

Flexible Enterprise Resource Planning and Testing Techniques

Anu,
Assistant Professor (Resource Person)
University Institute of Engineering and Technology
Mahrishi Dayanand University
Rohtak-124001, India
E-Mail: anukadyan01@gmail.com

Abstract:- The testing and development of enterprise resource planning (ERP) can't be performed manually with full profit accuracy due to the complexity of the system. The testing of ERF is having very high importance so that automatic approach is used for the testing and development as the data available for analysis is of broad variety and it is also very big data base to be solved and analyzed manually. The testing approach of ERP is different such as system level testing; block box testing & Model based testing. The model based testing process enables the user to a test more data with wide variety. But, in Model-based testing process a high degree of computation is required for black box testing. The concentration of present researcher usually does not address the proper functionality of UI. In this paper we have focused on the characteristics of various testing techniques and their area of implementation to assist the future research.

Keywords - *Model-based testing, UI testing, Service oriented architecture, Enterprise systems System-level testing.*

1.0 Introduction:

ERP is very essential tool for the modern day industry no matter what type of industry it is. The development of ERP deals with a very big data collection with wide variety which is changing almost every day. The testing of ERP software is more critical as it goes live. The testing of the system includes the special tasks which are not performed on regular bases in the industry and such problems may arise after a certain period of time or suddenly. The testing of the system should not affect the performance of the system and it should be quick process with less capital investment. The testing begins with the operation of the experts about a certain business scenario and then the feedback from the experts is maintained as record in the form of flow chart so that it can be easy to understand. The best feature of this approach is its flexibility this procedure can be changed throughout the development of the test script. The main concern with ERP testing is that the software developer have given a specific area to the end user for modification but every time the needs of the user may or may not match that specific area. So now for providing more freedom to their customer the software developer uses service oriented architecture (SOA) and Model based testing (MBT). The latest EPR must have SOA enabled software, the developer must know of SOA methodology guidelines, and finally the software is ready for professional services [4]. If our customer is such that it is having highly heterogeneity, highly distributive, dynamic, level of the service based systems Model-based testing (MBT) [13] is preferred.

2.0 SOA Testing

Service oriented architecture (SOA) is a very flexible method. SOA enables the developer to understand the customer requirements and the goal of the customer. The developer is having a collection of business services in a library. The developer selects the most feasible solution during the testing and implements it. The automatic integration mechanism of SOA enables the flow of information among the various departments and allow them to act according to the changes carried out. The flow of information may be web based. SOA can be implemented at any level of management to may be top to bottom and bottom to top. This flexibility is very positive for the customer but it is a very confusing for the developer. SOA testing requires the testing of service and interfaces that might be carried out simultaneously more over the security, privacy and diverse nature of the organization should be kept in mind. SOA testing have one more challenge that is the performance of the various departments and their sub units because every department may or may not be equally that much responsive to the change or the feedback received from the department may get delayed. For instance, SOA Implementation may help in integrating two or more internal applications which are independent, while testing of the ERP based system for a particular business process. These interlinking of the various departments becomes highly beneficial during integration testing in part testing as well as during complete testing of the business process. We have focused on the implementation of SOA in an efficient and reliable manner. We have also focused on the various testing categories as per the customer demand, then the comprehensive study of the suggested test strategy is carried out, and finally precise overview of the tools available in

the market is done that may be helpful for overall testing approach for SOA and ERP based systems.

2.1 Challenges before SOA:

SOA implementation brings fresh challenges for the company that may require changes in the ongoing test strategy of the company, because of the changing test inputs. Here are some of the examples of the new challenge before SOA [25]:

- a) The companies which get input from the user or have any other user interface will be having more data to be analysed so the new set of tools will be required.
- b) The variety of service customer and consumers are possible.
- c) The exchange of information between the various departments.
- d) SOA services modules can be used by the company outside their enterprise not only in the testing area that means it should be portable.
- e) SOA should get its inputs from business processes that may work of deferent technologies and span organizations for interconnection. Developer will need a versatile domain and versatile technical knowledge.

So, by looking at the challenges of SOA it may be concluded that MBT have a much greater impact and might be for influencing in the testing process in traditional industrial setups, because in such setups the modeling is carried out rather intermittently without any preplan. So as a result of this we can say that computer based regression tests should be carried out[16].

