



“They Keep Moving the Cheese”

A Framework for Evolutionary Acquisition of Large Software Intensive Systems

Cecilia Albert

Lisa Brownsword

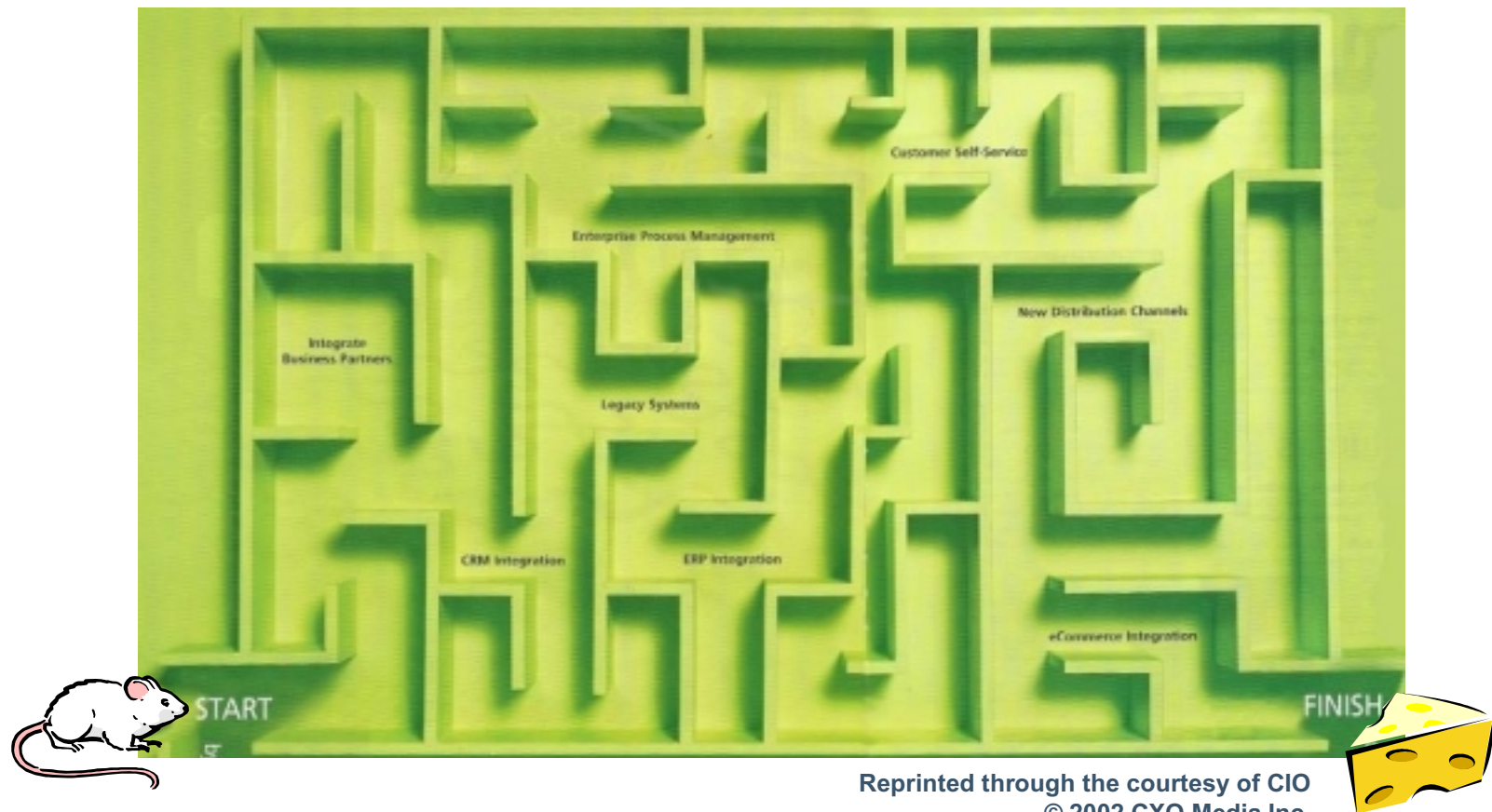
Sponsored by the U.S. Department of Defense
© 2003 by Carnegie Mellon University

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 26 JAN 2003		2. REPORT TYPE		3. DATES COVERED 00-00-2003 to 00-00-2003	
4. TITLE AND SUBTITLE 'They Keep Moving the Cheese' A Framework for Evolutionary Acquisition of Large Software Intensive Systems				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Carnegie Mellon University,Software Engineering Institute,Pittsburgh,PA,15213				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 24	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			



Carnegie Mellon
Software Engineering Institute

Who Moved My Cheese?



Reprinted through the courtesy of CIO
© 2002 CXO Media Inc.

A Story...

Program goal: provide a tool for strategic, operational, and tactical planners from all services and defense agencies to support joint and coalition engagements and peace keeping efforts

- Run on existing enterprise backbone (managed by another agency)
- Interface with multiple existing and developing systems
- Operate across multiple security levels

Program Start (late '90s)
• Automate manual process • Client-server architecture • Support 2-3 day planning cycle
6 increments delivered across 6-7 years
• First release in 18-24 months

2003
• New planning processes • Web-based architecture • Dynamic planning cycles • Collaborative planning
• Increment 1 is obsolete • Struggling to build/field increment 2 • Users have built “interim” solutions • Future is uncertain



Size Matters!

