

Carnegie Mellon University  
Software Engineering Institute

# Software Product Lines: Past, Present, and Future

Dr. Paul C. Clements

Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213-3890

Sponsored by the U.S. Department of Defense  
© 2005 by Carnegie Mellon University

[Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



# Software Engineering Institute

**Who We Are:** Applied R&D laboratory situated as a college-level unit at Carnegie Mellon University, Pittsburgh, PA, USA

**Purpose:** Help others make measured improvements in their software engineering practices

**First objective:** Accelerate the introduction and widespread use of high-payoff software engineering practices and technology identifying, evaluating, and maturing promising or underused technology and practices.

We are a small organization. We have to pick these practices carefully.

One we have chosen is **software product lines.**

© 2006 by Carnegie Mellon University



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## 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.**

- Product lines in manufacturing have been around for centuries.
- Software product lines have not.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Why software product lines?

Because of the high-payoff results they make possible in practice:

Improved productivity  
by as much as 10x

Decreased time to market  
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



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## Real-world examples

Successful software product lines have achieved these results

- across multiple domains
  - Mobile phones
  - Command and control ship systems
  - Ground-based spacecraft systems
  - Avionics systems
  - Pagers
  - Engine control systems
  - Billing systems
  - Web-based retail systems
  - Printers
  - Consumer electronic products
  - Acquisition management enterprise systems
- in large organizations and small
- in Government and private sectors

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)





© 2006 by Carnegie Mellon University

page 11

[Previous](#) [Next](#)

[Exit Slide Show](#)

- [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Product Lines in Building/Manufacturing

Product lines – building a family of products from interchangeable parts – have existed for centuries.

A thousand years ago, Li Chieh, the state architect of the Chinese emperor Hui-tsung, published a set of building codes for official buildings.

- Standard parts and ways to connect the parts
- Parameterized variations: lengths, loads
- Options for components
- Finishing with brackets, decorations

This book defined a set of reusable designs: a “product line” of buildings.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## A large, modern example



© 2006 by Carnegie Mellon University

page 16

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)





Carnegie Mellon  
Software Engineering Institute

**“A common type-rating enables pilots qualified to fly any 757 or 767 to fly all the others with minimal additional familiarization, saving training time and costs. The 757/767 allows crews to fly more models with a greater variety of fuselage widths, capacities, and range capabilities than any other common-rated family”** -- [www.boeing.com](http://www.boeing.com)



© 2006 by Carnegie Mellon University

page 17

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Another Example: The Airbus A-320 Family

- 100-200 seats
- 103-146 feet in length
- Same wing
- Same flight deck



© 2006 by Carnegie Mellon University

page 18

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Yet another example of a product line



Hamburger



Cheeseburger



Dbl. hamburger



Double cheeseburger



Veggie burger

What are the core assets here?

What do these products have in common?

Source: [www.burgerking.com](http://www.burgerking.com)

© 2006 by Carnegie Mellon University

page 21

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Questions

What does a company gain from producing a product line of sandwiches? Of commercial passenger aircraft?

- Simplified production of components
- Simplified management of inventory
- Simplified training
- Streamlined production facilities and process
- Market recognition and mind share
- Flexibility – ability to add new products quickly



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## Examples are everywhere!

Manufacturing technology – particularly the principle of interchangeable parts – makes the product line concept appear almost everywhere.

Look around this room. How many product lines do you see?

In 1964, IBM introduced the System/360 family of computers. Each model differed widely in performance and features, but they all ran the same programs. The operating system was also a family.



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## What about Software?

Software engineers knew about interchangeable parts.

We knew it was better to avoid designing and building software from scratch.

In other words, it seemed that *reuse* was the way to achieve greater productivity.

We began with small-scale reuse.

- Subroutines were the first example.
- We recognized that different systems shared common parts.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Moving to Large-Scale Reuse



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

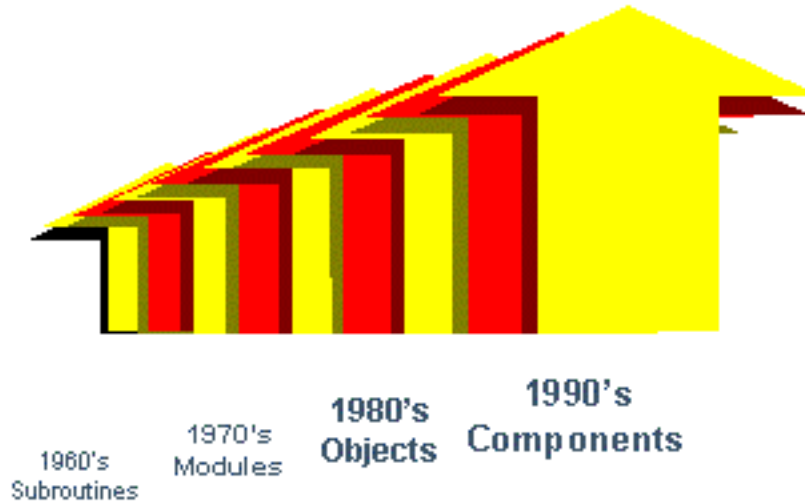
*Reuse* should pay off.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## But reuse has a disappointing history



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

[Previous](#) [Next](#)

[Exit Slide Show](#)

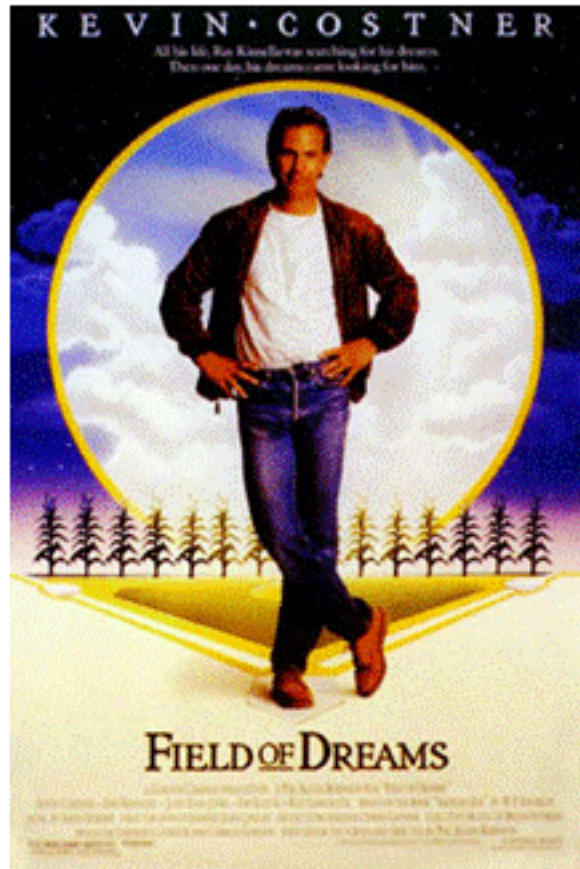
[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



*“If you build it...  
...they will come.”*

It turns out they don't.

