# A framework to address the impact of system of systems integration using commercially off the shelf (COTS) technology

Mary D. VanLeer*

The Sustainability Consortium,
University of Arkansas,
534 West Research Center, Blvd.,
ENTR 120, Fayetteville, AR 72701, USA
E-mail: mary.vanleer@att.net
*Corresponding author

Rashmi Jain

Department of Industrial and Systems Engineering,
National University of Singapore,
1 Engineering Drive 2, 117576, Singapore
E-mail: iserashm@nus.edu.sg

**Abstract:** For systems engineering, systems integration (SI) establishes linkages between hardware (HW), software (SW), products, services, processes and humans. Over the last decade the world of systems development has evolved rapidly particularly in the use of commercial-off-the-shelf (COTS) products as elements of larger systems. The growing trend toward COTS-based systems (CBS) architectures is based on modular components available within the market. This trend has presented various challenges for systems engineering practitioners attempting to understand the implications of using COTS products within these large and complex projects. This paper analyses those unique aspects of COTS products that influence the SI process differently than the integration of 'in-house' custom developed products.

**Keywords:** systems integration; systems life cycle; commercial off the shelf; COTS; COTS-based systems; CBS; systems of systems architecture; systems requirements; systems of systems integration complexity.

**Reference** to this paper should be made as follows: VanLeer, M.D. and Jain, R. (2013) 'A framework to address the impact of system of systems integration using commercially off the shelf (COTS) technology', *Int. J. System of Systems Engineering*, Vol. 4, No. 1, pp.23–43.

**Biographical notes:** Mary D. VanLeer received her Masters of Engineering in Systems Engineering from Stevens Institute of Technology. She is currently employed at The Sustainability Consortium as Director, IT Strategy where she is defining the technology platform for sustainability reporting. Prior to the Consortium, she was the Director of IT for the Arkansas Lottery, Director of Engineering Operations for International Game Technology, Director of Software Engineering at Sun Microsystems, and Manager of Product Engineering at StorageTek. In her over 30 years she has created centres of excellence in quality management, project management, and systems engineering.

Copyright © 2013 Inderscience Enterprises Ltd.

Rashmi Jain is currently an Associate Professor with the Department of Industrial and Systems Engineering at the National University of Singapore (NUS) since August 2011. Prior to joining NUS she was an Associate Professor at Stevens Institute of Technology, USA from 2003 to 2011. She is the Chair of the System Integration Study Group on ISO/IEC JTC1/SC7. She is member of the Systems Engineering Technical Committee of IES (The Institution of Engineers Singapore). She has had several international recognitions and positions such as Head of education and research for INCOSE, and Visiting Associate Professor at Keio University, Japan. She is on the editorial boards of several international journals. She has authored over a 100 technical papers, reports, and monographs, and co-authored a book.

## 1 Introduction

The use of commercial off the shelf (COTS) components or products in system development has significant effects on integration, and verification and validation (V&V) within the system of systems environment. A COTS item can be defined as one that is sold, leased, or licensed to the general public; offered by a vendor trying to profit from it; supported and evolved by the vendor who retains the intellectual property rights; available in multiple, identical copies; and used without modification of the internals (Albert and Brownsword, 2002; Money and Gansler, 2000). Over the last decade there has been a paradigm shift towards COTS-based systems (CBS). More and more systems development programs are mandated to use CBS due to the pressures to achieve cheaper, faster, and better system development. Consequently, system of systems engineering practitioners must understand the implications of using COTS, particularly before initiating large and complex projects.

Adoption of COTS systems are driven by the time-to-market constraints and the higher technology readiness level (TRL) of the COTS products. But the longevity and lack of design flexibility with COTS systems constrains their integration. Difficulties and lessons learned from COTS integration in practice are reported in Abts (2002), Abts et al. (2000), Boehm et al. (1995), and Bansler and Havn (1994). Interfaces with and between COTS products is one of the major challenges for COTS integration. There are no widely agreed upon COTS standards (Voas, 2001) mainly due to marketing strategies aimed at obtaining vendor lock (Morisio and Torchiano, 2002). COTS integration can be high risk activity as COTS components make several assumptions about architectural issues. When these assumptions conflict, the simplicity of using COTS is quickly replaced with complexity and integration scaffolding (Egyed et al., 2005). However, over the past decade, data and control integration mechanisms and standards have matured significantly, offering systems developer's options in utilising COTS technologies.

Different approaches have been suggested in the literature for categorising off-the-shelf (OTS) products. Several authors suggest that COTS products have become end-user oriented and their impact on the development process depends on the source, customisation, bundle, and role (Morisio and Torchiano, 2002; Egyed et al., 2005; Yakimovich and Basili, 1999; Abts et al., 2000). Morisio and Torchiano (2002) identified COTS attributes based on their possible values and have characterised COTS-based attributes and impact on the development process. According to Abts et al. (2000), the more granular attributes of COTS that impact systems integration (SI) in particular are

correctness, availability/robustness, security, product performance, understandability, ease of use, version portability, functionality, precision required to meet specifications, cost, maturity, vendor support, training, and vendor concessions (Abts et al., 2000). These attributes impact the effectiveness (or lack thereof) of the CBS integration process. A framework that delineates these attributes, identifies their impact over the lifecycle of system integration, and guides design decisions would be a significant tool for providing systematic guidelines and analysing the unique aspects of CBS to understand their overall implications on system of systems integration (SoSI). Such framework is presented in this paper.

## *1.1 Background*

According to Mark Maier, there is an emergent class of systems built from components that are evolutionary in nature and are independent resulting in a greater emphasis on interface design than in traditional systems engineering (Maier, 1998). It is broad and ubiquitous concept consisting of various elements that impact the early aspects of engineering large complex and emergent systems and the systems management processes (Sage and Lynch, 1998). SoSI facilitates the integration of systems both legacy and integrated systems hardware, software, net-centric, products, services, business processes, and human activities into a total system of systems solution (Grady, 1994) where independent systems behaviours emerge through voluntary and collaborative interaction (Boardman and Sauser, 2006). From a process perspective, the SoSI process creates the links within the systems engineering process from requirements collection to V&V and ultimately to implementation of the system (Morisio et al., 2002) ensuring the ability to define systems optimisation strategies that support technology refresh as components change or become obsolete. The SoSI process begins at project inception and continues throughout the entire systems lifecycle (Jain et al., 2010).