Project Size	People	Time (mos)	Success Rate
< \$750K	6	6	55%
\$750K-\$1.5M	12	9	33%
\$1.5M-\$3M	25	12	25%
\$3M-\$6M	40	18	15%
\$6M-\$10M	+250	+24	8%
>\$10M	+500	+36	0%

Source: The Standish Group, 1999



Definitions

A software-intensive system is one that

- Relies on software to provide core/priority mission function(s)

A large software-intensive system is one whose software

- Takes longer than 6 months to implement
- Takes more than 6 people to implement
- Takes more than \$750,000 to implement

and/or

- Is comprised of multiple interrelated systems or independently developed components implemented in software (system of systems, family of systems, etc)



Carnegie Mellon
Software Engineering Institute

Outline

Change Happens

Adapting to Change

Be Ready to Change Quickly



Change Happens

Large software-intensive systems change at a rate faster than the full system capability can be implemented – and they change during development and operation

Sources of change

- Enterprise priorities shift
- Business or operational needs change
- New technologies introduce new opportunities
- COTS products add and delete key features
- Participants rotate
- ...



Adapt to Change

DoD 5000* provides mechanisms for coping with change

Evolutionary Acquisition

Delivers capability in increments, recognizing, up front, the need for future capability improvements

- Success of the strategy depends on the consistent and continuous definition of requirements and maturation of technologies that lead to disciplined development and production of systems that provide increasing capability towards a material concept.

Spiral Development

A desired capability is identified but the end-state requirements are not known at program initiation

- Those requirements are refined through demonstration and risk management; there is continuous user feedback; and each increment provides the user the best possible capability. The requirements for future increments depend on feedback from users and technology maturation.

* The Operation of the Defense Acquisition System, 30 Oct 02



Lessons Learned



Going after “low hanging fruit” in the absence of an overarching architecture and coherent plan results in incompatible and stove-piped solutions



System requirements defined without sufficient insight into what can be realistically built, results in systems that cannot be built



There are no “silver bullets” that avoid disciplined system and software engineering (doing the right engineering correctly)



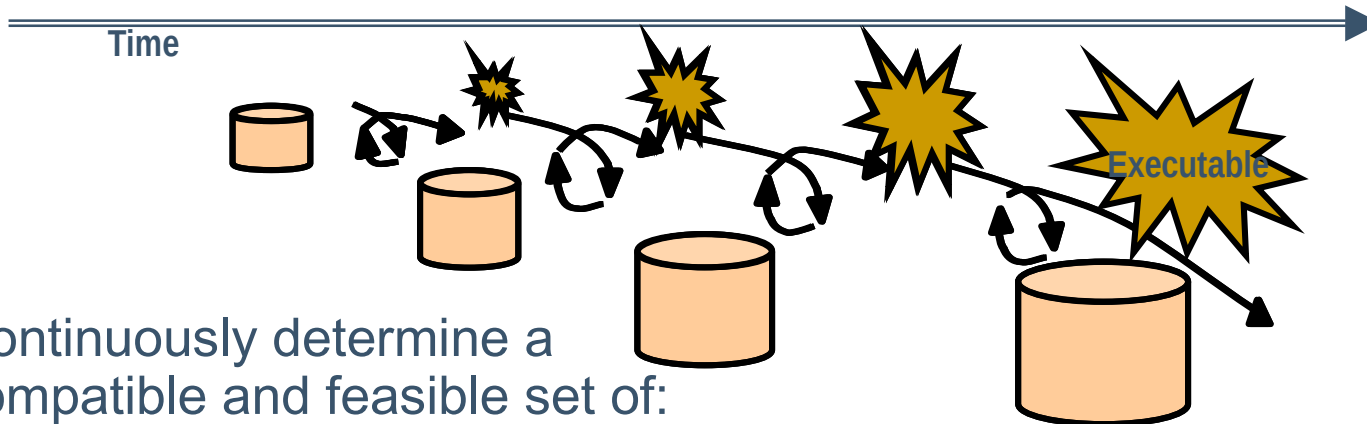
Be Ready To Change Quickly

Consciously apply spiral development practices at 2 (or more) discrete levels – with continuous interaction between the levels

- Program or system level
 - Evolve definition and implementation plan for system end-state
 - Define and spawn increments of useful capability that will build to full system functionality and performance
- Project or increment level
 - Define and implement plan for delivering the defined increment in the context of the system end-state