The *opportunistic* model  
of reuse has been a  
thorough failure.



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Some companies were trying a new approach

CelsiusTech's ShipSystem 2000: A family of 55 shipboard Command-and-control systems

- Began in 1985 with two systems
- Built from single core asset base
- Hardware-to-software cost ratio changed from 35:65 to 80:20
- Software staff went from 210 to 30
- Cycle time went from 7-10 years to 2-3.
- Reuse ~ 90%
- Integration test of 1-1.5 million SLOC requires 1-2 people
- Re-hosting to a new platform/OS takes 3 months
- Cost and schedule targets are predictably met
- Customer satisfaction is high

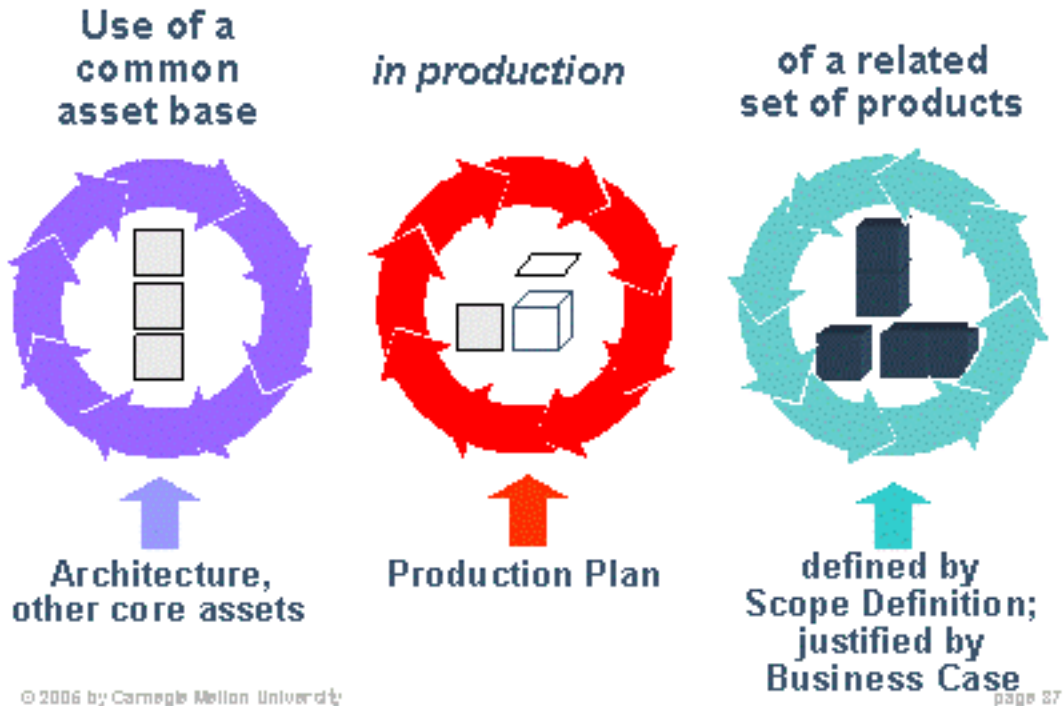


[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## The Key Concepts



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## What happened next?

Other companies were trying this approach

- On varying scales
- With varying success

Topics of interest at the time

- Domain analysis
- Software architecture
- Process improvement

- In 1996, the SEI published the CelsiusTech case study
- In 1997, we held our first product line workshop.
- In 2000, the first International Software Product Line Conference was held.

[Previous](#) [Next](#)

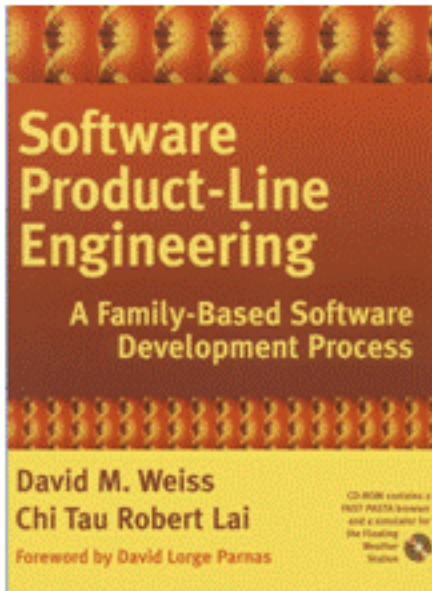
[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

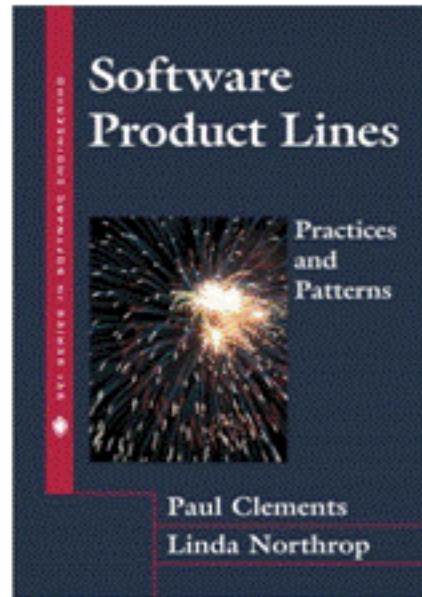




## Knowledge began to be codified



1999



2002

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## Other Forces Fell Into Place

Rapidly maturing, increasingly sophisticated software development technologies including *object technology*, *component technology*, *standardization of commercial middleware*.

A global realization of the *importance of architecture*

A universal recognition of the need for *process discipline*.

*Role models and case studies* emerging in the literature and trade journals.

*Conferences, workshops, and education programs* including product lines in the agenda.

Company and inter-company *product line initiatives*.

