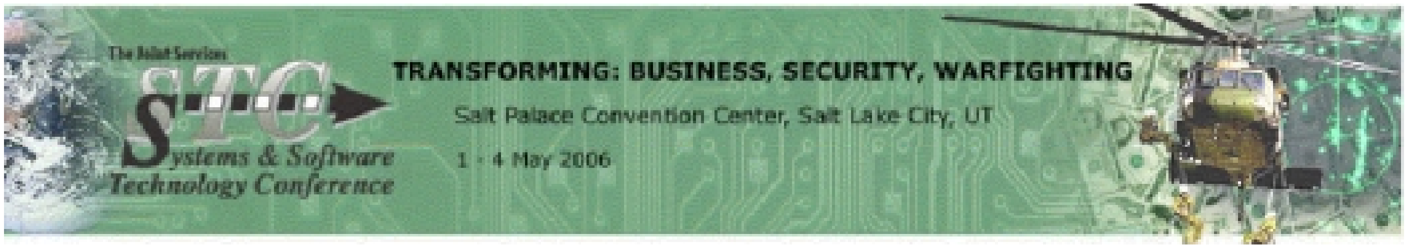
Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Integrating Warfighter-Driven System of System Innovation into the Acquisition Life Cycle

Ira Monarch and Jim Wessel


The Joint Services
STC
Systems & Software Technology Conference
TRANSFORMING: BUSINESS, SECURITY, WARFIGHTING
Salt Palace Convention Center, Salt Lake City, UT
1 - 4 May 2006

03/26/06

What’s Ahead

Users in Systems/Software Engineering

User-Driven Military Technology

User Innovation

Military Users and User Representatives

From Requirements to Capabilities: JCIDS

What JCIDS did not Consider: The DAPA Report

What the DAPA Report did not Consider: ASSIP’s CEF

Potential Key Focus Areas of the CEF

Conclusions

No Place for the User in SwE?

> **“The notion of ‘user’ cannot be precisely defined, and therefore has no place in Computer Science or Software Engineering.”**
>
> **– Edsger Dijkstra, ICSE 4, 1979**

But if this is the case, how does software engineering deal with

- user innovation
- user requirements (and now capabilities)
- user driven requirements change
- user resistance or rejection?

Military Technology as User Driven

The US military engages in large-scale client- and/or user-driven technology production that has infrastructure and processes for

- developing user requirements
- transforming user requirements into system requirements

However, Army warfighters are not seen as a source of technological innovation.

A new military perspective on requirements engineering should include both

- **user innovation**
- **disciplined user driven requirements generation and change.**

User Innovation

Innovating users include both firms and individuals (e.g., Boeing is a user-innovator of metal-forming machinery).

User-centered innovation offers advantages over manufacturer-centric innovation systems.

Users can

- develop exactly what they want, rather than relying on manufacturers to act as their (often very imperfect) agents.
- benefit from innovations developed and freely shared by others.

Empirical studies show that many users—from 10 percent to nearly 40 percent—develop or modify products.

User innovation research reported on this and the following slides is summarized in *Democratizing Innovation* by Eric von Hippel 2005 freely available online at http://web.mit.edu/evhippel/www/democ.htm

Reasons for Users to Innovate

Mass production strategies of “a few sizes fit all” leave many users dissatisfied.

For example, users of the security features of the Apache web server found that 19% actually innovated to tailor it.

The seller’s interest in making a profit is never the same as the user’s interest in overall solution quality.

And users know their context of use better than Sellers do.

Buy in – Users are more likely to use an innovation (more intuitive also) they have hand in defining.

Enjoyment – Volunteer contributors of code to widely used software are motivated by the joy they find in the work.

Problems with User Innovation

Lack of discipline and/or vision can bring risks

Local user innovations may inadvertently cause Enterprise level issues e.g., disparate databases, non-uniform messaging

Innovation may not be extendable to different contexts

- not scaleable for larger applications,
- lack global security features
- bring about survivability and maintenance risks

Disciplined processes must harmonize user innovation with total lifecycle acquisition

Enablers: Low Cost Resources and Toolkits

The cost of high-quality resources for design and prototyping is becoming very low.

