



## PRACTICAL GUIDELINES FOR SUCCESSFUL ERP TESTING

<sup>1</sup>AMEL AL-HOSSAN, <sup>2</sup>ABDULLAH S. AL-MUDIMIGH

<sup>1</sup>Department of Computer Science, King Saud University, Riyadh KSA

<sup>2</sup>Department of Information Systems, King Saud University, Riyadh KSA

### ABSTRACT

Testing of Enterprise Resource Planning (ERP) system comprises the activities that are used to validate business processes and the rules that govern them and removing or reducing maximum operational risk within available resources and time schedule constraints. This paper reviews some researches performed in ERP testing to identify the common challenges, failures and the proposed solutions. It also presents guidelines to help achieving success ERP testing. To support this paper with real life example, we included the testing methodologies and the best practices that are applied in the Government Resource Planning (GRP) system of King Saud University.

**Keywords:** MADAR, ERP, Testing, Hasib, Organization, GRP.

### 1. INTRODUCTION

Testing is one of the most critical factors in ERP implementation. It is usually conducted before a company fully deploys an ERP software package and it also continues even after the stage where the software goes live. So, the testing is considered to be a continuous activity. ERP companies have spent years of testing and debugging their systems. [2] Pointed out that in line with the volume and criticality of ERP implementations, testing occupy nearly 50% of the total ERP budget. The increasing ratio of testing from project budget is causing a big problem reflected in failed projects, missing deadlines and critical risks [4]. The success of the testing significantly drives the overall success of the ERP project. There are many different ways to manage the testing of ERP systems. All strategies and methods used in previous researches aim to reduce the cost of testing considering its coverage degree and the capability to reveal errors. The first part of this paper presents the testing levels and types of ERP systems. The second part discusses Manual and Automatic testing. The third part lists the obstacles and challenges during ERP testing. It also provides suggested solutions collected from previous literatures. The fourth part presents the relationship between the ERP testing and the quality of the system. The fifth part lists number of best practice methodologies and guidelines for success ERP testing. The sixth part presents a case study of the GRP system of King Saud University "MADAR" which is developed by software

provider "Hasib". The last section states the conclusion.

### 2. TESTING TYPES AND LEVELS OF ERP SYSTEMS

The first step of ERP testing is to determine every component in the system that needs be included in the test plan. In order to perform an effective testing, we must test the individual components and their interactions with the rest components of the system [1]. The testing types and levels differ among enterprises according to their activities, goals and objectives. [21] pointed that the performance testing and functional testing form the basic building blocks to ERP testing. Oracle Certified Partner [25] listed three key components of testing: Usability, Performance and Functionality. Software Testing Center of Excellence [24] presented eleven types of testing: Functional Testing, Data Handling and Integrity Testing, Systems Security Testing, Reliability Testing, Usability Testing, Integrated Performance, Stress Testing, Interface, Interoperability Testing, Regression Testing, Infrastructure Testing, Image Testing and Installation Testing. [27] Presented three levels of testing, Unit Testing, Integration Testing, and System Testing.

Nowadays, most modern enterprise systems have the Service Oriented Architecture (SOA) as underlying paradigm in this architecture the application is partitioned into logical units and encapsulates related functional units in a service [13]. The SOA systems often have more integration

points. The increase in environmental touch points leads to a very complex testing and data management problem [16]. Also the dynamic and adaptive nature of SOA makes most testing techniques not directly applicable to test services and service-oriented systems [17]

[13] Presented four testing layers for Service Oriented Architecture (SOA) systems as illustrated in figure 1.

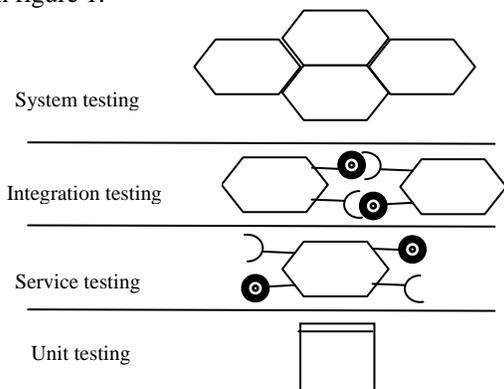


Figure 1: SOA testing layers [13]

