



Gotchas of COTS for Acquisition Managers: What You Don't Know Can Hurt You

Carol Sledge and Tricia Oberndorf

**Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213**

This work is sponsored by the U.S. Department of Defense.



Presentation Outline

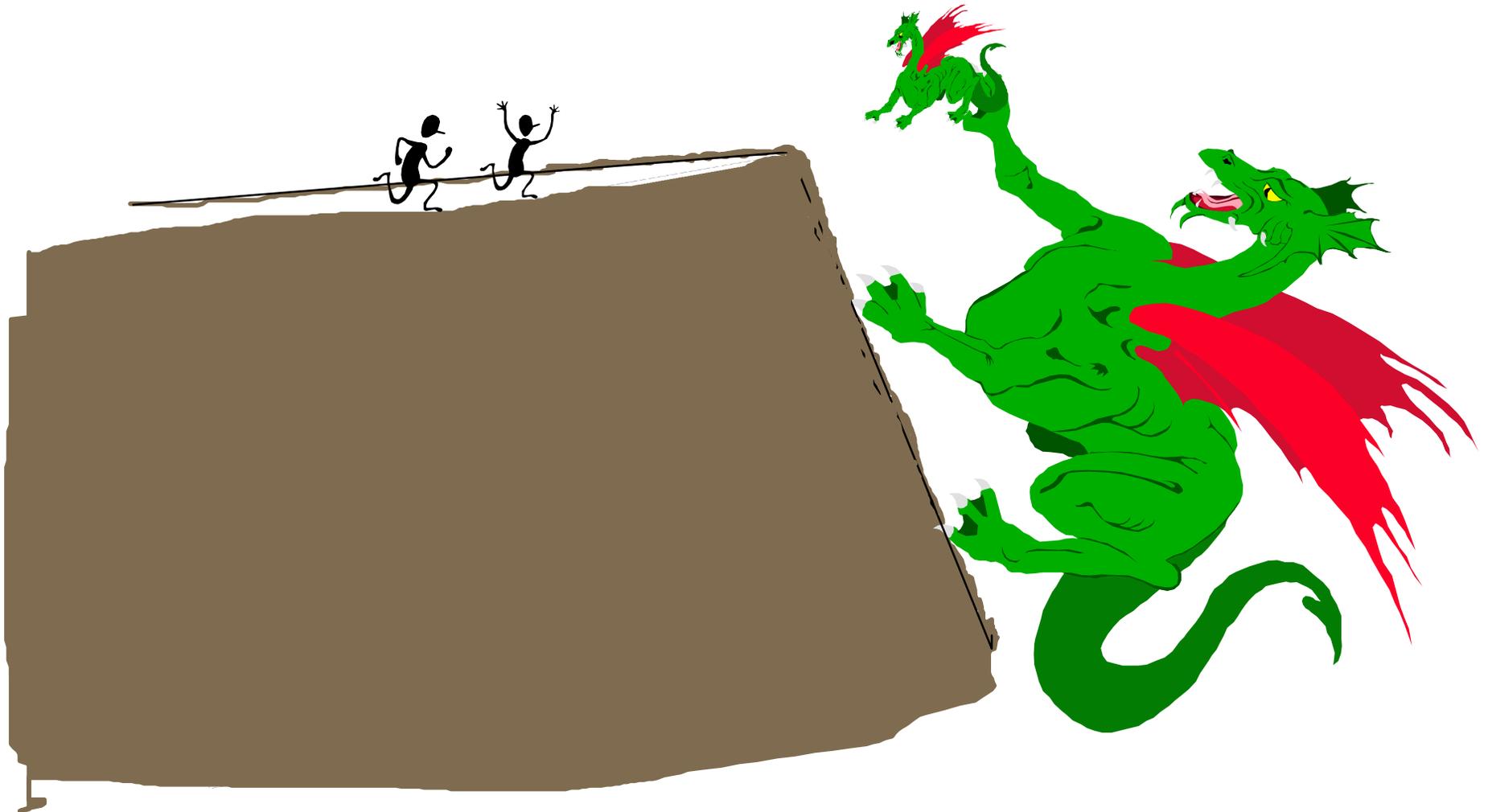
The “Gotchas” of commercial off-the-shelf (COTS) products

- **COTS is only a technical change.**
- **We’ll just modify the COTS.**
- **We never did it that way before.**
- **I don’t have to test any more.**
- **One size fits all systems for all time.**
- **Industry/My contractor will do it all.**
- **COTS will save me money.**

Conclusions



COTS is only a technical change.