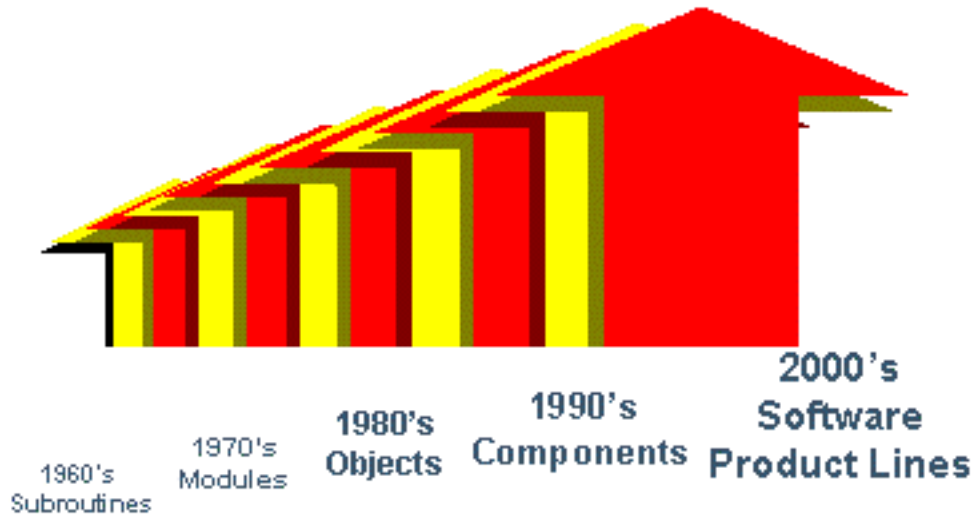
Rising recognition of the *amazing cost/performance savings* that are possible.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

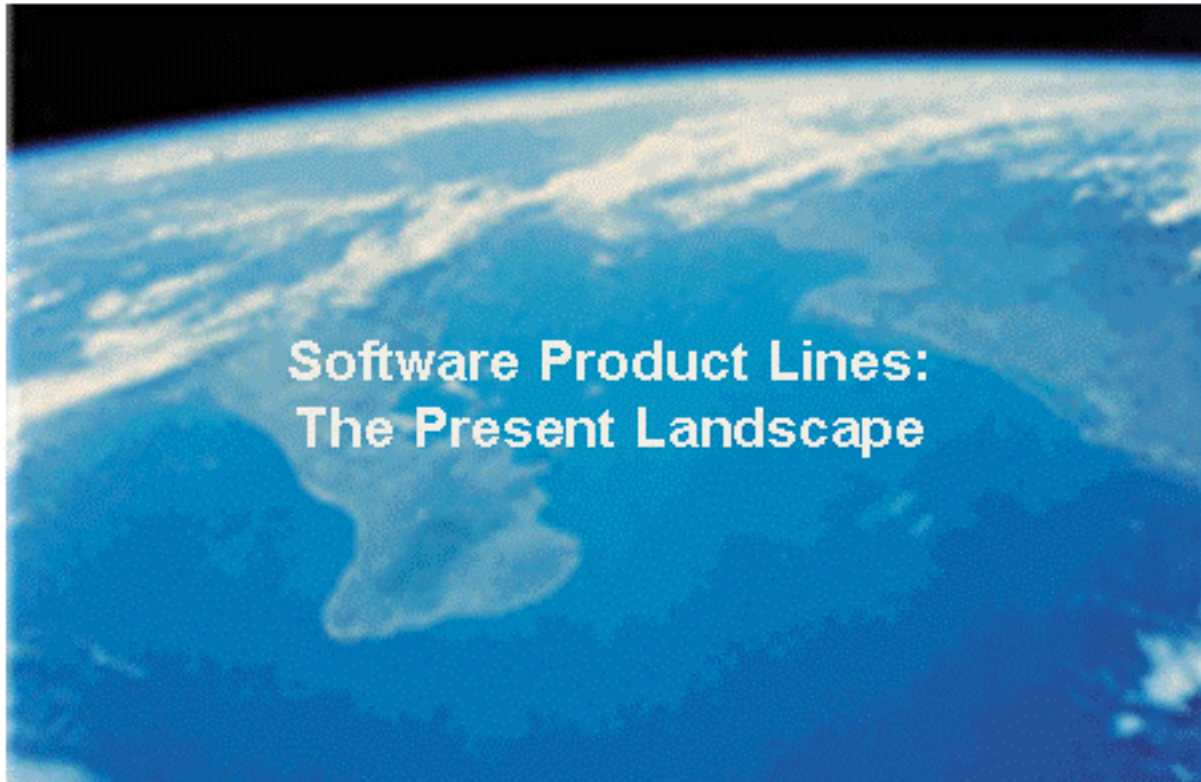
## Reuse: From Ad-Hoc to Systematic



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## What Do We Have Now?

Software product lines have emerged as an important, viable paradigm for software development.

- ➔ • Convincing evidence that software product line practice can bring about significant improvements in software development
- A body of knowledge and a set of standard models for software product lines
- A growing and energetic community of software product line practitioners

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## National Reconnaissance Office / Raytheon: Control Channel Toolkit

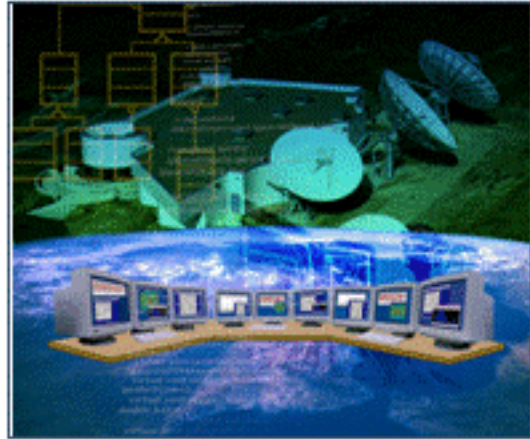
Ground-based spacecraft  
command and control systems

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



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Cummins, Inc.

World's largest manufacturer of large diesel engines.

- 20 product groups launched, yielding over 1000 separate engine applications
- 75% reuse, 360% productivity gain
- Product cycle time has plummeted. Time to first engine start went from 250 person-months to a few person-months.
- Software quality is at an all-time high.
- Product line approach let them quickly enter and then dominate the *industrial* diesel engine market.



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Nokia Mobile Phones

Product lines with >30 new products per year. Before product line, only 5-10 products per year.

Software product line is so successful, Nokia is selling their core asset base as a product.

