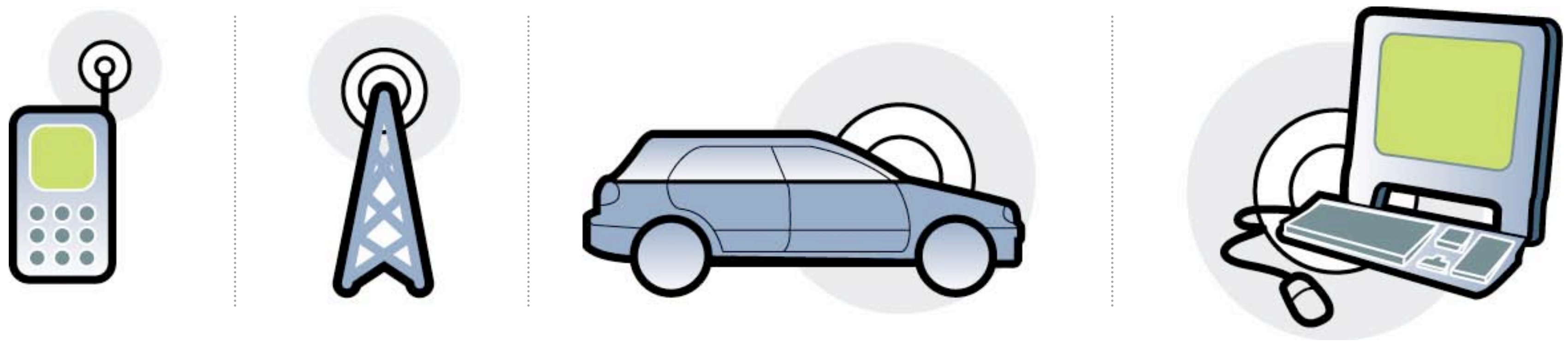
Software Product Lines: Reuse That Makes Business Sense

Linda Northrop

ASWEC 2006

BUSINESS SUCCESS REQUIRES SOFTWARE PROWESS



Software pervades every sector.

Software has become the bottom line for many organizations who never envisioned themselves in the software business.

UNIVERSAL NEEDS

- Deploy new products (services) at a rapid pace
- Accommodate a growing demand for new product features across a wide spectrum of feature categories
- Connect products in increasingly unprecedented ways
- Exploit a rapidly changing technology base
- Gain a competitive edge

UNIVERSAL BUSINESS GOALS

HIGH QUALITY

QUICK TIME TO MARKET

EFFECTIVE USE OF LIMITED RESOURCES

PRODUCT ALIGNMENT

LOW COST PRODUCTION

LOW COST MAINTENANCE

MASS CUSTOMIZATION

MIND SHARE

IMPROVED EFFICIENCY & PRODUCTIVITY

THE ULTIMATE UNIVERSAL GOAL

SUBSTANTIAL

QUICK

SUSTAINABLE

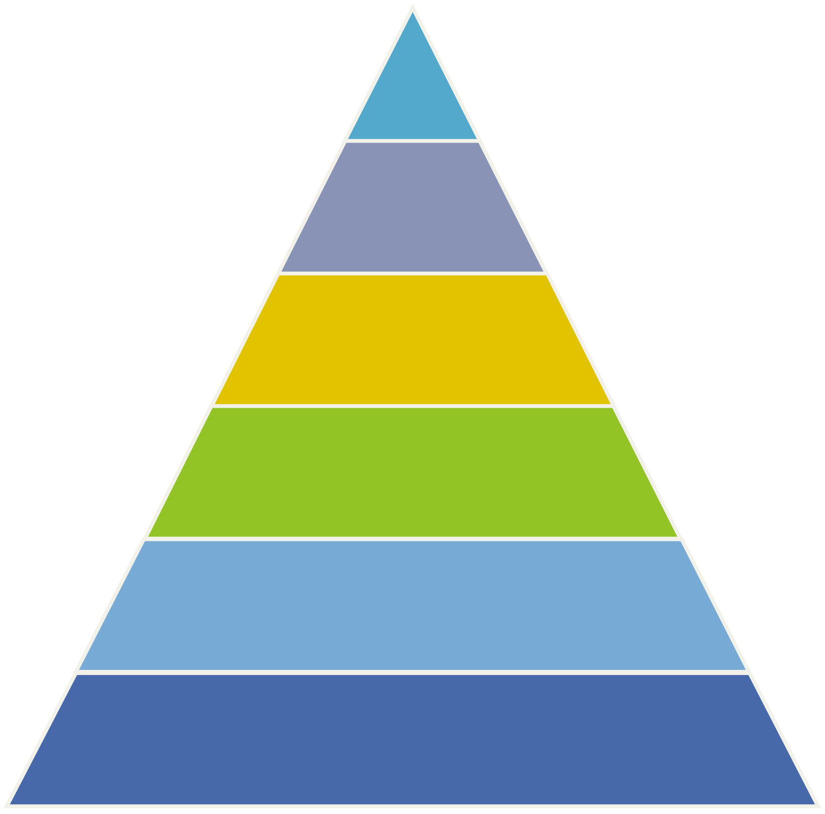
PROFIT

SOFTWARE (SYSTEM) STRATEGIES

PROCESS IMPROVEMENT

TECHNOLOGY INNOVATION

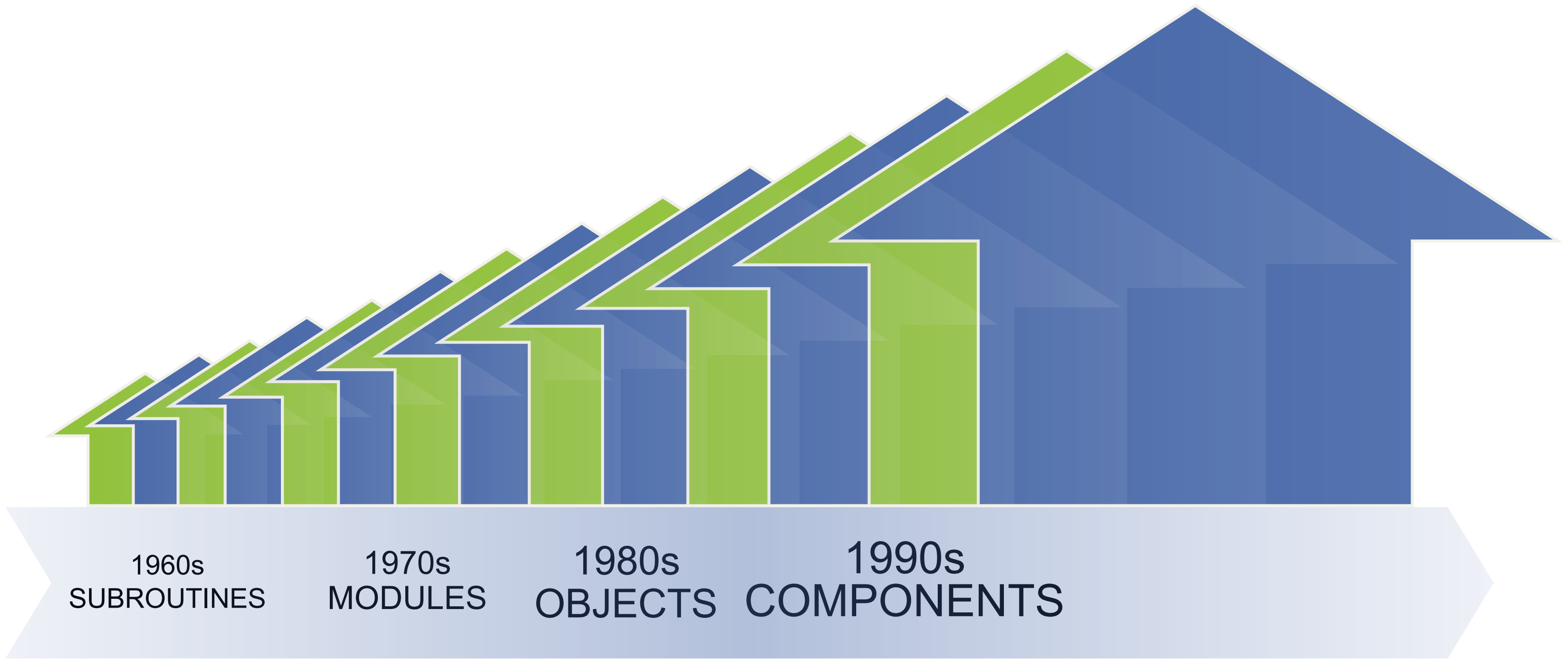
REUSE



FEW SYSTEMS ARE UNIQUE

Most organizations produce families of similar systems, differentiated by features.

REUSE HISTORY



Focus was small-grained and opportunistic.
Results fell short of expectations.

BUT WHAT IS REUSE?

REUSE MEANS TAKING SOMETHING DEVELOPED FOR ONE SYSTEM AND USING IT IN ANOTHER.

“The Army XYZ System is built with 80% reuse.”

A statement like this is vacuous.

- It is not clear what is being reused.
- It is not clear that the “reuse” has any benefit.

Reusing code or components without an architecture focus and without pre-planning results in

- short-term perceived win
- long-term costs and problems

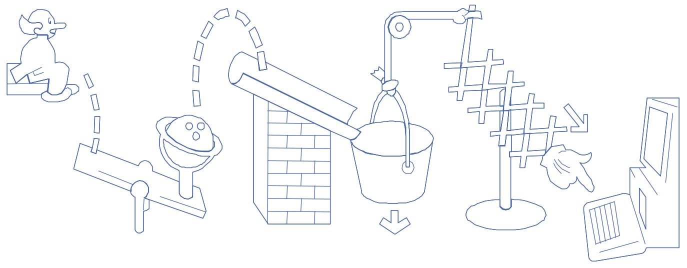
SOFTWARE REUSE FACT AND FICTION

THE FICTION:

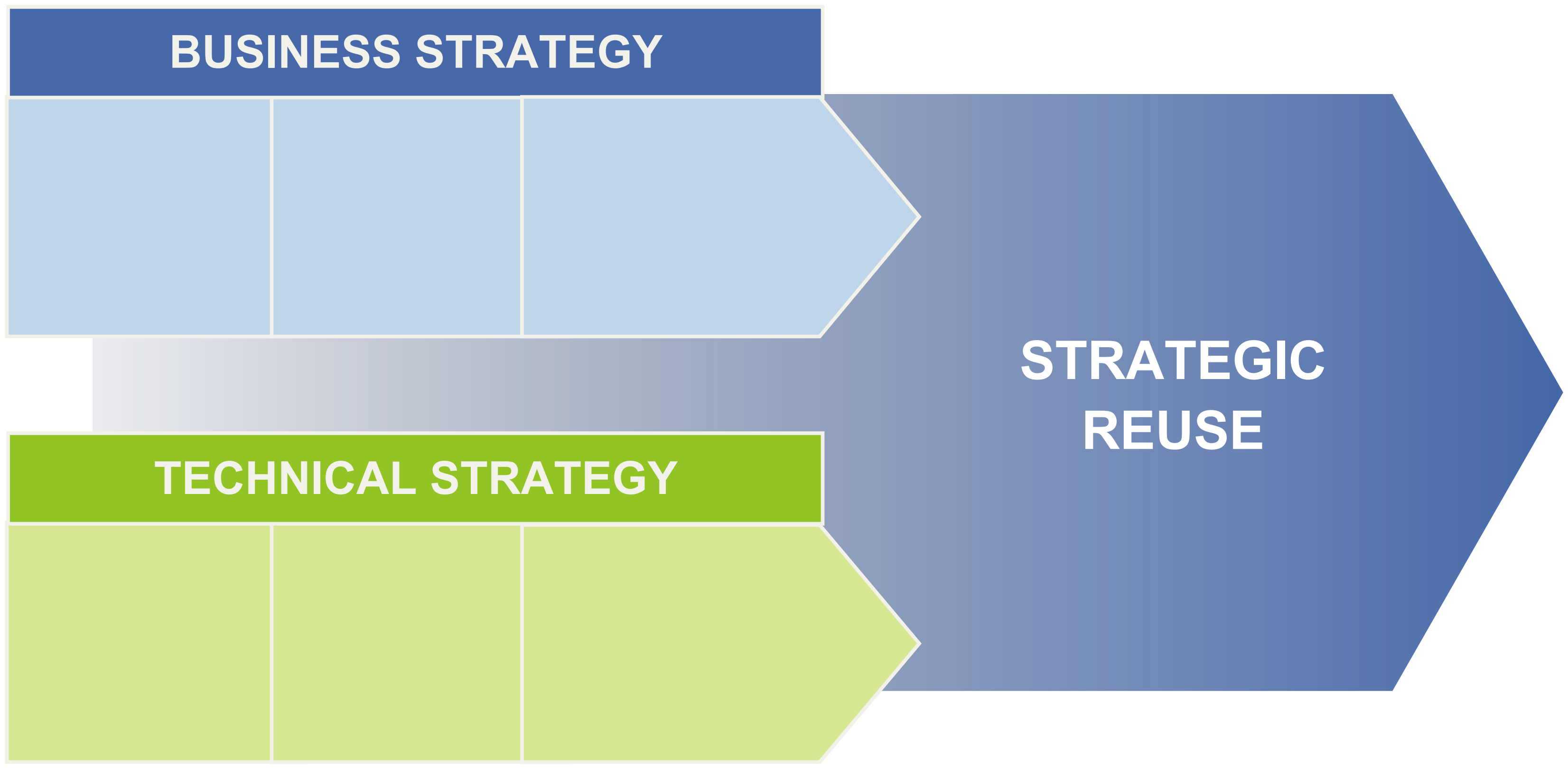
"... and then we'll be able to construct software systems by picking out parts and plugging them together, just like Tinkertoys ..."

THE FACT:

"It's more like having a bathtub full of Tinkertoys, Legos, Erector Set parts, Lincoln Logs, Block City, and six other incompatible kits -- picking out parts that fit specific functions and expecting them to fit together."



IMAGINE STRATEGIC REUSE



CELSIUSTECH: SHIP SYSTEM 2000

A FAMILY OF 55 SHIP SYSTEMS

- Integration test of 1-1.5 million
- SLOC requires 1-2 people.
- Rehosting to a new platform/OS takes 3 months.
- Cost and schedule targets are predictably met.
- Performance/distribution behavior are known in advance.
- Customer satisfaction is high.
- Hardware-to-software cost ratio changed from 35:65 to 80:20.

CUMMINS INC.: DIESEL CONTROL SYSTEMS

OVER 20 PRODUCT GROUPS WITH OVER 1,000 SEPARATE ENGINE APPLICATIONS

- Product cycle time was slashed from 250 person-months to a few person-months.
- Build and integration time was reduced from one year to one week.
- Quality goals are exceeded.
- Customer satisfaction is high.
- Product schedules are met.

NATIONAL RECONNAISSANCE OFFICE/ RAYTHEON: CONTROL CHANNEL TOOLKIT

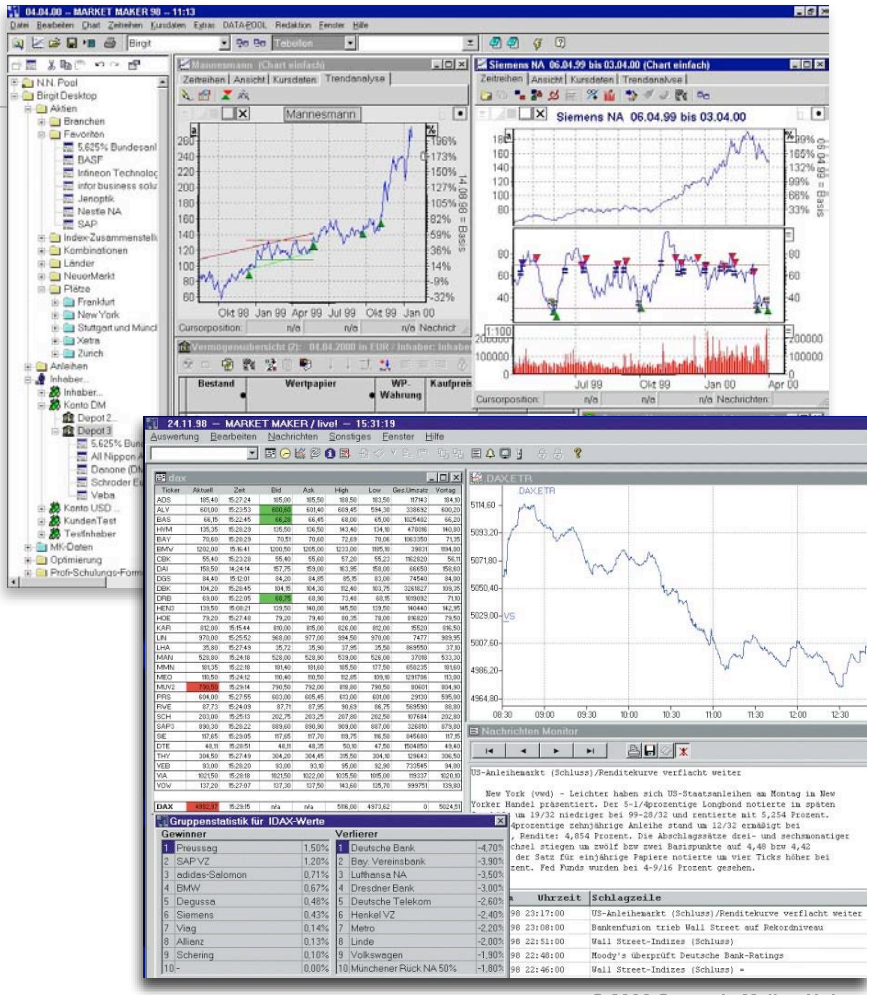
GROUND-BASED SPACECRAFT COMMAND AND CONTROL SYSTEMS

- increased quality by 10X
- incremental build time reduced from months to weeks
- software productivity increased by 7X
- development time and costs decreased by 50%
- decreased product risk

MARKET MAKER GMBH: MERGER

INTERNET-BASED STOCK MARKET SOFTWARE

- Each product is "uniquely" configured.
- Putting up a customized system takes three days.



04.04.00 – MARKET MAKER 98 – 11:13
Mannesmann (Chart einfach)
Siemens NA 06.04.99 bis 03.04.00 (Chart einfach)
24.11.98 – MARKET MAKER / live! – 15:31:19
DAXETR
Nachrichten Monitor
Gruppenstatistik für IDAX-Werte
Gewinner
Verlierer

NOKIA MOBILE PHONES

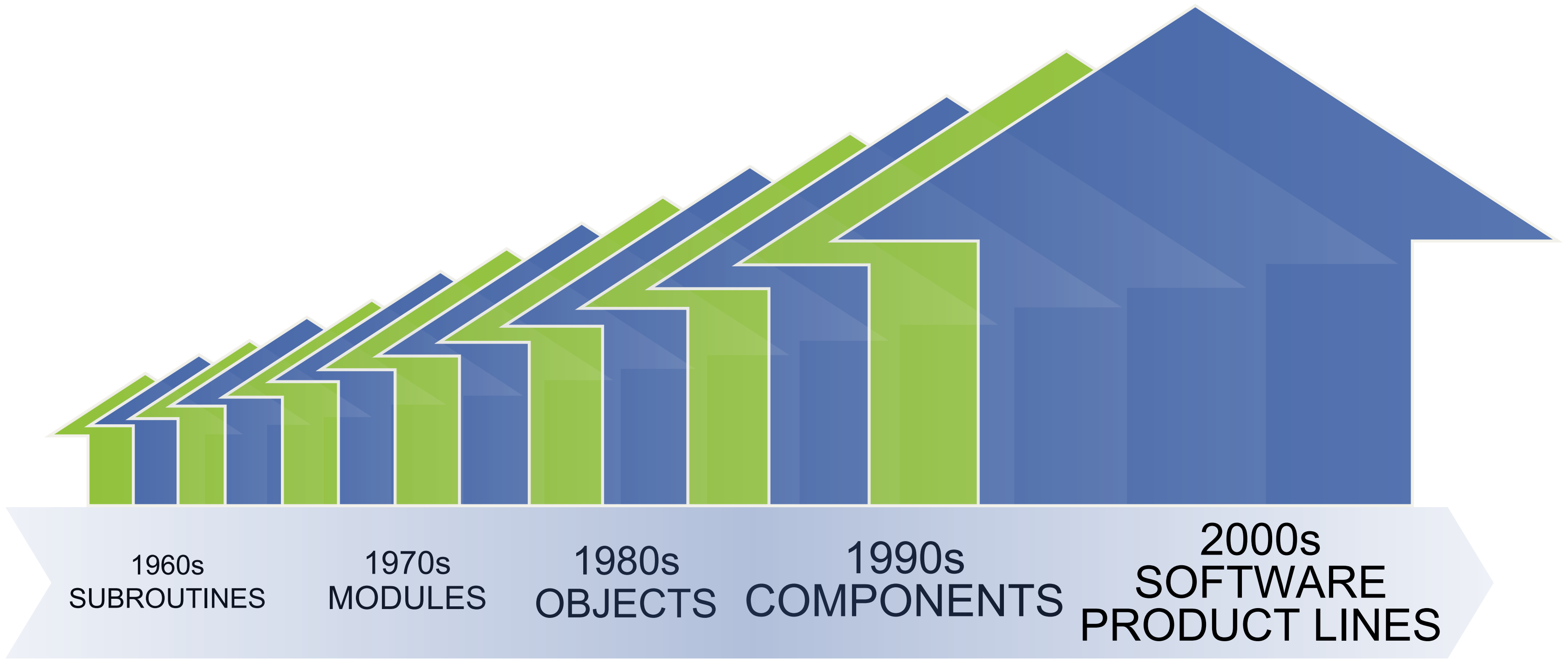
PRODUCT LINES WITH 25-30 NEW PRODUCTS PER YEAR

Across products there are

- varying number of keys
- varying display sizes
- varying sets of features
- 58 languages supported
- 130 countries served
- multiple protocols
- needs for backwards compatibility
- configurable features
- needs for product behavior
- change after release

HOW DID THEY DO IT?

