

COTS and Safety – Are They Mutually Exclusive?

Warren P. Naylor; System Safety Manager;
BAE SYSTEMS
Rockville, MD

Ron Stroup; Certification and Safety Lead
Federal Aviation Administration;
Washington, DC

Keywords:

COTS
PDS
NDI
Assurance
Safety
Obsolescence

Abstract

The lifecycle of Commercial Off The Shelf (COTS) products are limited and lifetime buys are no longer permitted. This policy and the reality of COTS have placed the safety and certification communities in a difficult position. How can we certify a mission/safety critical system when the very foundation, the microprocessors and OS's are castles built in sand?

Economic pressures and the much larger market place drive COTS products. The Government is no longer the leader or even a trendsetter in the market place. The Government has taken the position of Better, Faster, Cheaper and has identified COTS as the vehicle towards that end. Reality has not matched expectations, however, that has not dampened the initiative.

The objective of this paper is to provide the readers with the latest guidance available primarily provided by a joint RTCA Special Committee 190 and EUROCAE Working Group 52. This paper will propose methods of implementing COTS in hopes of positively impacting the safety and ultimately the success of your programs.

Introduction

Initially, the introduction of COTS into safety critical environments caught the safety community by surprise. The safety community raised objections and provided warnings and cautions about the use of COTS in safety critical systems. However, like most warnings when not

backed by empirical data, these warnings went largely unheeded, and the COTS revolution has proceeded with vigor.

The COTS movement is primarily driven by cost, schedule, the immediate need to replace aging systems to meet the evolving needs of the new millennium and the goal of keeping abreast of the emerging technologies. The goals have largely remained intact, even with the current heightened awareness of the COTS Issues.

It is the safety community's responsibility to take a proactive leadership role in mitigating the risk of COTS. We cannot, as a community, only present concerns and objections; we must also suggest solutions and alternatives. Hopefully, this paper will stimulate the identification of additional alternatives, methods, and solutions that we all can benefit from.

Note: Large portions of this paper were derived from the work of the RTCA SC-190 to which both authors are members. This paper is designed as more of an infomercial than a replacement for the CNS/ATM Guidelines for Software Assurance which should be published late this calendar year and certainly contains substantive information not provided in this paper. It is highly recommended that all readers of this paper purchase the Guidelines when they become available.

A Definition of COTS

The following definition of COTS is provided for the sole purpose of understanding the intent and scope of this paper. It is not meant to be 'the' definition, although we would argue that when 'the' definition is finally prescribed that it should possess the artifacts provided in our definition:

COTS products encompass a wide variety of general-purpose off-the-shelf products, Non

Developmental Items (NDI) and Previously Developed Software (PDS).

Note: Some of these products are designed to be user selectable/modifiable (e.g., a compiler). Vendor supplied modifications or selectables are still considered COTS. However, it must be understood that once a program modifies or enhances COTS software to meet their respective system requirements, than the modified COTS must then be considered application code, subject to all certification requirements without exception.

Examples of COTS software are operating systems, real-time kernels, graphical user interfaces, communication and telecommunication protocols, language run-time libraries, mathematical and low-level bit and string manipulation routines, etc. COTS software can be purchased apart from or in conjunction with COTS hardware such as workstations, mainframes, communication/network equipment or hardware items (memory, storage, I/O devices, etc.). There also may be some instances where the use of COTS software is impractical to avoid, e.g. library code associated with certain compilers.

COTS deliverables vary by the contract with the COTS supplier. They may extend from license rights, executable code, user documentation and training to the full set of COTS software lifecycle data including the source code as resulting from the COTS software development. COTS information disclosure relates to cost, protection of intellectual properties and legal questions such as ownership of the software, patents, liability and documentation responsibility. Those aspects are beyond the scope of this paper, as only those aspects that are specific impact safety will be addressed.

COTS Issues

COTS issues have become fairly well known over the past several years, however, for reader ease and understanding a brief itemized list is provided. The focus of this paper is on mitigating these issues rather than reemphasizing them. The most commonly recognized issues are:

- Obsolescence
- Maturation of product
- Version control
- Undisclosed issues/problems

- Vendor support
- Absence of available COTS data (e.g., source code, validation data, etc)
- Testing issues (regression testing of new upgrades)
- Robustness of Vendor's testing unknown
- Vendor's developmental processes unknown
- Structural coverage
- Selection/acquisition of the best COTS product
- Maintenance
- Training
- Security.