COTS is only a technical change. -1

Processes change.

- Processes may not match.
 - COTS products have inherent processes.
 - End-users have their own established processes.

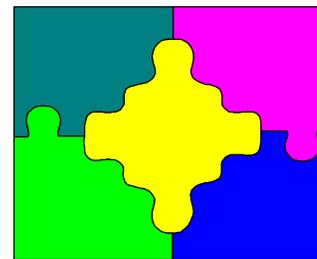
Consider Business Process Reengineering.

- Disciplined development processes are affected.



creator

vs.



integrator

Redefine existing development processes accordingly.



COTS is only a technical change. -2

Culture changes.

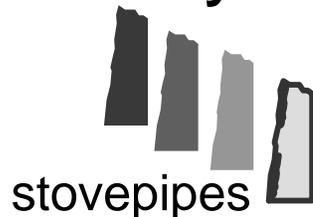
- Caveat emptor.

Become a smart consumer.

Empower Integrated Product Teams (IPTs).

Educate and train.

- There may be cultural issues.

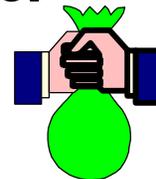


adversarial
attitudes

- People need incentives to change.



- sharing cost savings
- recognition

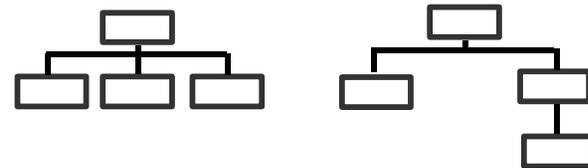




COTS is only a technical change. -3

Organizations change.

- **Changes in roles and responsibilities can reshape organization charts.**



Infrastructure changes.

- **Acquisition structure has a long lead time compared to the much faster turnover of technology.**



- **“Color of money” conflicts can cause problems.**
- **Some guidance has not caught up with the shift to COTS.**



We'll just modify the COTS.





We'll just modify the COTS -1

If the vendor does it

- **bring \$\$\$**



- **may not track with COTS releases**
- **may no longer work with other products**

Include in license agreement that the vendor feeds changes into next COTS release.



We'll just modify the COTS -2

If the vendor does not agree to do it, and you or the prime decide to do it

- still bring \$\$\$
- probably have inadequate maintenance and design documentation
- difficult to track vendor's new COTS releases
- modified off-the-shelf (MOTS) will stagnate as other products it depends on change and evolve



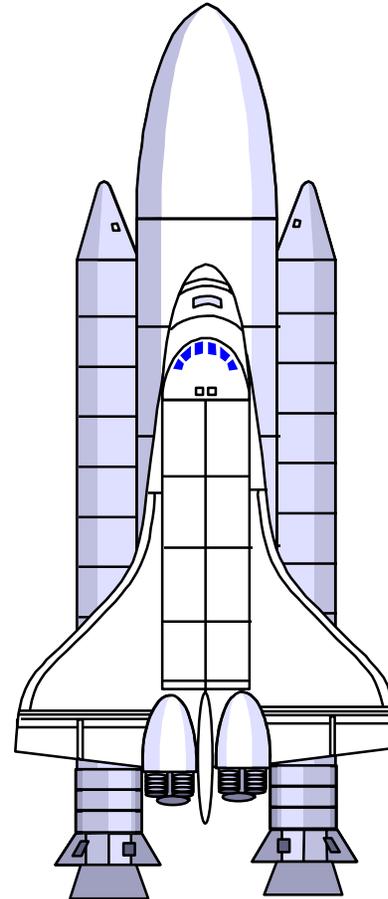
– You just bought the worst of both worlds!

