Abstract— Component-Based Software Engineering (CBSE) means the development of systems using in-house and third party software components. It helps in decreasing the cost of building the systems, the time it takes to deliver to the customer, enhances the maintenance of the software and improves dependability on the system. The quality of such systems relies upon the quality of individual components which form such systems and can be ensured by testing the individual components. The paper proposes a suitable component repository which allows storage and retrieval of components to enhance reusability and also focuses on unit testing of the components using JUnit. The proposed repository provides features such as user authentication, storage and retrieval of code components and test components.

Keywords— Component Repository, Component Based Development, Component Testing, Unit Testing, Storage and retrieval of Components

I. INTRODUCTION

Software systems have become more complicated which causes higher cost of building the systems, lesser productiveness, and lowering in the system-quality. Thus need of finding a good economical development strategy arises. CBSE helps in decreasing the cost of building the systems, the time it takes to deliver to the customer, enhances the maintenance of the software and dependability on the system. Component-Based Software Engineering (CBSE) means the development of systems through using already developed software components. Component-based development approach builds the software systems by choosing appropriate components and then integrating them by using a chiseled architecture. Fig. 1 shows how the component based development takes place.

There are a lot of existing software component repositories such as +1Reuse Repository[2], Software Asset Library Management System (SALMS) [3], Automated Software Reuse Repository (ASRR) [4], The Universal Repository[5,6], AIRS[7,8], Reuse Library Toolset (RLT) [9], HSTX Reuse Repository [10], Defense Software Repository System (DSRS)[11,12,13], Library Interoperability Demonstration (LID) [14], Integrated Computer Aided Software Engineering (I-CASE)[15], Multimedia Oriented Repository Environment(MORE)[16], Asset Source for Software Engineering Technology (SAIC/ASSET)[17], The Public Ada Library (PAL)[18], The CAPS Software Reusable Component Repository[19], The Ada Library and the Reuse Library at the Defense Information Systems Agency (DISA) [20] which uses different search and retrieval schemes. Effective search and retrieval methods include Text Based Search Method, Keyword Based Search Method, Signature Based Search Method and Operational Semantic Based Search Method. The Fig. 2 shows the already developed software repositories with the details about the retrieval method they use, whether they have security control, are they web based and are they integrated into CASE environment.
Component Based Software Development builds the systems by making use of already existing components for which a warehouse is required. But the problem occurs in the efficient searching and retrieving of the software components. Also the quality of such systems relies upon the quality of individual components which form such systems. The quality of such systems can be ensured by testing the individual components. Thus another challenge is component based testing. We can find the reliable and bug free components only if the components are tested by a suitable testing tool or some testing technique.

II. PROPOSED APPROACH, IMPLEMENTATION AND RESULTS

The aim is to develop a suitable component repository which allows storage and retrieval of components to enhance reusability and also testing the components. The proposed repository provides features such as user authentication, storage and retrieval of code components and test components used for testing the original code components.

A. Storage of Components

1) Manual Storage: The information about every component is added by the person using the system and it involves entering the details of component in the repository and uploading the components for future use.

B. Retrieval of Components

1) User Priority-Based Component Retrieval: Select Component Type, Select Component Language, Select Component Domain and Select Component Name.

2) Keyword-Based Component Retrieval: Keyword is entered to repository and relevant components which match the keywords are retrieved.

The Fig. 3 shows the login page which provides authentication to the users who can access the repository.
The home page of the developed repository is shown in the Fig. 4, describing the features provided by repository such as storage and retrieval of code and test components.

![Welcome To Component Repository](image1)

**Fig. 4 Home page of repository**

The component can be added to the repository by providing the details as shown in the Fig. 5, with the mandatory fields as component Id i.e. a unique id and keyword by which the component will be searched. Also the source file of the component can be uploaded by clicking on upload component button.

![Fig. 5 Add component](image2)

**Fig. 5 Add component**

Fig. 6 shows that the components can be retrieved by two ways one is using keyword and another way is specific retrieval.

![Retrieve by Keyword](image3)

**Fig. 6 Retrieve component**
Fig. 7 shows how a component can be searched and retrieved using keyword. In order to retrieve using keyword, the keyword is to be entered by user and the details of the relevant components will appear as the result.

![Retrieve component using keyword](image1.png)

The Fig. 8 shows the details of the retrieved component, having a download component link which the user can use to download the component.

![Details of retrieved component using keyword](image2.png)

C. Testing the Components

Testing of the components has been done by using JUnit, which is a framework for unit testing. It identifies test methods by using annotations and the class which contains test methods is called test class. For creating the JUnit test case firstly open the retrieved component and the corresponding test component. Then run the test component as a JUnit test.

The Fig. 9 shows the retrieved code component which is to be tested.

![Retrieve code component to be tested](image3.png)
The Fig. 10 shows the retrieved corresponding test component which will be used to test the code component.

Fig. 10 Retrieved test component

Now right click on the test component class and select Run -> Run As-> JUnit Test to execute the test component. The green bar on the left side of Fig. 11 shows that the test component has been executed successfully showing that there were no errors in the component being tested in accordance with the test component being used.

Fig. 11 Result of test component execution using JUnit

III. CONCLUSIONS

Development of Component Repository is an effort towards Component Storage and Extraction. Component Repository finds its scope of application in a firm which wants to reuse components, as it is capable of generating, retrieving, storing components effectively. A component repository has been developed, which provides storage and retrieval of components using manual storage and keyword and specific retrieval techniques. Also testing of the components is done using white-box testing technique at unit level by using JUnit. Future work can be extended on automatic storage of different types of components like test cases, source code. Auto-generation of tags can be applied on large number of dynamic links to reduce the human efforts and error, which enhances the possibility of getting most relevant software components.

ACKNOWLEDGMENT

We thank Thapar University for providing us opportunities to implement our work.

REFERENCES