Development processes used by COTS suppliers and procurement processes applied by acquirers may not be equivalent to processes used within the safety critical industries (e.g., Aviation, Department of Defense).

The use of COTS often requires that alternate methods be used to gain assurance to the systems predefined acceptable residual risk levels are met. These methods include product service history, prior assurance, process recognition, reverse engineering, restriction of functionality, formal methods, audits and inspections. Data may or should also be combined from more than one method to gain assurance data or an acceptable level of confidence is met.

It should be noted that alternate methods are not the prescribed solution; they are what they are called, alternate methods, only to be used when acceptable safety/certification data is unobtainable from the COTS vendors and cannot be produced by the developer.

System Aspects Relating to COTS

Necessity may dictate integration of COTS software into high integrity mission/safety critical systems or equipment. The higher the criticality (FAA) or severity (DoD) level of a system, the more demanding the assurance requirements are for the system and the software. Risk mitigation techniques must be considered to reduce the targeted system's reliance on the COTS. The goal of these mitigation techniques is to accommodate the required criticality/severity level by reducing the effect of anomalous behaviour of COTS on the system functionality.

COTS Planning Process

The purpose of the COTS Planning Process is to strategically plan for, coordinate, and ensure the COTS lifecycle issues are adequately addressed.

It is highly recommended that a Computer Resources Lifecycle Management Plan (CRLCMP) or what we propose to call a Strategic Lifecycle Technology Refresh Plan be developed and approved by the appropriate authority prior to proceeding down the path of a COTS based system.

Defining a COTS Process

The current ad hoc informal implementation of COTS into potentially hazardous systems cannot continue. Cost, schedule and safety are the victims of this ad hoc process. Our current methods or lack thereof is a kin to taking a stroll in a minefield. We may reach the other side without incident; however, the odds are certainly not acceptable.

Developmental processes used by the various COTS suppliers are most likely unacceptable to the mission/safety critical safety and certification communities. Alternate methods are used for the certification of these systems. These alternate methods are used to augment the certification and safety data provided by the COTS suppliers to give the certification authorities more confidence that the system will indeed operate safely, as required.

The successful implementation of COTS products into safety critical systems requires a formal standard process. The need to define a COTS Process is long overdue. The COTS Process consists of Planning, Assessment, and Verification.

COTS Planning Activities

The following are the activities a program should attempt to successfully achieve and document within a Strategic Technology Refresh Plan:

- a. COTS planning activities should include the following considerations:
 1. Product availability
 2. Requirements
 3. Availability of lifecycle data
 4. Ease of integration and extent of additional efforts such as glue code,

architecture mitigation techniques etc.

5. Product Service history
6. Supplier qualifications such as use of standards, history and length of service, technical support etc.
7. Configuration control including visibility into COTS supplier's product version
8. Modified COTS have additional considerations of warranty, authority to modify, continued technical support, etc., unless such modifications are allowed by the COTS supplier. The modifications themselves are new development. Change impact analysis should be performed to determine the extent of required re-verification.
9. Maintenance issues such as patches, retirement, obsolescence and change impact analysis
10. Relationships among COTS planning process, acquisition process, integral processes should be defined. Additionally, relationships between COTS processes and appropriate system lifecycle processes should be defined. Every input to a process need not be complete before that process can be initiated, if the transition criteria established for the process are satisfied.
11. Reviews should be conducted to ensure the COTS processes and the system processes are consistent. Ensure the COTS transition criteria are compatible with the system transition criteria. Ensure transition criteria are verified to assure that the outputs of each process are sufficient to begin the next process.

Note: COTS usage may require considerations of glue code, architectural mitigation techniques, derived requirements and COTS specific integration issues for checking consistency. Any supplemental software due to COTS software incorporation in systems is considered developmental software for which all of the objectives of this document apply.

COTS Assessment Process

The focus of this section is on the assurance aspects of acquiring COTS and should not be misconstrued as a tutorial on the acquisition of COTS. However, it must be understood that an unwise purchase of a COTS product could doom

your program to cost and schedule overruns and more importantly induce safety instability that in all likelihood will never be adequately mitigated. The COTS acquisition process is comprised of requirements definition, assessment, and selection.