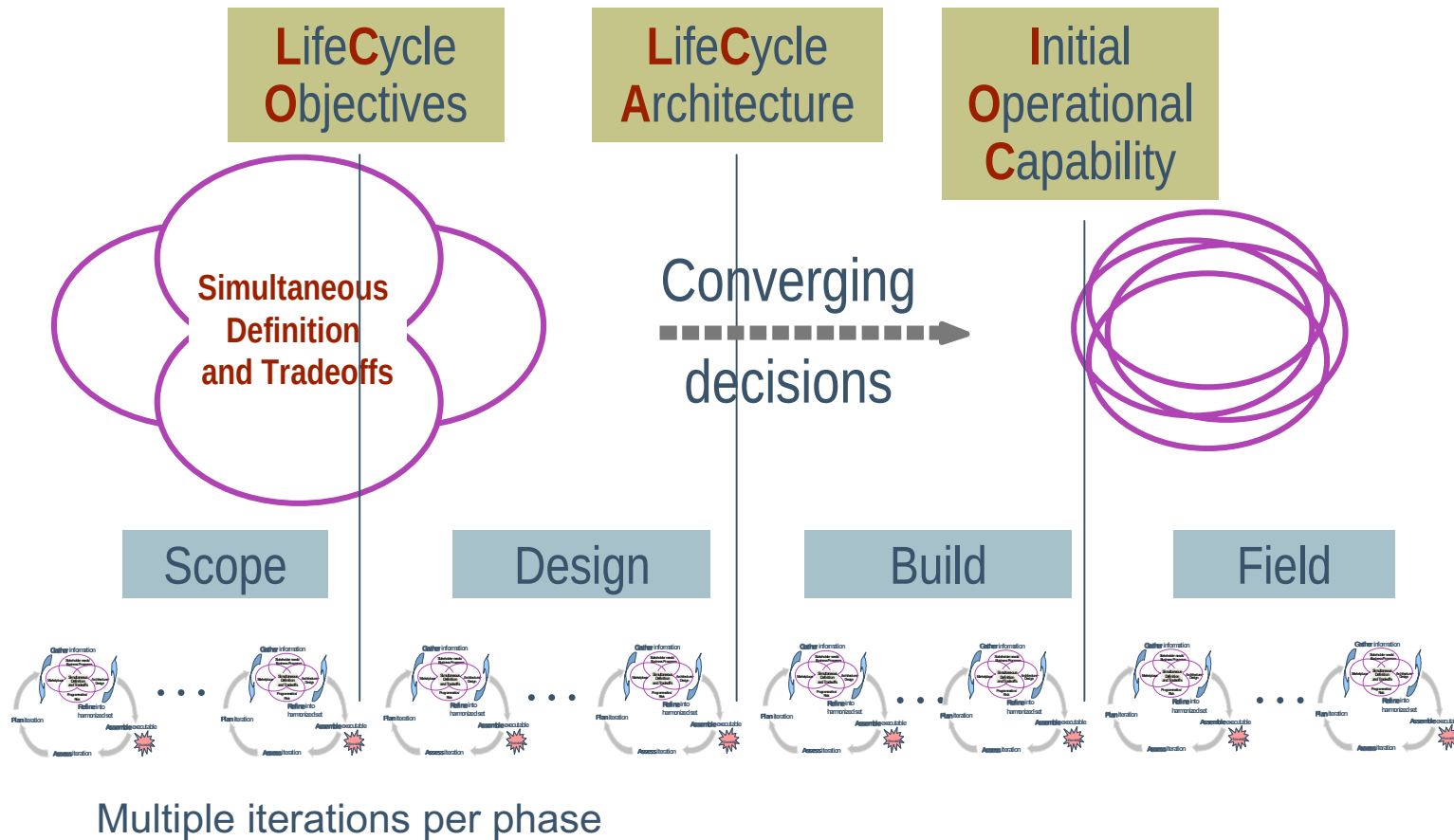
Disciplined Spiral Development



- Continuously determine a compatible and feasible set of:
business processes, requirements, plans, architecture, COTS products and other components
- Enterprise business objectives drive solution definition
- Risk considerations drive degree of detail
- Executable representations demonstrate current understanding and agreements

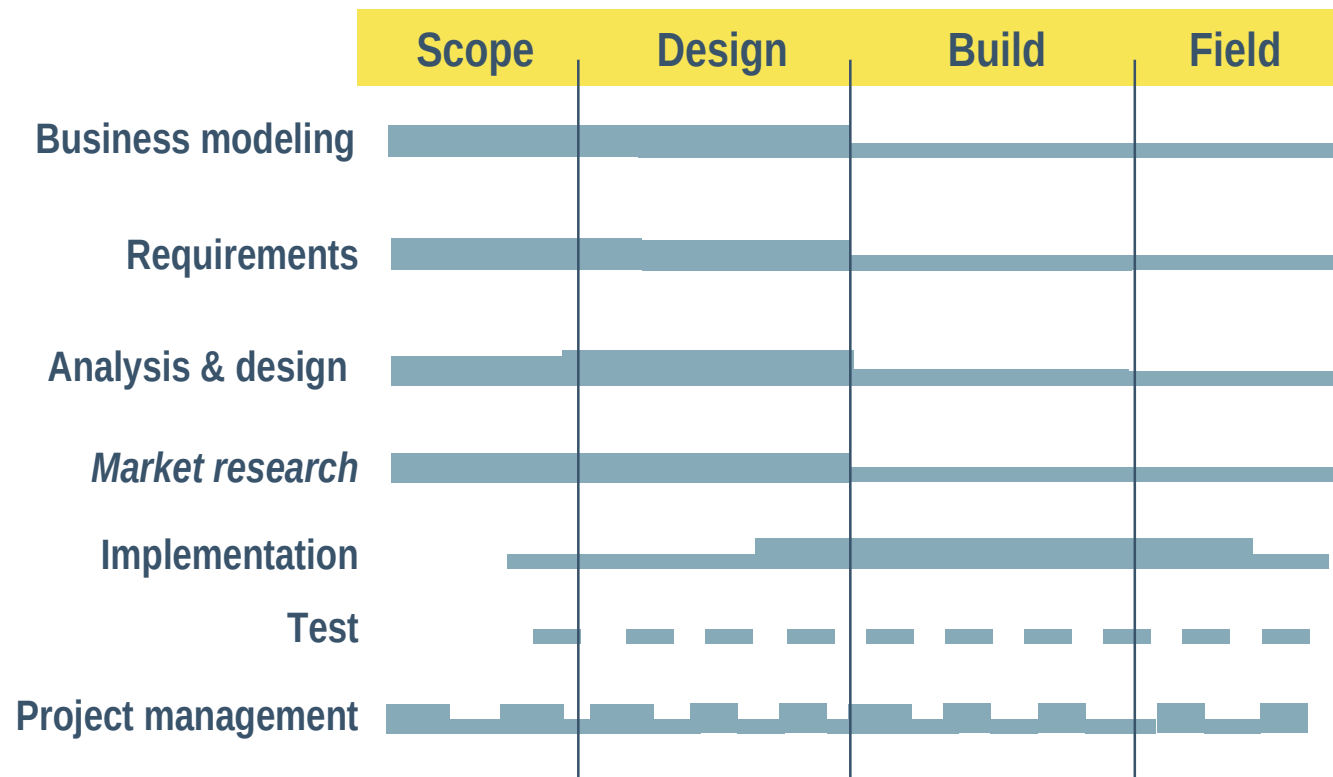
Spiral development facilitates evolving a viable solution –
at both system and increment levels