Today, user firms and individuals have access to sophisticated

- online environments for design and collaboration
- programming tools for software
- CAD design tools for hardware and electronics.

Some firms have provided toolkits to users for custom design.

- Customers tend to prefer designing on their own with the aid of a toolkit
- Innovation via toolkits can be cheaper for users and suppliers, and suppliers can channel how and what users develop.

The custom semiconductor industry was an early adopter of toolkits that in 2003, amounted to ~$15 billion in production.

User Innovation Information Dissemination
Open Source Commons

In “open source” software development contributors routinely and systematically freely reveal code they develop at their expense:

- hiding innovation is unlikely to last very long
- others will improve or suggest improvements to the innovation, to mutual benefit
- enhancement of reputation from positive network effects due to increased diffusion of the innovation, especially when reveal first

For many problems, SW user-innovators now have a choice:

- proprietary, closed software provided by Microsoft and other firms
- open source software that they can legally download from the Internet and legally modify to serve their own specific needs.

User Innovation Communities

Users joining together in networks and communities provide:

- useful structures and tools for their interactions
- distribution support to
  - increase the speed and effectiveness with which innovations are developed, tested and diffused.
  - facilitate building larger systems from interlinkable modules created by community participants
- enterprise level review – can mitigate risk of local user innovations that may ‘break’ enterprise rules and constructs

Prime Example:

Free and open source software projects that are Internet-based innovation communities.

Problems with Open Source Communities

There have been some spectacular successes in open source software development and innovation including

- Linux
- Apache
- Emacs

However, many open source communities fail.

Moreover,

- security of the software developed is uncertain
- there is no guarantee that changes a given user makes will be embraced by the community.

In this latter sense OSS is similar to COTS software.

Nevertheless, it is clear that users can and do innovate and such creativity should be tapped.

The Military Takes a Different Tack:
Formal User Organization

The US military does not rely on the emergence of user communities to develop new innovations and requirements.

US military user organizations are formally organized from the top down and laterally.

In the Army, there is a network of user and requirements related organizations including

- Combat Commanders
- Combat Developers
- Material Developers
- Testers

for which formal relations are defined in a whole host of military documentation.

Warfighter & Developer

How then might the military tap the creativity of users (warfighters and their representatives) in

- building new innovations
- defining new requirements?

New forms of linkage and interaction are needed among

- the network of user and requirements related organizations
- developers (the Industrial Base)

to enable the Industrial Base to power the development of user innovations cost effectively.

From Requirements to Capabilities

Military and Army user organizations are currently implementing a capabilities approach to requirements development.

This is demanded by the need for

- **the creation of Joint Forces and the engineering of**
  - **Software Intensive complex System of Systems (SOS)**
  - **Network Centric Systems (NCS) they require.**

The DOD has responded to this potential radical shift by calling for the **Joint Capabilities Integration and Development System** that

- consists in a joint concepts-centric capabilities formation process
- is designed to enable joint forces to meet the full range of military operations and challenges.