2.2 Testing Methodology of SOA:

We having discussed the challenges of SOA testing in a concise way, a elaborated study will be done on the particular testing activities, so that we may see which of testing activity is affected in what way. Legeard et.al. [12] discussed the layered testing approach which is very common for component-based systems (CBS). As a result four distinct testing layers are identified and are illustrated in figure 1 [17]. A brief introduction of each layer is as follows:

- A. Unit Testing: The Unit testing focuses on the basic functional correctness and flexibility of the various subsystems and is the easiest to understand testing layer in research as well as for customer use practice. So we may say that unit testing takes the single software unit and test it independently without interfere of the other units input. While performing the unit testing mock test the implementation context of the software unit is carried out.

- B. Service Testing: The service testing in SOA is similar to the component testing of the component-based systems. It mainly focuses on the fulfillment of the application based obligations of the component's interfaces. So generally the service testing focuses more on the integration of fundamental units inside the component but less on the correct implementation of algorithms. This also analogues to the definition of testing layers [18], were it is strongly said that everything which is not the unit testing is an example of integration testing.

- C. Integration Testing: The main feature of SOA is loosely bound service components which has high affinity towards each other can compatible enough to be integrated as per the requirements the loose coupling of service components. As compared to the component-based systems approach where cumulative testing was performed on similar components with very less flexibility so they are said to be tightly connected. The additional considerations are required for the SOA integration testing which are adaptability and distribution. The signal from one sub unit to another sub unit are studied under transportation and it is assured that the signal from the source doesn't get interrupted or delayed till to reaches to the receiver.[17]

- D. System Testing: The system testing in the SOA system testing fully integrated application, ready to be delivered for the customer for use is tested. It is checked for assuring the coustomer that all the sub systems are working well and their communication and integration is foolproof and quick.

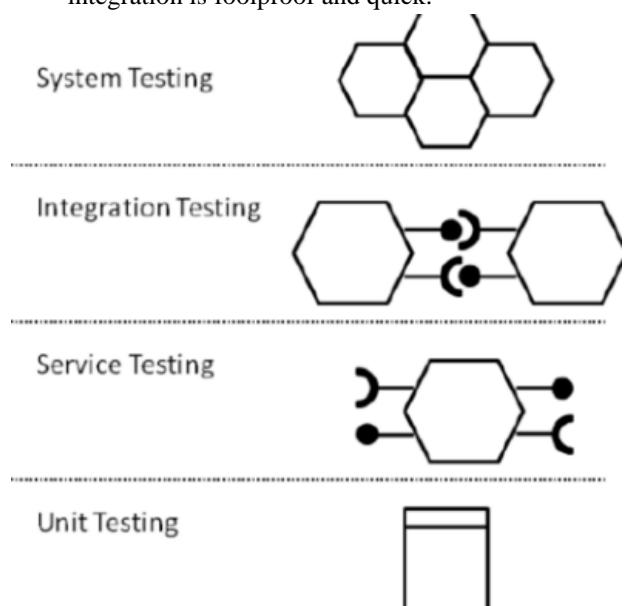


Figure No. 1: SOA Testing Layers [17]

3.0 GUI Test Automation:

Model based testing (MBT) has been used as a very interactive tool for testing through various APIs. Still, the application of this approach is a limited. There are very few software developers who want to design system-level APIs which has only testing applications solely. Moreover, the general-purpose testing tools require customization before it is adapt and use the API in an efficient way [26]. The general-purpose GUI testing tools available in a very good number and that can be directly utilized. Still the GUI tools are troubling for the developer as well as for the users due to its complexity. This skepticism often results in very troubling experience in utilizing input tolls such as mouse or key board or any other input media, and still if the input tool function in proper manor the utilization of those input signals is equally troubling task. After this there comes the testing and validation of the input signals which require highly skilled programmer to perform because there may be many faults in the input data which have to rectified by the developer and the data available to the end user should be in a user friendly language which is again a challenge for the developer. The maintenance and reprogramming cost in the system with GUI is also very high. Still this method has its applications due to the ease of input data as the input sources don't have any special features.[16]

3.1 Testing by Words and Keywords as Input:

Due to the extensive test data most of the test designers recommend those practices based on business process modeling which is having its roots in the high level language input due to its ease for the customer. The testing is done by changing one input and tracing its effects on all the sub systems. During tracking the crucial event are picked and the chain of events are studied to pick the potential failure. The high level inputs are called "action words". Action words require high improved implementations for the automation of the test. In the case of a personal computer as a input device, for example, action words can be such as "send an email", "writing a document", "checking inventory record" etc. Keywords sequences study every input in a sequence of key strokes, e.g. internet or intranet browsing, text inputting etc. An example of a keyword in windows operating system could be kw Press Key which gets the input and process it for further action. This keyword could be used, for example, in an action word that starting the internet explorer, aw Internet Explorer. The keywords may also include input that specify their functionality. For example, typing "A" from the keyboard could be described as kwtypeKey <A>. Therefore the mapping of keywords and action words which are pre defined should be kept in a library for further investigation [14].

