



COTS-BASED SYSTEMS

With Open Source Components

**Patricia Oberndorf and
Scott Hissam**

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

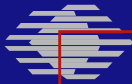
Sponsored by the Office of the
U.S. Secretary of Defense



Some COTS-Based System Fundamentals

Enter Open Source

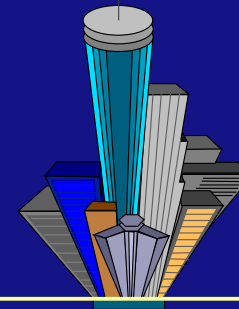
Conclusions



COTS: Attraction and Motivation

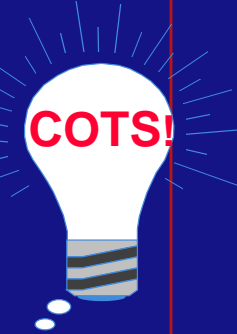


legacy systems



demands
for more
functionality

- develop faster
- adopt best commercial practices
- leverage commercial investment
- leverage new technology
- reduce costs

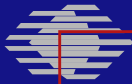




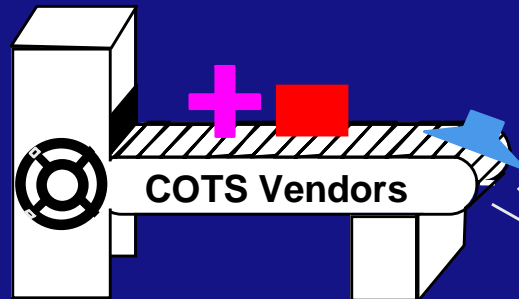
What is COTS?

A COTS product is a product

- **sold, leased, or licensed to the general public**
- **offered by a vendor trying to profit from it**
- **supported and evolved by the vendor, who retains the intellectual property rights**
- **available in multiple, identical copies**
- **used without modification of the internals**

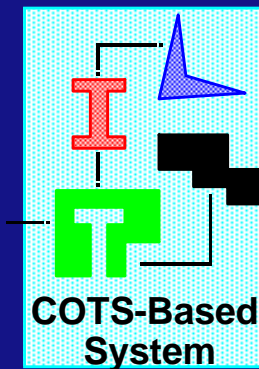


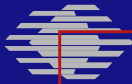
What Makes COTS Challenging? -1



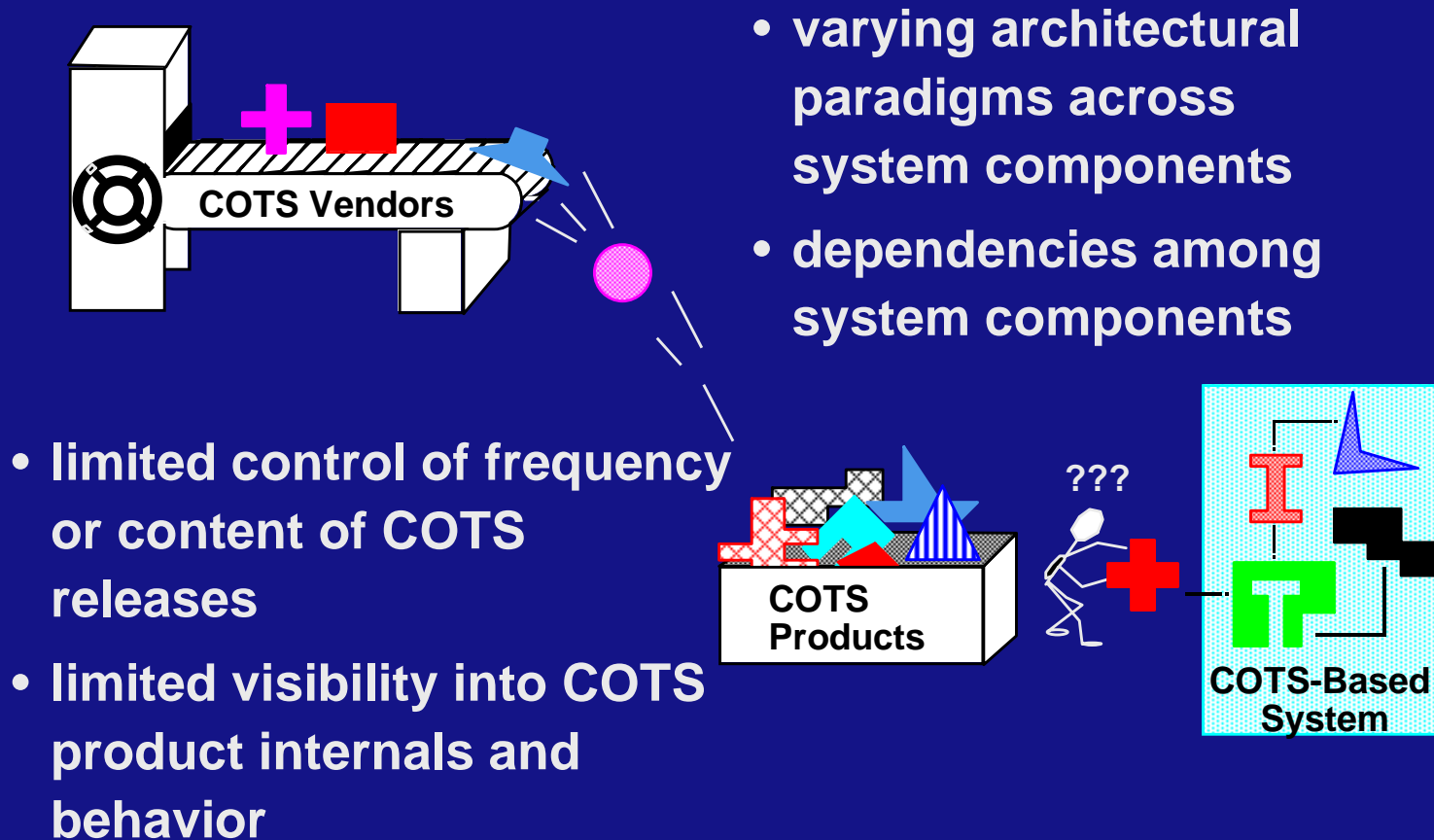
- built-in assumptions of end-user processes that may not match yours
- licensing, data rights, warranties

