Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 56 (2013) 13505-13509

ISSN: 2229-712X

Ontological Reliability Quantification Method

Shilpa Sharma[1] and Maya Ingle[2]

[1]Medi-Caps Institute of Technology and Management.

[2]Devi Ahilya University.

ARTICLE INFO

Article history:

Received: 31 January 2013;

Received in revised form:

11 March 2013;

Accepted: 15 March 2013;

Keywords

ORQM,

Project Category,

Project Parameters,

Reliability.

ABSTRACT

Software reliability quantification plays a very significant role for software consistency and excellence. However, the conventional software quantification method mostly focuses on evaluation by use of failure data which is gained only after testing or usage in the late phase of the software life cycle. Therefore, to obtain and quantify the software reliability with the help of architecture style may be introduced. Ontology allows developers and users to better understand software architecture and reliability terminologies, assess software reliability, and communicate effectively with the software reliability engineers. Therefore, an Ontological Reliability Quantification Method (ORQM) is instigated in this paper, which focuses on various project categories correlative with architecture style and concerned project parameters. Finally, some case studies are presented to demonstrate the viability of this method.

Introduction

Software reliability evaluation is playing an important role in software reliability engineering, which can give information taken as the reference or accordance to guide the software's design, analysis and testing and so on. Finally it will provide the quantitative estimation result for the issued software product. In recent years, software reliability evaluation based on failure data has been deeply developed, as the main means of software reliability estimation, lots of software reliability growth models have been proposed [1, 2, 3, 4, 5]. But with the shortcoming of not very good evaluation quality, many new models and technique were proposed to effectively improve the reliability estimation performance, such chaos deduce model, Bayes networks model, fuzzy theory model and so on [7,8, 9]. New technologies are also proposed, such as the failure data trend analysis and prediction quality improvement [10,11]. Based on the statistical theory, David et al. proposed several software reliability assessment methods which established the sampling theory for software reliability evaluation [12]. With the shortcoming of only applied in the late phase of the software life cycle, such as testing and maintenance process, its application is hindered. Whether it can be used in the early phase of software development becomes a difficult. Therefore, ontology is introduced to allow developers and users to better understand software architecture, assess software reliability, and communicate effectively with the architect. Ontology further allows the developer to make appropriate decisions in the context of architecture modelling, resource usage [13].

In this paper, we have introduced Ontological Reliability Quantification Method (*ORQM*) for various categories of projects such as communication, deployment, domain and structured oriented. The terminology used in our proposed method is described in Section II. In Section III, step wise procedure for reliability quantification using *ORQM* is discussed. We cover some case studies in Section IV. Finally, we present results and conclusion in Section V.

Terminology

Our Ontological Reliability Quantification Method (*ORQM*) use some standard terms along with some new terms needed for reliability quantification. We discuss these terms with the suitable examples in this section.

- **Project Category (PC)**

Project Category is defined on the basis of high level patterns and principles commonly used for application development. For example, these categories may include communication, deployment, domain and structured etc.

- **Project Parameters**

Individual project attributes that affect reliability quantification in a project are known as project parameters. we have classified project parameters affecting reliability in three classes mainly; quality attributes, devise ideologies and crosscutting concerns and described these classes as follows:

- ***Quality Attributes (Q)***

It is defined as the overall factors that affect run-time behaviour, system design and user experience. There exist various kinds of quality attributes depending upon various project categories. Q_i, i=1…, l; represent l kinds of quality attributes.

- ***Devise Ideology (D)***

It pertains to the key design principles using some specific criteria such as costs minimization and maintenance requirements. There may exist various kinds of devise ideologies. Let D_j, j=1…, m; represent m kinds of devise ideologies.

- ***Crosscutting Concerns (C)***

Crosscutting concerns are the features of a project that may apply across all layers, components, and tiers. These are also the areas in which high-impact design mistakes are most often made. Therefore, it represents key areas of design that are not related to a specific application. Let C_k, k=1…, n; represents n kinds of crosscutting concerns.

Tele:
E-mail addresses: Shilpa_1819@yahoo.com

- ***Weights (w)***

Weight is a value assigned to each project parameter depending upon its priority in specific type of project category. All the project parameters such as quality attributes, devise ideologies and crosscutting concern may have different priorities in quantifying reliability of a project.

