Knowledge Management in Software Enterprise

S.Mala, *Research Scholar, Madurai Kamaraj University, Madurai, Tamil Nadu.*
Dr. K. Alagarsamy, *Associate Professor, Madurai Kamaraj University, Madurai, Tamil Nadu*
A.Saranya, *Assistant Professor, V.V.V College for Women, Virudhunagar, Tamil Nadu*

Abstract

Effective knowledge management of the testing process is the key to develop the quality of software testing. Knowledge management has dissimilar features in software testing. One of the maximumsignificant research questions is how to efficiently integrate the knowledge management with the software testing procedure so that the knowledge effects can be extent and reclaimed in software testing organizations. In this paper, the current state of knowledge management in software testing was evaluated; knowledge management a process was proposed towardsa knowledge management system in software testing was intended and executed. Simultaneously numerous key technologies are reflected, such as knowledge map, validity threats, software testing, Test management and Metaphysical.

Keywords: *Knowledge management, Software testing, Metaphysical, knowledge map.*

I. INTRODUCTION

The significance of knowledge management that has developed a hot spot in international achieves fields. In order to improve its modest power, numerous enterprises have initiatively taken knowledge management into their essential business process, which efforts the development of consultation commercial whose new business scope is knowledge management, and then a lot of software tools and systems about knowledge management mustremained developed by IT enterprise. The functions and services of knowledge management are executed by knowledge management technology. The initiatives cannot effectivelyimplement knowledge management without the support of knowledge management technology, because it is the foundation for constructing knowledge management system and the driving force for executing knowledge management. Over ten years of exploration in knowledge management stimulates and markets many IT tools for knowledge management, but not all of them content the particular necessities of the enterprise. It is particularly obvious in particular fields. Knowledge about software testing, skills, experience, and stimulation are all very significant. Software testing is a knowledge-based movement. Though, if they do not have open views, abundant testing capabilities and skills, the testing quality can't be guaranteed.

Though the existing multipurpose knowledge management models and technologies have extra or fewer touched numerous problems, we select theories and technologies that are combined strongly with the field of software testing, which could help us find activeapproaches to solve the problems.

The aggregate complexity of software systems mutual with the beginning of distributed improvement models put more force on software organizations to achieve organizational knowledge and intellectual assets. Also, there is animportant loss of intellectual capital due to operatethroughput, restricted or limited knowledge. The acceptance of knowledge managementvalues can help software testing specialists to advance knowledge reuse and to inspire management discussions across the organization. There are numerous benefits of relating knowledge management in software testing such as:

a) Increasing test effectiveness,
b) Decreasing prices, time and effort,
c) Determination and application of more suitabletesting techniques,
d) Determination and application of more suitedtesting techniques,
e) Increasing the quality of results,
f) Supporting decision-making procedure.

Explicit knowledge testing can be recognized and retrieved by multiple personalities, e.g., in test manuals, processes, test items, test preparation, test enterprisequalifications, testing logs. Implicit testing knowledge is subjective and hard to document as it regularlyprocesses test executioninvolvements and considerations with software testers etc. Insufficient knowledge managementcomplete software testing mains to numerous negative significances. Lowreprocess of software testing knowledge, difficulties in software testing knowledge modification, a poor sharing environment of software testing knowledge, difficulties in optimal planning possessions.Testing experience, as well as testing knowledge, isnecessary to gain a deeper thoughtful ofthe used testing techniques.Though,testers do not incline to part the knowledge ormaterial that they increase when using numerous testing techniques.

This suggests that they missachance of sharing involvements and learningfrom each other, which restrictions their overallknowledge.Numerous testers are self-educated and have incompleteeducation on the subject. They necessitateadditional preparation. This limited knowledgealso consequences in a limited view about softwaretesting methods. Technology transfer amongresearch and industry is often limited; insignificance, not all new testing methods arestraightpractical in industry.Testers improvementnumerous types of knowledge andexperiences from their work in software projects.Sharing this knowledge can help to avoid makingsimilar errors and enhance testing accomplishments.Competent organizational knowledge sharingnecessitates establishing efficient knowledge managementperforms forknowledge creation, documentation, and management.The primary impartial of knowledge management in softwaretesting is to transfer testing knowledge and involvementbetween personalities in the same way astesting certification as well as developing tacitknowledge for supporting test design, implementation,and interpretation.Knowledge management supports test planning,test resultexploration and test consequences. Thetest design stage is also deeply dependent onknowledge management as it includesanswers the test conditions and objectives and choosing the relevant information to implement planned test cases.

