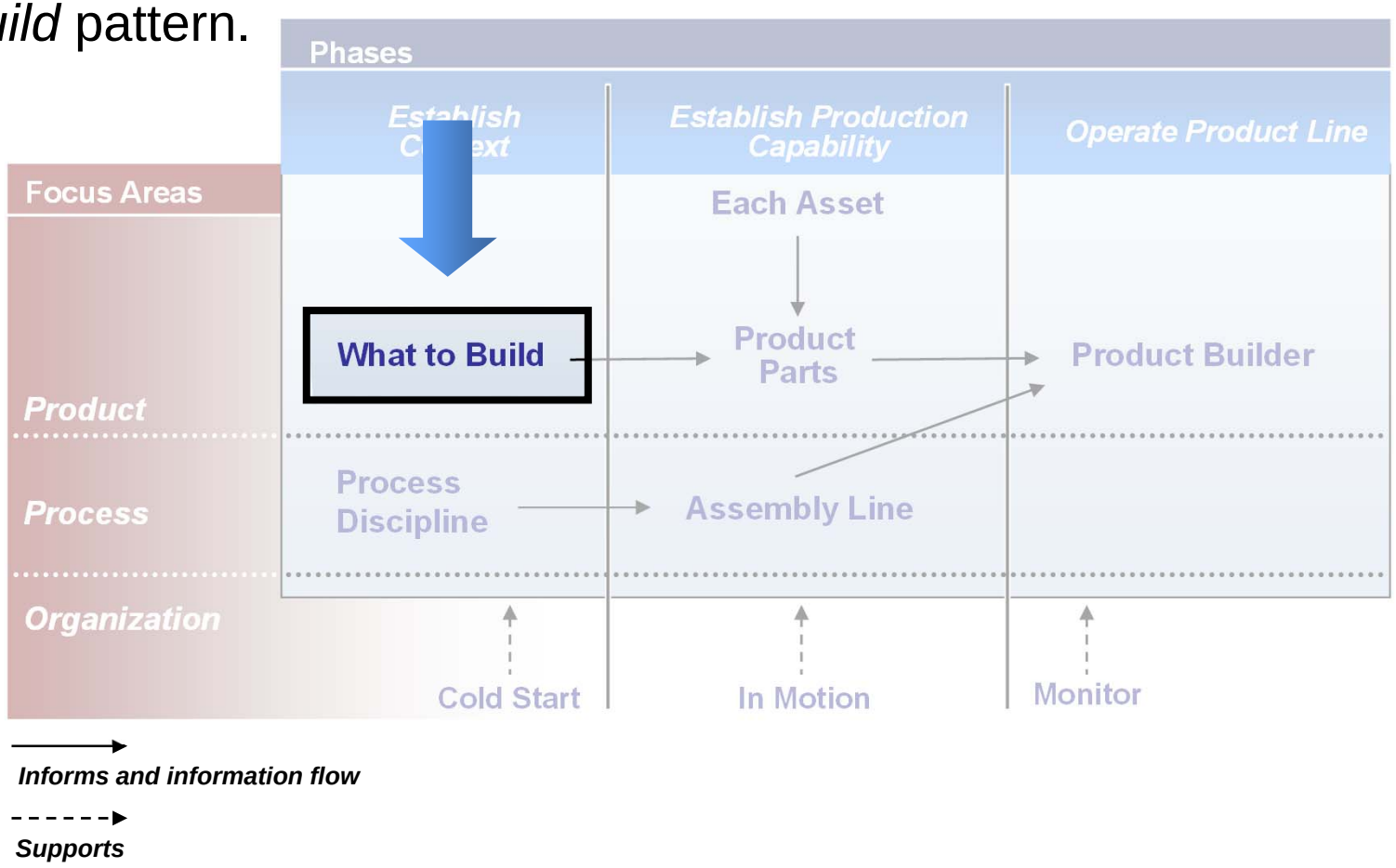
** High Priority
* Medium Priority

You are here

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

Detailed Results

We will see examples of some detailed results related to the *What to Build* pattern.



Some PLTP Findings for What to Build Practice Areas - 1

Understanding Relevant Domains

- There is good domain expertise, but the knowledge is incompletely documented, not consolidated, and not easily accessible (in various papers, reports, presentations).
- relative priority of addressing: **low**

Market Analysis

- Market targets have been informally identified but not vetted against a defined scope or business case.
- relative priority of addressing: **medium**

Technology Forecasting

- Hardware forecasting is done well; software forecasting lags somewhat.
- relative priority of addressing: **low**

Some PLTP Findings for What to Build Practice Areas - 2

Developing a Business Case

- There are known, prioritized business drivers but no formal business case to support product line scoping decisions.
- relative priority of addressing: **high**

Scoping

- There is only an informal understanding of the probable scope of the product line and thus no vetting with the business case or examination of how the current software architectures support a product line.
- relative priority of addressing: **high**

Et Cetera

A real PLTP would have more details and results for all 29 practice areas.

Response to the PLTP - 1

After considering the PLTP results, the managers decided it would be too risky to tackle a “big bang” adoption given the large number of asynchronous, critical customer deliveries.

Thus, the managers identified a relatively small pilot project that took advantage of a new development based on an existing, well-understood product. This pilot only covers a small part of the organization (the Mini-Widget project) and a subset of Widget World’s product portfolio.

Funding has been provided for the pilot effort but not the larger product line effort.

Response to the PLTP - 2

The “sketch” adoption plan is

- Plan and conduct the chosen pilot project using “lightweight” implementations of certain practice areas.
- Identify opportunities to pipe-line work on selected practice areas for the complete product line before the pilot is completed.
- Evaluate the results of the pilot and re-plan the large-scale product line adoption.

The planning for the pilot will result in an action plan supporting the adoption plan.

Our Approach for the Example

Discuss a specific part of the product line action plan and the planning processes steps to create it.

- definition
- criteria
- examples
- process steps

Illustrate the execution of the process steps by both showing example results and discussing other possibilities.

The SEI Product Line Planning Approach

When conducting SEI product line planning we

- Overview the process and plan structure.
- For each part of the plan
  - Present the definition, criteria, examples, and process steps.
  - Execute the process steps.

It typically takes 1½ to 2 days to create a skeleton draft product line adoption plan.

- Specific action items are assigned to create a complete draft plan.

Session Topics

Scenario

Goals and Rationale

Barriers and Enablers

Guidelines and Prerequisites

Strategies

Objectives

Approach

Activity Details

Putting it all together

Goal

Definition: A single, well-defined and desired future state to be achieved by the organization.

Criteria:

- traceable to an enterprise strategic initiative
- aligned with stated vision
- separable from other goals
- important to the organization and its mission
- realistic for the organization
- clear
- strategic level

Advice: around 4 or 5

Example: *A core asset base for satellite ground-based command and control exists.*

Process for Establishing Goals

Describe desired states.

Align with results of product line diagnostic (and other kettle considerations).

Cluster and refine into goals.

Validate against criteria.

Compare with diagnostic recommendations (if any).

Document rationale for each.

Rationale

Definition: The rationale is a collection of the key reasons justifying the choice of a goal.

Criteria:

Brief

Complete

Expressed in terms of existing data, as in the following examples:

- higher identified goals
- findings from a diagnostic
- identified organizational needs
- directives
- recognized standards
- historical or organizational context

Advice: Answers the *why* of the plan

Rationale: Examples

A priority recommendation of our organization's recent SEI Product Line Technical Probe was to create a product line concept of operations.

The May 5, 2004 corporate headquarters memo directs us to define and streamline our proposal-writing process.

The Management Steering Group has determined that improving our software quality assurance process is the most effective way to improve customer satisfaction.

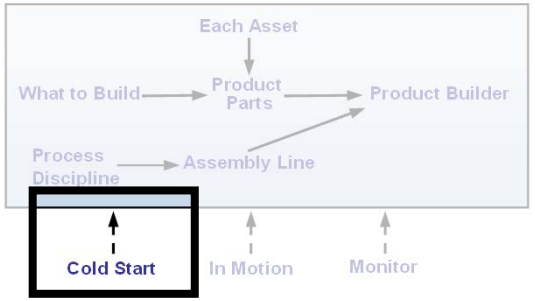
The CTO has directed us to pursue a product line approach for all diesel engine control software.

Example Scenario: Goals and Rationale

Goal 1. There is a mutually agreed upon set of Mini-Widget products to be supported by our product line over the next two years.

Rationale. Our organization's recent SEI Product Line Technical Probe revealed that we lacked a focus of what products would constitute our product line. Understanding this is crucial in order to provide the basis for our product line effort. A well defined scope, supported by a business case, will also provide the basis for our software product line architecture. Learning how to define the set of products for a limited pilot will reduce risks for adopting these practices company-wide.

Goals Discussion - 1

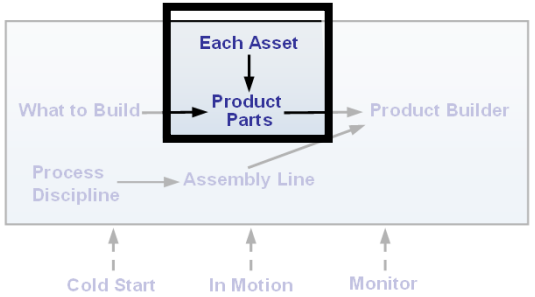


What goals might be appropriate to address the *Cold Start* high priority PLTP findings in the example scenario?

Reminder: In the Establish Context Phase, the high priority practice areas were

- Launching and Institutionalizing
- Funding
- Structuring the Organization
- Operations
- Training

Goals Discussion - 2



What goals might be appropriate to address the *Each Asset* and *Product Parts* priority PLTP findings in the example scenario?

Reminder:

- In the Establish Production Capability Phase, the high priority practice area was Architecture Definition.
- The medium priority practice areas were Requirements Engineering and Component Development.

Session Topics

Scenario

Goals and Rationale

Barriers and Enablers

Guidelines and Prerequisites

Strategies

Objectives

Approach

Activity Details

Putting it all together

Barriers and Enablers

Barriers - obstacles to achieving stated goals

Enablers - existing support for achieving stated goals

Process for Barriers and Enablers

Revisit current strengths and challenges from diagnostic results and select key barriers and enablers to achieving goals.

Brainstorm additional barriers and enablers to achieving goals.

Refine.

Barriers and Enablers Discussion

Based on the overall PLTP findings for the example scenario

- What are some of Widget World's organizational barriers?
- What are some of Widget World's organizational enablers?

What additional barriers might you identify in your organization?

What additional enablers might you identify in your organization?

Session Topics

Scenario

Goals and Rationale

Barriers and Enablers

Guidelines and Prerequisites

Strategies

Objectives

Approach

Activity Details

Putting it all together

Guidelines - 1

Definition: A statement of all applicable processes, structures, definitions, and artifacts that provides overall organizational guidance for all strategic initiatives

The guidelines will

- specify overarching processes or fundamental notions
- specify standards or models to be used
- identify supporting organizational structures
- specify standard methods

Criteria:

- Complete
- Applicable
- General

Guidelines - 2

Examples:

Overarching processes

- Adoption Factory pattern
- IDEAL model
- piloting
- communication processes
- risk-management processes
- change and resistance management processes

Standards or models

- Framework for Software Product Line Practice
- UML
- DoDAF

Supporting improvement structures

- Management Steering Group

Standard methods

- ATAM®-based architecture evaluation
- PLTP

Prerequisites - 1

Definition: An explicit statement listing all clarifications, documents, and cultural and structural necessities that are critical to the achievement of the primary goal

Criteria:

- Necessary and sufficient
- Out of the jurisdiction of the responsible team and the team's plan but necessary for success.

Prerequisites - 2

Examples:

The following are necessary to achieve the primary goal of this plan, e.g.,

- Quality culture foundations
  - meeting management process
  - team development principles
  - visible management support
- Resources
  - time (how much?)
  - money (how much and applied to what?)
  - people (how many and what roles/skills?)

Guidelines and Prerequisites Discussion

Guidelines

- What are some potential guidelines for the Widget World scenario?
- What other guidelines are appropriate for your organization?

Prerequisites

- What are some potential prerequisites for the Widget World scenario?
- What other prerequisites are appropriate for your organization?

Session Topics

Scenario

Goals and Rationale

Barriers and Enablers

Guidelines and Prerequisites

Strategies

Objectives

Approach

Activity Details

Putting it all together

Strategies

Definition: Stated approach toward achieving a goal.

Criteria:

- must achieve the associated goal (at least in part)
- is realistic for organization to implement
- has minimal adverse impact on other goals
- makes progress toward other goals, if possible
- should attack barriers and capitalize on enablers

Advice:

- should not exceed three per goal
- some should address known barriers
- some should capitalize on enablers

Example: *Develop a product line architecture to support a pilot project.*

Process for Establishing Strategies

Brainstorm.

Refine.

Validate against strategy criteria.

Validate set of strategies against goals for coverage.

Example Scenario: Strategies

Goal 1. There is a mutually agreed upon set of Mini-Widget products to be supported by our product line over the next two years.

- **Strategy 1-1:** Create an initial product line scoping document.
- **Strategy 1-2:** Create an initial product line business case.
- **Strategy 1-3:** Consolidate informal domain understanding and market analysis information.
- **Strategy 1-4:** Reconcile initial scope, business case, domain understanding and market analysis documents.

Session Topics

Scenario

Goals and Rationale

Barriers and Enablers

Guidelines and Prerequisites

Strategies

Objectives

Approach

Activity Details

Putting it all together

Objectives

Definition: Specific measurable, time-related statements of what is to be accomplished.

Criteria:

- measurable
- time-related
- realistic for the organization
- tied to accomplishment of at least one strategy

Advice: about 4-6 objectives per strategy

Examples

- *Product line pilot has been selected by two months after project start*
- *Product line pilot has been approved and funded by 1 Oct.*

Process for Establishing Objectives

Brainstorm.