3.2 Three-Tier Test Model Architecture:

The automation of Model based testing (MBT) will be successful after the full proof testing of the GUI model and also its implementation. So while designing the architecture these points should be kept in mind. Kervinen et al. [27] developed architecture of the three-tier test model with the intention of performing a case study with it in windows operating system. This architecture (Figure 2) is the conceptual basis for the event capturing tool development and utilized in TEMA Tool. Three tiers in GUI testing:

- Defining keywords navigating and for guiding in the GUI.
- Input the Action words as high level language to describe functional concepts.
- Define the control words to the test control related matters.

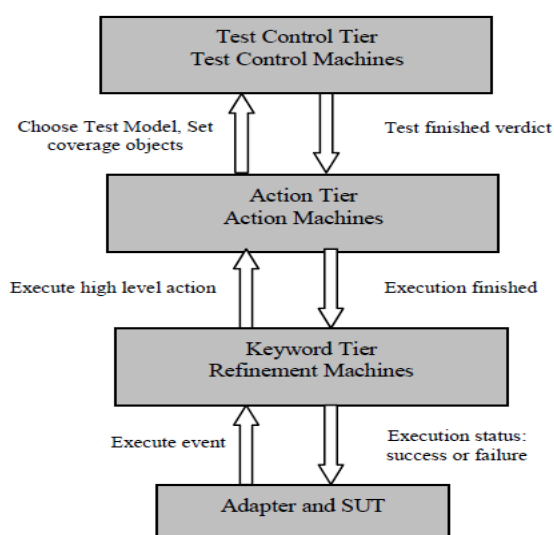


Figure 2: Three Tier Test Model Architecture

4.0 Brief View of State of the Practices:

The software industry is trying to make every software safe, efficient and user friendly. This effort begins at right at the start while defining the problem and designing the architecture till it is delivered to the customer. The testing and customer requirements can be meet by the computer controlled mechanism. The most effective way is the automatic testing of the software and then the changing the program according to the customer input. The keyword driven testing principle is usually followed for transformation from abstract test cases in SAP. The action word testing and Keyword-driven testing uses action keywords and data values into executable tests [13]. The user specific keywords or action words are the most significant as per the user friendly software is considered test language of SAP's eCATT [19]. Therefore the experience and skills of developer are used along with the knowledge for developing a appropriate test data. SAP contains a special feature called Test Data Migration Server (TDMS), this feature extract the presaved data from the data library so this data may be referred

automatically by the software according to the user defined changes. Sometimes the developer may consult the customer for the reference data and the customer input or knowledge will act as the back bone of the software development. There is one more feature in SAP known as Test Workbench, this feature controls everything related to the test.

5.0 Conclusions and Future Scopes:

In this paper we tried to compare the most common techniques for ERP testing in a brief and effective way. The paper will be helpful in guiding the researchers in for user friendly software which interactive as well as efficient in the field of system testing for enterprise resource planning, with a focus on either GUI or SOA testing. The developers who aim to develop the high level language program for more user friendly approach may adopt GUI or MBT as it is not possible in SOA. The researches may divide their ERP structure in smaller and flexible elements and then use the as per the requirements of the customer. This test may be performed according to SOA or GUI. The systematic modelling will result in a very dynamic UI specified program or UI input during running of program of the existing models for business processes.