Requirements Definition: The system's software requirements definition process identifies software requirements that COTS can satisfy. Since COTS are built for general purpose, COTS software may contain more features than the requirements needed by the system. A definition of these features may be available from the supplier or derived from COTS user manuals, technical materials, product data etc.

In the model, depicted in the figure 1, the COTS software requirements are the intersection of COTS software specifications and system's software requirements. COTS software requirements define system software requirements that should be satisfied by COTS software.

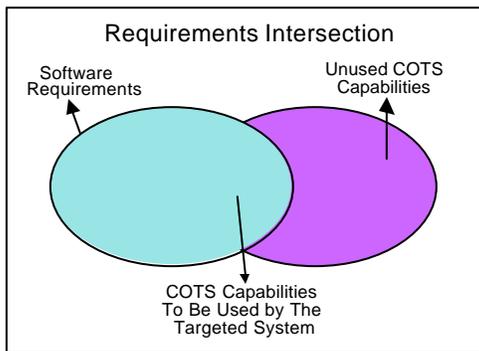


Figure 1 – COTS/Application Requirements Intersection

Due to the use of COTS, there may be derived requirements (e.g. platform dependent requirements, interrupt handling, interface handling, resource requirements, usage constraints, error handling, partitioning) to be added to the system software requirements.

All COTS software requirements and the resulting derived requirements should be provided to program's system safety assessment.

Assessment: COTS candidates should be assessed for their capability to implement the software requirements, for the effect of their respective derived requirements, and to support the assurance/severity level of the system.

During the COTS assessment process more than one COTS candidate may be examined to determine the extent of intersection of requirements with the system's software requirements as depicted in the figure below. Availability and relevance of COTS life-cycle data to support the assurance level of the system should also be assessed. The impact of any unneeded features present in the COTS software should be assessed.

Selection: The selection is an iterative process based on results from the assessment process and comparison of COTS suppliers (COTS supplier experience in the respective system, COTS supplier capability to support COTS software version control and maintenance over the expected lifetime of the respective system, COTS supplier commitment to keep the system design agent informed of detected errors, COTS supplier willingness to address the issue of escrow, etc.). Analyses may be conducted to compare advantages of using COTS versus developing the software.

COTS Assessment Activities

The following are the COTS acquisition activities a program should attempt to achieve and document within a Strategic Technology Refresh Plan:

- a. The COTS software specification should be examined, and a coverage analysis should be conducted against the system software requirements. The purpose of this analysis is to determine the COTS software requirements, to aid in the comparison of candidate COTS.
- b. Available COTS software life cycle data should be assessed. A gap analysis should be performed against the objectives of Section Three of this document for the proposed COTS software assurance level. This analysis aids in comparison of candidate COTS. This analysis may also be used to identify any alternate methods that may give partial or full assurance credit.
- c. Analysis should be conducted to identify derived requirements. This analysis should include all COTS functions, needed and unneeded. Derived requirements may be classified as follows:
 - Requirements to prevent adverse effects of any unneeded functions of any COTS software. This may result in isolation,

partitioning, wrapper code, coding directives etc.

- Requirements that the selected COTS may impose on the system including those for preventing adverse effects of needed COTS functions (e.g. input formatting, call order, initialisation, data conversion, resources, range checking). This may result in interface code, coding directives, architecture considerations, resource sizing, glue-code etc.
- d. All COTS software requirements, the resulting derived requirements and any pertinent supplier-provided data should be provided to system safety assessment.
- e. The selected COTS should be shown to be compatible with the host computer(s) and interfacing systems.

COTS Verification Process

The COTS Verification Process identifies verification objectives that cannot be met using traditional means. For those verification activities where compliance to existing requirements cannot be demonstrated by the available COTS data (e.g. structural coverage), alternate methods such as reverse engineering, prior assurance, process recognition, formal methods, audits, inspections and service history are often used.

COTS Verification Activities

Typical verification activities for COTS software achieved:

- Software reviews and analyses of COTS requirements
- COTS requirements based testing
- Verification of development of any supplemental software due to COTS (e.g. glue code, partitioning, wrappers)
- Verification of integration of COTS into system

Use of alternate methods should be considered upon the following two conditions:

- Justification supported by system safety assessment processes
- Acceptance by the appropriate approval authority.