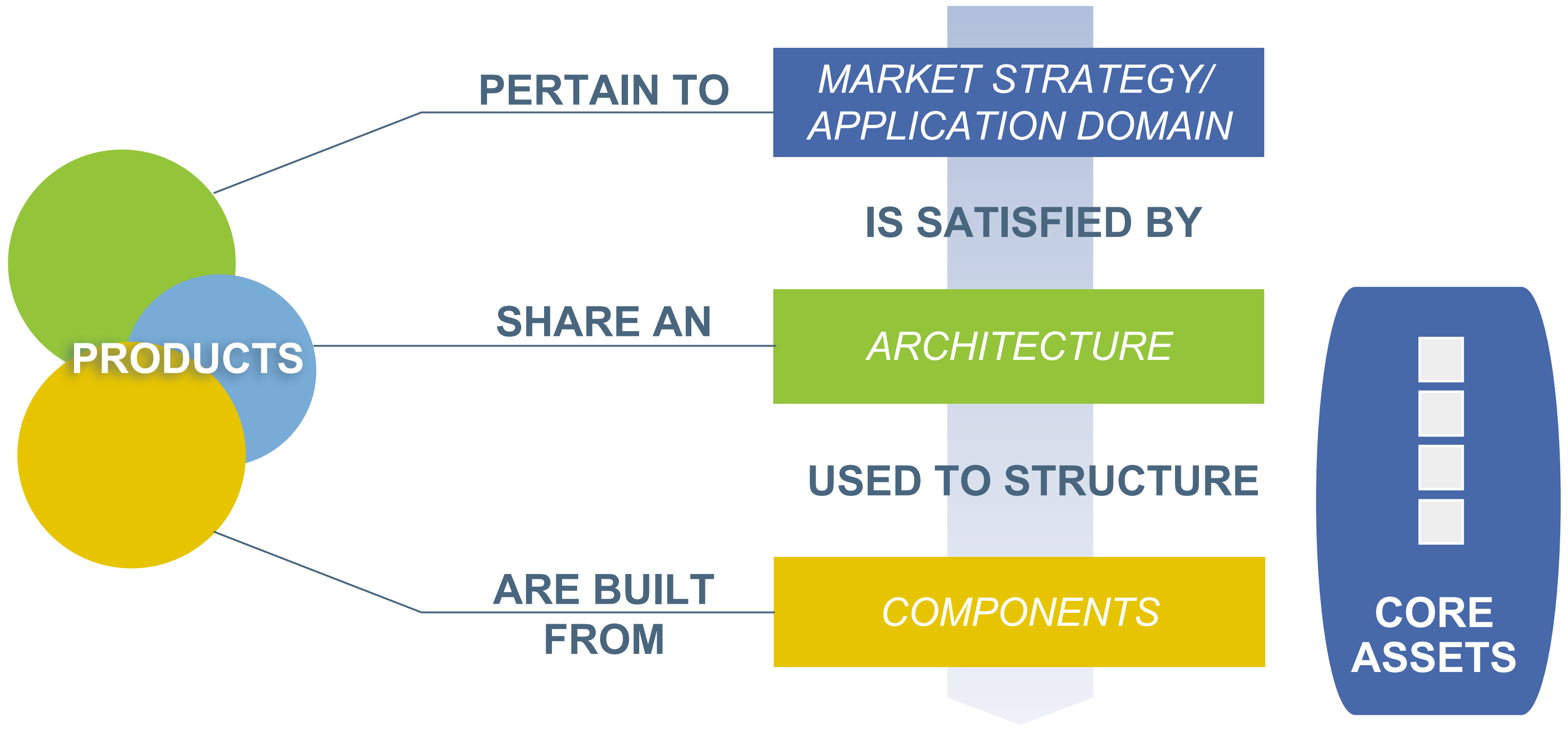
REUSE HISTORY: FROM AD HOC TO SYSTEMATIC



WHAT IS A SOFTWARE PRODUCT LINE?

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

SOFTWARE PRODUCT LINES



Product lines
- take economic advantage of commonality
- bound variability

HOW DO PRODUCT LINES HELP?

PRODUCT LINES AMORTIZE THE INVESTMENT IN THESE AND OTHER *CORE ASSETS:*

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

PRODUCT LINES = STRATEGIC REUSE

EARLIER LIFE CYCLE REUSE

⇓⇓

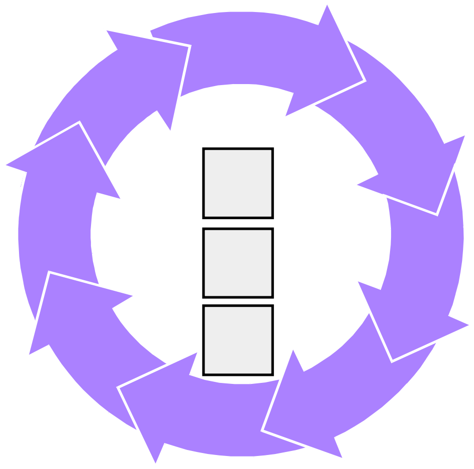
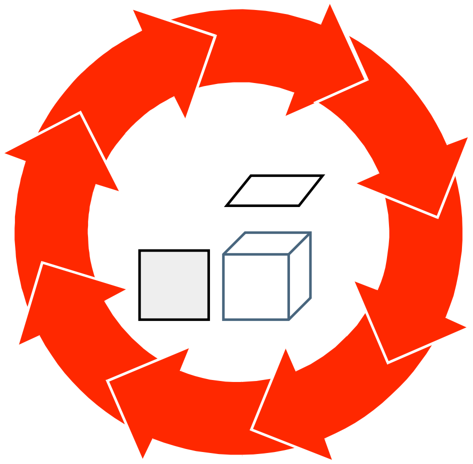
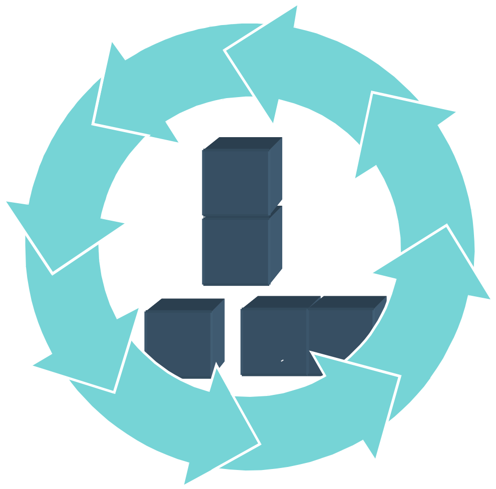
MORE BENEFIT

THE KEY CONCEPTS

Use of a core asset base

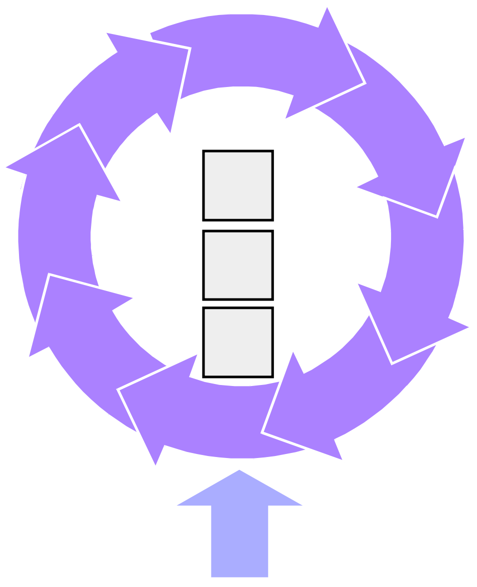
in production

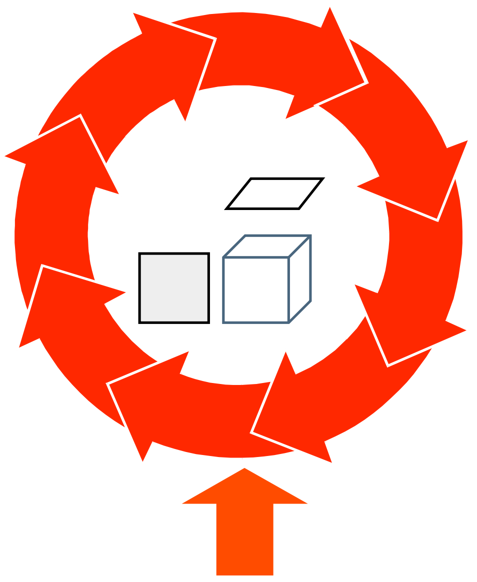
of a related set of products

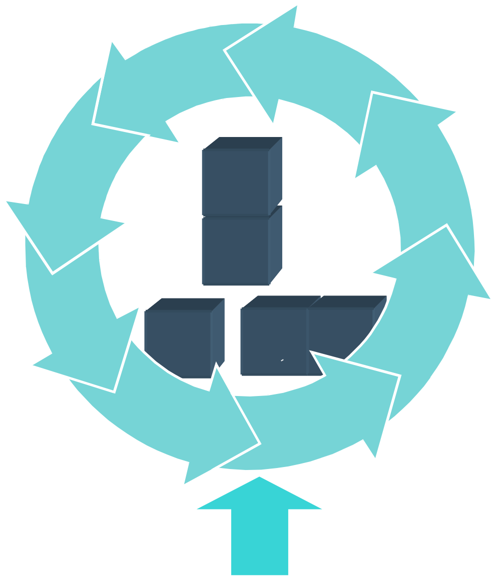




THE KEY CONCEPTS

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







SOFTWARE PRODUCT LINES ARE NOT

FORTUITOUS SMALL-GRAINED REUSE

- reuse libraries containing algorithms, modules, objects, or components

SINGLE-SYSTEM DEVELOPMENT WITH REUSE

- borrowing opportunistically from previous efforts
- modifying code as necessary for the single system only

JUST COMPONENT-BASED DEVELOPMENT

- selecting components from an in-house library or the marketplace with no architecture focus

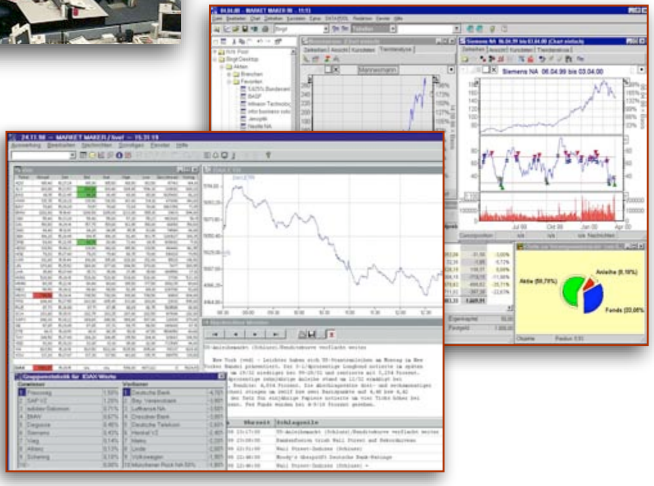
JUST A CONFIGURABLE ARCHITECTURE

- a good start, but only part of the reuse potential

JUST A SET OF TECHNICAL STANDARDS

- constraining choices without an architecture-based reuse strategy

PRODUCT LINES ARE



Software product lines involve strategic, planned reuse that yields predictable results.

COMMERCIAL EXAMPLES

SUCCESSFUL SOFTWARE PRODUCT LINES HAVE BEEN BUILT FOR FAMILIES OF

- mobile phones
- command and control ship systems
- ground-based spacecraft systems
- avionics systems
- command and control/situation awareness systems
- pagers
- engine control systems
- billing systems
- web-based retail systems
- printers
- consumer electronic products
- acquisition management enterprise systems
- financial and tax systems
- medical devices

REAL WORLD MOTIVATION

ORGANIZATIONS USE PRODUCT LINE PRACTICES TO:

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

SUMMARY: ORGANIZATIONAL BENEFITS

IMPROVED PRODUCTIVITY

- by as much as 10x

DECREASED TIME TO MARKET (TO FIELD, TO LAUNCH...)