- ***Effective Mean (EM)***

Average of weights assigned to quality attributes (*Q*), devise ideologies (*D*) and crosscutting concerns (*C*) of the project is said to be an effective mean.

- ***Deviation Factor (DF)***

It is defined as the variability for every class of parameter under consideration and is denoted by *DF*. For example, *DF(i)* for quality attributes, *DF(j)* for devise ideology and *DF(k)* for crosscutting concerns.

- ***Total Deviation Factor (TDF)***

It is the sum of deviation factors (*DFs'*) corresponding to every class of parameter.

- ***Project Reliability (R)***

It is ratio of total observed variability (*TDF*) captured across class of parameters to the total ideal variability of equivalent class of parameters (TDF_{ideal}).

Ontological Reliability Quantification Method (*ORQM*)

We propose Ontological Reliability Quantification Method (*ORQM*) that includes project parameters on the basis of project category for reliability quantification of project. We incorporate three classes of project parameters such as quality attributes; devise ideology and crosscutting concerns which are differ in numbers and weights as per the project category. The stepwise description of *ORQM* is as follows:

Step I: Identification of project category.

User must identify the PC of current project first.

Step II: Identification of project parameters and allocation of corresponding weights.

Identify project parameters depending on project category and assign weights.

Step III: Computation of Effective Means (*EMs*) of various parameters.

EMs corresponding to each class of project parameter related to each project i=1,2,3,…, N are shown in equations (6.1), (6.2) and (6.3) respectively.

$$EMQ(i) = \frac{1}{l}\sum_{\alpha=1}^{l} Q(i,\alpha) \quad \text{... (6.1)}$$

$$EMD(i) = \frac{1}{m}\sum_{\beta=1}^{m} D(i,\beta) \quad \text{… (6.2)}$$

$$EMC(i) = \frac{1}{n}\sum_{\gamma=1}^{n} C(i,\gamma) \quad \text{… (6.3)}$$

Step IV: Computation of Deviation Factors (*DFs*) and Total Deviation Factor (*TDF*) of various parameters.

DFs for each class of project parameters as well as *TDF* related to each project i=1,2,3,…, N are shown in equations (6.4), (6.5), (6.6) and (6.7) respectively.

$$DFQ(i) = \frac{1}{l}\sum_{\alpha}^{l}\left(Q(i,\alpha) - EMQ(i)\right)^2 \quad \text{… (6.4)}$$

$$DFD(i) = \frac{1}{m}\sum_{\beta}^{m}\left(D(i,\beta) - EMD(i)\right)^2 \quad \text{… (6.5)}$$

$$DFC(i) = \frac{1}{n}\sum_{\gamma}^{n}\left(C(i,\gamma) - EMC(i)\right)^2 \quad \text{… (6.6)}$$

$$TDF(i) = \sum_{i=1}^{N}\left(DFQ(i) + DFD(i) + DFC(i)\right) \quad \text{… (6.7)}$$

where N stands for total number of projects.

Step V: Calculation of Ideal Total Deviation Factor (TDF_{ideal}) of various project categories.

TDF_{ideal} refers to total deviation factor of an ideal project (i.e. the project possessing all the project parameters) and calculated using equations (6.1) to (6.7). TDF_{ideal} varies with the type of project category.

Step VI: Calculation of Project Reliability.

$R(i) = TDF(i) / TDF_{ideal}$

Case Studies

We consider different applications to analyze results obtained from *ORQM*. Our study included petite projects of four PC namely; communication, deployment, domain and structured oriented projects. We use three classes of project parameters namely; quality attributes, devise ideologies and crosscutting concerns corresponding to each project category.

Case I- Communication Oriented Projects

It is assumed that communication oriented PCs can accommodate many quality attributes *Qi* from q1 to q10, devise ideologies *Dj* from d1 to d5 and crosscutting concerns *Ck* from c1 to c6 as shown in Table I. Each project parameter is assigned some weight depending upon the frequency of its occurrence in maximum number of projects of that category. For example, domain alignment quality attribute is present in very few projects and therefore assigned 1. Whereas coupling quality attribute is present in every project under study and hence assigned 10. Now, we consider autonomous, distributable, loosely coupled share schema and contract, compatibility as devise ideologies allocated weight from 1 to 5 respectively. Then, crosscutting concerns such as instrumentation and logging, authentication, authorization, exception management, communication and caching are assigned weights from 1 to 6 respectively. We attempt to quantify reliability for communication-oriented projects P1 to P7 using *ORQM* as shown in Table II.