Knowledge also benefits to establish the satisfaction standards against the testing consequences. Knowledge management supports testing techniques assortment as it is often based on testers' knowledge and perception, gained from numerousbases, such as testing the precedingforms of the system, involvement in analyzing and fixing the imperfections, working on improvement and maintenance as well as working with similar software systems. Finally, knowledge management approaches help to escalation the effectiveness and efficiency of product testing. Relating Knowledge managementin software testing is important to increase the testing level and improve software quality.

II. KEY TECHNOLOGIES

A) Metaphysical-based knowledge representation

Metaphysical is consequent from philosophy and is aboutexistence and essence. In recent years it is used as atechnique of knowledge demonstration, knowledgedistributionand reuse in computer modeling.In knowledge demonstration of software testing,relative thoughts, qualities and relations are designatedby using metaphysical. There are forms, positions, projects, staff and knowledge level insystem.

Metaphysical are classified according to the degree of formality

a) Semi formal
b) In- formal
c) Rigorously formal

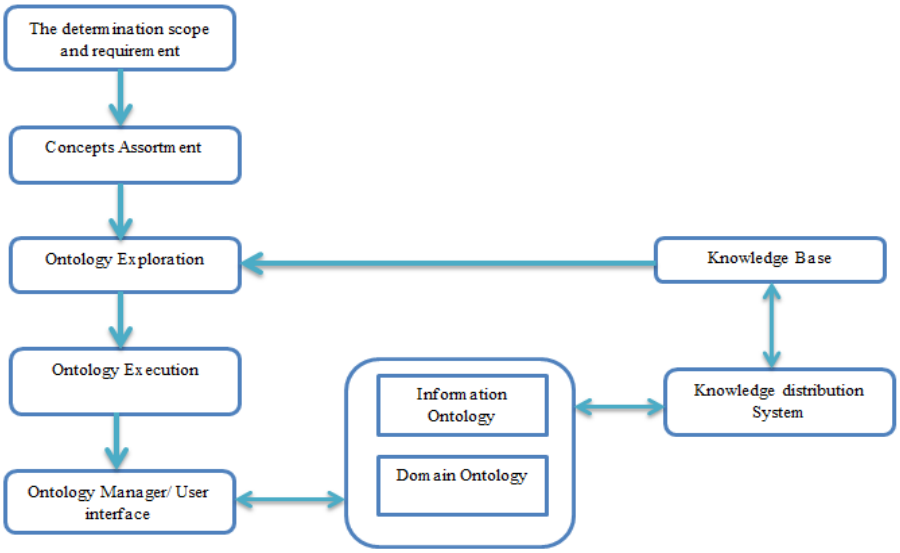


Figure 1 The framework of metaphysical-based knowledge management system

B) Knowledge management model

In knowledge executive of software analysis,active information distribution necessity is implemented toestablish channels of statement amongorganizing controls. Authorizing to the important for informationin software analysis, relative information transfer can beachieved in time. The level of knowledge reclaim canalso be enlightened by using effective knowledgetransfer.

Because the knowledge is changingpowerfully, an effective organization must be completeto meet those functions stated exceeding. Thus, acknowledge management model oriented softwareanalysis process is planned and shown in figure.

