

FLEXIBILITY: A KEY FACTOR TO TESTABILITY

Abdullah¹, Dr. M.H. Khan², Dr. Reena Srivastava³

¹Research Scholar, School of Computer Application, BBD University, Lucknow, India

²Associate Professor, Department of C.S. E., I.E.T., Lucknow, India

³Dean, School of Computer Application, BBD University, Lucknow, India

ABSTRACT

Testability is an important software quality factor that is ineffective if it is not available at an early stage in the development life-cycle. It becomes more essential in the case of object oriented design. Flexibility is an important key factor to testability analysis and measurement for delivering high class testable and maintainable software. Flexibility is a criterion of crucial significance to software developers, designers and the quality controllers. It constantly guides and supports to avoid wastage of resources as well as enable the designers for continuous improvement in the development process. Flexibility is concerned with building high quality and reliable software within the constraints of cost and time. It greatly influences cost, quality and reliability at software evolution process. Despite the fact flexibility is vital and highly significant aspect for software development processes, it is poorly managed. This paper focuses the need and importance of flexibility early at design phase. A model has been proposed for flexibility measurement of object oriented design by establishing multiple linear regression. Finally the proposed model has been validated using experimental tryout.

KEYWORDS

Flexibility Measurement, Testability Estimation, Software Testability, Software Quality, Design Phase, Object Oriented Design.

1. INTRODUCTION

Quality has become more important with our increasing dependence on software. In the last decades the demand for quality in software products has been increasingly emphasized. Computer industry has been delivering exponential improvement in cost, performance, but the problems with software are not declining. Software still delivered is late, goes beyond budget and is full of residual faults. As per the newest IBM report, 31% of the projects get cancelled prior to they are completed, 53% over-run their cost estimates by an average of 189%, and for each 100 projects, there are 94 restarts [1]. A key problem of software industry is its lack of ability to develop bug free software. If software developers are asked to officially state that the developed software is bug free, no software would have ever been released. Objective of software engineering is to create high quality software in time and within budget. If a product is meeting its requirements, we may say it is a superior quality product. The whole thing is measured with respect to requirements and if it matches, product is a quality product.

Software has become vital to advancement in almost all areas of human endeavor. The ability of programming only is no longer satisfactory to build large programs. There are major problems in the cost, timeliness, maintenance and quality of many software products. Software engineering has the goal of solving these problems by producing good quality, testable, maintainable software, on time, inside budget [10]. According to software engineering principles, if the process for development of any software is right, the chance of success of the software projects is greatly increased [22]. To attain this objective, study has to focus in a disciplined manner on both the quality of the product and on the process used to develop the product. However, due to increase in price of testing and *maintenance* of software, objective is now changing to deliver quality software. Software testing is an important and necessary activity of software development life cycle for producing high quality software. Testing is very important and challenging task. The time spent and effort required for software testing is very significant and consumes about 40% to 50% of the total cost for the entire development life cycle [3]. The most important problem during testing is that before correcting a program (error/fault), the programmer must first trace and understand it and it is possible with the help of its flexibility. For the same reason, traceability and understandability is the two most important criteria of quality factor flexibility [4].

It is important that cost effective testing technique must be applied during development life cycle and maintenance. The basic factor contributing to the development of these cost effective techniques is the testability of program. Program testability is defined as a measure of the effort required to satisfactorily test the program according to some well defined testing criteria [18]. To a large perspective, testing depends on how difficult the error is to trace. Program testability and error/fault traceability are two most important concepts: the more difficult an error is to trace, the more difficult it is in order to be fixed. The more difficult it is to correct, the higher its testability risk is [20]. The overall effort spent on testing not only depends on human factors, process issues, test techniques, and test tools, but also on characteristics of the software development artifacts. The degree to which a software artifact facilitates test tasks in a given test context is called testability [5]. If we want to improve testability we have to trace those parts of a program that lack testability. In view of the fact, it is clear that flexibility holds an important place as part of testability and traceability is an important criterion of flexibility [4]. The tester can use testability information to determine on what code to focus during testing [4]. Testability has been identified as one of the major issues in the field of software engineering for producing high quality software. It provides insights that are found to be very much valuable for the duration of software design, coding, testing and quality assurance [12].

2. TESTABILITY FACTOR

Testability is a major quality factor for producing high quality software. Lack of testability contributes to increased test and maintenance effort. The IEEE Standard Glossary defines testability as the degree to which a system or component makes possible the establishment of test criteria and performance of tests to conclude whether those criteria have been met [24]. ISO defines it in a parallel way: an attributes of software that bear on the effort needed to validate the software product [25]. The most well-known definition of testability is easiness of performing testing [13, 14]. The insight provided by software testability is significant for the extent of development life cycle, and quality promise. Design-for-testability is a very important issue in software engineering.

Testability is one of the most important factors determining the time and effort required to test software system [2]. A lower degree of testability outcome means increased test effort. It is essential in the case of OO designs where control flows are normally not hierarchical; it is costly

to redesign a system during implementation or maintenance. An effort has been put forth to recognize the key factors having positive impact on testability measurement at design phase. It has been concluded that Flexibility and Modifiability are the two most important factor affecting software testability measurement at design phase [8]. Taking into consideration the significance of their involvement, in this paper we have proposed a model to measure software flexibility at design phase.

