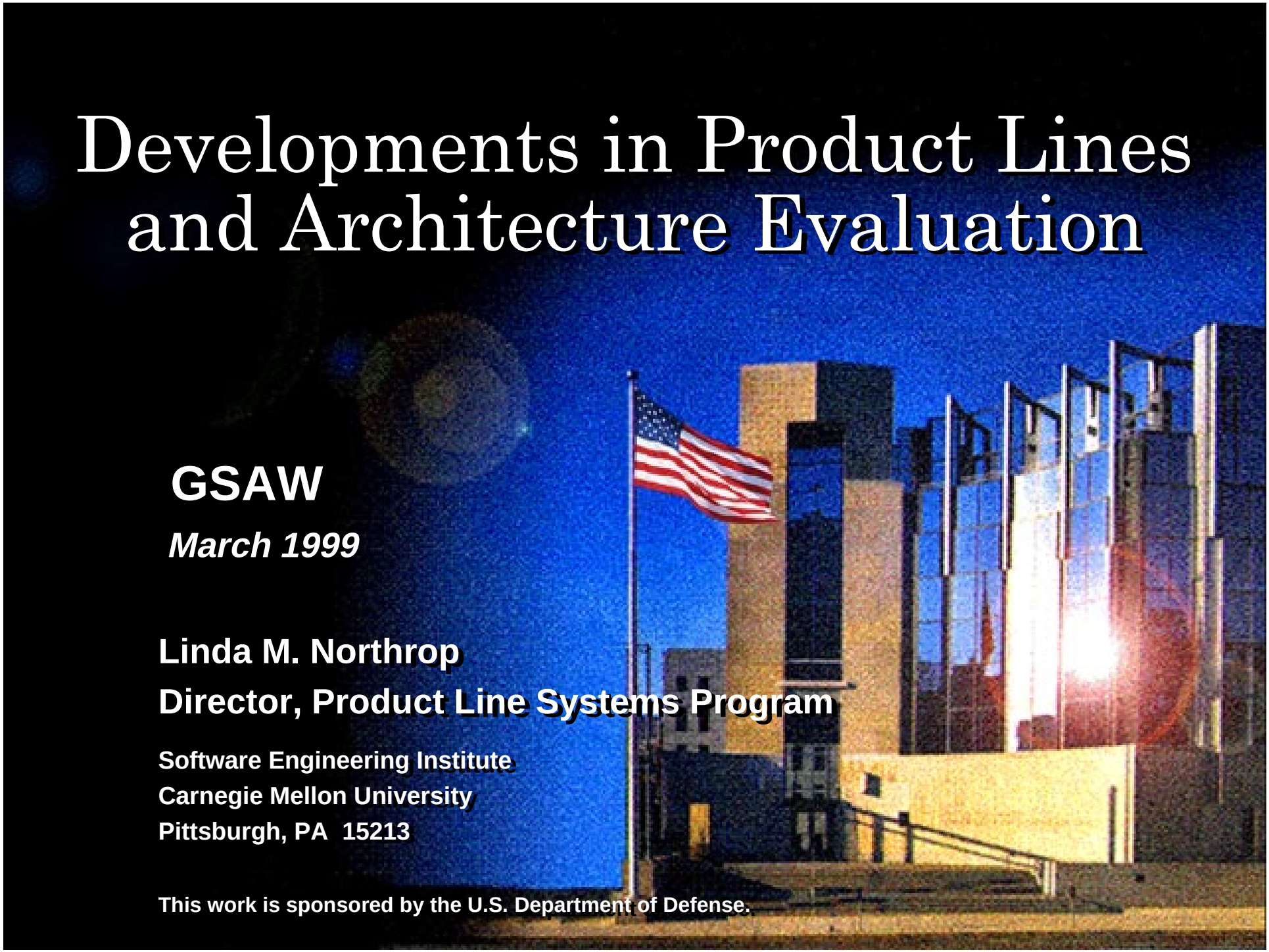
Developments in Product Lines and Architecture Evaluation



GSAW

March 1999

Linda M. Northrop

Director, Product Line Systems Program

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This work is sponsored by the U.S. Department of Defense.