Nowadays, there are few systems where all system components/sub-systems are newly designed ‘in-house’. For such ‘in-house’ systems, the integration approach differs from the approach for CBS. In today’s world, complexity of systems development is increasing rapidly due to the expansion in the number of system components, sub-systems, links, input, output, data, and control interfaces, and standards that must be integrated. This situation has created a shift from an all-inclusive in-house development to an integration of COTS components in an effort to reduce the time to market, cost, and system development time. There are several factors that can impact the success or failure of CBS integration differentiating it from the traditional SoSI approach. These factors include: technology refreshment, vendor responsibility, usability, inter-operability, architecture compatibility, scalability, interconnectivity, COTS marketplace, safety and security, portability, performance, systems failure risk, and life cycle cost.

In a traditional engineering approach, a system architect would make architectural decisions based on system requirements, constraints, and business objectives alone. After the system architecture plans are stabilised, a set of COTS products would be evaluated. According to Brownsword et al. (2000), this traditional engineering approach is not suitable for CBS since there may be no suitable COTS products available to suit the specific needs of the envisioned system. Thus, a new problem arises for such COTS-intensive systems, namely that an early understanding of the available COTS products in the appropriate marketplace. This implies a need to iterate between

requirements definition, development, and COTS product evaluations much earlier in the system life cycle than in traditional development (Kohl, 2001). More specifically, it requires engineers capable of and committed to performing these design iterations and trade-offs.

Building systems utilising components available from the market offers some unique advantages and challenges. Dawkins and Riddle (2000) point out that the safety related systems market cannot sustain the rate of technological advancements stimulated by the huge commercial market (Redmill, 2004). COTS components exist in an environment where there are multiple vendors and users who continuously drive technology advancements. Even if the design rules or standards are under consideration in certain environments, companies collaborate with their vendor partners in 'plugfests' (Gunderson and Minton, 2011) to ensure compatibility in a market where the standard is still evolving. The challenge for systems designers is when to drive market evolution or know when to follow it. The effect of COTS components on a project or system depends on the degree which the program intends to use COTS components, the extent to which introducing these components alters the characteristics of the system, and the complexity of integrating commercial and custom developed items (Jain et al., 2009).

Architectures which utilise COTS can reduce the time to market (Sankaran et al., 2011; Bolloju, 2009), but the trade-off maybe introducing an architecture that does not meet all stated requirements. Systems developers must strike a balance between the advantage and disadvantages of using COTS to create a solution that closely meets the needs of the market without additional expense and overhead of custom development.

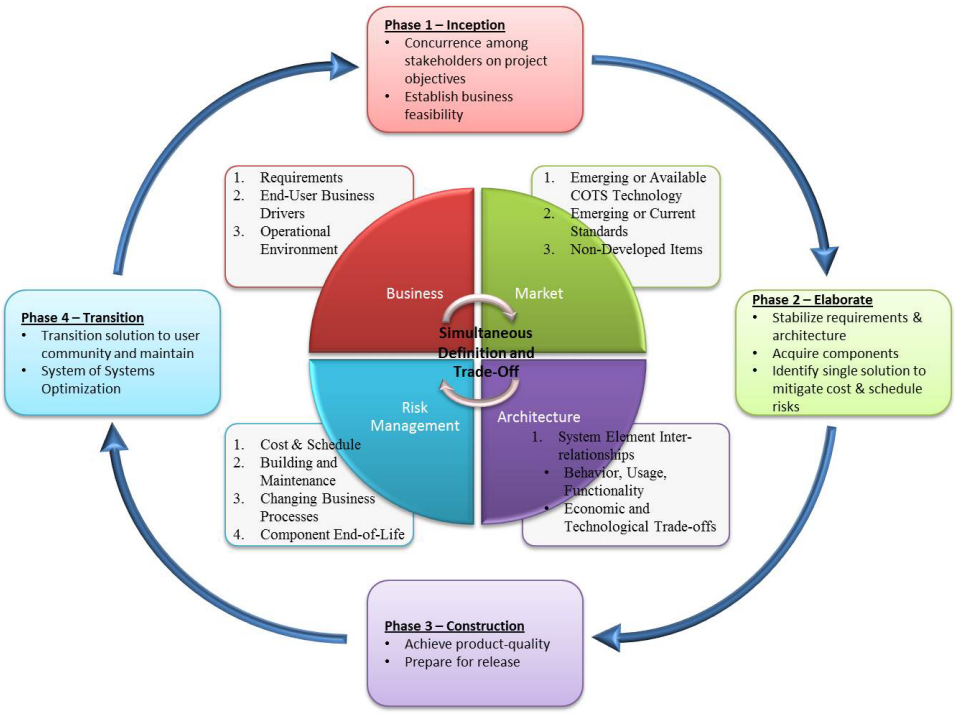
A common misconception when COTS products are to be used is that the required skill set is reduced. Stakeholders may believe there is no need to have an understanding of the overall architecture and design of the COTS products. It is a mistake to assume that all COTS products will seamlessly integrate (Blanchette, 2005). While vendors may advertise compatibility, when failures occur there is often a finger pointing exercise as to who is to blame. It is important that System of Systems Integrators possess the requisite knowledge of the COTS products and the skills necessary to determine the failure mode. As more companies move towards integratable system of systems consisting of COTS components and away from all in-house development, there is a concern that valuable intellectual capital will be lost. Intellectual capital is not necessarily lost it may be reduced in areas of proprietary development and enhanced in the ability effectively evaluate varying technologies then possessing the knowledge necessary to integrate them into a total solution (Sametinger, 1997).

## 2 A proposed CBS approach for SoSI

The increasing acceptability and introduction of COTS products into existing systems or legacy systems necessitate understanding the process of integration for CBS. It becomes mandatory to identify and analyse the unique aspects namely stakeholder needs and business processes, marketplace, architecture/design, other systems to include in the integration, and programmatic risk of CBS to differentiate between traditional SI approach and customised CBS approach. Using pre-existing components, particularly COTS components, introduces significant challenges. These challenges drive the life cycle processes of SI. Building a solution based on integrating pre-existing COTS products is different from typical custom development in that the COTS products are not

necessarily designed to meet the system's specifications. COTS products are built to satisfy the requirements of a market segment. Therefore, an understanding of what the product's functionality how it is likely to change over time must be established in order to modify the requirements and end-user business processes as appropriate, and thus to drive the resulting architecture (Albert and Brownsword, 2002). As an example of this approach, the largest train traffic control system was designed as a system of systems and introduced the advantage of how COTS technologies in hardware, software, and net-centric systems were integrated to address the needs of rapidly implementing a dependable system solution (Tomita et al., 2008).

**Figure 1** The four spheres of influence supporting the COTS iteration process (see online version for colours)




It is important to understand the influencing aspects of CBS namely marketplace, stakeholder needs/business processes, architecture/design, and programmatic/risk, in order for evolution of customised CBS approach. Key to building solutions based on COTS products is the need to simultaneously define and tradeoff among four 'spheres of influence' (Albert and Brownsword, 2002). While tradeoffs are common in any engineering endeavor, tradeoffs in this case are driven by the desire to leverage COTS products from marketplace.