The author [13] stated that the unit testing and service testing can be carried out just like any other Component Based System (CBS) or software architecture. Unit testing requires the knowledge of the source code; so this type of testing is known as “glass box” or “white box” testing. Typically, it is performed by members of the development team [27]. The focus of service testing is on the integration of the functional units inside the (service) component. In contrast to the CBS approach, integration testing has heterogeneous components with loosely connected interfaces. So, the adaptability and distribution of SOA demands additional considerations for integration testing [13]. Integration testing may use “glass box” or “black box” testing or both and it should be performed by an independent test team (not members of the development team) [27]. System testing in the SOA world can be defined analog to the classical definition for CBS. In practice, system testing (the upper testing levels) is based on high-level usage of scenarios and business requirements that have been defined by business analysts or customers. Therefore, UI-based testing is most appropriate method to carry out this test [13]. This level often includes operating the system in a variety of environments, different communication networks and other products that will interoperate with the system under test and it is validated under stress and load conditions [27].

The Testing Strategy Plan prepared for Accountable Business Transformation (ABT) program [19], uses V-model testing which consists of two levels of testing: Unit (Component) Test and Integration Test as shown in Figure 2.

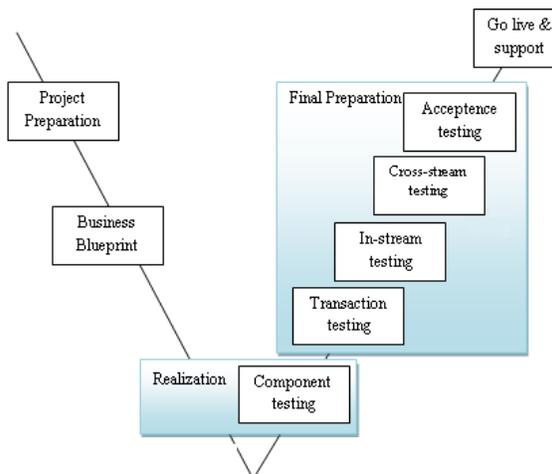


Figure 2: Testing Lifecycle [19]

The integration testing is composed of five basic levels: Transaction (testing of a single transaction), In-stream (covers chains of transactions), Cross-stream (end-to-end testing of integrated processes), Regression (similar to cross-stream testing but for unintended consequence) and User Testing or Acceptance testing (in-stream or cross-stream tests by the user community with the objective of formal validation). Testing Strategy Plan presents six types of testing, User Testing, System Testing, Load Testing, Stress Testing, Parallel Testing, and Security Testing. Most ERP systems mentioned in the researches have similarities in their testing levels, but they have differences in their testing types according to the level of security and reliability they require.

### 3. MANUAL AND AUTOMATIC TESTING IN ERP SYSTEMS

SAP can be considered as the world's leading provider for business software solutions. It has a large history of developing complex ERP systems [15]. SAP ERP users are required to perform comprehensive tests before the system goes live. Traditionally, organizations have been dealing with this challenge by using a manual trial-and-error approach to test their customized system. However, the shortcomings of this method make the ERP system more complicated. This includes, for example, inability to detect all problems, mounting

costs due to increased utilization of development and testing resources, extended implementation time, and inability to accurately calculate the project timeline and costs. To tackle the shortcomings of traditional approaches, there is a growing need for solutions to automate testing processes [3]. Smart automation of testing processes enable machines to complete tedious, repetitive tests on the system, while testing team members and process owners can perform other, more analytical tasks[4]. The test planning for automation must focus on the areas that are worth to automat [11]. Any test that has predictable results and meets one of the following criteria is a candidate for automation: The test is iterative, evaluates high risk conditions, impossible or costly to perform manually, requires multiple data values to perform the same action or the test is a baseline test run on several different configurations [5].

### 3.1. Automated testing methodologies

The Record/Playback Methodologies: in this methodology the testers record their manual test, and then play back the recorded scripts. It does not require programming skills [6]. When this methodology is used incorrectly, it can lead to untimely death of an automation effort [7, 8]. Data-Driven Testing Methodologies: in this methodology the tester uses the same script with the same or different data tables [6]. Data-Driven is a powerful methodology for both Hardware and Software engineering and testing [10]. A well-designed data-driven approach can make it easier for non-programming test planners to specify their test cases by writing them into the matrix [9].

Limitations of test automation are: [11] Listed the following limitations of Automation:

- Immature testing processes and methods
- Technical limitations of tools
- Need for development/training of test developers
- Difficulty in measuring return on investment of test automation

## 4. ERP TESTING CHALLENGES

Based on the previous researches of ERP testing, we can classify the challenges of testing into three areas as illustrated in figure 3.

