

# A Comparative Analysis and Evaluation of Different Agile Software Development Methodologies

Saleh M. Al-Saleem<sup>1,2</sup> and \*Hanif Ullah<sup>2</sup>

<sup>1</sup>Department of Information Systems, College of Computer and Information Sciences,  
King Saud University, Riyadh, Saudi Arabia

<sup>2</sup> Department of Computerized Based Testing, National Center for Assessment in Higher Education,  
P.O. Box 68566, Riyadh 11537, Saudi Arabia

## Abstract

Numbers of Information System development methods or agile software development methodologies are in practice. When any organization is going for IS development they will be looking for the IS development methods that are most complete and/or significant and also to fulfill their requirements and give them a complete solution. Agile software development methods have made their way into the software mainstream and have got the attention of software engineers and researchers due to their rapid growth. The purpose of this paper is to compare and investigate the three most practiced methods, that is Extreme Programming, Rational Unified Process and Dynamic System Development Method by using the Multi Criterion Decision Making tool. In addition, we analyze them on the basis of different aspects in order to find out which one is the most complete and significant technique for the development of Information System.

## Keywords:

*Information System, agile software development methodologies, XP, RUP, DSDM.*

## 1. Introduction

From the last two and half decades a number of IS development methods have been introduced and some of them have been practiced more or less [1]. All of these methods have their own pros and cons and as we know that software development is one of the volatile field, new technologies and concepts are taking place very rapidly and it seems that these methods are incomplete and are not competing with latest technologies [1]. Therefore new methods and techniques are taking place in which most of them are based on the old methods with some amendments and additions and some might be with new concepts [1]. Software engineering is a discipline, which is introducing new techniques and methodologies for the software development day by day [2]. Every organization is searching for a software development method that is organized in a way to deliver faster, better and cheaper solutions [2].

In this paper three of the most popular I.S development methods Extreme Programming (XP), Rational Unified

Process (RUP) and Dynamic System Development Method (DSDM) which are in practice nowadays in the market are discussed and compared with respect to different aspects such as process of the methods, practices, and pros and cons etc. Based on the facts and figures we will draw a conclusion regarding a method that is complete and more significant than other methods. The rest of the paper is organized as follows:

Section 2 consists of a background study regarding the techniques to be evaluated. Section 3 presents related works conducted in relevant areas. Section 4 describes the implementation of the proposed work and the results gained using the MCDM (Multi Criterion Decision Making). Finally section 5 briefly presents the conclusions drawn from the evaluation process.

## 2. Background

In this section we will explain Extreme Programming, Rational Unified Process and Dynamic System Development Method with respect to their life cycle and other development phases.

### 2.1. Extreme Programming

Extreme Programming is a disciplined way of software development which simplifies the project, gives quick feedback and brings courage to the team. It combines the whole team to the presence of a simple practice, where all the team members are getting enough feedback and could easily find that at what particular stage of the development stage they are and how they can get the target [3].

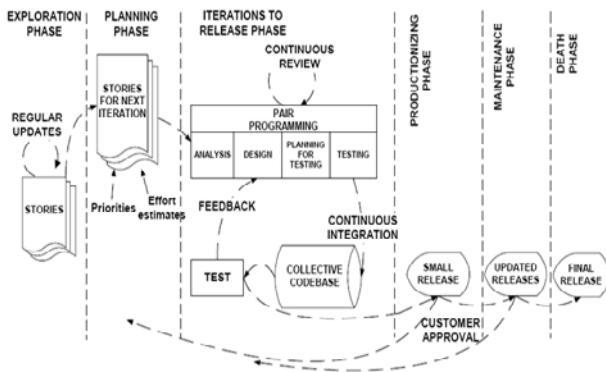
Extreme Programming came in to being due to some problems raised in traditional methods having lengthy development life cycle. It is simply the way to get the work done rapidly and easily [1].

### 2.2. Life cycle of XP Process

Extreme programming life cycle consists of the following phases which are discussed according to [1], [4], [5].

### 2.1.1. Exploration Phase

In this phase the customer wrote down the stories which are actually the requirements of the system. The customer puts these stories on story cards and in the meantime, the development team practice the tools which they are going to use in the entire project. So the tools are tested and the system get visualized by building its prototype. Depending on the size of the project this phase may takes weeks to months [1], [4], [5].