Across products there are

- varying keys, displays, features, protocols
- 58 languages and 130 countries
- need for backwards compatibility
- need for product behavior change after release



[Previous](#) [Next](#)

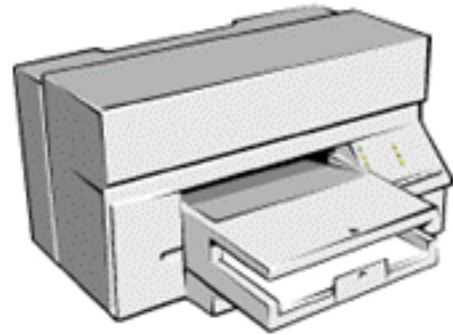
[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Hewlett Packard

### Printer systems

- 2-7x cycle time improvement (some 10x)
- 400% productivity improvement
- Sample Project
  - shipped 5x number of products
  - that were 4x as complex
  - and had 3x the number of features
  - with 4x products shipped/person



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Market Maker GmbH: MERGER

Internet-based stock market  
software

Each product "uniquely"  
configured

Three days to put up  
a customized system



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)





## What Do We Have Now?

Software product lines have emerged as an important, viable paradigm for software development.

- Convincing evidence that software product line practice can bring about significant improvements in software development
- ➔ • A body of knowledge and a set of standard models for software product lines
- A growing and energetic community of software product line practitioners

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Information Sources

Case studies,  
experience reports,  
and surveys

Workshops,  
and  
conferences



Applied research

Collaborations  
with customers  
on actual product lines

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Three Essential Activities



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## The Nature of the Essential Activities

All three activities are interrelated and highly iterative.

There is no "first" activity.

- In some contexts, existing products are mined for core assets.
- In others, core assets may be developed or procured for future use.



There is a strong feedback loop between the core assets and the products.

Strong management at multiple levels is needed throughout. Management orchestrates the processes to make the three essential activities work together

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Software Product Lines are Not About

**Just**

- **Libraries of objects, components, or algorithms**
- **Reuse when the software engineer is so inclined**
- **Reuse with no repeatable production process**
- **A configurable architecture**

=



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## Different Approaches - 1

*Proactive:* Develop the core assets first

- Define the product line's scope first and use it as a "mission" statement for the architecture and other assets
- Products come to market quickly with minimum code-writing.
- Requires up-front investment and predictive knowledge.

*Reactive:* Start with one or more products

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

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Different Approaches - 2

*Incremental:* Develop in stages with the plan 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.
- . . . . .

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

# Building the core asset base

## Core assets include:

Requirements and requirements analysis

Domain model

Software architecture

Performance engineering

Documentation

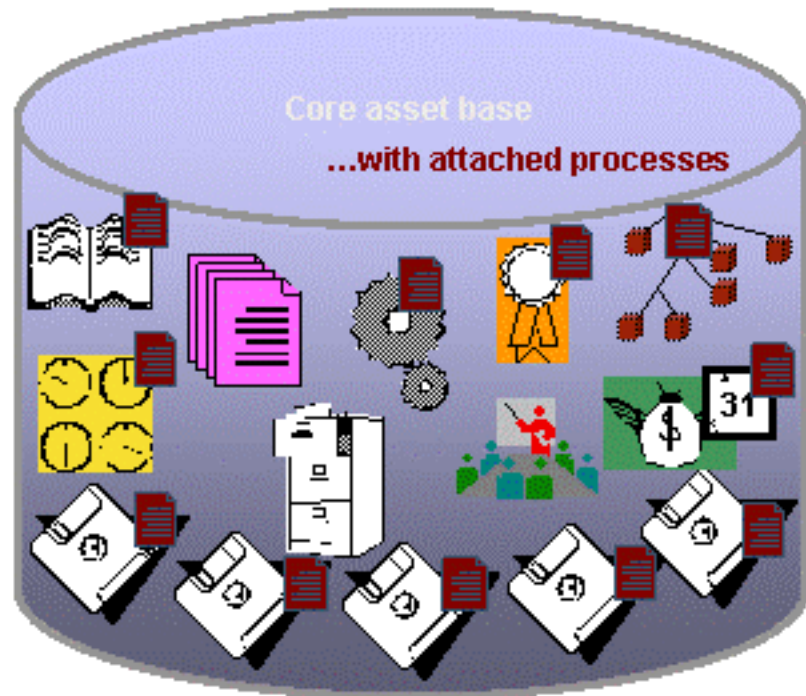
Test plans, test cases, and data

People knowledge and skills

Processes, methods, and tools

Budgets, schedules, workplans

...and Software

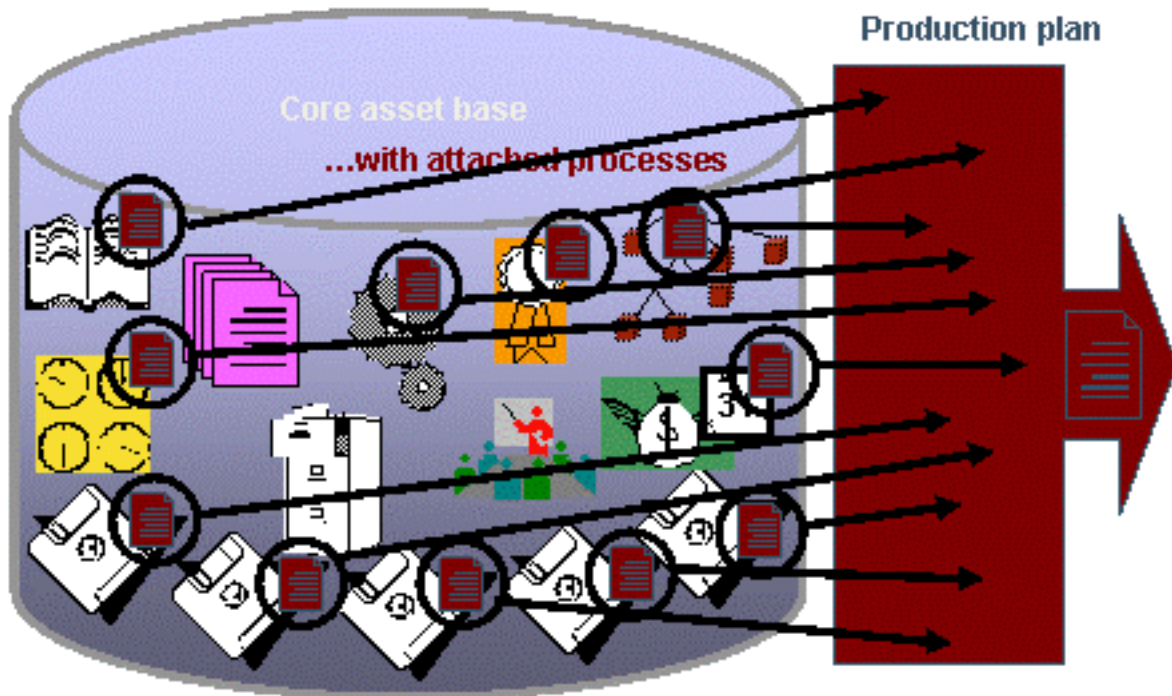


[Previous](#) [Next](#)

