IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)
ISSN(E): 2321-8843; ISSN(P): 2347-4599
Vol. 3, Issue 10, Oct 2015, 75-86

TOOLS FOR SOFTWARE ENGINEERS

HINA MAHMOOD[1] & MUHAMMAD SAJID REHMAN[2]

[1] Lecturer, Computer Science, Qurtuba University of Science & IT, Peshawar, Pakistan

[2]Assistant Professor, Computer Science, Qurtuba University of Science & IT, Peshawar, Pakistan

ABSTRACT

The technological advancements in 21st era expand the field of software engineering by introducing new products on daily basis. Depending upon the organization, the software developers are using different techniques and methods for software development. These development techniques must be unblemished in order to produce a high quality product. To achieve this goal, software engineers are using several tools that are required during various software development stages. These tools vary from organizations depending on the process. In this paper the researcher will try to cover tools used by software engineers in different phases of software development.

KEYWORDS: Software Engineering, Tools, Requirements Elicitation, Analysis, Design, Coding, Testing

INTRODUCTION

According to Roger Pressman, an application which is developed using a systematic procedure and following specified rules and regulations for its construction, operation and maintenance is termed as Software Engineering [1]. Every software goes through various stages during its development. A software process comprises of requirement elicitation, analysis, design, coding, testing and maintenance. There is a wide variety of tools that are used by developers to build softwares efficiently and effectively with minimum defects. Software developer select appropriate tool and evaluate its performance to ascertain the scope to which it meet the specified requirements [2].

This paper focuses on a literature based survey of software development tools.

SOFTWARE PROCESS MODELS

Several methodologies have been proposed by researchers and used by developers for software development. This portion identifies some of the software development approaches which are:

- Linear Sequential Model
- Prototyping
- Incremental Model
- Spiral Model
- XP
- SCRUM

These approaches have been categorized in the major phases of requirements gathering, requirements analysis, design, coding, testing and maintenance. Each and every phase have some particular tools which tries to covers the defects and produce a fruitful artifact [1]

TOOLS REQUIRED FOR DIFFERENT PHASES

Requirements Elicitation

The first phase of every software development methodology is requirements elicitation. It is a complex process for acquiring, understanding, seeking, uncovering and elaborating requirements. This section involves customers, stakeholders, documentation and other existing system [3]

TOOLS FOR REQUIREMENTS ELICITATION

Doors (Dynamic Object-Oriented Requirement System)

This tool is developed by IBM Rational to facilitate stakeholders that can actively participate in communication and provides a requirement management environment for managing change in requirements and accessing requirements database [2]

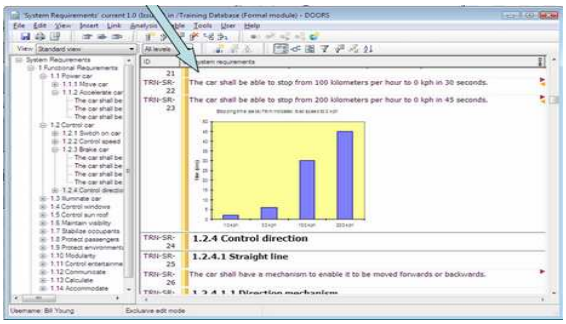

Figure 1: IBM Rational DOORS Tool

Requisite Pro

Requisite Pro is developed by IBM Rational for management of requirements, its documentation and to ensure requirements traceability and impact analysis and organize changes that occur during requirements gathering [2]

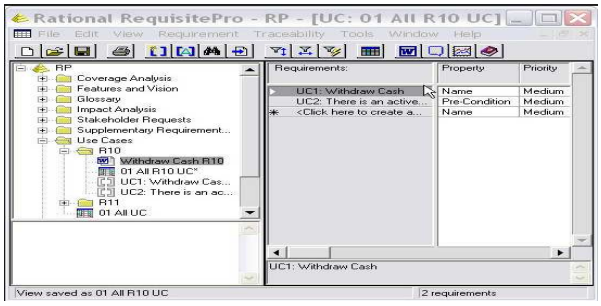

Figure 2: IBM Rational Requisite Pro

Groupware

Requirements gathering are done in a variety of ways. Developers may get requirements from stakeholders directly having face to face communication, having discussion boards or through audio and video conferencing.