Today's Talk

What I Told You Last Year

Update

- Commercial
- DoD
- SEI Product Line Practice Framework

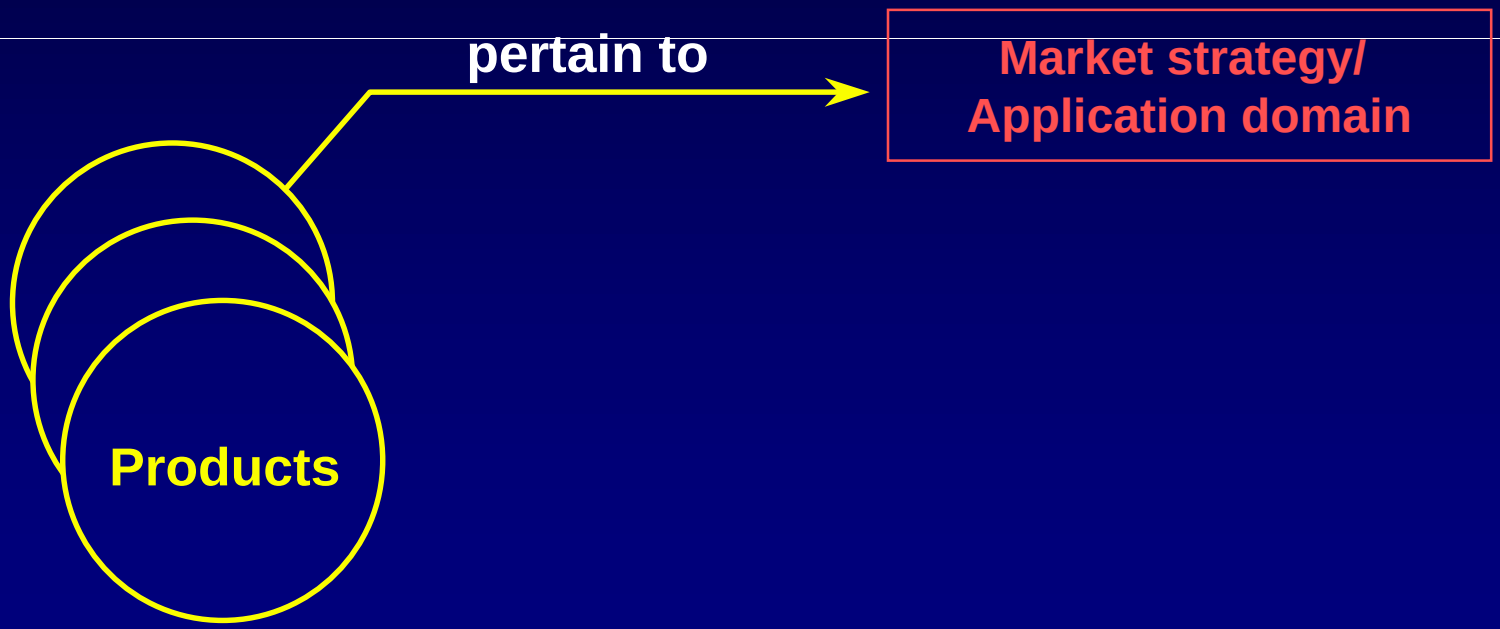
Architecture Evaluation

- Why?
- ATAM
- DALI

Conclusion

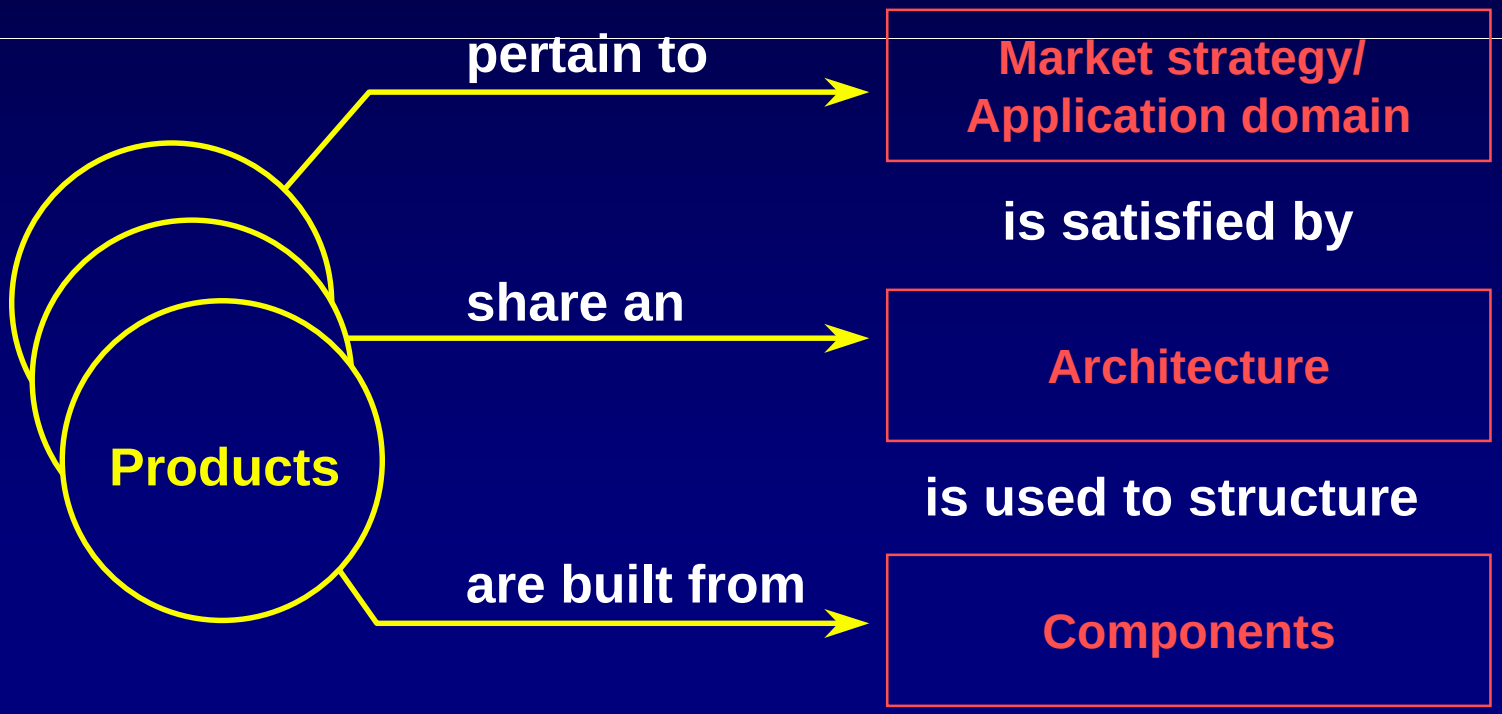
What Is a Product Line?

A product line is a group of products sharing a common, managed set of features that satisfy specific needs of a selected market or mission.



Software Product Lines

A product line is a group of products sharing a common, managed set of features that satisfy specific needs of a selected market.

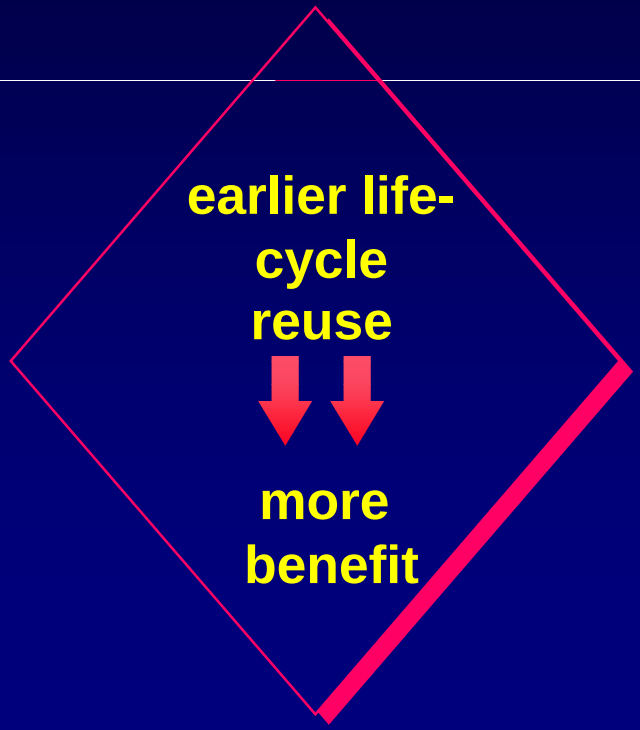


How Do Product Lines Help?