As shown in Figure 1, an emphasis is placed on the four spheres of influence and their synchronisation throughout the lifecycle. The first sphere of influence is the definition of the stakeholder needs and business processes. The marketplace influence is defined by the available and emerging COTS technologies. The architectural influence defines the essential elements of the system and their interdependencies. The

programmatic and risk sphere of influence considers the project management aspects of the system such as, cost, schedule, and long term sustainability of the solution (Albert and Brownsword, 2002).

The four spheres are simultaneously defined and trade-offs made iteratively throughout the system life cycle. While a decision in one sphere can have a positive or negative impact on the other spheres due to their interdependencies these influences create the foundation for success of CBS.

The iterations are managed by four phases: Inception, Elaboration, Construction, and Transition. These are presented in Table 1. Each phase consists of one or more SI iterations designed to accommodate changes induced by the COTS marketplace. It is a phase gate approach that has defined objectives, deliverables, exit criteria and review processes designed to build a knowledge base and continuous evaluation of the components available and alternatives that may impact the four spheres.

**Table 1** Phase definitions

| *Phase* | *Definition* |
|---|---|
| Inception | Defines the scope of the solution and provides the high-level plan for the project (Albert and Brownsword, 2002). |
| Elaboration | Provides the next level of detail of the project management process. It evaluates the requirements, business processes, and architecture with a goal of selecting the appropriate components for implementation (Albert and Brownsword, 2002). |
| Construction | Prepares the system for release ensuring it meets the stated goals of the stakeholder needs and supports the design goals of the architecture. Within the Construction phase, focused effort is placed on the preparation of the business environment for acceptance of the system. |
| Transition | The solution is released to the defined environment. It ensures that the necessary training and support structures are in place for successful implementation. It also includes the need for technology refresh should components reach end-of-life. It is the phase of system of systems optimisation. |

## 3 COTS integration throughout the life cycle

Based on the literature review and evaluation of SI processes and models throughout life cycle, the author presents a systems integration framework (SIF) for CBS (SIFCBS) in Figure 2. This framework provides a fundamental view that identifies a comprehensive set of end-to-end activities that may constitute and define the scope for CBS. This approach is based on the premise that system integration occurs throughout the lifecycle and is not a one-time activity. The process of SI using CBS has been divided into number of life cycle phases: Concept, Design, Development, Testing, Production, and Deployment/Optimisation and integrates the concepts of the spheres of influence. By extending the SIF we define the SI process as a set of phases that transforms the stakeholder requirements into an operational system by unifying the process components and product components into a whole while ensuring compliance to the specified levels of component operations and interoperability (Jain et al., 2010). The scope and objectives of SI for CBS should be clearly identified to ensure that new systems and components

(COTS products) are able to integrate seamlessly with the existing systems and components.

A life cycle view of SI will create an understanding of the context of SI for CBS and its scope across system engineering life cycle phases. It also helps in identifying and addressing SI issues related to CBS and when they may occur throughout the system development, implementation, and operation. The SIFCBS framework shown in Figure 2 identifies the necessary elements of SI and illustrates the dependencies among them. These six phases constitute a comprehensive scope of SI.

**Figure 2** Life cycle view of a SIFCBS (see online version for colours)


## *3.1 Concept phase*

Once a conceptual design for a system is chosen and all operational scenarios (use cases) to understand the context are analysed, the broader category of stakeholder requirements are then refined and derived to form system requirements or specifications. During the concept phase of systems development it is critical to identify requirements that will impact the SI activities of CBS. These requirements termed as COTS integration requirements address the required level of integration and quality. These integration requirements fall under the category of Adaptability, Interoperability, Applicability,

Standards, Flexibility, Requirement conformance, and Requirement stability. The aspects of COTS integration requirements are defined as:

*Adaptability* – The measurement of the system's ability to adapt to requirements changes, whether as a result of system redesign or to accommodate multiple applications. System requirements for CBS should be written in such a way that will allow COTS to meet the integrated environmental conditions. Selection of COTS products should be done based on adaptability of systems requirements to COTS capabilities or requirements (Redmill, 2004).

*Interoperability* – Defined as the ability of the system to 'play well with others' both with the systems it was originally designed to work with and with future systems. Perform functional decomposition and generate derived requirements to align required capabilities with capabilities of widely available COTS products.

*Requirement Conformance and Stability* – The ability of developers to use requirements metrics to measure the CBS's conformance and stability so they can monitor specifications, translations, and volatility, as well as the level of adherence to the requirements. COTS components are often unstable and this instability can affect the requirements stability if the system developers adapt these requirements to incorporate changes to selected components (Sedigh-Ali et al., 2001).

*Requirements Flexibility (Negotiating Requirements)* – "Don't go for COTS if you can't bend your requirements. If you can be flexible, COTS is cheaper. If not, it's more expensive" (Lewis et al., 2000). Requirements for COTS-based systems should not be so strict that it either excludes the use of COTS or that it requires large product modification in order to satisfy very specific requirements (Alves and Finkelstein, 2002). In fact, a wide range of conflicts can arise during the matching between customers' requirements and COTS features, ranging from a simple misalignment of desirable features to severe problems of product integration into the organization (Sivzattian and Nuseibeh, 2001).

*Applicability* – Applicability of COTS products is often restricted to certain operating systems (Unix or Windows), to specific languages (C, C++, or Java), or to specific environments or tools. This information provides the COTS users with a general knowledge about the environment in which the components cooperate (Dong et al., 2005).

*Standards* – Specifications of COTS components should include the information about the standards these components conform to, such as CORBA, DCOM or .NET (Dong et al., 2005). This information gives systems integrators confidence on the compatibility between COTS components during COTS integration. To build COTS components following a *de facto* standard is a good strategy in the market. Explicitly conveying this information to systems integrators increases the chance of selecting a specific component. The process of systems requirements for CBS should be standardised in order to get compliance with COTS capabilities.

### *3.2 Design phase*

The necessity for system engineering when integrating a number of COTS components is often ignored because designers view the system under construction as simply a

procurement of a set of qualified COTS components. However, this is not a 'one time' activity, and changes in COTS components and in the marketplace often drive frequent reengineering of the CBS throughout the life of the system (Money and Gansler, 2000). Therefore, the developers are required to analyse requirements, evaluate COTS products, design, integrate, and test the system at various phases of the system development. Modularity, technology refreshment and insertion, decomposition of functions, and architectural compatibility are all aspects of the design phase that impact the SI process.