The Difference Between:
Requirements and Capabilities

Capabilities are a heterogeneous mix of what the military calls

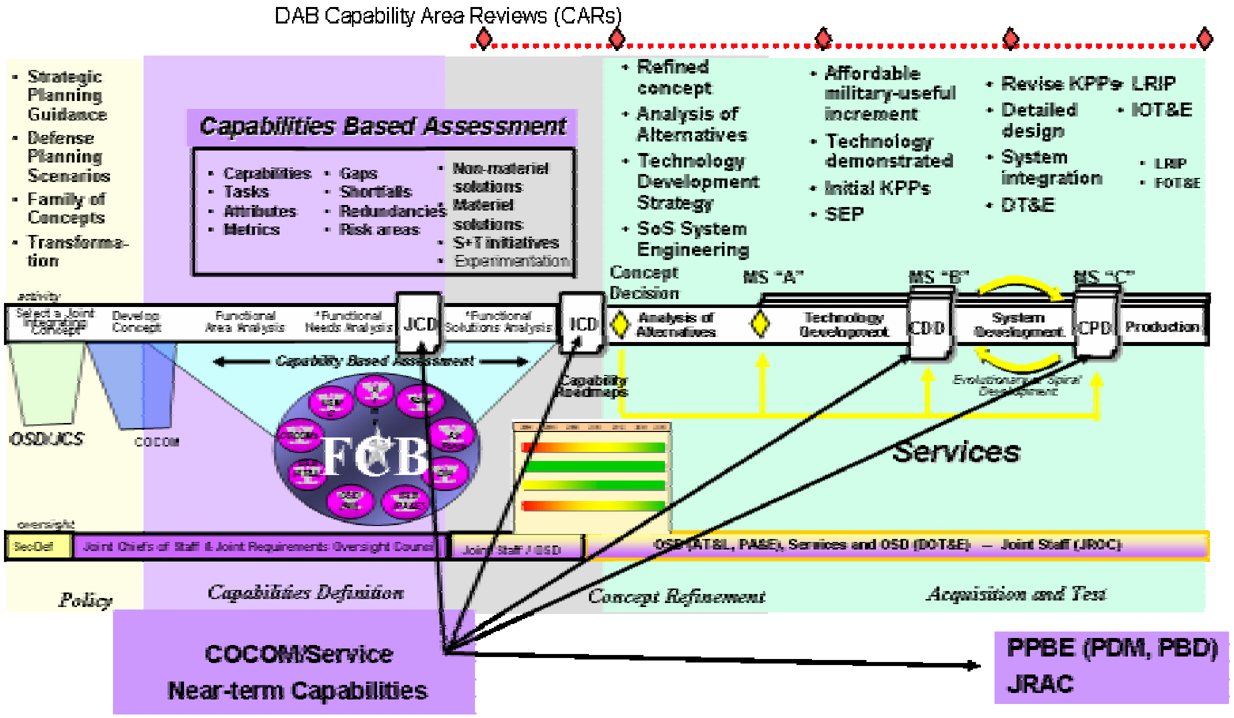
- DOTMLPF (Doctrine, Organization, Training, Materiel, Leadership and Education, Personnel and Facilities)

that taken together enables the application of military force.

Requirements are detailed descriptions pertaining to two separate and very different decompositions covering:

- User needs
- System makeup and attributes

The JCIDS Framework: *Still a Waterfall*



DAB Capability Area Reviews (CARs)

- Strategic Planning Guidance
- Defense Planning Scenarios
- Family of Concepts
- Transformation

Capabilities Based Assessment

- Capabilities
- Tasks
- Attributes
- Metrics

- Gaps
- Shortfalls
- Redundancies
- Risk areas

- Non-materiel solutions
- Materiel solutions
- S+T initiatives
- Experimentation

- Refined concept
- Analysis of Alternatives
- Technology Development Strategy
- SoS System Engineering

- Affordable military-useful increment
- Technology demonstrated
- Initial KPPs
- SEP

- Revise KPPs
- Detailed design
- System integration
- DT&E

- LRIP
- IOT&E

- LRIP
- FOT&E

This diagram describes the current capabilities based end-to-end process starting with Guidance and concept development, through capability identification and the ***handoff*** to the acquisition process where new materiel systems are required.

CBP & JCIDS, Col Gino Dellarocco and LTC Todd Key, Joint Staff/J8, 7 Feb 2006

The JCIDS Framework: Beyond the Handoff

JCIDS and current capabilities based approaches still assume a more or less simple **handoff** of capabilities descriptions to the acquisition community…

However, transforming joint capabilities into joint solutions is a complex engineering task requiring the discipline of CE.

The current capabilities based approach is therefore insufficient.

Full-fledged CE is needed and the CEF has to address how CE can be made to achieve this.

This will include establishing a viable evolutionary approach that is timely in meeting warfighter needs.