Refine.

Validate against

- objective criteria
- strategies.

Example Scenario: Objectives - 1

Goal 1. There is a mutually agreed upon set of Mini-Widget products to be supported by our product line over the next two years.

- **Strategy 1-1:** Create an initial product line scoping document.
  - **Objective 1-1-1.** Education about scoping is completed by September 30.
  - **Objective 1-1-2.** Criteria for inclusion in scope is agreed upon by October 15.
  - **Objective 1-1-3.** Initial scope document is agreed upon and submitted for Steering Group approval by December 1.

Example Scenario: Objectives - 2

- **Strategy 1-2:** Create an initial product line business case.
  - **Objective 1-2-1.** Education about business case contents is completed by September 30.
  - **Objective 1-2-2.** Alternative business case scenarios are developed by October 15.
  - **Objective 1-2-3**. Initial business case is agreed upon and submitted for Steering Group approval by December 1.

Example Scenario: Objectives - 3

- **Strategy 1-3:** Consolidate informal domain understanding and market analysis information.
  - **Objective 1-3-1.** Existing sources of domain information are identified by September 30.
  - **Objective 1-3-2.** Domain information consolidation report is completed by October 15.
  - **Objective 1-3-3.** Market analysis workshop is conducted by October 15.
  - **Objective 1-3-4.** Market analysis workshop report is presented to Steering Group by December 1.

Example Scenario: Objectives - 4

- **Strategy 1-4:** Reconcile initial scope, business case, domain understanding and market analysis documents.
  - **Objective 1-4-1.** Participant workshop is completed by December 15.
  - **Objective 1-4-2.** Revised scope document is completed by February 1.
  - **Objective 1-4-3.** Revised business case is completed by February 1.
  - **Objective 1-4-4.** Revised market analysis is completed by February 1.
  - **Objective 1-4-5.** Revised domain document is completed by February 1.

Session Topics

Scenario

Goals and Rationale

Barriers and Enablers

Guidelines and Prerequisites

Strategies

Objectives

Approach

Activity Details

Putting it all together

Approach and Activities

Definitions:

- Approach: the set of planned activities
- Activities: the basic elements of an operational plan

Criteria:

- decomposable into tasks
- actionable and assignable
- traceable to specific objective(s)
- answers the how of the plan

Advice: Aim for a maximum of 12-18 activities per strategy

Examples:

- *Document the current organizational roles and responsibilities.*
- *Design a new organizational structure for the product line effort.*

Process for Establishing Approach

Define approach.

Validate against

- activity criteria
- objectives.

Modify objectives as appropriate.

Activity Relationship and Validation

Activity relationships: A diagram of the temporal relationships among activities (data flow, dependency graph, or other suitable tool)

Activity/objective validation: A matrix to show which activities contribute to the achievement of which objectives. This validation can reveal:

- missing objectives
- superfluous activities
- scoping problems with primary goal
- critical paths
- inappropriate decomposition of activities
- missing prerequisites
- needed common assets

Activity Validation Matrix

Activities \ Objectives									

Example Scenario: Activities

Strategy 1-1: Create an initial product line scoping document.

- Activity 1-1-1: Form scoping team.
- Activity 1-1-2: Identify appropriate training & send team.
- Activity 1-1-3: Identify and interview key experts
- Activity 1-1-4: Identify scope alternatives & scope criteria.
- Activity 1-1-5: Complete Draft 1
- Activity 1-1-6: Conduct external review of Draft 1
- Activity 1-1-7: Complete initial scope document & submit

(Reminder about Objectives)

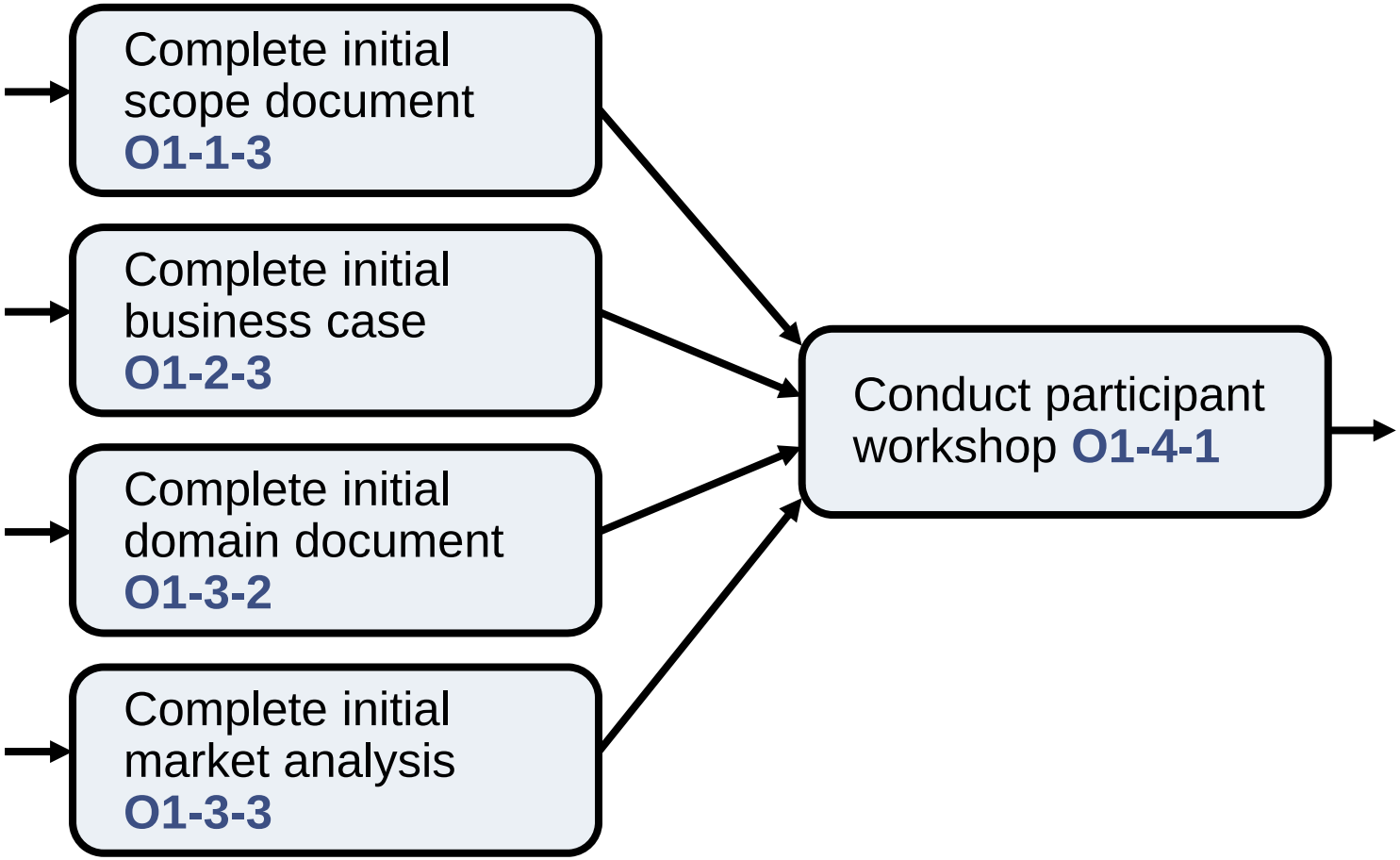
- 1.1.1 education completed
- 1.1.2 scoping criteria agreed to
- 1.1.3 initial scope document submitted)

Example Activity Validation Matrix

Objectives / Activities	Educ 1.1.1	Crit 1.1.2	Sub 1.1.3						
1-1-1 Form team	?								
1-1-2 ID trg & train team	X								
1-1-3 Interview experts		?							
1-1-4 scope alts & crit		X							
1-1-5 Draft 1			X						
1-1-6 Review Draft 1			X						
1-1-7 complete & submit			X						

? – Is this a stretch? Are we missing an objective?

Example Dependency Graph



Session Topics

Scenario

Goals and Rationale

Barriers and Enablers

Guidelines and Prerequisites

Strategies

Objectives

Approach

Activity Details

Putting it all together

Activity Details

Definition: The specific information necessary to begin task breakdown and assignment.

Criteria:

- appropriate to targeted activity
- aligned with organizational policy
- sufficient to begin task breakdown

Advice: include such information as

- major deliverables
- required communication and reviews
- resource estimates (time, effort, tools, etc.)
- responsible individuals
- major risks

Process for Establishing Activity Details

Define activity details.

Validate against detail criteria.

Modify objectives as appropriate.

Example Activity Details

These are often put into electronic planning tools such as Microsoft Project.

Session Topics

Scenario

Goals and Rationale

Barriers and Enablers

Guidelines and Prerequisites

Strategies

Objectives

Approach

Activity Details

Putting it all together

Next Steps

Note that our example only takes the organization through the pilot.

According to the action plan, it is now necessary to evaluate lessons learned and plan the next steps.

Depending on the organization's commitment and whether the initial pilot experience is favorable, it can start some parallel activities of product line adoption during the pilot to prepare for the next steps.

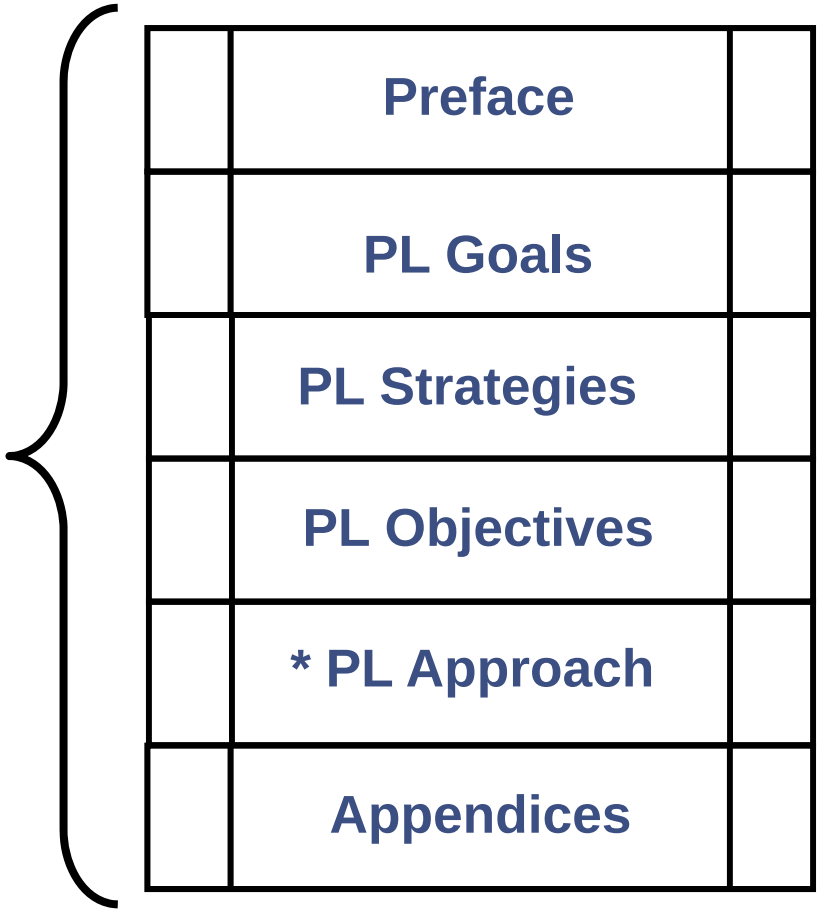
Other Considerations

The example doesn't explicitly address general change issues directly.

What additional supporting goals (or strategies) might you add to address these considerations?

Formatting the Product Line Action Plan

Product Line Action Plan



* Typically managed using project tracking tool

Parts of the Action Plan

Preface

- introductory, context-setting material including: purpose, linkage to other plans, owner, prerequisites for success, guiding principles, barriers, enablers, etc.

Goals

- well-defined and desired future states to be achieved by the organization (about 4 or 5 in the plan)

Strategies

- approaches toward achieving a goal (about three/goal)

Objectives

- measurable, time-related statements of what is to be accomplished along the way towards goals

Approach

- the set of planned activities to execute strategies

Plan Examples

Refer to the handouts.