*Modularity* – is the degree to which a system is structured as a configuration of smaller, self-contained units with well-defined interfaces and interactions (i.e., independently testable), moderating design complexity and enhancing its clarity, and enabling design and functional flexibility and variety for the system as a whole (McCabe and Pollen, 2004). Systems that integrate multiple COTS components require extensive engineering to define system architecture with a modular design that is open enough to facilitate the insertion of new COTS technology. Modularity is usually associated with open systems design, but in this case, open systems design is considered to be a design standard and modularity a best practice in architecting for CBS.

*Technology refreshment and insertion* – creates a defined process and management plan that supports COTS updates so that they can be synchronised with each other and the organisation's release and business cycle. Without synchronisation and management, updates might occur sporadically during the maintenance part of the cycle and the risk of technology obsolescence might increase dramatically (Reifer et al., 2003). The architecture of CBS should be capable of allowing periodic COTS technology refresh and insertion cycles throughout the system life cycle by implementing an open architecture, and providing an upgrade path for performance enhancements and cost reductions. Before implementing any new technology refresh it is important to evaluate the timing and value of the change. If the system is not in urgent need for updating, it may be prudent to suspend the change until a major refresh is needed (Lewis et al., 2000).

*Decomposition of functions* – partitions the system to maximise the use of widely used commercial standards for logical and physical interfaces e.g., network topologies, protocols, and how operating system interface with applications (Redmill, 2004). An analysis of the interfaces and integration requirements provides an estimate of the complexity of the interfaces, middleware, or glue code required for integrating different COTS products. Overly complex interfaces complicate testing, debugging, and maintenance which can degrade the system's quality.

*Architecture Compatibility* – can create a significant issue that needs to be addressed when considering COTS integration. For example, incompatible COTS design assumptions can cause serious interoperability problems affecting data, control, timing and service provisions. An architecture that allows efficient evolution of system is a strategic asset for any system design and critical for CBS (Brownsword et al., 2000). For example, incompatible COTS design assumptions can cause serious interoperability problems affecting data, control, timing and service provision (Boehm et al., 1995). Presentation integration can also be affected. For instance, using commercial software development frameworks for producing COTS implies

that certain compromises must be made on system requirements. According to Boehm et al. (1995), the problems created by the use of COTS can be addressed by addressing the technology needs such as mismatch detection, assessment and conciliation, and guidance for architecture design, construction and documentation. Addressing these needs, Garlan et al. (1995) suggest the need to develop better ways to document architecture assumptions, especially non-functional properties, first class component connectors such as interaction protocols, and architecture analysis tools beyond compilers.

### *3.3 Development phase*

The development phase converts system design into a complete system including the various activities of acquiring and installing the system within its intended environment, preparing test case procedures, preparing test files, developing the components, and procurement of the COTS products. It is important to monitor these activities to ensure compliance with required standards. By using prototypes to evaluate key areas of the system design, developers can gain insight into the performance of the system early in the development cycle thus reducing risk that the system will not meet the performance criteria. Upfront costs and time to market are two measurements within the development phase that can impact the SI process (Redmill, 2004).

*Upfront cost* – is the overall expense incurred during the course of system development. Expenses include the costs of COTS component acquisition and integration and quality improvements to the system (Sedigh-Ali et al., 2001).

*Time to market* – is the measured time required to release the system/product, from the beginning of development and COTS component acquisition to delivery. A modified version of this measurement aspect can evaluate the speed of incremental delivery, measuring the amount of time required to deliver a certain fraction of the overall application functionality (Sedigh-Ali et al., 2001; Redmill, 2004).

### *3.4 Testing phase*

Within the testing phase there are several activities which support the SI process.

*Vendor testing* – is generally carried out by vendor before delivery of COTS component/items. Vendor supplied products are tested through configuration item testing prior to delivery to the systems developer. Vendor testing eliminates the need for the system integrator to establish the test environment and test processes for two levels of testing (module testing and configuration testing). As COTS products are tested by the vendor, they should be validated by systems integrator for congruence between vendor testing requirements and systems integrator testing requirements. The system integrator validates the test process of vendor by reviewing the vendor test procedure, witnessing testing at vendor site, and incoming inspection and validation (i.e., sample testing, integrate with spares and repairs testing) (Jain et al., 2008).

*Managing the interface between the vendor and systems integrator*– interface management between vendor and systems integrator is the most important aspect of the testing phase, as the responsibility of testing of COTS components are transferred

from systems integrator to vendor. Unique testing requirements provided to the vendor can drive up the cost and reduce the benefits of using COTS product in the system (Sametinger, 1997). In order to manage the testing activity between vendor and systems integrator, the system integrator must validate COTS products against systems requirements by 'visual inspection' and 'hot-box testing'. There are certain issues associated with vendor like warranty repairs, technology refresh, and technology insertion which cannot be controlled by systems integrator. The vendor cannot validate failures found by the system integrator; therefore, vendor testing process changes may be required to accommodate product upgrades creating interface difficulties between vendor and systems integrator. In order to manage the vendor interface effectively it is a good practice to establish a document of understanding (DOU) as part of procurement contract.

*Hot box testing (receiving and inspection)* – a system comprised of COTS components can utilise hot box testing to check the functional behaviour, performance, and standards compliance by the systems integrator as an integral part of system validation process. Hot box testing establishes an environment that replicates the operational system.

*Immediate testing* – a COTS -based infrastructure design that supports components that use an open standards interface will create the foundation for 'plug and play' enabling rapid introduced into the system for immediate testing (Giangarra and Semple, 2011).

### *3.5 Production phase*

During the production phase of the CBS, it is important to measure vendor response to system issues such as failures, and changes in COTS product configuration and interact with the vendor on the evolution of technology. The following aspects of production phase impact the SI process:

*Configuration management* – the vendors release COTS products according to their own schedules, requiring the support of a configuration management system that could select from among multiple versions of COTS products in order to produce different system configurations. Configuration management (CM) establishes and maintains system artifact integrity and traceability throughout the CBS's lifetime beginning with the first evaluation of candidate products. Since the managed baseline is expected to change more frequently due to changes in COTS products and the marketplace, the CM system will need to track new artefacts, including COTS product versions, tailoring, patches, installation procedures, and possibly licence management information (Brownsword et al., 2000).