**Figure 1: Life cycle of XP process [1]**

### 2.1.2. Planning Phase

In this phase the requirement stories wrote down by the user in the exploration phase are prioritized according to the requirement of the system and the contents of the initial release are prepared [1], [4], [5]. The developer's team then estimates the requirements and works for the first release, manage the time and schedule the resources. This phase takes a couple of days and first release are prepared within two months.

### 2.1.3. Iterations to Release phase

In this phase the scheduled plan is divided into multiple iterations. Architecture of the system is created in the first iteration mainly concerned with the selected stories that focuses on the structure of the required system [5]. The customer decides which stories should be included in every iteration and perform the necessary tests that the customer wants. Each iteration take approximately 3-4 weeks to implement, and the system will be ready to use at the end of last iteration [5], [6].

### 2.1.4. Product ionizing Phase

In this phase the developed system is presented for its first release and are tested for extra performance and functionality. Some new amendments are also decided if they are part of the first release. New ideas and suggestions might be raised and are recorded for possible implementation later. The duration of this phase is normally very short and take almost one to three weeks [1], [3].

### 2.1.5. Maintenance Phase

The system is released to the customer in this phase. The XP team will be working on both sides i.e. customer support side and also on new iteration which will reduce the speed of the developer team. Some new people may also be introduced to the XP team in this phase, so the team will also be restructured [1], [3].

### 2.1.6. Death Phase

This is the final phase where the system is completed, and customers are satisfied from the performance of the system[1], [3]. All the entire work are documented and design, architecture, code etc. are finalized. The death phase also occurs if the desired outcomes are not fulfilled or if the system is going to be too expensive to afford.

### 2.1.7. XP pros and cons

#### Pros:

- XP is an iterative development method
- Trusts the developer
- Customer makes business decisions.
- Continual process improvement.
- Not tied to expensive tools.
- Development makes technical decisions.

#### Cons:

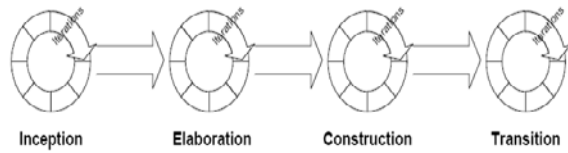
- Lightweight on process side.
- Dose not specify artefacts.
- A lot of customer involvement.
- Does not support too many tools

## 2.3.Rational Unified Process

RUP or Rational Unified Process introduced by "Philippe Kruchten, Ivar Jacobsen and others at Rational Corporation" based on unified modeling language (UML). It is an iterative method for Object Oriented system based on use cases that are used to model requirements and build foundation of the system [1].

### 2.2.1. Process of RUP

The life cycle of RUP is divided into four phases: Inception, Elaboration, Construction, and Transition. Each phase is then further divided into iterations. Each iteration have a purpose to produce an integral part of the software. Time required for each iteration may be as low as two weeks or as long as 26 weeks [1], [7], [8].



**Figure 2: RUP phases diagram [1]**

## 2.2.2. RUP phases

### 2.2.2.1. Inception phase

In this phase the system business state is studied. The most important use cases are determined, and the cost of the project is estimated in order to find that wither the project is acceptable or not. It is also determined that which resources will be needed for the entire project [9], [10].

### 2.2.2.2. Elaboration Phase

In this phase the developers have a close look at the system to determine the system architecture and to decide the system plan [4], [11]. This is an important phase in RUP where the developers analyze the risk associated with the system.

### 2.2.2.3. Construction Phase

In this phase all the development work is completed and different modules are integrated. Also some modules are released before going to the transition phase [9], [10].

### 2.2.2.4. Transition Phase

In this phase the software is in full operational form. User feedback are recorded and any critical problem exist in the system are removed. Any delayed part due to time constraint are completed [4], [11]. Beta tests for the entire system are carried out, users are trained, and the user documentation are prepared [4], [11].

## 2.2.2.1. Pros and Cons of RUP

**Table 1. RUP pros and cons**

<i>Pros</i>	<i>Cons</i>
<ul style="list-style-type: none"> <li>• RUP is an iterative development method.</li> <li>• It is a use case driven method.</li> <li>• Manageable, and traceable.</li> <li>• Lots of artefacts</li> <li>• Process is tuneable.</li> </ul>	<ul style="list-style-type: none"> <li>• Heavyweight on process side.</li> <li>• Customer involvement is not as much as in XP.</li> <li>• Tied to expensive toolset.</li> </ul>

## 2.4. Dynamic System Development Method (DSDM)

DSDM is a skeletal structure for rapid application development (RAD), originated in 1994 and maintained by an organization called DSDM Consortium. Unlike other methods, DSDM keep the time and resources fixed and functionality can be variable. All the developed systems will be delivered in a fixed amount of time [1], [3].