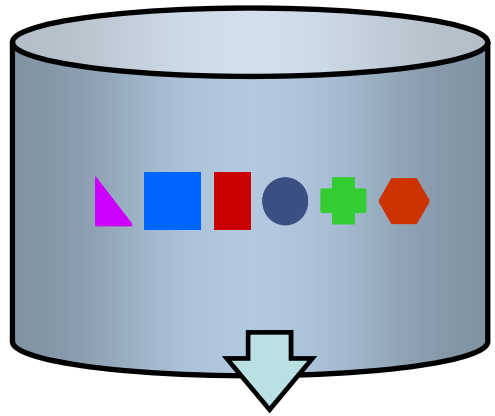
Factors Influencing Adoption

Organizational Context

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

Adoption Support

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



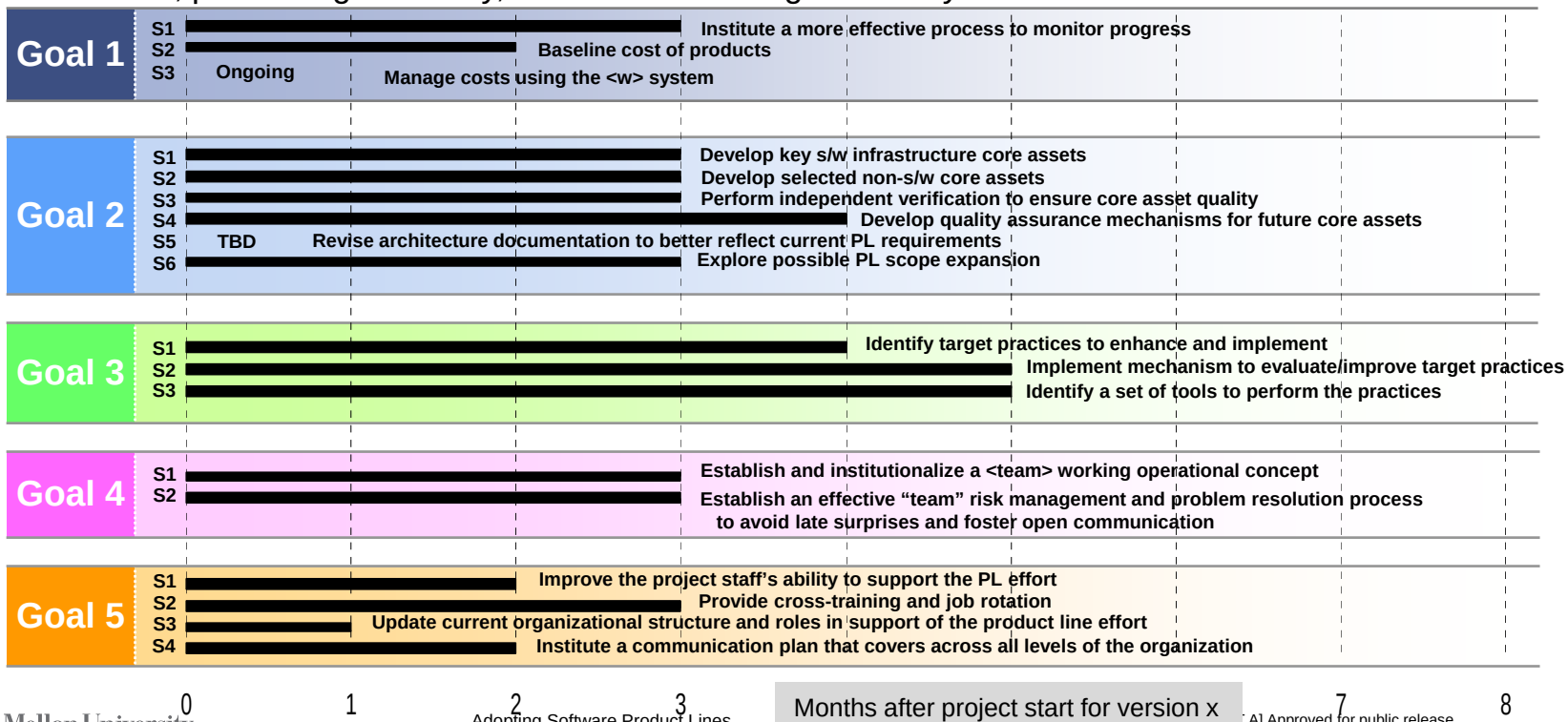
Product Line Adoption Plan

. . .

Product Line Action Plans

Example 1: Action Plan Summary

- **Goal 1**: Business: We deliver <version x> products within budget and schedule and reduced software problem reports at system integration.
- **Goal 2**: Core Assets: The core asset base is systematically grown.
- **Goal 3**: Practices: There is an improved set of institutionalized engineering practices and tools. (both Product Line and Software Engineering)
- **Goal 4**: Customer Interface: <customer>, <prime>, and <consultant> have good working relationship, communication, and clarity of priorities.
- **Goal 5**: People: We have the appropriate organization staffed with people with right knowledge, skills, abilities, performing efficiently, and communicating effectively.



Example 2: Action Plan Summary

Goal 1: (“Business” Goal) We deliver <version x> products within budget and schedule and reduced software problem reports at system integration.

- Strategy 1 (S1): Institute a more effective process to monitor progress.
- Strategy 2 (S2): Baseline cost of products (to allow comparison of using PL approach versus with non-PL approach).
- Strategy 3 (S3): Manage costs using the <w> system.

Goal 2: (“Core Assets” Goal) The core asset base is systematically grown.

- Strategy 1 (S1): Develop key software infrastructure core assets.
- Strategy 2 (S2): Develop selected non-s/w core assets.
- Strategy 3 (S3): Perform independent verification to ensure core asset quality.
- Strategy 4 (S4): Develop quality assurance mechanisms for future core assets.
- Strategy 5 (S5): Revise architecture documentation to better reflect current PL requirements.
- Strategy 6 (S6): Explore possible product line scope expansion.

Goal 3: (“Practices” Goal) There is an improved set of institutionalized engineering practices and tools.

- Strategy 1 (S1): Identify a set of target practices that will be enhanced and implemented within the <ver x> timeframe.
- Strategy 2 (S2): Implement a mechanism for evaluating and improving the target practices within the organization.
- Strategy 3 (S3): Identify a set of tools to perform the practices and specify how to use the tools.

Goal 4: (“Customer Interface” Goal) <customer>, <prime>, and <consultant> have a good working relationship, communication, and clarity of priorities.

- Strategy 1 (S1): Establish and institutionalize a working operational concept that defines how the team will work together.
- Strategy 2 (S2): Establish an effective “team” risk management and problem resolution process to avoid late surprises and foster open communication (escalation mechanism).

Goal 5: (“People” Goal) We have the appropriate organization staffed with people with the right knowledge, skills, abilities, performing efficiently, and communicating effectively.

- Strategy 1 (S1): Improve the project staff’s ability to support the PL effort.
- Strategy 2 (S2): Provide cross-training and job rotation to improve flexibility and promote understanding.
- Strategy 3 (S3): Update current organizational structure and roles in support of the product line effort.
- Strategy 4 (S4): Institute a communication plan that covers across all levels of the organization.

Software Engineering Institute | Carnegie Mellon

Adopting Software Product Lines

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Part 4: Exploiting Other Current Activities
Module 9: CMMI® and Process Improvement

Session Outcomes

After this session participants should

- know some of the relationships between the CMMI models and the Framework for Software Product Line Practice
- know how to identify CMMI and product line linkages that can be exploited during product line adoption
- know some ways that an existing process improvement structure can be exploited for product line adoption

Linking Product Line Adoption to Other Current Activities

Product line adoption will not occur in a vacuum.

- Current management and technical practices provide a starting point (good or bad).
- Other change initiatives are often underway.

As we have seen, the current way of doing business provides input to the “kettle.”

We will examine some ways to build on common, on-going activities.

- process improvement
- architecture-centric development
- hardware platform engineering

Product Line Adoption and Process Improvement

Many organizations considering product line adoption already have a process improvement effort underway.

You can build on an existing improvement initiative to accelerate software product line adoption.

In this session we will examine how to build on

- Capability Maturity Model Integration (CMMI) adoption activities
- an existing process improvement infrastructure

Session Topics

Process improvement and CMMI models

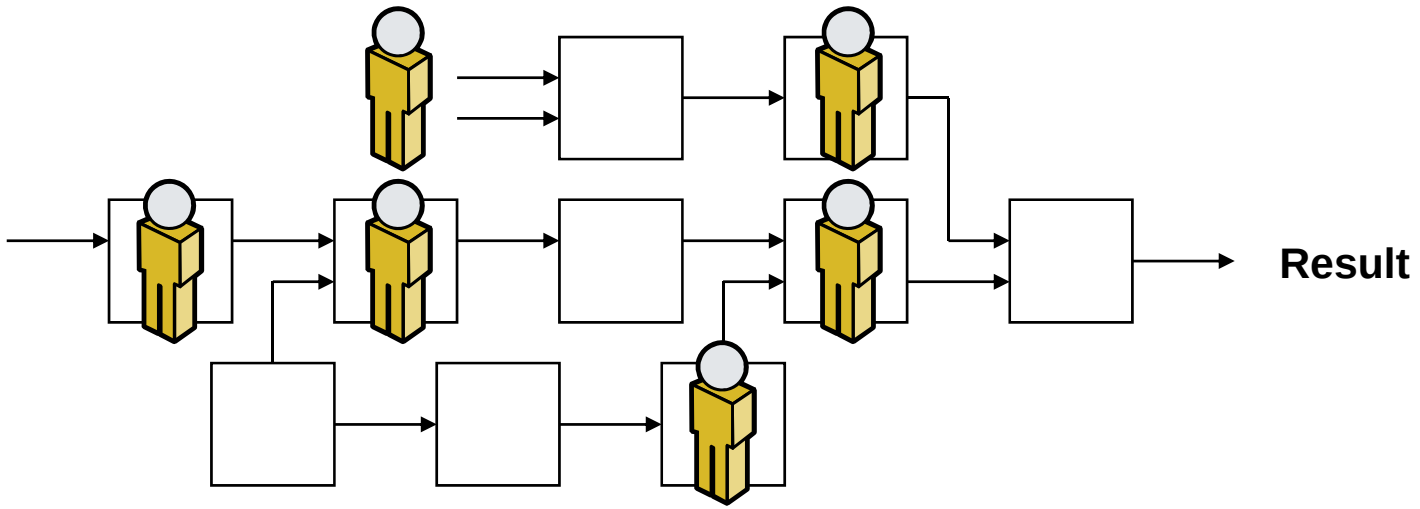
Building on CMMI-based process improvement

Building on a software process improvement infrastructure

Summary

What is a Process?

A process is a logical organization of people and technology into work activities designed to transform information, materials, and energy into a specified result.



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 and what kind of process discipline?”

Many organizations use Capability Maturity Model Integration (CMMI) models for process improvement.

What is CMMI? - 1

CMMI is a process improvement approach that provides organizations with essential elements of effective processes.

From an ancestral focus on software development and maintenance, CMMI models have evolved to cover other disciplines.

A CMMI *constellation* is a collection of CMMI components that includes a model, its training materials, and appraisal-related documents for an area of interest.

What is CMMI? - 2

A CMMI model contains the essential elements of effective processes.

CMMI for Development (CMMI-DEV) – V1.3 Nov 2010

- systems engineering
- software engineering

Other models

- CMMI for Acquisition – V1.3 Nov 2010
- CMMI for Services – V1.3 Nov 2010

For each model, there are two representations

- staged
- continuous

Staged Representation Models - Maturity Levels

Maturity levels, used in the staged representation, organize selected process areas into five evolutionary plateaus to support and guide process improvement across the organization.

Maturity levels are defined, evolutionary plateaus of process improvement in which each level is supported by the characteristics of the process areas implemented within the lower levels.

Maturity levels represent a process improvement evolution for the entire organization.

CMMI-DEV Process Areas (Staged)

Maturity Level	Process Areas
5 Optimizing	Organizational Performance Management 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 Risk Management Decision Analysis and Resolution
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-DEV 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 process area 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.”

Process Area Categorization (Continuous)

Category	Process Areas
Process Management	Organizational Process Focus
	Organizational Process Definition
	Organizational Training
	Organizational Process Performance
	Organizational Performance Management
Project Management	Project Planning
	Project Monitoring and Control
	Supplier Agreement Management
	Integrated Project Management
	Risk Management
	Quantitative Project Management
Engineering	Requirements Development
	Requirements Management
	Technical Solution
	Product Integration
	Verification
	Validation
Support	Configuration Management
	Process and Product Quality Assurance
	Measurement and Analysis
	Decision Analysis and Resolution
	Causal Analysis and Resolution

Session Topics

Process improvement and CMMI models

Building on CMMI-based process improvement

Building on a software process improvement infrastructure

Summary

CMMI and Framework Comparisons - 1

Area of Comparison	CMMI-DEV	Framework
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	SCAMPI	PLQL PLTP

CMMI and Framework Comparisons - 2

Area of Comparison	CMMI-DEV	Framework
Contains "How To"	No*	Yes
De facto standard	Yes	No
Maturity Levels	Yes (staged)	No
Capability Levels	Yes (continuous)	No

** There is limited, generic "how to" in the informative material.*

Process Areas (CMMI-DEV) and Practice Areas (Framework)

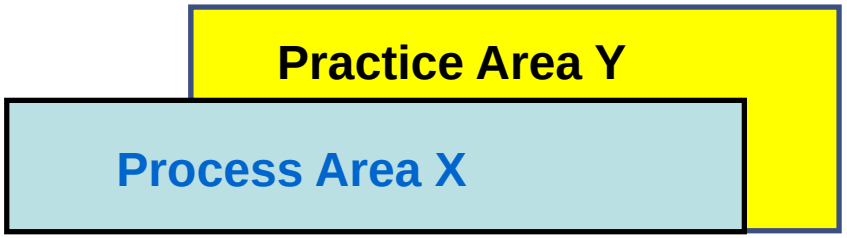
The most appropriate units for detailed comparison

- CMMI-DEV Process Areas
  - Describe where an organization should have processes
  - 22 within CMMI-DEV
- Framework Practice Areas
  - Describe where an organization should have expertise (sometimes this includes processes)
  - 29 within the Framework

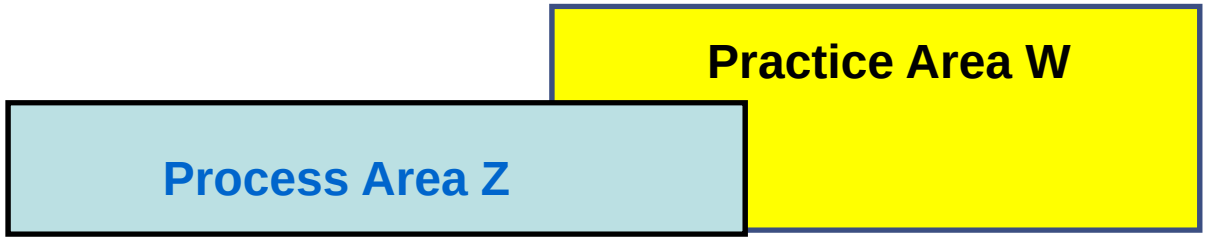
Process Areas and Practice Areas

Certain CMMI-DEV 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-DEV Process Areas	Framework Practice Areas
Configuration Management	Configuration Management
Requirements Management	Configuration Management
Project Planning	Technical Planning
Organizational Training	Training
Measurement and Analysis	Measurement and Tracking
Risk Management	Technical Risk Management
Decision Analysis & Resolution	Make/Buy/ Mine/Commission Analysis
Technical Solution	Make/Buy/ Mine/Commission Analysis

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

CMMI-DEV Process Areas	Framework Practice Areas
Organizational Process Definition	Process Discipline
Supplier Agreement Management	Acquisition Strategy, Using Externally Available Software, Make/Buy/Mine/ Commission Analysis
Project Monitoring and Control	Measurement and Tracking
Project Planning	Organizational Planning
Requirements Development	Requirements Engineering
Risk Management	Organizational Risk Management
Technical Solution	Architecture Definition, Component Development, Using Externally Available Software
Product Integration	Software System Integration
Verification	Testing, Architecture Evaluation
Validation	Testing

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

CMMI-DEV Process Areas	Framework Practice Areas
Integrated Project Management	Measurement and Tracking, Customer Interface Management, Structuring the Organization
Organizational Process Definition	Structuring the Organization
Organizational Performance Management	Technology Forecasting

In CMMI-DEV, but not addressed explicitly in Framework

Organizational Process Focus

Process and Product Quality Assurance

The following CMMI-DEV 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 CMMI-DEV

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

Architecture Practices in CMMI V1.3

For Version 1.3, CMMI provides better coverage of architecture-centric practices (mostly by changes to informative material).

