Implementation of CBSE in Small Businesses

William T. Councill
Mannatech, Inc.
600 South Royal Lane
Coppell, TX 75019
bcouncil@mail.mannatech-inc.com

A position paper is proposed for the Second International Workshop on Component-Based Software Engineering

Introduction

The U.S. Small Business Administration defines small businesses as having fewer than 500 employees. Over 99 percent of all U.S. businesses are small businesses, and from 1992 to 1996, small businesses created all of the net new jobs [1]. Yet, reference to the literature on software engineering suggests strongly that implementation of software engineering process models, such as the Software Engineering Institute's Capability Maturity Model (CMM) and ISO 9000, occur among large organizations and on Department of Defense projects.

Additionally, object-oriented and componentware methodologists—such as Grady Booch, Ivar Jacobson, Bertrand Meyer, and James Rumbaugh, among others—generally conduct consulting assignments that eventuate in their articles and books within organizations having more than 500 employees. Numerous small, independent software vendors (ISVs) and information technology (IT) organizations operating in small businesses will adopt component-based development or solutions.

The benefits of software engineering are undeniable. Likewise, once component-based development (CBD) is skillfully adopted, and after the appropriate infrastructure is established, the summary of the first workshop on component-based software engineering [2] instructs that some goals of CBSE are to:

- Enhance reuse of core functionality across applications;
- Permit the flexible upgrade and replacement of parts of software independent of their production (manufacture); and
- Provide the means for establishment of collective organizations' best practices so that extant processes may be replaced because of changing business or market-driven scenarios.

Many, if not most, of the business components developed for numerous industries will be developed by small businesses as defined by the U.S. Small Business Administration. The issue is this: How can small business software organizations assure quality and trusted components that comport with all established or prevailing standards, whether those components are internally developed or purchased from third parties?

Software Engineering and Small Business

Small companies, especially those in the range of 200 to 500 employees, often develop complex software to assure their companies' competitive positions. This software is often perceived as mission-critical; that is, businesses cannot succeed without the software. Software engineering in small companies, especially entrepreneurial organizations, must realize the needs of continually more erudite market experience. To survive, many small companies interpret market changes as requirements for modifications to business processes and business rules, thus necessitating frequent, rapid revisions to the companies' core software.

Software engineering processes, including the CMM, generally are based on experiences with DOD contractors and large corporations for whom change occurs more slowly than in small companies. In large organizations that adopt change management, change is perceived as a deliberate, methodical process. In small businesses, change is considered a "make-or-break" situation; the competitive window of opportunity is perceived—often rightly so—as narrow and one that must be traversed immediately.

Nevertheless, software engineering is required to assure consistent access to the software. Strict software engineering guidelines--for example, adherence to the key process areas (KPAs) of the CMM--are considered necessary to assure, at a minimum, a repeatable, controlled, and well understood, software development process. Many small companies, especially those that do not function as subcontractors for DOD projects, consider software engineering a hindrance to efficient, realizable development of software.

Additionally, the roles identified by Paul Allen in his article, "Planning Team Roles for CBD," [3] are daunting for small businesses. Allen identifies an array of roles that, ideally, should be assumed by members of any component development team and support personnel to achieve optimal results [4]. That article relies greatly on work published by the Dynamic Systems Development Method consortium (DSDM) [5].

The roles are effectively delineated, and Allen states that roles may be assumed by various employees at different times. The problem is that many small organizations have insufficient numbers of employees to assure that employees assigned to various roles will function in each role well. For example, an experienced configuration management engineer likely will have assumed too many existing roles to learn to function effectively in the new role of reuse librarian.

Budgets in many small companies, despite the commitment to component technologies, will not support the recruitment of a reuse librarian. Yet, many small companies necessarily will adopt CBD. How can CBSE develop so that small companies can assure themselves, as well as purchasers of their components, of the confidence and trust in their components?

CBSE and Small Business

CBD for small businesses makes sense. Many small businesses are unlikely to purchase enterprise resource planning (ERP) software because of the great expense and significant number of personnel required for implementation. Small companies, for example, may anticipate the purchase of a third party order processing component to build into a proposed new application. Or, such companies may develop a business component and, to further fund their IT functions, likely will develop generic versions of the business component for resale to other, non-competing companies with similar business processes.