- by as much as 10x

DECREASED COST

- by as much as 60%

DECREASED LABOR NEEDS

- by as much as 10X fewer software developers

INCREASED QUALITY

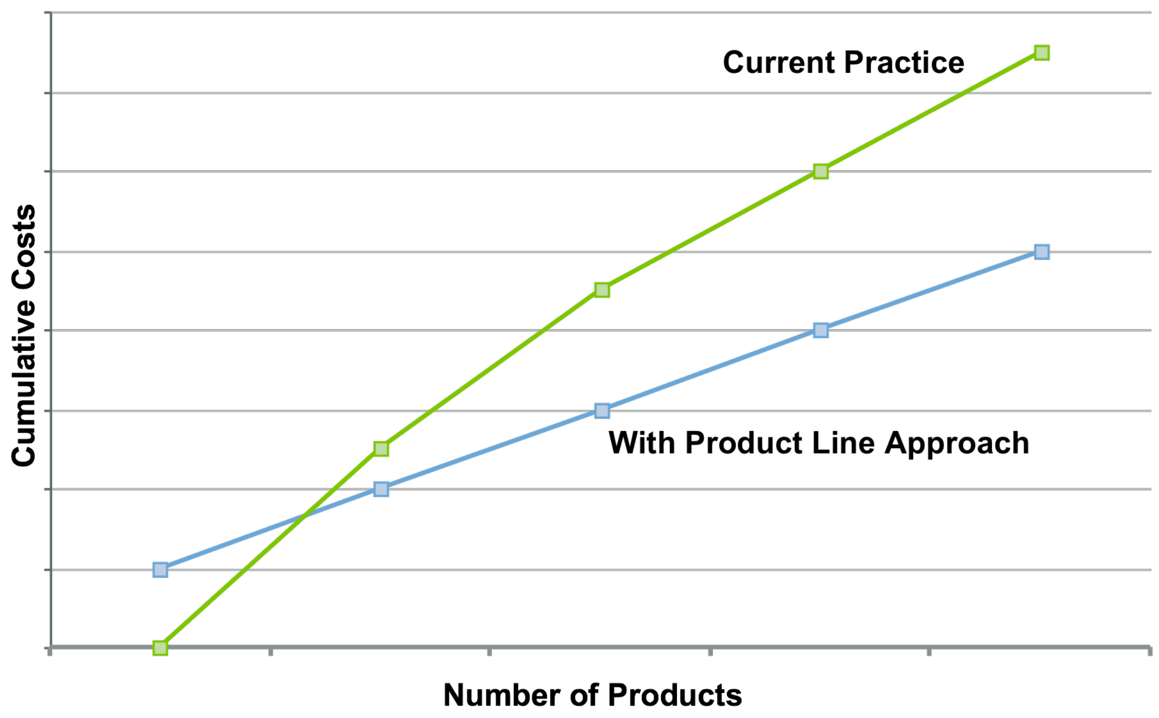
- by as much as 10X fewer defects

*Product line practice permits **predictable** "faster, better, cheaper."*

COSTS OF A SOFTWARE PRODUCT LINE

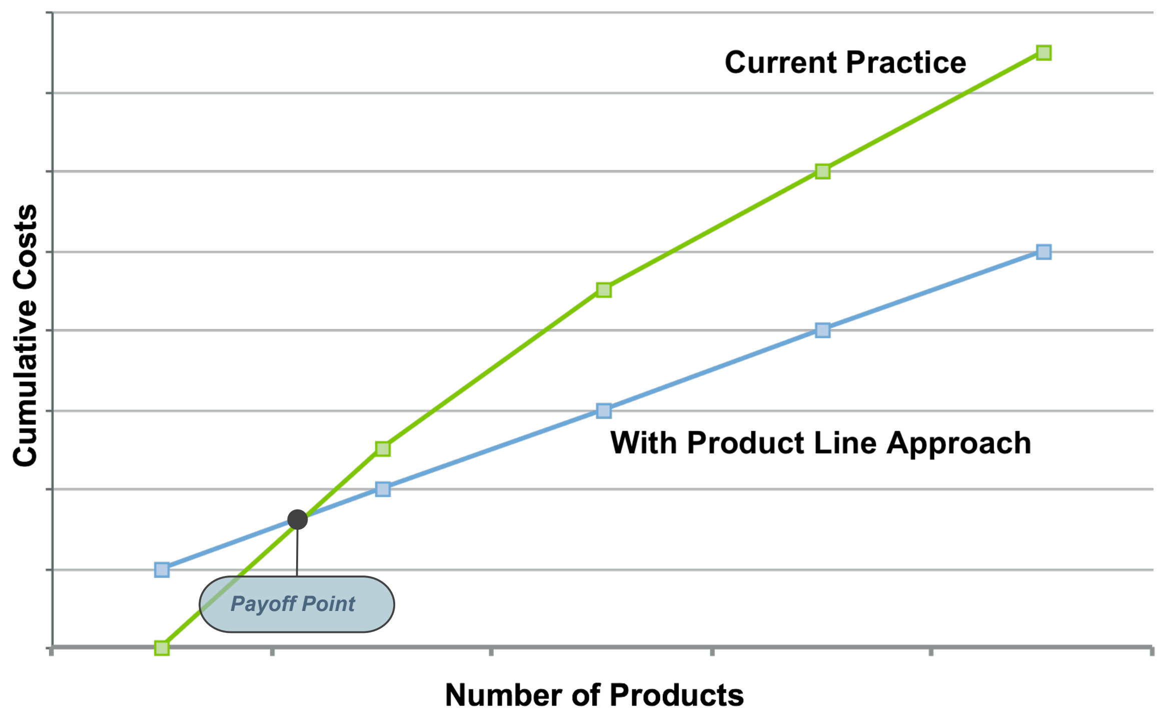
Core Assets	Costs
Architecture	**Must support variation inherent in the product line**
Software Components	**Must be designed to be general without a loss of performance; must build in support for variation points**
Test Plans, Test Cases, Test Data	**Must consider variation points and multiple instances 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**

ECONOMICS OF PRODUCT LINES



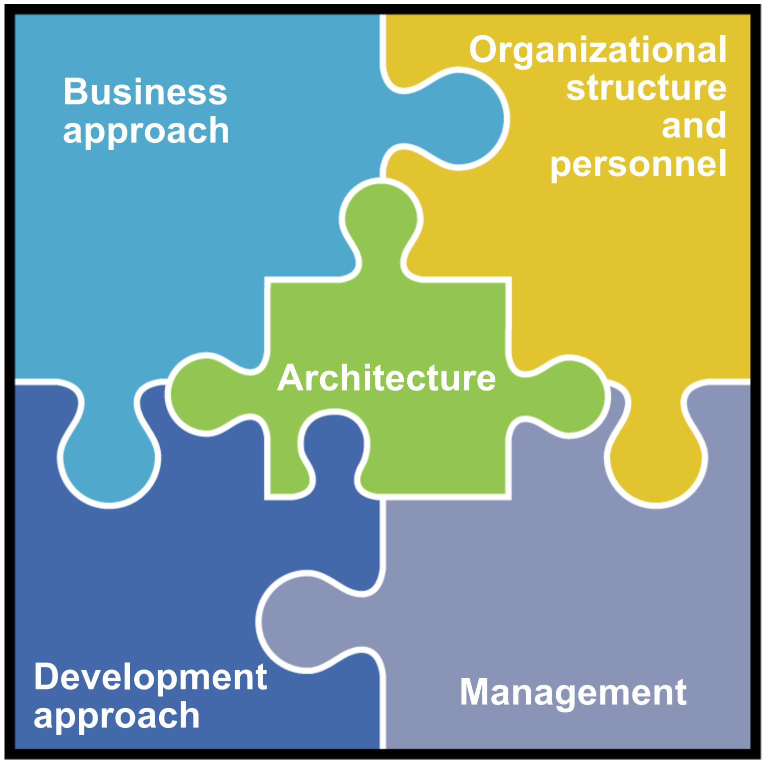
Weiss. D.M. & and Lai, C.T.R..
Software Product-Line Engineering: A Family-Based Software Development Process
Reading, MA: Addison-Wesley, 1999.

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.

NECESSARY CHANGES



The product line architecture is the foundation of everything.

WHY IS SOFTWARE ARCHITECTURE IMPORTANT?

Represents ***earliest*** design decisions

- hardest to change
- most critical to get right
- communication vehicle among stakeholders

First design artifact addressing

- performance
- modifiability
- reliability
- security

Key to systematic ***reuse***

- transferable, reusable abstraction

The **right architecture** paves the way for system **success.**
The **wrong architecture** usually spells some form of **disaster.**

PRODUCT LINE PRACTICE

CONTEXTS FOR PRODUCT LINES VARY WIDELY, BASED ON

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

But there are universal essential activities and practices.

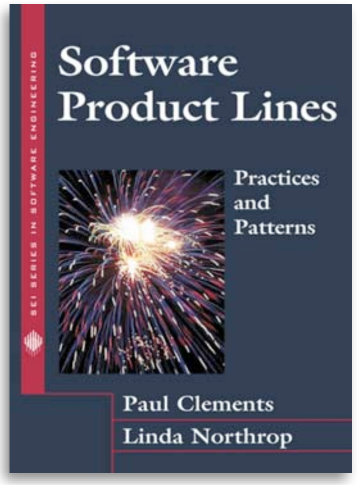
THE SEI FRAMEWORK FOR SOFTWARE PRODUCT LINE PRACTICE^{SM}

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 4.0 – in *Software Product Lines: Practices and Patterns*

Version 4.2 – http://www.sei.cmu.edu/productlines/framework.html



SEI INFORMATION SOURCES

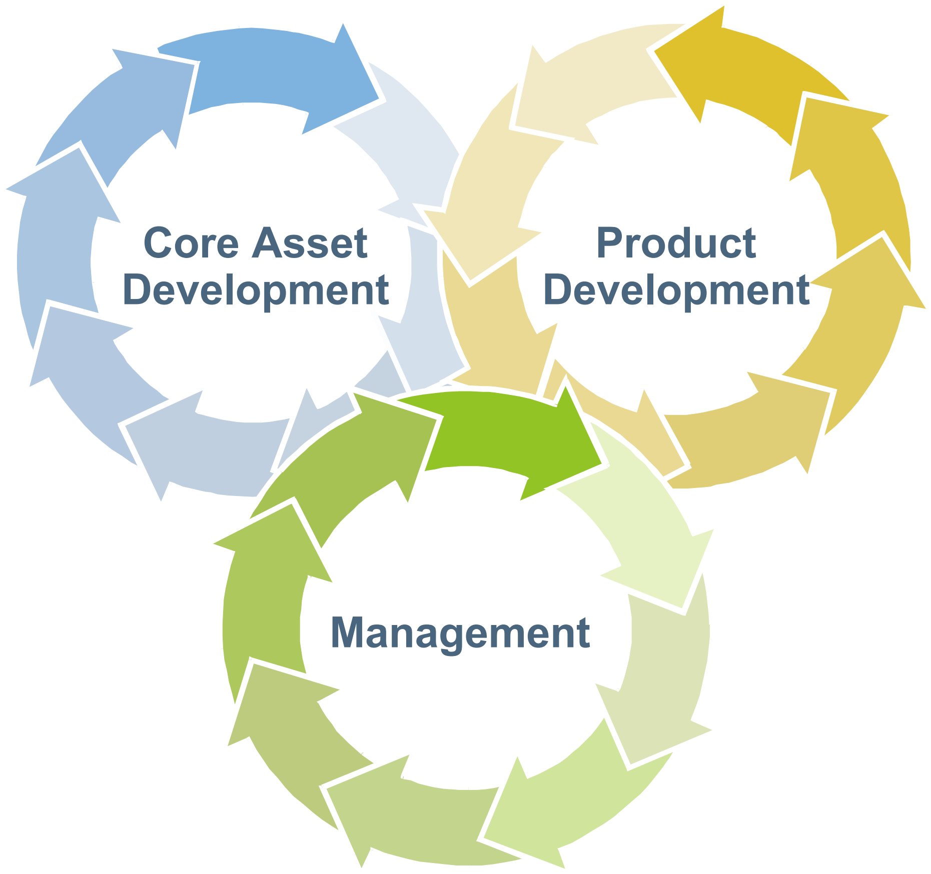
Case studies, experience reports, and surveys

