NRO Industry Advisory Working Group

COTS/GOTS/FOSS Business Model Action Teams Report

October 1, 2017
COTS/GOTS/FOSS Open Source Software Business Models
Implementation & Action Team Map

Business Model Differences

Drivers: Make-Buy-Integrate Decisions

Terms of Reference

Software Business 101

Integrate/Hybrid?

SW Licensing Decision Framework

COTS?

Body of Knowledge and Industry/Government Dialog Advance
Adoption of ISP-ASP Vision
# COTS/OSS/GOTS Business Model

## Action Team Participants

<table>
<thead>
<tr>
<th>Business Models Team</th>
<th>Terms of Reference</th>
<th>Make-Buy Drivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Hays</td>
<td>John Farrell</td>
<td>Nick Buck</td>
</tr>
<tr>
<td>Mike Miller</td>
<td>Keith Barber</td>
<td>David Beddoe</td>
</tr>
<tr>
<td>Suzanne Sincavage</td>
<td>Ben Avicolli</td>
<td>John Amazigo</td>
</tr>
<tr>
<td>Marc Kriz</td>
<td>Jim Metsala</td>
<td>Andy Cibula</td>
</tr>
<tr>
<td>Nick Buck</td>
<td>Pam Arya</td>
<td>John Farrell</td>
</tr>
<tr>
<td>John Farrell</td>
<td>Matt Mattigan</td>
<td>Animesh Gupta</td>
</tr>
<tr>
<td>Keith Barber</td>
<td>Andy Murren</td>
<td>Doug Harts</td>
</tr>
<tr>
<td>Shannon Silverstein</td>
<td></td>
<td>Ken Kincel</td>
</tr>
<tr>
<td>Scott Thomas</td>
<td></td>
<td>Marc Kriz</td>
</tr>
<tr>
<td>Animesh Gupta</td>
<td></td>
<td>Mike Manning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jim Metsala</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jared Putman</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Allen Shepherd</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Leigh Thompson</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shannon Silverstein</td>
</tr>
</tbody>
</table>

## Terms of Reference

- John Farrell
- Keith Barber
- Ben Avicolli
- Jim Metsala
- Pam Arya
- Matt Mattigan
- Andy Murren
- Nick Buck
- David Beddoe
- John Amazigo
- Andy Cibula
- John Farrell
- Animesh Gupta
- Doug Harts
- Ken Kincel
- Marc Kriz
- Mike Manning
- Jim Metsala
- Jared Putman
- Allen Shepherd
- Leigh Thompson
- Shannon Silverstein
- Steve Thomas
- Keith Barber
Body of Knowledge and Industry/Government Dialog Advance Adoption of ISP-ASP Vision
Terms of Reference Action Team

Participants & Objectives

TOPICS

• Misunderstandings: policy, definitions of OSS/COTS (what it is, what it’s not), who uses it?
• Miscommunications: “Open Software” vs “Open APIs” vs “Proprietary” vs “Closed”
• Government PM biases against paying for commercial licenses

OBJECTIVES Completed: Create clarity & level set
– Aggregated a list of terms
– Defined the terms based on a set of known references
– Voted on the definitions OR provided multiple definitions
– Identified areas of confusion:
  • Misunderstandings: policy, definitions of OSS/COTS (what it is, what it’s not), who uses it?
  • Miscommunications: “Open Software” vs “Open APIs” vs “Proprietary” vs “Closed”
  • Government PM biases against buying commercial licenses

http://usgif.org/system/uploads/4990/original/Terms-of-Reference-v1_0_4-20170410.pdf
Terms of reference: GOTS/OSS/COTS

- **GOTS**
  - Government off-the-shelf (GOTS) is a term for software and hardware government products that are ready to use. They were created and are owned by the government.

- **OSS**
  - Open Source Software is software for which the human-readable source code is available for us, study, reuse, modification, enhancement, and redistribution by users of that software.

- **COTS**
  - Clinical: Commercial Off-The-Shelf (COTS) is a term that references non-developmental items (NDI) sold in the commercial marketplace and used or obtained through government contracts. The set of rules for COTS is defined by the Federal Acquisition Regulation (FAR).