If you use built-in customizations

- may not migrate with the product
- need vendor “guarantee” that customizations will migrate



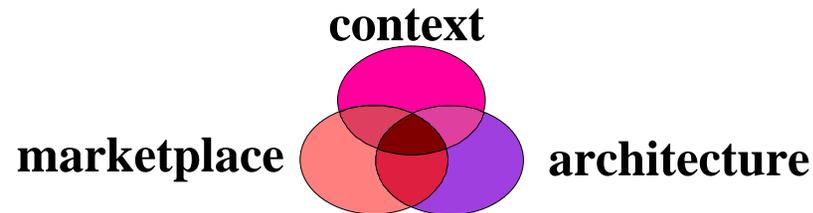
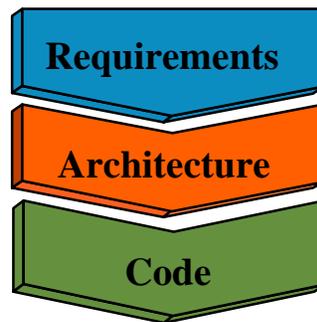
We never did it that way before.





We never did it that way before. -1

Say goodbye to the waterfall!



- Marketplace changes constantly, so keep a finger on the pulse.

Start a market research group.

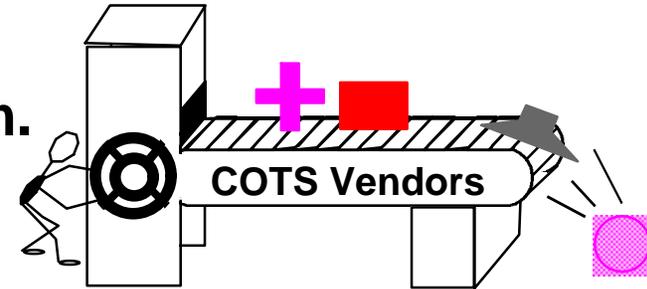
Make COTS decisions to balance system stability with the avoidance of obsolescence.



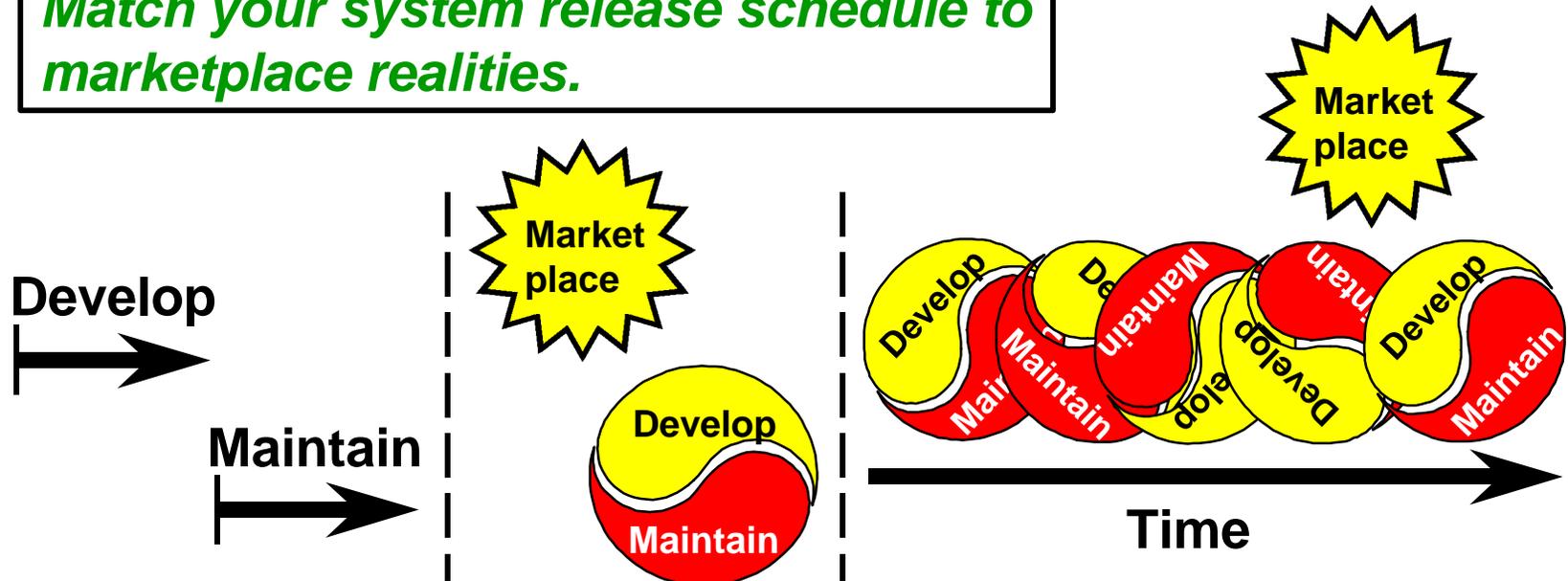
We never did it that way before. -2

COTS for life ... cycle!

