**REUSE OF COMPONENTS IN SOFTWARE ENGINEERING;**

Mostly system are designed in such a way that they are able to reuse. Software engineering is basically more focused on use of original components and reuse of components is not so common in software engineering. However many applications and systems now a days promote reuse of components as it is beneficial in many ways. So in past few years many techniques are developed to reuse software. The system have built in components which can be reuse and some systems have extreme potential for reuse because components that used abstract data types can be reuse in many situations. However the reuse in software engineering is possible at various levels and from single component to whole application, a software reuse is possible at different levels and in different ways and environments. There are different ways for software reuse. When someone tries to reuse software. He must be considered the following factors.

**SCHEDULE OF THE DEVELOPMENT OF SOFTWARE;**

1. If quick and advance use of software is required and software development is not cumbersome than whole new development of entire software system is recommended.

2**. LIFETIME OF SOFTWARE;**  if software is used over a long period of time or for long life purposes. The maintenance of software should be the first priority of developer and he should not compromise on maintenance of software for

term or temporary benefits of reuse. You can adopt system for reuse easily by making changes in its components if you have enough time and there is no need to hurry or to make quick or swift changes in software.

**3. SKILLS AND EXPERTISE OF DEVELOPMENT TEAM;**

When making some amendment in software to make it fit for use for specific purposes, the skills and expertise of software team means a lot. Software reuse techniques are bit complex and it requires a lot of skills and expertise to use old software in some new ways.

**4 THE FUNCTIONAL AND NON FUNCTIONAL REQUIREMENTS OF SOFTWARE**;

A security or safety case is required for critical systems.

**5. THE PLATFORM.**

The platform on which software system runs is most important and we would able to reuse software if it is redesigned for same platform. For example! '.Net' is specific to Microsoft platforms and many other applications are also platform specific like android apps so you only reuse the software if it is reuse in same platform.

**REUSE TECHNIQUES.**

Ther are many reuse techniques that are available. However few are easy to implement and better to use for diverse business and enterprise requirements. Reusing components of software is a challenging task for software developers and they use many approaches to do it.

**SOFTWARE PRODUCTS LINE.**

The effective approach to reuse components of software is to define software product line. For example, a printer control software is designed to use it to control specific type of printer by the company. The company then built a different model with little changes in its hardware. So the software built for specific device can be adapted to use with different models with little or few changes in its source code.

A software product line is a software having a same architecture but shared components. So the components can be reuse in specific situations.

**WHAT IS THE COMPONENT.**

A component us advance form of structured data. it is more advanced than objects. There are different definitions of components circle in software development communities.

' a software element that conforms to a standard component model and can be independent deployed and composed without modification and according to a composition standard '( councill and heinamann, 2001) '.

A standard based definition of component is

' a software unit that conforms to the specific composition standard is component' ( szypersky , 2002).

So the definition clarified to us that a component is element that is embedded into system. It is also agreed by developers that components are independent so that they can be reuse and the developers can reuse components in different situations according to requirements.

**WHY REUSE OF COMPONENTS IS NECESSARY.**

components are basically services providers so they can provide one ir more service according to need and their design. When specific component us embedded in a system, system calls the component whenever it needs the service provided by the component and system does not need information about where component is executed in the system. For example! A system used in public library use component that provides search services for catalogue or a system which transfer files format from one format to another use component that transfer jpeg files into png files format. So these are examples of components used in system. Reusing components is necessary because it makes system more efficient and advanced and also makes the system more functional.

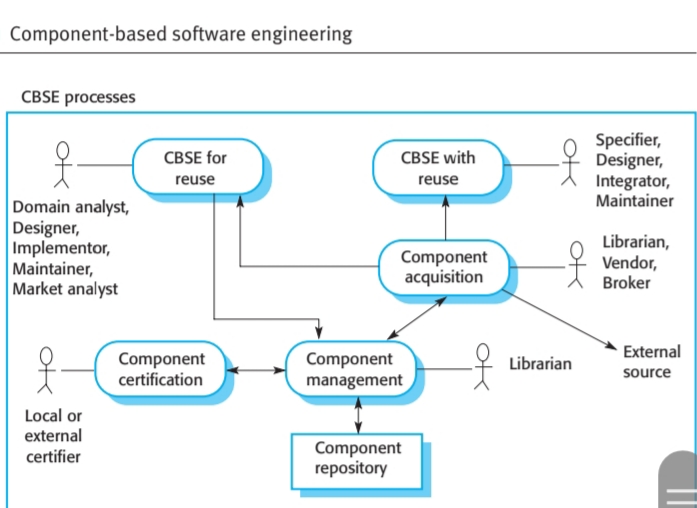
**REUSING COMPONENTS.**

The software processes like CBSE processes support components based software engineering take account of possibility of reuse of components. Different process activities involved in development and using reusable components.

The CBSE PROCESSES are.

1. Development for reuse. The process deal with developing components that will be reuse with other applications. General components are involved in the process.

2. Developing with reuse. In this process, new applications are developed that used already existing components.



Basic process of reuse are illustrated in the figure. For reuse there are also supporting processes concerning with components acquisition, components management and components certification.

**1. COMPONENTS ACQUISITION.**

It is a process of development of components into reusable components.

**COMPONENTS MANAGEMENT.**

It deals with management of reusable components so that they are available for reuse.

components certification is the process of certifying the components that they are according to their prescribed specification.