Phases Bounded by Anchor Points





Disciplines* Extend Across Phases



*adapted from Kruchten; shows partial set of disciplines



Carnegie Mellon
Software Engineering Institute

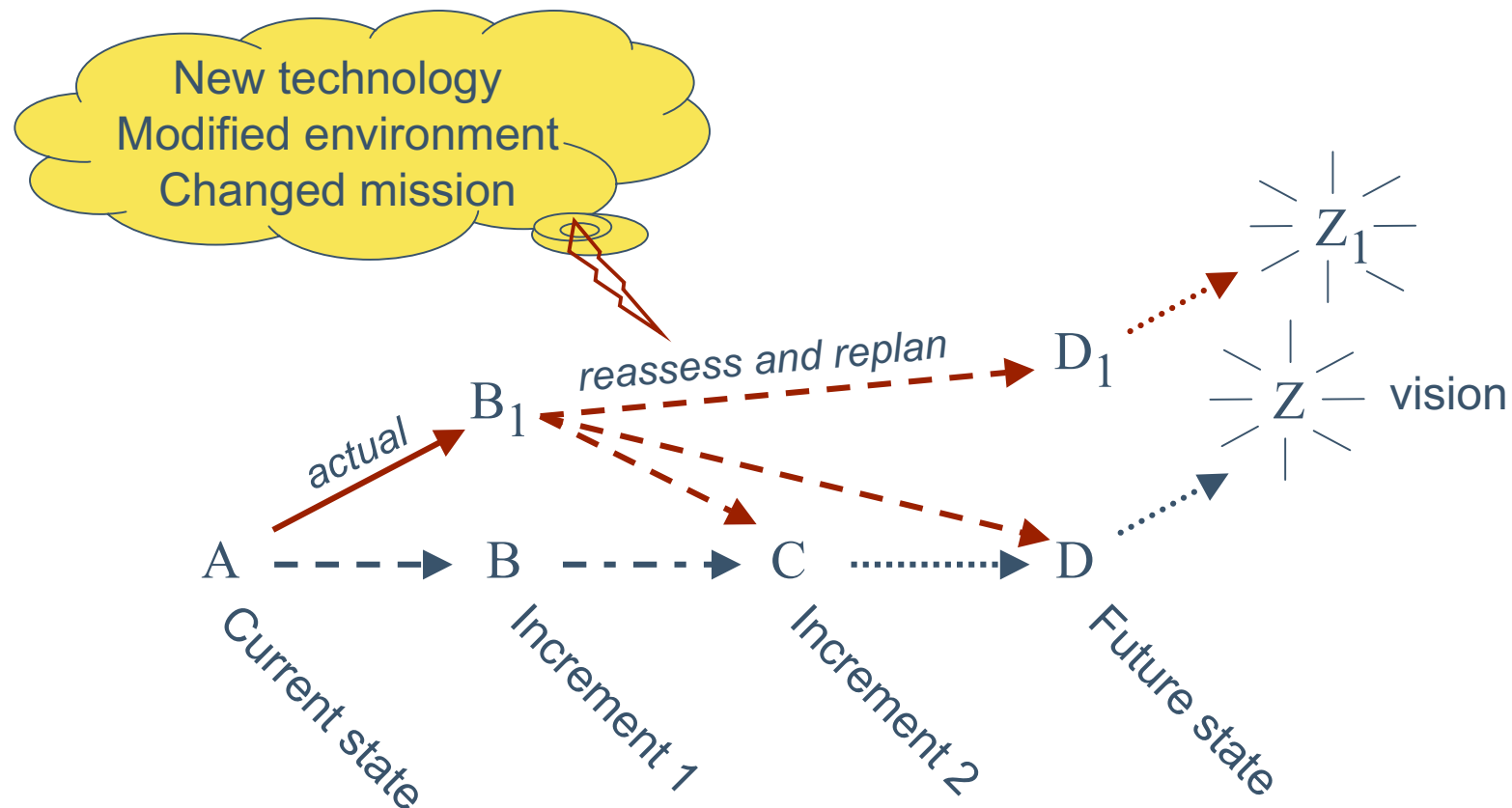
Keep a Long View in Systems Planning