[Exit Slide Show](#)

- [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
- [31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)
- [58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Building the production plan



© 2006 by Carnegie Mellon University

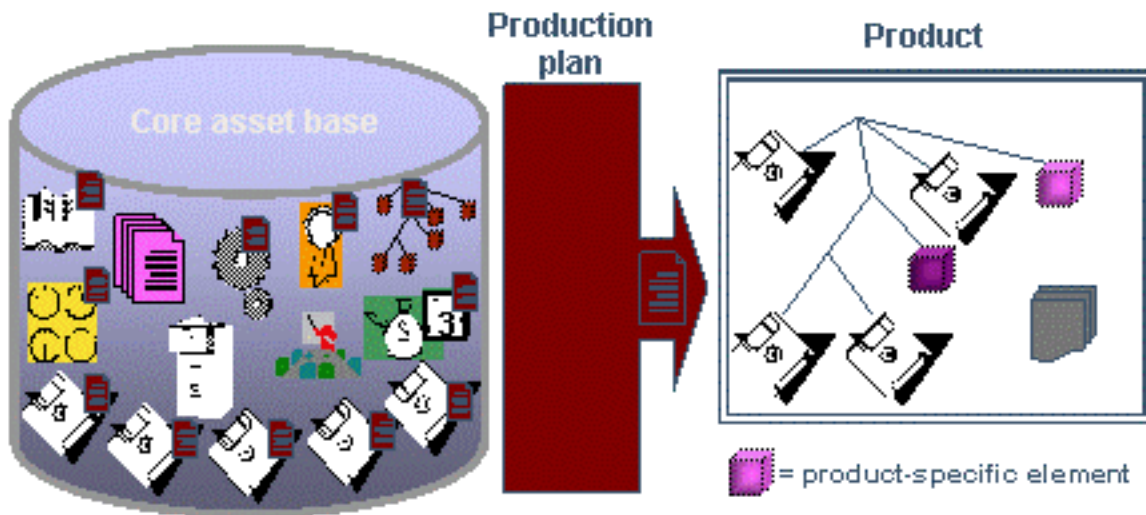
page 77

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

# Building a product...



[Previous](#) [Next](#)

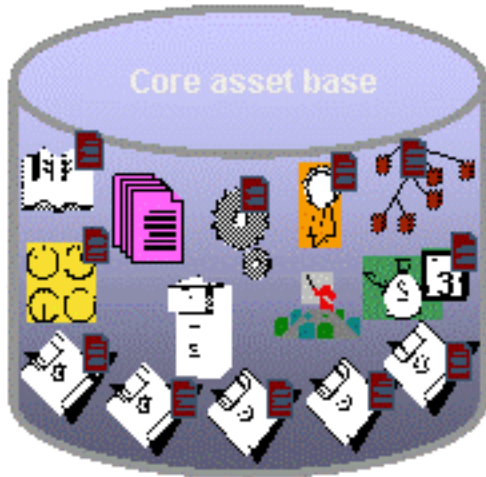
[Exit Slide Show](#)

- [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

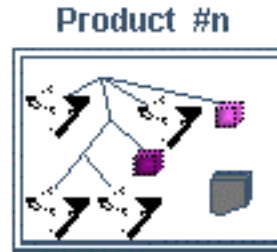


# Building subsequent products...

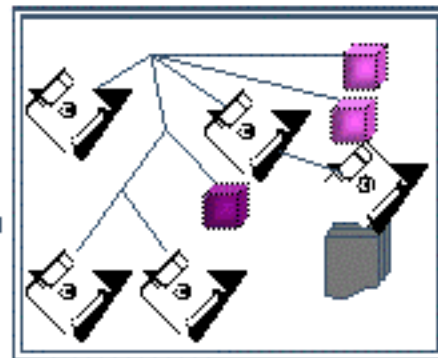
(proceed when ready)



Production plan



Product #n+1



 = product-specific element

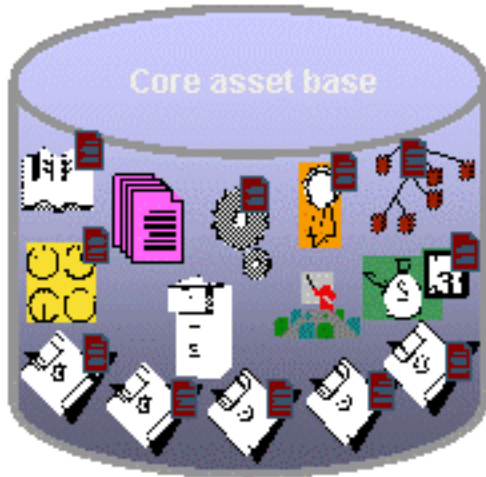
[Previous](#) [Next](#)

[Exit Slide Show](#)

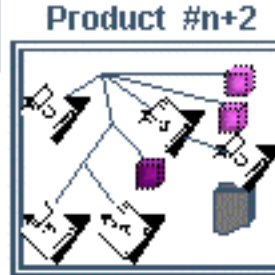
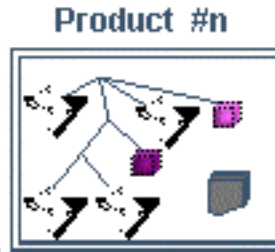
- [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
- [31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)
- [58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

(proceed when ready)

# Feedback



Updates  
to  
core  
assets



[Previous](#) [Next](#)

[Exit Slide Show](#)

- [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
- [31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)
- [58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

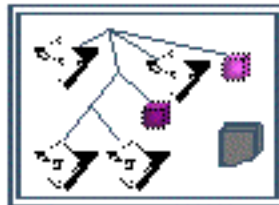
# Organization



The core asset base may be distributed and maintained by separate groups...



Product #n



...who cooperate to build a product.



[Previous](#) [Next](#)

[Exit Slide Show](#)

- [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
- [31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)
- [58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



# Product line experience has yielded important lessons

## Lessons in software engineering

- Architectures for product lines
- Testing variable architectures and components
- Importance of having and capturing domain knowledge
- Managing variations
- Important of large, pre-integrated chunks

## Lessons in technical/project management

- Importance of configuration management; why it's harder for product lines
- Product line scoping: What's in? What's out?
- Tool support for product lines

## Lessons in organizational management.

- People issues: how to bring about change, how to launch the effort
- Organizational structure: Who builds the core assets?
- Funding: How are the core assets paid for?
- Interacting with the customer has whole new dimension

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## SEI Framework for Product Line Practice

We've distilled these lessons into *practice areas*, areas of expertise necessary for engineering any software system, but which take on a different flavor in a product line context. These form a *framework for product line practice*.