Elevates quality attribute requirements to equal importance with functional requirements

- SP 3.2 Establish a Definition of Required Functionality and Quality Attributes

CMMI V1.3 also provides an improved terminology to support understanding and use of architecture-centric practices

- Updated the glossary to include new terms (and modified some old terms)
- Updated the informative material (especially Requirements Development, Technical Solution, and Verification in CMMI-DEV) to:
  - make use of the new terms
  - bring more emphasis to quality attributes and thus strike a better balance between functional and non-functional requirements
- Replaced selected uses of overloaded terms such as “performance” with an appropriate qualifying phrase.

Architecture and Product Lines in CMMI V1.3

Added and revised the informative material throughout the Engineering Process Areas (in particular) to appropriately mention the following engineering concepts:

- quality attributes (i.e., non-functional requirements or “ilities”)
- architecture-centric practices
- product lines, system of systems

These concepts are mentioned in example boxes, in examples provided in the notes, and in discussions that mention various approaches that can be used.

When functional requirements are discussed, mention of quality attributes is added to balance the view of requirements.

Which CMMI-DEV Model Representation Supports Software Product Lines?

Product line practice is supported by both CMMI-DEV 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-DEV Process Areas
- cross-project strategic business processes not address by CMMI-DEV

Leveraging CMMI-DEV Process Areas to Software Product Lines

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

- Requirements Management
- Project Planning
- Project Monitoring and Control
- Configuration Management
- Requirements Development

It would be **even more useful to** be able to standardize across the organizational unit (Capability Level 3).

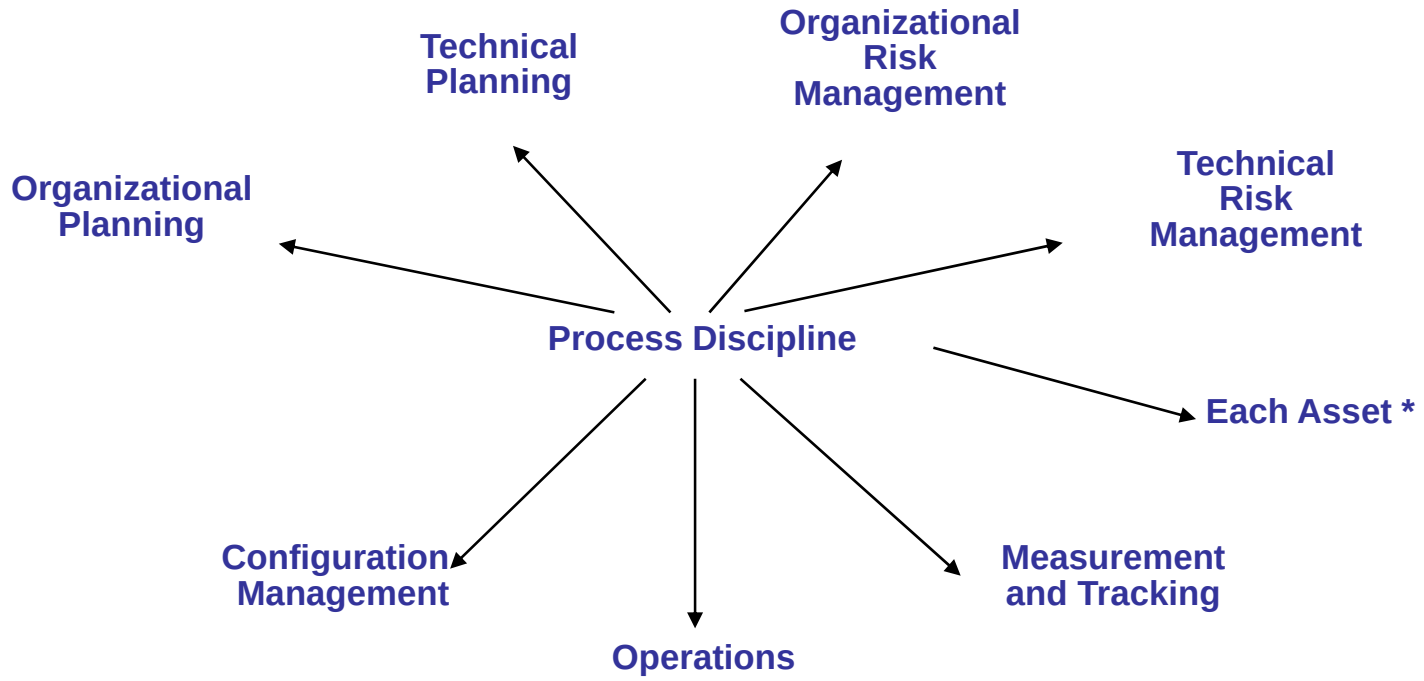
Even if you have mature CMMI-DEV processes in place, product line processes **always have special aspects,** many with process implications.

The Process Pattern

The ***Process*** pattern consists of Framework practice areas that should be used to support product line activities that require processes.

If you are choosing which CMMI-DEV process areas to implement, consider the process areas that correspond to these Framework practice areas.

Process Pattern



Dynamic Structure

But There's More …

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

For each practice area, these special aspects are found in the Framework under

- Aspects Peculiar to Product Lines
- Application to Core Asset Development
- Application to Product Development

Example: Configuration Management - 1

CMMI-DEV puts an emphasis on what to do.

CMMI-DEV 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, for example,

- 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.

Building Your Processes to Support Software Product Lines

In a manner similar to the CM example

- Compare the practices in the Framework to those in the CMMI-DEV. (See previous slides for rough mappings.)
- Identify gaps in current process.
- Add appropriate process aspects to address gaps and support product line practice.

Session Topics

Process improvement and CMMI models

Building on CMMI-based process improvement

Building on a software process improvement infrastructure

Summary

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

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 training 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 include:

- 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

Session Topics

Process improvement and CMMI models

Building on CMMI-based process improvement

Building on a software process improvement infrastructure

Summary

Summary

When adopting a software product line approach you seldom start with a blank sheet of paper.

Process improvement can provide a good basis for product line adoption.

- The organization will be better inclined to follow product line processes.
- An existing process improvement infrastructure can help address typical technology change issues.

CMMI models can provide a foundation for product line practice but more product line-specific practices and guidance will always be necessary.

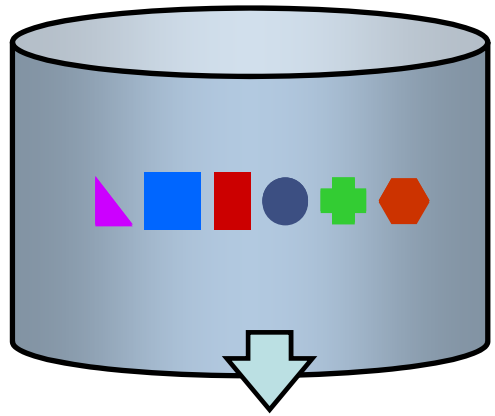
Factors Influencing Adoption

Organizational Context

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

Adoption Support

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



Product Line Adoption Plan

. . .

Product Line Action Plans

Software Engineering Institute | Carnegie Mellon

Adopting Software Product Lines

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Part 4: Exploiting Other Current Activities
Module 10: Architecture-Centric Development and Hardware Platform Engineering

Session Outcomes

After this session participants should

- know some ways that an existing architecture-centric development approach can be exploited for product line adoption
- know some ways that an existing hardware platform engineering approach can be exploited for product line adoption

Product Line Adoption and Product Engineering

Many organizations considering product line adoption already have engrained product engineering practices.

You can build on these practices and in some cases use them to accelerate software product line adoption.

In this session we will briefly examine how to build on

- architecture-centric development activities
- hardware platform engineering

Session Topics

Building on architecture-centric practices

Building on hardware platform engineering

Summary

Software Architecture - 1

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

[1] Bass, L.; Clements, P. & Kazman, R. *Software Architecture in Practice, 2nd Edition.* Reading, MA: Addison-Wesley, 2003.

Software Architecture - 2

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

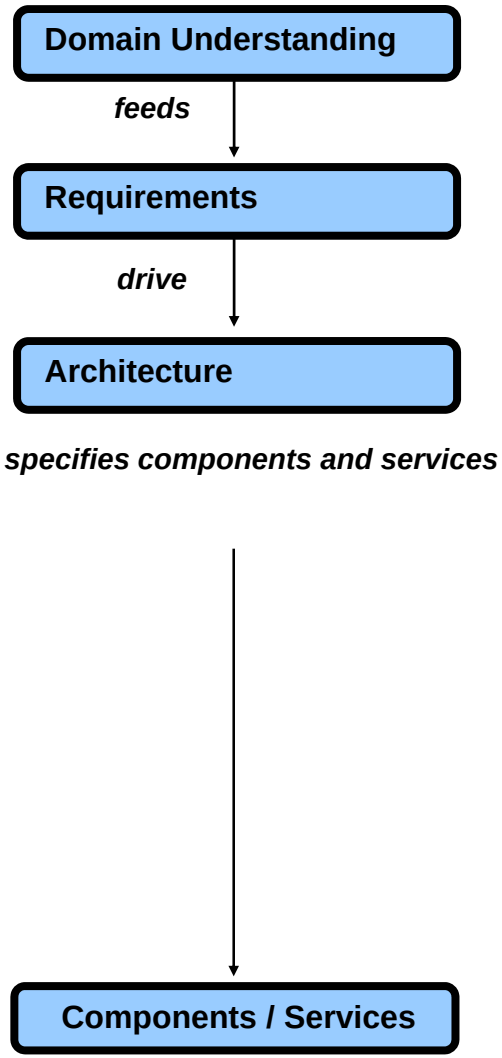
Direct linkages to the following practice areas in the Framework include:

- Building a Business Case
- Requirements Engineering
- Architecture Definition
- Architecture Evaluation
- Component Development
- Testing

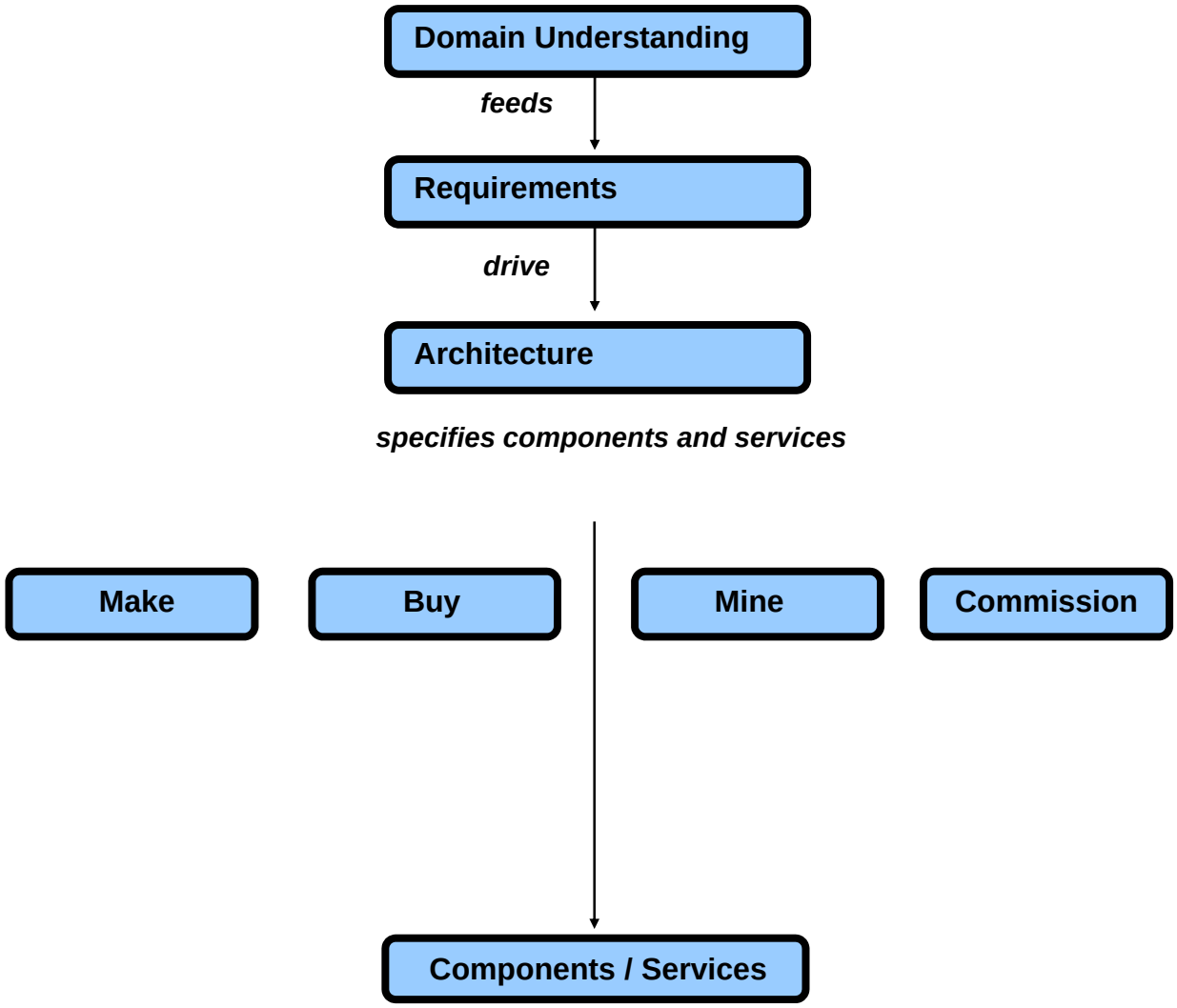
There are also weaker linkages with

- Mining Existing Assets
- Using Externally Available Software
- Software System Integration