Product lines amortize the investment in these and other *core assets*:

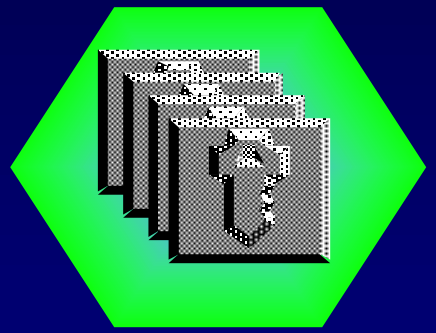
- requirements and requirements analysis
- domain model
- software architecture and design
- performance engineering
- documentation
- test plans, test cases, and data
- people: their knowledge and skills
- processes, methods, and tools
- budgets, schedules, and work plans
- components

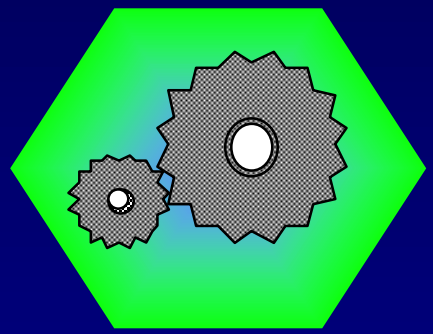
product lines = strategic reuse

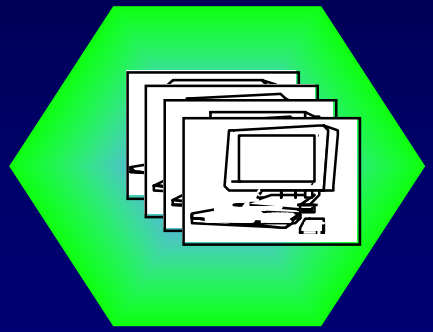


The Key Concepts

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







Real World Motivation

Product Line Practice needed to

- to achieve large scale productivity gains
- to improve time to market
- to maintain market presence
- to sustain unprecedented growth
- to compensate for an inability to hire
- to achieve systematic reuse goals
- to improve product quality
- to increase customer satisfaction

State of Product Line Practice

Commercial Sector

- has succeeded with systematic reuse through software product lines in many domains (HP, AT&T, Lucent, Nokia, Raytheon, Lockheed Martin, Hughes, Motorola, Ericsson, Bosch, CelsiusTech, ALLTEL, Phillips, Caterpillar, Cummins, Buzzeo, etc.)
- is seeing a growing trend toward product lines (Boeing, Ford, Chrysler, Allied Signal, etc.)
- has realized reduced time to market, improved quality, reduced development and maintenance costs, improved efficiency, increased interoperability through software product lines
- has developed proprietary technical, business, management, and acquisition product line practices

How to Bridge the Gap???

- validate
- distill
- codify
- tailor
- transition

DoD

- has requirements for faster, better, cheaper, architecture-based reuse
- has need for systematic reuse
- has had many reuse efforts and a few successes
- is attracted to product line idea but has not been able to make it happen as a practice
- wants proven, off-the-shelf practices and support tools

Today's Talk

What I Told You Last Year

Update

- Commercial
- DoD
- SEI Product Line Practice Framework

Architecture Evaluation

- Why?
- ATAM
- DALI

Conclusion

Progress on Commercial Side

Cummins
- 4.5 years in product line development
- over 20 products successfully launched
- system build and integration went from roughly 1 year to 3 days