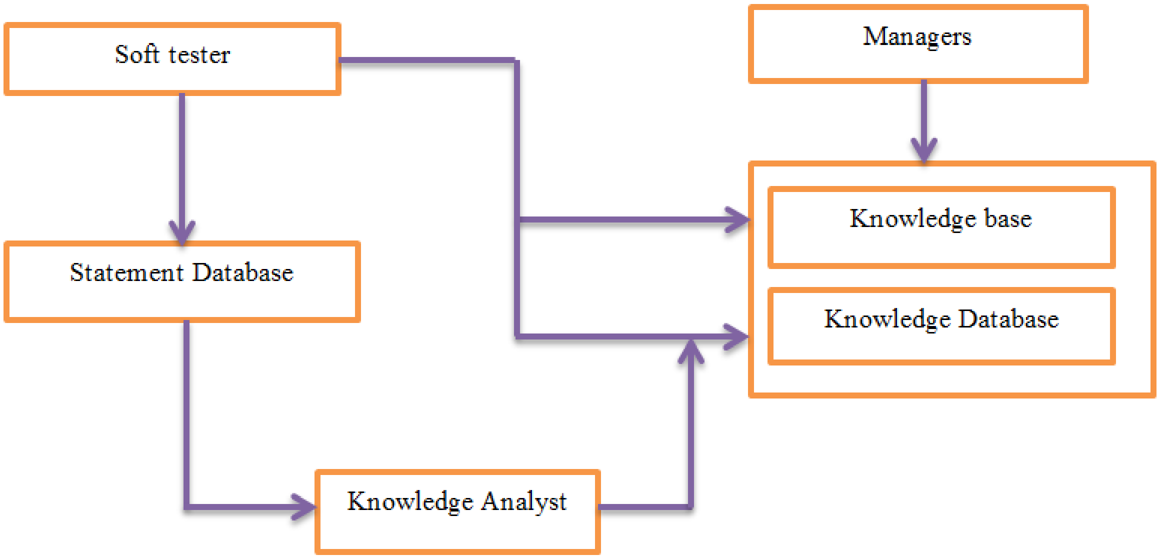


Figure 2 Knowledge Management Model

C) Assembling knowledge map

Knowledge map is a standard catalogue about knowledge. The knowledgesource shown by knowledge map could be departmentname, team name, specialist's name, connected person'sname, filename, bibliography, event number, patentnumber, or knowledge database index. However, theknowledge content is not comprised. It is a guide to savetime in tracing the knowledge source.Good software testing knowledge managementplatform must also offer tough software testingknowledge cataloging. According to the appliedexperience and methods, five knowledgefields are added into software testing. They aredeveloping semantic, database, operating system, and software testing tools, connected knowledge for thetesting project. In this knowledge map, the knowledgelevel is separated into five levels in each filed. They areunderstanding, associate, mastership, conversance,and authority. Description for respectively level is obviouslydesignated and effortlessly to be assessed. Every employee'ability is assessed by these standards. The assessmentprocedure should be done by the employee, team,executive, and knowledge analyst cooperatively.

Thedefinitions of five levels of knowledge are given asfollows:

1. Conception*:*

Controls has used the technologyfor half a year and occupied part in a project. Theycan orientation technology documents or helpfiles to need their necessities, know the consequenceof the technology.

2. Associate*:*

Staffs can grip more than 50%of key technology and has charity it for a yearand occupiedportion in more than two projects byusing it.

3. Mastership:

Staffs can grasp more than 60% ofkey technology and has used it for two yearsand taken part in more than three projects byusing it.

4. Conversance*:*

Staffs can grasp more than 85%of key technology and has used it for five yearsand taken part in more than five projects byusing it.

5. Specialist*.*

Staffs can grasp more than 95% ofkey technology and has used it for seven yearsand taken part in more than eight projects byusing it.

Affording to our research, whenassessing staffs knowledge stages, the topthree levels can be promotedinevitably interms of staffs' working knowledge and utilitytime. But when staffs will be assessed asconversance or specialist, the usefulness time andschemes which are taken portion in by them areonly essentialsituations. Even if they meetthe environments, they are not always to beevaluated as conversance or authority; theirlevels must be modified by knowledgeanalyst by hand.

D) Metaphysical-based search and sorting

Metaphysicalgenerally acts as knowledge database inknowledge repossession subsystem. It categorizesperceptionsof software testing, designates relation restrictionamong concepts, and conceptsthorough knowledgedatabase which contains concrete perceptions, qualities,associations, and incidencesamongmodels.

Whenrecovering knowledge, the correlative ideas orcharacteristics are found conferring to the users' requirements.Start from there, the metaphysicaldata will bepatterned to see if it is connected to the concepts orqualities. Reasonablyintellectualrepossession is thenattained.

E) Design of knowledge management

The method is knowledgeexecutivestage oriented software analysisprocedure. The system usages the knowledge life-cycle organization as a guide. It can help originalities to store, achieve, search, and share all types of knowledge by using the knowledge forms. It can evaluate the knowledge level of staff by using the knowledge map. Using knowledge map will develop a symbol for intelligent staff. The system will confirm the staff who has knowledge by indicators, and that will develop the culture of knowledge-sharing in the initiative.