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

© 2006 by Carnegie Mellon University

page 38

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



# Scoping

**Scoping** bounds a system or set of systems by defining those behaviors or aspects that are in and those that are out.

A scope definition lets you determine if a proposed new product can be reasonably developed as part of the existing (or planned) product line.

We want to draw the boundary so the product line is profitable.

- If the scope is too limited, there will be too few products to justify the investment in the core assets.
- If the scope is too large, the core assets will need to be impossibly general.
- If the scope encompasses the wrong products, the product line will not succeed.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)





## Growing a Scope Definition

The scope starts out broad and very general.

In a product line of Web software

- Browsers are definitely in.
- Aircraft flight simulators are definitely out.
- Email handlers are... well, we aren't sure yet.

The scope grows more detailed as our knowledge increases and the product line matures.

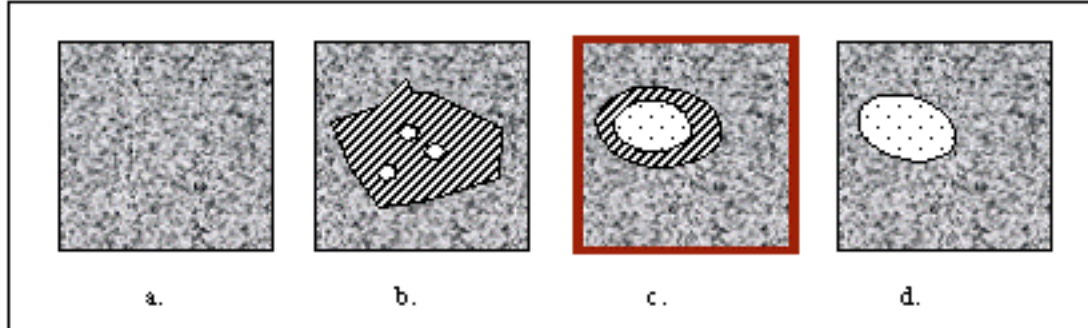
Initially, many possible systems will be "on the cusp," meaning their "in/out" decision must be made on a case-by-case basis. That's healthy.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Scope precision increases as we learn more...up to a point.



- a: space of all possible products
- b: early, coarse-grained "in/out" decisions
- c: product line scope with a healthy area of indecision
- d: product line scope = product line requirements

If many products appear on the cusp over time, you may need to reactively adjust the scope.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## Proactively Adjusting the Scope

Companies highly skilled at product line engineering purposely adjust their scope to take advantage of "nearby" market opportunities. Examples:

### **CelsiusTech**

- Ship system product line → Air defense system product line
- Forty percent of new system was complete on day one.

### **Cummins, Inc.**

- Automotive diesel engine product line → industrial diesel engine product line
- Quickly entered and dominated industrial engine market

### **Motorola**

- One-way pager product line → Two-way pager product line
- Same product line architecture used for both

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## A Snapshot of the Body of Knowledge

Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures  
by Hassan Gomaa

Domain Architectures: Models and Architectures for UML Applications  
by Daniel J. Duffy

Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools  
by Jack Greenfield

Domain-Driven Design: Tackling Complexity in the Heart of Software  
by Eric Evans

Component-Based Product Line Engineering with UML  
by Colin Atkinson, et al.

Source: [amazon.com](http://amazon.com)

Software Product Lines : Experience and Research Directions,  
by Patrick Donohoe (Editor)

Design and Use of Software Architectures, By Jan Bosch

Software Reuse : Architecture, Process and Organization for Business Success,  
by I. Jacobson, et al.

Managing Software Re-Use,  
by Wayne C. Lim

Reuse-Based Software Engineering: Techniques, Organizations, and Controls,  
by Hafeedh Mili, et al.

Practical Software Reuse,  
by Donald J. Reifer

Software Product Lines: Practices and Patterns,  
by Paul Clements and Linda Northrop

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## What Do We Have Now?

Software product lines have emerged as an important, viable paradigm for software development.

- Convincing evidence that software product line practice can bring about significant improvements in software development
- A body of knowledge and a set of standard models for software product lines
- ➔ • A growing and energetic community of software product line practitioners

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## We now have a *community* of software product line practitioners

We seek to

- understand the principles and practices behind software product engineering.
- help others successfully adopt the paradigm
- share experience

Major forums, workshops and conferences

- Software Product Line Conferences (SPLC 1-3)
- Product Family Engineering (PFE 1-5)
- **Merged in 2004 to become SPLC / SPLC-Europe**

Community web sites

- [www.sei.cmu.edu/productlines](http://www.sei.cmu.edu/productlines)
- [www.softwareproductlines.com](http://www.softwareproductlines.com)

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)





## SEI Software Product Line Curriculum

### Five one- and two-day courses

- Software Product Lines
- Adopting Software Product Lines
- Developing Software Product Lines
- Product Line Technical Probe Team Training
- Product Line Technical Probe Leader Training