Groupware includes a large variety of tools that are required for requirements gathering. These tools facilitate customers through online group discussions using TeamWave and GroupSystem [3].

Enterprise Architect

Sparx Systems developed this UML tool covering several phases of development from elicitation, analysis up to maintenance. It incorporates requirements management with other activities of software development [2]

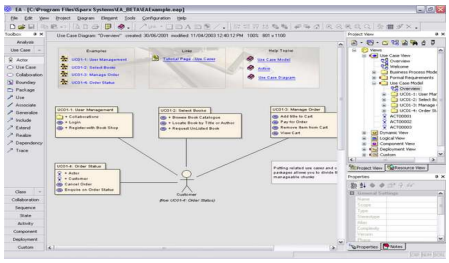

Figure 3: Enterprise Architect (Sparx System)

CORE (Vitech)

Core Product suite was developed by Vitech which manage requirements, its modeling, validation and verification and provide documentation in an effective way. The behavior of the System and its flow control is assessed through UML activities and sequence diagram.

This tool provides traceability generating documentation from database, presenting multiple modeling notations and change impact analysis [2]

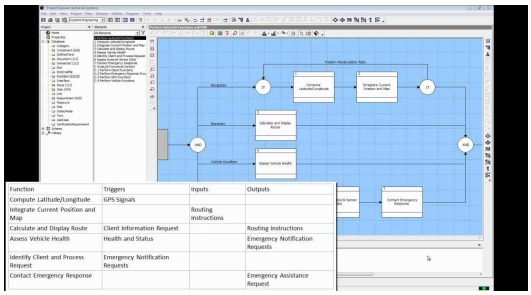

Figure 4: CORE (Vitech)

MKS Integrity

The integrity tool was developed by MKS and its main focus is on requirements validation. All changes in requirements are managed by this tool.

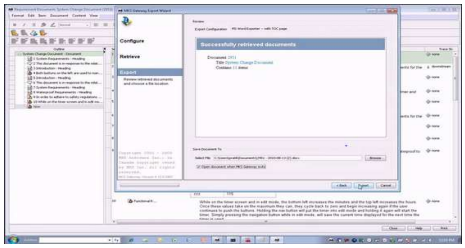

Figure 5: MKS Integrity

XPSwiki

It is an open source tool which acquires and schedule requirements using web browser. This content management system is very effective and popular among developers. Swiki enables the development team by defining input forms and add structure to pages. Web is used for different user stories, task assignments, and any relevant data and it can be accessible from programmer's workstations [4].

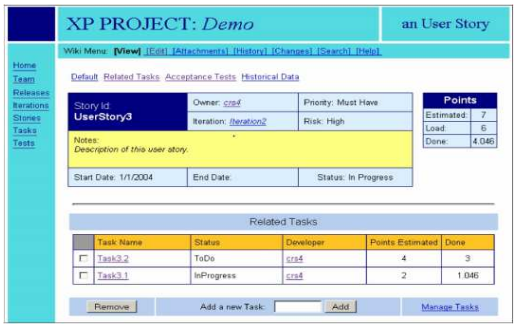

Figure 6: GUI of XPSwiki

XP4IDE

It is an internet based tool use XPSwiki server to connect. This tool is used for user stories, documentation of file, work flow and time estimation during different phases [4].

