Overview of *Software Reuse Initiative Technology Roadmap*

Barbara A. Fleming

May 1994

Defense Information Systems Agency
Joint Interoperability & Engineering Organization

Center for Information Management
Software Systems Engineering Directorate

Agenda

- **Part 1: Background**
- **Part 2: Contents of the Roadmap**
- **Part 3: Plans**

Part 1: Background

- **Motivation**
- **Approach/Status**

Motivation

- Software is becoming more significant portion of cost in DoD
- Software reuse identified as significant potential contributor to cost reduction (50%) in draft *DoD Software Technology Strategy* (1991)
- SRI set up to coordinate DoD software reuse activities
- Development and maintenance of Reuse Technology Roadmap identified as "one of the most crucial technical activities" in draft of SRI Program Management Plan
- Software reuse is immature area; investment decisions involve risk
- Software reuse related to many technologies; multiple investment candidates
- Roadmap intended to address overall reuse picture and assist in making software reuse investment decisions

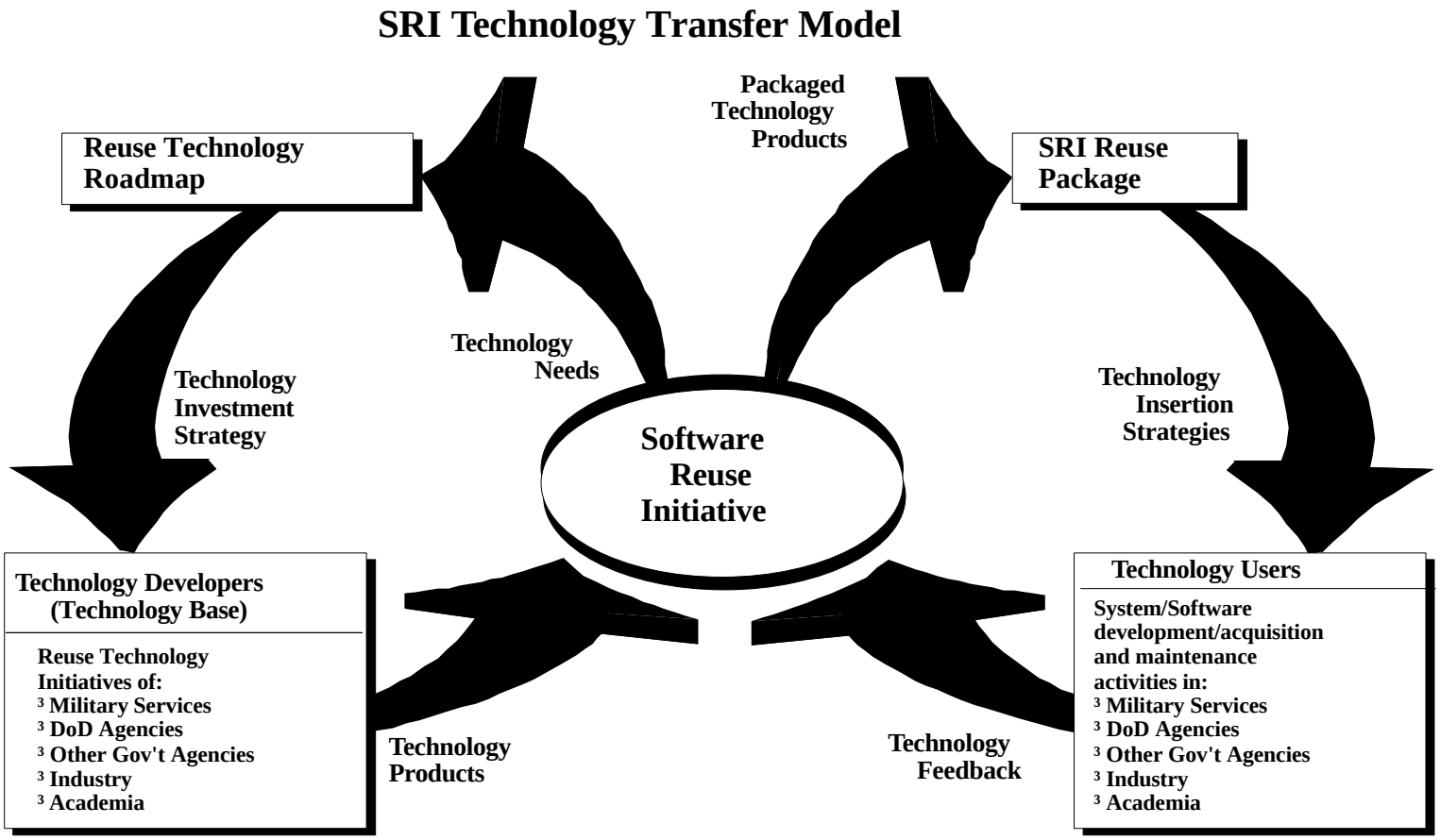
Approach/Status

- Task managed by DISA/JIEO/CIM (Barbara Fleming) for SRI
- Task performed by MITRE and the University of Houston
- Extracted information from experience, literature survey, data gathered from software engineering managers / researchers
- Followed guidance of expert review panel (DoD, industry, academia)
- Incorporated comments from broader DoD review of draft
- Began May 1993; delivered Version 1.2 May 4, 1994