Motorola
- product line initiative

Thomson-CSF
- product line initiative

Computas
- toolkit-based information systems product line

Philips
- medical imaging product line

Siemens
- medical imaging product line

10Fold
- information systems product line

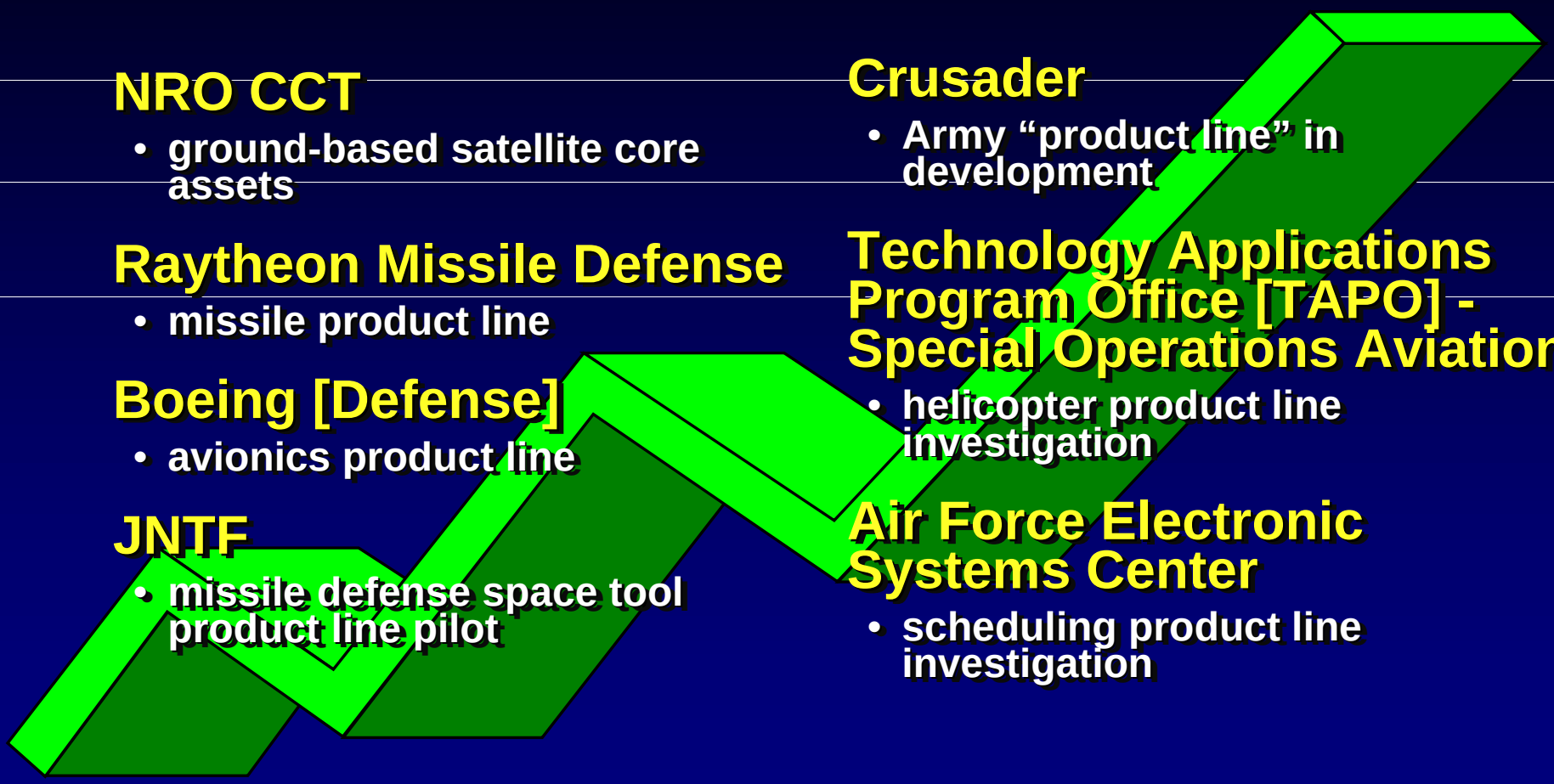
PRAISE Project
- inter-company product line research

Telesoft S.p.A.
- telecom product line in acquisition context

Bosch
- product line business unit

Deimler-Benz
- configuration generator

Progress on DoD Side



NRO CCT
- ground-based satellite core assets

Raytheon Missile Defense
- missile product line

Boeing [Defense]
- avionics product line

JNTF
- missile defense space tool product line pilot

Crusader
- Army "product line" in development

Technology Applications Program Office [TAPO] - Special Operations Aviation
- helicopter product line investigation

Air Force Electronic Systems Center
- scheduling product line investigation

Evidence of Growing Maturity

Universities have latched onto software product lines as an area of research.

Software product line concepts are being targeted in some European universities.

Product line workshops are being organized [OOPSLA, ECOOP, ARES].

SEI Working to Bridge the Gap - 1

NRO's Control Channel Toolkit (CCT) Program, begun in 1997, provides a common architecture and components for multiple ground satellite systems.

IMPACT: increased quality, decreased time to field, projected 27.8% savings in sustainment costs projected for the next 9-year period

Joint National Test Facility launched a pilot missile defense space tool (MDST) to test product line architecture concepts.

IMPACT: Warfighter will have access to most current and realistic SBIRS representations faster than ever before.

Robert Bosch

Caterpillar

US Army Special Operations Aviation

SEI Working to Bridge the Gap - 2

PLP Technical Reports & Framework

1st PLP Workshop (SEI) Dec 1996
commercial, DoD participants

3rd PLP Workshop (SEI) Dec 1998
commercial participants

Reuse and Product Lines Workshop (WISR8) Mar 1997
commercial, academic participants

