

COTS RISKS AND RISK MITIGATION APPROACHES

For OSD

Ronald J. Kohl

R. J. Kohl & Assoc.

May 19, 2009

CONTENTS

- ✘ Definitions
- ✘ Benefits of using COTS
- ✘ Risks with using COTS
- ✘ Mitigation approaches
- ✘ 'Open' problems
- ✘ Summary
- ✘ References

DEFINITIONS

- ✘ Commercial Off The Shelf (COTS)
 - + Commercially available product acquired in 'as is' condition, perhaps with 'reconfiguring' capabilities
- ✘ Other Non-Developed Item (NDI) types
 - + MOTS (Modified Off The Shelf)
 - + GOTS (Government Off The Shelf)
 - + Reuse products
 - + Shareware
 - + OSS (Open Source Software)
- ✘ Custom
 - + Home grown or home maintained, control of source code and development team

POTENTIAL BENEFITS OF USING COTS

- ✘ Reduced development cost/schedule
- ✘ Reduced maintenance cost
- ✘ No cost product ‘improvements’
- ✘ “Proven” product
 - + Wide user base to identify problems
 - + Wide user base to build shelf life
- ✘ Available skill base
- ✘ Industry investment in technology base

COTS RISKS (1 OF 3)

- ✘ Is COTS the right choice for this program?
- ✘ Security capabilities vs COTS
 - + Information Assurance
 - + Software Assurance
 - + Accreditation/Certification
- ✘ User acceptance of Human/Machine I/F and procedural changes
 - + Has the user been involved, early and often?
 - + Has product training been accounted for?
- ✘ Product Evolution
 - + Product features change when and to what the Vendor chooses
 - + Insight into changes is dependent upon market space

COTS RISKS (2 OF 3)

- ✘ Multiple COTS product integration
 - + Common or compatible interfaces?
 - + Consistent vs inconsistent functionality
 - + Problem resolution amongst multiple vendors
 - + Requires skill base, training, personnel across products
- ✘ No/little insight into product
 - + Limited, poor, or no documentation (e.g. product flaws)
 - + No source code
 - + Unknown development processes or skills
- ✘ May not meet program requirements
 - + Product features not as advertised (more or fewer)
 - + Product not suited for intended operational use
 - + Difficult to find safety rated products (e.g. DO 178B)

COTS RISKS (3 OF 3)

- ✘ Underestimated total program costs
 - + Integration costs, Verification costs and O&M costs
- ✘ Risk to maintenance
 - + Unpredictable vendor support and vendor stability
 - + Dependency on vendor to identify flaws that are applicable to program
 - + Dependency on license agreements
 - + Vendor resistant to accepting/fixing externally identified flaws (requires “proof”)
 - + Product knowledge/skills needed over life of system
 - + Product lifetime may be less than program life (e.g. obsolescing)
- ✘ High probability of mods or ‘wrappers’
 - + Interfaces/protocol not standard with industry
 - + Unique operational environment

MITIGATION TECHNIQUES (1 OF 3)

- ✘ Gain Marketplace and vendor knowledge
 - + Shop early and often
- ✘ Gain product knowledge prior to baselining requirements
 - + Keep product training up-to-date
 - + Learn all you can, as early as you can, however you can
- ✘ COTS standards for program
 - + Stay aware of software portability, do not use COTS product extensions when programming
 - + Flagship product vs loosely coupled federation of products
- ✘ Use of alternate vendors
- ✘ Early vendor involvement throughout the life cycle
 - + Be nice to your vendors, it helps

MITIGATION TECHNIQUES (2 OF 3)

- ✘ Product and/or Vendor certification
- ✘ Flexibility in Requirements changes
- ✘ Early prototyping, allowing time for design/requirements changes
- ✘ Overall robust system design that can identify and withstand the unexpected
- ✘ License and support agreement negotiations
- ✘ Ensure Roles/Responsibilities are established for all of the above!!!
 - + Developers
 - + Vendors
 - + Government
 - + Others?

MITIGATION TECHNIQUES (3 OF 3)

- ✘ Source code escrow
- ✘ Up front systems engineering evaluations
 - + Complete 'make vs buy' trade studies, product suitability, etc
- ✘ Product 'insight' requirements
 - + Processes, skill base, tools, change history, etc
- ✘ Product simulators/models
- ✘ Early User involvement/assessments that include real end user and system support cadre

COTS EVALUATION CHECKLIST – STARTER SET

- ✘ Maturity of COTS marketplace in the domain
- ✘ Stable/quality vendors in this marketplace
- ✘ Quality products in this marketplace
- ✘ Similar usage of this COTS in related applications and environments
- ✘ Certifiable for Critical Applications
- ✘ Insights into development products and processes
- ✘ Fidelity of product simulations/models
- ✘ Product Change/Maintenance Plans
- ✘ Quality of and alternatives for product support
- ✘ Cost impacts for all the above

SOME 'OPEN PROBLEMS'

- ✘ Compatibility between the COTS product lifecycle and systems development/ops lifecycle
- ✘ Effective cost estimation algorithms
- ✘ Multiple vendor/product integration

SUMMARY

- ✘ COTS is just another candidate solution
 - + Know when to use, not use
- ✘ Recognize system elements that may be ‘volatile’ or ‘weak’, make final decisions as late as possible
- ✘ Establish acceptance criteria, early!
 - + Review acceptance criteria, often!
- ✘ COTS vs other NDI options (e.g. OSS)
- ✘ Develop a COTS Management Plan
 - ✘ For Acquirer and Developer!!

REFERENCES

- ✘ SEI COTS-Based Initiative products
 - + <http://www.sei.cmu.edu/cbs/>
- ✘ AIAA Guidebook, “Managing Use of COTS in Mission Critical Systems”
 - + <http://aiaa.org/content.cfm?pageid=178>
- ✘ “The Use of Commercial Software in Ground Systems Development”, R. Adams, Suellen Eslinger, Aerospace Crosslink
 - + <http://www.aero.org/publications/crosslink/spring2006/03.html>
- ✘ “Sustaining Software – Intensive Systems”, CMU/SEI-2006-TN-007, Mary Ann Lapham, Contributor: Carol Woody,
www.sei.cmu.edu/publications/documents/06.reports/06tn007.html
- ✘ ICCBSS (major COTS conference)

QUESTIONS?

- ✘ Ron Kohl, R. J. Kohl & Assoc.
 - + 301-874-3509
 - + rjkohl@prodigy.net