*Evolution of technology*– to take advantage of advancements and investments in COTS technology and to avoid technology obsolescence or product incompatibility, (Verma and Plunkett, 2000) systems integrators must be able to anticipate changes in the market and proactively manage the integration of new technologies. Upgrades are frequently not upwardly compatible, old releases become obsolete and unsupported by the vendor. If COTS architecture mismatch doesn't get you initially, COTS architecture drift can easily get you later (Boehm and Abts, 1999). During

production phase, it is critical to interact with the vendor on the evolution of technology to establish technology refresh plan and coordinate with the technology insertion plan. Therefore, there must be a strategy and plan for upgrade at pre-determined intervals as part of the initial system project plan. With COTS-based systems, the strategies change due to span of control issues namely technology refresh, and technology insertion. It is recommended to create an operational test environment to use for loading new COTS component versions in order to understand any and all impacts of these changes (Lewis et al., 2000). It is important to carefully evaluate COTS vendors' track records with respect to product evolution predictability. Widely varying feature sets, too-frequent updates, and dramatic shifts in product capabilities can cause problems in the long term. Finally, it is important to establish a proactive system release strategy, synchronising COTS upgrades with system releases. Planning the ongoing integration of evolving COTS products with an internally developed system helps ensure that both continue to function harmoniously (Boehm and Abts, 1999).

### *3.6 Maintenance and obsolescence phase*

During maintenance phase of system life cycle, COTS products undergo a technology refresh and technology insertion. As a part of maintenance activity, maintainers have to decide whether to allow upgrades of COTS products or retain old versions. If they choose to retain old versions, they will eventually reach the point where the vendor no longer supports those versions. If they choose to upgrade, they must synchronise the associated update with their release cycle and with product updates other vendors are making. A major challenge during maintenance phase is striking a balance between system stability and the need to stay current with the marketplace (Brownsword et al., 2000). The most important aspect of maintenance phase is the planning and accounting for the maintenance cost throughout the system's lifecycle. It may be discovered that the cost to maintain CBS equals or exceeds that of developing custom system. Maintenance in this context involves updating CBSs with new technology refreshes, technology insertion, modifying wrappers and glue code, and incorporating rating fixes and repairs into the system as part of the optimization (Reifer et al., 2003).

This phase also takes into account components obsolescence throughout the system of systems lifecycle. Obsolescence can an extensive impact on the overall cost to maintain a system based on the length of time the system is in service. Obsolescence may impact: availability of hardware components; software within the system and the software components needed to maintain it; supporting applications, documentation and data formats needed to access and maintain the system; procedures and methodologies; human capital skillsets. The obsolescence problem cannot be avoided, but the impact can be minimised through risk mitigation planning (Sjoberg and Harkness, 1996; Kang et al., 2012).

### *3.7 Role of vendor*

The COTS product embodies the vendor's expectation of how it will be used. This includes the concept of operation it supports, interface and data standards, architecture and design, and characteristics of form, fit, and function. Equally important are the vendor's business approaches and management strategies in areas such as development,

maintenance, distribution of updates, and availability of spare parts (Money and Gansler, 2000). System engineers must adjust the system requirements to match with vendor's anticipated uses of the COTS products and the vendor's business approaches in order to improve the effectiveness and feasibility of systems requirements.

There must be an understanding and control over total ownership cost of the CBS. Frequently, the sustainment cost associated with COTS integration has been underestimated by systems integrator. These costs include market research, evaluation, test, and integration for version upgrade, COTS replacement, technology refresh, and annual licensing fees. Sustainment cost becomes the critical cause of failure of COTS integration. Sustainment cost is commonly the shared responsibility between the vendor and systems developer. Success of COTS integration depends on the issues regarding cost by embracing total ownership cost models that incorporate both routine maintenance costs and costs for frequent technology updates of COTS products (Money and Gansler, 2000).

Having vendors as part of integrated product teams helps foster a more trusting partnership among the vendor, contractor, and the organisation. Licencing is the primary vehicle for securing the use of COTS products such as H/W or S/W. Data rights or detailed technical specifications are the marketplace drivers for protecting a vendor's intellectual property. Licence agreements and data rights can and should be negotiated to decide tradeoffs parameters of the COTS integration process. Licence agreements should be signed in such manner that they will remain flexible to address any unanticipated issues/risks during the system development (Albert and Brownsword, 2002).

### *3.8 Marketplace*

The marketplace plays a key role to the success of CBS integration. Some of the marketplace issues are:

1 COTS product maturity

2 marketplace maturity

3 vendor responsiveness

4 suitability of licences for user application

5 release schedule

6 unpredictable content and quality

7 rapid technology turnover

8 limited support of past releases (Redmill, 2004).

Properly documented market research enables the systems integrator to create selection criteria used to evaluate vendors based on business practices, vendor performance, and relative size of the system to the vendor's business base. The criteria are used to justify a vendor selection that best aligns with the organisation. Business relationships must be established with contractors and vendors through communication of the COTS product

and business requirements forming partnerships to ensure the CBS solution provides maximum value to all stakeholders. Market research data can also provide the necessary information to create a set of criteria used in a trade-off analysis to select the most suitable COTS product(s) for integration throughout the life cycle. The system developer must conform to the behaviour of the other vendors in the marketplace, and then exert control by managing and verifying requirements in a manner that optimises that use of COTS products by adopting the requirements of the other vendors as closely as is practical. Finally, the system must be engineered to accommodate marketplace driven changes to COTS products that may occur at any point in the lifecycle (Money and Gansler, 2000).

## 4 Validation

In an effort to validate the SIFCBS framework that is discussed in this section a survey was designed based on the SIFCBS elements to evaluate the applicability of the end-to-end activities that constitute and define the scope for a CBS. The list of statements related to each of the CBS integration activity or method by phases is illustrated in Table 2. The survey was designed to gather data from the SE professionals who have experience of systems development and integration with a focus on the unique aspects of COTS activities that could influence the SI process differently compared to the integration of 'in-house' custom development. The survey data was collected primarily from the commercial industry sector software and systems projects where the impact of COTS on large long-term projects is clearly more evident given the variety of COTS products that are integrated and maintained over the life time of the project. The respondents were asked to rate each statement on a scale of 1–5 (1 being highly disagree and 5 being highly agree) with regard to their agreement that the statement provides a valid method or activity that aids in the integration of COTS products into existing systems. The survey consisted of 26 questions along with an introduction and a few questions on demographics of the respondents.