- Benefit occurs over the long term.
- Change is constant.
- Changes will occur before anything is fielded.
- Development and maintenance are no longer distinct.



Match your system release schedule to marketplace realities.

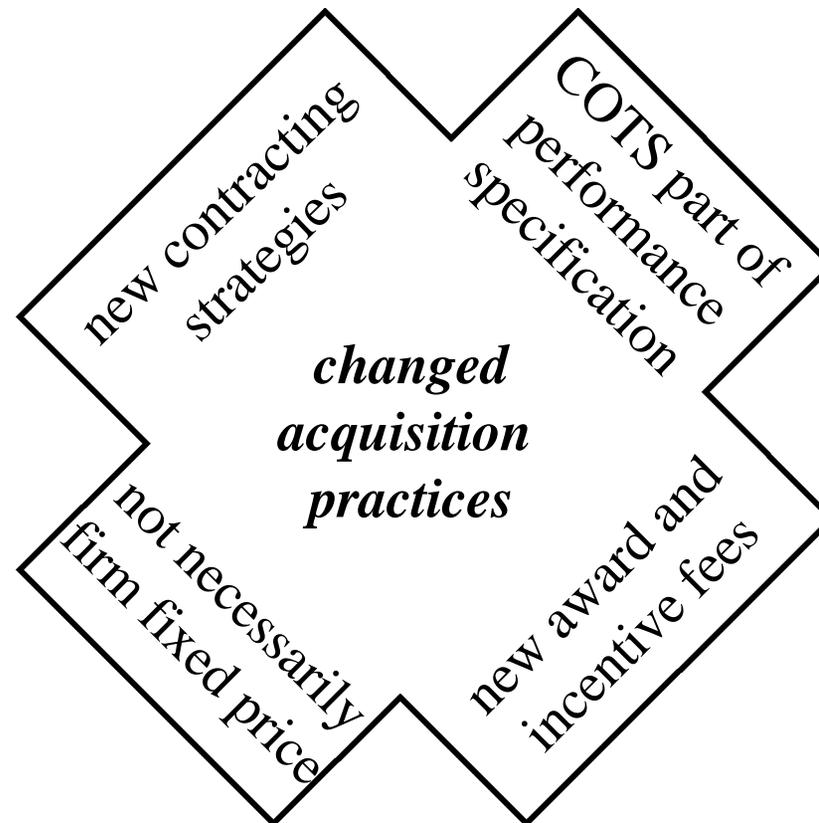




We never did it that way before. -3

But I can't tell the contractor what to do.

- **Use of COTS changes the way you do acquisition.**





We never did it that way before. -4

But I can't tell the contractor what to do.

- Prime may have strategic alliances with certain vendors
- License negotiation is critical

*enterprise-wide vs. per seat
vs. per user vs. ...*

*no expiring
license keys*

*negotiation of other kinds
of maintenance/support
critical to product*

*development vs.
run-time licenses*

*contractors negotiate
passthrough licenses*



We never did it that way before. -5

**Contracting officers
Buyers
Lawyers**

can make you or break you.

Find good ones.

