International Journal of Computer Applications (0975 – 8887)
Volume 146 – No.10, July 2016

Implementing Quality Assurance Features in Component-based Software System

Navdeep Batolar
Student,
Computer Science Department
Guru Nanak Dev University
Amritsar, India

Parminder Kaur
Professor,
Computer Science Department
Guru Nanak Dev University
Amritsar, India

ABSTRACT

The increasing demand of component-based development approach (CBDA) gives opportunity to the software developers to increase the speed of the software development process and lower its production cost. The software developers have to face many challenges while developing the software and one of the major challenges is to develop the software with an appropriate level of quality. Quality assurance is considered as the most important element as without an appropriate level of quality, the software product will produce ruinous results and it will be of no use. Considering quality in case of CBDA, it is very important to use those components in the development process which will provide us with questionable quality.

CBDA offers various advantages including, reducing the complexity of the software by simplifying its design, reducing the production cost and time, as there is no need to start the development process from the scratch, it provides us with the feature of reusability. Important issues which should be properly handled while developing the software are: assuring the quality of each item which is being used, using that component which best suits our requirements. The main concern of this paper is to develop an application with an appropriate level of quality. In order to fulfill this demand, the concept of functional and extra- functional properties and validation of quality metrics is being used. Security and integrity, these two quality metrics are being implemented and experimental results are explained.

Keywords

Component-based development approach , quality assurance, quality metrics, security, integrity

1. INTRODUCTION

Component- based development approach have become quite popular for the development of efficient and effective software product. There are many reasons like it handles and minimize the complexity of the software product, deliver the product in timely manner , improve consistency and productivity which encourage the use of this approach. Apart from all these advantages, the most important motivation factor being offered by component-based development approach is its capability to improve and assure quality. Quality is the foremost and the most important feature of any software system. The main concern while using component-based development approach is to assure the quality of each component being used. The quality of a component can be considered as a guarantee that the software system being developed will be able to carry out the required services.

In order to assure the quality of the software product being developed, the concept of quality metrics is being implemented.

What are Quality metrics?
Quality metric can be considered as a guide to measure the quality of the software product.

- It improves the performance by identifying as how the task can be accomplishes in relation to standard.
- It allows and encourages the comparison among various products.
- It supports various business strategies which are used to improve the quality of the software product.
- It is a potential measure to reduce the shortages.

Why to use Quality metrics ?
Quality metric provides us with data which is used to

- Identify the latest trends in performance.
- Predict the performance of the system being developed.

If quality metrics are efficiently deployed for the development of the software product then the quality of the software system can be improved

Quality metrics are used to measure the quality of the software product. Quality metrics can evaluate and access the quality of the product being developed from customer point of view on the basis of certain parameters which are shown below.