Workshops and conferences

Applied research

Collaborations with customers on actual product lines

THE THREE ESSENTIAL ACTIVITIES



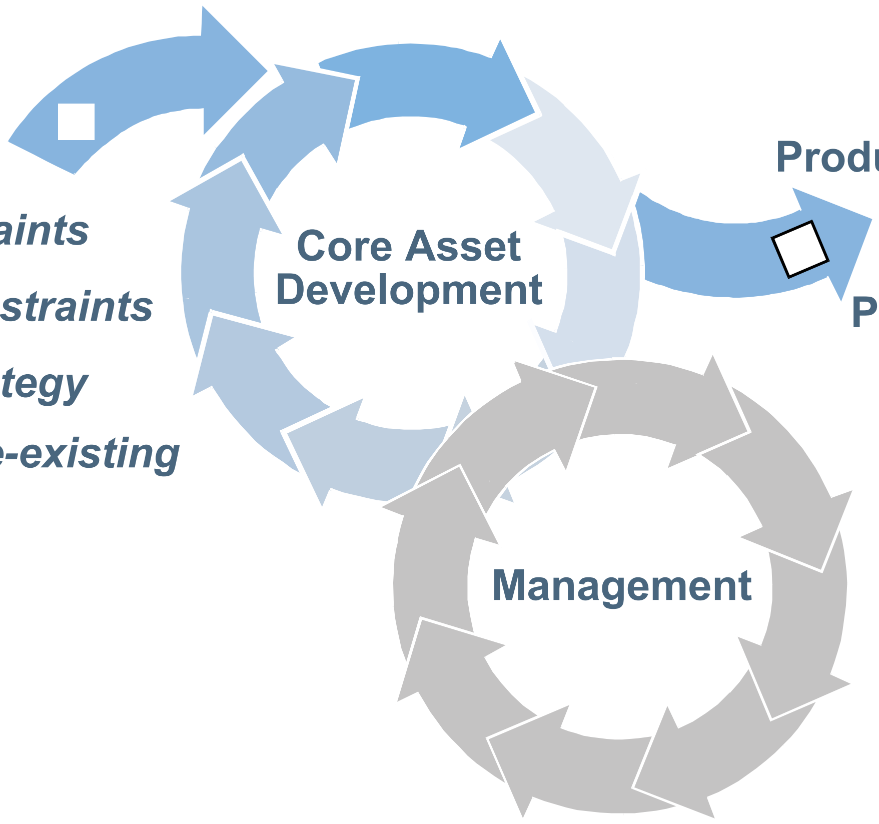
CORE ASSET DEVELOPMENT

Product Constraints

Production Constraints

Production Strategy

Inventory of Pre-existing Assets

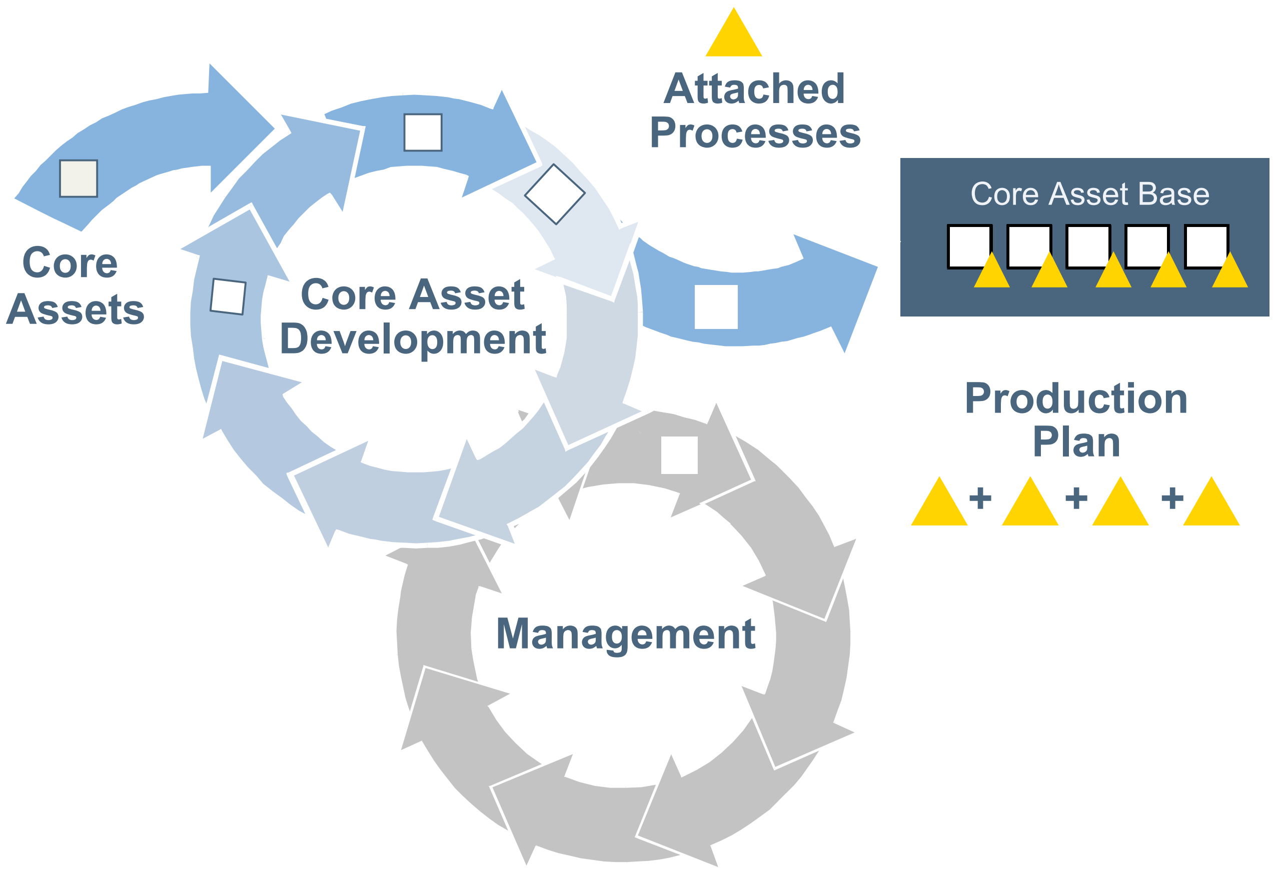


Product Line Scope

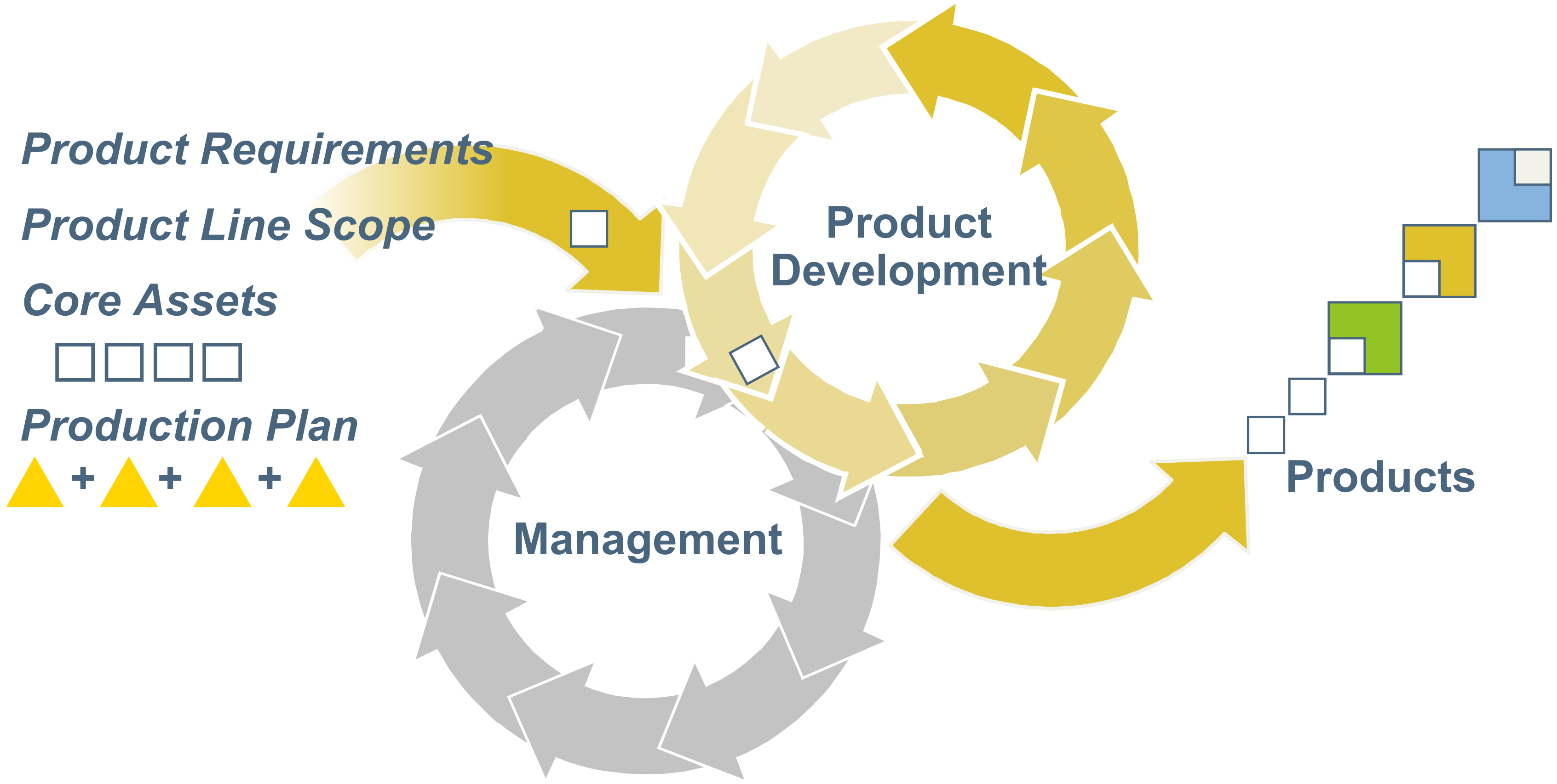
Core Assets

Production Plan

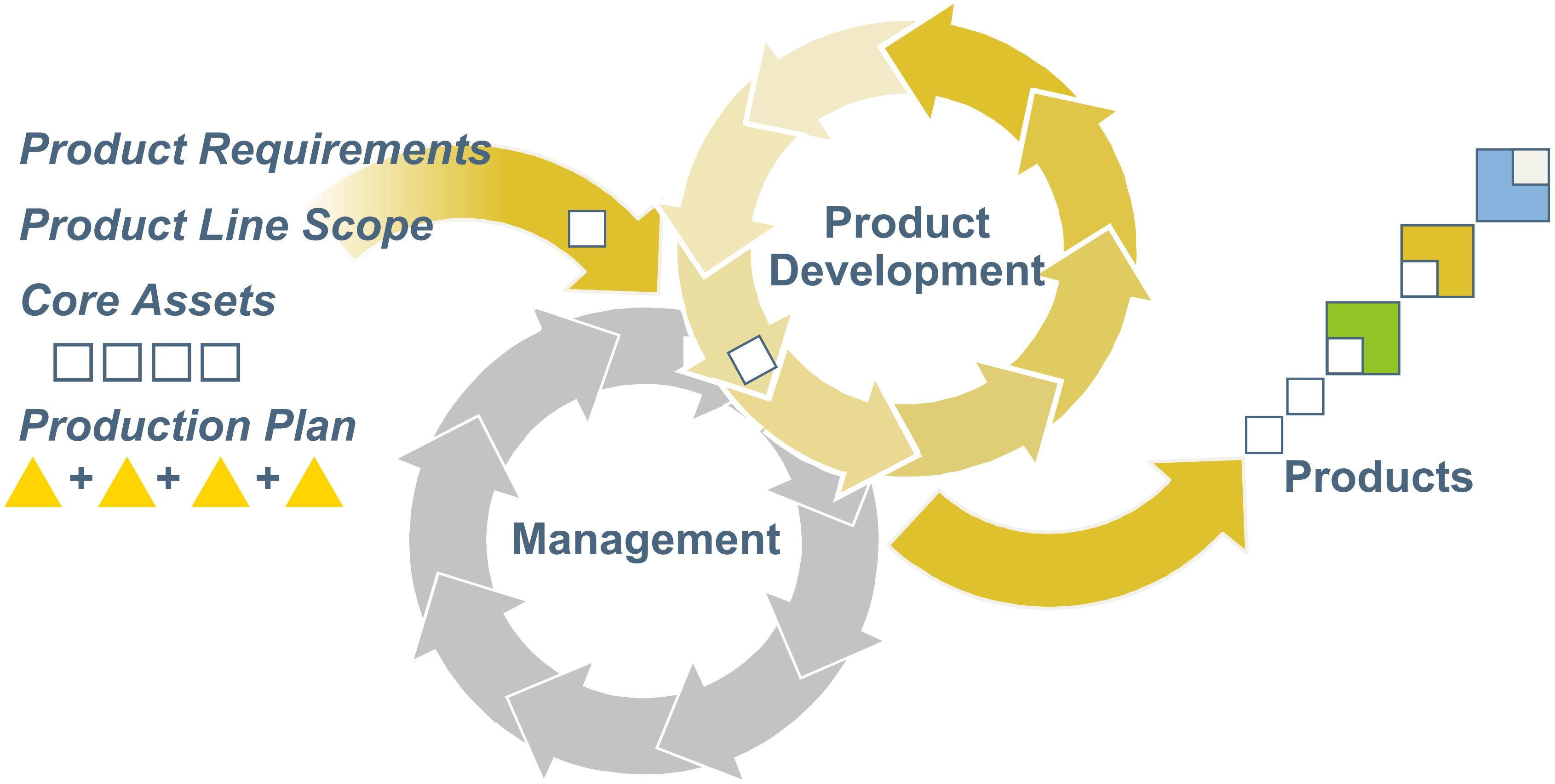
ATTACHED PROCESSES



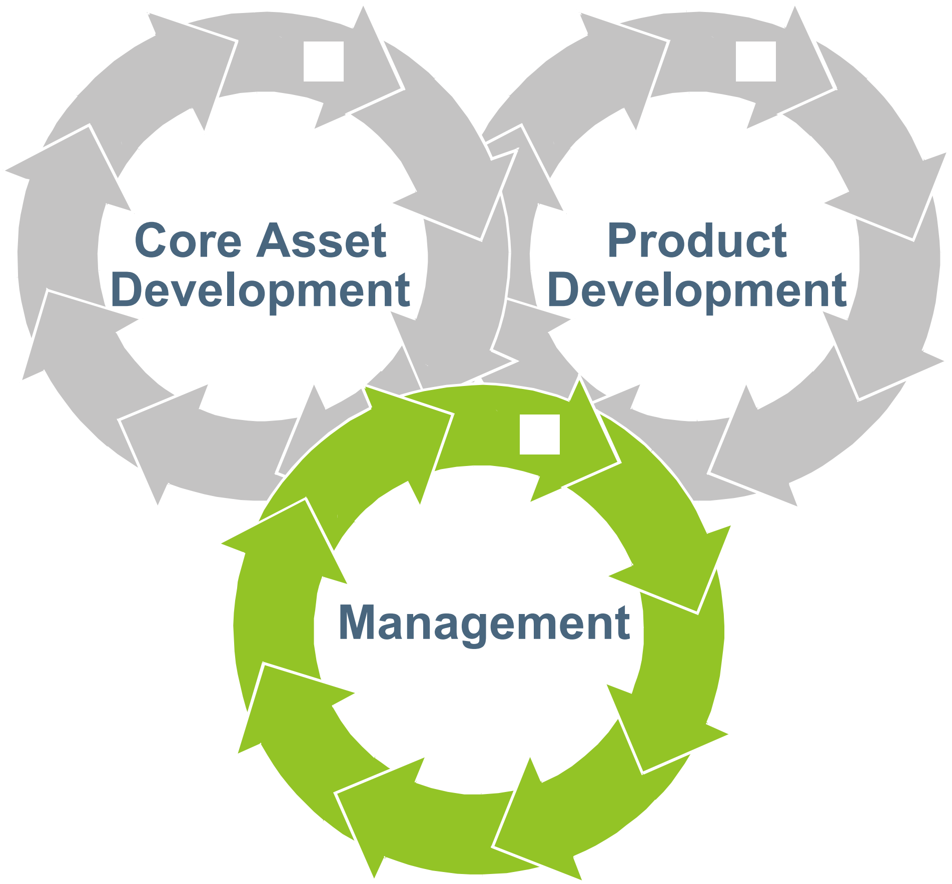
PRODUCT DEVELOPMENT



PRODUCT DEVELOPMENT



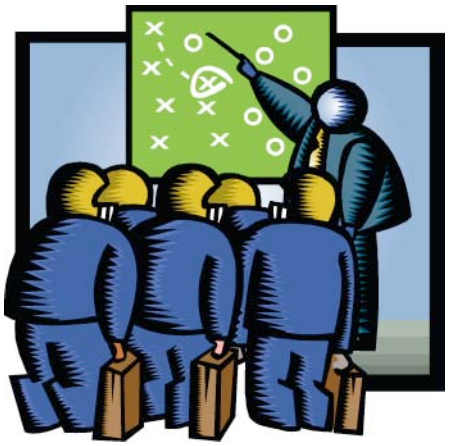
MANAGEMENT



MANAGEMENT

MANAGEMENT AT MULTIPLE LEVELS PLAYS A CRITICAL ROLE IN THE SUCCESSFUL PRODUCT LINE PRACTICE BY

- achieving the right organizational structure
- allocating resource
- coordinating and supervising
- providing training
- rewarding employees appropriately
- developing and communicating an acquisition strategy
- managing external interfaces
- creating and implementing a product line adoption plan