A Direction Beyond JCIDS

A Report by the Assessment Panel of the Defense Acquisition Performance Assessment Project for the Deputy Secretary of Defense

Defense Acquisition Performance Assessment (DAPA) **Report, January 2006** ***Panel***

Lieutenant General Ronald Kadish, USAF (Ret) Panel Chairman supported by several others including

- General Richard Hawley, USAF (Ret)
- General Paul Kern, USA (Ret).

The following slides are based on this report.

The Kadish Report addresses concerns – requirements being a primary one – that JCIDS does not.

DAPA on Requirements Socialization

CoCOMs participate but do not

- play a leading role in defining capability shortfalls
- have mechanisms to identify areas of excess capability.

Current acquisition encourages a “Conspiracy of Hope” allowing industry to

- propose unrealistic cost
- optimistic performance
- understate technical risk
- and cost plus contracts don’t help.

An environment of open and face-to-face communication is needed for

- **industry to understand government requirements**
- **government to understand industry capabilities.**

Other Requirements Socialization

Though not raised by DAPA,

open face-to-face environments may be needed for other mixes of requirements stakeholders since

- There is a significant disconnect between “requirements development” and budget processes
  - requirements being fixed before they are understood.
- There are many instances where systems formerly declared Not Operationally Effective by OT&E were actually fielded and useful.
  - On the other hand, Capabilities are often added to systems without checking with others the impact on life-cycle costs.
- There is an absence of systems engineering in requirements development process
  - resulting in poor conversion of capabilities into viable requirements.

DAPA on JCIDS

JCIDS

- has not sufficiently addressed the lack of technology maturity information.
- has no time-phased, fiscally and technically informed capabilities development/divestment plan.
- is slow and complex, ill-suited to respond to urgent needs.
- is structured for a “Cold War,” traditional opponent.

DAPA on
Time Certain Requirements Development

Time Certain Development enforces evolutionary acquisition by making time the focus of the up front requirement statement.

- Capabilities should be upgraded over time as technologies mature and operational requirements become clearer.
- Established durations are not adjusted to accommodate capability enhancements prior to fielding useful capabilities.
- Evolutionary acquisition, spiral development or block upgrades will be used for the inclusion of enhancements and increased requirements.
- Extra attention will need to be put on requirements traceability

The DAPA Report and Beyond

The DAPA Report makes some useful criticisms and recommendations re JCIDS

However, it does not address major issues in capabilities engineering:

- disciplined interaction among CoCOMs, CBTDEV and MATDEVs
- joint processes needed to produce requirements for SOS and NCW

ASSIP

US Army Strategic Software Improvement Program (ASSIP)

An AAE initiative:

- A long-term commitment to institutionalize software innovation and improvement
- Focused on people, programs, production and sustainment, and continuous process and product improvement

A partnership between ASA(ALT), PEO/PMs, Software Engineering Centers (SECs) and the Software Engineering Institute (SEI)

The Capabilities Engineering Framework is a new ASSIP initiative planned to extend through FY ’08.

Work in ‘06 is to build a **Capabilities Engineering Framework (CEF) straw man that can be seen as useful to user, development and acquisition communities.**

A Framework for Capabilities Engineering

Definition of Capabilities Engineering (CE):

Discipline for engineering needed capabilities into joint solutions (e.g., systems of systems) meeting war fighter needs with acquisition efficiency.

Capabilities Engineering Framework (CEF)

- **Articulates an efficient separation of concerns and shared concepts for aligning work**
- Identifies all relevant CE stakeholders and touch points (interfaces)
- Provides guiding principles, processes, methods and tools for a new way of doing business.

Opportunities for Improvement

New processes, techniques and tools for translating capabilities into viable product requirements

Reconfiguration of touch points and interaction to handle SOS, rapid changes in user needs and technology evolution.

Articulation of common ground or shared understanding – are new shared abstractions needed?