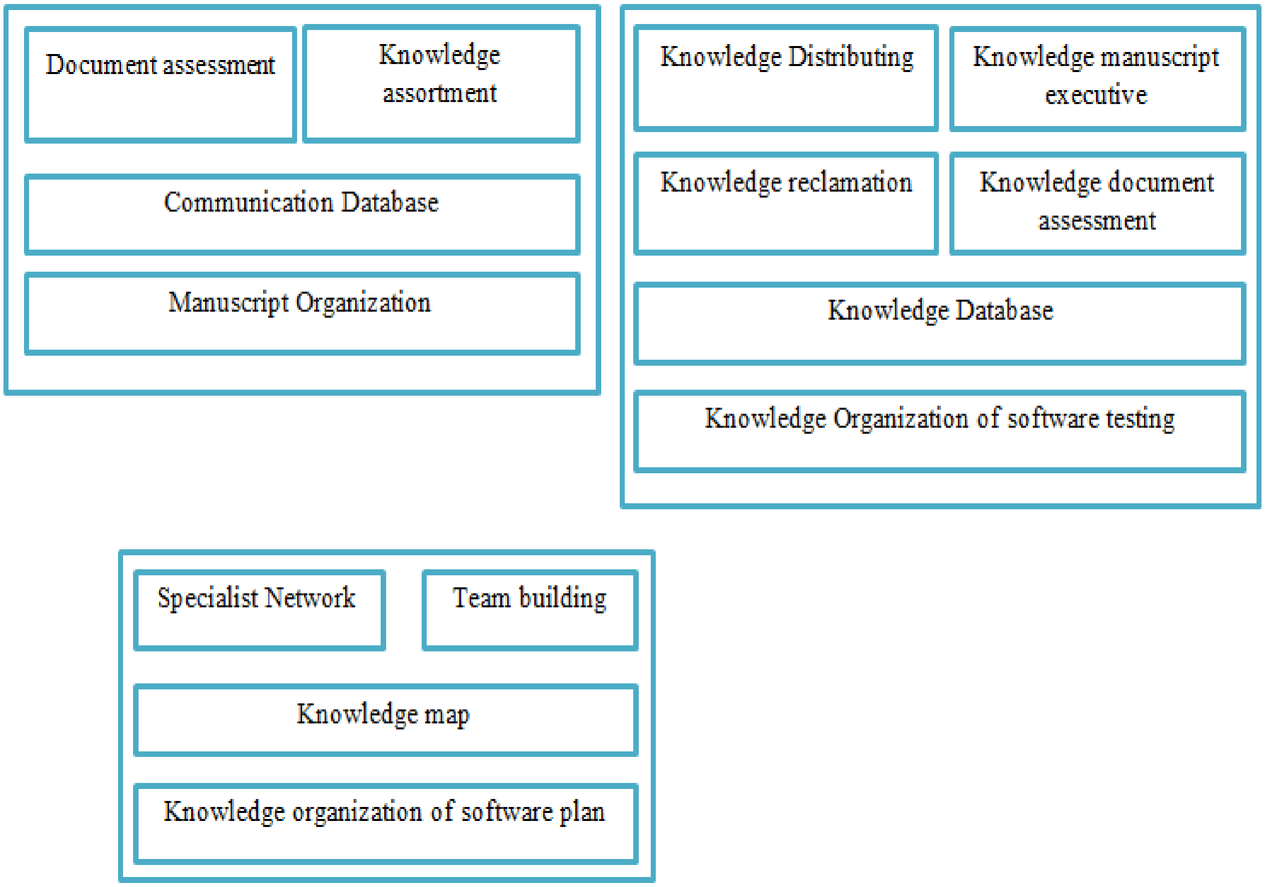


Figure 3 Knowledge Management Design

III. VALIDITY INTIMIDATIONS

Validity intimidations under the collectivesegment ofthe thesis are conversedconferring to the four validityclasses.

A) Internal validity

Threats are decreased by generatingand conservinganevaluationprocedure which includedthe particulars of the search sequencepreparationand start set identification, presence andprohibitingstandards used, the quality valuationbeing approvedout.

The risk for decisionerror was reduced by execution the independentassessment of the two authors who laterassociated and conferred the results. Both authorscontrolled closely composed and discussed anyquestionable cases. Furthermore, internal validitythreats are alleviated by following the planningandquality calculationnorms. Lastly,there is stagnant some risk that the studied optimistictesting consequences are the result of other phasesthan relating knowledge management techniques. It is intended toexplore this phase in future work when theserelations are explored in part.

B) Create validity

Applications on many potentialconfounding aspectsirrespective of whethera study could detention the planned knowledge,i.e. to attain the goals and purposes. One ofthe main concerns for this exploration is multipledescriptions of knowledge management. This threat was moderatedby accepting the well named definition by Davenport. Thesearch thread structure might be one of the conceptvalidity pressures in this study. Therefore,the examine string was iteratively expressed withextensive debatesamong the authors. Next,data extraction could also be the basis of validitythreats. To avoid these threats, manager'sassistance was established and all apprises at respectivelystep were sent for approval.