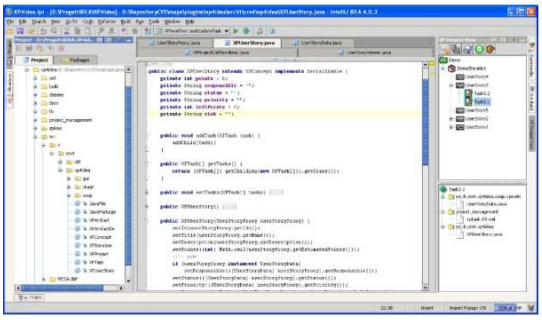

Figure 7: The GUI of XP4IDE

Requirements Analysis

In requirements analysis phase, all the requirements which are gathered during requirements elicitation phase are analyzed. All the behavioral details of the system, its performance and problem domain are analyzed thoroughly by a software analyst in order to build software [1]

TOOLS REQUIRED FOR ANALYSIS

Requirements Analysis Tool (Rat)

The syntactic and semantic analysis of a requirements document is done through user-defined glossaries and semantic web technologies. RAT uses two interfaces to describe its analysis process [5]

Requirements Checker: It is an interactive interface like MS Word used for spelling and grammar checking.

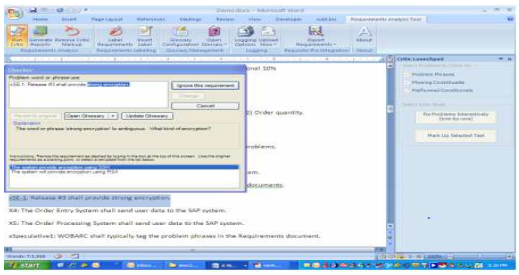

Figure 8: Interactive Requirements Checker

Requirements Tagger: Using this tool the entire document is analyzed. All the requirements and action are highlighted through its corresponding glossaries.

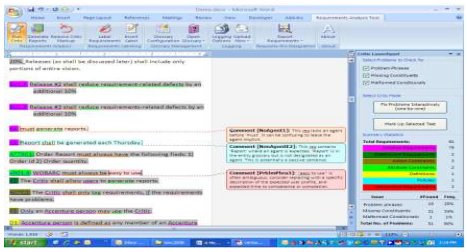

Figure 9: Comments Generated Using Requirements Tagger

QARCC (Quality Attribute Risk and Conflict Consultant)

The analysis tool QARCC is based on WINWIN process model. It is a knowledge based tool used to identify and analyze the conflicts in early development cycle. It emphasizes on the quality attributes to be explored, after which it suggests possible solutions [6]

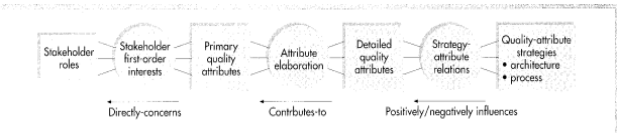

Figure 10: QARCC Knowledge-Based Structure

QuARS (Quality Analyzer for Requirement Specifications)

This tool allows an analyst to determine ambiguity problems in natural language and identify the linguistic defects to analyze the requirements to be more consistent and concise to a certain topic. The effectiveness of QuARS analysis depends on the completeness, precision and adequacy-to-domain of the dictionaries [7]

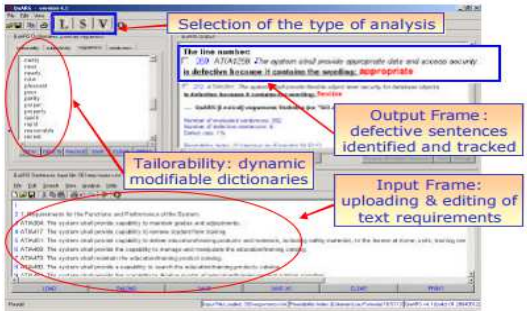

Figure 11: QuARS GUI for the Defect Identification Functionality

Design & Development

Designing is an important phase in software development life cycle. It comprises of four main attributes of a program: data structure, software architecture, interface representation, and procedural details [1].