**Table 2** Activity or method statements covered in the survey

| *Number* | *Activity or method statement* |
|---|---|
| *Requirements definition and analysis* | |
| 1_1 | Stakeholder needs which require large product modifications are typically not good candidates for a COTS-based system. |
| 1_2 | A measure of a system's flexibility is its ability to adapt to the COTS capabilities. |
| 1_3 | To fully evaluate the applicability of COTS products, it may be necessary to use a trade-off analysis to align required capabilities with the capabilities of the COTS products. |
| 1_4 | The environment in which the COTS product is created – operating system, software languages, or tools used to create – provides systems developers insights into the applicability of the COTS product to the system under consideration. |

**Table 2** Activity or method statements covered in the survey (continued)

| *Number* | *Activity or method statement* |
|---|---|
| *Requirements definition and analysis* | |
| 1_5 | Developers of COTS-based systems must understand what standards the COTS components conform to prior to the design phase. |
| *Architecture/design process* | |
| 2_1 | It is important to address the issues with architecture compatibility related to COTS integration such as interoperability, control, timing and service provisions, and interconnectivity early in the systems development lifecycle. |
| 2_2 | System design practices that leverage modularity in the design are best suited for CBS architectures. |
| 2_3 | The practice of involving the SI team early in the CBS design process can provide an analysis of the complexity of the integration efforts before they become too costly to address. |
| 2_4 | Planning for technology refresh and insertion of the COTS components must be considered during the design phase. |
| *Development process* | |
| 3_1 | When considering the overall expenses incurred during the course of the systems development project, the costs of the component acquisition and integration must be considered. |
| 3_2 | A measurement consideration of time to market in a CBS systems development process is to measure the rapidity of incremental feature releases in comparison to projected 'in-house' development. |
| 3_3 | Throughout the CBS development phase it best to use an iterative process to continually assesses the market needs, evaluate the requirements adaptability to respond to critical changes, and analyse the system architecture's ability to integrate alternative solutions. |
| *Testing process* | |
| 4_1 | As COTS products are tested by the vendor, the testing process must include checks to ensure congruence between vendor testing and SI testing. |
| 4_2 | Hot box testing establishes an environment that replicates the operational system providing the benefit of early evaluation of the CBS's functional behaviours, performance, and standards compliance. |
| 4_3 | Interface management between the vendor and systems integrator can be a source of unexpected costs if unique tests are needed to ensure validation of requirements against the COTS product. |
| *Production process* | |
| 5_1 | During the production phase of the CBS, it is important to measure a vendor's response to system issues such as failures and changes in COTS product configuration. |
| 5_2 | The evolution of technology can create a configuration management challenge during the production phase that would impact the SI process of a CBS. |
| 5_3 | In the production phase of CBS, it is important to establish a proactive system release strategy, synchronising COTS upgrades with system releases. Planning the ongoing integration of evolving COTS products with an internally developed system helps ensure continuous operation. |

**Table 2** Activity or method statements covered in the survey (continued)

| *Number* | *Activity or method statement* |
|---|---|
| *Maintenance process* | |
| 6_1 | Technology refresh evaluations and analysis are required to support changes in the technology and standards for CBS development as part of the maintenance strategy throughout the lifecycle. |
| 6_2 | Maintainers of the system are responsible for determining whether to allow upgrades of COTS products or retain older versions. |
| 6_3 | COTS product capability and quality evaluations need to be managed as a continuing task during the maintenance phase. |
| *Vendor responsibility* | |
| 7_1 | COTS integration is not a one-time exercise and should be supported by the vendor throughout the lifecycle to ensure maintainability of the CBS throughout its operational life. |
| 7_2 | Having vendors as part of the integrated product team helps foster a more trusting partnership among the vendor, contractor, and the organisation. |
| 7_3 | Having knowledge of the marketplace provides insight into the business practices and future goals to support COTS integration. |
| *Marketplace considerations* | |
| 8_1 | Market research data provides a set of criteria to be used in the selection process for identifying the most suitable COTS products for integration. |
| 8_2 | COTS product maturities as well as the market place maturity are critical issues to understand when selecting a COTS solution. |

## 5 Research analysis and survey results

The validation data collected from the 25 respondents was analysed for the purposes of discussion covered in this section. The respondents experience ranged from analysing COTS requirements, evaluating multiple vendor products, architecting for COTS integration, quality assessment, integration and testing, program management, and field support. The average experience of these respondents was a little over 26 years. Figure 3 provides a breakdown of the respondent's current positions within the engineering community.

Of the 26 statements covered in the survey on integration CBS, the respondents agreed or strongly agreed with 76% of the statements. Figure 4 provides the percentage of agreement to the statements. While none of the respondents strongly disagreed to any of the statements, only 3% of statements were responded to with disagreement and 21% of the statements were marked as neutral.

**Figure 3** Respondents current position (see online version for colours)

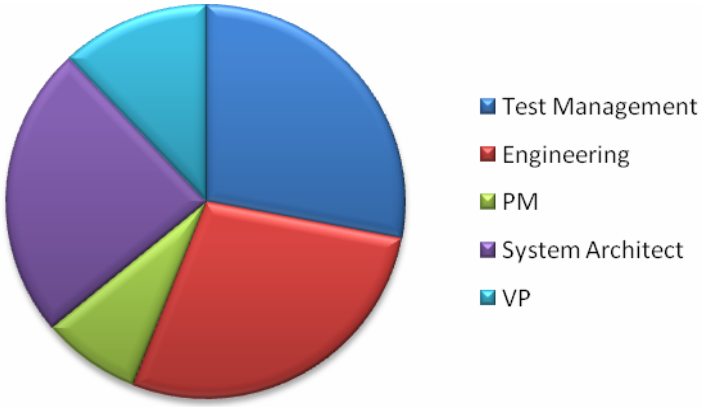
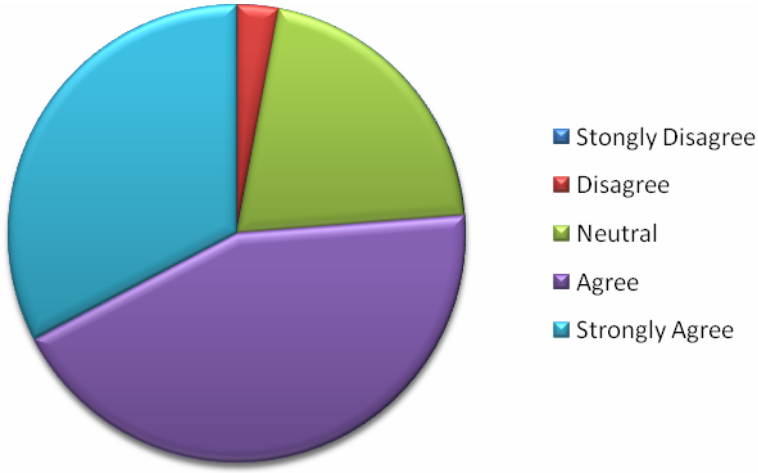



**Figure 4** Percentage of agreement for the CBS integration activity (see online version for colours)




The weighted means and standard deviations (SD) were calculated using weighted averages (–2 to 2 for strongly disagree to strongly agree) to determine any significant differences between the individual statements. The weighted means and SD can be reviewed in Figure 5. It illustrates the respondents' level of agreement to the statements where the closer to the centre, the higher the agreement (indicated by a high mean score) to the method – the smaller was the difference in their level of agreement (indicated by a small SD) across all the respondents. The results indicate that seven out of the 26 statements have a mean score above 7.0, therefore being most relevant across the respondents' community. Table 3 presents the seven statements and the associated phase within the SIFCBS that received the highest level of agreement.