- launching and institutionalizing the approach in a manner appropriate to the organization



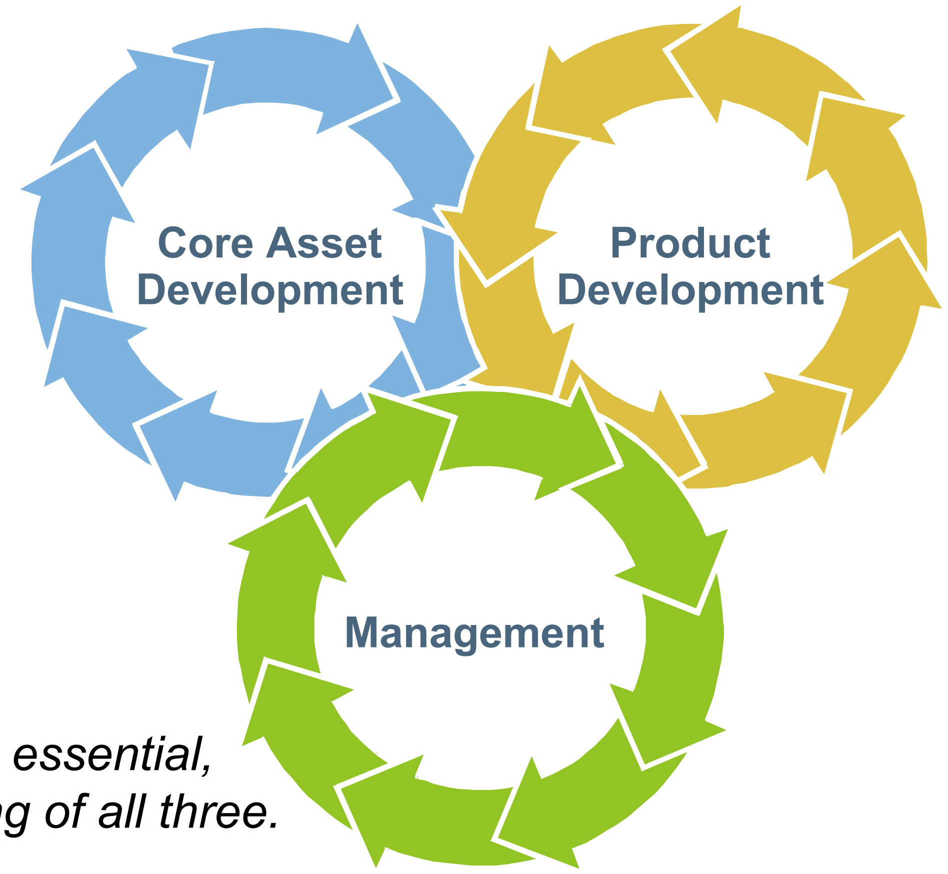
MANAGING A SOFTWARE PRODUCT LINE REQUIRES LEADERSHIP

A KEY ROLE FOR A SOFTWARE PRODUCT LINE MANAGER IS THAT OF CHAMPION.

The champion must

- set and maintain the vision
- ensure that the appropriate goals and measures are in place
- “sell” the product line up and down the chain
- sustain morale
- deflect potential derailments
- solicit feedback and continuously improve the approach

ESSENTIAL PRODUCT LINE ACTIVITIES



Each of these is essential, as is the blending of all three.

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.
- …

ALTERNATE TERMINOLOGY

OUR TERMINOLOGY		ALTERNATE TERMINOLOGY
Product Line		Product Family
Core Assets		Platform
Business Unit		Product Line
Product		Customization
Core Asset Development		Domain Engineering
Product Development		Application Engineering

DRIVING THE ESSENTIAL ACTIVITIES

BENEATH THE LEVEL OF THE ESSENTIAL ACTIVITIES ARE ESSENTIAL PRACTICES THAT FALL INTO PRACTICE AREAS.