DoD PLP Workshop (SEI) Mar 1998
DoD participants

2nd PLP Workshop (SEI) Nov 1997
commercial participants

Essentials of PLP Presentation (STC) April 1997 April 1998
DoD audience

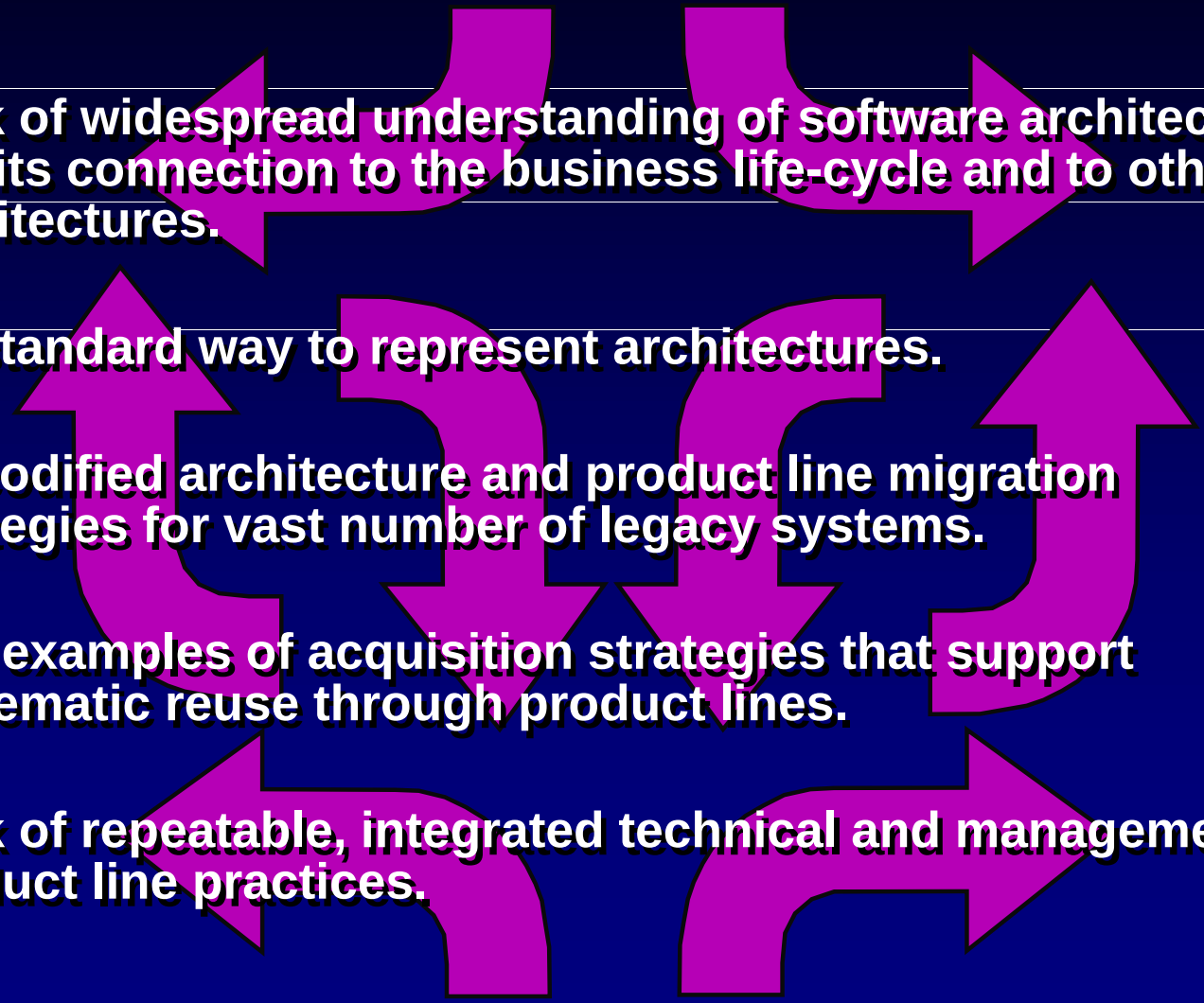
Software Architectures in PL Acquisitions (with PLIAT) June 1997, June 1998
DoD participants

SEI Symposium Presentations Aug 1997 Sept 1998
DoD, commercial audience

Object Technology and Product Lines (OOPSLA '97, '98) Oct 1997, Oct 1998
commercial, DoD participants

2nd DoD PLP Workshop - March 1999

Current Challenges



Lack of widespread understanding of software architecture and its connection to the business life-cycle and to other architectures.

No standard way to represent architectures.

No codified architecture and product line migration strategies for vast number of legacy systems.

Few examples of acquisition strategies that support systematic reuse through product lines.

Lack of repeatable, integrated technical and management product line practices.

Product Line Practice

Contexts for product lines vary widely

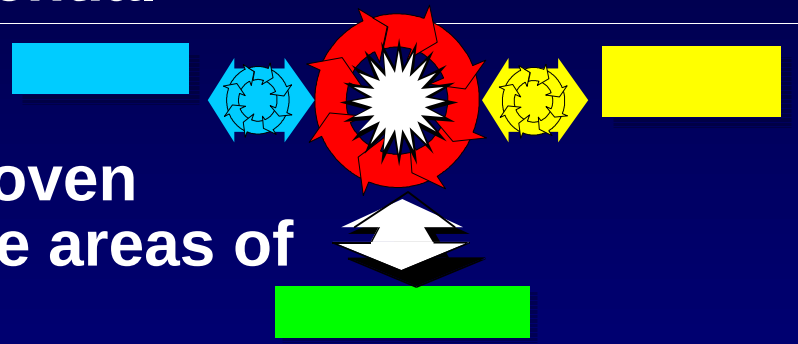
- nature of products
- nature of market or mission
- organizational infrastructure
- process maturity
- artifact maturity

But there are universal essential elements and practices.

SEI Product Line Practice Framework

Web-based, evolving document

Describes product line essential activities



Describes essential and proven product line practices in the areas of

- **software engineering**
- **technical management**
- **organizational management**

Addresses development and acquisition contexts

Framework Goals

Identify the foundational concepts underlying the software product lines and the essential issues to consider before fielding a product line.

Identify practice areas that an organization creating or acquiring software product lines must master.

Define practices in each practice area where current knowledge is sufficient to do so.

Provide guidance to an organization about how to move to a product line approach for software.

Framework Audience

Members of organizations who are in a position to make or influence decisions regarding the adoption of product line practices

SEI Information Sources

Case studies, experience reports, and pilots

Workshops

Surveys

Collaborations with customers on actual product lines

Current Status of Framework

Version 1.0 is now on our Web Site

http://www.sei.cmu.edu/plp/framework.html

Version 1.0 differs from earlier drafts

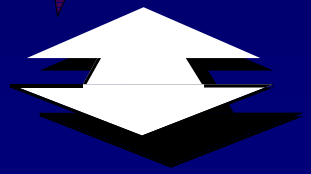
- ✔ shorter introduction
- ✔ slightly different “signature” pictures
- ✔ “Enterprise Management” category is now “Organizational Management”
- ✔ modified list of practice areas
- ✔ now addresses acquisition context
- ✔ describes six practice areas

Practice Area Categories

SOFTWARE ENGINEERING

TECHNICAL MANAGEMENT

ORGANIZATIONAL MANAGEMENT



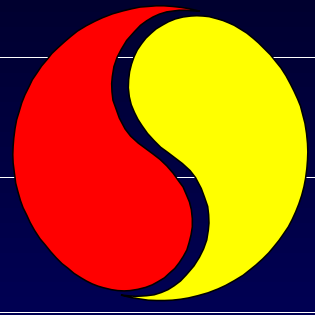
Practice Area Descriptions

For individual practice areas the framework has

- **introductory description**
- **aspects peculiar to product lines**
- **how applied to core asset development / acquisition**
- **how applied to product development / acquisition**
- **specific practices**
- **practice risks**
- **references**