Table 4 presents four statements that rated the least relevant in their application to SIFCBS. These statements received the highest number of respondents who marked the statements as either neutral or disagree. In discussions of the survey with the respondents

who were neutral towards the evolution of technology statement, they felt the question was poorly worded and that the evolution of technology impacts the SI process throughout the lifecycle and not just within the production phase.

For the other three statements, this could be a reflection of the types of respondents. Only the one programme manager and two executive level respondents agreed with the necessity of marketplace knowledge on the decisions of CBS.

**Figure 5** Survey results indicating respondents' agreement to statements (means and standard deviations) on integration of CBS (see online version for colours)

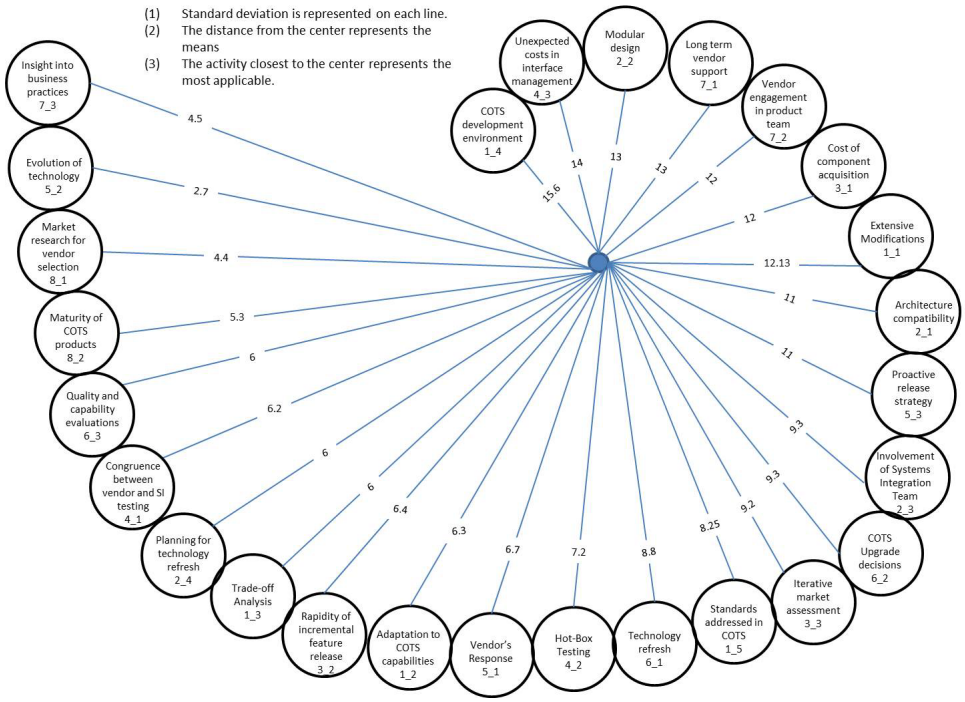



**Table 3** Most relevant statements within the SIFCBS

| *Phase* | *Statement* |
|---|---|
| Requirement definition and analysis | The environment in which the COTS product is created – operating system, software languages, or tools used to create – provides systems developers insights into the applicability of the COTS product to the system under consideration. |
| Testing | Interface management between the vendor and systems integrator can be a source of unexpected costs if unique tests are needed to ensure validation of requirements against the COTS product. |
| Architecture/design | System design practices that leverage modularity in the design are best suited for CBS architectures. |
| Vendor support | COTS integration is not a one-time exercise and should be supported by the vendor throughout the lifecycle to ensure maintainability of the CBS throughout its operational life. |

**Table 3** Most relevant statements within the SIFCBS (continued)

| *Phase* | *Statement* |
|---|---|
| Vendor support | Having vendors as part of the integrated product team helps foster a more trusting partnership among the vendor, contractor, and the organisation. |
| Development | When considering the overall expenses incurred during the course of the systems development project, the costs of the component acquisition and integration must be considered. |
| Requirement definition and analysis | Stakeholder needs which require large product modifications are typically not good candidates for a COTS-based system. |

**Table 4** Least relevant statements within the SIFCBS

| *Phase* | *Statement* |
|---|---|
| Vendor support | Having knowledge of the marketplace provides insight into the business practices and future goals to support COTS integration. |
| Production | The evolution of technology can create a configuration management challenge during the production phase that would impact the SI process of a CBS. |
| Marketplace | Market research data provides a set of criteria to be used in the selection process for identifying the most suitable COTS products for integration. |
| Marketplace | COTS product maturities as well as the market place maturity are critical issues to understand when selecting a COTS solution. |

## 6 Conclusions and future work

The specific nature of CBS influences the SI process aspects throughout the SI process cycle. Those aspects of SI for CBS have been identified and analysed to question certain facts related to CBS which distinguish traditional SI approach from customised SI approach for CBS. Within this paper, we have discussed critical aspects on: What are the unique challenges and issues, and factors affecting COTS integration? What are the advantages, disadvantages, critical success factors, best practices, managing vendor relationships, contractual agreements? What are the factors affecting SI process for CBS and how do they impact the complete SoSI process throughout the systems development lifecycle starting from concept through the maintenance and obsolescence phase?

While this exploratory study indicates a good beginning towards application of the practices defined the SIF, there is caution in drawing conclusions from the limited number of participants and industry experience. The value of the paper comes from its credible research in creating the framework for application of practices for successful CBS development project.

Future research work will build on the proposed SIFCBS specifically for system of systems application. The focus will be on identifying and implementing a measurement

system for each of the elements of the proposed framework. The SI preparedness for the CBS will be indicated through these metrics and utilised to determine the predictability and accuracy of SI planning, and related risks for COTS-based system development. Once the integration of the CBS is completed then the planned estimates can be compared with the actual achieved milestones and feedback can be iterated to improve the CBS integration predictability process and related risks.

## References

Abts, C. (2002) *COTS-Based Systems (CBS) Functional Density – A Heuristic for Better CBS Design*, USC Center for Software Engineering, Los Angeles.

Abts, C., Boehm, B. and Clark, E. (2000) *COCOTS: A COTS Software Integration Cost Model - Model Overview and Preliminary Data Findings*, pp.325–333, USCCSE, Munich, Germany.

Albert, C. and Brownsword, L. (2002) *Evolutionary Process for Integrating COTS-Based Systems (EPIC): An Overview*, Software Engineering Institute, Pittsburg.

