RESEARCH ARTICLE OPEN ACCESS

STUDY OF REUSEABILITY OF SOFTWARE

Asfiya Shireen Sheikh Mukhtar[1], Meenal Meshram[2], Shubhum Mahatme[3]
(MCA Department, Smt. Radhikatai Pandav College Of Engineering & Technology, Nagpur)
(Email: asfiyashireen@gmail.com)
(MCA Department ,Smt. Radhikatai Pandav College Of Engineering & Technology, Nagpur)
(Email: meenalmeshram.it@gmail.com)
(MCA Department, Smt. Radhikatai Pandav College Of Engineering & Technology, Nagpur)
(Email:shubhammahatme1@gmail.com)

Abstract:

Software reuse has been considered as a means to help solve the software development crisis. This paper surveys recent work based on the broad framework of software reusability research, and suggests directions for future research. We address general, technical, and non- technical issues of software reuse, and conclude that reuse needs to be viewed in the context of a total systems approach. We also envision a software system or reuse support system(RSS) that helps document and elucidate existing application systems so that the ideas and design decisions involved in their creation can be reused either in the context of maintenance or when building new systems.

Keywords **— hypertension, health care, iceberg disease, modifiable risk factor.**

1. INTRODUCTION

Organizations face many problems in software development including increased costs, delayed schedules, unsatisfied requirements, and software professional shortages. This situation is often referred to as the software development crisis. Increases in software development productivity and improvement in software quality are necessary to allow organizations to maximize the return on investment in information technology. The new business environment, which is characterized by increased competition, global markets, and the need to cut costs, makes this improvement in software development productivity even more important. In this paper, we examine software reusability as a means to improve the process of software development and also the quality of the software produced. Software reuse refers to the use of previously developed software components in new applications. Traditionally, this has involved code reuse by other programmers in the same organization.

2. A FRAMEWORK FOR SOFTWARE REUSE

There are many approaches to the concept of software reuse. To organize and place various concepts and models of reuse (or reusability research), a number of conceptual frameworks for software reuse have been proposed. A framework which classifies the available technologies for reusability into two major groups, composition technologies and generation technologies .Another framework based on three research and development questions, what is being reused?, how should it be reused?, and what is needed to enable successful reuse?, is developed by [Freeman 871. In Freeman's framework, five levels of

reusable information (code fragments, logical structure, functional architecture, external knowledge (such as application domain knowledge and software development knowledge), and environmental knowledge related to organizational and psychological are defined. For each of the five information levels, typical projects of three different expected payoff periods are identified to answer research and development questions .In this paper, the framework for software reusability research shown in Figure 1 is used to organize our discussion. This framework is inclusive in the sense that most issues in other frameworks are discussed. Center for Digital Economy Research Stem School of Business IVorking Paper IS-9 1- 15. In Figure 1, research on software reuse is divided into three groups according to the point of view: general issues, technical issues, and non-technical issues. General issues are classified into definitions and scope of software reuse and economic issues. Technical issues are classified into reuse methodologies and software approaches. lion-technical issues are classified into organizational issues and psychological issues. software reusability research general issues technical issues non-technical issues I I I softwire I I and issues reuse pproaches psychological organizational scope A methodologies issues issues I I 1 I I economic cost generation composition object benefits benefit methods methods CASE oriented approach models I 1 methods I I I 1 1 language ransformation based systems systems application generators retrieval strategies application software composition classification l~brar~es principles and Figure 1. A Framework of Software Reusability Research

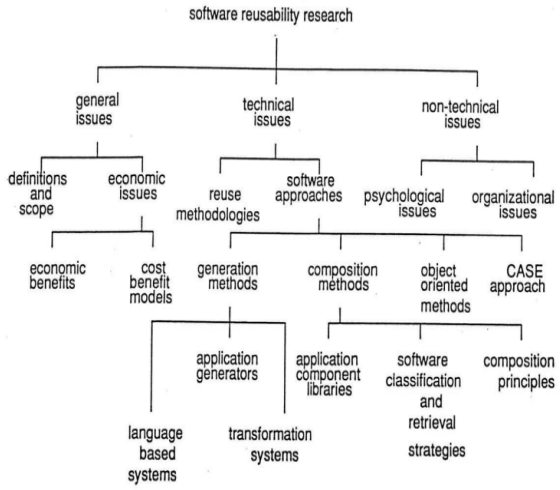

Figure 1. A Framework of Software Reusability Research

3. DEFINITIONS AND SCOPE OF SOFTWARE REUSE

Questions related to software reuse (such as what is software Center for Digital Economy Research Stem School of Business IVorking Paper IS-9 1- 15 reuse?, what do we reuse?, when do we apply software reuse?, and who reuses software?) have been considered by a number of researchers . Software reuse is defined as the use of previously developed software artifacts such as design, code, documentation, etc., in new applications by various users such as programmers and systems analysts.To provide an organized and inclusive point of view, we define the concept of widespread software reuse with respect to the following criteria: user types, reusable resource types, and task types.

User Types

When reusable software resources are well classified and easily retrievable, anyone in the same organization should be able to use them for software system development. For organizations such as the Department of Defense that normally involve a number of different software contractors, reuse across different organizations can be extremely important both economically and from the

point of view of developing sound, cohersnt, and maintainable systems [Myers 871 [DOD 863.Software reuse by people in different organizations implies such problems as standardization and legal rights that will not be considered in this paper.