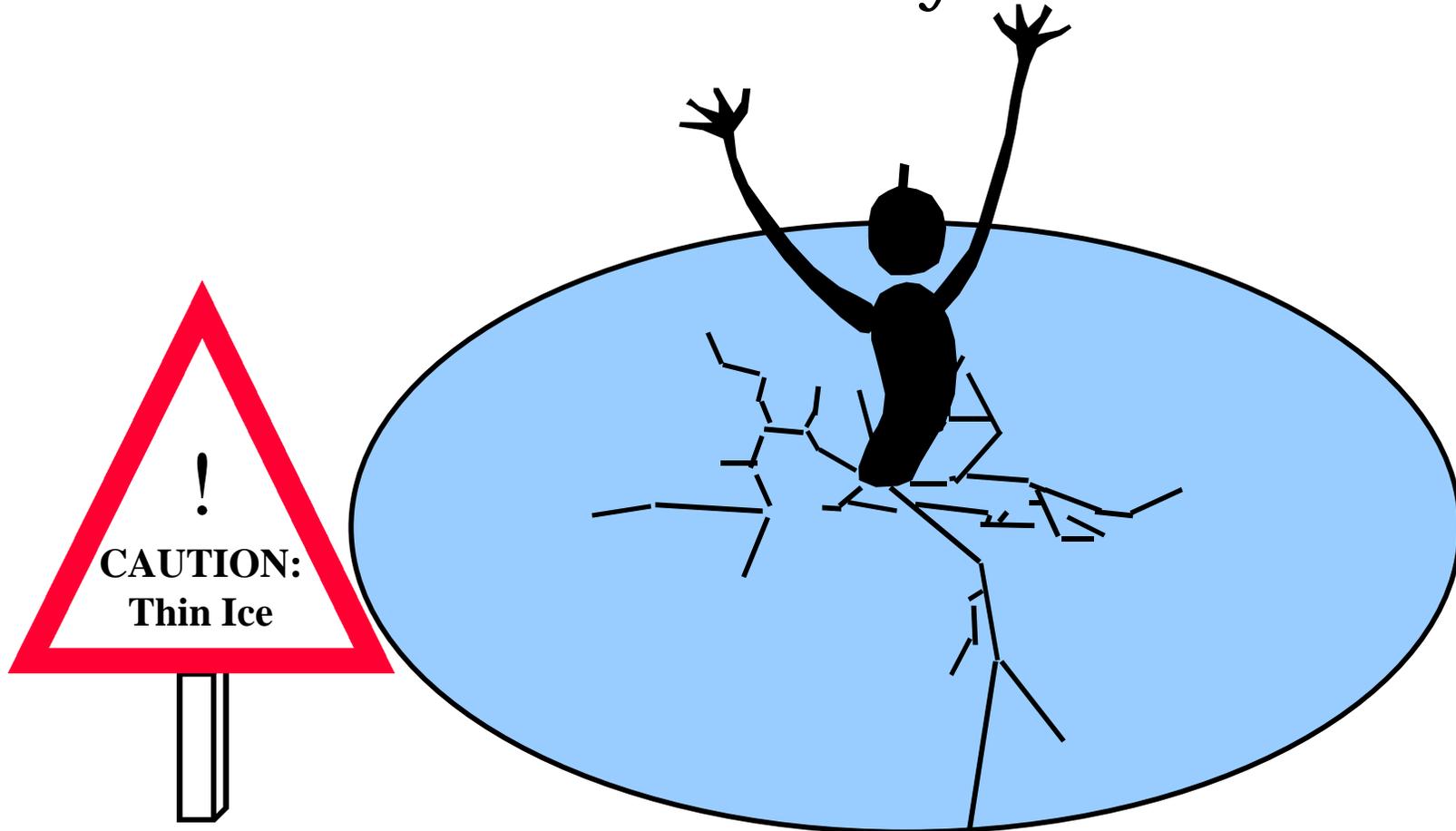
Involve them early (e.g., via IPTs).

If I don't own the code, what *do* I own?

- **the architecture**



I don't have to test anymore.





I don't have to test anymore. -1

Experiments/"testing" are critical to feature discovery.

No one else can test for everything you need to know.

- many kinds of tests



Quality



Conformance



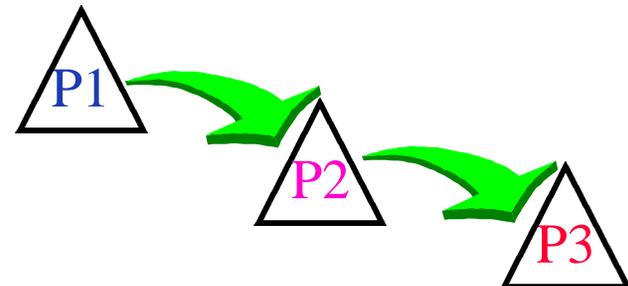
Users

- testing of new versions/releases

Determine what changed and the effects on your system.

Test continuously.

COTS changes have a ripple effect.