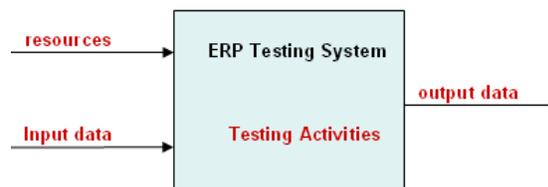


Figure 3: Testing System challenge areas

The first area concerns with the input of the ERP testing such as the resources and data used for testing, the second area concerns with how the testing activities has been processed, and the last area concerns with how to evaluate the output data of a testing system. In the following paragraphs, the challenges that were discussed in the literature will be presented.

### 4.1. The Size of Test Suite

One of the ERP testing problems is the size of the test suite. The test suite size becomes larger and larger due to the frequently generated test cases as the demands are continuously changed during the development of the software. This size of the test suite leads to the increase of the cost to re-run all test cases in the test suite [12]. [12] Presented two ways to optimize test suite. The first way is to reduce the test suite using algorithms that identify the reduced test suite with the same coverage of the test requirements of the original test suite. The other way is the prioritization of test suite. In this algorithm the test suites are sorted according to some coverage criteria. All reduction algorithms should consider both the coverage degree of test case for test requirements and the capability of test cases to reveal error.

### 4.2. Testing SOA enterprise systems

Another challenge is the need for new testing methods for the modern enterprise systems that are based on SOA and whether the existing testing approaches can be adapted. [13,20] Presented the key distinguishing factors for SOA systems that generate unique challenges for the testing activities. These factors are lack of code access, Dynamicity and adaption, lack of control, lack of trust and the cost of testing. The first three factors deal with technical challenges, while the remaining factors may have to be solved on the management level.



The management challenges group can be addressed by providing means of interaction between stakeholders, in order to share information and rights. The technical challenges group requires the black box testing techniques to be applied, as code access in a SOA environment is limited. Dynamicity and adaption require providing detailed information about interfaces to incompatibilities that are detected and handled automatically [13]. Therefore [13] assumed that Model Based Testing (MBT) will influence much greater impact in the testing process than in traditional industrial development systems, where modeling is carried out. The lack of control over system parts implies that tests will have to be carried out not only in the development of a service-based application, but also regularly after deployment, since the services run on an independent infrastructure and it might be updated without informing the consumer. Therefore, having automatic regression tests in this case is a natural conclusion [14]. [13] Pointed out that the GUI testing might be in many cases, the only possible approach for testing the functional and non-functional properties of a service-based system.

#### **4.3. Providing data for ERP testing activities**

ERP systems posed new challenges regarding test data that is used in regression testing [15]. The author defined the regression testing as the testing that is used for assuring the quality of software systems produced in several development cycles. This type of test guarantees that additional code or any changes do not affect the functionality already implemented in the system. It also guarantees that the requirements are always satisfied. Regression testing usually occurs at the end of the execution of a test phase. Depending on the number of fixes that have been incorporated during the test phase and the extent of any design changes required to fix defects, significant portions of the testing may need to be run again in a new test pass [27]. This type of testing requires a large amount of data [15]. The Author of [15] Identified four main challenges regarding the provision of test data for automatic testing of ERP software: system test data supply, system test data stability, input test data constraints and test data correlation. The following sections will present these challenges and the different solutions presented by the author.

**System test data supply:** ERP testing demands the insertion of common test data to the system during the testing preparation. The first solution is to write the data directly into the database, but this is difficult and the data could be inconsistent. The

second solution is more realistic where we fill the empty system with common test data by using the application and enforcing system data consistency.

**System test data stability:** It means keeping the core system data unchanged for regression testing. Since the changing of system data is part of common ERP functionality, it is strongly connected to this requirement. The first solution is to use the system data access rules to prevent the alteration of common test data, but enforcing write protections during test execution will cause a difference in behavior of the SUT compared to the delivered system. The second solution is to bind the concrete system data to abstract test cases during runtime. Another strategy to supply test data stability is the copying of master data tables in an initial system state to ease and speed up regular data resets.

In the context of ERP system testing the **input test data** should respect the syntax constraints (data types and ranges), semantic constraints (relation between the data) and contextual input data constraints (application context such as current time in the SUT).