C) External validity

Itconsiders the capability togeneralize results outside the studied context.Most of the studies fall under the instance study research category with high rigor and significance scores as most of them wasaccompanied in industrialcontexts. Thus, the outcomes can beconsidered industry pertinent and are more generalized.For the studies that received low rigorand relevance scores, it remains to be determinedif the ideas suggested in these studies have highgeneralizability.

D) Reliability

It considers the degree of repeatabilityand whether the data and analysis depend ona specific assistant. To strengthen dependability,each step of the snowballing procedure was recognized,including the database exploration. Thesame relates to each step of data assembly andanalysis and they can be backpedaled, if needed.The quality valuation of the selected papers was confirmed by using rigor and significance criteriaconferring to objective calculation criteria. Thepossessions and featuresrecognized from the paperswere mapped with the research questions toattain the objectives of the study.

IV. SOFTWARE TESTING PERSPECTIVE

Revisions on knowledge management in software testing havecaptivated on dissimilarphases of software testing. Recognizedon the selected alterations, six highestmodules were restrained, designated below. Notethat, translation to association scheme, one study can impartiality more the one explorationthoughtfulnessregardingthe testingviewpoint.

1) Testing Procedure:

The focus is on managing knowledge in the perspective ofagiven testing process.

2)Test Item:

The focus is on managing knowledge about test item, forsupporting, e.g., testing item reuse.

3)Testing Stage:

The study reflects the application of knowledge management in a specificlevel test, such as unit testing, system testing, combination testing, and reversiontesting.

4)Testing Method:

The focus is on handling knowledge about testing methods, directing at helping testers to select better suited testing methodfor scheming test cases.

5)Other Party Testing:

The study converses knowledge executiveapplied to situations inwhich testing is capable by a other party.

6) All-purpose:

This type is used to categorize those papers that deliberate knowledge executivein software testing in general,without concentrating in any exactphase ofsoftware testing.

A) Knowledge management perception:

Similarlyto thepreceding,trainings on knowledge executive in software analysis also kindness on dissimilar aspects of knowledge management. Based onthe designatedstudies; the classesdesignated below were measured. Again, a studycan distance more the one exploration focus regarding the knowledge executive perspective.

B) Knowledge Management Model:

The study deliberates a model for organizationknowledge, considering knowledge procedures, and, ultimately somefeatures of it, such as knowledge carriers.

C) Knowledge Representation:

The study deliberatesfeaturesconnected to howto signify testing knowledge.

D) Knowledge Packing:

The study goes outsidephases related to knowledgedepiction, focusing on how to pack it. Studies classified in thistype are also classified in the previous.

E) Knowledge Catching:

The study addresses phasesconnected to how toobtain and store testing.

F) Knowledge Elicitation:

The study goes outsideparts related to knowledgeapprehending, conversing also methods to produce knowledge from experts.Studies classified in this kind are also classified in the Knowledge Capturinggroup.

G) Knowledge Recovery:

The study statements aspects regardingrepossessionof testing knowledge. In this case, the user is answerable for searchingknowledge items.

H) Knowledge Dissemination:

Respects pro-actively distributing testingknowledge.

I) Knowledge Assessment:

The study methodsfeatures related to the developmentof the testing knowledge previously stored, such as assessment andmaintenance.

V. PROPOSED SYSTEM

We are evolving the testing knowledge management thresholdfor this proposed work.The theoretical model of Roost was used as the initial point for requiring theTesting knowledge managementthreshold. However, this conceptual model does not deliverspecificsconcerningthe possessions of the concepts considered. Such properties can be recognizedconferring to the characteristics of the group´s test environment, such as informationproviding by the tools used for managing software testing. Thus, informationfrom one of the schemesdeliberate in this work was used as the source for recognizingcharacteristics and specifying the Testing Knowledge organizationthreshold.Use cases in white are testing-specific features comprised in thiswork. Developer is the central actor, signifying all kinds of professionals involvedin the software development procedure. Knowledge Manager signifies a user withspecific authorizations, assuring access to structures inherent only to a KnowledgeManager.

A) Use cases:

The different kinds of use cases are mentioned below:

***a) Create Knowledge Element*:** This use case permitsdesigners to make aknowledge element.

b)Create Conversation-connected Knowledge: This use case permits developersto register a Conversation-connected Knowledge.

***c)Create Lesson Knowledgeable*:** This use case permits developers to register aLesson Knowledgeable.

***d)Create Extracted Item*:** This use case permits the developer to register anExtracted Item.

***e)Create Test Case*:** This use case allows designers to register a TestCase.

f)Comprise Test Consequence:

This use case permits the inventor to comprise a testconsequencecomparative to a test case.

g)Comprise Issue:

This use case permits the developer to register an issuereporting an occurrence.