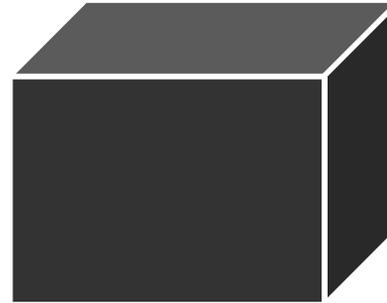
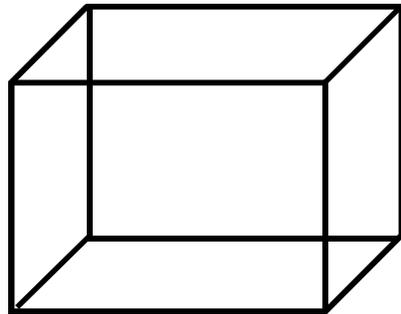




I don't have to test anymore. -2

Testing changes at the unit level.

- **need alternatives to white box testing**

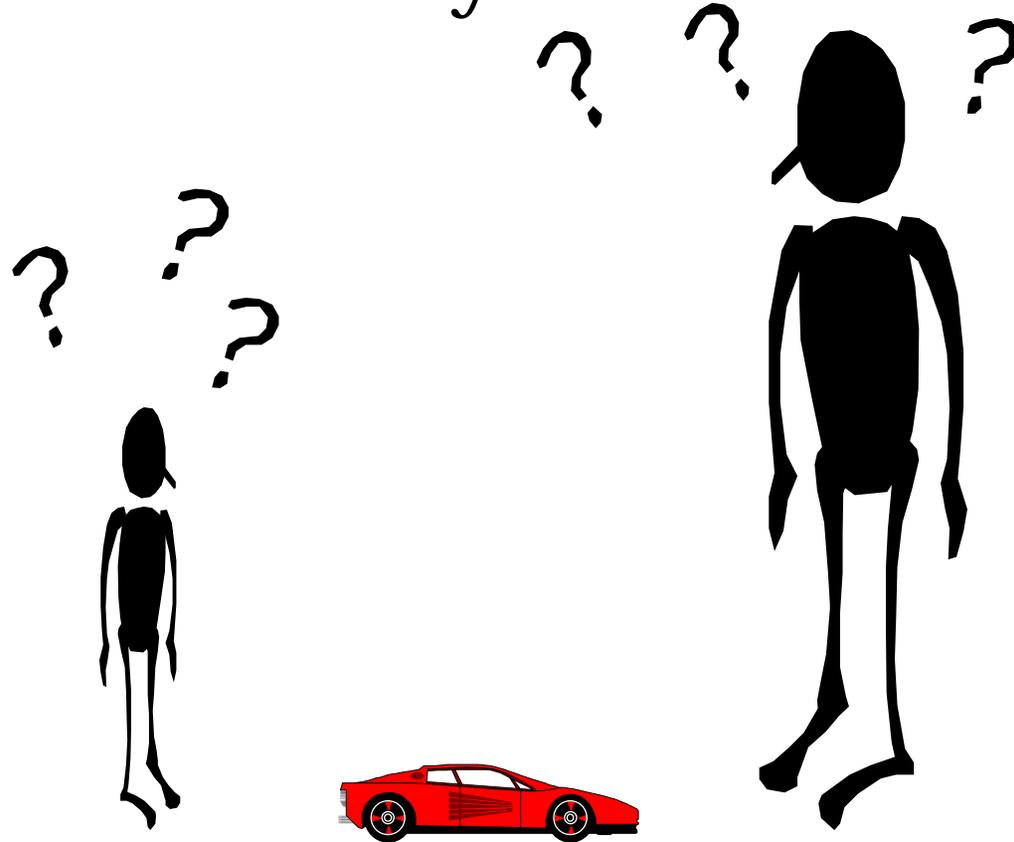


Integration and system testing do not go away.

- **old processes: some may still apply,
others need adjustment for COTS**



One size fits all systems for all time.





One size fits all systems for all time. -1

One process

- There is no one evaluation or design process for all.

Find a process for your situation.

Tailor processes wisely.

One architecture

Find appropriate architectural patterns and commonalities.

Consider similar systems and other work.

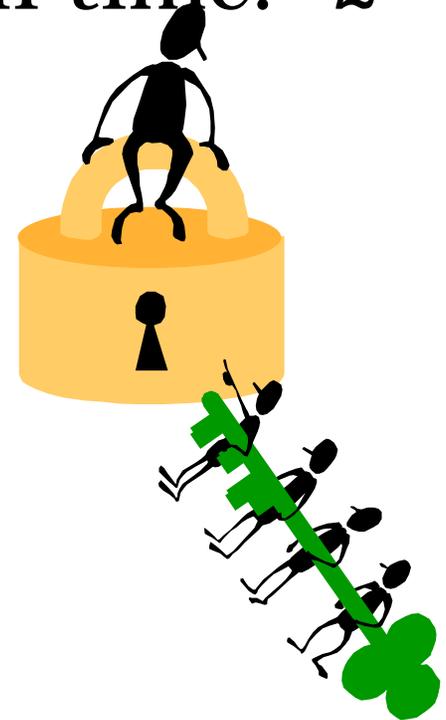
Watch for effects of COTS products on architectural decisions.