Nevertheless, purchasers of third party componentware must trust their business processes to components developed by others. Components can only be trusted when, especially considering components developed using object-oriented techniques, the following requirements converge:

1. The architecture demonstrates adherence to the basic laws and principles of object-oriented analysis, design, and programming [6]:

    a. The Law of Demeter – Avoidance of coupling a client to knowledge of indirect objects and the internal representations of direct objects.

    b. Principle of Low Coupling – Assurance that a class is not dependent on too many other classes. Enhances reusability.

    c. Principle of High cohesion – A measure of how strongly related and focused the responsibilities of a class are. Enhances maintenance, comprehension, and reusability.

2. Services can be accessed only through a consistent, detailed, published interface that represents a contract between the provider of the capability and potential consumers. This is routinely referred to as component encapsulation.

3. New interfaces can be defined to service new requirements with minimal disruption to the component's internals and to component consumers.

4. CBSE is demonstrable through:

    a. Repeatable processes (at a minimum, as defined by CMM, Level 2);

    b. Documentation of all phases of the software development life cycle;

    c. Configuration management of code; and

    d. Separate, verifiable component management, with automated search capabilities on metadata.

The Challenge

Components will require a great degree of trust; otherwise, as Bertrand Meyer states, "the spread of less-than-optimal components could lead to a *worsening* of the [software development] situation [7]. In that article he refers to another of his articles, concerning the Ariane project, where reuse of an improperly specified component (module) created an industrial disaster [8]. Code libraries from sources other than compiler vendors generally will be reviewed with suspicion; the code is likely to be readily investigated. The question that must be answered by Workshop participants is: Against what standards will components with an acceptable, published interface be adjudged, thereby permitting the purchaser a reasonable sense of trust?

The author proposes that the admonition "Caveat Emptor" ("Let the buyer beware") is an insufficient standard and one that promotes lawsuits in the event the purchased software does not work as published. Rather, it is recommended that all vendors of business components establish a warranty attesting that basic software engineering principles have been adopted for the general design, construction, and testing of the component. Since many business components, it is proffered, will be developed by ISVs and other small businesses, how can the quality, merchantability, and "fitness for a particular purpose" be assumed, without recourse to the courts?

The Butler Group has introduced the concept of CBD levels of maturity [9]. The issues are similar to those for the CMM. However, what is the assurance that ISVs and small business IT organizations will adopt the proprietary model, or even become aware of the Butler Group's attempts to propose such a model?

The author proposes for the Workshop's consideration an approach for certification of components analagous to the international Underwriter's Laboratories, Inc. model. For those vendors that volunteer to participate in component certification, publication of compliance, as well as all known defects, should be prescribed. The organization's approach to software engineering, and especially the testing of components, likewise would be published.

A separate standards body would establish minimal standards for acceptable types of defects and the numbers of errors in their internals, interfaces, and metadata. Components then could be tested in UL-type certification laboratories according to standards promulgated by approved national or international standards' organizations. Rigid testing would assure vendors' or publishing users' intent to offer for consumption a trusted component that is reliable and reusable.

Summary

Small businesses require implementation of software engineering processes to assure continuous access to reliable software. Yet, the business dynamics of small businesses necessitate a review and likely revision of software engineering practices that support their needs for frequent, rapid change. The development or purchase of reliable, reusable components has the potential to assist small businesses with dynamic and sudden business, and therefore, software change. Component technology can assure quality software, but only once standards for trusted components are established and testing facilities are established for those organizations, both small and large, that desire to submit to the voluntary process of certifying their components.

References

[1] Small Business Administration, "Small Business Profile, 1998."
www.sba.gov/ADVO/stats/profiles/98us.html

[2] Brown, A. W. and K.C. Wallnau, "An Examination of the Current State of CBSE: A Report on the ICSE Workshop on Component-Based Software Engineering," 1998.
www.sei.cmu.edu/cbs/icse98/summary.html

[3] Allen, P. And S. Frost, "Planning Team Roles for CBD," *Component Strategies*, (August 1998).
www.selectst.com/downloads

[4] Allen, P., "Practical Strategies for Migrating to CBD," *Cutter IT Journal*, Vol. 11, No. 12 (December 1998).

[5] DSDM Consortium, *DSDM Version 3*, Tesseract Publishing, Farnham, UK, 1997.

[6] C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*, Prentice Hall PTR, Upper Saddle River, NJ, 1998.

[7] Meyer, B., C. Mingins, and H. Schmidt, "Trusted Components for the Software Industry."
www.eiffel.com

[8] Jezequel, J-M. and B. Meyer, "Design by Contract: The Lessons of Ariane." www.eiffel.com

[9] Wilkes, L. and D. Sprott, "Understanding Component-Based Development Through the Butler Group Maturity Model: What Level Are You At?" *CBD Forum Journal* (June 1998).