Unintended emergent consequences (often negative) of Systems of Systems

Integration of user innovation and rapid acquisition into the formal acquisition process including end user feedback mechanisms, processes and techniques

Redirection of budgetary constraints on people who fill skilled systems and software engineering roles

Defining the CEF – Stakeholders

Major CEF Developers and Users will include

- End Users (Combat Commanders and other warfighters)
- Combat Developers (who represent end users – TRADOC)
- Material Developers (who run the acquisition process and work with the industrial base)
- Testers and Testing Facilities (ATEC, CTSF, etc.)
- Systems/Software Engineers from the Army, research communities and commercial organizations.
- The Industrial Base

Senior Leadership Buy-in: Involvement of senior Army leadership minimally from TRADOC and Acquisition communities is crucial.

Defining the CEF – Guiding Principles

Bottom Line: Stakeholders collaborate early and throughout the lifecycle so that end users (warfighters) get quality solutions in a timely manner…and

- Ideal touch points, interfaces, protocols and respective organizational behaviors are characterized and understood by all stakeholders.
- Current formal acquisition is streamlined so that user driven and rapid acquisition are accommodated as a matter of course.
- Viable emergent* approaches for formulating (joint) capabilities for agile SOS evolution are created and applied.
- Capability formulation co-evolves with Enterprise/SOS architectures, *emphasizing quality attributes e.g., interoperability.*
- *Accuracy of* capability definition is not compromised by premature precision and completeness.

* See Monarch, Fisher and Wessel, "Requirements Engineering from an Emergence Perspective in the Era of Systems of Systems and Net-Centric Warfare," submitted to 14th International RE'06 Conference

CEF Key Focus Area 1:
User Driven Innovation

Need environments for Combat Developers (CBTDEVs) and Material Developers (MATDEVs).

- user-driven innovation, in wartime
- new means for achieving shared understanding

A possible institutional starting point are the Battle Labs.

Some of the Battle Labs have been key to integrating user driven rapid acquisition with more Army-wide system acquisition undertaken by PMOs.

If this is a good starting point, how would the interaction between CBTDEVs and MATDEVs continue into acquisition settings?

CEF Key Focus Area 1: WRAP 1

Warfighting Rapid Acquisition Program (WRAP) provides both a:

- Bridge linking TRADOC experimentation with systems acquisition
- Mechanism to accelerate the acquisition of selected operational warfighting enhancements.

Determine whether experimental candidates warrant rapid acquisition and are affordable, effective, sustainable, and suitable.

CEF Key Focus Area 1: WRAP 2

WRAP continued:

Though highly touted in the first few years after its inception, in 1998, the GAO had this to say,

“… the Army's criteria for selecting WRAP candidates are open-ended and allow room for different interpretations…”

Again a lack of shared understanding…

But perhaps the real reason is lack of funding.

In any case, the WRAP appears not to have been used much for several years.

CEF Key Focus Area 1: Recommendations

Stakeholders interact continuously at all stages of the requirements development and acquisition process

Stakeholder networks need to be created and maintained, i.e.,

- new links or revitalizing existing links
- new processes for making use of these links.

Past attempts at building institutional bridges between CBTDEVs and MATDEVs need to be leveraged.

Embed SEs and SwEs in the field.

Modeling and Simulation are needed for CBTDEVs and MATDEVs re inserting SOS technology in the battlefield.

CEF Key Focus Area 2:
Shared Abstractions

Shared understanding requires shared abstractions.

Possible Starting Point:

Doctrine such as Universal Joint Task List (UJTL) and in the Army Universal Task List (AUTL)

Capability engineering requires an entirely new lexicon permitting relationships to be drawn between mission areas, capabilities, systems.

This requirement extends to the component systems that comprise a capability enabling effective decomposition.

What is especially needed is a method for decomposing software to minimal functional units.

Key Focus Area 2: Recommendations 2

Need abstractions to be used unambiguously across boundaries, especially between CBTDEVs and MATDEVs.

Methods and processes need to be developed for developing and disseminating these abstractions by

- making use of UJTL and AUTL to the extent possible
- developing shared representations of systems down to software components
- developing modeling and simulation tools that no longer "assume away" crucial information
- organizing SW architecture patterns on the basis of shared abstractions.