***h) Include Occurrence*:**

This use case consents the developer to report an occurrencerelated to a test result.

i)Conversion Knowledge Item:

This use case allows the knowledge managerto conversion a knowledge item.

***j) Remove Knowledge Item*:**

This use case permits the knowledge managerto remove a knowledge item.

k)Pre-assess Knowledge Item:

This use case allows the knowledgemanager to pre-assess a knowledge item, manufacture it obtainable, refusingit or choosingauthorities to assess it.

l)Assess Knowledge Item:

This use case allows a developer to make adetailed assessment of a knowledge item, to support the knowledge managerin making choices about whether the item should be accepted or rejected.

***m)Imagine Knowledge Item*:**

This use case permitsdesigners to imaginethe particulars of a knowledge item.

n)Imagine Test Case:

This use case permits developers to imagine thedetails of a test case.

o)Search Knowledge Item:

This use case allows the developer to searchfor knowledge items available according to knowledgeable parameters.

p)Exploration Test Case:

This use case permits the developer to exploration for testcases conferring to informed parameters.

q)Significance Knowledge Item:

This use case permits the designer to significance theeffectiveness of a knowledge item checked.

r)Find Authorities:

This use case permits the creator to find and choicespecialistswith a desired profile, as well as inspecting the profiles of experts found.It works as a Yellow Pages system.

As seen in Figure 4, Testing Knowledge Management threshold is divided into several panels:

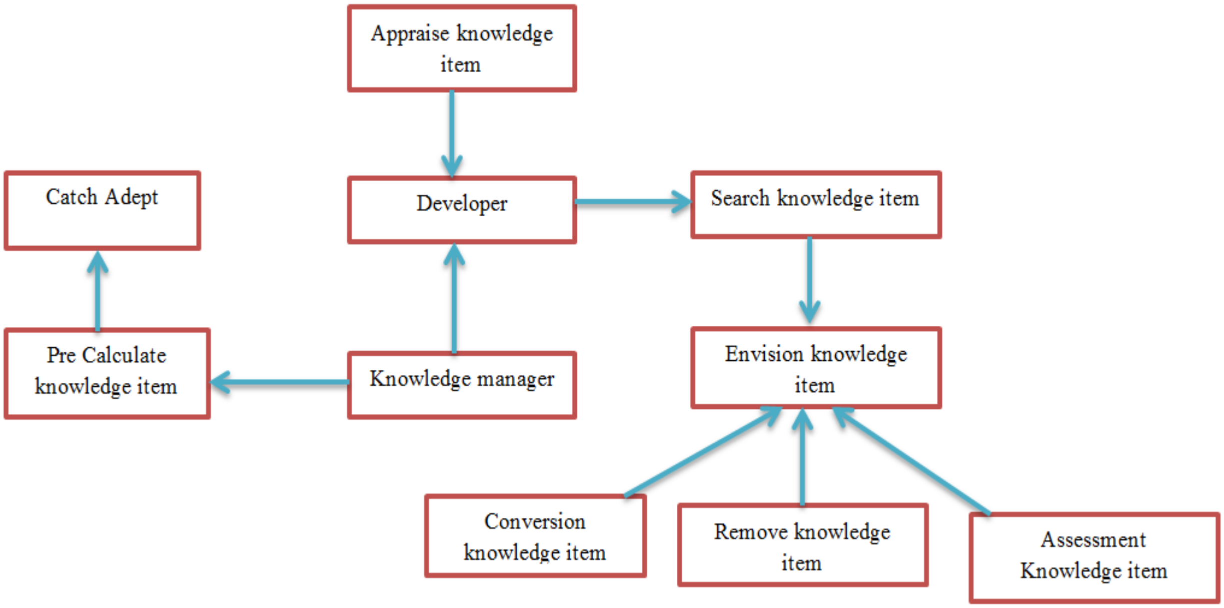


Figure 4 Testing Knowledge Management Portal Use Case Diagram

- **Functionalities:**

Comprises the main threshold functionalities, which are formationand search for knowledge substances.

- **Tools to Support Association:**

Integrates tools for associate collaboration,specifically Yellow Pages and Debate Forums.

- **Contribution in threshold:**

Comprises information about current authenticateduser'scontribution in the threshold. From this board, the user canassessment the knowledge items created, assessed, valued and to be evaluated byit.

- **Number of Associates:**

Reports the number of active members in theassociation. From this board, the user can view a list of all the associationmembers.