**Test data correlation:** The relationship between system data and input data needs to be observed. It might even be impossible to determine in advance whether a certain input value should trigger a positive or negative system response, and hence the satisfaction of constraints might become nondeterministic. This constraint especially becomes prominent in regression tests where the provision of unique input test data might be problematic [15].

#### **4.4. Test Data Management (TDM)**

The automated tests create large quantities of data, so the companies have difficulty accessing, organizing, collating and reporting these data [22]. Test Data Management (TDM) system provides the capability to correlate and analyze test results from several engineering design groups, at different locations and over different time-periods to quickly understand and assess information you need [23]. So with (TDM), a company can scale as desired without linearly increasing the testing or data management effort [22].

A company may have large quantities of legacy data generated by old test systems that the company would like to include in future test data analysis. Companies can continue to use legacy test systems for many years if data from these systems is



successfully integrated into a modern enterprise TDM solution. To import the data from any data source in the legacy system, a test engineer first establishes a mapping between the structures of the legacy data and the database format of the enterprise TDM system. The tools and utilities provided with a good TDM system assists with this mapping and imports the data into the new TDM system [23].

## 5. ERP TESTING AND SOFTWARE QUALITY

ERP software quality evaluation system, developed on the bases of the testing. The testing can help to collect the information for evaluating the software quality. The quality evaluation system for ERP software can be established from the following issues that have been tested such as availability, functionality, performance, reliability and services [6,26].

The robust approach for testing an ERP system is centralized testing where there is a quality Assurance (QA) team for defining the standards and procedures for the various testing cycles and documenting the test plan, lessons learned, User's Acceptance Test (UAT) plan and the test strategies. The test procedures and test strategies include what test case templates will be used, how and what metrics will be reported, how peer reviews will be conducted, ensuring that all test requirements have been met, etc. In addition to the quality Assurance team under the centralized model, the testing team provides expertise for executing test cases whether manually or with automated test tools. The testing team also provides independent verification for the test results since the testers were not associated with the configuration of the transactions, workflow development, creation of user roles, or the development of the Reports, Interfaces, Conversions, and Enhancements (RICE) objects[28].

## 6. PRACTICAL GUIDELINES FOR ERP TESTING

### 6.1. Best practice in test automation

[5] Listed number of best practices that should be considered before implementing automated software testing. The first step is to prepare the applications for automation testing, identify the correct search criteria and define a standard display resolution. The second step is to prepare

automation team and processes by identifying the skills of team members, document the functions and how to use them, create an automation plan, define measurements for success, develop test cases and choose the right tool. The last step is to implement automated testing by documenting automated tests and create reusable and maintainable automated tests. Remember, incorrect use of a test tool's management functionality results in wasted time [18].

[11] Summarized the lessons learned from experts in test automation in the following points:

- Record and Playback approach is not effective when software application is data dependent.
- Test Automation is the same effort as software development
- It is not possible to achieve automated testing in all types of testing
- Error Recovery routine is very important for automated test suite to be successful.
- Test Automation could have bugs.
- Improved automation exercise test developers have been recruited instead of transferring software developers to automation team.
- Test scripting standards developed and automation team trained on the same.
- Reuse of test Scripts designed and developed
- Developed an efficient Error Recovery routine.
- Test Scripts developed for test data setup
- Test Script generation based on functional specification designed and used.
- Identified test areas that are most critical and adequate focus was given to achieve success in those areas.

### 6.2. Guidelines for best practice of testing

We are proposing in this paper the following points as general guidelines that we consider very important for ERP testing:

- Test manager should Encourage and support self-motivation within and outside test team to increase job satisfaction.
- Test manager should monitor team skills and performance
- Identify, analyze, and prioritize test objectives to guide all testing activities.
- Use flexible and adaptable approaches for testing.



- The requirements should be testable, reflect goals and objectives of customers.
- Design test cases that force defects to appear early in testing.
- Create a custom Test Metrics according to the enterprise.

## 7. CASE STUDY

### 7.1. *The Testing Experiment of King Saud University GRP system*

The Government Resource Planning (GRP) system of King Saud University is called "MADAR". It is deployed by Hasib software provider and it is based on Service Oriented Architecture (SOA). This system automates Finance, HR, Payroll system, Purchasing and Accounting. It is developed using .NET technology and it uses web based client interface using ASP.NET. The database management system for MADAR is MS-SQL Server. Hasib developed custom development package called "HGS.NET". It is an object oriented component based services.