- **Hybrid**
  - Applying COTS/OSS as available to meet functionality with GOTS to integrate, extend or augment mission specific functionality
  - Integrating vetted open source to meet specific market or customer unique requirements development
  - Contribute to OSS community projects
Simplified Taxonomy of Software Types

Applications – Software designed to perform a variety of functions

Apps – Software designed for a single purpose and performs a single function

Infrastructure – Software used as a platform for applications (e.g., OS, DB)

Management Tools – Software designed to manage and secure infrastructure

DEVOPS – Software tools used to create software, deploy and evaluate performance
Open Source and Open Standards: Procurement-Related “Facts”...

- Open Standards ≠ Open Source ≠ Open Data
- Open Source is used extensively by COTS and GOTS developers alike
- It’s not Proprietary Code that cause vendor lock in, it’s Proprietary Interfaces
- Vernacular (terminology) really matters, especially in procurements
- GOTS is a custom software baseline, even if it uses OSS components
- Open Source isn’t free. Must consider Total Cost of Ownership

GOTS vs COTS vs FOSS is a Multi-Factor “Make-Buy” decision
COTS/GOTS/FOSS Software Business Models
Implementation & Action Team Map

Body of Knowledge and Industry/Government Dialog Advance
Adoption of ISP-ASP Vision
COTS/OSS/GOTS Business Model Team

Topical Activities

TOPICS

- Warranty, Licensing & Liability
- Version control & certification
- CM, modifications/updates
- Buy vs Build decisions
- Business models (capital investment vs labor/ FTE model)
- Metrics & cost realism

- Characterize different business models for developing, deploying, integrating and maintaining OSS, COTS and GOTS
  - Funding source models
  - How are they acquired?
  - Life cycle costs and cost recovery
  - Intellectual property influences on business model
  - Maintenance and licensing, royalties
  - Protections and indemnification
  - Retirement and refresh
  - Identify Representative models
  - Risk models
**COTS Investment Model**

**Supplier Bears Development Cost, Amortizes Maintenance Cost**

- Large Initial investment Against future sales
- Government License subscriptions on existing product
- Predictable cost Limited ownership
- Larger than expected maintenance may add to subscription costs
- Government License subscriptions on new product
- Reinvest and Refresh and New Product development
- 1-3 yrs to return

- Government coverage of investment costs
- Maintenance costs
- Marketing, Training, Enablement, Services
- 20-50% OSS reduce dev costs
- OSS maintenance & participation

~ 5 yrs to return
COTS and COTS/OSS

Funding Sources
- Venture capital or startup funding, research grants, InQTel, academic partnerships
  - Business case / pricing models based on expected revenues over market and its risk
  - Operational income (license fees, etc.) reinvested in new product development
- Congressional mandates, presidential directives, agency sponsored programs
- Government funding for specific application with IP ownership and rights to resell maintained by supplier

How are they acquired?
- Commercial purchase, GSA (limited usage), and procurement
- Integration of OSS (20-50%) use of available components and open standards
- Contracting vehicles often not well suited for COTS, COTS/OSS business and licensing models.
  - Funding cycle not tuned for rapid response
    - Government has flexibility here, not often exercised
    - Uneven adoption of commercial practices
  - GSA can be helpful, used less than expected
    - Internal investment to maintain GSA schedule
    - Small companies have less access to GSA
    - Some GSA Schedule holders may provide GSA discounts to select customers
  - Opportunity to leverage SEWP
COTS and COTS/OSS (Cont’d)

- Maintenance and updates
  - Trend is to move to subscription model instead of perpetual, with no in-year pricing adjustments
  - All updates and support available as long as subscription license is current.
  - No change order required
  - Options: Move from product focus to trusted vendor focus

- Licensing, liability and warranties
  - Evaluate licensing terms: Trend away from perpetual licensing to term subscription allows for Government flexibility and manages risk
  - Yearly subscription terms works well for some suppliers and sponsors
  - Cost based on risk of amortization across expected subscription renewals.
  - Indemnification and protections are additional costs carried by supplier