- **EssentialBoard:**

It is the boardsituated in the central section of the page.It displays information connected to the portal functions being achieved.

- **Number of Knowledge Items**:

Informs the quantity of existing knowledgesubstances per type.

- **Most Recent Items:**

Presents the knowledge items that were extra recentlycreated.

- **Most Accessed Items:**

Presents the most retrieved knowledge matters inthe portal.

VI. PERFORMANCE ANALYSIS

One of the mainchallenges in managing testing knowledge is to efficiently participate knowledge management withsoftware testing so that knowledge substances can be common and reclaimed in testingorganizations. Moreover, managing testing knowledge is not an informal task, andthus it is improved to start with a small-scale enterprise. Initially, it is required toidentify important knowledge items of a sub-topic of software testing to be distributedwith the knowledge management. The review aim was to define a situation to apply knowledge managementin software testing.

The following testing activities were considered:

1. Scheduling Test
2. Design Test Case
3. Coding Test
4. Test Execution
5. Test Results Analysis

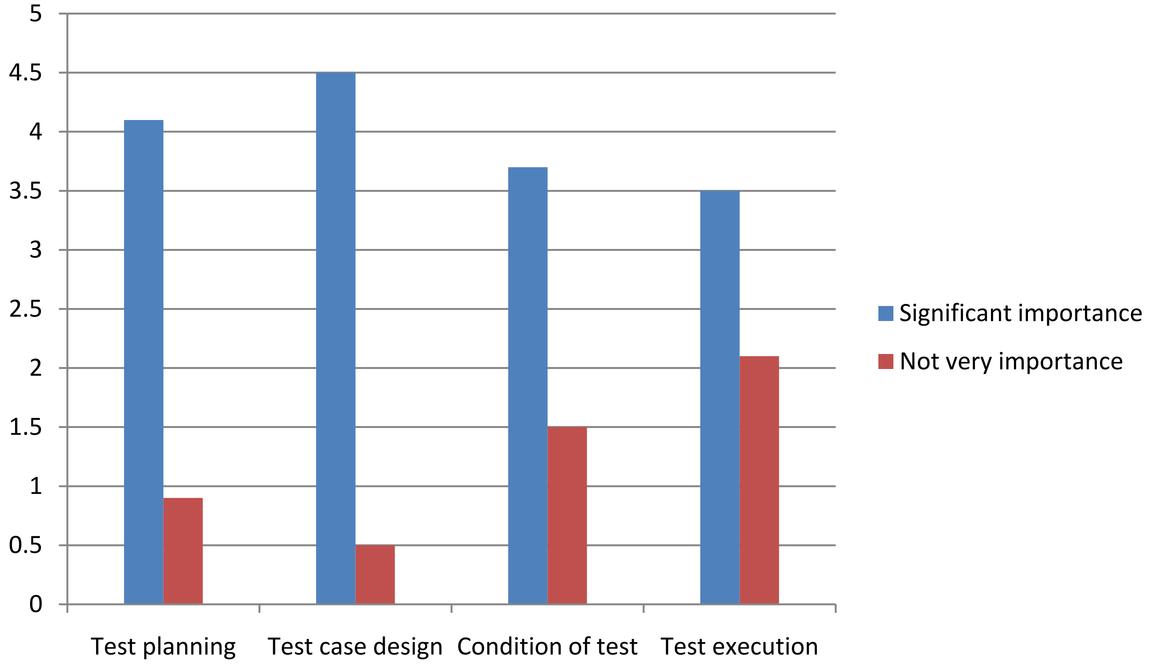
Table 1displaysconsequences of associating the standard measures. From table 2 can createnumerousbroad explanations.The amount, to which each measurement or the dimensions average associates with the standard scores, is indication of their standard validity

	Standard Item 1	Standard Item 2	Standard Average	Evidence Quality	System Quality	Contentment	Specific Impact	Association Influence
Internal Knowledge	0.44	0.41	0.45	0.44	0.43	0.41	0.29	0.42
External Knowledge	0.26	0.22	0.25	0.32	0.33	0.29	0.31	0.28
Knowledge Average	0.42	0.38	0.42	0.45	0.44	0.41	0.45	0.41

Table 1 Correlation of Knowledge Management

	#	**%**	**Cum: %**
Strategic User	124	39.1	39.1
Operational User	110	34.7	73.8
Process Owner	36	11.4	85.2
Technical Staff	47	14.8	100.0
Total	317	100.0	