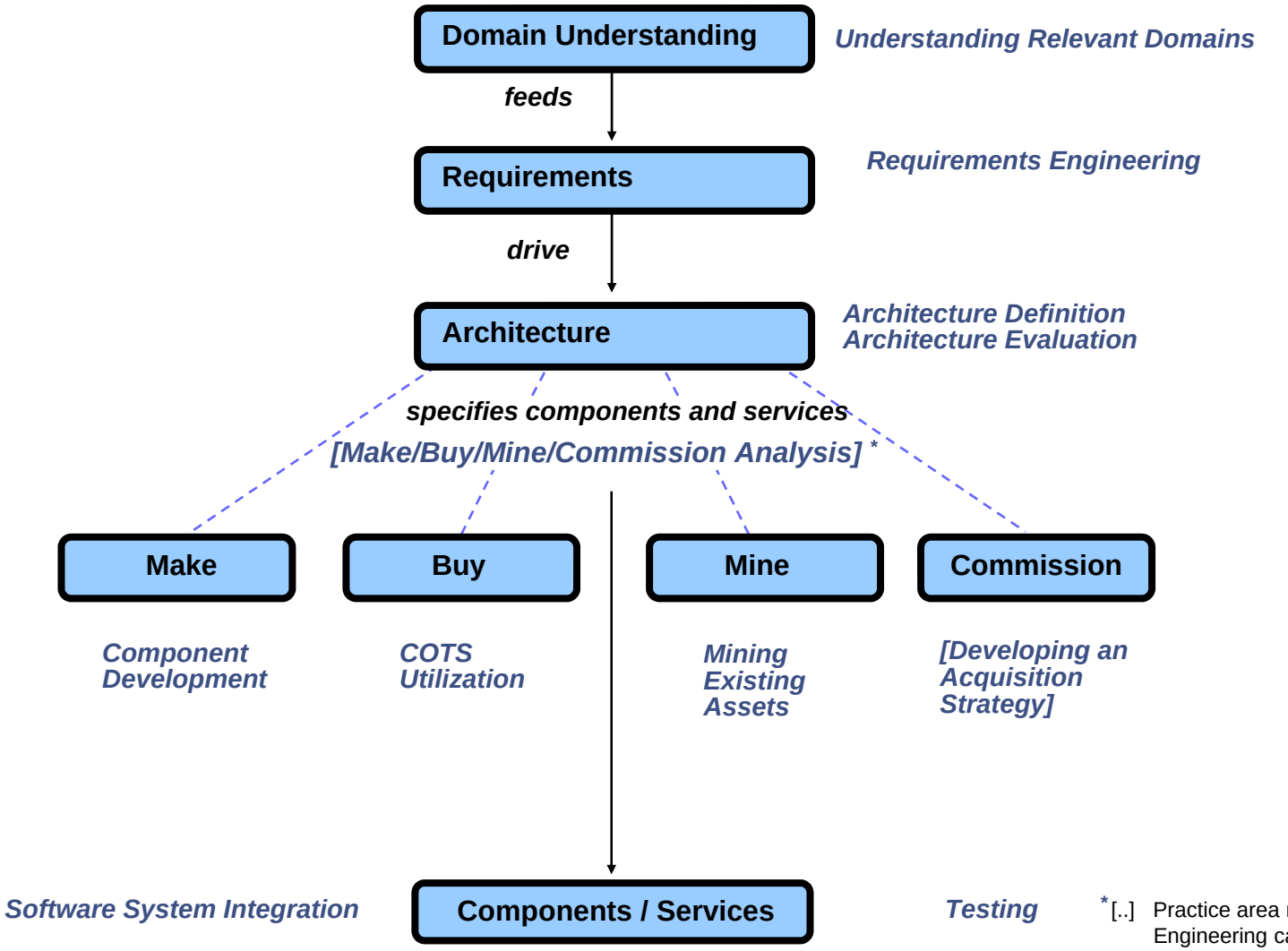
Practice Area Relationships



Practice Area Relationships

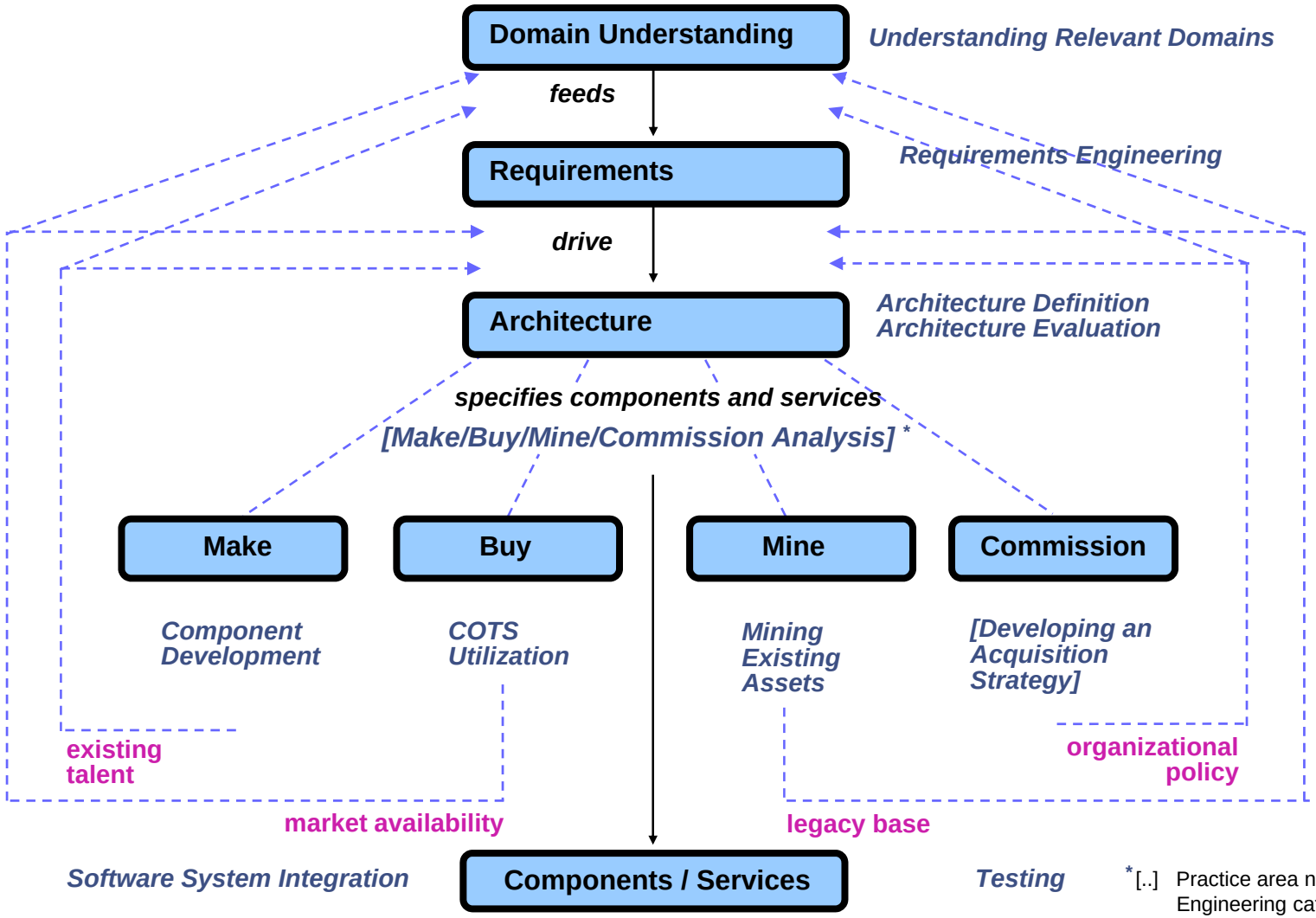


Practice Area Relationships



*[..] Practice area not in Software Engineering category.

Practice Area Relationships



*[..] Practice area not in Software Engineering category.

Influence on Adoption Factory

** Strong support
* Weak support

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

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.

Adapting Architecture-Centric Practices - 1

Creating the business case for the system

- The **business case** must justify the product line and not just a single system. It might also need to justify use of the product line approach.

Understanding the **requirements**

- Product line requirements will need to specify common requirements and variation points.
- Product-specific requirements will need to be aligned with the product line requirements.
  - match common requirements
  - specify variation requirements for variation points

Adapting Architecture-Centric Practices - 2

Creating and/or selecting the architecture

- The product line architecture must
  - apply to all members of the product line (even if their functions and qualities differ)
  - embody the commonalities and variations of the family members
  - include specific mechanisms for variation
- Each product architecture will be an instantiation of the product line architecture.

Adapting Architecture-Centric Practices - 3

Documenting and communicating the architecture

- The product line architecture needs to be communicated to the entire product line organization.
- Its documentation must have views for all of these stakeholders.
- In particular, the documentation must make variation mechanisms clear.

Analyzing or evaluating the architecture

- The product line architecture must be evaluated to determine its fit for structuring all the members of the product line (current and future).
- Each product architecture must be evaluated to determine its fitness for the product's unique intended purpose.
- Of particular interest, is support for variability and the inherent tradeoffs with other quality attribute requirements.

Adapting Architecture-Centric Practices - 4

Implementing the system based on the architecture

- Components that will be core assets must be developed to align with the commonality and the variation mechanisms specified in the product line architecture.
- Components that are product-specific must be developed in accordance with the dictates of the product architecture, which is instantiated from the product line architecture.

Ensuring that the implementation **conforms** to the architecture

- Processes and/or tools must be used to ensure that both the product line and product-specific architectures are adhered to in all code development.

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

Session Topics

Building on architecture-centric practices

Building on hardware platform engineering

Summary

Product Lines and Hardware Engineering

A product line approach to software was inspired by product line approaches in manufacturing.

Though the Framework and the product line practice patterns were written for software product line practice, in structure both are entirely applicable to

- non-software product lines
- product lines that are combinations of software and hardware

The essential practice areas are not different.

How they apply to hardware are different than how they apply to software.

Hardware Platform Engineering - 1

The use of a product line approach for hardware is often referred to as **hardware platform engineering.**

- The platform is the name for the hardware core assets from which products are made.
- The platform can be configured differently for different products in the product line.

Hardware Platform Engineering - 2

The following aspects of a hardware platform engineering environment could be exploited for product line adoption:

- product line concepts (conceptual underpinnings)
- product line mentality (cultural underpinnings)
- practices from the practice areas in the Organizational focus area of the *Adoption Factory* pattern, which includes
  - *Cold Start* pattern
  - *Monitor* pattern
  - *In Motion* pattern
- process architecture