Case II- Deployment Oriented Projects

Let us consider now deployment oriented projects having illustrious combinations of *Qi* from q1 to q9, *Dj* from d1 to d3 and *Ck* from c1 to c6 as shown in Table III. Quality attributes such as maintainability, scalability, flexibility, availability, security, central access, supportability, usability and integrity as quality attributes allocated weights from 1 to 9 respectively; devise ideologies projects having separation of concerns, event based notification and delegated event handling with corresponding weights from 1 to 3; and crosscutting concerns projects possessing authentication, authorization, exception management, communication, cryptography and sensitive data acquired with weights from 1 to 6 respectively. *ORQM* is executed for deployment-oriented projects P8 to P14 for reliability quantification and depicted in Table IV.

Case III- Domain Oriented Projects

While domain oriented projects are studied, it is observed that the quality attributes such as *Qi* from q1 to q9, devise ideologies *Dj* from d1 to d5 and crosscutting concerns *Ck* from c1 to c5 as shown in Table V helps to quantify project reliability. Therefore, quality attributes weighted from 1 to 9, devise ideologies having weights from 1 to 5. Whereas, crosscutting concerns hold weights from 1 to 5 correspondingly. Next, P15 to P21 domain-oriented projects are considered for reliability quantification as shown in Table VI.

Case IV- Structured Oriented Projects

Lastly, we quantify reliability for structured oriented projects.

Table I Communication Oriented Projects

			P1	P2	P3	P4	P5	P6	P7
Q	q_1	Domain Alignment	1	1	0	1	0	1	0
	q_2	Abstraction	0	2	0	2	2	0	2
	q_3	Discoverability	0	3	3	3	0	3	3
	q_4	Interoperability	0	0	4	0	0	4	4
	q_5	Rationalization	5	5	5	5	5	5	0
	q_6	Extensibility	6	6	6	6	6	6	6
	q_7	Flexibility	0	0	7	7	0	7	7
	q_8	Scalability	8	8	0	0	0	8	0
	q_9	Simplicity	0	0	0	0	9	0	0
	q_0	Coupling	10	10	0	10	0	0	0
D	d_1	Autonomous	0	1	0	1	1	1	0
	d_2	Distributable	2	0	2	2	0	2	2
	d_3	Loosely coupled	3	3	3	3	3	3	3
	d_4	Share schema and contract	4	4	4	4	4	0	4
	d_5	Compatibility	5	0	5	0	0	0	0
C	c_1	Instrumentation and logging	0	1	0	1	1	1	0
	c_2	Authentication.	2	2	2	0	2	2	0
	c_3	Authorization	3	3	3	0	3	3	0
	c_4	Exceptn mgmt	4	0	4	4	4	0	4
	c_5	Communication	5	0	5	5	0	0	5
	c_6	Caching	0	0	6	6	0	0	0

Table II Reliability Computation of Communication Oriented Projects

Project	*Quality Attributes*										*Devise Ideologies*					*Crosscutting Concerns*						*TDF*	*R*
	q1	*q2*	*q3*	*q4*	*q5*	*q6*	*q7*	*q8*	*q9*	*q10*	*d1*	*d2*	*d3*	*d4*	*d5*	*c1*	*c2*	*c3*	*c4*	*c5*	*c6*		
P1	1	0	0	0	5	6	0	8	0	10	0	2	3	4	5	0	2	3	4	5	0	11.70	0.888383
P2	1	2	3	0	5	6	0	8	0	10	1	0	3	4	0	1	2	3	0	0	0	11.37	0.86329
P3	0	0	3	4	5	6	7	0	0	0	0	2	3	4	5	0	2	3	4	5	6	10.15	0.770691
P4	1	2	3	0	5	6	7	0	0	10	1	2	3	4	0	1	0	0	4	5	6	13.16	0.999349
P5	0	2	0	0	5	6	0	0	9	0	1	0	3	4	0	1	2	3	4	0	0	9.06	0.687927
P6	1	0	3	4	5	6	7	8	0	0	1	2	3	0	0	1	2	3	0	0	0	6.32	0.479589
P7	0	2	3	4	0	6	7	0	0	0	0	2	3	4	0	0	0	0	4	5	0	4.36	0.330802
P_{ideal}	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	1	2	3	4	5	6	13.17	1