TOOLS REQUIRED FOR DESIGN & DEVELOPMENT

IBM Rational Rose

It is a UML based development of applications. Software developers, designers, architects and analysts manage their requirements and communications with the help of different models using this software. Rational Rose family comprises of many different product editions for small and enterprise level business modeling. It also supports development of UNIX and Linux based applications [8].

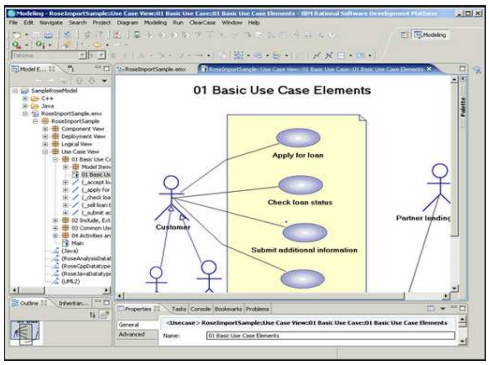

Figure 12: IBM Rational Rose

IBM Rational Software Modeler

It is a UML based tool for designing and it allows information to be communicated from all aspects of development project [2].

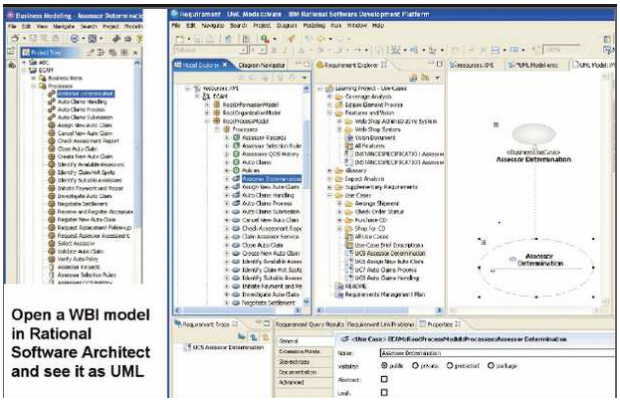

Figure 13: IBM Rational Software Modeler

Microsoft Visio

A tool by Microsoft which is used for diagrammatic representation of models using flow charts, network diagrams and work flow diagrams [2].

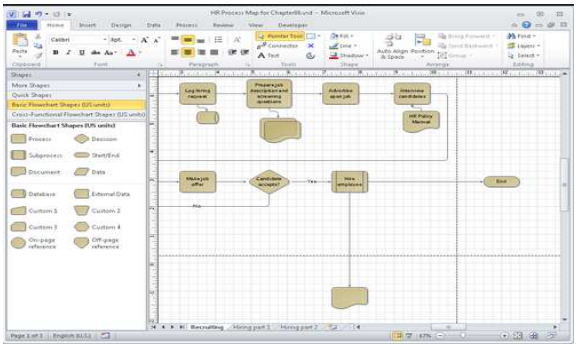

Figure 14: Microsoft Visio

IBM Rational Rhapsody

It is a visual environment based on UML used for graphical notations to understand requirements in order to generate code and speed up the process of development [2].

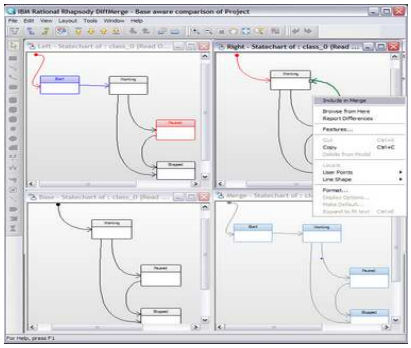

Figure 15: IBM Rational Rhapsody

IBM Rational Software Architect

This tool is used for developing and designing architecture for different programming languages applications like C++ and java [2]

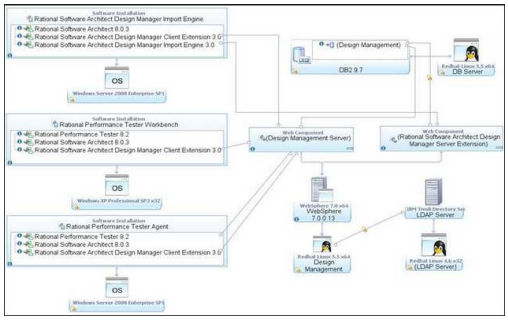

Figure 16: IBM Rational Software Architect

ENTERPRISE ARCHITECT (Sparx Systems)

IT systems and business modeling uses a UML design and analysis tool. It facilitates built-in reporting and documentation. Sparx systems support traceability from analysis to design, coding, testing and maintenance [2].