Entity types

There are three kinds of entities that can be reusable: process, data, and object. An object resource is a combination of data and process resources [Wegner 901 [Micallef 881. Note that in this paper, we focus on process and data resources only.Process resources are usually considered the main target of
software reuse. However, the importance of data resources as reusable software objects is recognized by the emergence of data base management systems(DBMS), standard data interchange format [Fylstra and Gill 801, expanded data definition which includes various data types such as
graphics and voice, and many data-related applications.

Task Types

Maintenance includes two task types: modifying existing software
systems and adding new components to enhance existing software systems.
In both cases, maintenance can be viewed as a reuse-oriented task in
which the appropriate requirements, design, and code from earlier
versions of the system has to be accessed and understood by the
maintenance programmer.

4. ECONOMIC ISSUES

Reusing software resources can result in increases in productivity and improvements in quality [Standish 841 [Horowitz and Munson 841 [Boldyreff 891 [Rubin 901 and reliability [Lubars 861. Economic issues concern (1) actual evidence of productivity and quality increase from software reuse and (2) cost benefit models of software reuse.

5. REUSE METHODOLOGIES

In the framework in Figure 1, technical issues are divided into two categories: reuse methodologies and software approaches. In this section, we group the challenges involved in developing a reuse methodology. Section 6 will discuss the various software approach that can be used to support watever methodology is adopted. As the development costs of software systems increase, the role of reuse becomes more important in software engineering. For this reason,a software engineering methodology should support the notion of developing and leveraging reusable software resources [Rubin 901.
eve loping a software methodology that supports reuse is an active focus of current research [Hall 891 [Freeman 87a] [Wirfs-Brock and Wilkerson 891.
Here, we look at reuse methodologies rather narrowly in terms of the process steps that might be performed by a software development group. Approaches to reusability that involve broader organizational strategies are discussed in section 7. Finally the sixth process is required to combine the new and reused software resources into the target software system.

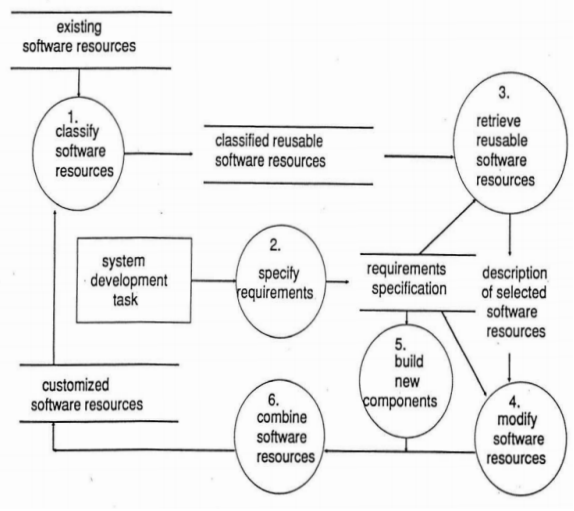

Figure 2. System Development with the Concept of Software Reuse.

6. NON-TECHNICAL ISSUES

Most research on software reusability deals with technical issues. such research concentrates on the "WHAT and HOWw of the reuse issue, but rarely explains ItWHYtt. Non-technical issues explain the difficulties that have been experienced in promoting widespread reuse in industry. They can be classified into two categories:psychological issues and organizational issues.

7. CONCLUSION AND REFERENCE

In the above, general, technical, and non-technical issues of Center for Digital Economy Research Stem School of Business Working Paper IS-9 1- 15 software reuse were broadly addressed. Software reuse is regarded as a key to improving software development productivity and quality [Tracz 871 [Gruman 881 [Biggerstaff and Richter 891. As outlined above, researchers and practitioners have proposed many approaches to increase the potential of software reusability. The definition of widespread software reuse developed in this paper places equal emphasis on data and process resources, abstract and concrete resources, specific application-oriented and generic function-oriented resources. It also emphasizes a wide range of tasks from maintenance of existing systems to development of new systems. The field of software reuse is at a formative stage. Major research opportunities exist in all of the areas of software reusability research that are depicted in Figure 1. These research question have been suggested in the body of the paper. Briefly, we need better tools for modeling the productivity gains from reuse to help motivate its adoption and guide research into fruitfuldirections. We need to understand better how to build reusability into our software development methodologies. In the technical area, we need to understand the role of reusability in the generation, composition, object- oriented, and CASE approaches to software development. A key will be to develop improved methods for classifying, organizing, and retrieving software resources. Finally, to make the technical solutions work, we need to know how to build a supportive organizational environment and to solve the psychological problems of motivation and bounded rationality.

In our opinion, the key conclusion that can be drawn from this review is that reuse needs to be viewed in the context of a total systems approach. Thus, we do not envisage only libraries of useful data definitions, software routines, or objects that can be reused by a motivated user. Rather, we also envision a software system or reuse support system(RSS) that helps document and elucidate existing application systems so that the ideas and design decisions involved in their creation can be reused either in the context of maintenance or Center for

Digital Economy Research stem School of Business Working Paper IS-9 1- 15 30 when building new systems. In the latter case, the reuse support system
should encourage the use of standard data definitions, and software design approaches both through the organization and also between
organizations. The LaSSIE system [Debanbu et al. 911 and Telos [Mylopoulos et al. 901 are two approaches to building such an
information system about an information systemw.