One size fits all systems for all time. -2

One standard

- Interoperability is dependent on standards, but not everything interoperates.
- There are other reasons for standards besides interoperability.
- Different circumstances dictate different choices.





One size fits all systems for all time. -3

One product

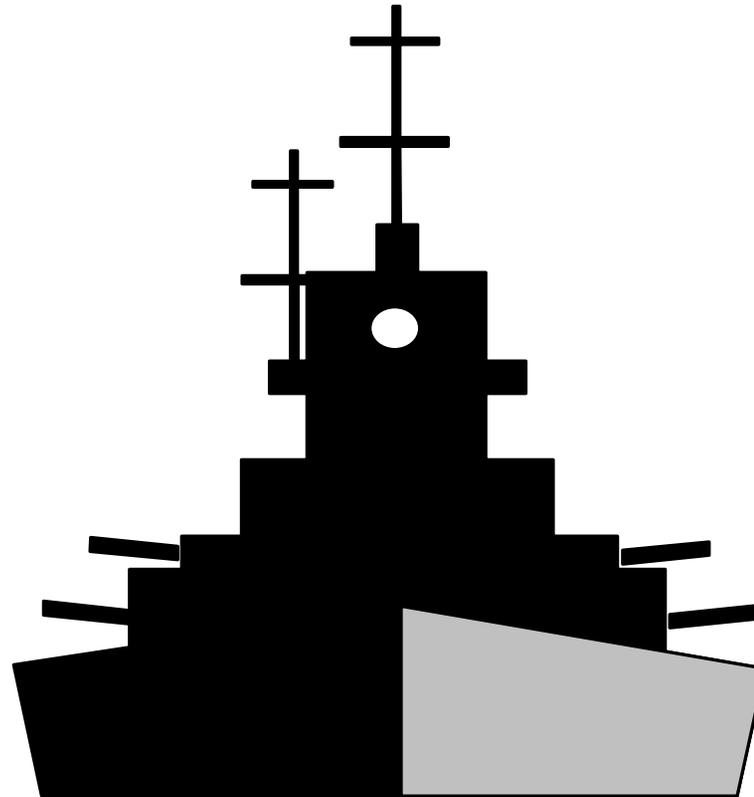
- Do not expect anyone to “just tell me the answer.”



Be a smart consumer.



Industry/My contractor will do it all.





Industry/My contractor will do it all. -1

My contractor will know how.

- This is new for the contractors, too.
- Responsibility for the system ultimately lies with the government.



- Life cycle/lifetime is what matters.

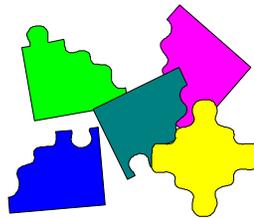
Attend to migration and maintenance.



Industry/My contractor will do it all. -2

Industry's already solved it.

- **“Plug-and-play” hasn't happened yet for software.**



- **Military-unique requirements exist (Merrill Lynch doesn't mobilize).**



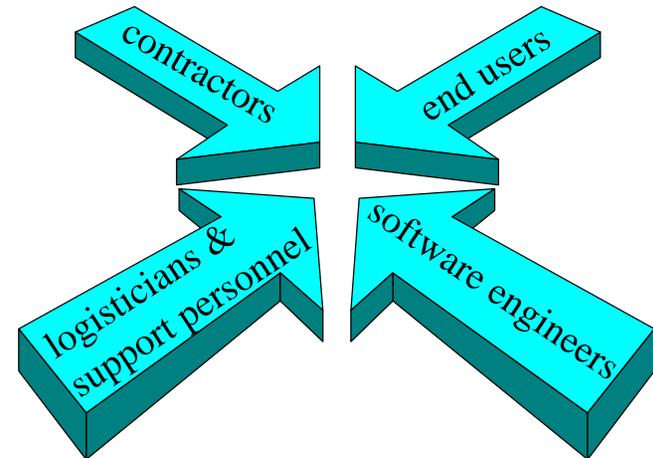
- **Military performance demands are not the norm in industry.**
- **Industry systems are commodity-based and more disposable.**
- **Architectures in an analogous industrial application may not fit the military context.**



Industry/My contractor will do it all. -3

IPTs will take care of it.