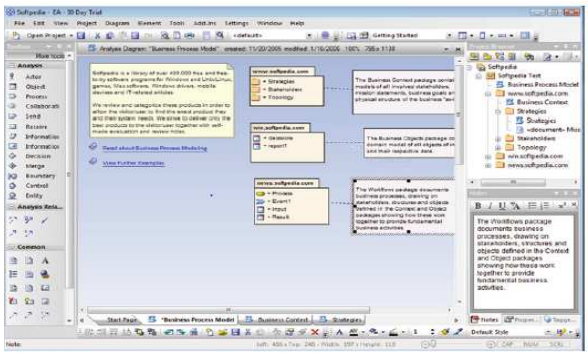

Figure 17: Enterprise Architect (Sparx Systems)

The Reload Learning Design Editor (LDE)

It is a user friendly interface using Eclipse platform. It supports various suitable user roles and design approaches [9]

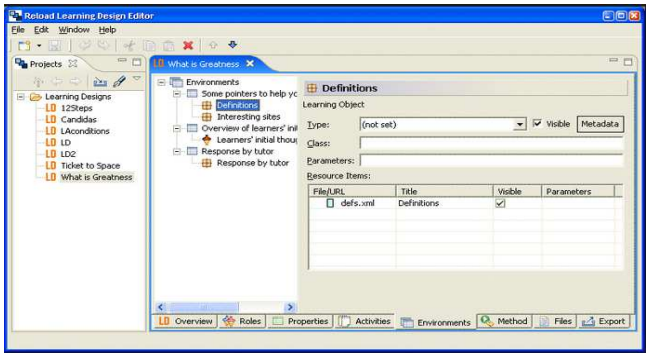

Figure 18: Reload Learning Design Editor (LDE)

MSC: Message Sequence Chart

This tool supports consistent and automated checking of message exchanges as machine readable. (Holzmann, et.al (1997))

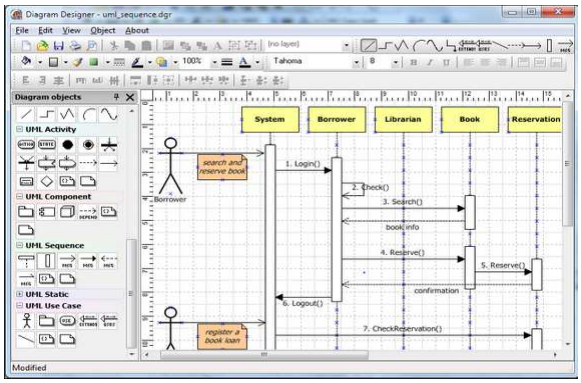

Figure 19: Message Sequence Chart

POGA pictures of graph algorithms

It is a graphical tool for constructing and analyzing directed label graphs. (Holzmann, et.al (1997))

Code Generation

In code generation phase the design made by software engineer must be translated in machine understandable format. Coding depends on the designing of the software.

