

Software Reuse – an Overview

Ewak Creative Compusoft Limited
A1&A2, Block AA101, First Street,
Anna Nagar,
Chennai 600040
Ph: 91-44-2621-1545
www.ewaksoft.com



Software Reuse

This is an article basically written for Software companies and the managers to considerably increase the reuse of the Software assets created by them.

Software assets can be converted as components to enable the programmers, managers to use those components as and when required. This is similar to the way a Personal Computer is assembled using pre-manufactured components. Likewise a software can also be assembled using the pre-assembled software components.

For software business houses developing an application, achieving it at a cost effective manner, with the best of quality and with a faster time to market is a top priority. In most cases, component based development and reuse of the software would be a more effective strategy.

Why reuse

Software Reuse is the process of creating software systems from predefined software components. Software reuse has two sides:

- (1) the systematic development of reusable components.
- (2) the systematic reuse of these components as building blocks to create new systems.
- (3) Transformation of the Client Server legacy application into Software components

A reuse software need not only be code , but also designs, tests cases, data, prototypes, plans, documentation, frameworks, and templates can all be material for reuse.

Some major advantages of software reuse could be

- (1) Increased productivity.
- (2) Shorter development time.
- (3) Develop software with fewer people
- (4) Reduction of Defect density
- (5) Quicker time to market
- (6) Reduced maintenance costs
- (7) Produce more standardized software
- (8) Produce better quality software
- (9) Provide a powerful competitive advantage

While comparing software systems, it is usually found that over 50 % generality between each software application. This includes code, design, functional and architectural similarities. Some software similarities can be predefined and built into software tools (such as reusable code patterns in generators); others can be created as reusable components, which are stored in software, reuse libraries.

The potential for reuse is huge since most of the new applications can be assembled from reusable components rather than reinventing from scratch. The same applies to

	6 th February 2004
Article – Software Reuse	Page 2 of 5

your time-tested Client Server Application. It makes more business and economic sense to migrate those applications as software components , so that they can be deployed in future developments be it Client Server or a Web Based application. It has also to be considered that development of component architecture comes with its own set of challenges than developing a Client Server architecture.

Since the last 30 years , many Commercial enterprises such as Toshiba , Hughes, Boeing Aircraft Company Command and Control Systems Division , Celcius Tech Systems AB in Sweden , Fujitsu Electric Switching Divisions etc have been practicing reuse of their technologies. There has been a significant savings in time and cost as a result of Reuse.

Many authorities have quoted a significant reduction in costs as mentioned

- (1) Time to market: reduction of 2 to 5 times
- (2) Defects : reductions of 5 to 10 times
- (3) Maintenance Costs: reductions of 5 to 10 times
- (4) Overall software development costs: reductions of 15 to as high as 75 times for long term projects(Which includes the overhead cost of developing reusable assets and supporting their use).

Some Points of discussion in software reuse

- (1) A component has to be used a minimum 3 to 5 times in application projects to recover the initial cost of creating and the future support costs.
- (2) It may take two to three product cycles before the benefits of reuse is gained.
- (3) Your existing client Server application can be migrated as components for further reuse.
- (4) There will be a substantial reduction in costs by migrating your existing Client Server application as components. The costs are substantially lesser compared to fresh development.
- (5) The same components can be reused in fresh client server application as well as in fresh development of new N tier application.

Systematic Institutionalizing Reuse

Software Reuse works best when applied above the single system level where there are more opportunities to reuse components. The migrated components and to get the ROI from the investment in reuse. Ideally, reuse programs should extend across multiple systems and projects.

A systematic reuse process is the creator of reuse components and the other is the re-user of the components over a number of projects.

	6 th February 2004
Article – Software Reuse	Page 3 of 5

A third entity is that of the support and maintenance of the reuse components. A powerful system to be placed for managing the reuse component to maintain the components as well as for version control. A simple tool or utility can be written for the same.

Strategic Business Systems Planning for migration and Reuse

The primary objective of strategic business systems planning is to align systems planning with business goals.

In the plan, each proposed system is described in terms of what business needs it fulfills and prioritized in terms of the strategic business benefits it can provide.

Management commitment and support of system projects is easier to obtain when the business value of a system can be clearly demonstrated to management. Likewise, the support for reuse is easier to obtain when reuse is linked to business strategies. Only those components that are likely to provide a return on the investment for making them reusable should be the focus of reuse. The guiding principle is that economic/business analysis should be part of reuse.

Migrating the client server components and Reusing them is more of a business strategy and must be market-driven in the sense that the business develops components for reuse that can be used to develop future systems.

Reuse of Migrated Client Server components.

1. Migration and reuse.

Software design takes place in an environment where a number of potentially reusable components are available. Reusable components are not normally produced in a normal project development. We have to put more effort to reuse these components. The objective is to make those pieces of code as reuse components.

The more practical approach would be to migrate your client server application as a n-tier application using the business logic and re-architect the components for future reuse whenever required.

2. Ewak's Role in helping you reuse your components.

Ewak has developed a process and methodology where we enable business enterprises like yours , to reuse your very valuable software assets.

	6 th February 2004
Article – Software Reuse	Page 4 of 5

We study your code, understand your business logic, extract the code, re-architect and convert the code as components. You can reuse these components in re-deploying them in fresh developments as well as use the same while architecting web applications.

Ewak's methodology and process objective is to produce components, which are potentially reusable. They should be easy to reuse because they have been designed for reuse. Such developed components can be reused over a number of projects and in various situations

They are globalized for reuse with ensuing improved paybacks including better quality, reliability, and faster production and at a lower cost. As we are migrating the same code, the cost incurred is considerably lower.

While migrating the components for reuse, there are many stages we follow which starts from identifying an application domain, identification & classification of reusable abstractions, domain-oriented reuse, design components, assessment for reuse, and deliver potentially reusable components

Focusing Reuse on the Future, using the Past

Using a strategic plan to direct its reuse strategy by the migration and reuse approach, an enterprise can assure that its reuse efforts are focused on identifying and building the "right" reusable components using the existing code and the components that can be used as building blocks in developing new systems that are of the greatest business value to the enterprise now and in the future