References

- [1] D. E. O'Leary, Enterprise Resource Planning systems, life cycle, electronic commerce and risk. Cambridge University Press, 2000.
- [2] G. Pike, "Supporting business innovation while reducing technology risk," SAP AG, Tech Rep., 2006.
- [3] S. Wiczorek, A. Stefanescu, and I. Schieferdecker, "Test data provision for ERP systems," in Proc. of Int. Conf. on Software Testing (ICST'08). IEEE Computer Society, 2008, pp. 396–403.
- [4] D. Woods and T. Mattern, Enterprise SOA - Designing IT for Business Innovation. O'Reilly 2006.
- [5] Forrester, "Enterprise and SMB software survey, North America and Europe, Q4 2008," Forrester Research, Business Data Service Survey, 2008.
- [6] R. Heffner, "Across all vertical industry groups, the majority of SOA users are expanding its use," Forrester Research, Research Report, May 2009.
- [7] C. Bartolini, A. Bertolino, E. Marchetti, and A. Polini, "Towards automated WSDL-based testing of web services," in Intern. Conf. on Service-oriented Computing (ICSOC'08), ser.
- [8] J. Offutt and W. Xu, "Generating test cases for web services using data perturbation," SIGSOFT Softw. Eng. Notes, vol. 29, no. 5, pp. 1–10, 2004.
- [9] L. Baresi and E. Di Nitto, Test and Analysis of Web Services. Springer, 2007.
- [10] G. Canfora and M. D. Penta, "Service-oriented architectures testing: A survey," in Software Engineering: International Summer Schools, ISSSE 2006-2008, Revised Tutorial Lectures. Springer-Verlag, 2009, pp. 78–105.
- [11] A. Barbir, C. Hobbs, E. Bertino, F. Hirsch, and L. Martino, "Challenges of testing web services and security in SOA implementations," in Test and Analysis of Web Services.
- [12] M. Utting and B. Legeard, Practical model- based testing, a tools approach. Morgan Kaufmann, 2007.
- [13] P. Baker, Z. R. Dai, J. J. Grabowski, Ø. Haugen, I. Schieferdecker, and C. Williams, Model-Driven Testing: Using the UML Testing Profile. Springer, 2008.
- [14] A. Milanova, A. Rountev, and B. Ryder, "Parameterized object sensitivity for points-to analysis for Java," ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 14, no. 1, pp. 1–41, 2005.
- [15] M. Greiler, H.-G. Gross, and K. A. Nasr, "Runtime integration and testing for highly dynamic service oriented ict solutions," in Proc. of Testing: Academic & Industrial Conference - Practice and research techniques (TAICPART'09). IEEE Computer Society, 2009.
- [16] M. Acharya, A. Kulkarni, R. Kuppli, R. Mani, N. More, S. Narayanan, P. Patel, K. Schuelke, and S. Subramanian, "SOA in the real world - experiences," in Service- Oriented Computing (ICSOC), vol. 3826, 2005, pp. 437–449.
- [17] S. Wiczorek and A. Stefanescu, "Service integration: A soft spot in the SOA testing stack," in Proceedings of the 5th Central and Eastern European Software Engineering Conference in Russia (CEE-SECR'09). To appear in IEEE Computer Society, 2009.
- [18] S. Ali, L. C. Briand, M. J.-U. Rehman, H. Asghar, M. Z. Z. Iqbal, and A. Nadeem, "A based approach to integration testing based on UML models," Information & Software.
- [19] M. Helfen, M. Lauer, and H. M. Trauthwein, Testing SAP Solutions. SAP Press, 2007.
- [20] J. Jacky, M. Veanes, C. Campbell, and W. Schulte, Modelbased Software Testing and Analysis with C#. Cambridge University Press, 2008.
- [21] A. M. Memon, "An event-flow model of GUI-based applications for testing," Softw. Test., Verif. Reliab., vol. 17, no. 3, pp. 137– 157, 2007.
- [22] M. Vieira, J. Leduc, B. Hasling, R. Subramanyan, and J. Kazmeier, "Automation of GUI testing using a model driven approach," in AST '06: Proceedings of the 2006 International workshop on Automation of software test. New York, NY, USA: ACM, 2006, pp. 9–14.
- [23] I. Craggs, M. Sardis, and T. Heuillard, "AGEDIS case studies: Model-based testing in industry," in proceedings of the 1st European Conference on Model Driven Software Engineering. Imbus AG, 2003.
- [24] A. Paiva, J.C.P. Faria, N. Tillmann, and R. F. A. M. Vidal, "A model-to-implementation mapping tool for automated Model-based GUI testing," in ICFEM'05, ser. Lecture Notes in Computer Science, vol. 3785. Springer, 2005, pp. 450–464.
- [25] SOA Testing, Available online at <http://www.crestechglobal.com/sts-soa-testing.html>.
- [26] Mikko Satama, "Event Capturing Tool for Model-Based GUI Test Automation", Available online at: practise.cs.tut.fi/files/publications/TEMA/Satama_final.
- [27] Kervinen, A., Maunumaa, M., Katara, M., "Controlling Testing Using Three-Tier Model Architecture", In Proc. of the 2nd International Workshop on Model-Based Testing, MBT'06, Vienna, Austria, March 25-26, 2006.