- not if they're stovepiped, not empowered, or inadequately staffed
- can be powerful tool for teaming



I have no control any more.

- Control may be *reduced*, but you still must exercise *influence*.

Be aware of the marketplace, including standards.

Establish rapport (and influence) with second-tier suppliers.



COTS will save me money.

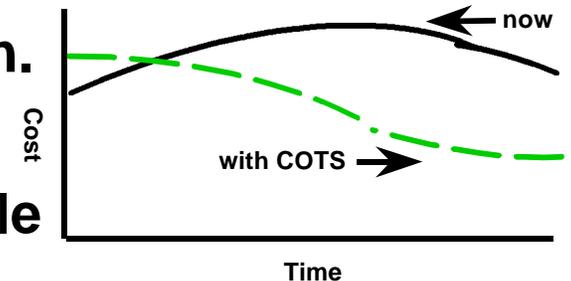




COTS will save me money. -1

The economics change, but how is unknown.

- not enough experience yet
- cost curve changes
- potential savings not immediately available

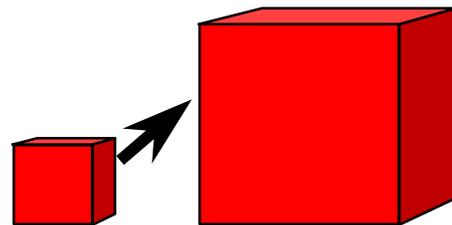


ROI may not be monetary.

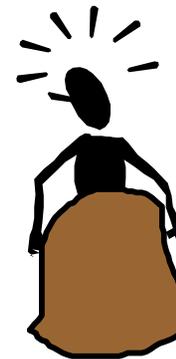
- may enjoy system improvements for the same investment
- when price tag for custom development outstrips the available budget, COTS may be the only way to field something



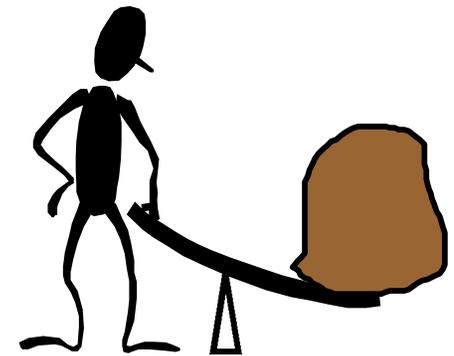
cost savings/
cost avoidance



increased capability



VS.



otherwise unachievable results



COTS will save me money. - 2

Savings may be hard to see at the level of a single system or site: benefits may only accrue at a higher level.

Short-term thinking can undermine long-term achievements

- **COTS decisions must be based on life-cycle considerations.**

Many aspects affect the bottom line:

- **requirements**
- **market research**
- **architecture/design**
- **prototyping**
- **evaluation**
- **integration**
- **negotiative relationship of context/architecture/marketplace**
- **testing**
- **technology forecasting**
- **standards**
- **per-unit cost amortizations**
- **reacting to new product releases**
- **reacting to marketplace changes**



Conclusions

COTS is not as easy as it sounds.

- need to understand
- need to do your homework
- need to adapt
 - development process
 - culture
 - organization
 - ...
- share
 - lessons learned
 - architecture
 - nondevelopmental items (NDI)

***COTS is like fire: If controlled it will help you,
if uncontrolled it will rise up and destroy you ...***

(adapted from Theodore Roosevelt quote.)



For Additional Information

Tricia Oberndorf

**Software Engineering Institute
Carnegie Mellon University
4500 Fifth Ave.
Pittsburgh, PA 15213-3890**

Voice: 412/268-6138

Fax: 412/268-5758

E-mail: po@sei.cmu.edu

Carol A. Sledge

**Software Engineering Institute
Carnegie Mellon University
4500 Fifth Ave.
Pittsburgh, PA 15213-3890**

Voice: 412/268-7708

Fax: 412/268-5758

E-mail: cas@sei.cmu.edu