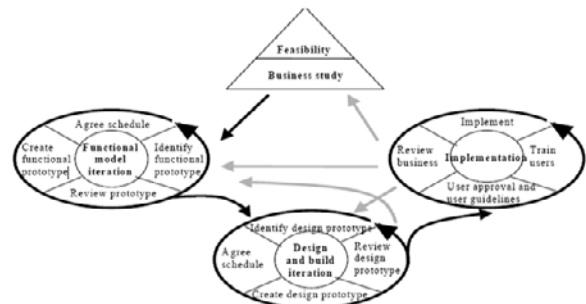
### 2.3.1. DSDM Process

DSDM process consists of few phases, that are feasibility study, business study, functional model iteration, design and build iteration, and implementation [1]. The first two phases are done at once, while the rest of the three phases are iterative and incremental. Every iteration must be completed in a fixed amount of time also called a time box [1], [3].

#### 2.3.1.1. Feasibility Study

In this phase the project at hand is judged for DSDM compatibility which means that the project is checked in order to conclude that whether it is suitable to do it with DSDM or not [10]. Also this phase has a concern with risk

and technical issues. Moreover feasibility report and outline plans are prepared in this phase [10].



**Figure 3: DSDM Process diagram [1]**

#### 2.3.1.2. Business Study

Basic properties of the business and technology are studied, the customer decides the priorities, and the customer from the affected area of business is given more consideration because they will be more involved at initial stage [1], [12]. Process is described according to the business definition. System architecture and outline plans are also defined [1], [12].

### 2.3.1.3. Functional Model Iteration

This is the starting iterative and incremental phase where the actual development starts. On the completion of each iteration, the next iteration is planned on the basis of the last iteration, and in this way the analysis model is implemented. Prototype code and analysis model is also prepared [2], [13].

### 2.3.1.4. Build and Design Iteration

Here the system is just ready to fulfill the basic agreed user requirements. This is an iterative as well, so the user will review the functionality and design and may ask for further developments [4], [14].

### 2.3.1.5. Implementation

In this phase the system is handed over to the users, users are trained and if the system has a large number of users then it may be done in a period of time [2], [5], [9]. This phase may also be iterated.

### 2.3.1.6. DSDM Pros and Cons

#### Pros:

- User is consider as the owner of the solution
- Risk is minimized up to enough extent by due to it iterative and incremental nature.
- The solution obtained fulfils the exact requirement of the user all the times.
- User is trained before the system implementation.
- The system implementation goes in very smooth way.

#### Cons:

- More user involvement can be danger some time if the user is not an appropriate one.

## 3. Related Work

The work that we have seen more relevant to our work is that of Osama Shoaib and Khalid Khan. In their work they have evaluated the agile methodologies i.e. Extreme Programming (XP), Rational Unified Process (RUP) and Dynamic System Development Method (DSDM) based on certain quality practices such as iterative and incremental development, communication and team work, pair programming, configuration and change management, simplicity and customer feedback etc. [2]. Also they have evaluated the above mentioned agile methods based on certain software quality factors such as efficiency, integrity, re-usability, maintainability, flexibility, timeliness and cost-effectiveness etc. [2].

Similarly the work done by M. A. Awad in his thesis is also more relevant to our work. In his thesis he discussed the two kind of methodologies i.e. heavyweight and lightweight. Heavyweight methodologies are considered as the traditional way of software development, detailed

documentation and expensive design [6]. While lightweight methodologies are also known as agile methodologies. Based on certain characteristics he evaluated the two methodologies. Also he discussed the strengths and weaknesses of the two opposing methodologies with some of the challenges that are associated with the implementation of agile processes in software industry [6].

Moreover the work done in [8] for the evaluation of three agile methodologies i.e. RUP, Catalysis and XP on the basis of CMM (Capability Maturity Model) framework seems similar to our work. In this book chapter the authors evaluated the three mentioned methodologies by considering certain characteristics such as requirements management, software configuration management, software quality assurance etc. at CMM level 2 and integrated software management, peer reviews, software product engineering, training program etc. at CMM level 3 [8].