Software Engineering Practice Areas

- Domain Analysis
- Mining Existing Assets
- Architecture Exploration and Definition
- Architecture Evaluation
- Component Development
- Testing
- Requirements Management
- COTS Utilization
- Software System Integration

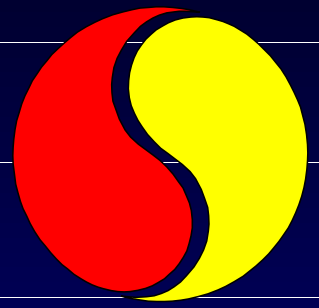


in Version 1.0

in Version 2.0

Technical Management Practice Areas

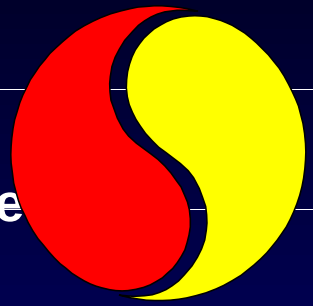
- Data Collection, Metrics and Tracking
- Product Line Scoping
- Configuration Management
- Process Modeling and Implementation
- Planning and Tracking
- Make, Buy, Mine, Outsource Analysis
- Technical Risk Management
- Tool Support



in Version 1.0

in Version 2.0

Organizational Management Practice Areas



- Achieving the Right Organizational Structure
- ~~Building and Communicating a Business Case~~
- Funding
- ~~Market Analysis~~
- Developing and Implementing an Acquisition Strategy
- Operations
- Training
- Customer and Supplier Interface Management
- Technology Forecasting
- Launching a Product Line
- Product Line Institutionalization
- Organizational Risk Management

❄ in Version 1.0

in Version 2.0

Remarks

The SEI Framework for Software Product Line Practice is intended to be a living document.

Version 1.0 is the first step.

Future versions will

- **build on current foundation**
- **complete other practice area descriptions**
- **describe a small number of product line scenarios**
- **provide a list of frequently asked questions and their answers**

In addition, the SEI is producing generic product line artifacts, case studies, technical reports, and workshop reports.

Today's Talk

What I Told You Last Year

Update

- Commercial
- DoD
- SEI Product Line Practice Framework

Architecture Evaluation

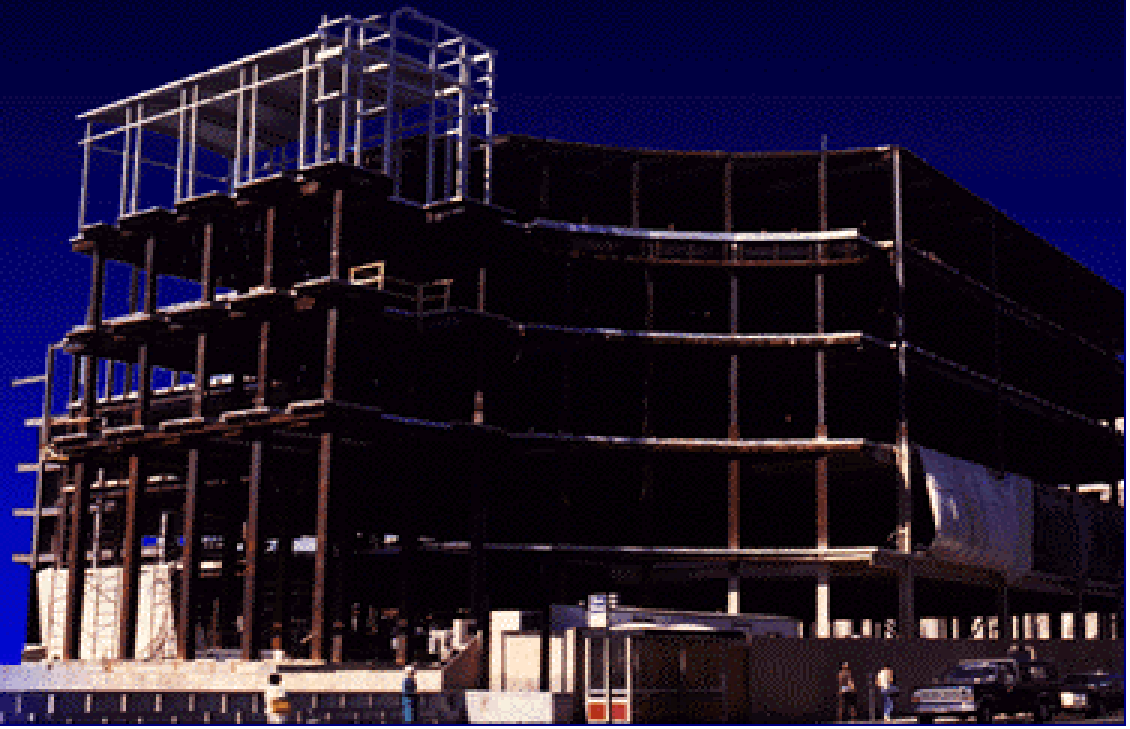
- Why?
- ATAM
- DALI

Conclusion

What Is Software Architecture?

Software architecture is the structure or structures of the system, which comprise software components, the externally visible properties of these components, and the relationships among them.

The exact structures to consider and the ways to represent them vary.



Why Architecture?

Represents *earliest* design decisions

- hardest to change
- most critical to get right

First design artifact addressing

- performance
- modifiability
- reliability
- security

Key to systematic *reuse*

The right architecture paves the way for system success.

The wrong architecture usually spells some form of disaster.

Why Architecture Evaluation?