Hardware Platform Engineering and the Adoption Factory Pattern

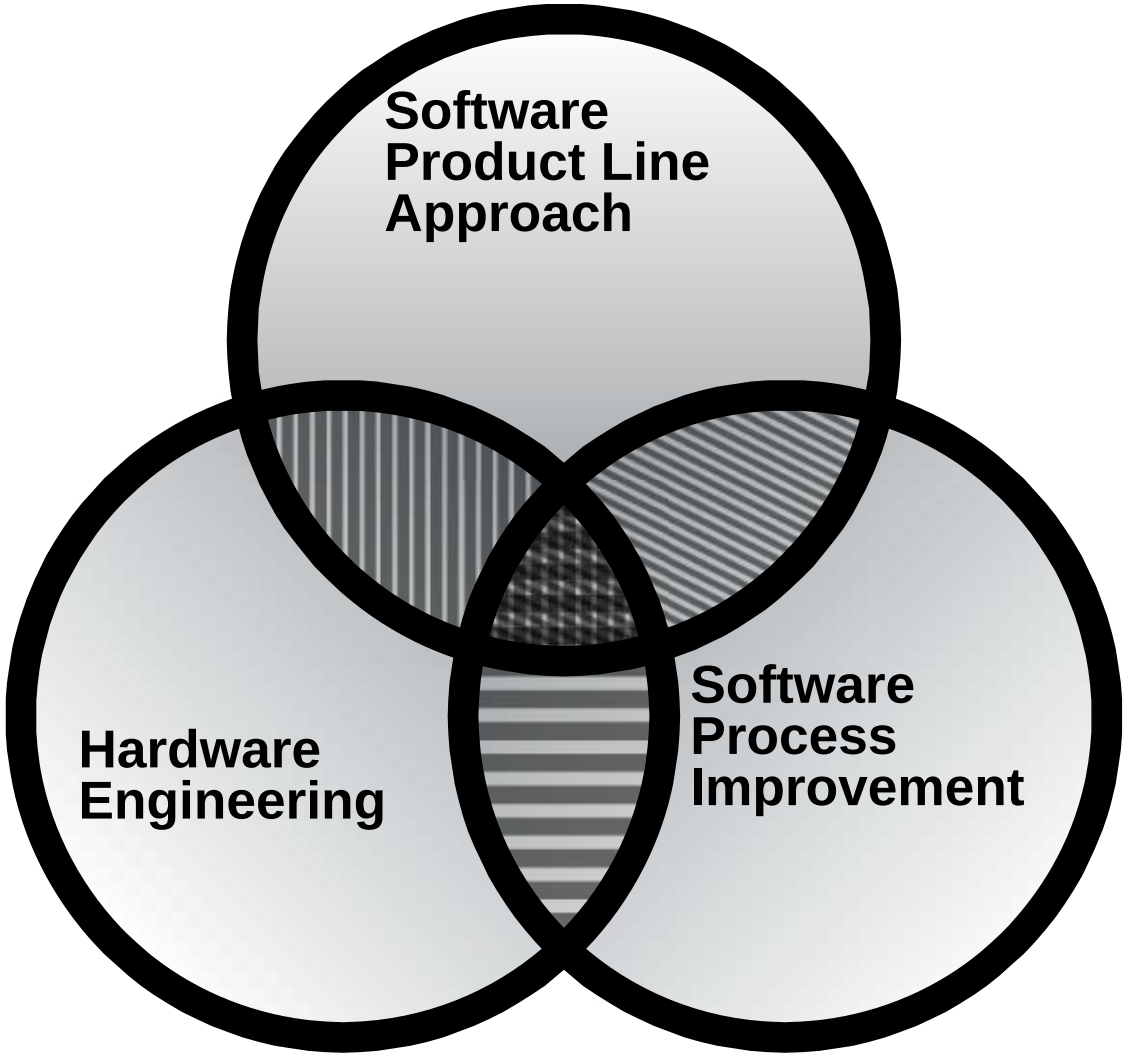
Practice areas are the same but in some cases the practices are very different.

Influence on Adoption Factory

** Biggest practice deviation

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

Overlapping Activities - 1



Overlapping Activities - 2

Many of the activities involved in software engineering process improvement have relevance in hardware engineering.

Many of the software product line practices can be applied to hardware when the systems being developed involve software embedded in hardware.

The overlaps can be rich areas to exploit these other current activities during product line adoption.

Discussion

There are other current activities that can provide a foundation for software product line adoption.

- What are they?
- How do they relate to software product line practice?
- What can you exploit during product line adoption?

Session Topics

Building on architecture-centric practices

Building on hardware platform engineering

Summary

Summary

The product line architecture is one of the key assets in the core asset base for a software product line.

If an organization uses architecture-centric practices for its software, these practices provide an excellent starting position for the software engineering practice areas necessary for software product line adoption.

Hardware platform engineering is a product line approach for hardware.

- The associated practices, knowledge, and culture can jumpstart software product line adoption.

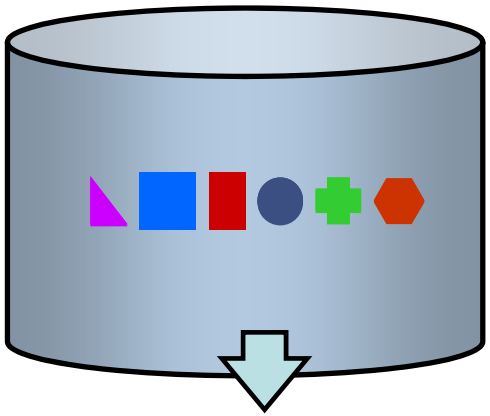
Factors Influencing Adoption

Organizational Context

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

Adoption Support

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



Product Line Adoption Plan

. . .

Product Line Action Plans

Software Engineering Institute | Carnegie Mellon

Adopting Software Product Lines

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Module 11: Course Wrap-Up

Software Product Lines

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.**

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.

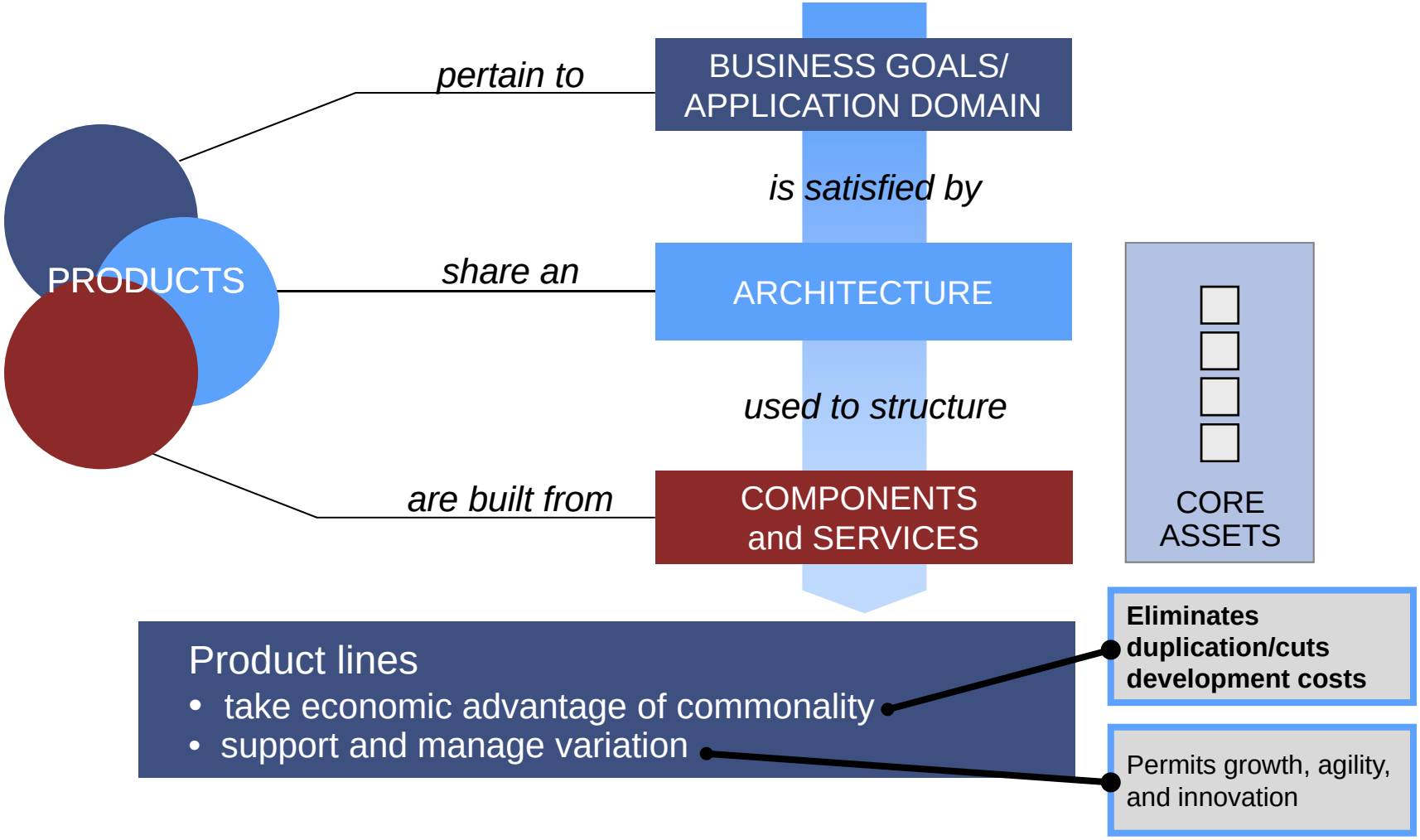
The Adoption Endgame

To have an **operational software product line.**

To do that, an organization must

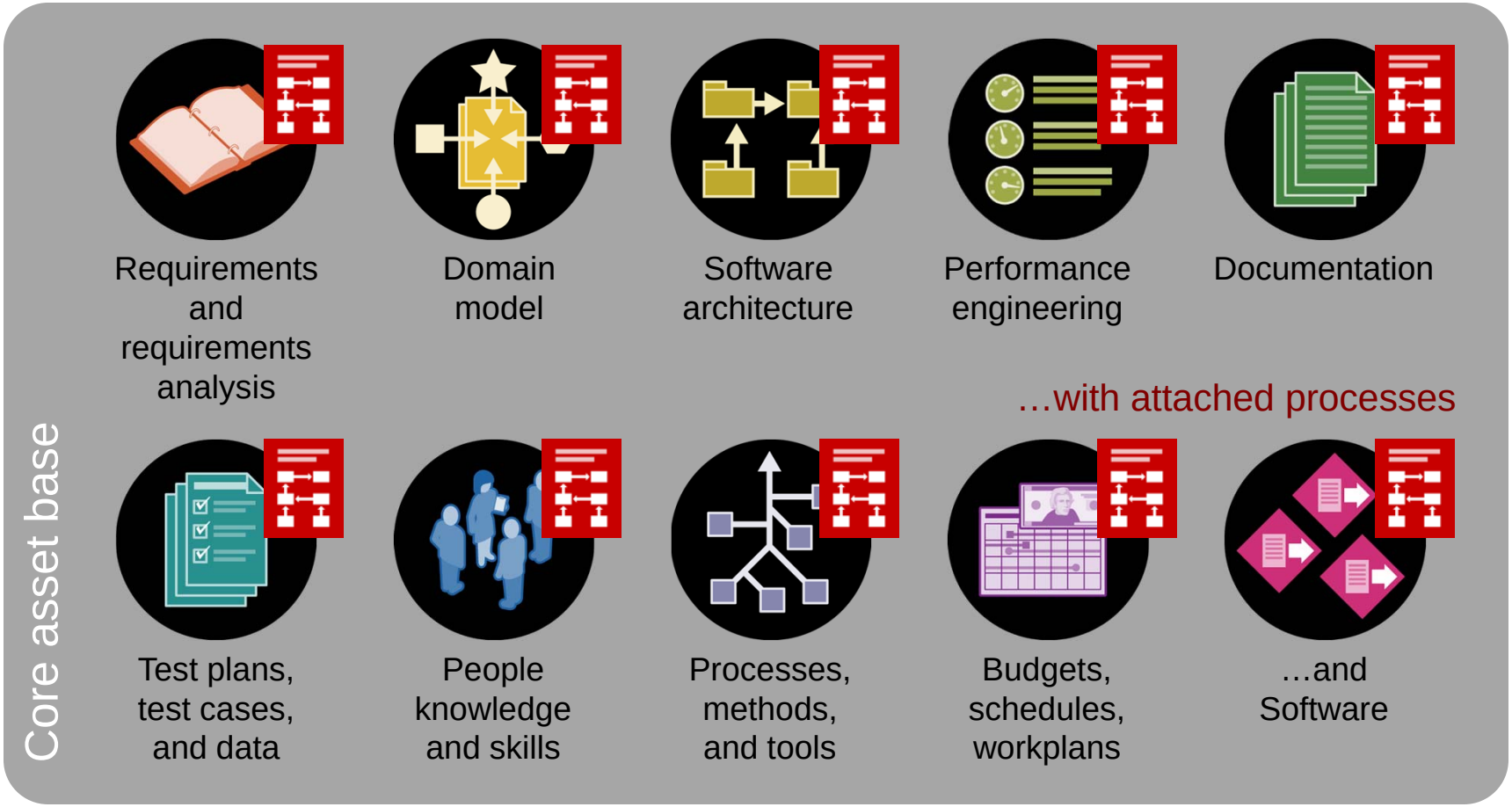
- have
  - a core asset base
  - supportive processes and organizational structures
- develop products from that asset base in a way that achieves business goals
- prepare the organization to institutionalize product line practices