Table III Deployment Oriented Projects

			P8	P9	P10	P11	P12	P13	P14
Q	q_1	Maintainability	1	1	1	1	1	1	1
	q_2	Scalability	2	0	2	0	2	0	0
	q_3	Flexibility	0	3	0	3	0	3	3
	q_4	Availability	4	4	0	4	0	4	4
	q_5	Security	5	5	5	5	5	0	5
	q_6	Central Access	6	0	6	0	6	6	6
	q_7	Supportability	7	7	0	7	7	0	0
	q_8	Usability	8	8	8	8	8	8	0
	q_9	Integrity	0	9	9	0	0	9	9
D	d_1	Separation of concerns	1	1	1	1	1	1	1
	d_2	Event based notification	2	2	2	2	2	0	2
	d_3	Delegated event handling	3	3	3	0	3	0	0
C	c_1	Authentication.	1	1	1	0	1	1	1
	c_2	Authorization	2	2	2	2	2	2	2
	c_3	Exceptn mgmt	0	3	3	0	3	3	0
	c_4	Communication	4	4	4	4	0	0	4
	c_5	Cryptography	5	0	0	5	0	0	0
	c_6	Sensitive data	6	0	0	0	0	0	6

Table IV Reliability Computation of Deployment Oriented Projects

Project	*Quality Attributes*									*Devise Ideologies*			*Crosscutting Concerns*						*TDF*	*R*
	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	d_1	d_2	d_3	c_1	c_2	c_3	c_4	c_5	c_6		
P8	1	2	0	4	5	6	7	8	0	1	2	3	1	2	0	4	5	6	7.78	0.821944
P9	1	0	3	4	5	0	7	8	9	1	2	3	1	2	3	4	0	0	7.10	0.749919
P10	1	2	0	0	5	6	0	8	9	1	2	3	1	2	3	4	0	0	7.50	0.792467
P11	1	0	3	4	5	0	7	8	0	1	2	0	0	2	0	4	5	0	6.77	0.714678
P12	1	2	0	0	5	6	7	8	0	1	2	3	1	2	3	0	0	0	7.37	0.778718
P13	1	0	3	4	0	6	0	8	9	1	0	0	1	2	3	0	0	0	5.59	0.590074
P14	1	0	3	4	5	6	0	0	9	1	2	0	1	2	0	4	0	6	9.03	0.953502
P_{ideal}	1	2	3	4	5	6	7	8	9	1	2	3	1	2	3	4	5	6	9.47	1.000235

Table V Domain Oriented Projects

			P15	*P16*	*P17*	*P18*	*P19*	*P20*	*P21*
Q	q_1	Communication	1	0	1	1	1	1	1
	q_2	Extensibility	2	0	2	2	2	2	0
	q_3	Testability	3	3	3	3	3	0	3
	q_4	Simplicity	4	4	0	4	0	4	0
	q_5	Highly cohesive	5	5	5	0	0	5	0
	q_6	Understanding	6	0	6	0	0	0	0
	q_7	Manageability	7	0	7	0	0	0	0
	q_8	Integrity	0	8	0	0	8	8	8
	q_9	Decoupling	0	9	0	0	9	0	0
D	d_1	Pensiveness	1	0	1	1	1	1	0
	d_2	Composition	2	2	2	2	2	2	2
	d_3	Legacy	3	0	0	0	0	0	3
	d_4	Encapsulation	4	4	0	4	0	0	4
	d_5	Binding	0	5	0	0	0	5	5
C	c_1	Cashing	1	1	1	1	0	1	0
	c_2	data validation	2	2	2	2	2	2	0
	c_3	Config. Mgmt	3	0	0	0	3	3	0
	c_4	Authorization	0	4	0	4	0	4	4
	c_5	Exceptn Mgmt	0	5	0	0	5	0	5