Activities used for specific alternate methods or for combination of alternate methods are considered on a case-by-case basis. An example

of activities associated with usage of service history for assurance credit is provided below.

COTS Testing

A common testing misconception is that the inclusion of COTS would reduce the level of testing. The theory is that the vendor prior to purchase has previously tested the COTS products. This theory has serious flaws as the thoroughness of vendor testing cannot be verified or validated. In fact one cannot even verify whether known problems were corrected. Additionally, products from one manufacturer must be integrated with products from others and incompatibilities are not uncommon.

Again, reality has not fulfilled the vision. The inclusion of COTS products into mission critical and safety critical systems has actually increased the necessity and duration of testing. The test and safety engineers cannot assume the adequacy of vendor testing, therefore,

When and to what extent should a program regressively test? Regression testing was performed in legacy systems whenever safety critical or mission critical requirements were modified. The robustness of the testing corresponded directly with the assumed risk of the modification. This remains true in a COTS based system. The inclusion of COTS has introduced additional testing requirements.

Portability is the key issue involved with retesting or regression testing. The portability question has tremendous cost and schedule impact when selecting processor upgrades. A portable processor is one that can take the code from processor 'A' and import it into processor 'B' without a recompile. The implementation of any replacement processor that requires a recompile requires a complete and comprehensive retest of the system. A fully portable processor may only require regressively testing the safety specific tests.

Non-critical hardware specific upgrades or replacements such as output devices for data, etc. normally require no retest at all. This determination can only be made after a risk analysis has been performed and documented on the proposed upgrades.

Figure 2, System / Safety Process Flow Diagram illustrates a simplified top-level look at

a typical safety program containing a COTS upgrade. It is highly recommended that once COTS are utilized within any mission critical or safety critical program's operational environment, the assumed risk inherently can not be lower than medium as defined in MIL-STD-882C/D. This holds true regardless of the assumed risk of the program receiving the COTS upgrade. A standalone closed system that does not interface with any other mission critical or safety critical system maybe the only exception to this rule.

Configuration Management

This section describes the configuration management process for a system using COTS. The configuration management system of the COTS supplier is not under the control of system. This requires the system configuration management system to include control of the COTS versions.

COTS Configuration Management Activities

The activities associated with configuration management of COTS are:

- a. A COTS identification method should be established to ensure that the COTS configuration and data items are uniquely identified.
Note: The identification method may be based on COTS identification from the COTS supplier and any additional data such as release or delivery date.
- b. The system's problem reporting should include management of problems found in COTS, and a bi-directional problem reporting mechanism with the COTS supplier should be established.
- c. The system's change control process for the incorporation of updated COTS versions should be established. An impact analysis of changes to the COTS software baseline should be performed prior to incorporation of new releases of COTS software.
Note: The list of changes (problem fixes and new/changed/deleted functions) implemented in each new release may be available from the COTS supplier.
- d. The system's archive, retrieval and release should include COTS specific configuration and data items.

Quality Assurance

The quality assurance process should also assess the COTS processes and data outputs to obtain assurance that the requirements associated with COTS are satisfied.

Note: It is recommended that the COTS supplier quality assurance is coordinated with the system's quality assurance process when feasible.