Software Product Lines

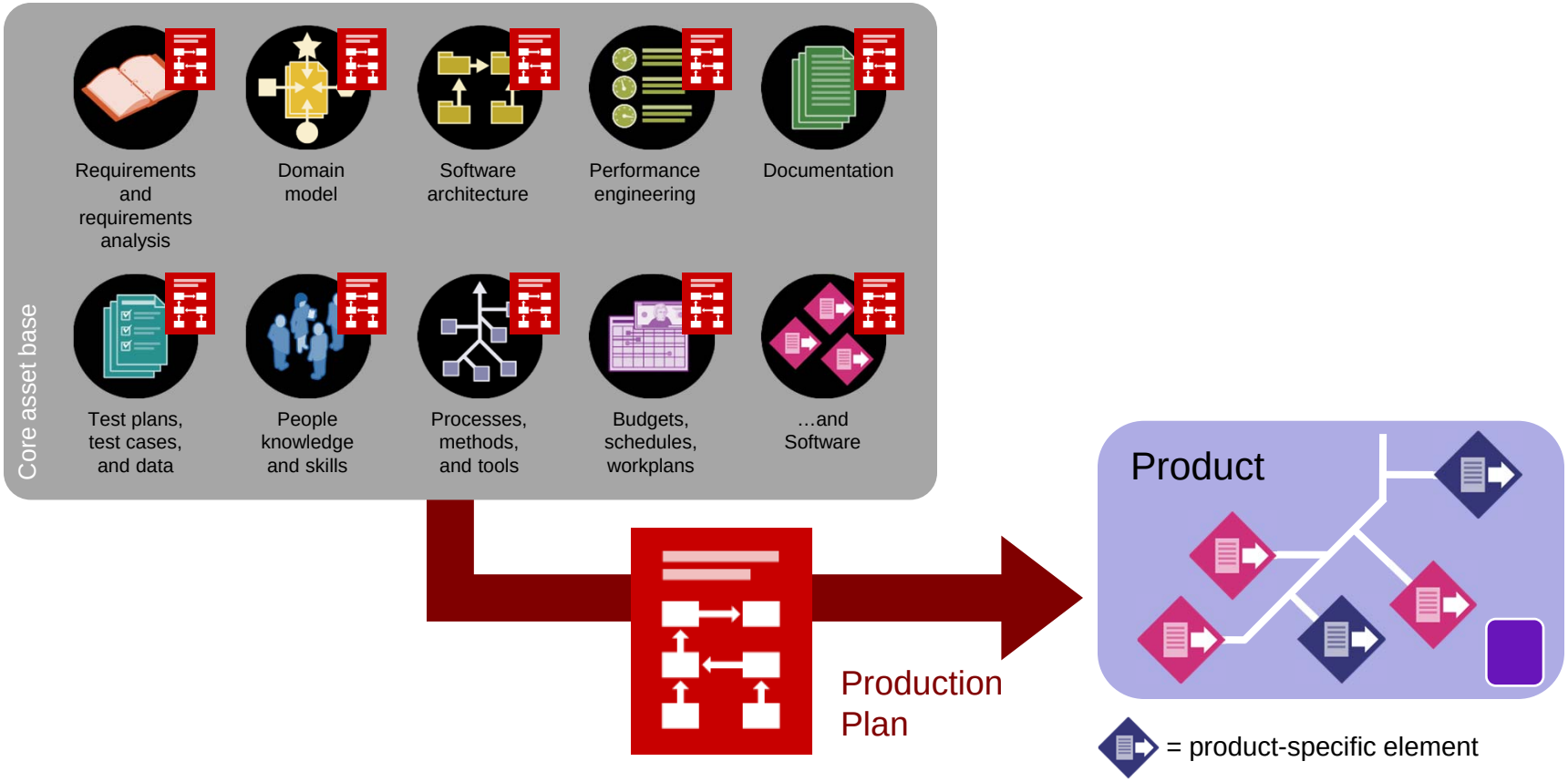


Building the Core Asset Base

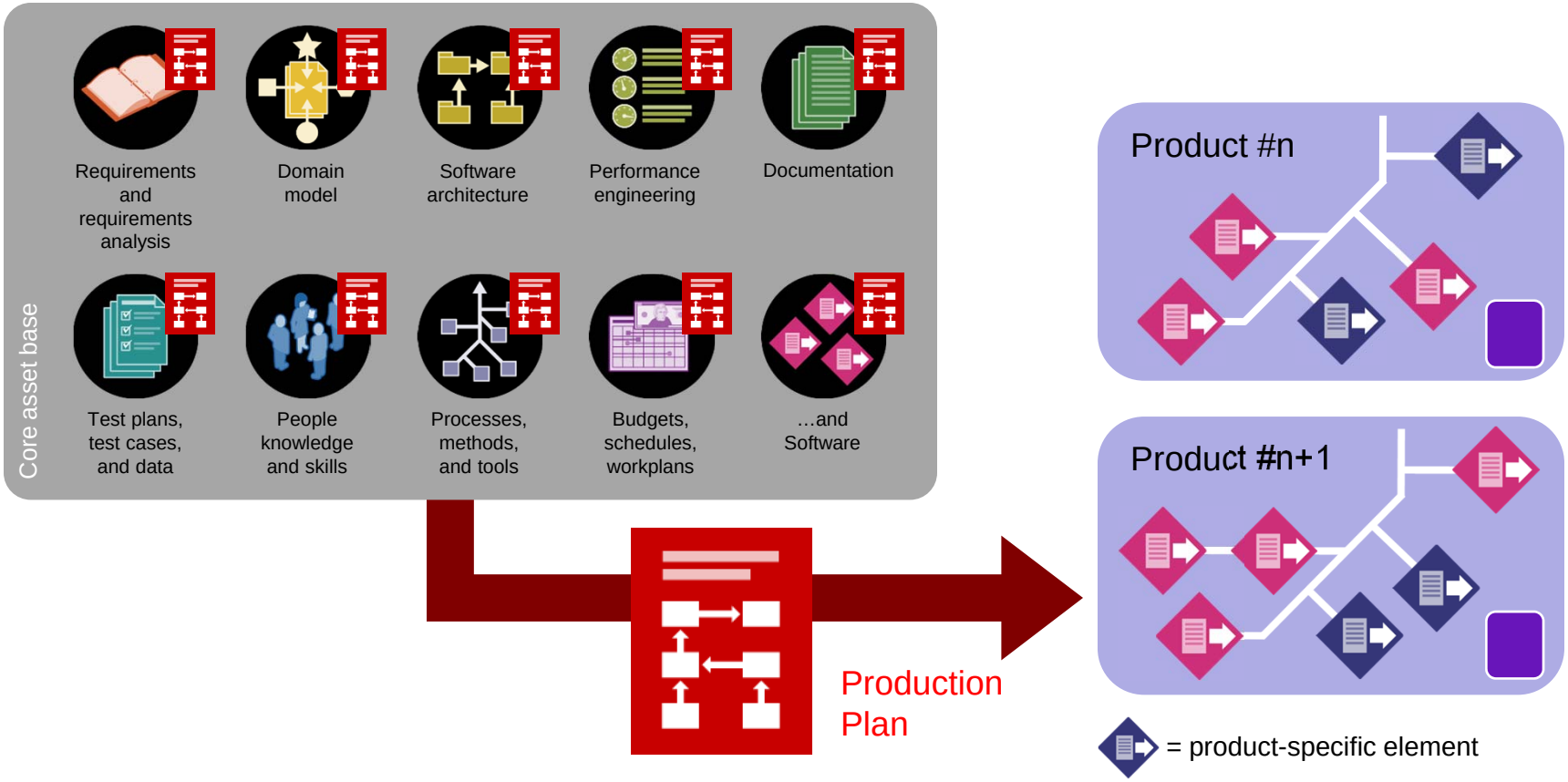
Core assets include:



Building a Product…

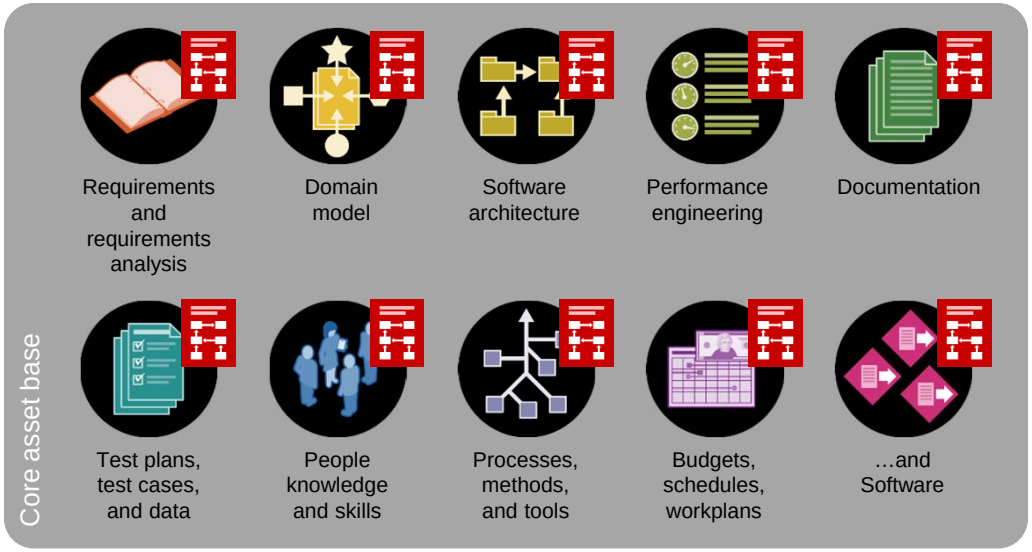


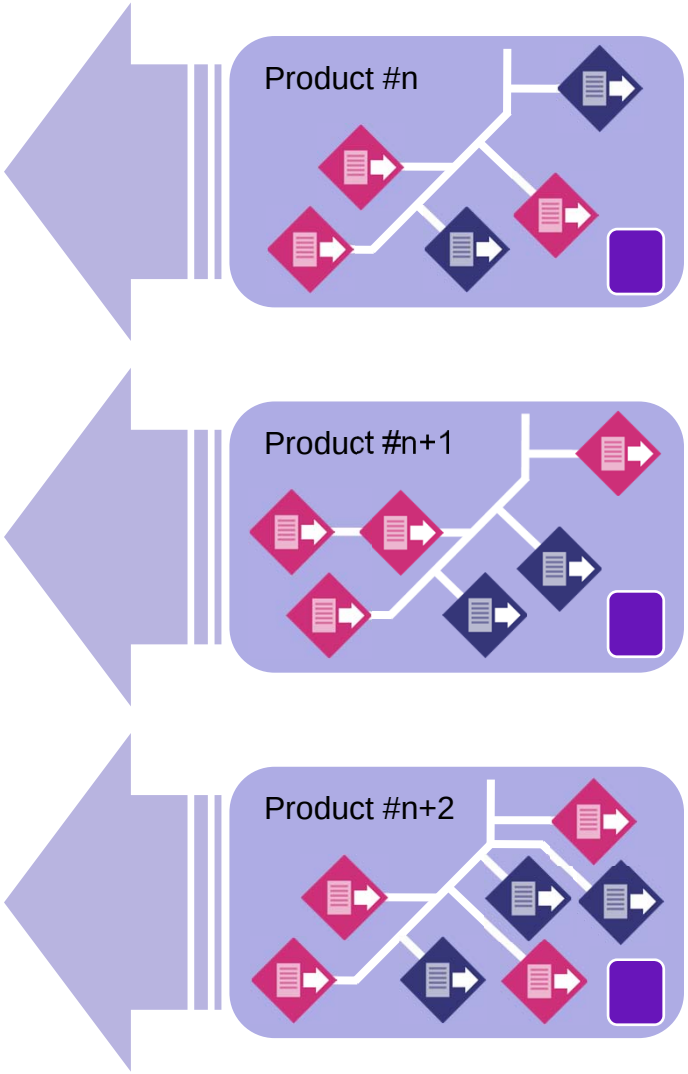
Building Subsequent Products…



Feedback

Updates to core assets





Pitfalls

Some pitfalls seem to challenge all would-be product line organizations.

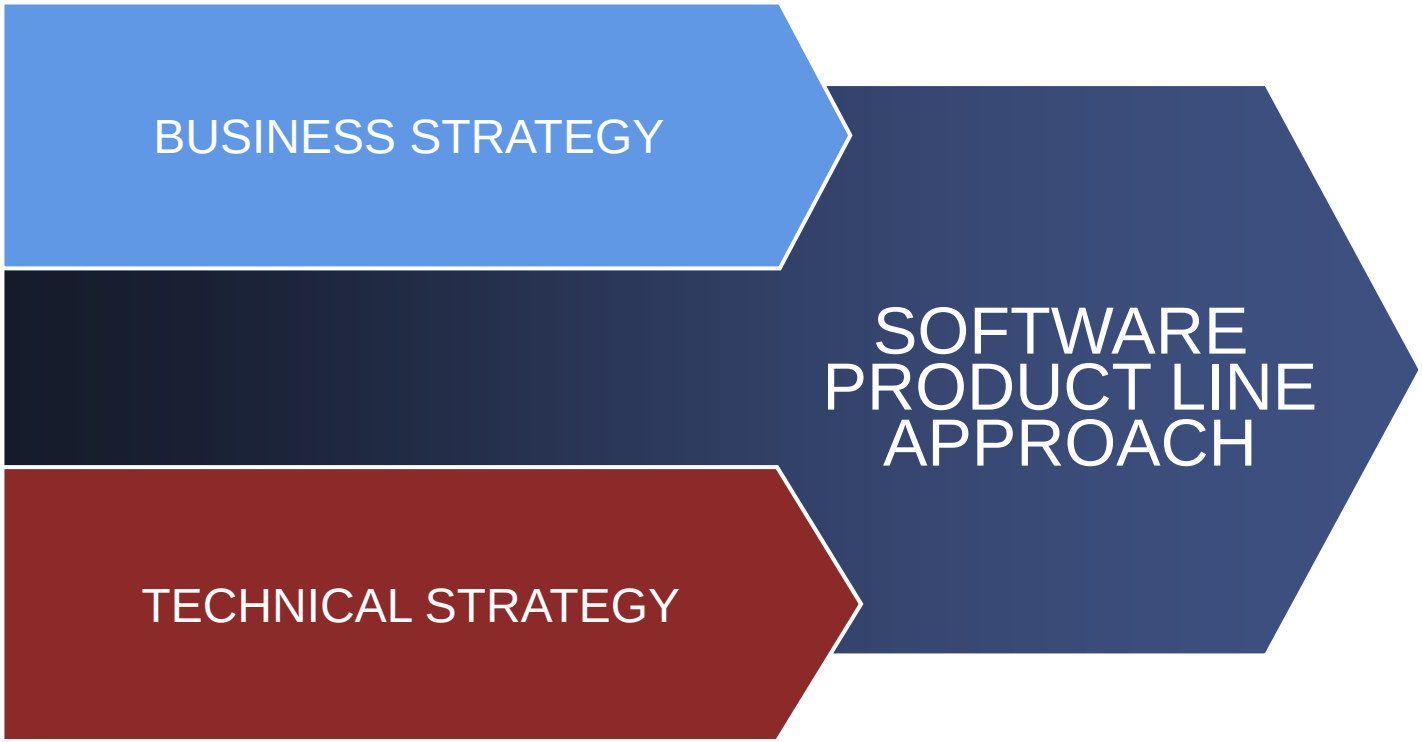
Some are most typically associated with certain organization types.