Part 2: Contents of the Roadmap

- SRI Context of Roadmap
- Purpose
- Scope
- Cause-Effect Framework
- Process Framework
- Enablers of Application and Domain Engineering
- Summary of Enabling Technologies
- Basis for Assessing Technology Maturation
- Maturation Status
- Near-term Recommendations
- Recommendation Considerations

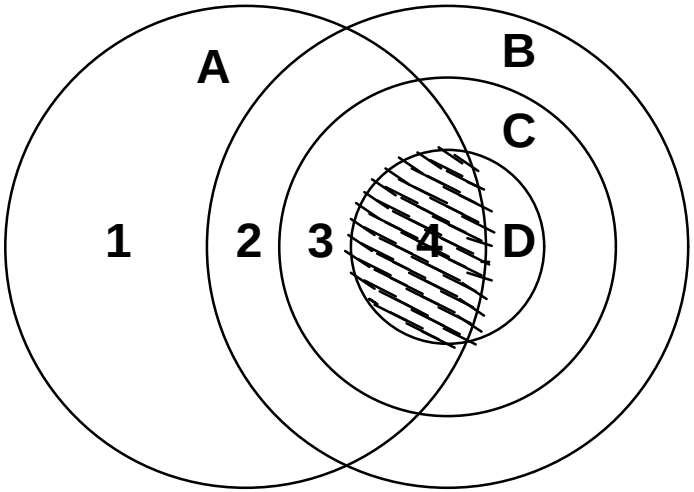
SRI Context of Roadmap



Purpose

- **Purposes of this DoD Software Reuse Initiative Technology Roadmap are as follows:**
  - **Identify those technologies which are critical in implementing software reuse within DoD**
  - **Assess the current and projected maturation of the critical technologies**
  - **Recommend investment in specific critical technologies to accelerate their maturation sufficiently to support the DoD software reuse goals in a timely manner**

Scope



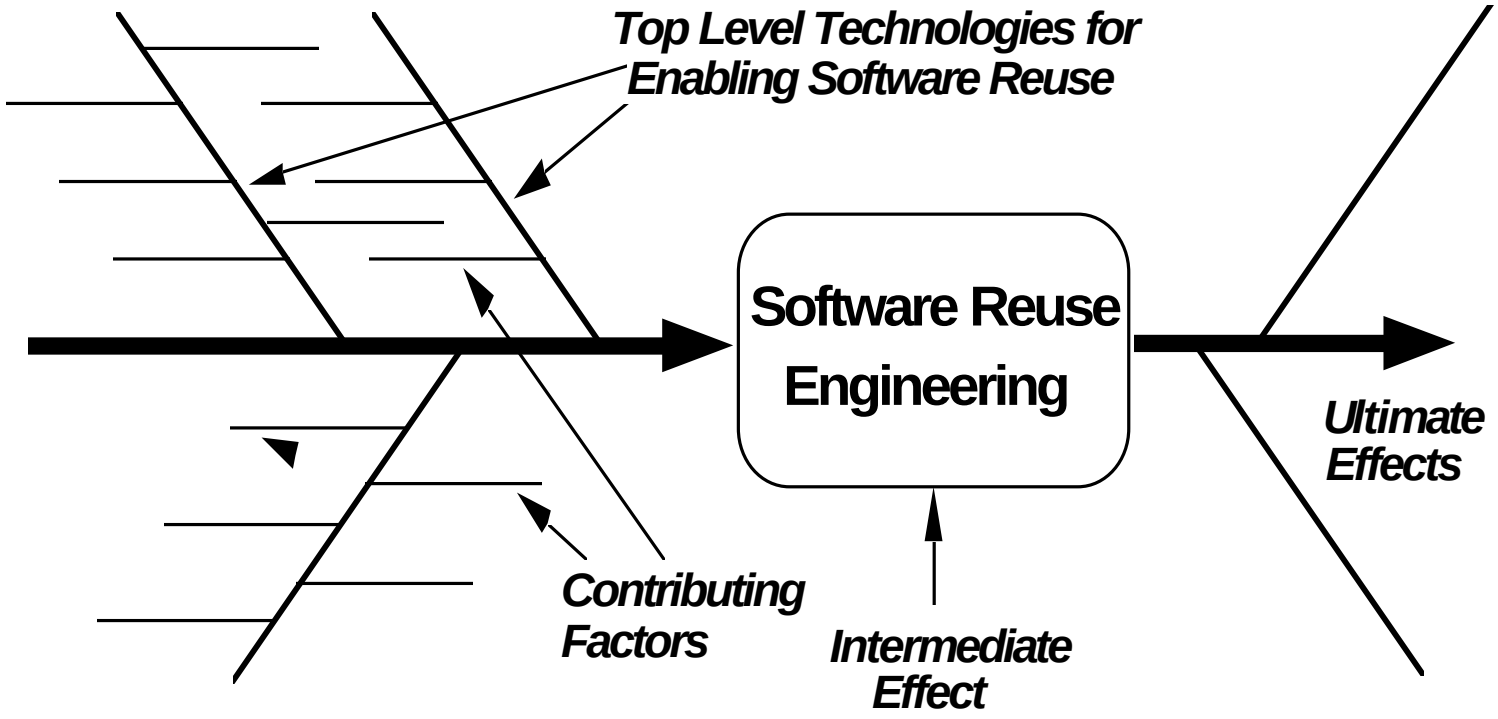
Key: A,B,C,D = full circles
1,2,3,4 = within A only