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

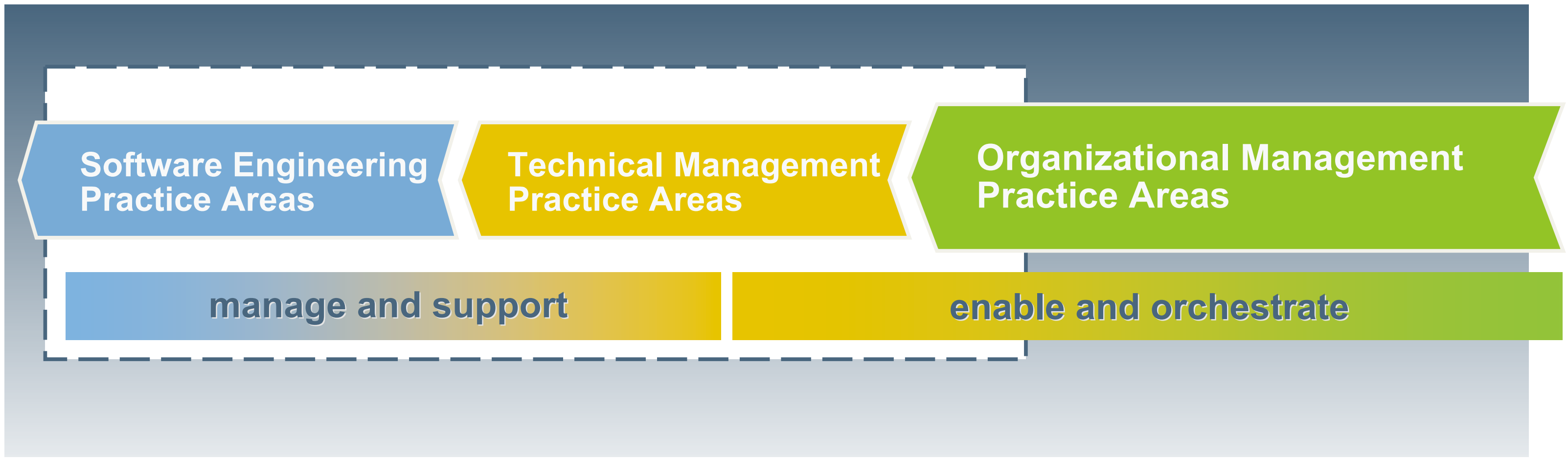
PRACTICE AREAS CATEGORIES

Software Engineering

Technical Management

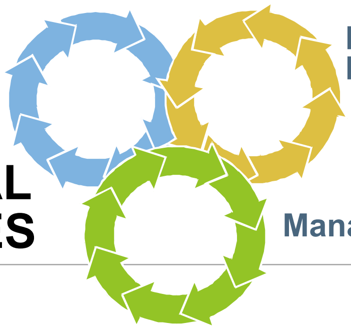
Organizational Management

RELATIONSHIPS AMONG CATEGORIES OF PRACTICE AREAS


Software Engineering Practice Areas
Technical Management Practice Areas
Organizational Management Practice Areas
manage and support
enable and orchestrate

FRAMEWORK

ESSENTIAL ACTIVITIES



PRACTICE AREAS

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

DILEMMA: HOW DO YOU APPLY THE 29 PRACTICE AREAS?

ORGANIZATIONS STILL HAVE TO FIGURE OUT HOW TO PUT THE PRACTICE AREAS INTO PLAY.

Twenty-nine is a big number.

HELP TO MAKE IT HAPPEN

ESSENTIAL ACTIVITIES

PRACTICE AREAS

Software Engineering	Technical Management	Organizational Management

GUIDANCE

Case Studies	Patterns	Probe

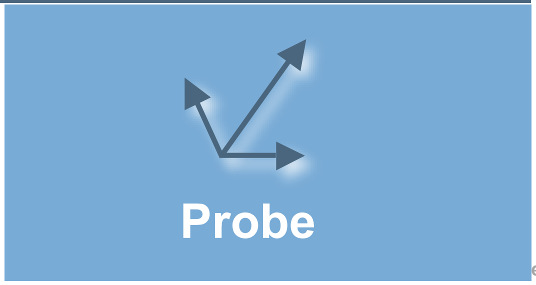
HELP TO MAKE IT HAPPEN

ESSENTIAL
ACTIVITIES

PRACTICE AREAS

Software Engineering	Technical Management	Organizational Management

GUIDANCE



e Mellon Univer

HELP TO MAKE IT HAPPEN

ESSENTIAL ACTIVITIES

PRACTICE AREAS

Software Engineering	Technical Management	Organizational Management

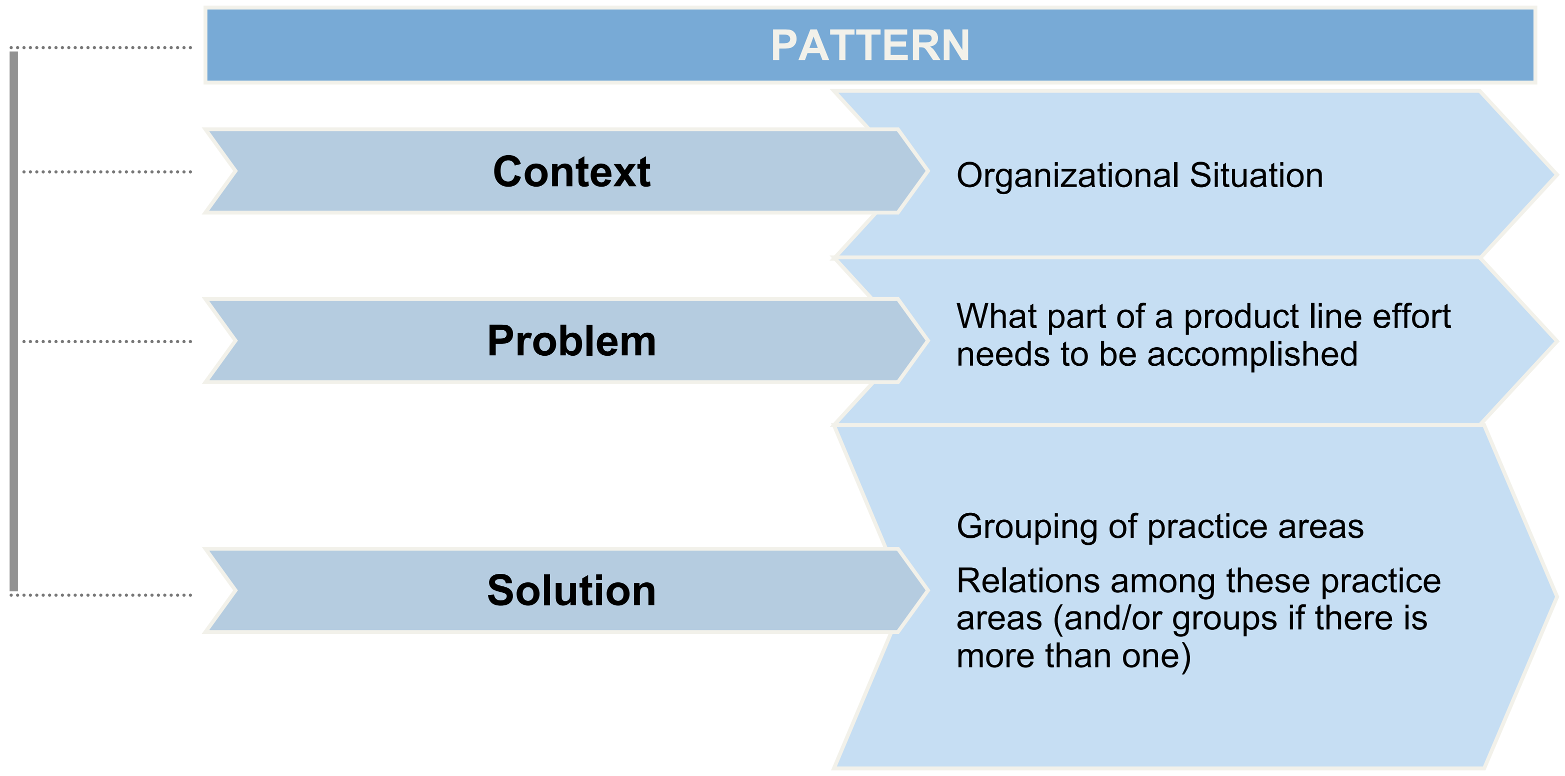
GUIDANCE

Case Studies	Patterns	Probe

PATTERNS CAN HELP

- 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 others.
- Patterns assist in effecting a divide and conquer approach.

SOFTWARE PRODUCT LINE PRACTICE PATTERN


PATTERN
Context
Organizational Situation
Problem
What part of a product line effort needs to be accomplished
Solution
Grouping of practice areas
Relations among these practice areas (and/or groups if there is more than one)

WHAT TO BUILD PATTERN - 1

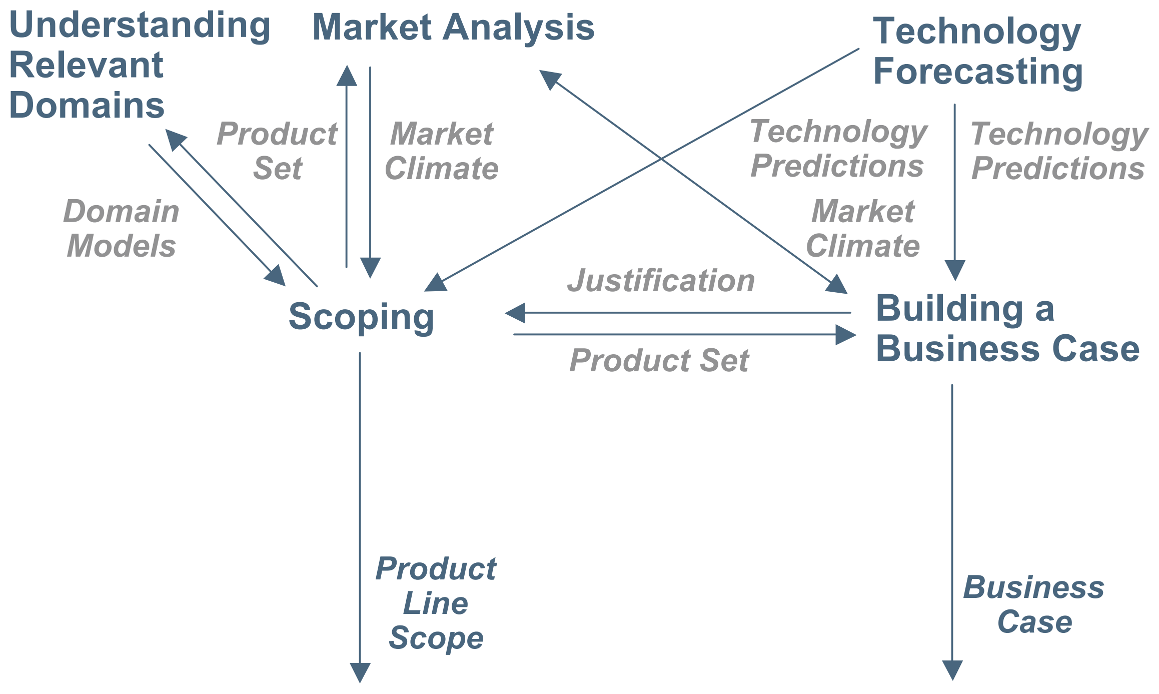
NAME:

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



Dynamic Structure

FACTORY PATTERN - 1

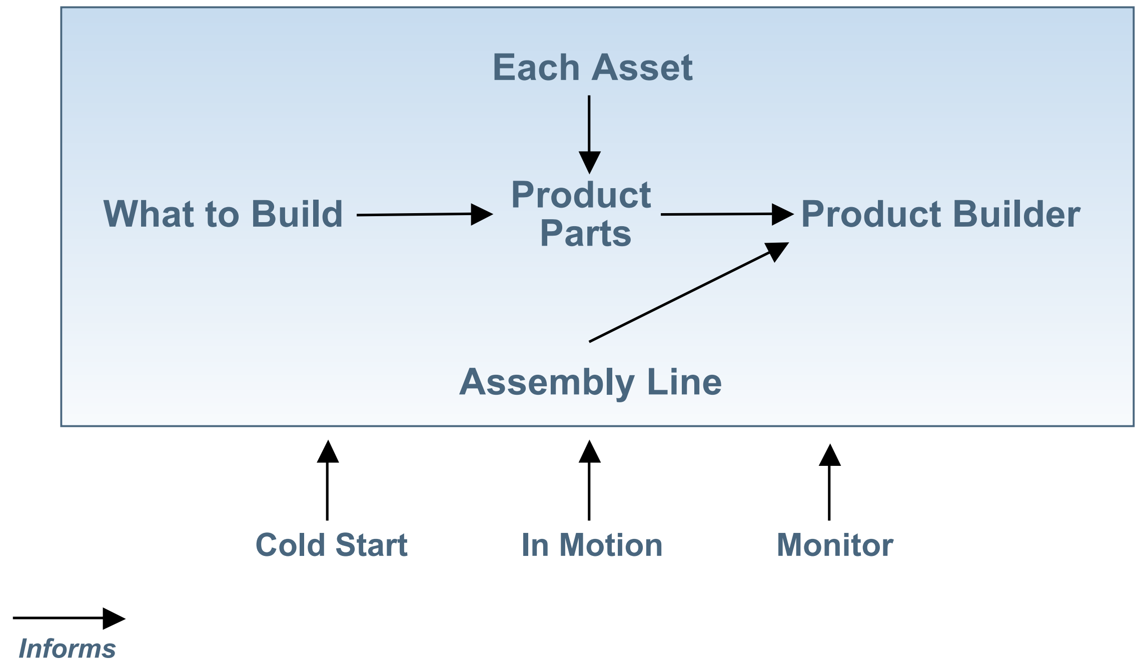
NAME:

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

CONTEXT:

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

FACTORY PATTERN - 2



Dynamic Structure

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 Parts	Green Field Barren Field Plowed Field
What to Build	Analysis Forced March

THE ADOPTION ENDGAME

EFFECTIVELY ACHIEVE AN *OPERATIONAL PRODUCT LINE*.

- have
  - a core asset base
  - supportive processes and organizational structures
- develop products from that asset base in a way that achieves business goals
- improve and extend the software product line adoption effort as long as it makes sense

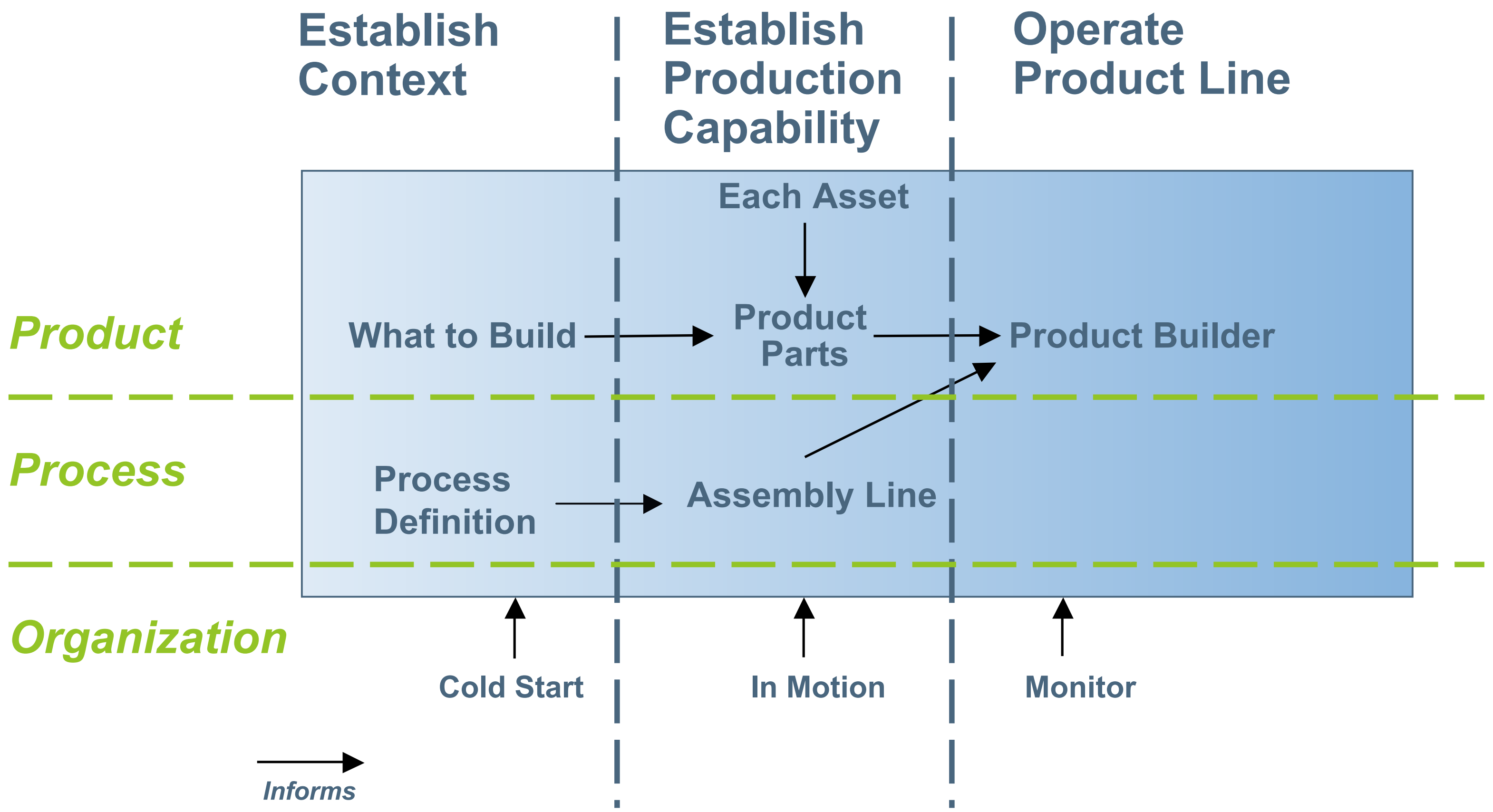
BARRIERS TO PRODUCT LINE ADOPTION

Cost, cost,
&
cost….

You have to
invest to
eventually
SAVE.

BARRIERS TO PRODUCT LINE ADOPTION

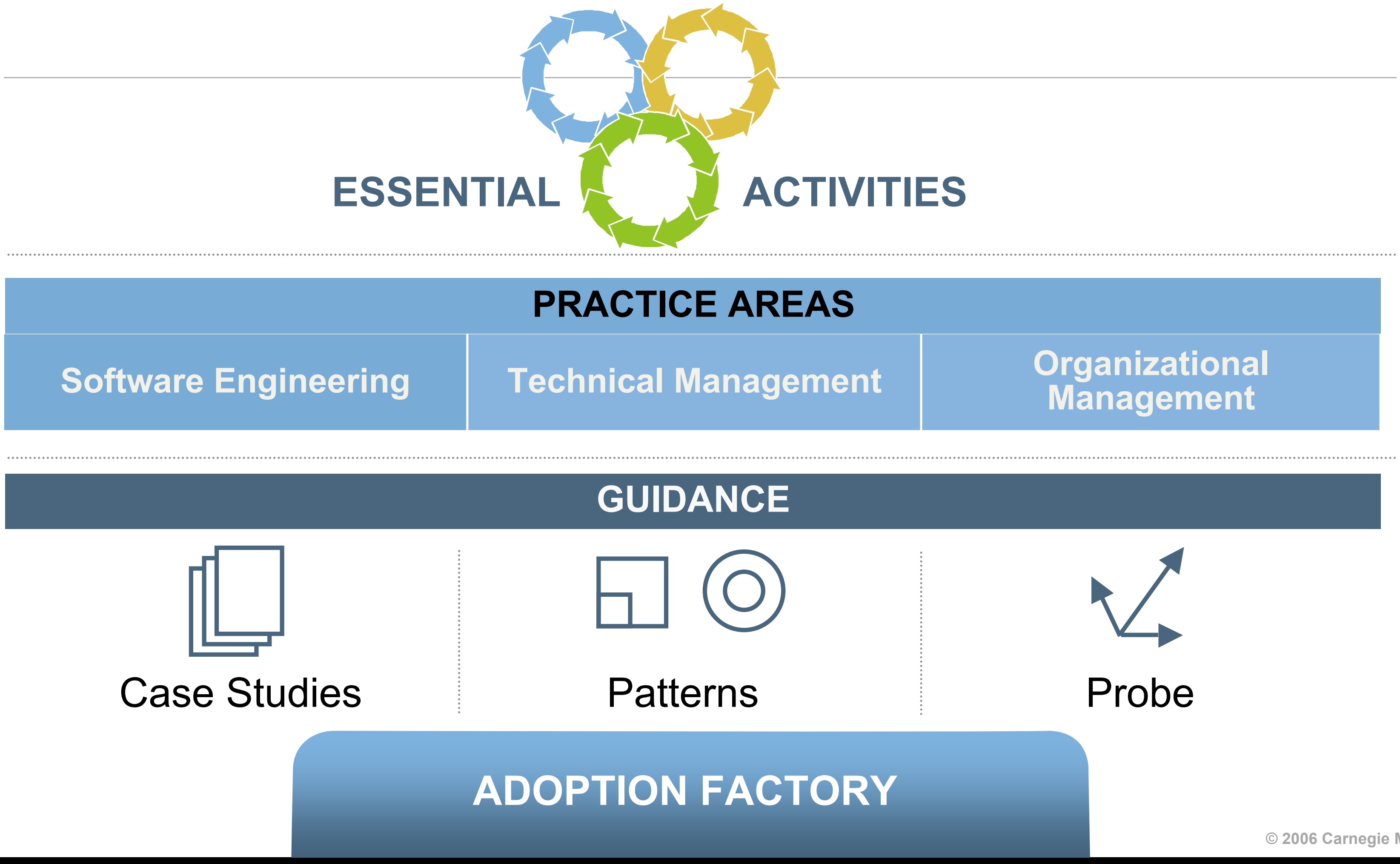
THE ADOPTION FACTORY PATTERN



ASSOCIATED PRACTICE AREAS

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

THE ENTIRE PICTURE



IN A NUTSHELL

- Software product lines epitomize the concept of strategic, planned reuse.
- The product line concept is about more than a new technology. It is a new way of doing one's software business.
- There are essential product line activities and practices areas as well as product line patterns to make the move to product lines more manageable.

WHAT'S DIFFERENT ABOUT REUSE WITH SOFTWARE PRODUCT LINES?

- Business dimension
- Iteration
- Architecture focus
- Preplanning
- Process **and** product connection

AT THE HEART OF SUCCESSFUL PRODUCT LINES

- A pressing need that addresses the heart of the business
- Long and deep domain experience
- A legacy base from which to build
- Architectural excellence
- Process discipline
- Management commitment
- Loyalty to the product line as a single entity

FINAL WORD

If properly managed, the benefits of a product line approach far exceed the costs.

Strategic software reuse through a well-managed product line approach achieves business goals for:

- efficiency
- time to market
- productivity
- quality
- agility

SOFTWARE PRODUCT LINES:
REUSE THAT MAKES BUSINESS SENSE.

QUESTIONS – NOW OR LATER

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

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

World Wide Web:

http://www.sei.cmu.edu/productlines/plp_init.html

SEI Fax: 412-268-5758