## 4. Proposed Methodology

The research work in this article is carried out by using the MCDM tool (MakeItRational). MCDM is basically a decision making tool which draws the results based on certain steps [15], [16]. The main steps for making the decision are goal selection, Alternatives, criteria, sub-criteria, preference and finally result. The description about each step is given below.

### 4.1. Goal Selection

This part is basically concerned with the selection of goal. The main goal in our case is to evaluate XP, RUP and DSDM Techniques and to come up with a conclusion that which technique is more appropriate for the development of Information System [16].

### 4.2. Alternatives

This portion is concerned with the main alternatives that we are going to evaluate. In our case we have three main alternatives which are summarized in table 2.

**Table 2: Main Alternatives**

<i>Number</i>	<i>Name</i>
1	Extreme Programming
2	Rational Unified Process
3	Dynamic system development method (DSDM)

### 4.3. Main Criteria

We have identified seven main criteria based on which we will evaluate the agile methodologies. The identified criteria include Efficiency, integrity, Re-usability, Ease of use, Maintainability, Testability and Cost-effectiveness. The hierarchical block diagram of the entire process

clarifies the idea about the goal selection, main criteria and alternatives.

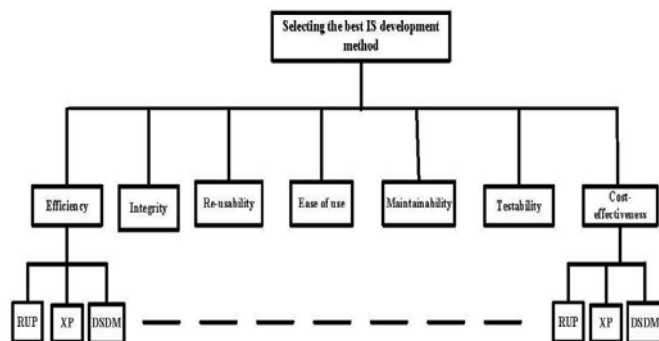


Figure 4: Hierarchical block diagram

#### 4.4. Preferences

Preference is basically concerned with the assignments of priorities. Priorities are assigned based on the following table.

Table 3: Priorities with their Importance

No:	Name	No:	Name
1	Equal Importance	6	Strong Importance plus
2	Weak Importance	7	Very Strong Importance

Efficiency vs. Maintainability	2 : 1	Cost-effectiveness vs. Ease of use	2 : 1
Re-usability vs. Maintainability	2 : 1	Cost-effectiveness vs. Integrity	3 : 1
Maintainability vs. Ease of use	3 : 1	Testability vs. Cost-effectiveness	2 : 1
Integrity vs. Maintainability	2 : 1	Cost-effectiveness vs. Maintainability	2 : 1
Testability vs. Efficiency	2 : 1		

3	Moderate Importance	8	Very Strong Importance plus
4	Moderate Importance plus	9	Extreme Importance
5	Strong Importance		

The detail about each preference assigned to each criterion is summarized in the following table.

Table 4: Evaluation in context of: Selecting the best IS development method

Criteria	Ratio	Criteria	Ratio
Efficiency vs. Integrity	3 : 1	Testability vs. Re-usability	2 : 1
Efficiency vs. Re-usability	1 : 1	Testability vs. Ease of use	2 : 1
Re-usability vs. Integrity	3 : 1	Testability vs. Integrity	2 : 1
Efficiency vs. Ease of use	2 : 1	Testability vs. Maintainability	2 : 1
Re-usability vs. Ease of use	2 : 1	Efficiency vs. Cost-effectiveness	1 : 1
Integrity vs. Ease of use	2 : 1	Cost-effectiveness vs. Re-usability	1 : 1

#### 4.5. Results

This section consists of some graphs illustrating the overall results of the evaluation process. From these graphs a number of conclusions are discussed.