Table VI Reliability Computation of Domain Oriented Projects

Project	Quality Attributes									Devise Ideologies					Crosscutting Concerns					TDF	R
	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	d_1	d_2	d_3	d_4	d_5	c_1	c_2	c_3	c_4	c_5		
P15	1	2	3	4	5	6	7	0	0	1	2	3	4	0	1	2	3	0	0	6.69	0.631492
P16	0	0	3	4	5	0	0	8	9	0	2	0	4	5	1	2	0	4	5	6.76	0.637933
P17	1	2	3	0	5	6	7	0	0	1	2	0	0	0	1	2	0	0	0	4.38	0.413126
P18	1	2	3	4	0	0	0	0	0	1	2	0	4	0	1	2	0	4	0	3.44	0.324689
P19	1	2	3	0	0	0	0	8	9	1	2	0	0	0	0	2	3	0	5	6.52	0.615293
P20	1	2	0	4	5	0	0	8	0	1	2	0	0	5	1	2	3	4	0	7.83	0.739208
P21	1	0	3	0	0	0	0	8	0	0	2	3	4	5	0	0	0	4	5	4.78	0.451066
P_{ideal}	1	2	3	4	5	6	7	8	9	1	2	3	4	5	1	2	3	4	5	10.59	1

Table VII Structured Oriented Projects

			P22	*P23*	*P24*	*P25*	*P26*	*P27*	*P28*
Q	q_1	Abstraction	1	1	1	1	0	0	1
	q_2	Isolation	2	2	0	0	0	0	2
	q_3	Manageability	3	3	3	3	3	3	3
	q_4	Performance	4	4	4	0	4	4	4
	q_5	Reusability	5	0	0	5	5	5	0
	q_6	Testability	6	0	0	6	0	0	6
	q_7	Ease of Deployment	7	7	0	0	0	0	7
	q_8	Reduced outlay	0	8	8	0	8	8	8
	q_9	Ease of development	0	0	0	0	0	9	0
	q_0	Techcomplexity	0	0	10	10	10	0	0
D	d_1	Reusable	1	0	1	1	1	0	1
	d_2	Replaceable	2	2	2	0	2	0	2
	d_3	No context spec	0	3	3	3	3	3	3
	d_4	Independent	4	4	0	0	0	4	0
	d_5	High Cohesion	5	0	0	0	0	0	0
C	c_1	Authentication	1	1	1	0	1	0	0
	c_2	Audit &logging	2	2	2	2	2	0	2
	c_3	Communication	0	0	0	3	3	3	0
	c_4	Exceptn Mgmt	4	0	0	0	0	0	0
	c_5	Validation	5	5	0	0	0	5	0

Table VIII Reliability Computation of Structured Oriented Projects

Project	*Quality Attributes*										*Devise Ideologies*					*Crosscutting Concerns*					*TDF*	*R*
	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_0	d_1	d_2	d_3	d_4	d_5	c_1	c_2	c_3	c_4	c_5		
P22	1	2	3	4	5	6	7	0	0	0	1	2	0	4	5	1	2	0	4	5	9.00	0.734694
P23	1	2	3	4	0	0	7	8	0	0	0	2	3	4	0	1	2	0	0	5	10.03	0.818601
P24	1	0	3	4	0	0	0	8	0	10	1	2	3	0	0	1	2	0	0	0	5.92	0.482993
P25	1	0	3	0	5	6	0	0	0	10	1	0	3	0	0	0	2	3	0	0	6.25	0.510204
P26	0	0	3	4	5	0	0	8	0	10	1	2	3	0	0	1	2	3	0	0	8.13	0.663946
P27	0	0	3	4	5	0	0	8	9	0	0	0	3	4	0	0	0	3	0	5	6.61	0.539592
P28	1	2	3	4	0	6	7	8	0	0	1	2	3	0	0	0	2	0	0	0	6.63	0.540892
P_{ideal}	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	1	2	3	4	5	12.25	1

The required quality attributes, devise ideologies and crosscutting concerns are observed to allocate weights. Table VII illustrates Abstraction, isolation, manageability, performance, reusability and testability etc. and possesses weight in a order of 1 to 10. While devise ideologies consist of reusable, replaceable, not context specified etc. are having 1 to 5 weights correspondingly. In addition, authentication, audit and logging and communication hold 1 to 5 weights in that order and constitute crosscutting concerns. Finally, Table VIII shows *ORQM* execution for P22 to P28 structure-oriented projects.