A = What DoD can do to make reuse successful
B = Technology essential to reuse goals
C = Technology essential to reuse goals and not yet mature
D = Technology essential to reuse goals and not yet mature and won't be mature soon enough

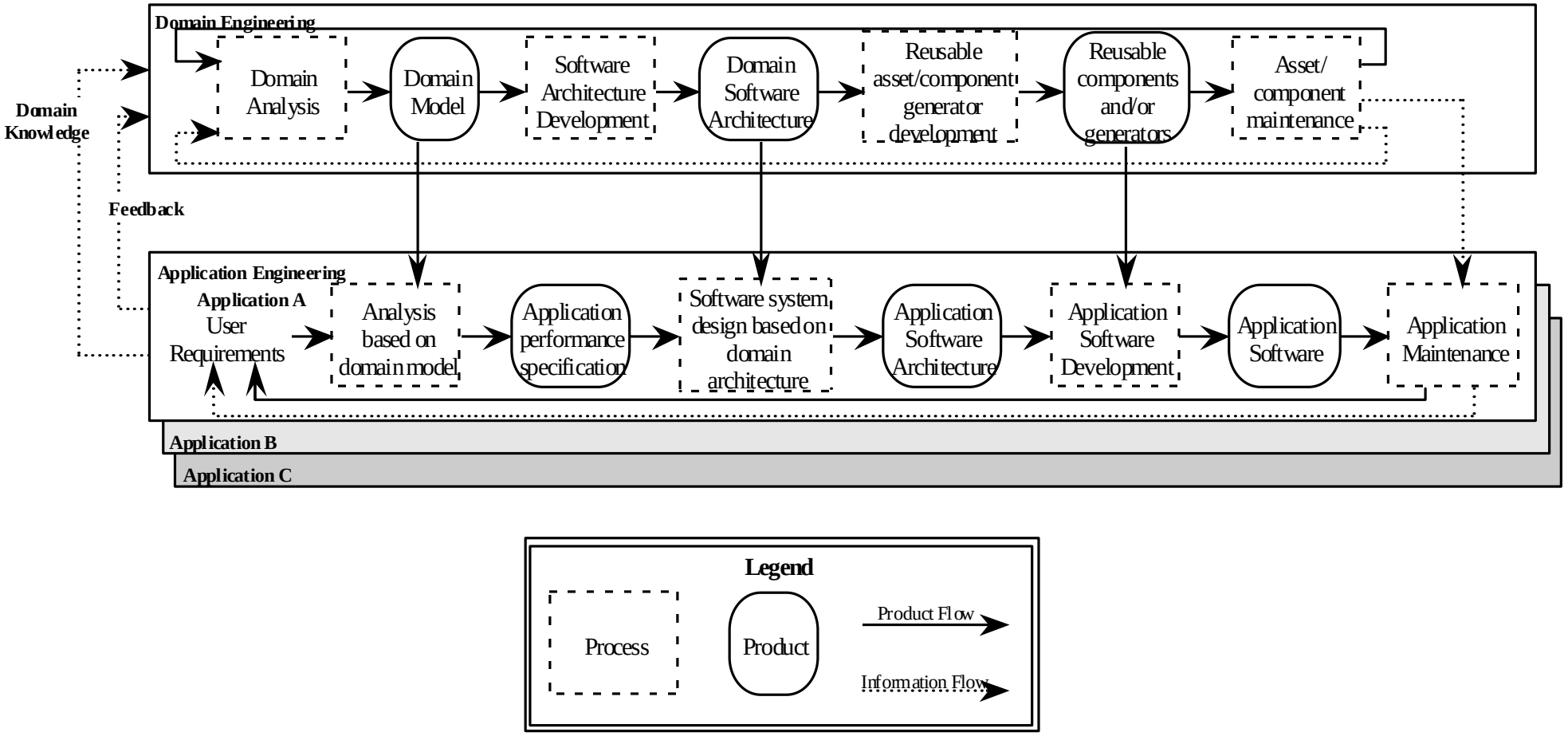
1 = What DoD can achieve from actions that are not technology constrained
2 = What DoD can achieve using existing technology
3 = What DoD can achieve using emerging technology
4 = What DoD can achieve by investing in technology to speed it up