- Cost recovery
  - Licensing fees, services labor, installation labor
  - Speed of delivery and incentives drive margins
  - Emerging options: maintain as open source, monetize the data instead of the tools
  - Free license in exchange for services, cover maintenance costs and not investment costs
    - Small margins on services extends investment cost recovery and delays new product development
  - Pay-per use such as node or server pricing
GOTS Investment Model

Gov’t Bears Development & Life Cycle Maintenance Cost

GOTS model

Initial Development Costs

Leverage OSS reduces development costs

Maintenance plateau

Retire maintenance to support new refresh and reduce maintenance costs

Mission unique with OSS integration
Hardening

OSS reduces dev costs and schedule

OSS maintenance & participation, version control

Gov’t Bears Development & Life Cycle Maintenance Cost
GOTS and GOTS/OSS

Funding Sources
- Government funding for specific application with unlimited rights desired by Government.
- Collaborative effort for cost assessment and payment methods (payment methods, usage, development, infrastructure, etc.)
- GOTS generally fits into services (RDT&E). O&M to maintain only after development
- Suppliers may invest IR&D toward Government relevant general market areas

How are they acquired?
- Maintains a national security advantage
- No equivalent COTS solution is available
- Key differentiator is defined in what is unique vs. commodity
- Procurement (RFP, BAA, SEWP, sole source justification etc.), funding vehicle
- Can often be supported by either development effort or service model
  - Services vs. Software (different budgets)
- Integration of OSS with increasing availability of components and open standards
  - Advantage cost and time, access may limit availability of OSS in secure environments
  - Preapproved software (versioning) impacts schedules and vulnerabilities

Note: OSS is defined by DoD as COTS, included here when used with GOTS, as it is acquired by a different model.
GOTS and GOTS/OSS (Cont’d)

- Maintenance and updates
  - Services and personnel, counted in cost for GOTS//OSS
  - COTS accounted for in license fees or FFP Installation and start up services
  - Maintain OSS and participant in upgraded in the OSS framework
    (sometimes not considered in overall cost)
  - Parallel applied to hardware as well
  - Options: service contracts to suppliers vs. Government in-house maintenance

- Risks
  - Manage requirements and performance against requirements.
  - Vendor or integrator takes on risk if not fully tested.

- Cost/Payments
  - Delivery terms
  - Contracting labor and service basis
Consider a comparison of program costs for a core development of $20M of code – For purpose of comparisons assumptions are:

- **GOTS development model** - 5 year development with 5 year overlapping operations and maintenance (O&M)
  - KTR:: core code development, SEIT, and program costs
  - Gov – PMO, SETA and Integration staff
  - Gov and KTR O&M

- **COTS model** (little code modification is required)
  - COTS vendor investment creates and services and maintains the core product
  - Government cost to assess requirement, and SI for I&T and maintenance support and licenses
  - Licenses – assumes Vendor costs are amortized over 40 enterprise users over 9 years

- **Hybrid model** (805 COTS and 20% GOTS for modification of code for mission requirements)
Notional Comparison of Government Costs
GOTS vs COTS Life Cycle

Reality: there is wide use of Open Source components in both COTS & GOTS

Program cost GOTS - $107M
Program cost COTS - $63M
Program Costs Hybrid - $80M

Equivalent Vendor investment

Hybrid reduces new investment & adds mission requirements

GOTS model detail

Hybrid O&M roughly equivalent

Cumulative cost $107M

Source: US Geospatial Intelligence Foundation NRO Industry Advisory Group
Comparing GOTS / Open Source to COTS

Compare Apples to Apples

• COTS
  - Amortizes the cost of development across multiple organizations
  - Value exceeds any one user
  - Research and Development is a reaction from multiple users (Diversity of requirements)
  - Typically leverages OSS for subcomponents…represents a commercial component integration

• OSS
  - Voluntary efforts of developers
  - “No one works for free” i.e., “bulk of the work that is developed is on those features desired by the top few largest users who pay their developers to build the software to their requirements. In other words, the future development ’road map’ is anonymous and collective.”
  - Security ramifications fall to using organization

• COTS costs represent:
  - Research & Tech Roadmap;
  - Development Processes, Protocols, Standards;
  - Robust Testing, Support, Compliance Testing/Evaluations
  - Security to include IAVA Patching

