

# Life Cycle Activity Areas for Component-Based Software Engineering Processes

Robert C. Seacord  
Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213 USA  
+1 412-268-3265

Kingsley C. Nwosu  
Lucent Technologies, Inc.  
600-700 Mountain Ave  
Room 3D-435  
Murray Hill, NJ 07974  
+1 908-582-7131

## Abstract

*Although traditional software engineering focuses on development Component-based software engineering (CBSE) processes must focus on integration. In this paper, we elaborate on this focus into a process framework for CBSE consisting of four major activity areas: engineering, business, management, and overarching. We show how these activities are concurrent with respect to an iterative and incremental development model. Detailed discussions are also presented on the consequent issues, concerns, problems, and recommended solutions..*

## Introduction

Over the years, it has been the ambition and goal of the Software Engineering community to quickly build a software system from components that have been developed outside of the development organization. This has been motivated by similar abilities in the companion hardware areas resulting in associated benefits such as shortened time to market and increased productivity. Due to developments in several areas of the industry and technology, that long cherished ambition is coming to fruition. Today, there are commercial off-the-shelf (COTS) products that perform functions that previously required custom-built software components.

Component-Based Software Engineering (CBSE) process departs from the conventional software development process in that it is integration-centric as opposed to development-centric. A true CBSE process depends largely on selection, acquisition, and integration of components obtained from external vendors. While this is a positive trend, there are numerous issues, questions, and risks involved in component selection, acquisition, and integration that must be adequately addressed and resolved in a CBSE project.

Multiple process models exist today, primarily focused on the building of custom systems. The exact nature of the relationships between traditional build process models and process models that incorporate the differences imposed by CBSE are unknown. There is an increasing

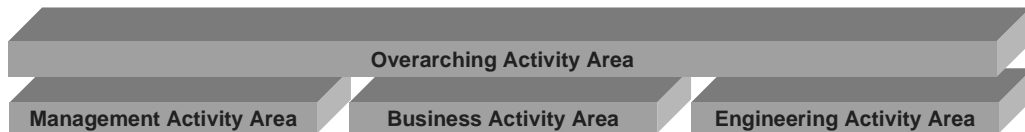
belief that existing process models are inadequate to address the unique CBSE concerns of development as primarily an integration effort as opposed to a custom development. Some of the issues to be determined include what the appropriate process model for CBSE should look like.

In [Oberndorf 98a], Oberndorf, Sledge, Brownsword developed a *process framework* that defines the activity areas that cover the development and evolution of COTS-based systems for government programs. In this paper we adopt the broad outlines of this process framework and discuss the needs of component-based systems within this context. The exact nature of activity areas described in this paper differs in some regards from the COTS-based systems process framework defined in [Oberndorf 98a].

In our discussion of the process framework, we first present an overview of the integrated relationship between the different activity areas within the development life cycle. Secondly, we discuss the different activity areas, and the issues within each activity area, followed by our conclusions.

## Development life cycle overview

The CBSE process framework consists of four system activity areas: engineering, business and management, and overarching. The *engineering activity area* covers activities associated with the technical conceptualization, construction, and sustaining of a system (hardware, software, and people). The *business activity area* includes activities associated with developing a business case for a component-based system, determining business process implications, and developing cost estimates. The *management activity area* covers activities that a project manager is directly responsible for such as cultural transition, information sharing, and CBSE policies. The overarching activity area covers organizational activities, or activities that may be common to several projects and are accordingly better addressed at an organizational level. Figure 1 shows the relationship of the major activity areas with respect to each other.

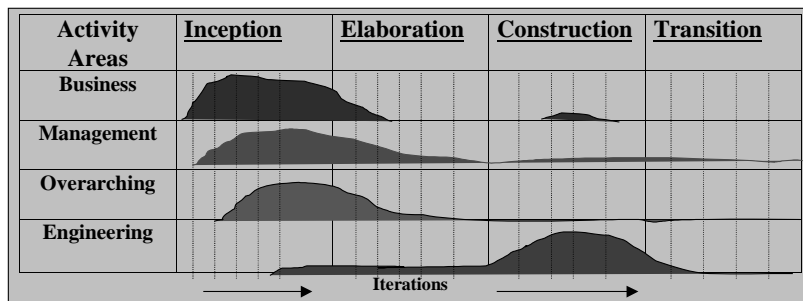


**Figure 1. Activity areas**

The activity areas are not an ordered sequence of phases, rather they involve iterative and incremental development activities that usually characterize and are often needed to accommodate the uncertainty that exists with most CBSE projects.

Incremental development consists of a series of iterations involving activities from one or more activity areas, the successful completion of which results in the delivery of a final system. To elaborate these incremental life cycle activities, we adopt the Rational Objectory Process model [Quantrani 97] that structures the development process along two dimensions - division of the life cycle into phases and iterations (time) and production of a specific set of artifacts with well-defined activities (process components). The time dimension consists of the inception, elaboration, construction, and transition phases. The inception phase specifies the project vision; the elaboration phase defines the planning and architecture specification and design. The construction phase builds the product incrementally, while the transition phase does the manufacturing, delivering, and training. In our case, we use the same time

phases, but the process components are the activity areas identified above. As a result, Figure 2 shows the application of the Rational Objectory Process model on the proposed framework development process.



**Figure 2. Development life cycle phases and iterations**

As shown in Figure 2, successive iterations are required to complete each development life cycle phase. The inception phase, for example, comprises successive iterations of activities in the business, management and overarching activity areas. Specifically, these include business case analysis, cost estimations, addressing cultural issues, defining policies, and risk management. During the construction phase, these iterations are principally of activities in the engineering area.