The graph in figure 5. shows the ranking of alternatives according to their importance and severity. In the case of DSDM all the alternatives have the maximum value which concludes that DSDM is the most significant tool in all

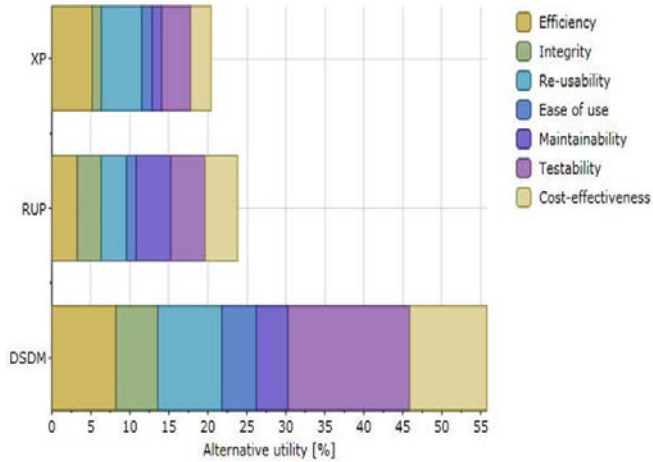
Alternative	Total	Efficiency	Integrity	Re-usability	Ease of use	Maintainability	Testability	Cost-effectiveness
XP	20.41	5.18	1.17	5.18	1.34	1.23	3.7	2.62
RUP	23.81	3.26	3.07	3.26	1.22	4.46	4.39	4.15
DSDM	55.77	8.22	5.37	8.22	4.42	4.05	15.61	9.89

Table 5: Alternatives Ranking

agile techniques [16]. A more detailed description of the above figure can be clarified from table 5 given above.

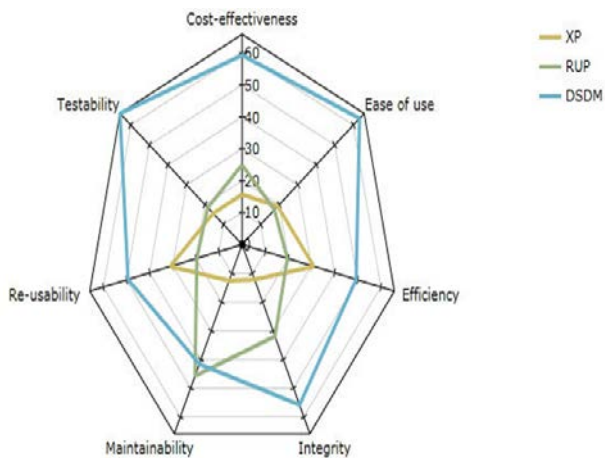
The above table shows the value in % for each criterion against each alternative. For example DSDM give an efficiency of 8.22%, while XP and RUP gives an efficiency

of 5.18% and 3.26% respectively. Similarly in the case of testability DSDM is more accurate because it has a value of 15.61%, while the others have a value of 4.39% and 3.7% which means that DSDM provides the best testing facility to its end users.



**Figure 5: Ranking in context of: Selecting the best IS development method**

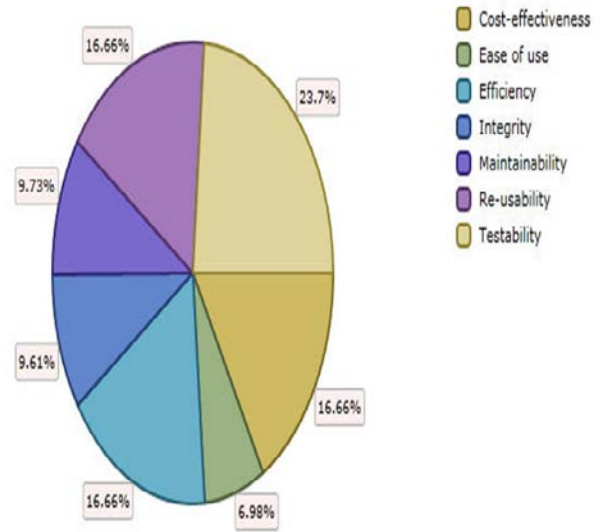
The graph in figure 6 illustrates the comparison of each criterion against each alternative. Based on the values against each criterion in the graph, it is clear that DSDM is the most significant and complete method in all three agile methodologies which can be used for the development of any information system [16].



**Figure 6: Alternatives Comparison**

Similarly the graph in figure 7 shows the importance and severity of each criterion. As we can see from the graph that testability has a value of about 24%, which means that testability is one of the most important factor that must be considered in order to evaluate the agile techniques. Moreover efficiency, re-usability and cost-effectiveness are

also the most important factors that must be brought into consideration in order to evaluate the agile methodologies.



**Figure 7: Alternatives weights**

## 5. Conclusion

From the above discussion and according to our understanding each one of these methods have some strengths and weaknesses, but according to our research study, understandings and based on the facts and figures concluded from the above graphs, DSDM is the most complete and significant method as compared to other agile methods that can be used for the development of any information system. But as we have mentioned earlier that each one have some pros and cons, DSDM also have some limitations but they can be ignored as the other methods have a lot of problems as compared to DSDM.

