The Role of System Representation and Collaboration in Design: Why Are Some Programs More Adaptable than Others?

Presented By
Lt. Col. Rob Dare
ESC/AE
March 26, 2003

Agenda

- Why are Some Programs More Adaptable than Others?
- Research Method and Definitions
- System Representations
- Collaborative Stakeholder Roles
- Recommendations for practitioners

Why are Some Programs More Adaptable than Others?

- Adaptability: stakeholder-driven changes to existing requirements baseline or to contractor design choices
  - Add value
  - Work within programmatic constraints
- Two complementary aspects support adaptability:
  - Use of system representations (e.g. prototypes, interim software releases) - a mechanism for knowledge sharing
  - Specific roles of stakeholders in collaborative interaction - provide flexibility and structure

Research Method

- Case studies of 8 Air Force Command & Control (C2) programs with incremental delivery strategies
- Data collection
  - Interviews with SPO, user, and contractor - patterns of collaboration
  - Program documentation review – level of adaptability
- Research questions
  - How does a system representation enhance adaptability?
  - What characteristics make system representations effective at promoting adaptability?
  - What are the roles of stakeholders in facilitating program adaptability?

Research Definitions

- System representation (SR)
  - A representation of the system design as it is envisioned at a given point in the design process
- Collaboration
  - Sharing knowledge between stakeholders about the system during design with the intent to identify and disposition emergent issues and opportunities

System Representations (SR) Finding 1

- Higher degrees of knowledge sharing (depth and frequency of SR usage) corresponded to higher levels of adaptability
- An SR facilitated adaptability when used to:
  - Identify potential changes
  - Evaluate potential changes (perform “what if’s”)

Adaptive programs used a system representation to share knowledge between stakeholders

System Representations (SR) Finding 2

- Higher SR fidelity enhanced adaptability
- Two elements of SR fidelity
  - Level of detail (system, sub-system or minimal)
  - Coverage of stakeholder emphasis areas (e.g. technical performance, user interface, reliability, etc.)

More adaptable programs had higher fidelity system representations (system-level detail, coverage of emphasis areas)

SR Observation – Effectiveness of SR vs. Analysis

SR \ Analysis	Low	High
Low	❖Interoperability	❖Life Cycle Cost
Medium		❖Reliability ❖Development Cost
High	❖User Interface ❖Technical Performance ❖Maintenance	

SR's were particularly effective at portraying user interface, technical performance, and maintenance aspects of designs.

Stakeholder Roles in Support of Adaptability

- Stakeholder actions must provide both flexibility and structure
  - Encourage innovation
  - Stay within cost and schedule constraints
- Roles (best practices) observed
  - SPO
  - User
  - Contractor

SPO Roles

- Flexibility
  - Encourage and facilitate user engagement
  - Facilitate contractor evaluation of changes (studies clause, resources, etc.)
- Structure
  - Manage user expectations
  - Evaluate risks of changes

SPO Roles for Adaptability

Activity	SPO Role	High Performance (Examples from cases)	Low Performance (Examples from cases)
Demonstrate partial design	**Encourage and facilitate user engagement during the design phase**	Emphasized importance of user involvement from the start of design; user shown that inputs made a difference	Discouraged user from participation during design phase (SPO felt user role was limited to defining requirements)
	Manage user expectations	Briefed user on current and future SR capabilities in preparation for user interaction with SR	Decided not to share SR with users until substantial functionality was available due to concern that user would criticize program
Identify potential design or requirements changes	Provide design feedback: system considerations and "ilities" (reliability, maintainability, interoperability…)	Tracked system's ability to flow data to meet all user needs; Analyzed technical risk areas (e.g. antenna interference and COTS performance in operating environment) to ensure system reliability	Allowed contractor to develop and demonstrate system in non-integrated segments; Lacked process for tracking related systems that were in development to spot future interoperability issues
Evaluate potential changes	**Facilitate contractor evaluation**	Established contract provisions for studies; Encouraged contractor "what if" exercises	No resources planned for contractor "what if"" exercises
	Evaluate risks	Assessed realism of cost estimates; Weighed added risk to meeting constraints	Underestimated resources required to implement changes
Incorporate changes into baseline	Issue rapid approval	Added work scope and funds to contract quickly	May have delayed timely implementation due to under staffing (not conclusive)

Notes: Roles in **bold with gray backgrounds** are best practices supported by case study data
Non-bolded roles are either common practice (well understood) or are only partially supported by data