Table 2 Respondent Composition of Knowledge Management



Graph 1 Importance of Knowledge management to Software testing activities

VII. CONCLUSION

Exploration of knowledge management in software analysis is samesignificant to development the testing level and sprainingvolume, improve the quality of software crops and the financialbenefit, and stimulate the nuclear sensible power of software enterprise. Software analysis technologies, knowledge management and informationexecutive model are conversed in this exertion. On the basis of systemically instant of these fields, we evaluated the knowledge service model, and then constructed informationexecutive model oriented software testing. At last this system is intended and executed which is the basis of additional investigations of informationexecutive. In this system concerned with software testing process is executed based on evaluating current difficultiesconnected with knowledge management in software testing. It is actualsignificant to transmit out informationexecutive in software analysis, and it is also respected in other fields. This system is a model of informationexecutive system oriented software testing process.

REFERENCE

[1] I.Rus and M. Lindvall, "Knowledge management in software engineering,"IEEE software, vol. 19, no. 3, pp. 26–35, 2002.

[2] S.Vasanthapriyan, J. Tian, and J. Xiang, "A survey on knowledgemanagement in software engineering," in Software Quality, Reliabilityand Security-Companion (QRS-C), 2015 IEEE International Conferenceon. IEEE, 2015, pp. 237–244.

[3] N.Suresh, "An overview of object oriented software testability" volume 1 Issue 1–Feb 2014.

[4] T.R. Gruber, "Toward principles for the design of ontologies used forknowledge sharing?" nternational journal of human-computer studies,vol. 43, no. 5-6, pp. 907–928, 1995.

[5] S.Kokila1 , T. Princess Raichel, "Software as a Service, a Detailed Study on Challenges and Security Threats" volume 2 issue 12 December 2015

[6] S.S. E. Committee et al., "IEEE standard for software and systemtest documentation," Fredericksburg, VA, USA: IEEE Computer Society,

[7] 2008.

[8] D.Graham, E. Van Veenendaal, and I. Evans, Foundations of softwaretesting: ISTQB certification. Cengage Learning EMEA, 2008.

[9] O.K. Wei and T. M. Ying, "Knowledge management approach in mobilesoftware system testing," in Industrial Engineering and EngineeringManagement, 2007 IEEE International Conference on. IEEE, 2007,pp. 2120–2123.

[10] X.Li and W. Zhang, "Metaphysical-based testing platform for reusing," inInternet Computing for Science and Engineering (ICICSE), 2012 SixthInternational Conference on. IEEE, 2012, pp. 86–89.

[11] I.Douglas, "Testing object management (TOM): A prototype for usabilityknowledge management in global software," in InternationalConference on Usability and Internationalization. Springer, 2007, pp.297–305.

[12] V.Santos, A. Goldman, and C. R. De Souza, "Fostering effective intertermknowledge sharing in Agile software development," EmpiricalSoftware Engineering, vol. 20, no. 4, pp. 1006–1051, 2015.

[13] I.Horrocks, "Ontologies and the semantic web," Communications of theACM, vol. 51, no. 12, pp. 58–67, 2008.

[14] Arik Johnson. An introduction to knowledgemanagement as a framework for competitive intelligence.[White Paper]. International Knowledge ManagementExecutive Summit, California,1998

[15] Gerhard Fischer, Jonathan Ostwald.Knowledge Management: Problems, Promises, Realities, andChallenges. IEEE Intelligent Systems, January/ February,2001,pp. 60-72

[16] IoanaRus, Mikael Lindvall, and SachinSuman Sinha, Knowledge Management in SoftwareEngineering A State-of-the-Art-Report[M], The University ofMaryland, 2001.

[17] He Zhitao. Research and Implementation ofKnowledge Management System Oriented Software TestingProcess(D). Beijing□Beijing University of Aeronautics andAstronautics, 2003.

[18] R.Brent Gallupe, “Knowledge ManagementSystems: Surveying the Landscape”[M], Queen’s University atKingston, October 2000.

[19] George Lawton. Knowledge Management:Ready for Prime Time?[J] IEEE Computer,vol 34, 2001, pp.12-14

[20] loanaRus, Mikael Lindvall. KnowledgeManagement in Software Engineering. IEEE Software,May/June 2002, pp. 26-38

[21] Nakkiran N Sunassee, David A Sewry. Atheoretical framework for knowledge managementimplementation. In: Proc of the SAICSIT 2002 ACM

[22] International Conference Proceeding Series, 2002, pp. 235-245

[23] Seija Komi-Sirvio, AnnukkaMantyniemi,VeikkoSeppanen. Toward a Practical Solution for CapturingKnowledge for Software Projects. IEEE Software, May/June,2002, pp. 60-62