Alves, C. and Finkelstein, A. (2002) *Requirements Negotiation for COTS-based Systems: Challenges and Open Issues*, pp.1–19, Elsevier, Essen.

Bansler, J.P. and Havn, E.C. (1994) *Information Systems Development with Generic Systems*, pp.707–718, Nijenrode University Press, The Netherlands.

Blanchette, J.R. (2005) *Pros and Cons of Using COTS Products*, pp.472–476, IEEE, Orlando.

Boardman, J. and Sauser, B. (2006) *System of Systems – The Meaning Of Of*, pp.118–123, IEEE, Los Angeles, CA.

Boehm, B. and Abts, C. (1999) 'COTS integration: plug and pray?', *IEEE Computer*, Vol. 32, No. 1, pp.135–138.

Boehm, B., Scherlis, W.L. and Wulf, W.A. (1995) *COTS and Systems Intgration: A Technical Perspective*, pp.14–19, Carnegie Mellon University, Pittsburg.

Bolloju, N. (2009) 'Conceptual modeling of systems integration requirements', *IEEE Software*, Vol. 26, No. 5, pp.66–74.

Brownsword, L., Oberndorf, T. and Sledge, C.A. (2000) 'Developing new processes for COTS-based systems', *IEEE Software*, July/August, Vol. 17, No. 4, pp.48–55.

Dawkins, S. and Riddle, S. (2000) *Managing and Supporting the Use of COTS*, Springer-Verlag, Southampton, UK.

Dong, J. et al. (2005) *A COTS Architectural Component Specification Stencil for Selection and Reasoning*, pp.1–4, ACM, St Louis.

Egyed, A., Muller, H. and Perry, D. (2005) 'Integrating COTS into the development process', *IEEE Software*, Vol. 22, No. 4, pp.16–18.

Garlan, D., Allen, R. and Ockerbloom, J. (1995) 'Architectural mismatch: or why it's hard to build systems out of existing parts', *IEEE Software*, Vol. 12, No. 6, pp.17–26.

Giangarra, P. and Semple, B. (2011) *A COTS (Commericial off the Shelf) Based Approach to Plug-an-Play Launch Control Systems*, pp.152–158, IEEE Computer Society, Palo Alto, CA.

Grady, J.O. (1994) *Systems Integration*, 1st ed., CRC Press, Boca Raton.

Gunderson, C.R. and Minton, D.H. (2011) *Rapid Evolutionary Acquisition – An In-Progress Review of an Exemplar Pilot Initiative*, pp.19–24, IEEE, Albuquerque, New Mexico.

Jain, R., Chandrasekaran, A. and Erol, O. (2009) 'A framework for end to end approach to systems integration', *International Journal of Industrial and Systems Engineering*, Vol. 5, No. 1, pp.79–109.

Jain, R., Chandrasekaran, A. and Erol, O. (2010) 'A systems integration framework for process analysis and improvement', *Systems Engineering Journal*, Vol. 13, No. 3, pp.274–289.

Jain, R., Chandrasekaran, A., Elias, G. and Clouter, R. (2008) 'Exploring the impact of systems architecture and systems requirements on systems integration complexity', *IEEE Systems Journal*, Vol. 2, No. 2, pp.209–223.

Kang, C.M., Hong, Y.S. and Huh, W.T. (2012) 'Platform replacement planning for management of product family obsolescence', *IIE Transactions*, 04 April, Vol. 44, No. 12, pp.1115–1131.

Kohl, R. (2001) *Changes in the Requirements Engineering Processes for COTS-based Systems*, IEEE Computer Society, Toranto.

Lewis, P. et al. (2000) *Lessons Learned in Developing Commercial Off-The-Shelf (COTS) Intensive Software Systems*, FAA, Washington DC.

Maier, M.W. (1998) 'Architecting principles for systems-of-systems', *The Journal of Systems Engineering*, Vol. 1, No. 4, pp.267–284.

McCabe, R. and Pollen, M. (2004) *Evaluating Architectures with Systems Attributes*, Systems and Software Consortium, Herdon.

Money, A. and Gansler, J. (2000) *Commericial Item Acquisition: Considerations and Lessons Learned*, Office of the Secretary of Devense, Washington DC.

Morisio, M. and Torchiano, M. (2002) *Definition and Classification of COTS: A Proposal*, pp.165–175, Springer-Verlag, Orlando.

Morisio, M. et al. (2002) 'COTS-based software development: processes and open issues', *The Journal of Systems and Software*, Vol. 61, No. 3, pp.189–199.

Redmill, F. (2004) 'Analysis of the COTS debate', *Safety Science*, Vol. 42, No. 5, pp.355–367.

Reifer, D., Basili, V., Boehm, B. and Clark, B. (2003) 'Eight lessons learned during COTS-based systems maintenance', *IEEE Software*, Vol. 20, No. 5, pp.94–96.

Sage, A.P. and Lynch, C. (1998) 'Systems integration and architecting: an overview of principles, practices, and perspective', *The Journal of Systmes Engineering*, Vol. 1, No. 3, pp.176–227.

Sametinger, J. (1997) *Software Engineering with Reusable Components*, 1st ed., Springer-Verlag, Berlin.

Sankaran, K., Kannabiran, G. and Dominic, P. (2011) ,Determinants of software quality in COTS products: an exploratory study', *International Journal of Business Information Systems*, Vol. 8, No. 1, pp.4–22.

Sedigh-Ali, S., Ghafoor, A. and Paul, R. (2001) 'Software engineering metrics for COTS-based systems', *IEEE Computer*, Vol. 34, No. 5, pp.44–50.

Sivzattian, S. and Nuseibeh, B. (2001) *Linking the Selection of requirements to Market Value: A Portfolio Based Approach*, pp.202–213, Interlaken, Essener Informatik Beitrage.

Sjoberg, E.S. and Harkness, L.L. (1996) *Integrated Circuit Obsolescence and its Impact on Technology Insertion Definition for Military Avionics Systems*, pp.792–799, Institute of Electrical and Electronics Engineers, Dayton, OH.

Tomita, K. et al. (2008) *The Schemes to Develop Dependable System Using COTS*, pp.32–39, IEEE, Taipei, Taiwan.

Verma, D. and Plunkett, G. (2000) *Systems Engineering and Supportability Analysis: Technology Refreshment for COTS-Intensive Systems*, Lockhee Martin, Minneapolis.

Voas, J. (2001) 'Faster, better, cheaper', *IEEE Software*, Vol. 18, No. 3, pp.96–97.

Yakimovich, D. and Basili, V. (1999) Software Architecture Classification of Estimating the Cost of COTS Integration, pp.296–302, ACM, Los Angeles.