- Engineering-based organizations
- Developer-focused organizations
- Government acquisition organizations

Two Strategic Pitfalls

1. Failure to recognize that a software product line approach is both a business and technical strategy
2. Failure to manage the product-line-unique aspects of both governance and roll out appropriately

Strategic Reuse for Business Benefits



Sound Business Strategy

There is a business case with incentives at all levels.

Key individuals from the business side of the organization are involved.

Business practices are aligned with the product line approach.

All levels of managers are appropriately involved.

Sound Technical Strategy

Technical staff have software engineering skills and know how to apply them in a product line context.

There is a dominant architecture focus.

There are variation mechanisms that are based on business goals and a business case and are reasonably resilient to the possible impact of emerging technologies.

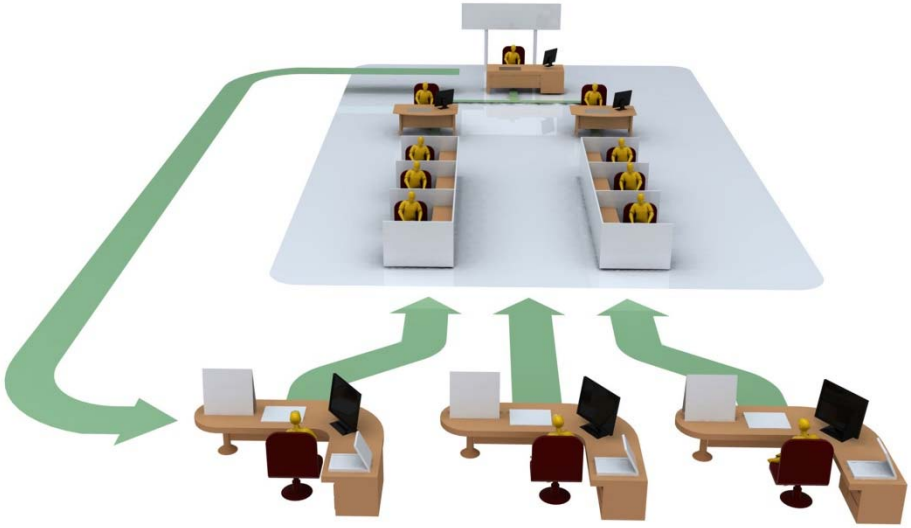
There is a production strategy and a production plan that ensures repeatable and predictable application of the chosen variation mechanisms.

There are non-code core assets.

Appropriate Governance

More than an organization chart and a workflow diagram are required.

- The organizational structure and its associated operating concept must support the product line.
- Both the business and the technical dimensions of the product line effort have to be coordinated and managed.
- There must be open communication and knowledge transfer throughout the organization



Effective Rollout Strategy - 1

Know what you need to do in order to move to a product line effort.

- SEI Framework or other similar models can help.
- Tips for using the SEI Framework
  - Practice area does not equal process area.
  - All practices areas don't play at the same volume.
  - People assigned to carry out the product line tasks must have the requisite skills to do so.

Have a plan or roadmap for how you will phase in the product line activities.

- The Adoption Factory pattern can help
  - Set the product line context.
  - Establish the production capability.
  - Operate the product line.
  - Recognize there will be iteration.

Effective Rollout Strategy - 2

Include change management mechanisms in your roll out strategy.

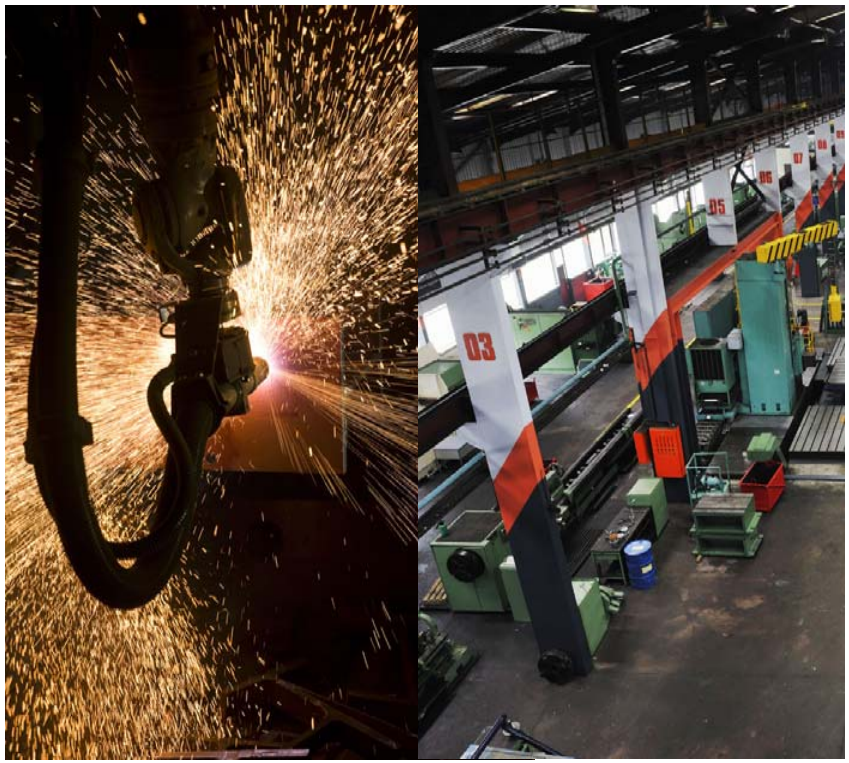
- Manage the “people” aspects of change.
- Use proven change management models and techniques that fit your organizational culture.

Make sure you know how to stay in business while effecting the necessary changes.

Engineering-Based Organizations

Organizations where

- The products in the product line have both hardware and software components
- The individuals involved in the product line have an engineering background



Pitfalls in Engineering-Based Organizations

Core assets lack business relevance

Lack of organizational support and visibility

Technical thrashing among the software engineers

Cost overruns for the product line ramp-up

Why These Pitfalls?

Engineers tend to tackle the technical activities either first or to the exclusion of business activities.

- Only code core assets are considered.

Engineers often have an inability to bring the rest of the organization on board with the effort.

Engineers might make decisions based on technical merits without sufficiently considering relevant business needs, goals, and strategies.

- They might not even know the business goals and strategic directions.

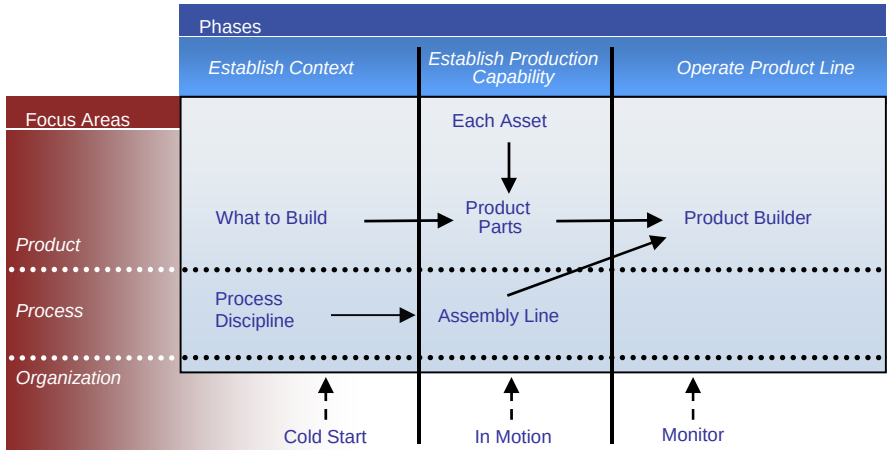
What's the Remedy?

Be sure to establish the product line context.

- Define the product line scope and its supporting business case.
- Base variations and variation mechanisms on business needs.

Pay proper attention to the organizational practice areas.

Make sure that the engineers engage their business counterparts.



Developer-Focused Organizations

Organizations

- That produce software applications and enterprise systems.
- Have typically not grown out of a hardware or engineering culture, but rather an IT culture.

Pitfalls in Developer-Focused Organizations

Lack of strategic technical focus

- Focus is on technology and code reuse.
- Product line architecture and production plan are foreign concepts.

Lack of consistently applied software engineering practices across teams

Stagnant integration and deployment efforts

Lack of technical agreement across team boundaries

Waffling commitment to the product line endeavor

Why These Pitfalls?

Usually

- The business leaders drive the product line effort.
- An assortment of fairly autonomous technical teams with a code-level focus do the technical work.
- These teams
  - Vary in software engineering skills and practices
  - Focus exclusively on the code and functionality
  - Use different definitions for key concepts (release, build, test, architecture, etc.)
  - Love their autonomy

What's the Remedy?

Embrace more disciplined technical practices that involve.

- A modest injection of software engineering in key areas
- A more strategic view

Pay attention to organizational change management principles.

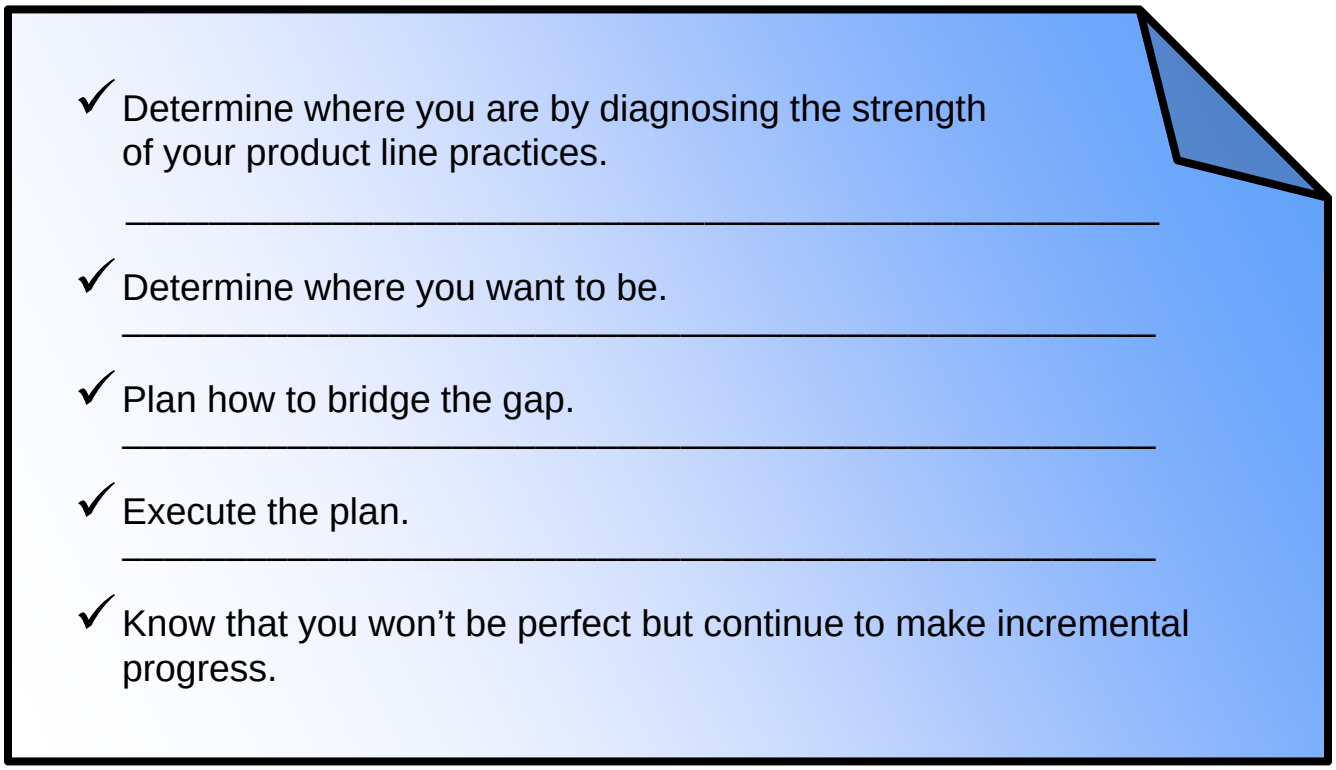
- Establish a vision that speaks to all levels of the organization.
- Make sure that interfaces across team and process boundaries are well defined.
- Establish common terminology and quality expectations.

In any case, don’t panic!

Learn from the mistakes of others.

The first step in getting out of a predicament is to recognize it.

Then



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:

- generic guidance
  - for product lines
  - for technology change
- organizational context

Factors Influencing Adoption

Organizational Context

product line readiness

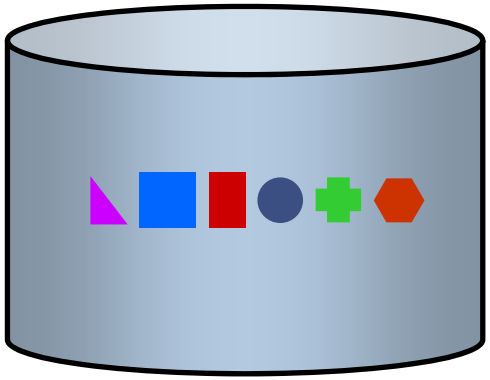
barriers

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

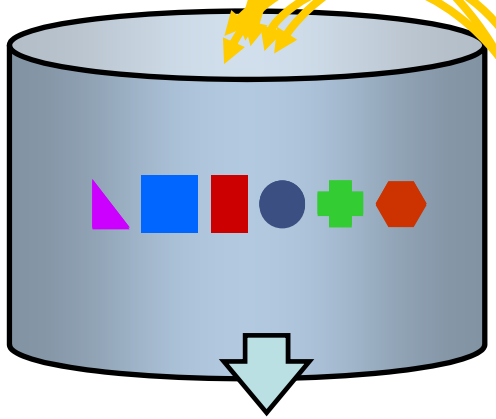
product line adoption roadmap

product line approaches

change models

change management mechanisms

planning process



Product Line Adoption Plan

. . .

Product Line Action Plans

Course in Review

We have examined all the 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 makes business sense.

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