Areas to invest in: 4 (lined)
Critical technologies: 3 + 4 (+ 2 as necessary for background)
Scope of roadmap document: 3 + 4 (+ 2 as necessary for background)

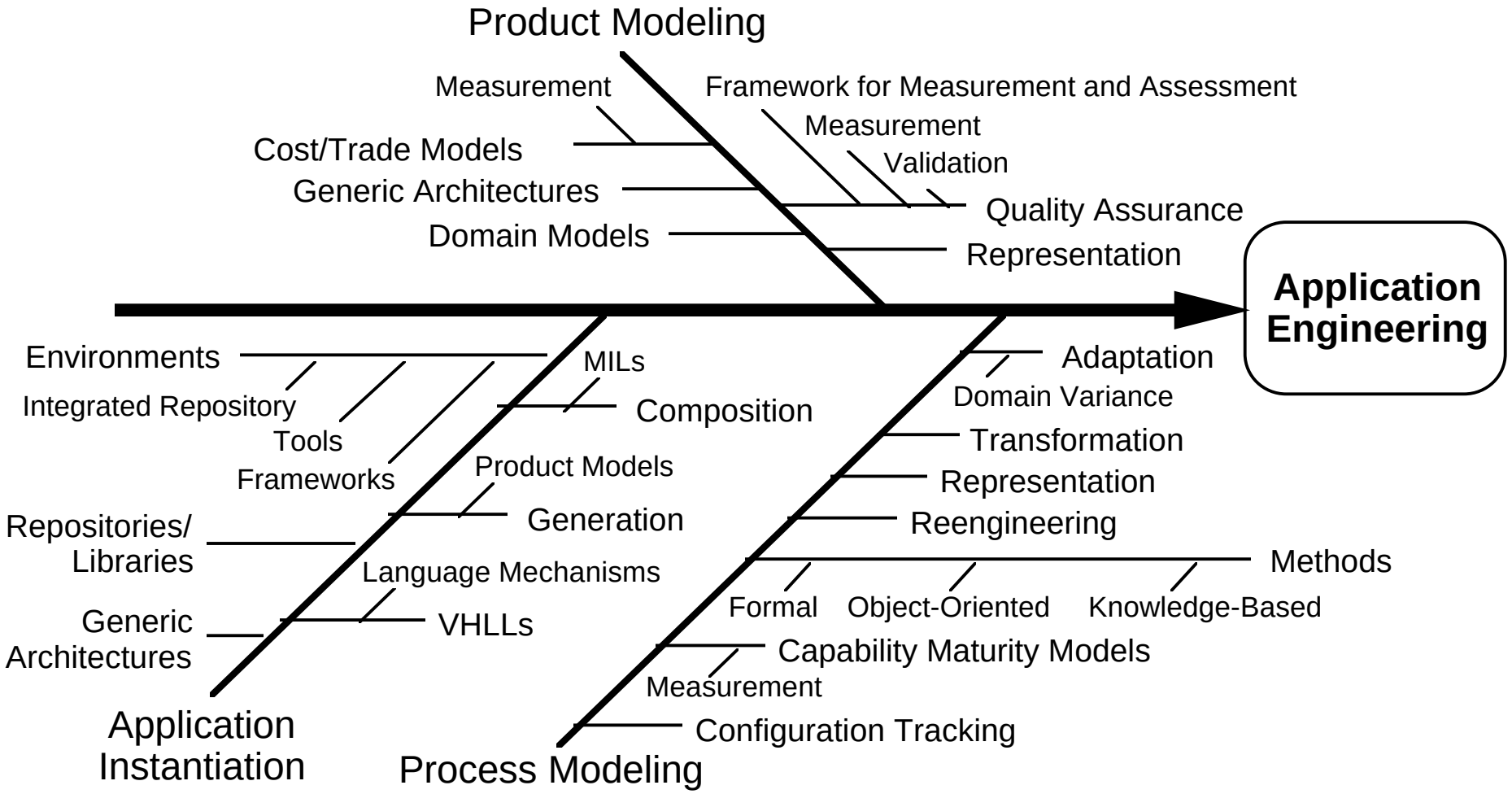
Cause-Effect Framework



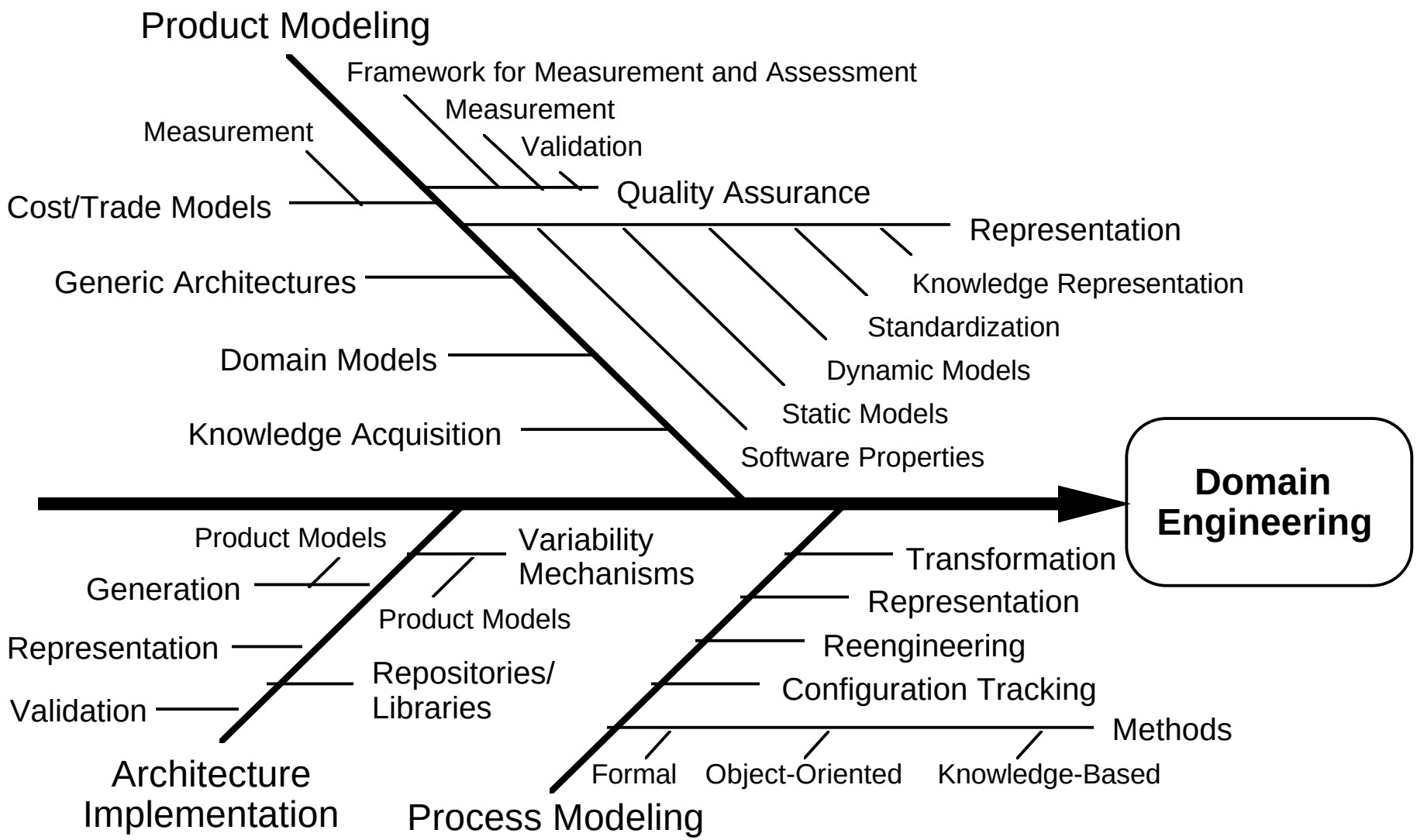
Process Framework



Enablers of Application Engineering



Enablers of Domain Engineering



Summary of Enabling Technologies (1 of 3)

Technology	Practice	Research	Gaps
Representation	Product template definitions Rigorous approach to defining and modeling processes and their relationships to evolving products not defined	Proposals for uniform data models Standard for software life cycle development	Formal machine-processable semantics Variable granularity Traceability evolution Extensibility of structured methods to the object paradigm Black box specifications Multilevel security and integrity
Process Modeling	Inadequate definition of connection of domain models and associated solution architectures Insufficient life cycle integration Insufficient standardization of semantic modeling within software engineering frameworks Insufficient tools and user interfaces	Semantic models for critical parts of the process Enactment mechanisms	Representation and enactment of those representations Adaptation and reusability of process definitions