1 Bob Palmer SAP National Security Services, Senior Director, Solution Marketing
Typical Trade-off dialogs
GOTS vs COTS

• Trade Space
  – Lifecycle Cost (TCO)
  – Scalability
  – Extensibility
  – Support
  – Reliability
  – Roadmap
  – Security (where is it developed; who is funding)

• Typical dialog when COTS Functionality < Required Functionality
  – if COTS “only” satisfies 85% of Requirements then Gov’t Builds, or
  – If OSS satisfies 85% of the requirements then “good enough” (We don’t need 100% solution)
  – Challenge: Understand commercial vendor’s ability to address Gov’t requirements

• Challenge When COTS Functionality = GOTS Functionality
  – GOTS represents a custom software development with life cycle cost impact and no other “committers”, i.e. “developer lock-in”
  – Complexity of “gluing OSS together” i.e., communication channels = N(n-1)/2
  – Represents potential competition between Government and Commercial industry

Emergence of “Open API” and “SDKs” changes the game to COTS + GOTS instead of COTS vs GOTS
Observations

- Contracting approaches can be a barrier to technology adoption in the IC
  - Holdover from legacy GOTS model can impact COTS viability
  - Delays reinvestment if new products
  - Viability of technology leadership and investment
  - Wrong color of money and no will to trade for another color
  - Easier to put development work on a services contract than procure a product

- Acquisition practices for GOTS and COTS differ
  - Milestone payments for GOTS development are generally acceptable, not so for COTS
  - Licensing and service payments are completely different
COTS/GOTS OPEN SOURCE SOFTWARE BUSINESS MODELS

Implementation & Action Team Map

Body of Knowledge and Industry/Government Dialog Advance Adoption of ISP-ASP Vision
Make-Buy Drivers Action Team: Deliverable

- **Goal:** Lift barriers to mission execution by providing guidelines to the two sides of the “agency” business: Mission & Acquisitions - working in concert and industry can serve more effectively

- **Deliverable:** Develop make/buy/integrate decision “best-practices”
  - Before RFP and/or After Award – address the continuum
  - “Something between an educational tool on decision factors and a decision matrix.”
  - Product or Compendium of Ideas:
    - As a Guidebook or/and
    - As a Recipe (step by step) or/and
    - As a Wiki (allowing community input) or/and
    - As a Decision Matrix (multiple?) or/and
    - As a TOP 10 or/and
    - As an FAQ (Fact vs Fiction) [this would be a good Appendix]
  - Potential inclusion in the NRO Acquisition Manual
  - As an IC Directive or an NRO Directive
COTS-GOTS Make-Buy: Factors to Consider

OPERATING MODEL FACTORS
- Acquisition Strategy
  - ABC (Adopt, Buy, Create) vs CBA?
  - “Speed to Need” (Day 1 vs Year 1 capability)
  - Incremental & Agile approaches (esp. COTS)
- Ease of development with Service contracts
  - Measures of Success
  - QA/QC Independent
  - Ownership – Make/Buy/Integrate components
- Procurement Factors
  - Complexity of NDI Product procurement
  - Different pots of money: Dev vs O&M?
  - Licensing (OSS contribution?/Middleware)
  - Warranties, Indemnification, Cyber liability

ARCHITECTURAL FACTORS
- Open Architecture
  - Level of componentization
  - Degree of API publication
  - Open I/Fs vs Open Source code

BUSINESS MODEL FACTORS
- Technology cycle compression
- Funding models (government & industry)
- Services vs. Licenses:
  - Obs & expenditures (Govt)
  - Revenue recognition (Industry)
- TCO (Govt) and Cost recovery (Industry)
- Intellectual property influences
- Maintenance and licensing, royalties
- Protections and indemnification
- Retirement and refresh
- Risk models

ARCHITECTURAL FACTORS
- Requirements
  - 80-90% fit vs 100% fit vs not avail in marketplace
  - Outcome based vs Activity based
  - Tech Readiness/Product Maturity
### Risk Comparison – COTS/GOTS/Hybrid Models