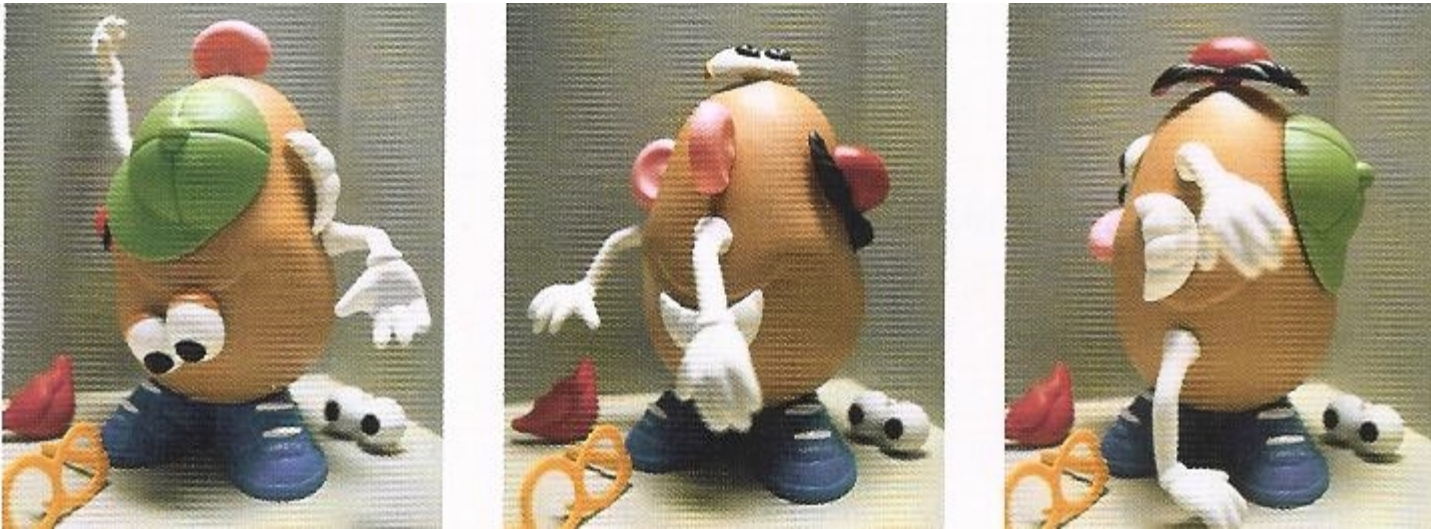
Reprinted through the courtesy of CIO
© 2002 CXO Media Inc.



Evolving System Definition



Take a Short View on Increment Planning



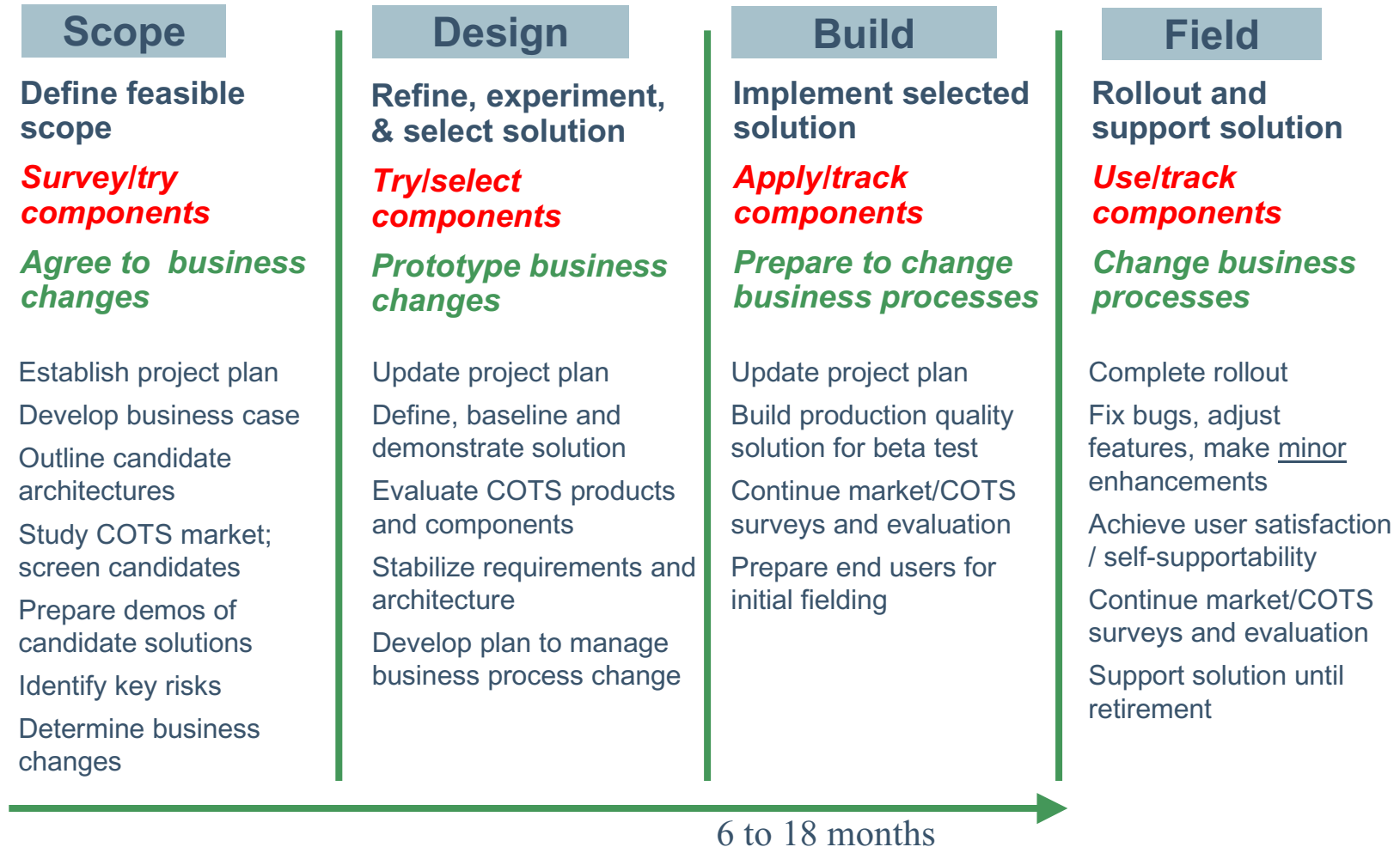
Reprinted through the courtesy of CIO
© 2002 CXO Media Inc.