The testing levels used in this study composed of eleven levels. We can describe them as follows: Unit Test, Smoke Test, Functional Test, Integration Test, System Test, Load Performance Test, Stress Performance Test, GUI Test, Negative Test, Acceptance Test and Regression Test.

In all the previous testing levels, they use different methodologies such as White Box Testing, Black Box Testing, Gray Box Testing, Manual and Automatic Testing. They also use tracking system to record facts about known bugs such as the time a bug was reported, its severity, the erroneous program behavior, and details on how to reproduce the bug; as well as the identity of the person who reported it and any programmers who may be working on fixing it. It also traces and manages any change request in the system.

The provider of KSU system lists number of points that affect the testing activities and the quality of the system:

- The customer forces the provider to quickly deploy the requested modifications with limited time that is not enough to fully implement the testing procedures as required by quality assurance policies.
- The customer has special type of functions and conditions that need separate test cases that were never done for previous customers before

such as the payment conditions for hourly paid instructors.

- The huge volume of data and the number of transaction performed by the system that required more time to test.
- The number of new requirements that are frequently requested by the customer after the deployment of the system that need more effort for testing

Above points should be considered during the process of planning and designing of flexible and robust test cases and scripts. They also emphasize on the importance of the completeness, accuracy and consistency of the test data, which affect the final results of the test.

The best practices of testing activities that Hasib had experienced can be summarized in the following points:

- Applying Quality Assurance standards and procedures over all SDLS.
- Simulate the client environment (HW, SW) to detect any problems early.
- Document the lessons learned from the previous project and plan for new strategies to avoid the problems in the future.
- Well documentation of the performed tests to permit the tester to create more accurate test cases that satisfy the system requirements.
- Improve the human and technical resources.
- Task allocation based on skills and experience of the test team members.

## 8. CONCLUSION

In this paper we have presented the challenges of ERP testing and the obstacles that are investigated by the previous literatures and the proposed solutions. The paper highlighted the important issues that should be considered in order to achieve successful testing for enterprise systems. We also presented some guidelines, best practices and lessons learned from the experts in this field. This paper presented the testing activities and best practices applied by Hasib to test MADAR system



in King Saud University. It also stated the difficulties appeared during test phases to enrich lessons learned from different experiments. We finally believe that the researches of ERP testing can be extended to cover major factors related to the environment where the ERP system is applied.

## REFERENCES

- [1] "An integrated approach to requirements and quality management. *The principles of requirements-driven testing*" IBM Corporation Software Group Route 100 Somers, NY 10589, U.S.A. Produced in the United States of America June 2009 Requirements and quality management solutions , White paper
- [2] Gerrard, P., "Test Methods and Tools for ERP Implementations", Windsor, 10-14 Sept. 2007, pp. 40 - 46 .
- [3] Yachin, D., "SAP ERP Life-Cycle Management Overcoming the Downside of Upgrading", External Publication of IDC Information and Data, September 2009, WHITE PAPER
- [4] Schliesser s., VP R&D Sapiens N.V., "An approach to ERP testing using services", [Software-Science, Technology & Engineering, 2007. SwSTE 2007. IEEE International Conference on](#) 30-31 Oct. 2007, pp. 14 - 21
- [5] Seapine Software, Inc., "Automated Testing Best Practices" , 12/2007, <http://downloads.seapine.com/pub/papers/AutomatedTestingBestPractices.pdf>
- [6] Wen, R.B. , "URL-Driven Automated Testing" , IEEE CONFERENCES, [Quality Software, 2001. Proceedings. Second Asia-Pacific Conference on](#) , 2001 , pp. 268 - 272
- [7] Boehmer B. , and Patterson B., "Software Test Automation – Developing an Infrastructure Designed for Success", Proceedings of the STAR East 2001 Conference, May
- [8] Shakil Ahmad, "ADVANCE DATA DRIVEN TECHNIQUES" Proceedings of the STAR East 2001 Conference, May 14-18, Orlando, Florida, USA, 2001.
- [9] Cem Kaner, "Software Realities Pitfalls and Strategies in Automated Testing", COMPUTER 00 18-9162 0 1997 IEEE, Vol. 30, No. 4; APRIL 1997, pp. 114-116
- [10] Scott Davidson, "The Scientific Method and Design and Test" IEEE Design & Test of Computers 0740- 7475 91 0 2000 IEEE, July-September 2000, pp. 142-143
- [11] Kishore C.S., "Automated Testing: A Practical Approach for ERP product" <http://www.cskishore.com/paperqai.asp>
- [12] Prabhu, J.; Malmurugan, N.; Gunasekaran, G.; Gowtham, R.; "Study of ERP Test-Suite Reduction Based on Modified Condition/Decision Coverage" IEEE CONFERENCES, Computer Research and Development, 2010 Second International Conference\_on\_2010 , PP. 373 - 378
- [13] Wiczorek, S.; Stefanescu, A.; " Improving Testing of Enterprise Systems by Model-based Testing on Graphical User Interfaces", Engineering of Computer Based Systems (ECBS), 2010 17th IEEE International Conference and Workshops on 2010 , PP. 352 - 357
- [14] M. Acharya, A. Kulkarni, R. Kuppili, 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.
- [15] Wiczorek, S.; Stefanescu, A.; Schieferdecker, I.; "Test Data Provision for ERP Systems" ,IEEE CONFERENCES, Software Testing, Verification, and Validation, 2008 1st International Conference on 2008 , PP. 396 - 403
- [16] howarth, J. ; ryan, R.;" Automating What You Can't See: *Testing Middleware for the Enterprise*"; june 29 2006
- [17] Canfora, G.; Di Penta, M; " SOA: Testing and Self Checking"; International Workshop on Web Services Modeling and Testing; (WS-MaTe 2006).
- [18] Dustin, E.; " Lessons in Test Automation" ; *Software Testing & Quality Engineering*