**Is Hybrid the “Best of Both Worlds”?**

<table>
<thead>
<tr>
<th>RISK AREA</th>
<th>COTS</th>
<th>GOTS</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Development</strong></td>
<td>• Developed in anticipation of market need</td>
<td>• Requirements must be well defined upfront to control risk. • Purpose built, custom coding.</td>
<td>• Lowest baseline risk. Development risk focused on extending COTS with new GOTS functionality.</td>
</tr>
<tr>
<td><strong>Integration</strong></td>
<td>• Dependent on API availability • May require wrapping • Legacy system complexity</td>
<td>• Purpose built integration • Code can be modified if req’d • External Interface complexity</td>
<td>• COTS integration points allow use as open platform • Minimize custom integration</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>• Market driven performance • Unique mission configuration or adaption to legacy may be required.</td>
<td>• Requirements driven testing and performance. “Have it your way.”</td>
<td>• Requirements driven performance &amp; testing PLUS widely available components</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>• Cost amortized over larger expected market.</td>
<td>• Labor throughout lifecycle. Cost from requirements, testing and service rates.</td>
<td>• Minimize direct labor TCO. • Leverage available COTS • Maintain OSS</td>
</tr>
<tr>
<td><strong>Schedule</strong></td>
<td>• Available off the shelf with min modification. May require labor to configure or adapt.</td>
<td>• Minimum 6 months to capability. Timeline driven by requirements, testing and available services.</td>
<td>• “Day 1” capability + Requirements, testing and available services driven</td>
</tr>
<tr>
<td><strong>Maintenance</strong></td>
<td>• Services and version control per licensing parameters.</td>
<td>• Maintenance services. Version Control and promotion to operations. • Sustaining outage</td>
<td>• GOTS &amp; COTS delineated terms • Managed via maintenance model</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>• Indemnification, warranties, IA risks/costs carried by supplier.</td>
<td>• Supports unique security needs. • Requires labor to maintain.</td>
<td>• Indemnifications and protections, w/support to unique security needs.</td>
</tr>
</tbody>
</table>
Potential COTS-GOTS Acquisition Models

COTS
- Govt buys 100% functionality, but it addresses less than 100% of reqts
- Govt request new requirement
- Vendor assesses market ROI and pays to develop “100%” solution, OR Govt defers requirement and waits for Vendor roadmap

“Accelerated” COTS
- Govt buys 100% functionality, but it addresses less than 100% of reqts
- Govt identifies “delta” requirements
- If no COTS Market ROI, Govt pays vendor for accelerating capability
- If potential Market ROI, Govt/Vendor cost share accelerating capability

COTS-GOTS “Hybrid”
- Govt buys COTS but it meets less than 100% of requirement
- Govt assigns “delta” reqts to integrator
- Integrator extends COTS via API/SDK

GOTS
- Govt builds 100%
- Govt assigns reqts to integrator
- Govt self-indemnifies & ensures compliance with Economies Act
- Vendor-free supply chain

Assumptions: both COTS/GOTS providers use open source
IDEAS for Further Discussion

- Government can change the model: shift toward buy vs build
  - Identify root cause: proprietary interfaces are the culprit in vendor lock in…not proprietary code!
  - Provide training for government, FFRDC, and SETA on COTS Integration vs GOTS Development

- Explore concept of an ODC line for licenses in RFPs: level playing field for all and avoids biasing against off the shelf software due to BOM impacts to bid price

- Have every contract award include requirement for prime to complete a market survey and make/buy assessment for government approval
  - Include life cycle costs and the various factors identified by the IAWG COTS/GOTS/Hybrid matrix
  - Market survey…Life Cycle Costs….Risk Factors
  - CO/COTR/PM emphasis on Market Research beyond RFI to include industry visits, symposia, etc.

- Recognize make-buy tradespace as a requirements trade space…government should include ALL requirements in RFPs.
  - Stop allowing programs to cherry pick the requirements in order to justify a “Create” (vs “Adopt”/”Buy”)
  - Require programs to determine what portion of the requirements have viable commercial solutions prior to deciding on the acquisition/procurement strategy.

- Consider an “outcome based” requirements model (additional dialog needed)

- Encourage Programs to consider incremental capability approach
  - Stress “Day 1” capability as high value to support “speed to need”
  - If there is an incremental approach does it change the make-buy?