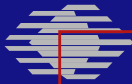
- frequent, continuous change of COTS products and marketplace
- COTS products driven by marketplace, not your system context





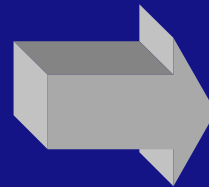
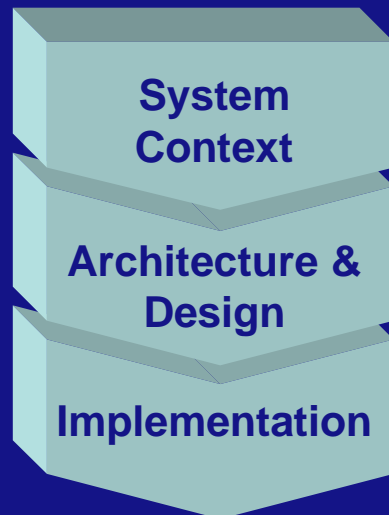
What Makes COTS Challenging? -2



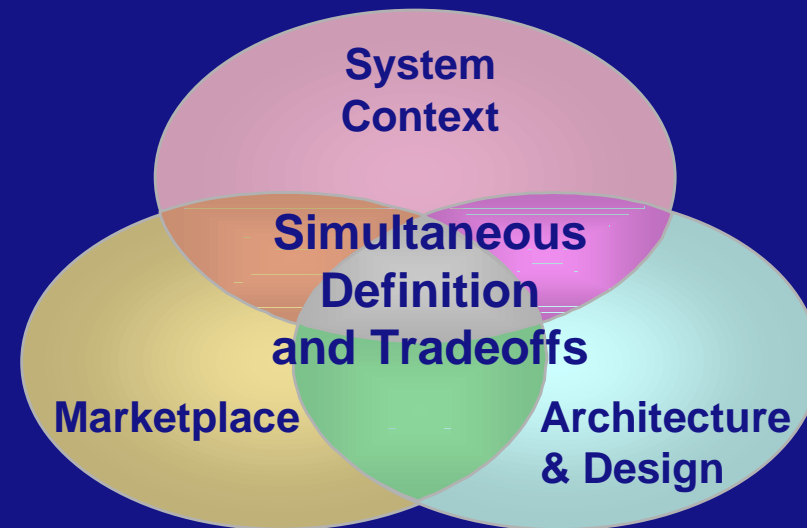


CBS Fundamental Change

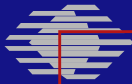
Traditional Development Approach



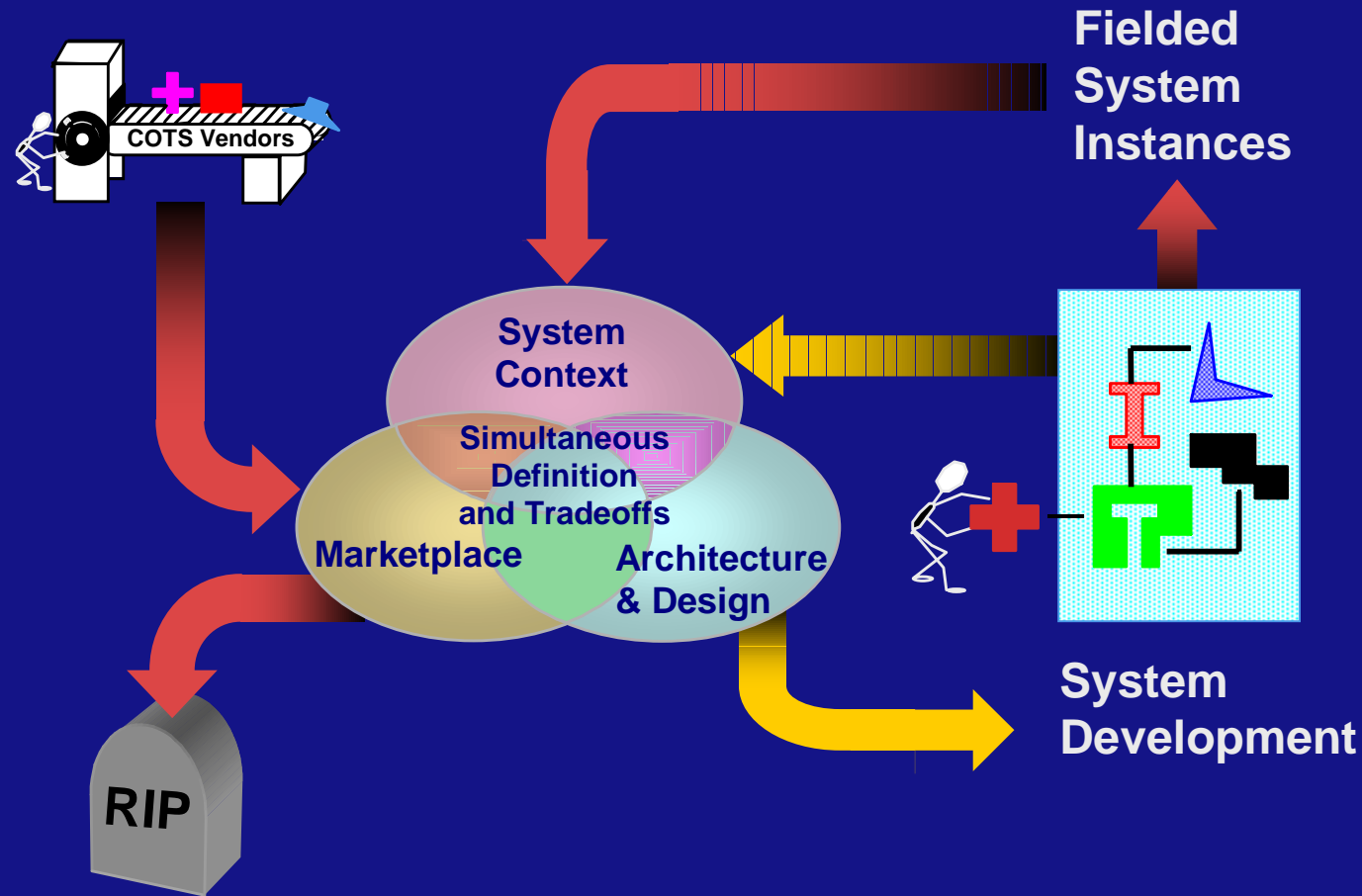
Required COTS Approach



This change applies to *business* and *contractual* activities, as well as the *engineering* ones.



Cyclic Nature of CBS

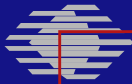




Some COTS-Based System Fundamentals



Enter Open Source
Conclusions



What is Open Source Software?

A special category of software that is distributed in a source code form that can be interpreted by software developers.

Open source software:

- **may or may not be free**
- **often has licenses requiring further sharing**

“Open sourcing is considered a premier means to ensuring rapid technology development, as well as to ensuring that the best technology is available to both users and developers.” --

http://linux.corel.com/resources_faqs.htm#2



Examples of Open Source Software

Operating Systems

- Linux, NetBSD, FreeBSD, OpenBSD (Unix derivatives)
- ReactOS, a WinNT-like OS supporting drivers and applications

GNU tools

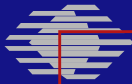
- suite of POSIX utilities (e.g., file, email, compilers)

Languages

- Perl, Tcl

Specialized applications

- MIT's X Windows, XFree86
- WWW Browsers (Communicator, Mozilla, Lynx)
- WWW Servers (Apache, Samba)



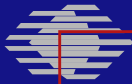
Who's Using Open Source Software and Development?

Just about any company that is using some variant of Unix

- **it is just behind the veil of the OS vendor**

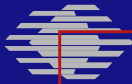
More directly...