Results And Conclusion

Reliability quantification of software projects is an exigent job due to its varying setting. Major challenge that lies in quantification is kind of project categories and affecting project parameters. With the help of *ORQM*, quantification may be accomplished on the basis of weight allocation to project parameters with corresponding project categories. These are useful in identifying the early scale of project reliability and establishing the software excellence. We also presented four cases concerned with project categories such as communication, deployment, domain and structure. Further, we computed certain statistics such as effective mean EM, deviation factor DF and total deviation factor TDF of these project cases. Further, we have computed software reliability R of every project of various project categories. And, we have observed some facts as follows:

- *ORQM* provides the facility to improve the traditional reliability che*Ck*ing mechanism by considering the architecture style thereby providing the scope of improvement in reliability estimation and the actual facts to user and developer.
- *ORQM* computes minimum reliability for communication oriented projects as 47% and maximum reliability as 99%. Thus, a direct measure may be provided for reliability quantification for any project.
- It is observed that, reliability of most of deployment oriented projects under study ranges from 71.4% to 79.2%. For these projects, quality attributes such as maintainability, security and usability as well as devise ideologies such as separation of concerns and event based notification and crosscutting concern mainly authorization plays vital role.
- *ORQM* quantifies maximum reliability of domain oriented project under study as 73% and minimum as 32%. It is found that, these values are typically less than corresponding maximum and minimum values of reliability for other projects of different categories.
- *ORQM* eliminates the need of failure data and experts. Therefore, an average project developer can quantify reliability more precisely.
- *ORQM* provides flexibility on number and type of vital project parameters and project category depending on the project behavior and team makeup.
- *ORQM* also resolves the limitations of reliability engineering by associating weights to each of the project parameter according to project category.
- Ontological approach for reliability quantification of software projects leads to a step towards the engineering practices thereby establishing the fact that these methods are not informal methods.

References

[1] R. Z. Xu, M. Xie and R. J. Zheng. Software Reliability Model and Application. Beijing: Tsinghua University Press,1994(in Chinese).

[2] S. Yamada Osaki. Software reliability growth modeling: Models and applications. IEEE Trans. SoftwareEngineering SE-11(12), 1431-1437, 1985.

[3] J. D. Musa, A. Iannino and K. Okumoto. Software Reliability: Measurement, Prediction, Application. Mc-Graw-Hill, New York, 1987.

[4] H. Pham. Software Reliability. Springer-Verlag, Singapore, 2000.

[5] S. Yamada. Software reliability models In Stochastic Models in Reliability and Maintenance. Springer-Verlag, Berlin, 2002, 253-280.

[6] N. Karunanithi, D.Whitley and Y. K. Malaiya. Prediction of Reliability Using Connectionist Models. IEEE Transactions on Software Engineering, 18(7), July 1992, 563-574.

[7] J. H. Guo,X.Z.Yang and H.W.Liu.Software Reliability Nonlinear Modeling and Its Fuzzy Evaluation.4th Wseas Int. Conf. on Non-linear analysis, non-linear systems and chaos, Sofia, Bulgaria, October 27-29, 2005,49-54.

[8] B. Littlewood. Forecasting Software Reliability. CSR Technical Report,1989.

[9] S. Bro*Ck*lehurst, P. Y Chan and B. Littlewood. Recalibrating Software Reliability Models. IEEE Trans.on Software Engineering, SE-16(4), 458-470, Apr.1990.

[10] L. P. David, A. John, S. P. Kwan. Evaluation of Safety-critical Software. Communication of ACM, 1990(6).

[11] W. H. Howden. Good enough versus High Assurance Software Testing and Analysis Methods. In Proc. of IEEE HASE, 1998.

[12] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," http://ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html, 2006.

[13] Shilpa Sharma, Maya Ingle, “Ontology Based Specifications for Software Reliability Advancement”, International Journal of Computer Applications, Volume 43– No.13, 2012 , pp18-26