TOOLS REQUIRED FOR CODE GENERATION

Source Code Editor

A text editor which is used for editing the source code and performs syntax checking of code as the code is written

Example: Microsoft Visual Studio [2]

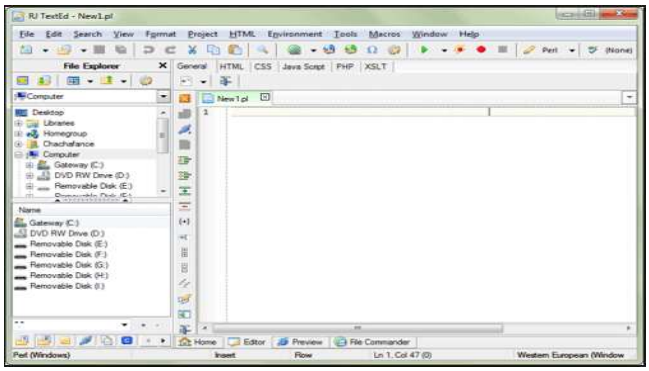

Figure 20: Source Code Editor

Compiler or Interpreter

Compiler is that computer program which produces an executable program by translating source code to target code. Throughout the process of conversion of languages it produces lexemes by lexical analysis, do parsing and generating code.

An interpreter is that program which executes instructions written in a programming language. It do translation of code line by line. Code is translated into intermediate representation which is then executed immediately [2]

Build Automation Tools

It involves scripting to automate the build process. Tasks that are performed are compilation of source code, linkage of the object code and building the executable software; performing automated tests and reporting results; reporting the build status; and generating release notes [2].

IBM Rational Purify Debugger

Software programs are tested and debugged using this tool. It identifies several breakpoints in the code and execution is carried out step by step [2].

Visual Source Safe (VSS)

VSS is a control management system for source code and binary files developed by Microsoft Corporation. It is used by small software development organizations. It allows multiple users to place their source code and work products under version control management [2].

The PV Tracker Tool

This tool is used for requesting changes in communication and tracking issues. It allows the developers to link the affected source code files with issues and changes. With the help of this tool, managers are allowed to determine team progress and to prioritize tasks [2].

The IBM Rational Clearcase

This IBM Clearcase tool allows software code and other software deliverables to be placed under version control management. It is an efficient tool for handling large number of files and supports standard configuration management tasks such as checking in and checking out of the software assets as well as labeling and branching. Objects are stored in repositories called VOBs [2].

Clearquest Tools

Clearquest tool tracks the defects in source code and allows the versions of source code modules that were changed to be linked to a defect number in Clearquest [2]

LDRA Testbed Tool (Liverpool Data Research Associates)

The complexity of source code is determined by LDRA tool. It automatically generates reports of code assessments including examined files and covers the clarity, maintainability and testability of code [2]

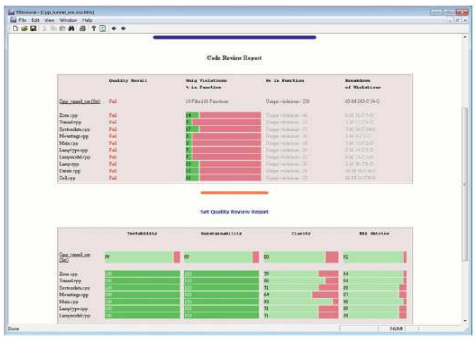

Figure 21: LDRA Testbed tool

LDRA TBvision tool

The LDRA TBvision tool is used to check the quality of code and investigate faults and avoidance measures. The results are represented using various reports and graphical notations [2]

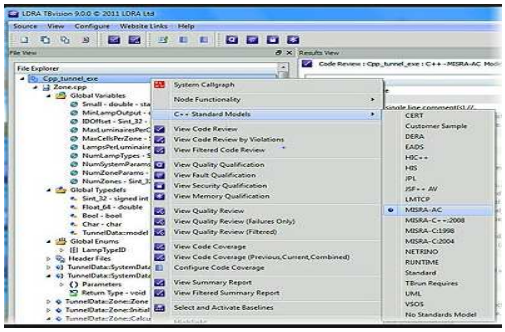

Figure 22: LDRA TBvision tool

RATS Tools (Remote Administration Tools)

For static code analysis, an open-source tool RATS tool is used that provides multi-language support for C, C++, Perl, and PHP [2]

Fortify Tool

Security vulnerabilities in several languages such as C, C++ and java are identified by developers and software engineers using this tool [2]

Parasoft Tool

It helps developers to identify coding issues that lead to security, reliability, performance and maintainability issues later [2].

