



GSAW 2002

Integration:  
Reinventing an *Expensive*  
Wheel

Larry McQueary (larry@talarian.com)

Talarian Corporation

14 March 2002

<http://www.talarian.com>

**Talarian**<sup>TM</sup>  
The world works in real-time<sup>TM</sup>

# Introduction

- **Talarian Background**
- **Infrastructure Software**
- **COTS Best Practices and Benefits**
- **Commonly heard COTS Observations**
- **Top Issues: Resistance to COTS Software Use**
- **Top Issues: Vendor-Client Communication**
- **Summary and Suggestions**
- **Q&A**

# A Vendor Perspective: Talarian

## ● Talarian Background

- ◆ **Founded 1989, from Lockheed Martin roots**
- ◆ **1989 – 1995**
  - Specializing in infrastructure building blocks for command and control systems (RTworks)
- ◆ **1996 – Present**
  - Generic infrastructure solutions, aerospace, financial, telecom, OEM

## ● Talarian in Aerospace

- ◆ **Command and control of NASA's Hubble Space Telescope**
- ◆ **Real-time monitoring of data for Space Shuttle**
- ◆ **Raytheon ECLIPSE**
- ◆ **Aerospace STARS**
- ◆ **SMC Det 12: RSC/CERES/"COBRA"**

# Infrastructure Software: A Common Build vs. Buy Decision

- **Every ground system depends on common infrastructure software**
  - ◆ The fundamental framework or glue that provides inter-process communication and extensibility
  - ◆ The integrator's toolkit and information bus
- **The most:**
  - ◆ Overlooked/underestimated
  - ◆ Complex
  - ◆ *Expensive!*
- **Conflicting directives can drive costs up**
  - ◆ Not just buy vs. build, but re-use/rebuild vs. buy

# COTS Best Practices and Benefits

- **Reduce lifecycle costs and system complexity by relying on native features of COTS products.**
- **Reduce lifecycle costs by:**
  - ◆ **Relying on vendor/supplier maintenance support for COTS products.**
  - ◆ **Adhering to applicable (and sensible) design and implementation standards**
    - POSIX, ANSI, IEEE, OMG, etc.
  - ◆ **Provide abstractions for replaceable components (i.e. “wrappers”)**
  - ◆ **Small, specialized, loosely coupled components**
  - ◆ **Focus on your team’s core competencies, not problems that have been solved by experts outside your domain (e.g. middleware)**
- **Distributed Systems**
  - ◆ **Model the architecture on events, not the GUI/HMI**
  - ◆ **Distributed message- and event-driven processing**

# Most Commonly Heard COTS Software Observations

- **Inter-product/platform version incompatibility**
  - ◆ (Too) frequent version releases
  - ◆ Changing COTS vendor priorities
- **50% of features go unused**
  - ◆ V&V concerns
  - ◆ “Build the piece you need vs. buying more than you need”
- **Some integrations too tightly coupled with COTS**

# Top Issues We See: Resistance to COTS Software

- **General: “Space is really hard”**
  - ◆ Integration issues are riskier and more complex in aerospace than in most other industries
  - ◆ COTS? We don't need no stinkin' COTS! 😊
- **Engineering: “Programmers like to program”**
  - ◆ Using COTS is not an intuitive choice for anyone educated prior to the 1990s
- **Management: Conflict of Interest**
  - ◆ Reduction of programmer hours is almost anathema to program managers

# Top Issues We See: Vendor-Client Communication

## ● Polarized Attitudes

- ◆ One extreme: Customers believe that vendors are looking to make you dependent on them, while turning a quick buck in the short term
- ◆ The other: Vendors believe that customers want to pay next to nothing, develop in a vacuum, and leave a bullseye on the vendor's back

## ● How do you ensure success?

- ◆ Open, honest, and early discussion and agreement of requirements between vendor and client
- ◆ Pick vendors that will be your *partners*

## ● Success is a two-way street

- ◆ If the customer isn't successful, the vendor has a short lifespan
- ◆ No one wins, least of all the vendor, if the vendor sells you something you don't need

# Summary and Suggestions

- **“Don’t worry, be happy!”**
  - ◆ **Despite discussion to the contrary, using COTS is still cheaper and more effective than building**
  - ◆ **A large percentage of home-grown software projects that re-invent complex COTS functionality will fail.**
    - **Analyst studies (Tower, Gartner) of real world projects have confirmed this notion time and again**
- **Understand the risks and weight them by probability of occurrence**
  - ◆ **Again, communication of lifecycle issues w/vendor is key**
- **Pick vendors who understand the issues and be willing to make commitments up front in return for commitments from the vendor**
  - ◆ **Training, premium support**

# Q&A

- **No tomatoes, please!**