Wang

A few components can be managed by hand. But it

will become difficult to manage thousands of

components manually. CBSD process includes

requirement analysis, retrieving components, getting

components, and integration of applications software.

Retrieving and getting components is a process of

communicating with domain component library (asset

library).

Reusable asset specification (RAS) [7] is an Object

Management Group standard to describe and package

asset. RAS is described in two major categories, Core

RAS and Profiles. Core RAS represents the

fundamental elements of asset specification. Profiles

describe extensions to those fundamental elements.

The Core RAS is not instantiated therefore an asset

must be of a particular profile. A profile may extend

Core RAS or may extend another profile. This Default

profile is a realization of the Core RAS. The Default

Component Profile and Default Web Service Profile

derive from the RAS Default Profile.

Shreiff

Some of the barriers are directly related to the

complexity of the reuse technology. These revolve

around three factors: the process involved in building

reusable assets, the reusable assets themselves, and the

support available for utilizing and maintaining the

assets.

The major technical barrier pertains to difficulty of

locating reusable assets. Software repositories, while

not considered a success factor [17] are believed to be

responsible for dissatisfaction with reuse. The structure

of the repository may make it hard to locate an

asset.

Barriers manifested themselves during the three

stages of the reuse life cycle. The main ones during

the early stages of asset creation were lack of reuse

policies to govern the creation of reusable assets, lack

of funding, immature technology, and lack of reuseoriented

education and training. In the asset utilization

phase, reuse stakeholders complained about difficulty

in accessing an asset, limited marketing of assets

available in the repositories, and lack of support for

the integration of assets during application development.

Though stakeholders at all five sites acknowledged

an ongoing effort to manage assets, they

reported the same barriers that emerged during the

creation of assets for the limited role of asset management

Proposition 9 (Asset utilization—accessibility). The

lack of a well-organized and indexed software repository

with appropriate search engines is a barrier to

reuse adoption.

The problem at several sites was that developers

were unaware of assets at an organizational level and

relied heavily on word of mouth to find them. They

asserted that the lack of software repositories and the

lack of ways to promoting reusable assets constituted

significant barriers. At CBS-TCC and POM-OGC the

main problem was the lack of a central repository with

a good search mechanism and catalogue system.

Developers developed an attitude that: ‘‘it was more

trouble to find and use than to build on your own.’’

At SCC, asset utilizers complained they were not

aware of reusable components availability despite a

central on-line index. They commented that: ‘‘it is the

asset creators’ responsibility to inform us of what is

there,’’ and provide tools to make it easily accessible.

They suggested a knowledge-based repository.

At STAR II, the accessibility of the assets was not

felt to be a problem. Developers said they believed that

assets could be easily found within the repository

because the library was categorized under domains;

also web pages that search components in the library

have been published. The group also provided a use

scenario to help asset utilizers understand what each

component did and how it did it.

Lucredio

A reuse repository is a collection of reusable assets with

requirements such as search and retrieval mechanisms (Mili

et al., 1998). The use of a repository as an essential factor

for reusability has been target for debates. Frakes and Fox

(1995) and Morisio et al. (2002) showed that the construction

and use of repositories is important, but not fundamental

to the success of reuse programs. In this research,

we analyzed the use of component repositories in the software

organizations, aiming to analyze their impact on

reuse programs.

Analyzing the survey, we identified that the existence of

component repositories does not assure the success in reuse

levels. In fact, we observed the opposite effect: the percentage

of success in organizations without a repository was

superior to the ones that have one.

We verified that the organizations that had repositories

considered mainly: (i) source code; (ii) software components;

(iii) document models; (iv) reports and screen generators;

and (v) data base queries.

The reasons for the success in organizations with component

repositories were not identified, so a subsequent

study is necessary. However, it was identified that some

software organizations store their assets in simple structures

like Concurrent Versions System (CVS), without an

efficient search mechanism. It is possible that this may have

contributed to failure in the efforts of promoting reuse.

Based on these observations, we conclude that the existence

of a reuse repository has no influence on software

reuse success.

Frakes-Kang

A reuse library consists of a repository for storing reusable

assets, a search interface that allows users to search for

assets in the repository, a representation method for the

assets, and facilities for change management and quality

assessment. Much research on reuse libraries has been done

as reported in the papers in the reuse roadmap. Key ideas

are the application of indexing methods such as free text

keyword and faceted classification to reusable components.

There has been disagreement in the reuse research

community about the importance of libraries for reuse.

However, failure modes analysis of the reuse process shows

that in order to be reused a component must be available,

findable, and understandable. A reuse library supports all

of these. The argument has also been made that most

component collections are small and, therefore, do not need

sophisticated library support. However, the emergence of

the World Wide Web as a defacto standard library of

reusable assets argues against this point of view.

Experiments on reuse libraries indicate that current

methods of component representation could be improved.

Frakes-Fox

A reuse repository is a collection of reusable assets, along with a searching machanism for locating assets meeting development needs. The importance of a repository for promoting reuse has been the subject of debate. Many organizations have considered a repository central to ther reuse efforts, and many kinds of repository mechanisms have been reported. On the other hand, Tracz has agured that repositories are not of critical importante for reuse.

We examined the impact of having a repository on organizational code reuse levels, with the results shown in Figure 12. We found organisations with a repository have median code reuse levels 10 percent higher than organisations that do not have reuse repositories, but this difference is not statistically significant at the 0.05 level. Our analysis does not take into account what type of repository or what type of indexing was used.

We conclude that having reuse repositories does not improve levels of code reuse. Organizations trying to improve systematic reuse should probably not focus on repositories in their improvements efforts, at least initially.