Testing

The critical part of software development is testing which starts when the coding phase comes to an end. Testing includes the assessment of each and every portion of the statement to explore the issues, ambiguity and errors left in program.

TOOLS REQUIRED FOR TESTING

Test Director

A web-based test management tool is developed by Mercury. This management tool comprises of several different modules including requirements, test plan, test lab, and defect management. These four modules provide a consistent repeatable process for gathering requirements; planning and scheduling tests; analyzing results; and managing defects.

The Test Director tool is now a part of Quality Center developed by HP [2].

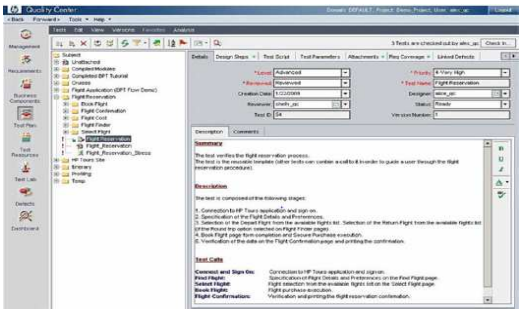

Figure 23: HP Quality Center

Winrunner Tool

It was developed by Mercury. The major focus of this tool is to enable defects in applications by capturing, authenticating and replaying user interactions automatically. Regression testing is major concern of Winrunner tool.

The Winrunner tool has been replaced by HP Unified Functional Testing software which includes HP Quick Test Professional and HP Service Test [2]

LoadRunner

Mercury developed the LoadRunner testing tool that allows the software application to be tested with heavy load having hundreds of users using simultaneously [2]

CONCLUSIONS

The process of developing software is a tedious job which is done by a software development team. The team members are divided into different categories according to their specialization in a particular stage of developing software. These developers include elicitors, analysts, designers and testers, each use specific tools for accuracy in the development process.

This research covers some of the software development tools in each and every phase that are used by different developers.

REFERENCES

1. Roger S.Pressman(2001) *Software Engineering: A Practitioner's Approach*, New York, NY, McGraw-Hill
2. O'Regan, G. (2010). *Introduction to software process improvement*. Springer Science & Business Media.
3. Zowghi, D., & Coulin, C. (2005). Requirements elicitation: A survey of techniques, approaches, and tools. In *Engineering and managing software requirements* (pp. 19-46). Springer Berlin Heidelberg.
4. Pinna, S., Mauri, S., Lorrai, P., Marchesi, M., & Serra, N. (2003). XPSwiki: An agile tool supporting the planning game. In *Extreme Programming and Agile Processes in Software Engineering* (pp. 104-113). Springer Berlin Heidelberg.
5. Verma, K., & Kass, A. (2008). *Requirements analysis tool: A tool for automatically analyzing software requirements documents* (pp. 751-763). Springer Berlin Heidelberg.
6. Boehm, B., & In, H. (1996). Identifying quality-requirement conflicts. *IEEE software*, (2), 25-35.
7. Lami, G., Gnesi, S., Fabbrini, F., Fusani, M., & Trentanni, G. (2004). An automatic tool for the analysis of natural language requirements. *Informe técnico, CNR Information Science and Technology Institute, Pisa, Italia, Setiembre*.
8. A UML visual modeling and application development solution(n.d.).In *IBM Rational Rose Family*, Retrieved 21st September,2015,*http://www-03.ibm.com/software/products/en/ratirosefami*
9. Milligan, C. D., Beauvoir, P., & Sharples, P. (2010). The Reload learning design tools. *Journal of Interactive Media in Education*, *2005*(1), Art-8.
10. Holzmann, G. J., Peled, D. A., & Redberg, M. H. (1997). Design tools for requirements engineering. *Bell Labs Technical Journal*, *2*(1), 86-95.