Architecture evaluation

- **can be done early, when there is time for mid-course correction**
- **is relatively inexpensive**
- **is best commercial practice**

Early quality evaluation is cost effective (AT&T: 10% productivity increase/project)

There are competing requirements; decisions must be made early

Software quality cannot be appended later

SEI's Architecture Tradeoff Analysis Method (ATAM)

ATAM is an architecture evaluation method that

- **focuses on multiple quality attributes**
- **illuminates points in the architecture where quality attribute *tradeoffs* occur**
- **generates a context for ongoing quantitative analysis**
- **utilizes an architecture's vested stakeholders as authorities on the quality attribute goals**

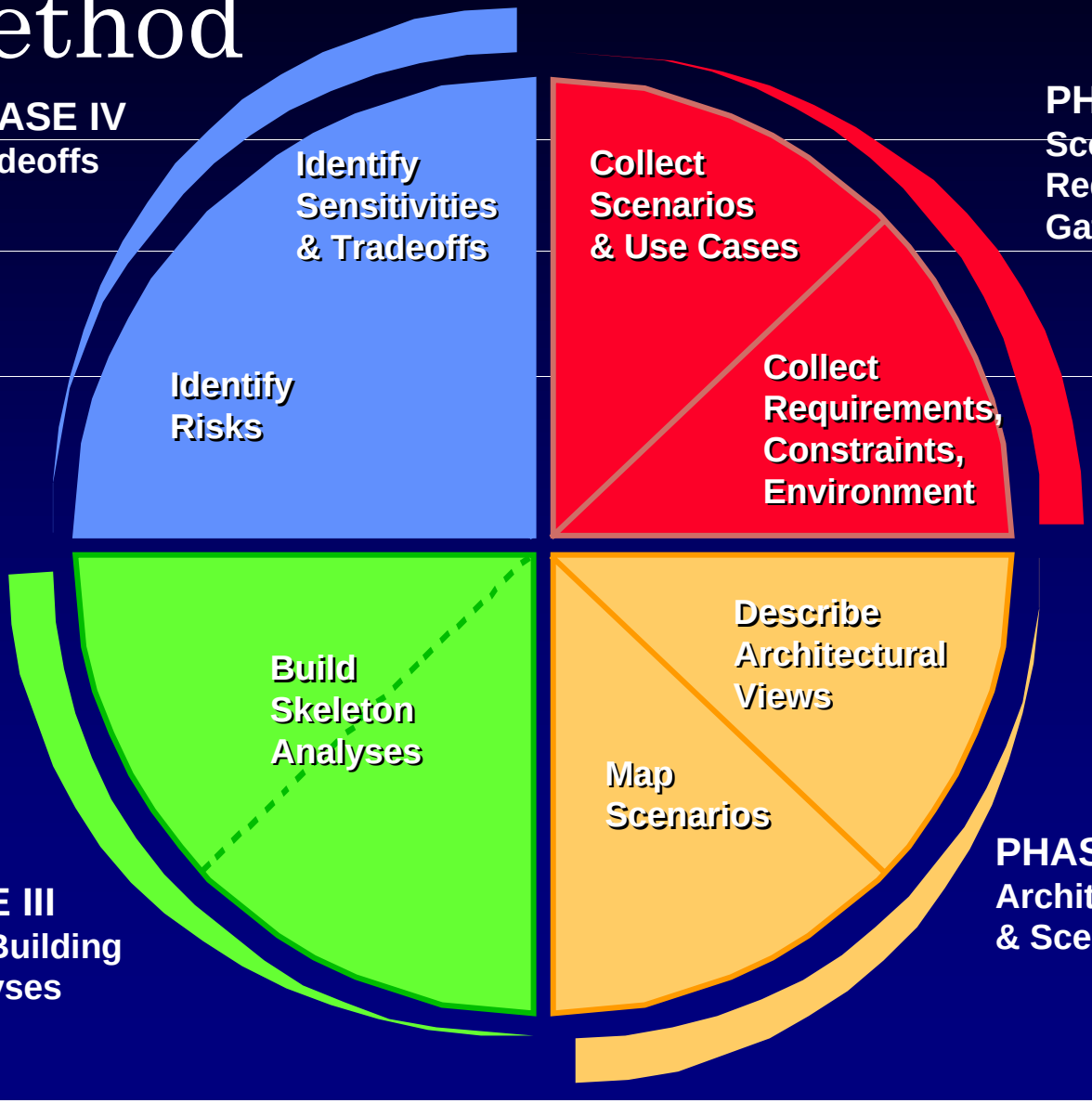
ATAM and Risks

The point of an ATAM analysis is not to provide precise analyses . . . the point is to discover areas of high potential risk in the architecture.

We want to find *trends:* correlations between architectural parameters and measurable properties.

These areas can then be made the focus of risk mitigation activities: e.g. further design, further analysis, prototyping.