Composition and Generation	Reuse of small, self-contained building blocks from existing systems Generative technologies Compositional technologies	Module Interconnection Languages Schema-Based Interconnection Languages Application of domain and generic architecture models Increased automated support for domain engineering Megaprogramming	Automated support for classification Schema-based interconnection languages Early life cycle composition strategies Domain-specific application generators Executable specification languages (VHLL and transformation)

Summary of Enabling Technologies (2 of 3)

Technology	Practice	Research	Gaps
Language Mechanisms	Program modularity is mature Object-orientation Exception handling mechanisms	Evolution of object-oriented systems Reuse improvements for concurrent and distributed systems	Reuse of hardware related features Levels of abstraction Semantic mappings
Libraries and Repositories	Local repositories for large systems by developing organization Database security and quality assessment for centralized reuse libraries	Sharing of assets across repositories – interoperability Indexing of reusable components	Interoperability, interface design, distributed databases, database security, quality assurance, change management Automated indexing, representations of library connections, experimentation
Methods	Information obtained from knowledge acquisition is represented in a knowledge base Object-oriented analysis	More automation in system development using knowledge obtained from domain engineering Use of formal methods with structured development methods – hybrid approach	Integration of methods (knowledge - based, object-oriented, formal) Integration with system life cycle Scale-up to support large systems