Allows a stable development environment – if a short timeframe (6-18 months)

Allows focused discovery, experimenting, and learning on a manageable scale to find optimum way to understand and meet user needs



Increment Activity Mapping



Leverage Feedback between Long- and Short-Term

Maintain long-term strategy (system level) aligned with enterprise improvement

Make short-term implementation decisions (increment level) aligned with long-term strategy

Use knowledge gained in short-term increments to evolve long-term strategy

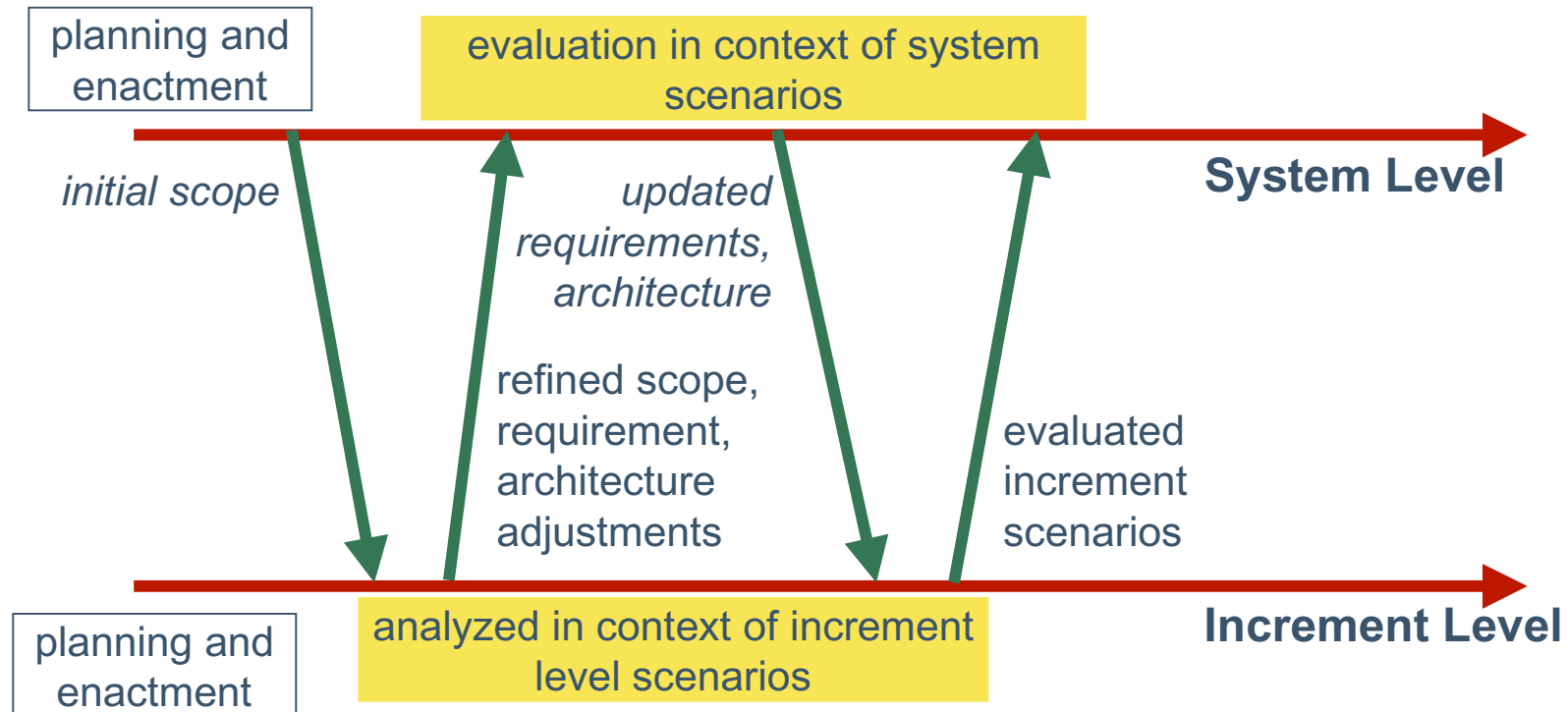


Reprinted through the courtesy of CIO
© 2002 CXO Media Inc.