### Activity areas

The activity areas should be considered as a notional model that would be used to guide the detailed planning of a specific development project. Depending on the particular needs of a project, some activity areas would have greater emphasis than other areas.

### Overarching activity area

The overarching activity area covers organizational activities, or activities that may be common to several projects and are accordingly better addressed at an organizational level such as managing vendor relationships. Examples of overarching activities include training, knowledge sharing and licensing. Although many of these activities correspond to activities within individual projects, they are on a significantly different level and have a different focus. For example, the licensing of a component or components at an organizational level may have a significant impact on the organization. Often an organization can negotiate a significantly better price on a component or component-line. This can assist projects in providing a core set of components they can use at a reduced cost. However, this approach may also have a negative impact by leading projects to select components that may not be the best fit for the particular needs of the project. Components may be selected because they are available at a reduced cost, particularly if project allowances for licensing components are cut as a consequence of spending at the organizational level.

The most important activity in the overarching activity area is the establishment of an effective component-based strategy. This strategy can be represented in very broad terms or in very specific guidelines. Each approach has risks and benefits. In the Clinger-Cohen act, the federal government strongly encouraged the use of COTS in government acquisitions [Oberndorf 98b]. This has been interpreted in many different ways, often with adverse effect

to the programs involved. This is illustrative of the risks involved in defining strategy in broad terms.

An alternate approach is to provide very specific guidelines. For example, an organization may go as far as to select a specific component model or framework and require projects to adhere to this framework. An organization may make a strategic decision, for example, to use only Microsoft COM components. This decision may have advantages in providing common components that can be re-used in different projects throughout the organization, and in having a common knowledge base that can be shared between development groups. The disadvantages, again, comes from encouraging projects to adopt technologies that may not be appropriate to the particular needs of the project. Strategic decisions are often made for reasons that have little or nothing to do with technology issues. However, it is important that the technical consequences of these decisions are understood.

### **Engineering activity area**

The engineering activity area covers activities associated with the technical conceptualization, construction and sustainment of a system. CBSE has significantly different characteristics from traditional, custom system development. In particular, the use of system architectures and frameworks take on additional importance in a CBSE process. In fact, with the advent of Enterprise JavaBeans and the continued evolution of CORBA architectures can be viewed as components and be selected through an evaluation process. The characteristics of the architecture vary between domains, but in general the architecture attempts to ensure system characteristics that need to be handled in a common manner over multiple components. One example of this can be distributed transaction processing, where multiple components may implement operations that are part of a larger transaction that may need to be completed or rolled back as a whole. Problems such as these are intractable without a consistent architecture.

Mismatches between an architecture and components often need to be addressed before a component can be successfully integrated. Architectural mismatch [Garlan 95] can often be addressed by implementing a wrapper, bridge or adapter for the mismatched component. Rarely should the architecture be altered to suit the needs of a specific component, unless this change can then be generalized and consistently applied to all components that comprise the system.

### **Business activity area**

The business activity area includes activities associated with developing a business case for a component-based system, determining business process implications, and developing cost estimates. A component-based software engineering effort often introduces new concerns that are not at issue in more traditional custom development approaches. Software licensing issues, for example, often dictate the system design, as certain configurations may be prohibitively priced.

### **Management activity area**

The management activity area covers activities a project manager is directly responsible for such as budgeting and risk management.

In a component-based development effort, the budget is distributed between human resources and the purchase of licenses for software components. In a custom development

effort, the development effort can be sized and an appropriate development team can be staffed for the necessary period to complete the project. In CBSE, a tradeoff exists between purchasing and integrating commercially available components and the development of custom components. It is often impossible to determine in advance the requirements in each of these areas until the process is sufficiently advanced. For example, it is normally impossible to determine if a particular component can be used to provide a required capability until that component has been evaluated. In this case, it is impossible to determine if funds should be allocated for the purchase of the component or for a development effort to create a custom component until the evaluation is completed. An advanced CBSE process in this area will provide for maximum flexibility in exchanging development and licensing resources.

Risk management exists in both a custom development effort and in a component-based software engineering effort, although the nature of the risks is often significantly different and must be understood and accounted for in a risk mitigation plan. CBSE must provide for continued evaluation of products so that the integration team can understand how market changes may affect their efforts.

## **Conclusions**

In this paper, we have discussed a number of activity areas that form a process framework for component-based software development. We presented the concurrency between the activities of these activity areas with respect to an iterative and incremental development model. Detailed discussions are also presented on the issues, concerns, problems, and recommended solutions to the different major activity areas. As alluded to in several places in this paper, a true realization of a CBSE project involves addressing and solving numerous issues, many of them presented here, that are novel to component-based software engineering. All the issues presented here do not necessarily apply to every CBSE project, however, every CBSE project will be faced with many of these issues.

## **References**

1. Garlan, D., Allen, R., and Ockerbloom, J.: Architectural Mismatch or Why It's Hard to Build Systems Out of Existing Parts, Proceedings of the International Conferences on Software Engineering, Seattle (1995)
2. Oberndorf, P., Sledge, C., Brownsword, L., COTS-Based Systems for Program Managers, Software Engineering Institute, Carnegie Mellon University, briefing.
3. Oberndorf, P., A Clarification of DoD COTS-Related Policies," , Software Engineering Institute, Carnegie Mellon University, briefing. Available WWW: URL: [http://www-preview.sei.cmu.edu/cbs/cbs\\_slides/index.html](http://www-preview.sei.cmu.edu/cbs/cbs_slides/index.html)
4. Visual Modeling with Rational Rose and UML, Terry Quattrani, Addison-Wesley, 1997, ISBN:0-201-31016-3.