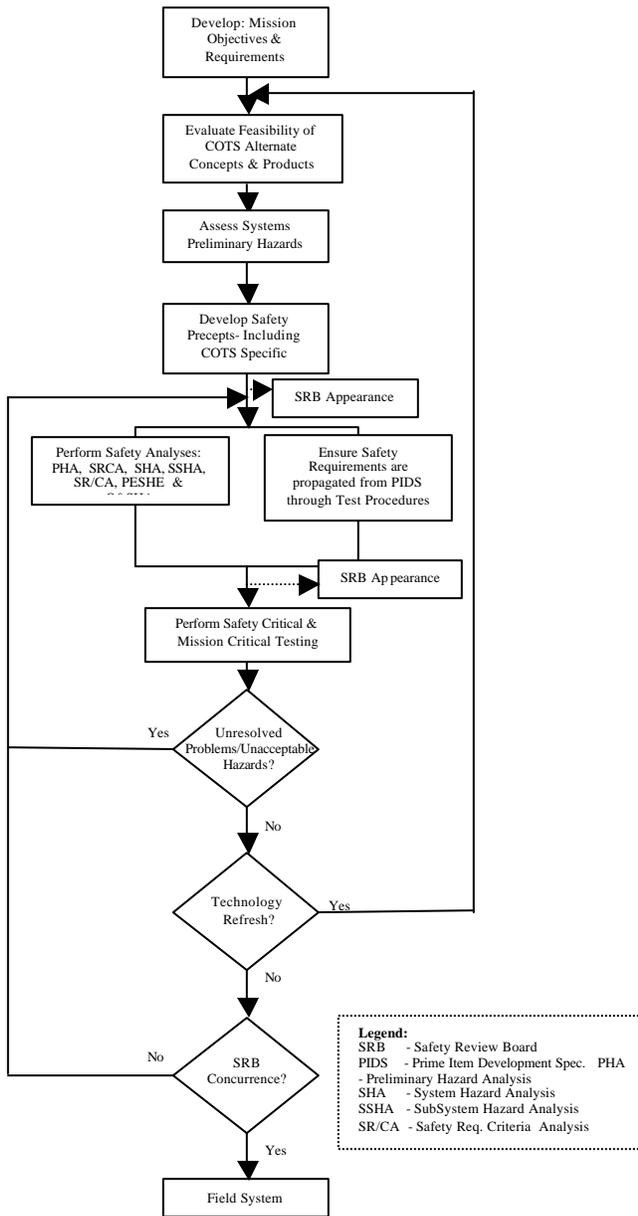


Figure 2 – Safety Process Flow Diagram

System Safety and COTS

With the advent of Acquisition Reform an emphasis on cost, schedule, and the use of COTS has been mandated. This mandate should not be viewed as lowering or relaxing any standards for mission critical or safety critical systems. It is strongly recommended that your safety program implement a strong MIL-STD-882 / STANAG 4404 safety program.

The safety community must also consider cost, schedule, mission, and program directives in the performance of safety evaluations, analyses, and recommendations. The safety community/engineer must understand that absolutely no one desires to build an unsafe system. The safety engineer should not be a stop sign in the development of a system. The safety engineer should assume the posture of caution and provide alternative cost, schedule, and mission effectiveness recommendations. Successful proactive safety programs work near the epicenter of a program. Unsuccessful reactive safety programs work from the circumference looking inward. The safety community must be viewed as productive members of the development team before they are truly permitted to influence the decisions taking place at the program’s epicenter.

System sponsors must also recognize that safety is a leg in their system’s stool. Safety should be included before the fact rather than after. A proactive vice a reactive system safety program does produce significant financial benefits over the projected life cycle of your system.

COTS Specific Safety Precepts

Systems developers and designers cannot be expected to inherently make the right safety decisions and selections without proper guidance. Developing and implementing safety precepts usually initiate proper guidance. Developing strong safety precepts during the conceptual phase is paramount to mitigating the risk of your COTS-based system. The wise selection and implementation of safety precepts prior to development of any system goes a long way in building the foundation of your system safety program.

Program specific precepts should be developed coincidental with or shortly after the Preliminary Hazards of your system are identified and should be updated as appropriate throughout the development process.

Each program’s safety precepts should be tailored to and receive concurrence from the Weapon System Explosives Safety Review Board (WSESRB), Certification Agent, and/or your program’s respective safety review board during an introductory appearance. Early acceptance is critical to avoid rework and design issues that are not easily correctable. These precepts should be explicit, encompassing, achievable, and enforceable. It may be beneficial to analyze the precepts of other like systems; however, thorough analysis must be performed to ensure their comprehensiveness and applicability to your program. Table 1 COTS Specific Safety Precepts were developed and adopted by the NSSMS MK 57 Mod 4-9 program and are recommended for inclusion into any COTS-based system:

COTS Specific Precepts	Intent of Precept/Remarks
1. No mission or safety critical functions shall be initiated or sustained by COTS.	This precept requires the development of application processing or operator action to initiate or sustain a mission critical or safety critical function.
2. Ensure operating systems and environment functionality shall not be accessible by the operator.	The intent of this precept is to prevent the operator from interfacing and either advertently or inadvertently modifying the underlying operating system(s) resulting in unknown, possibly catastrophic consequences.
3. All safety critical and mission critical resources shall be dedicated single use resources.	Stated simply, this requirement does not permit the use of shared resources. A fire or flight control computer is and remains only a fire or flight control computer. This requirement prohibits the loading of external or extraneous software to perform non-mission functions (e.g., tax software, games, word processors, etc.) to prevent data and memory corruption.