Architecture Tradeoff Analysis Method

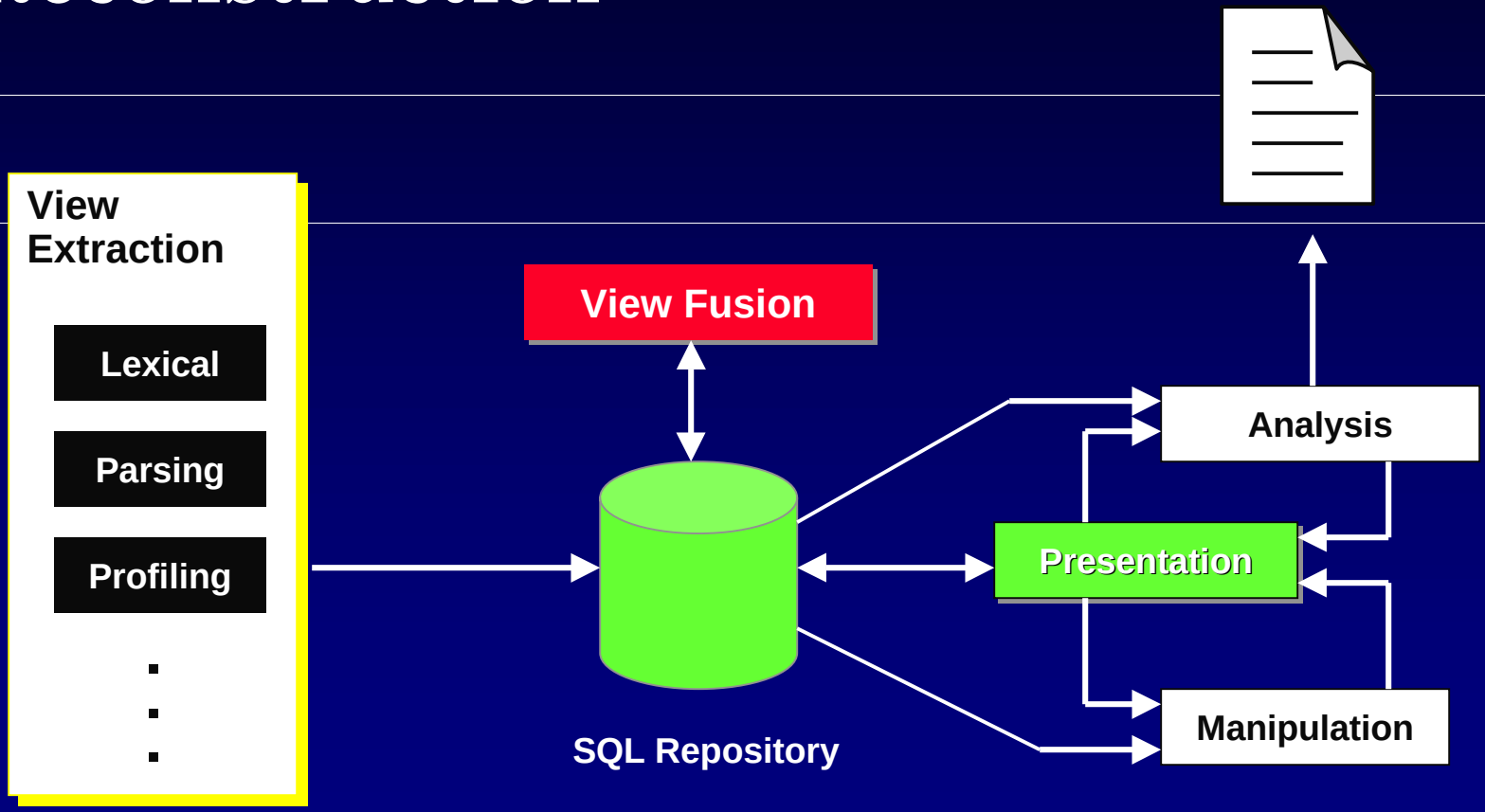


Demonstrated Value of Architecture Evaluation

SEI's Architecture Tradeoff Analysis Method (ATAM) pilot on Army's Mortar Fire Control System resulted in

- greatly improved architectural documentation
- better understanding of the requirements
- stakeholder buy-in
- discovery of missing performance and survivability requirements
- highlighting a previously unknown tradeoff point in the architecture
- delineation of design options to mitigate risks of this tradeoff

Dali: A Workbench for Software Architecture Extraction and Reconstruction



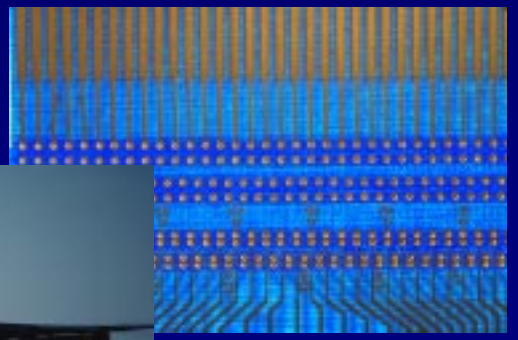
Why Extraction and Reconstruction?

Architectures are frequently undocumented.

Architectural drift and erosion are unavoidable.

However, we need to be able to reason about the architectures of existing systems:

- **for reuse**
- **in support of product line development**
- **for *analysis***



Architectural Conformance

Question: If my architecture was designed with a particular quality attribute in mind, does the property hold for my target system?

(Probable) Answer: Who knows?

The architecture of the implemented system must *conform* to the as-designed architecture.

Status

We have extracted architectures using:

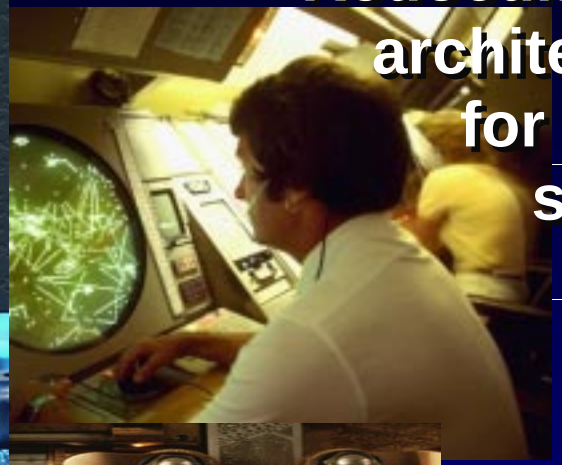
- **C**
- **C++**
- **Fortran**
- **Assembly language**
- **various other info: makefiles, shell scripts, log files, linker map files, instrumentation traces**

Typical system: 50 KLOC - 1 MLOC

Some Current Applications

Understanding legacy water quality models

Redocumenting architectures for physics simulations



Understanding architectural dependencies in embedded control software for reengineering

Today's Talk

What I Told You Last Year

Update

- Commercial
- DoD
- SEI Product Line Practice Framework

Architecture Evaluation

- Why?
- ATAM
- DALI

Conclusion

Conclusion

Architecture evaluation and product line practices hold great potential for reduction in time to market, cost reduction, and technical risk mitigation.

Considerable and exciting progress is ongoing in both areas.

For Additional Information

Telephone	**412 / 268-7638**
FAX	**412 / 268-5758**
Email	**lmn@sei.cmu.edu**
World Wide Web	**http://www.sei.cmu.edu/plp**
U.S. mail	**Linda M. Northrop** **Software Engineering Institute** **Carnegie Mellon University** **Pittsburgh, PA 15213-3890**