Magazine; September/ October 1999; Vol. 1, Issue 5; PP. 16-21

[19] Gauthier, L. " ABT program Testing Strategy Plan" ; Report, june 2008

[20] 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.

[21]ITC Infotech, Business friendly solutions; "ERP testing" ; 2010

<http://www.itcinfotech.com/erp/ERP-Testing.aspx>

[22] VI Technology, Inc; "Extending Test Into the Enterprise to Solve Test Data Management Challenges"; White Paper; 2005

[23] VI Technology, Inc ; "Are You Ready for Enterprise Test Data Management?"; White Paper; 2005

[24] Cigital All rights reserved; "Software Testing Center of Excellence" ; 2008

<http://www.cigital.com/services/quality/Software%20Quality.pdf>

[25]"Enterprise Applications Testing"; Leverage the Power of Objectivity; OraclePeopleSoft - ERP White Paper Series

[http://www.cognizant.com/html/content/bluepapers/Enterprise\\_Application\\_Testing.pdf](http://www.cognizant.com/html/content/bluepapers/Enterprise_Application_Testing.pdf)

[26] Xuesong, C.; Baoqiain, S.; Kai, S; " ERP Software Testing and Its Quality Evaluation System" ; Beijing UFsoft(group) CO. Ltd.

[27] LOVIN, C. ; YAPTANGCO, T. ; "Best Practices: Enterprise Testing Fundamentals" ; Reprinted from Dell Power Solutions, February 2006. Copyright © 2006 Dell Inc. pp 52-54

[28] Fajardo, J. ; " SAP R/3 Testing- Lessons Learned" ; article; 2005;

<http://sap.ittoolbox.com/documents/sap-r3-testing-14065>

#### **AUTHOR PROFILES:**

**Amel Al-Hossan** is working as a lecturer in department of computer science at Princess Nora University Riyadh, KSA. She got her Master degree in computer science and currently pursuing her PhD from King Saud University. Her research areas include: Software Engineering, Enterprise System, and Wireless Sensor Networks. She has published many research papers in her area of interest

**Dr. Abdullah Al-Mudimigh** is associate professor & senior IT consultant at college of computer and information sciences, King Saud University in Saudi Arabia. Currently he is a general manager of ERP project (MADAR) at King Saud University. Dr. Al-Mudimigh has many years of experiences working in IT. He also IT consultant to many government departments and firms. He is a member of many committees at many organizations, in Saudi Arabia and globally. He has been published at internationality recognized refereed journals and conferences. He also acted as a reviewer for many journals and conferences. His current research interests focus on the areas of IT business value, organizational and strategic impacts on IT, Electronic business, Customer Relationship Management (CRM), Supply Chain Management (SCM), and Knowledge Management (KM).