User Roles

- Flexibility
  - Provide design feedback: operational perspective (how the system will be used)
- Structure
  - Coordinate field participation (user HQ)
  - Define priorities (importance of potential changes vs. existing requirements)

User Roles for Adaptability

Activity	User Roles	High Performance (Examples from cases)	Low Performance (Examples from cases)
Demonstrate partial design	**Coordinate field participation**	Designated user headquarters coordinated involvement of future field users who had experience operating existing systems	Did not participate in review of contractor's design, or had review of design by user headquarters personnel only
Identify potential design or requirements changes	**Provide design feedback: operational perspective (how system will be used)**	Commented on how operators would use the system - led to design improvements and changes in requirements	Had minimal or no user interaction after initial requirements definition
Evaluate potential changes	**Define priorities (importance of potential changes)**	Updated priority list weekly; leadership emphasized importance of establishing and communicating clear priorities	Had minimal or no user interaction after initial requirements definition
Incorporate changes into baseline	**N/A**	**N/A**	**N/A**

Notes: Roles in **bold with gray backgrounds** are best practices supported by case study data
Non-bolded roles are either common practice (well understood) or are only partially supported by data

Contractor Roles

- Flexibility
  - Create and share system representation
- Structure
  - Evaluate cost, benefit and best implementation approach for changes

Contractor Roles for Adaptability

Activity	Contractor Roles	High Performance (Examples from cases)	Low Performance (Examples from cases)
Demonstrate partial design	Create and share SR	See SR discussion and findings in Chapter 6 regarding knowledge sharing and SR fidelity	See SR discussion and findings in Chapter 6 regarding knowledge sharing and SR fidelity
Identify potential design or requirements changes	Select design options to meet requirements	Standard part of work effort – no appreciable differentiation between programs	Standard part of work effort – no appreciable differentiation between programs
Evaluate potential changes	Evaluate cost, benefit and best implementation approach	Assessed the benefit of changes and the work effort required for implementation; explored implementation options	Responded to user requests with minimal consideration of cost and schedule impacts
Incorporate changes into baseline	Update SR	Incorporated changes and provided iterative opportunities for SR review	Made limited (or no) iterations of SR available for government review
	Update program documentation	Ensured thorough documentation of all changes	Captured agreements inconsistently

Notes: Roles in **bold with gray backgrounds** are best practices supported by case study data
Non-bolded roles are either common practice (well understood) or are only partially supported by data

Recommendations for Adaptability

- Use system representations during design to enhance stakeholder knowledge sharing
  - Depth and frequency of SR interaction determine effectiveness
  - Use SR to identify and evaluate potential changes
- Desirable SR characteristics
  - System-level representations
  - Coverage of stakeholder emphasis areas
- SR portrays: user interface, technical performance and maintenance (analysis may help other emphasis areas)
- Stakeholders provide essential mix of flexibility and structure by following observed best practices

Back-Up Charts

Definitions* (Policy)

- Evolutionary Acquisition

  An acquisition strategy that defines, develops, produces or acquires, and fields **an initial hardware or software increment** (or block) of operational capability…capabilities can be provided in a shorter period of time, **followed by subsequent increments**…allowing for full and adaptable systems over time.

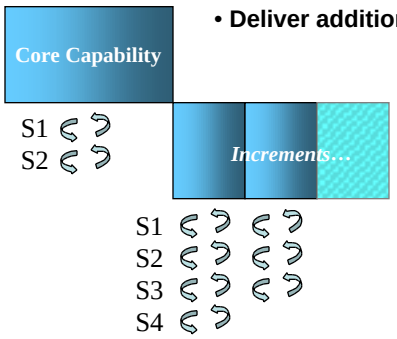
- Spiral Development

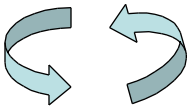
  An iterative process for developing a defined set of capabilities **within one increment**…provides the opportunity for **interaction between the user, tester and developer**…requirements are refined through experimentation and risk management, there is continuous feedback, and the user is provided the best possible capability within the increment.

*USD (AT&L) Aldridge memo, 12 Apr, 2002

Evolutionary Acquisition (EA) vs. Spiral Development (SD)

Evolutionary Acquisition is a Strategy
- Deliver core capability (rapidly)
- Deliver additional capability in increments





Spiral Development is a Process
- Within Increment
- Build a little, test a little
- Get interim customer feedback!
- Is not a "given" when using EA

Collaborative Acquisition Approach