Table 1 – COTS Specific Safety Precepts

Disable All Unused COTS Functionality: The disabling of all unused COTS functionality is a desirable goal rather than a precept due to the unavailability and/or limitations of the COTS software and hardware. There are also other issues such as vendor support, liabilities, and unfamiliarity with the product that make this mostly unachievable. However, there are configuration items that may be modified during initialization, such as, disabling screen saver functions.

Summary

Keeping pace with technology should be viewed realistically. Staying abreast of technology is not synonymous with maintaining the cutting or bleeding edge of technology. Each upgrade should be viewed as a management decision. Cost, schedule, system safety, and life cycle support are the four primary programmatic risks involved in any COTS upgrade or new development program.

Hopefully this paper will help the safety professional and the management authority in making better decisions regarding the selection and use of COTS in the development or upgrades of mission/safety critical systems. This paper is not a silver bullet, however, it is hoped that it may positively impact the safety and ultimately the success of your programs.

Biography

I. Warren P. Naylor, BAE SYSTEMS Applied Technologies, System Safety Manager, 1601 Research Boulevard, Rockville, MD 20910-3173, telephone – (240)-994-1765, facsimile - (202) -548-5504E-mail - warren.naylor@baesystems.com

Warren P. Naylor is the systems safety manager and lead systems safety engineer for BAE SYSTEMS Applied Technologies with over 8 years of software/systems safety experience and over 25 software development and field service support with Department of Defense weapon systems.

Mr. Naylor is currently providing software safety security, and human factors support the Federal Aviation Administration AIO 200. Additionally, Mr. Naylor is providing independent system safety consultation and safety management services to the

NATO SEASPARROW Project Office (NSPO / PMS471) and manages the safety efforts for the Naval Surface Fire Support (PMS 529) programs.

Mr. Naylor's proximity to PMS471, the FAA, and NSWCDD G71 has provided him with a unique insight into implementing, managing, and maintaining safety in highly complex, integrated COTS-based real-time mission and safety critical systems.

II. Ronald Stroup, Software Safety and Certification Lead, Federal Aviation Agency, AIO 200, 800 Independence Avenue, S.W. Washington, DC. 20591, telephone – (202)-493-4390, facsimile (202)-267-5080, e-mail - ronald.l.stroup@faa.gov

Mr. Ronald Stroup joined the Federal Aviation Administration as an Aerospace Engineer in 1989. He holds a Bachelor of Science in Avionics Engineering from Parks College of Saint Louis University. Mr. Stroup served, as a systems engineer in the Aircraft Certification Services' Chicago Aircraft Certification Office and in 1997 became the Software Technology Specialist for the Aircraft Certification Service. His responsibilities included providing technical expertise in the area of software approvals and acted as a focal point to improve the software approval process.

Since 1998, Mr. Stroup has served as the Software Safety and Certification Lead for the Office of Information Services and Chief Information Officer. His duties include developing and applying software assurance standards to the acquisitions of software-intensive National Airspace Systems.

Mr. Stroup is a member of the RTCA/SC-190 Committee, IEEE, and Co-Program Manager for the Streamlining Software Aspects of Certification initiative. He served as a Subject Matter Expert for the Software Fundamentals course to develop training in the application of RTCA/DO-178B Assurance Standard and software engineering practices. Mr. Stroup serves on the FAA's Systems Engineering Council, the System Safety Working Group and Eurocontrol's Software Task Force.

Acknowledgements

We would like to thank all of RTCA SC 190 and in particular those members of the COTS Subcommittee, lead by Subcommittee Co chairs Mr. Claude Secher (French DGAC/STNA) and Ms. Uma Ferrell (Ferrell and Associates).

References

1. DOXXX/ED.XX, Guidelines for CNS/ATM System Software Integrity Assurance, EOCY 2001
2. Naylor, W.P., Maintaining System Safety In A COTS Environment, ISSC, 1999
3. Naylor, W.P., Assessing Safety and Programmatic Risks of Proposed System Upgrades, Journal of System Safety, Volume 36, No 2 2Q 2000.