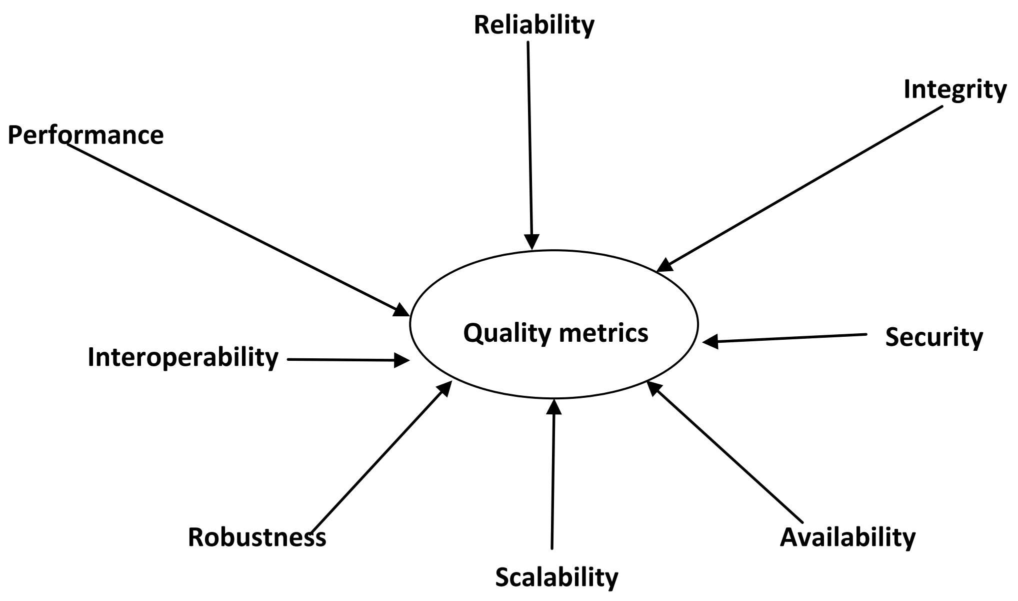


Fig 1: Parameters of Quality Metrics

2. RELATED WORK

(Reeta et al.,2014) have represented an approach by which the Quality of service (Qos) for dynamic reconfigurable CBSS can be improved. By dynamic reconfiguration, we can say that a system can be changed from one configuration to other without shutting down or rebooting the system. The major challenge faced during reconfiguration is to maintain the Qos while reconfiguration. The main aim is to reduce system disruption which can be achieved by using appropriate reconfiguration strategies and testing each strategy by using appropriate reconfiguration benchmark result and evaluating the results at the end**. (Li et al.,2012)** suggested that the true benefit of dynamic configuration can only be achieved if it causes minimum disruption to the ongoing application. If the quality of assurance features are preserved during the dynamic reconfiguration then the system has significant advantage over static reconfiguration system. **(Ramaswamy et al.,2013)** discussed about Component-based decisive architecture for reliable autonomous systems. In order to achieve the quality of

component- based software system, there is a need to develop a scalable framework which will assure the safety requirements. The functional and non-functional aspects for decision making strategies are also taken into consideration. **(Esposito et al.,2011)** have represented a systematic approach for qualifying and selecting off the shelf components for safety critical systems.

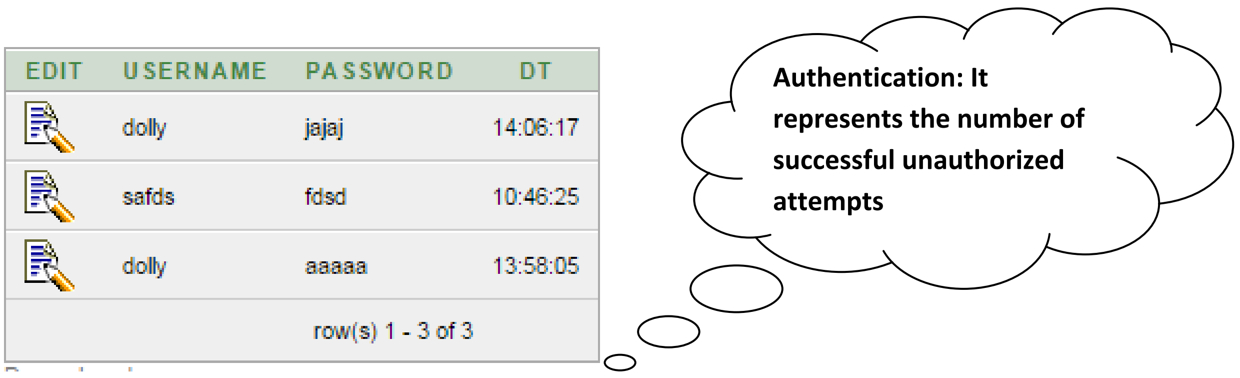
(Crelier et al.,2011) has given an overview about model-driven quality assurance tools. These tools are used to assure the traceability of existing models and artifacts and also support development phases like software testing and validation. **(Yan et al.,2011)** have described the autonomic trust management for CBSS. Trust is an important issue in the case of CBSS as components and their requirements vary very rapidly. An adaptive trust control model is introduced which can specify, evaluate the trust relationships. **(Liangli et al.,2006)** has discussed the methods by which the testability of component-based software system can be improved. Quality assurance is taken as a key research content in this paper. **(Alvaro et al.,2005)** represented the advantages of using component based approach for software development.

3. PRACTICAL IMPLEMENTATION

In order to assure security and integrity following methodology is being implemented.

An application is said to be secured if it possess the following features:

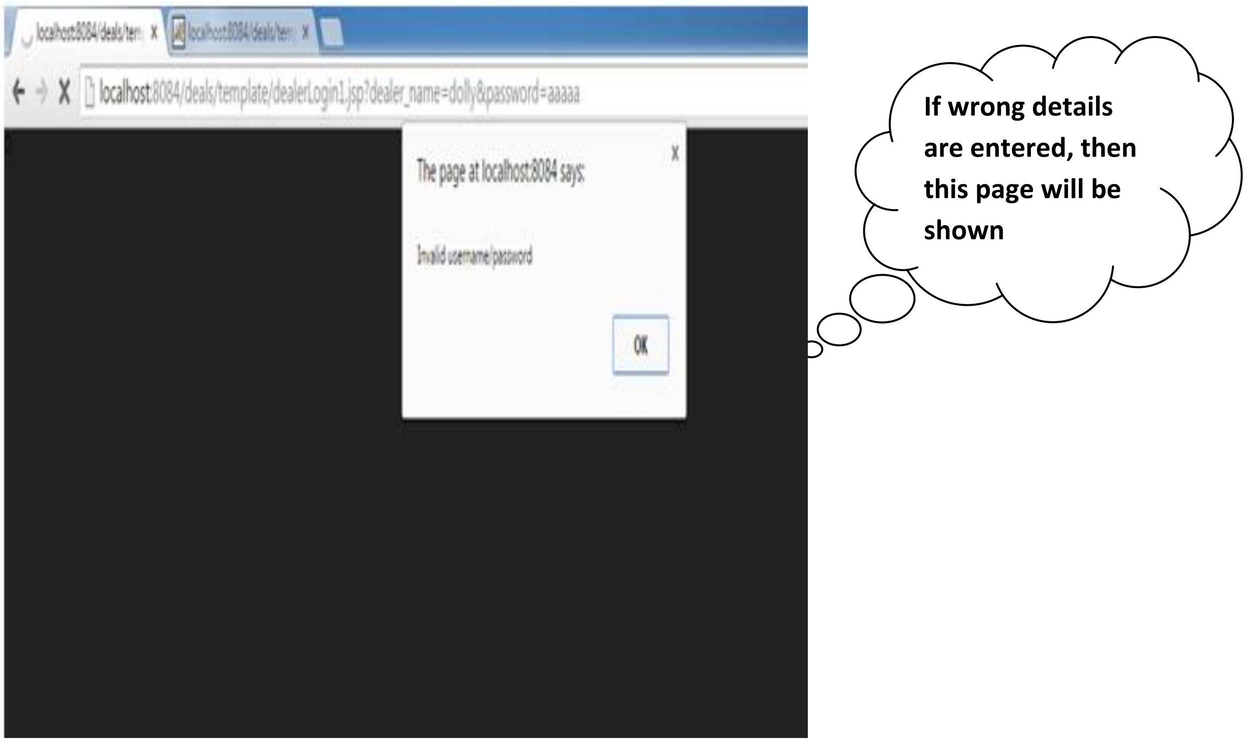
- Authorization and authentication.
- Confidentiality



EDIT	USERNAME	PASSWORD	DT
	dolly	jajaj	14:06:17
	safds	fdsd	10:46:25
	dolly	aaaaa	13:58:05
	row(s) 1 - 3 of 3		

Authentication: It represents the number of successful unauthorized attempts

Fig 2: Authorization and authentication


localhost:8084/deals/template/dealerLogin1.jsp?dealer_name=dolly&password=aaaaa

The page at localhost:8084 says:

Invalid username/password

OK

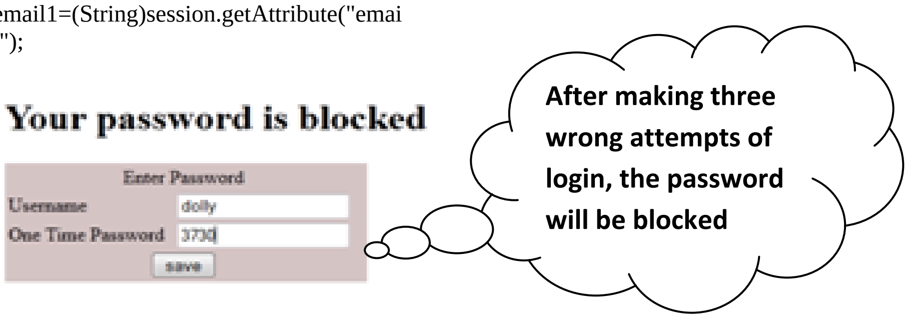
Fig 3: Authorization and authentication

- **Confidentiality**

```
else if(ab.size()>3)
        {
  a.setDealer_name(n);
        Random ch=new Random();
        onetimepass= String.valueOf(ch.nextInt(10000));

a.setOnetimepass(onetimepass);
        a.updatepass();

email1=(String)session.getAttribute("email");
        String from="Secure Login System";
        String to=email1;
        String subject="One Time Password";
 String message="Your password to access system is : "+onetimepass+"\r\n. Your password will be valid for 10 mins";
  mybeans.Common1.msgSend(from, to, subject, message);
  response.sendRedirect("msg.jsp")
        }
```


Your password is blocked

Enter Password

Username	dolly
One Time Password	3730

save

After making three wrong attempts of login, the password will be blocked

Fig 4: Confidentiality

- If the person enters the correct password after making three wrong attempts then its confidentiality will be checked by sending a one time password to its email id.