Anticipate continuous change



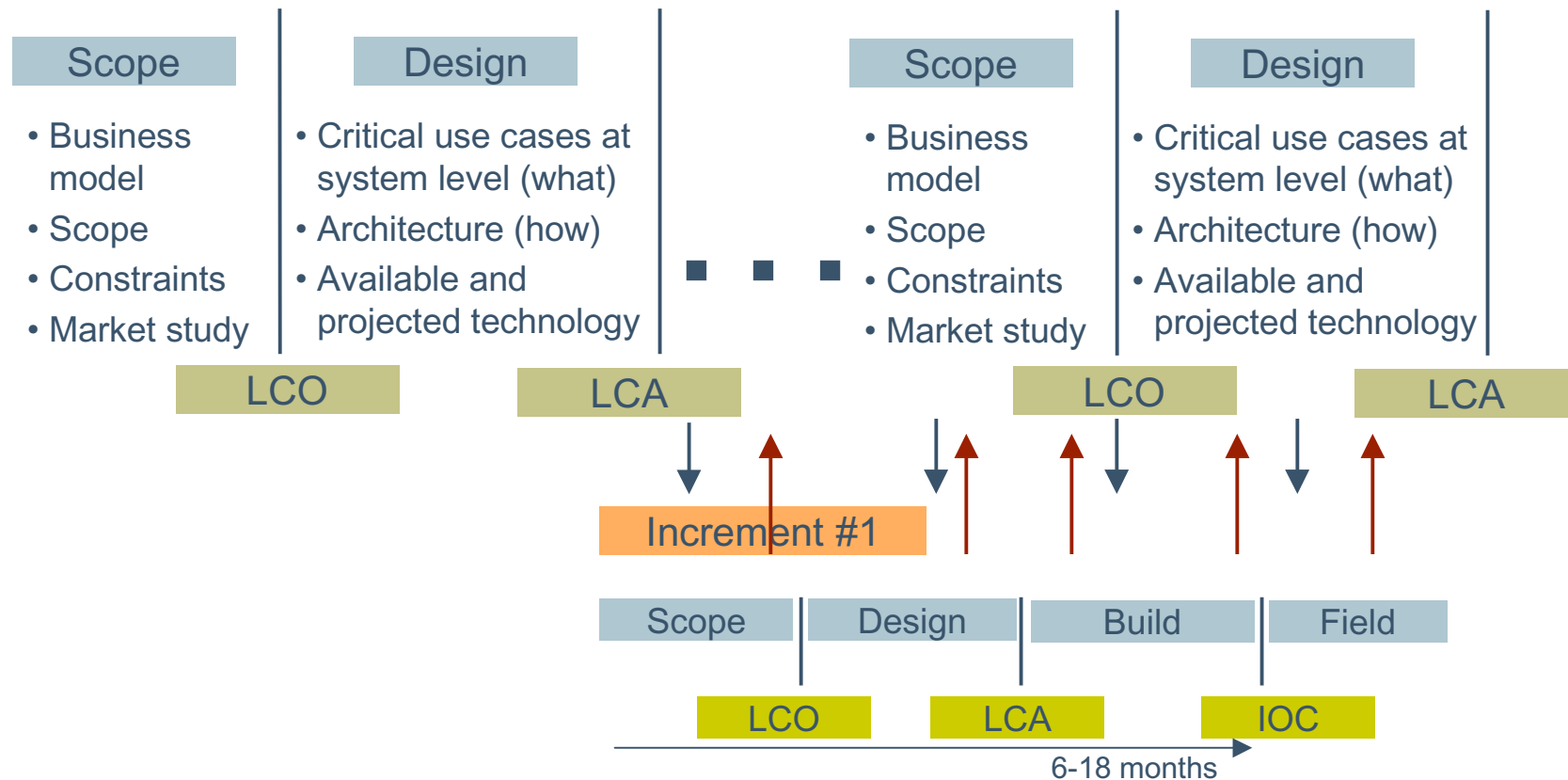
Plan and Manage *Efficient* Feedback



Decisions take place simultaneously at both levels –
one informs the other

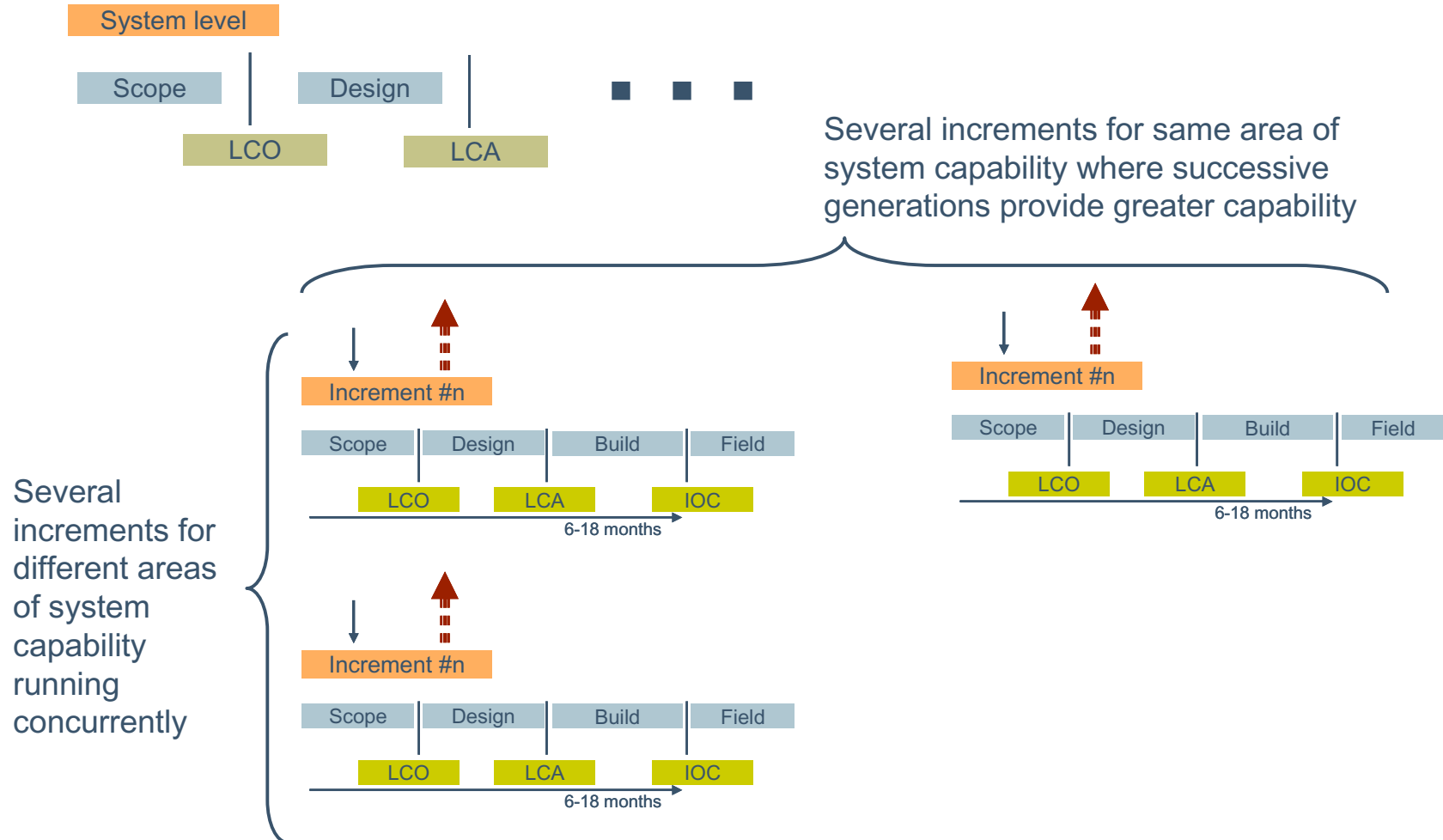
Managing Continuous Evolution

System level





Scenarios of Multiple Increments



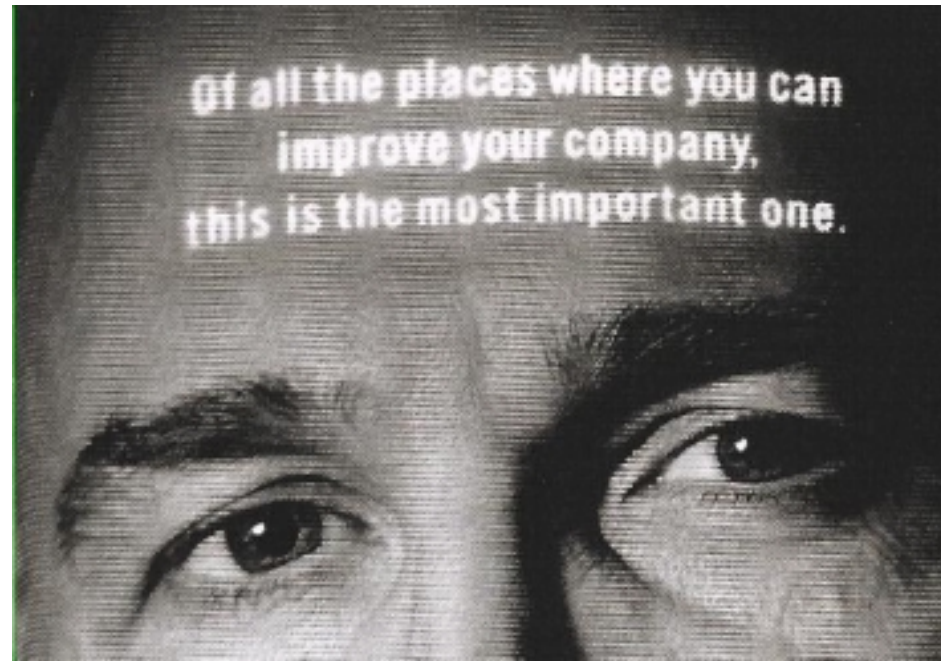
The Handwriting on the Wall

Change Happens

Adapt To Change Quickly

- Anticipate Change
- Monitor Change
- Change
- Enjoy Change!

Be Ready To Change Quickly
And Enjoy It Again



Reprinted through the courtesy of CIO
© 2002 CXO Media Inc.



Carnegie Mellon
Software Engineering Institute

Contact Information

Lisa Brownsword
llb@sei.cmu.edu

Cecilia Albert
cca@sei.cmu.edu



Lisa Brownsword is a senior member of the technical staff in the Commercial-off-the-shelf- (COTS)-Based Systems (CBS) Initiative at the Software Engineering Institute (SEI). Before joining the SEI, Lisa was on staff at Computer Sciences Corporation in support of NASA/Goddard's Software Engineering Lab. Prior to that, she was employed at Rational Software Corporation providing consulting to managers and technical practitioners in the use of and transition to software engineering practices, including architecture-centered development, product lines, object technology, Ada, and CASE.

Cecilia Albert is a senior member of the technical staff in the Commercial-off-the-shelf- (COTS)-Based Systems (CBS) Initiative at the Software Engineering Institute (SEI). Before joining the SEI, Cecilia was in the Air Force where she served in a variety of information technologies related positions including: developing major software programs for simulation, command and control, and mission processing of national satellite systems; teaching acquisition and leading an industry study on telecommunications and information systems at the Industrial College of the Armed Forces; and managing the archive and dissemination programs at the National Imagery and Mapping Agency.