- **BSDi**
- **SCO**
- **IBM**
- **Apple**
- **... many more, plus conferences & services**



Comparing COTS & Open Source

COTS	Open Source
sold, leased, or licensed to the general public	licensed to the general public
vendor trying to <i>profit</i> from it	author/source wanting to <i>share</i>
vendor <i>retains</i> the IPR	author/source <i>gives up IPR</i> - <u>required to share source</u>
available in multiple, <i>identical</i> copies	potential for many <i>nonidentical</i> copies
used <i>without modification</i> of the internals	<i>source modification</i> is the whole idea



Strengths of Open Source Software

Collaborative access to motivated, talented, and specialized software engineers

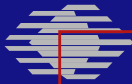
- **variability in the motivation, the degree of talent, and true capabilities**

Access to broad spectrum of peer reviewers

- **built-in “checks-and-balances” against the developers**

Patches can happen in a matter of hours

- **frequently changing**



Weaknesses of Open Source Software

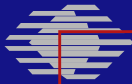
Talented pool of peer developers can lead to fractions in baseline source.

Quality Assurance is left to the masses.

- **has everything been checked?**

Open Source licensing agreements can *hinder* creativity

- **a company that hardens Linux cannot sell that hardened version without releasing the source - opens itself back up to vulnerabilities (Catch 22)**



Potential Impact to Organizations -1

Acquisition

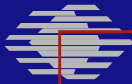
- is it free?
- who owns this stuff?

Development

- who did this?
- how does it work?

Maintenance

- when something is not right, whom do you call? (see bullet above)



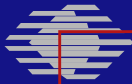
Potential Impact to Organizations -2

Organizations may find themselves a new line of business

- tempting to add a feature to save big \$\$\$
- added features must be repeatedly cut back in to new releases

Schedule

- easy victim to early or late releases



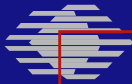
Potential Impact to Organizations -3

Test & verification

- how do you know this stuff does *only* what it is supposed to do?

Security

- off-shore development, malicious intent, poor peer review (bad business)
- malicious hackers can review code and find vulnerabilities

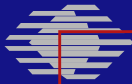


Some COTS-Based System Fundamentals

Enter Open Source

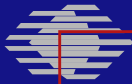


Conclusions



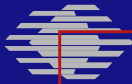
A Counterbalance to the COTS Challenges? -1

- frequent, continuous change of COTS products and marketplace
 - **accelerated by open source**
- COTS products driven by marketplace, not your system context
 - **still true, but you can make it what you want**
- built-in assumptions of end-user processes that may not match yours
 - **still true**



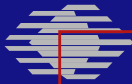
A Counterbalance to the COTS Challenges? -2

- **licensing, data rights, warranties**
 - **licensing still needs attention; warranties unlikely**
- **limited control of frequency or content of COTS releases**
 - **still true**
- **limited visibility into COTS product internals and behavior**
 - **the biggest change**



A Counterbalance to the COTS Challenges? -3

- **varying architectural paradigms across system components**
 - **still true**
- **dependencies among system components**
 - **still possible, and changes could create more, but at least you'll have more visibility into them**

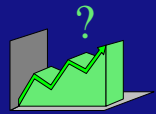


Keys to CBS Success



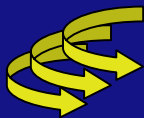
Make COTS-Based System Tradeoffs

- Reconcile Products and User Operations
- Leverage the Marketplace
- Engineer an Evolvable Architecture
- Make Tradeoffs Simultaneously
- Avoid COTS Modification



Think More Like a Business

- Live by the COTS Business Case
- Negotiate Licenses & Supplier Relationships
- Realign Budgets for COTS Realities

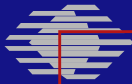


Establish Evolution as a Way of Life

- Evolve COTS-Based Systems Continuously
- Take the Long View on System Acquisition



Change the Culture



Some Open Source References

Frequently Asked Questions about Open Source

- <http://www.opensource.org/faq.html>
- also try <http://www.free-soft.org/>

The Cathedral and the Bazaar

- <http://199.183.24.253/redhat/cathedral-bazaar/>
- <http://computer.org/software/Interview.htm>

Linux on the Move

- IEEE Software Jan/Feb 1999
 - <http://computer.org/software/GEI.pdf>

The Halloween Documents

- <http://www.opensource.org/halloween>



COTS-BASED SYSTEMS

With Open Source Components

**Patricia Oberndorf and
Scott Hissam**

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Sponsored by the Office of the
U.S. Secretary of Defense



For Additional Information

URL: www.sei.cmu.edu/cbs

Patricia Oberndorf

Voice: 412 / 268-6138

Email: po@sei.cmu.edu

Fax: 412 / 268-5758

Scott Hissam

Voice: 412/268-6526

Email: shissam@sei.cmu.edu

Fax: 412 / 268-5758

**4500 Fifth Ave.
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213**