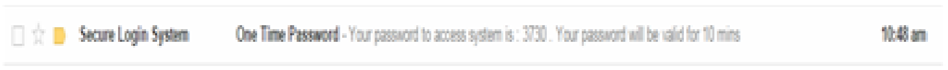

Secure Login System   One Time Password - Your password to access system is : 3730 . Your password will be valid for 10 mins   10:48 am

Fig 5: Confidentiality

Integrity:

Integrity measures to what level the services can prevent unauthorized access to, or modification of, data or programs.

- **Extent to get protected from unauthorized user is high**.

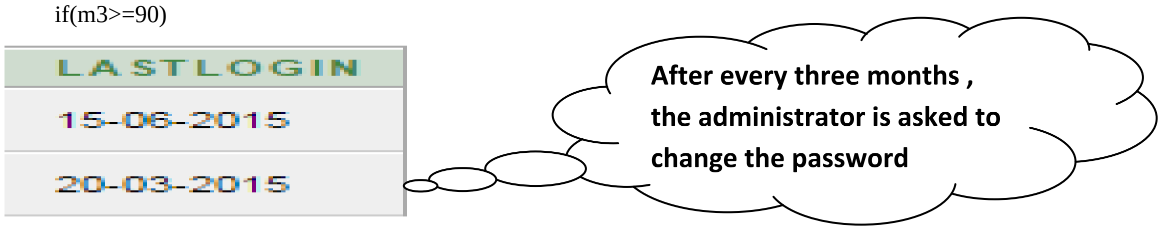
```
Dealer t=new Dealer();
    t.setDealer_name(n);
```

```
            t.setPassword(a1);
            ArrayList ar=t.login();
            out.println(ar.size());
            String email1="",lastlogin="";
            if(ar.size()==1)
            {
                  Dealer a=(Dealer)ar.get(0);
                  session.setAttribute("did",
a.getDealer_id());

                session.setAttribute("dname",
               a.getDealer_name());
        Date d1=s.parse(lastlogin);
         long m2=d1.getTime();
         long m3=(m1-m2)/(1000*60*60*24);
         if(m3>=90)
                  session.setAttribute("email",
a.getEmail());
                     email1=a.getEmail();
                     lastlogin=a.getLastlogin();

out.println(email1+"cccccccccccccccccccc");
                  Calendar c=Calendar.getInstance();
                  Date d=c.getTime();
                  long m1=d.getTime();
                 SimpleDateFormat                    s=new
SimpleDateFormat("dd-MM-yyyy");

         {
            response.sendRedirect("cpassword.jsp");
         }
```



LASTLOGIN
15-06-2015
20-03-2015

After every three months , the administrator is asked to change the password

Fig 5: Integrity

- In order to protect the application from unauthorized attempt, the administrator is asked to change the password after every three months, whwnever the time limit crosses three months the administrator is directly shown the change password link before he can do any further processing.