Summary of Enabling Technologies (3 of 3)

Technology	Practice	Research	Gaps
Software Engineering Environments	Life cycle assets are dependent on the environment in which they were created Lack of support for incorporating reuse assets into the development process	Emergence of concurrent collaborative software engineering Standard interface sets and interface specifications Functionality of tools and applications decoupled from presentation capability	Integration of Software Engineering Environments (SEE) based on open standards Support for use of domain knowledge artifacts by domain specific SEEs Consistent representation of semantics
Reengineering	Browsing – hypertext, multiple view systems, program slicing Code redocumentation Software restructuring	Increased maintainability through code restructuring, translation and redocumentation Encapsulation of code from legacy systems Program decompositions and knowledge capture	Software maintainability measurement Cost benefit models Expert systems for reengineering Software process instrumentation for maintenance
Measurement and Assessment	Traceability-based impact analysis Size-based metrics (lines of code, function point analysis) No software reuse metrics standards	General model of software reuse measurement Models for economic payoff Software reuse maturity Software reuse levels Impact analysis support	Framework for measurement and assessment Standard definitions of types of software reuse Definition and validation of software reuse metrics Better data collection mechanisms

Basis for Assessing Technology Maturation

Milestone 0 is marked by the appearance of a key idea underlying the technology or a clear articulation of the problem.

Milestone 1 is marked by a clear definition of a solution approach via a seminal paper or demonstration system.

Milestone 2 is marked by availability of usable capabilities.

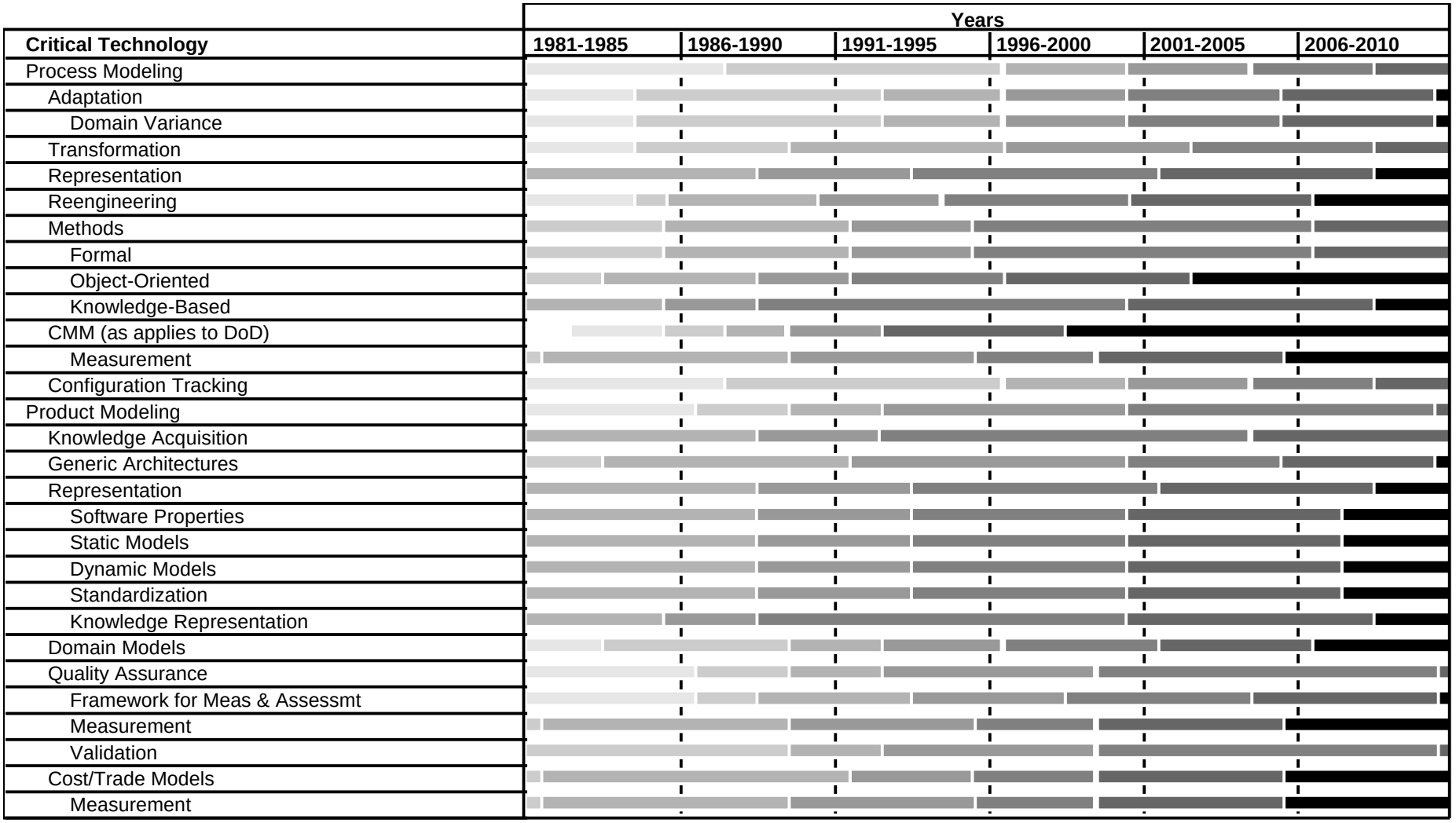
Milestone 3 is marked by a shift to usage outside of the development group.