CEF Key Focus Area 3:
Co-Evolution of SOS Requirements & Architecture

Can SOS architecture co-evolve with requirements development?

Can co-evolution of SOS architectures and requirements help build more effective relationships among

- user
- acquisition
- engineering communities?

Architecture repositories exist but are not sufficiently used for architecture development.

Key Focus Area 3: Design Patterns

Design patterns

- take into consideration context of use
- support the co-evolution of doctrine, requirements and architecture to
  - insure an architecture makes sense in the world in which it is intended to be implemented
  - facilitate capturing in an architecture what a CDD references
  - expose gaps and ambiguities in doctrine (shared abstractions)
  - identify troublesome requirements that may be
    - too detailed, not feasible
    - beyond the range of variability for a product line

Key Focus Area 3:
Design Pattern Recommendations

Establish a DoD/Army infrastructure service on the web for software architecture patterns and components that will be

- reused across multiple programs
- harvested from existing Army inventories.
- contractually required
- managed as critical assets of military intellectual property and sources of competitive advantage.

Aligning CEF to Army User Organizations
TRADOC

We are currently working with two groups in TRADOC:

- **Army Capabilities Integration Center (ARCIC)**
- **TRADOC Program Integration Office (TPIO)**

These organizations represent warfighters.

The aim of the CEF with respect to these organizations is to:

- identify touch points among all CE stakeholders
  - acquirers, testers, engineering specialists and industry
- provide all these CE stakeholders a disciplined set of interactions, infusing system and software engineering into the entire life cycle
  - from capabilities development to user and product requirements and to development, testing and fielding.

Conclusion 1

Software and Capability Engineering include users (warfighters and their representatives) as full-fledged participants/innovators.

The military has and continues to devote much time, thought and effort in creating a capabilities organizational infrastructure.

ASSIP’s Capabilities Engineering Framework (CEF) will align with this infrastructure and recommend changes where needed.

Work on the CEF began in FY ’06 and is planned to continue for the next several years.

A straw man CEF will be completed by the end of ’06.

We believe the CEF is relevant to all the Services and the Joint Force.

Conclusion 2

R&D re CE maturity and the way ahead will include investigating processes, techniques and tools for

- translating capabilities into requirements
- handling and leveraging SOS emergent behavior
- establishing shared abstractions across users, acquirers and developers
- reconfiguration of touch points and interaction discipline wrt
  - SOS
  - rapid changes in user needs
  - technology evolution
- integration of user innovation and rapid acquisition into the formal acquisition process
- end user feedback mechanisms, processes and techniques.

Acronyms 1

ABCS	Army Battle Command System
ARCIC	Army Capabilities Integration Center
ASSIP	Army Strategic Software Improvement Program
ATEC	Army Test and Evaluation Command
AUTL	Army Universal Task List
CBTDEV	Combat Developer
CDD	Capabilities Development Document
CE	Capabilities Engineering
CEF	Capabilities Engineering Framework
CoCOM	Combat Commander
CTSF	Central Technical Systems Facility
DAPA	Defense Acquisition Performance Assessment
DOTMLPF	Doctrine, Organization, Training, Materiel, Leadership and Education, Personnel and Facilities

JCIDS	Joint Capabilities Integration and Development System
JFCOM	Joint Force Command
JROC	Joint Requirements Oversight Council
MATDEV	Material Developer
NCS	Network Centric System,
NCW	Network Centric Warfare
OSS	Open Source Software
OT&E	Operational Test and Evaluation
OV	Operational View
PEO	Program Executive Office
PMO	Program Management Office
PPBES	Planning, Programming, Budgeting and Execution System
SOS	System of Systems
SV	System View
SwE	Software Engineering
TV	Technical View
TRADOC	Training and Doctrine Command
TPIO-BC	TRADOC Program Integration Office-Battle Command
UJTL	Universal Joint Task List
WRAP	Warfighting Rapid Acquisition Program