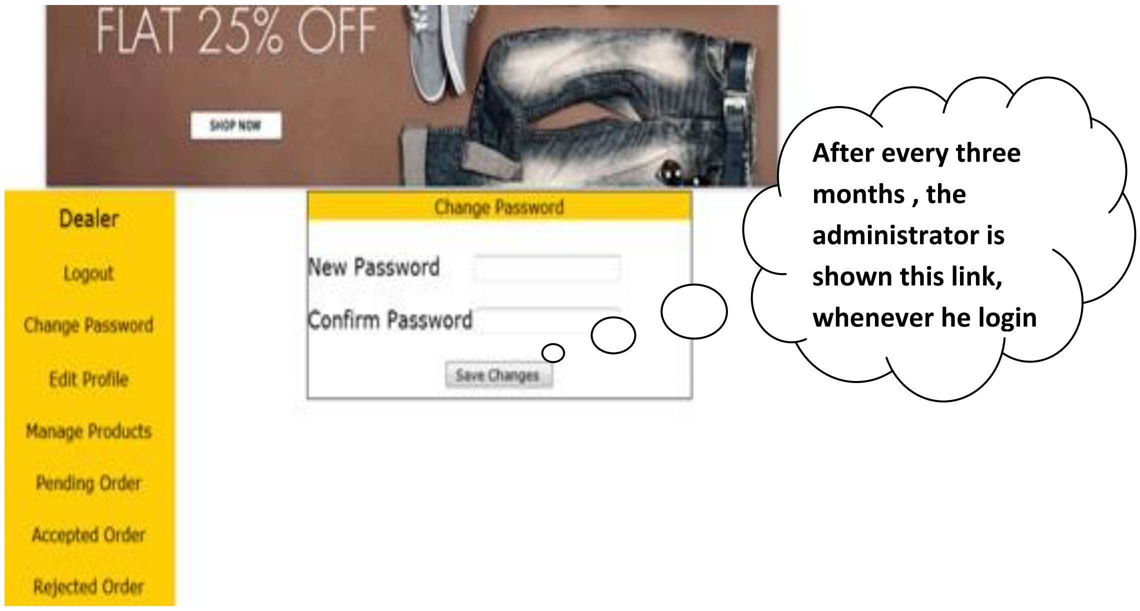


Fig 6: Integrity

4. CONCLUSIONS

Component-based development approach have become quite apparent in the field of software system. The improvement in technology and the need for quality requirements have encouraged the developers to create software system with an appropriate level of quality, the fact cannot be denied that low quality negatively effect the working of the software system and in order to develop a quality software, the developers must use the concept of functional and extra functional properties. In this thesis work, the main focus is given on proposals that are accustomed to assess the quality of component- based software system. The most important motivation factor behind using component-based approach for the development of software system is its capability to provide quality assurance. A software system which does not fulfill the quality requirements is of no use. In order to fulfill the quality requirements, various techniques are used which includes: simplification of software design by using the concept of functional and extra functional properties, developing appropriate understanding of the quality metrics and finally validating these metrics for practical implementation.

This thesis work summarizes the main points with the help of which the quality of component-based software system can be assured. Through practical implementation, we can conclude that in order to assure the quality of online application, the most important quality metrics which should be taken into consideration is security and integrity, other quality metrics are also important but these two quality metrics plays a vital role for quality assurance of such applications. Security and integrity can be implemented if the application possess features like Authorization and authentication, Traceability and auditability, Confidentiality, Data encryption.

5. FUTURE WORK

Considering quality as the main motivation factor for the development of the application, customer's view point plays a vital role ,but the perception of the customer with respect to the quality changes very rapidly. There is a customer satisfaction metric which measures the satisfaction level of the customer in accordance with five scale points. In this thesis work, the quality is assured by using the concept of quality metrics, but different types of applications demands the existence of different quality

metrics, for example, if we consider online based applications , security can be considered as the most important quality metric. The approach used in this thesis work have tried to implement security and integrity in online based applications.

Future work will consist of implementing other quality metrics like reliability, availability, interoperability, regularity, flexibility and so on in component-based software system for enhancing the quality of these system.

6. REFERENCES

[1] Li, W. (2012). Qos assurance for dynamic reconfiguration of component-based software systems. Software Engineering, IEEE Transactions on, 38(3), 658-676.

[2] Ladan, M. I. (2012). Web services metrics: A survey and a classification.*Journal of Communication and Computer*, *9*(7), 824-829.

[3] Reeta, R., & Mariappan, A. K. (2014, April). An approach to assure QoS for dynamically reconfigurable component-based software systems. InCommunications and Signal Processing (ICCSP), 2014 International Conference on (pp. 1353-1357). IEEE.

[4] Ramaswamy, A., Monsuez, B., & Tapus, A. (2013, May). Component based decision architecture for reliable autonomous systems. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on* (pp. 605-610). IEEE.

[5] Yan, Z., & Prehofer, C. (2011). Autonomic trust management for a component-based software system. *Dependable and Secure Computing, IEEE Transactions on*, *8*(6), 810-823.

[6] Liangli, M., Houxiang, W., & Yansheng, L. (2006, October). The Design of Dependency Relationships Matrix to improve the testability of Component-based Software. In Quality Software, 2006. QSIC 2006. Sixth International Conference on (pp. 93-98). IEEE.

[7] Liangli, M., Houxiang, W., & Yansheng, L. (2006, October). The Design of Dependency Relationships Matrix to improve the testability of Component-based Software. In Quality Software, 2006. QSIC 2006. Sixth International Conference on (pp. 93-98). IEEE.

[8] Crelier, O., Roberto Filho, S. S., Hasling, W. M., & Budnik, C. J. (2011, September). Design principles for integration of model-driven quality assurance tools. In *Software Components, Architectures and Reuse (SBCARS), 2011 Fifth Brazilian Symposium on* (pp. 100-109). IEEE.

[9] Alvaro, A., Almeida, E. S., & Meira, S. L. (2005). Quality attributes for a component quality model. 10th WCOP/19th ECCOP, Glasgow, Scotland.