Milestone 4 is marked by substantial evidence of value and applicability.

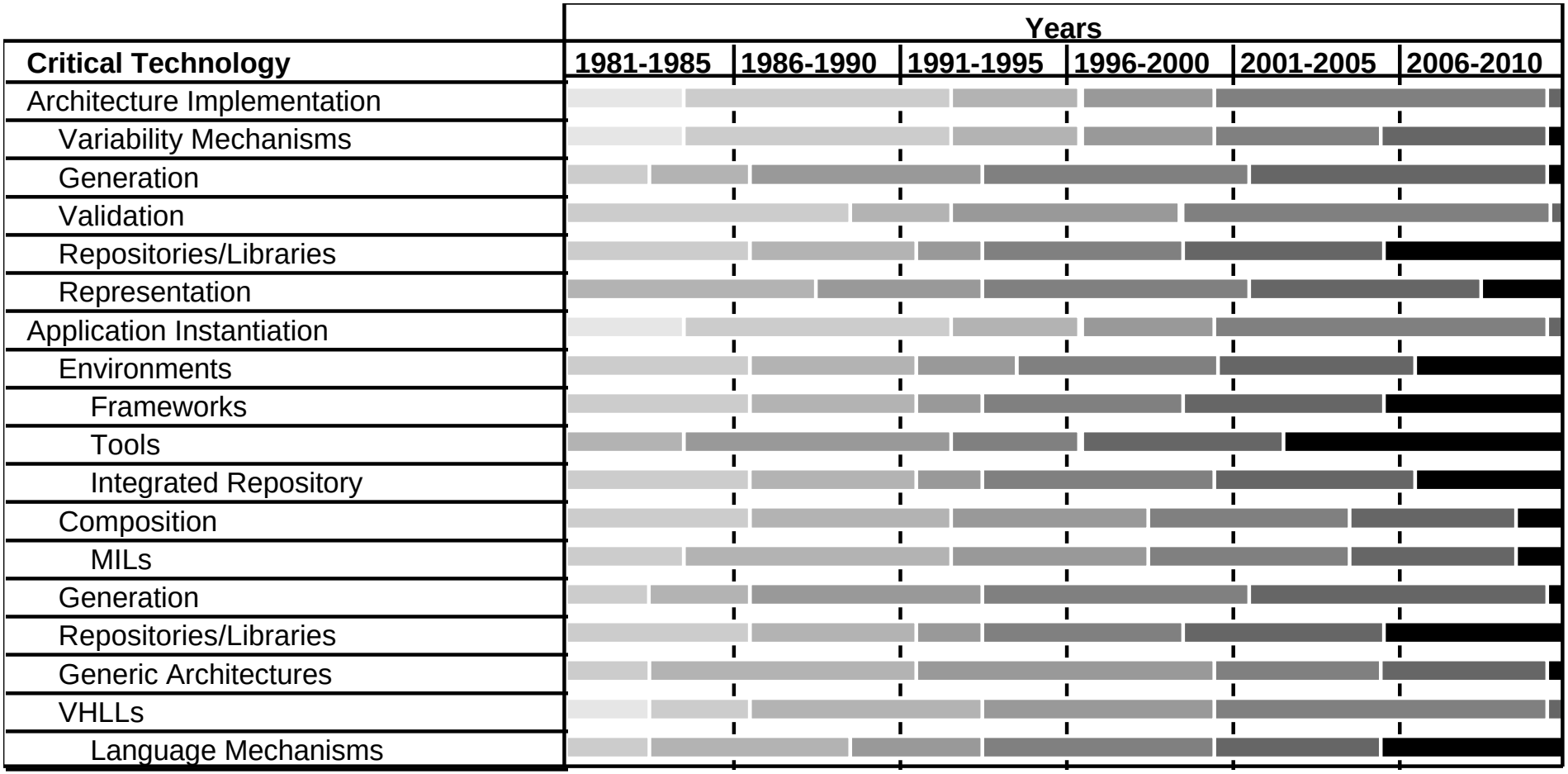
Milestone 4a is the point at which the technology has been propagated throughout 40% of the community.

