



True Program Success™



The Real Costs of Developing COTS Software

Arlene F. Minkiewicz
PRICE Systems, L.L.C.
17000 Commerce Pkwy, Suite A,
Mt Laurel, NJ 08054
arlene.minkiewicz@pricesystems.com

October 2006



Conclusion

- > **COTS solutions can save time and money in the development and life-cycle phases of a software product.**
- > **Care should be taken to:**
 - Select COTS solutions carefully
 - Understand all of the activities and the respective cost implications
 - Look at the long term & short term cost implications



The Real Costs of COTS Software

1. The Problem
2. Solution Methodology
3. Development Activities for COTS Based Systems
4. Sustaining Activities for Maintaining COTS Based Systems
5. Other Costs Associated with COTS
6. Conclusions

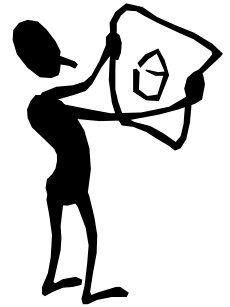


The Problem

- > COTS – Not always the low cost solution
- > Need to understand the activities required to develop and maintain COTS based software systems and cost drivers of those activities



Solution Methodology



- > Understand and bound the problem
- > Understand the process of including COTS and identify the activities associated with this process
- > Identify the factors that drive costs for these activities
- > Construct a mathematical model to determine cost from these cost drivers
- > Test the mathematical model against actual data and refine the model



Bounding the Problem

- > **Extended definition of COTS product (from USC CSE) to include modifications**
 - Commercially available software product - sold, leased or licensed
 - Source code sometimes unavailable
 - Periodic release with new features, upgrades for technology, etc.
 - Modifications to software

- > **Focus is on COTS products being embedded in new software systems**

- > **Some success extended this solution to enterprise software integrations**



Development Activities

For a COTS-Based Software System

- 1. Analyze software requirements**
- 2. Identify, evaluate and select COTS components**
- 3. Purchase / Lease / License the COTS components**
- 4. Tailor COTS software**
- 5. Design, code, and test glue code and modifications**
- 6. System level integration and test**
- 7. Evaluate and integrate upgrades**
- 8. Fix bugs**



Analyze Software Requirements

- > **Necessary whether software is being built or bought**
 - In fact part of the requirements discussion should be whether building or buying makes sense
- > **Selection criteria should relate back to requirements**
- > **Care should be taken to identify where there is flexibility – as no COTS solution will meet all software requirements completely**
- > **Cost drivers are:**
 - How much functionality?
 - What kind of functionality?
 - Where will the software operate?
 - Other project constraints

Identify, Evaluate, and Select

- > **Identify solutions that satisfy product and vendor requirements**
- > **Techniques for evaluation include:**
 - Progressive filtering
 - Puzzle approach
 - Keystone components
- > **Cost drivers are:**
 - How many functions replaced by COTS?
 - How many different COTS solutions considered?
 - Type(s) of evaluation taking place
 - Number and criticality of evaluation criteria



Tailoring

> Tailoring activities include:

- Database and parameter initialization
- Registration with operating system
- Screen and report scripting
- Security and access control

> Cost drivers are:

- How much functionality?
- What kind of functionality?
- Quantification of complexity to learn and use COTS solution
- Quality of user documentation, vendor support and training
- Presence of security and access control issues



Develop Glue Code & Make Modifications

- > **Glue code holds the system together**
 - References interfaces and interprets return codes
 - Converts data to format required by COTS
 - Makes up for short comings in COTS delivered functionality
- > **Modifications are sometimes necessary especially if it GOTS not COTS**
- > **Cost drivers for glue code development and modifications are the same as for any software development with adjustments for**
 - Quality of documentation, training and support
 - Experience of integration team with product and COTS product
 - COTS product maturity and volatility
 - Vendor cooperation





System Level Integration & Test

- > **System level testing complicated by unfamiliarity with COTS and unavailability of source code**
- > **Requirements relating to performance, reliability and security could be problematic**
- > **Cost drivers are:**
 - Total functionality being integrated
 - Quality, performance and reliability requirements
 - Integration team productivity
 - Integration difficulty
 - Number of COTS components in system



Evaluate Upgrades

- > **Upgrades are often required to fix bugs, accommodate changes in technology, make up for short comings in COTS product(s) selected**

- > **Cost drivers are:**
 - Measure of functionality (function points, tailoring requirements)
 - Vendor stability
 - Product stability and quality
 - Requirements stability
 - Upgrade cycle of COTS product compared to length of development project



Activities for Maintaining and Growing COTS-Based Systems

- > **Maintain license, subscription and royalty fees**
 - Cost drivers are the fees
- > **Evaluate and include upgrades**
 - Cost drivers the same as for evaluation upgrade during development
- > **Fix bugs associated with defects due to tailoring, glue code or delivered with the COTS components themselves**





Fix Bugs

- > **Repairs related to COTS software are often problematic due to unfamiliarity and unavailable source code**

- > **Cost drivers for repair activity costs include:**
 - Size of functionality utilized (function points or tailoring requirements)
 - Size of glue code developed
 - Quality of COTS product
 - Experience and productivity of maintenance team with product and COTS process
 - Vendor responsiveness

Other COTS Costs

- > **Costs to keep code in escrow**
- > **Substantial changes in license and service fees due to change of vendor or changes in the market place**





Conclusion

- > **COTS solutions can save time and money in the development and life-cycle phases of a software product**
- > **Care should be taken to:**
 - Select COTS solutions carefully
 - Understand all of the activities and the respective cost implications
 - Look at the long term and short term cost implications

Arlene F. Minkiewicz

Chief Scientist

PRICE Systems, L.L.C

856-608-7222

Arlene.minkiewicz@pricesystems.com

About PRICE Systems

- > Leader in Program Affordability Management solutions
- > Combine cost estimating, project control, and knowledge management – *ensuring project success at every decision gateway*
- > Customers increase visibility, minimize risk and cost, accelerate project development, and improve the effectiveness of project selection, control and delivery



POWERED BY
PRICE Research

www.PRICESystems.com