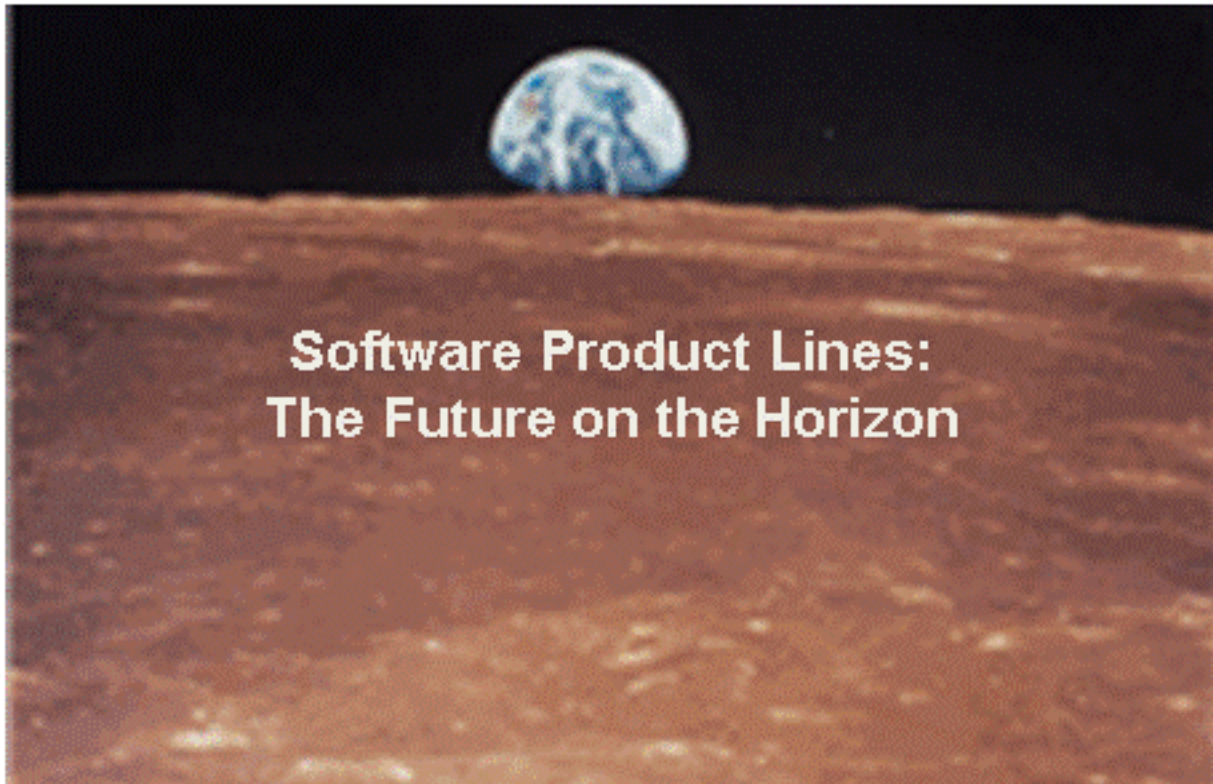
### Three certificate programs

- Software Product Line Professional Certificate
- Product Line Technical Probe Team Member Certificate
- Product Line Technical Probe Leader Certificate

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



© 2006 by Carnegie Mellon University

page 107

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Current research topics

### ➔ Product lines and aspects

- An *aspect* is a concern that cuts across units of code in a program.
- Aspects represent a way to achieve multi-dimensional separation of concerns.
- Making a "global" change merely requires changing the aspect and then weaving the aspect through the code.
- Example: Changing a program's overall fault-handling approach



Can aspects be used to represent the variation points of a software product line?

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



Carnegie Mellon  
Software Engineering Institute



9th International  
Software Product  
Line Conference  
26-29 September 2005  
Rennes, France

*Call for Participation*

**Aspects and Software Product Lines:  
An Early Aspects Workshop at SPLC-Europe 2005  
*Rennes, France*  
26 September 2005**

<http://www.early-aspects.net/events/splc2005ws/> / <http://www.sse.uni-essen.de/SPLC2005>

**Background**

Aspect-oriented software development (AOSD) techniques aim to systematically identify, modularise, represent and compose broadly-scoped properties such as security, mobility, availability and real-time constraints – the so called *crosscutting concerns* – throughout the software life cycle. The initial focus of AOSD techniques has been at the programming level resulting in a range of aspect-oriented programming (AOP) techniques, and a number of application studies validating them in real-world application scenarios. Recently, however, the term “Early Aspects” has emerged to refer to AOSD techniques that focus on treatment of crosscutting concerns that arise *before* implementation, especially in requirements analysis and architecture design. Five Early Aspects workshops have been held to date to explore this area (see <http://early-aspects.net/>). One very promising application of early aspects is in

[Previous](#) [Next](#)

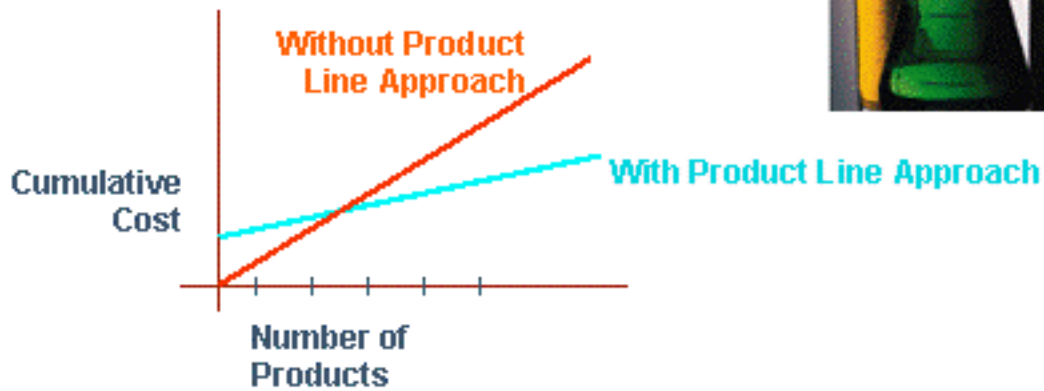
[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Current research topics

### ➔ Product lines economic models

Here is our "classic" economic model:



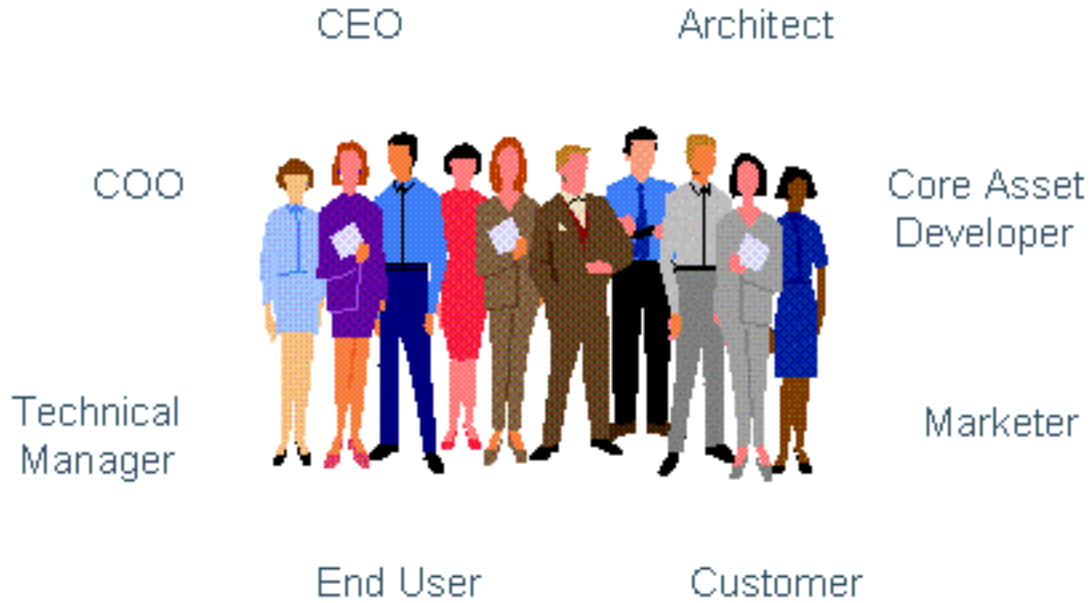
This comes from a world where economic advantage is anecdotal, intuitive, and based on single data points.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Individuals Who Benefit – Anecdotally