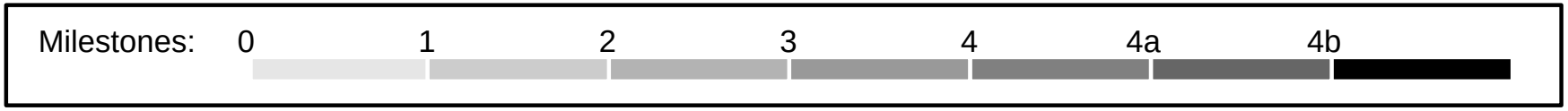
Milestone 4b is the point at which the technology has been propagated throughout 70% of the community.

Maturation Status (1 of 2)



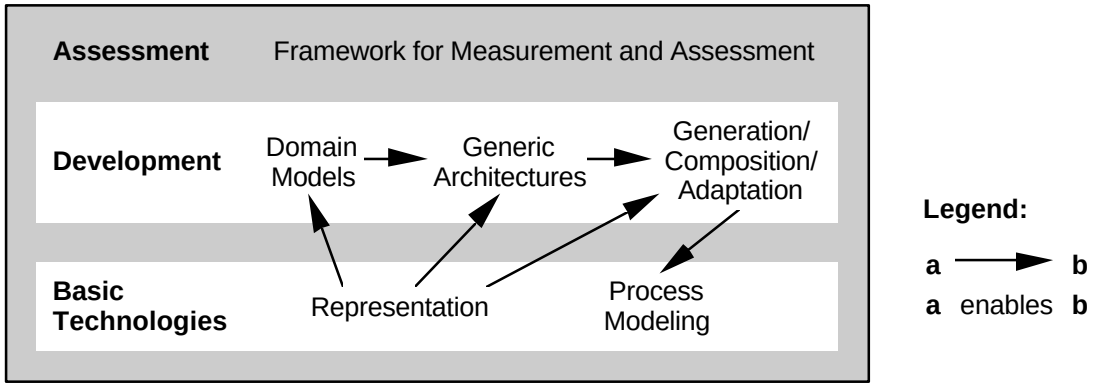
Maturation Status (2 of 2)





Near-term Recommendations

- **Near-term DoD investment is recommended in these technologies:**
  - **Framework for Measurement and Assessment**
  - **Representation**
  - **Domain Models**
  - **Generic Architectures**
  - **Generation Mechamisms**
  - **Composition**
  - **Adaptation Mechanisms**
  - **Process Modeling**



Recommendation Considerations (1 of 3)

- **Framework for Measurement and Assessment**
  - **Develop in concert with other technologies**
  - **Experimentation and data collection**
  - **Areas of emphasis: Standard reuse terminology, validation of reuse metrics, data collection mechanisms, costing models**
- **Representation**
  - **Machine-processable semantics (for systems and users, structure, function, etc.)**
  - **Variable granularity of representations**
  - **Standardize existing representations, not necessarily create new ones**

Recommendation Considerations (2 of 3)

- **Domain Models**
  - Select domains (mature, well-understood)
  - Include multiple views (including static, dynamic)
  - Relies on software representation, knowledge acquisition
- **Generic Architectures**
  - Specify domain architectures in standard or common form
  - Use architecture specification and definition languages
  - Tap extensive knowledge about existing architectures
- **Generation Mechanisms**
  - Develop domain-specific generators
  - Use experienced engineers in mature domains
  - Computer-assisted initially, rather than fully automated

Recommendation Considerations (3 of 3)

- **Composition**
  - **Represent interfaces and relations among components at varying levels of granularity, up to subsystems**
  - **Base on architectural modeling foundation**
  - **Explore hybrid of composition and generation**
- **Adaptation Mechanisms**
  - **Determine how to represent variances in a domain, including design and implementation alternatives**
  - **Explore multiple approaches: selection of alternatives; parameterization; class specializations; templates**
- **Process Modeling**
  - **Address resources, roles, responsibilities, activities, and products**
  - **Define and reuse process definitions for domain engineering and application engineering**

Part 3: Plans

- Reviewer Suggestions
- Plans for Continuation

Reviewer Suggestions

- **Response to Coverage**
  - **Generally positive response**
  - **Most thought technical issues were adequately addressed**
- **Suggestions for Additional Coverage**
  - **Refine recommendations, make more specific/actionable**
  - **Many requests to address technology transfer**
  - **Address other management concerns (cost, risks, obstacles, payoff)**

Plans for Continuation

- General approach:
  - Work the reuse issues closely with DoD community (DoD Software RESC, Roadmap review panel, services, R&D)
  - Focus in on small number of key investments (base on DoD needs and risks, refine and elaborate recommendations)
  - Tie in with budget planning (amount and timing of investment)
  - Plan, set up technology transfer (e.g., pilot projects, key areas)
- Potential tasks to implement approach (schedule to be defined)
  - Meet with DoD groups, conduct Roadmap workshops
  - Coordinate expert workshops to capture domain architectures
  - Analyze reuse cost avoidance and return on investment
  - Refine and prioritize reuse investment recommendations
  - Develop technology transfer/insertion plan