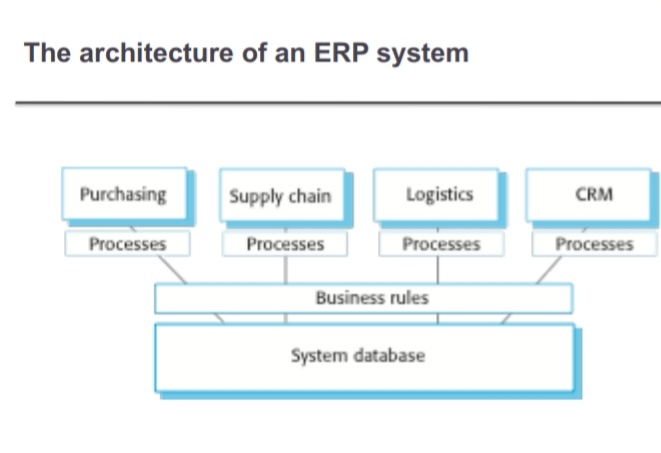
**EXAMPLE OF REUSE OF COMPONENTS IN SOFTWARE SYSTEMS.**

**COTS SYSTEM**.

it is a software system that is built to support various types of business environments, business development and activities and even sometimes the whole enterprise. Cots system used by small to medium to large scale industries and offices to fulfill their software requirements. It can be reuse for different types of customers with small or minor alterations and the source code of COTS system can be reuse for different purposes and for different customers without any significant changes. The system have special features so it can be reuse in different conditions.

**ERP SYSTEM**

Another example of reuse of components in software engineering is ERP SYSTEMS. An enterprise resource planning system is a system that is use to facilitate different business processes like ordering or invoicing etc. ERP system is most widely used software system in business and industry. So the system is basically built around the core. Which is altered and adopted for different types of uses in industry and business. It is a most common example of software reuse. The core is modified by inclusions of modules and further modifications is done by using knowledge of business processes and rules so that it can be fit for any use in diverse environments if businesses and industries.



**THE ARCHITECTURE OF ERP ;**

**1.** It built up of modules to support various different types of business functions.

2. A set of processes that facilitate the activities of modules are associated with each of the modules in ERP SYSTEM.

3. A database is built up to store all the information of business functions and facilitates the temporary storage activities performed during different business activities and processes. The information stored in a database in such a way that it can be operated and reused in various different business activities.

4. A set of rules is designed to use the information stored in the database and the rules are nearly applied to all information stored in the database. Various components are embedded in ERP SYSTEM SOFTWARE. These components are reusable and it provides system more depth, diversity, functionality, flexibility and power.

**BENEFITS OF REUSE OF COMPONENTS.**

Common advantages of reusing components are.

1. It increases productivity of software systems.

2. It decreases development time of software.

3. It improve interoperability of software system.

4. It lessens the need of skills and expertise required to develop software systems.

5. It makes easier the project development process.

6. It reduces costs of developing software system.

7. It increases standard of software.

8. It increases the potential of software and systems.

9. It improve the user experience in interfacing with software.

10. It increases overall performance and quality of the system.

11. It reduces implementation, training and maintenance costs.

12. It makes easier the upgrades of system.

13. It makes software more efficient and user friendly.

14. It gives access to looking and generalizing source code written for specific software component designed to provide specific services.

15. It makes source code more readable and understandable.

**DISADVANTAGES OR LIMITATIONS OF REUSE OF COMPONENTS IN SYSTEM**.

1. Reusing components in software engineering requires additional skill and expertise.

2. It sometimes increases the complexity of source code written for software.

3. Reusing required specifications, design documents etc which is sometimes difficult to obtain or provide.

4. Reuse of components might be hard than use of hardware equipment because framework for reuse is difficult.

5. It decreases the portability of software. Software might be used with specific platform.

**RISK IN REUSING COMPONENTS IN SOFTWARE ENGINEERING.**

Reusing software or developing it from scratch is often decide by management and not by developers. Some unknown risks are associated with components reuse in software. These are system failure, lack of performance, inability to achieve the objectives associated with reuse of components in software engineering, increasing maintenance costs, lack of skills and expertise of developers to deal with source code of components, difficult user interface, bulky software and some other risks are also associated with reuse of components in software systems.

However the benefits of reuse of components are great and development of new systems from scratch might impose more serious risks than the risks associated with reuse of components in software engineering so reusing components in software engineering should be applied and promoted by developers.

**REFERENCING**

The four articles i researched for writing assignment are .

1 Haefliger, S., von Krogh, G., &Spaeth, S. (2008). Code Reuse in Open Source Software. Management Science, 54(1), 180–193. http://www.jstor.org/stable/20122369

2 P Spector, A. Z. (1989). SOFTWARE, INTERFACE, AND IMPLEMENTATION. Jurimetrics, 30(1), 79–90. http://www.jstor.org/stable/29762156

3 Mills, H. D. (1977). Software Engineering. Science, 195(4283), 1199–1205. http://www.jstor.org/stable/1743573

4 Slebodnik, M., &Riehle, C. F. (2009). Creating Online Tutorials at Your Libraries: Software Choices and Practical Implications. Reference & User Services Quarterly, 49(1), 33–51. http://www.jstor.org/stable/20865173

5 KEITH, M., DEMIRKAN, H., & GOUL, M. (2013). Service-Oriented Methodology for Systems Development. Journal of Management Information Systems, 30(1), 227–259. http://www.jstor.org/stable/43590325

6 Haefliger, S., von Krogh, G., &Spaeth, S. (2008). Code Reuse in Open Source Software. Management Science, 54(1), 180–193. http://www.jstor.org/stable/20122369