## References

- [1] Abrahamsson, P., et al., Agile Software Development Methods: Review and Analysis. 2002: VTT Publications. 112.
- [2] OSama Sohaib and K. Khan, The role of software quality in agile software development methodologies. TECHNOLOGY FORCES (Technol. forces): Journal of Engineering and Sciences 2010: p. 6.
- [3] Jeffries, R., What is extreme programming. 30/01/2015.
- [4] Cohen, D., M. Lindvall, and P. Costa, An Introduction to Agile Methods, in Advances in Computers. 2004, Elsevier. p. 1-66.
- [5] Fojtik, R., Extreme Programming in development of specific software. Procedia Computer Science, 2011. 3(0): p. 1464-1468.
- [6] Awad, M.A., A Comparison between Agile and Traditional Software Development Methodologies, in School of Computer Science and software Engineering 2005, University of Western Australia. p. 84.

- [7] Chan, F.K.Y. and J.Y.L. Thong, Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems*, 2009. 46(4): p. 803-814.
- [8] Pooley, R. and P. Wilcox, Chapter 11 - Evaluating process models, in *Applying UML*. 2004, Butterworth-Heinemann: Oxford. p. 161-173.
- [9] Chow, T. and D.-B. Cao, A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 2008. 81(6): p. 961-971.
- [10] Conboy, K. and L. Morgan, Beyond the customer: Opening the agile systems development process. *Information and Software Technology*, 2011. 53(5): p. 535-542.
- [11] Williams, L., *Agile Software Development Methodologies and Practices*, in *Advances in Computers*, V.Z. Marvin, Editor. 2010, Elsevier. p. 1-44.
- [12] Germain, É. and P.N. Robillard, Engineering-based processes and agile methodologies for software development: a comparative case study. *Journal of Systems and Software*, 2005. 75(1-2): p. 17-27.
- [13] Petersen, K. and C. Wohlin, A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems and Software*, 2009. 82(9): p. 1479-1490.
- [14] Vavpotic, D. and M. Bajec, An approach for concurrent evaluation of technical and social aspects of software development methodologies. *Information and Software Technology*, 2009. 51(2): p. 528-545.
- [15] Abdullah S. Alghamdi, Hanif Ullah, and I. Ahmad, Evaluating Encryption Techniques for C4I System, in *ISCA 23rd International Conference on Computer Applications in Industry and Engineering*. 2010: Imperial Palace Hotel, Las Vegas, Nevada, USA.
- [16] Abdullah Sharaf Alghamdi, Hanif Ullah, and S.U. Ali. Evaluating Chaos-based vs. Conventional Encryption Techniques for C4I System. in *International Conference on Computer Communications and Networks (ICCCN 2011)*. 2011. Lahaina, Hawaii, USA.



**Hanif Ullah** is an Applied Researcher in Computerized testing department at National Center for Assessment in Higher Education, Riyadh, Saudi Arabia. He received his MSc. Degree in Information Technology from Quaid-e-Azam University Islamabad, Pakistan in 2007, and BIT (Hons) degree in Information Technology from Iqra University, Karachi, Pakistan in 2004 respectively. He worked as a Research Assistant and Lecturer in the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. His research interest includes: Wireless Sensor Networks, Biometric Security and Comparison & Evaluation of different web based and computer based testing tools.



**Saleh Al-Saleem** is an Associate Professor in the College of Computer and Information Science, King Saud University, Riyadh, Saudi Arabia. He is also a Manager of the Computerized testing department at National Center for Assessment in Higher Education, Riyadh, Saudi Arabia. Dr. Saleh received his PhD degree from Wayne

State University, Michigan, USA, in 2001, in the field of computer science (Evolutionary Computation). He received his Master degree in computer science from Ball State University, USA in 1996, and his BS degree in computer science from College of education, King Saud University, Saudi Arabia in 1991 respectively. He served as the dean of admission & registration as well as the head of IT and e-Learning in Shaqra University. Previously he worked as head of Information Technology department at Arab Open University. Before that he worked as the head of Computer Technology department and faculty member in Riyadh College of Technology. Dr. Saleh current research interest includes: evolutionary computation, Text Classification, ERP, BPM, and e-Learning.