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)





# SIMPLE

Structured Intuitive Model for Product Line Economics

- $C_{org}()$  : cost to an organization of adopting the product line approach
- $C_{cab}()$  : development cost to develop a core asset base
- $C_{unique}()$  : cost to develop unique software that itself is not based on a product line platform.
- $C_{reuse}()$  : cost to reuse core assets in a core asset base

Example: Cost of building a product line of  $n$  products =

$$C_{org}() + C_{cab}() + \sum_{i=1}^n (C_{unique}(product_i) + C_{reuse}(product_i))$$

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Current research topics

### ➔ Product line practice patterns

Patterns represent pre-packaged solutions to known problems.

Design patterns are well-known in software engineering.

Organizational or process patterns also provide solutions or solution strategies to attack recurring problems

- How do I determine the scope of my product line?
- How do I establish a production capability?
- How do I launch a product line effort?
- How do I adopt the product line approach across my whole organization?

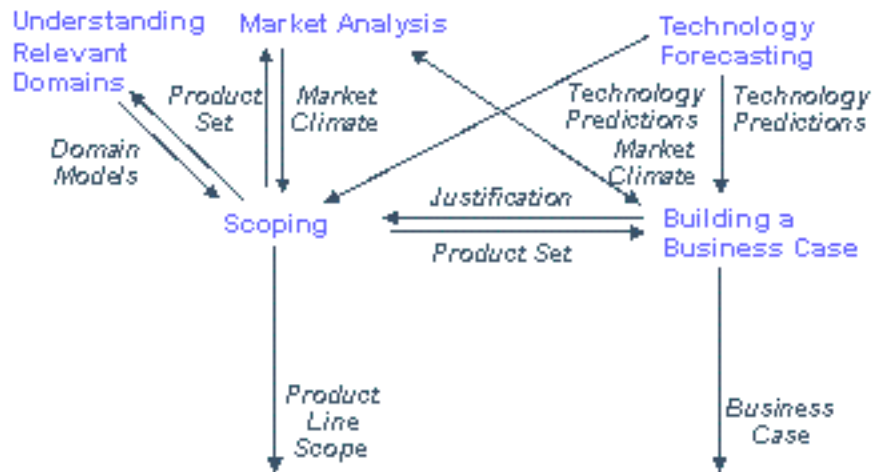


[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Example: “What to Build” Pattern



Patterns can provide

- Guidance for managers, blueprint for organizations
- Basis for process enactment, workflow and rules engines

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

## Other research topics

**Product line testing**

**Variability mechanisms and  
product line architectures**

**Lightweight, low-cost, low-barrier  
product line approaches**

**“Non-traditional” product lines**

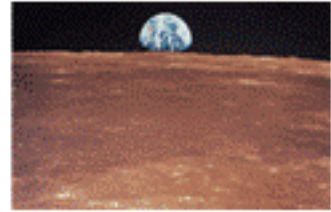
- **Product lines of components for other product lines**
- **Cross-organizational product lines**



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



## In the Future

Software product line practice will become a first-class part of software engineering professional practice.

- Product line engineering will be a standard part of software engineering curricula
- Software architects will routinely consider building architectures for a family of systems, using variability mechanisms wisely
- Product/project managers will always ask whether a project being launched can be a product line.
- Decision-makers will have the tools, models, and experience base necessary to help them make the right decisions and carry them out.
- Development tools will handle the development of a family of systems built and maintained in parallel.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)

# Key Themes Among Successful Product Lines

- Sophistication in the domain
- A legacy base from which to build
- Architectural excellence
- Process maturity
- Management commitment
- Capacity for introspection



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) 64 [65](#) [66](#)





## Three Observations

Reusing software is **not** the same as establishing a software product line.

It is essential to maintain the reused software (and its associated technical and developmental artifacts) and manage its evolution and growth separately from the places where it is used.

Mature product line organizations prioritize the health and welfare of their core asset base – that is, their *product production capability* – over that of any individual product.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)



Carnegie Mellon  
Software Engineering Institute



9th International  
**Software Product  
Line Conference**  
26-29 September 2005  
Rennes, France

---

**Home**  
**Organization**  
**Program Committee**  
**Workshops**  
**Tutorials**  
**Registration**  
**Local Information**  
**Sponsors**  
**Contact**

9th International  
**Software Product  
 Line Conference**  
**(SPLC-EUROPE 2005)**  
 26-29 September 2005  
 Rennes, France



**News**  
**Five Workshops** have been accepted. 2005-05-03  
**The submission system** is closed. 2005-04-12  
**Sponsoring:** Find Sponsorship rules [here](#). 2005-03-18  
**SPLC'05 Online:** The SPLC'05 Homepage goes online. 2004-12-23  
**Submission Deadlines:**  
 Paper submission: April 11, 2005  
 Author notification: June 1, 2005

With SPLC-Europe 2005 we celebrate the foundation of a new conference series, the International Software Product Line Conference (SPLC), which results from the merger of the former events SPLC (Software Product Line Conference started 2000 in the USA) and PFE (Product Family Engineering started 1996 in Europe).  
 SPLC is thus now the premier forum for practitioners, researchers and educators to present and discuss the most recent ideas, innovations, trends, experiences, and concerns in the area software product lines and software product family engineering. From 2005 on, the SPLC conference will be held yearly in either Europe, America or Asia.  
**Goal:** The technical program of SPLC-Europe 2005 will include research papers, case studies, experience and experimental reports, and problem statements in the area of software product line engineering. The objective is to continue the dialogue between software product line practitioners and researchers on the benefits, obstacles, and weaknesses of applying software product line principles, techniques, methods, processes, and tools in an industrial or organizational setting.

© 2006 by Carnegie Mellon University

page 151

[Previous](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)  
[31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#)  
[58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#)