3. FLEXIBILITY

In general, the objective of flexibility is to improve the quality of software systems within the development life cycle. Flexibility is a key concern to guarantee stability between software artifacts of succeeding phases of the development cycle. On the other hand only some works have addressed the theme of flexibility into its design. Flexibility is a testability factor, which allows the incorporation of changes in a design and gives an early focus to produce testable software within time and budget. The reported experience suggests that by emphasizing flexibility as a quality factor for testability measurement from the initial stage, the documentation will be clearer and extra reliable [16]. Flexibility of software artifacts has been recognized as an important key factor of testability for supporting various activities in the software system development process. More purposely, flexibility information can be used to support the analysis of implications and integration of changes that occurs in software systems. Flexibility enables system acceptance by allowing users to better understand the system and contributes to clear and consistent system documentation. Researchers and Practitioners advocated that flexibility aspect of software is highly desirable and significant for developing quality software. Literature survey reveals that there are various aspects of software, including flexibility factor that either directly or indirectly influence testability of software [4], [6], [7]. Flexibility plays a central role for producing quality software; it incorporates software quality objectives into the product and focuses on early error detection and design flaws. Aforementioned facts reveal that flexibility is a key factor to testability.

3.1. FLEXIBILITY QUALITY CRITERIA

Criteria are the characteristics which classify the quality factors. The criteria of the factors are the attributes of the software product or software production process by which the factor can be judged or characterized. The relationship between the flexibility and the quality criteria is listed below in Table 1.

Table 1: Flexibility Quality Criteria

S.No.	Factor	Quality Criteria	Mode
1	Flexibility	<ul style="list-style-type: none"> ▪ Structured ▪ Augment ability 	Criteria of Boehm quality Mode
2		<ul style="list-style-type: none"> ▪ Generality ▪ Independence ▪ Self- documentation ▪ Modularity ▪ Software independence 	Criteria of McCall quality Mode
3		<ul style="list-style-type: none"> ▪ Complexity ▪ Concision ▪ Consistency ▪ Generality ▪ Modularity ▪ Self-documentation ▪ Expandability ▪ Simplicity 	Criteria of Ming-Chang Lee Mode

There are following four major motivations for developing a list of criteria for flexibility:

1. Criteria provide a more absolute and real definition of factors.
2. Criteria common between factors help to show the interrelation among factors.
3. Criteria allow assessment and review metrics to be developed with greater easiness.
4. Criteria consent to identify that area of quality factors which may not be up to a predefined acceptable standard.

4. OBJECT ORIENTED DESIGN PROPERTIES

Object oriented design is the latest approach and provides a set of methods, tools and procedures for development of quality software. Object oriented design properties are real concepts that can be directly judged by scrutinizing the internal and external organization, association, and functionality of the design components, classes, attributes and methods. The feature of a design has a very important impact on the quality of a software product; but due to the multiplicity and complexity of design properties (e.g., coupling, encapsulation), their measurement and correlation with external quality attributes (e.g., testability, portability) is tough. Object-oriented design argues to continue essential software quality objectives like testability and reusability by mechanisms similar to encapsulation of data, inheritance and dynamic binding. Object-oriented programming is a fundamental technology that supports the above mentioned quality goals but only knowing the language policy or concepts of the object-oriented technology is not sufficient to produce good quality software [21]. A good quality object-oriented design wants design policy and practices that must be known and used. Their contravention will finally have a strong impact on the higher-level quality attributes. But as mentioned in advance, quality must be expressed in a quantified manner [15].

The design properties of abstraction, encapsulation, coupling, cohesion, complexity and design size are regularly used as being representative of design quality characteristics in cooperation with procedural development as well as object oriented development. Messaging, composition,

inheritance, polymorphism and class hierarchies correspond to new design concepts which have been introduced by the object oriented paradigm and are thus vital to the quality of an object oriented design [17]. Table 2 summarizes the various design properties and their benefits.

Table 2: Object Oriented Design Properties and Their Benefits

Design Property	Definition	Benefits
Design Size	A measure of the number of classes used in a design.	<ul style="list-style-type: none"> ▪ Representative of design quality characteristics of object oriented development
Abstraction	A measure of the generalization, specialization view of the design.	<ul style="list-style-type: none"> ▪ Flexibility ▪ Effectiveness ▪ Functionality
Encapsulation	Hides the implementation details.	<ul style="list-style-type: none"> ▪ Reduces complexity ▪ Flexibility ▪ Understandability ▪ Easier testing and maintenance
Coupling	Defines the interdependency of an object on other objects in a design.	<ul style="list-style-type: none"> ▪ Low Coupling Provides ▪ Flexibility ▪ Good understandability
Cohesion	Assesses the relatedness of methods and attributes in a class.	<ul style="list-style-type: none"> ▪ Flexibility
Inheritance	Allows child classes inherit the characteristics of existing parent class.	<ul style="list-style-type: none"> ▪ Flexibility ▪ Reusability ▪ Eliminates redundant code
Polymorphism	Ability to take more than one form. (Extensively used in implementing inheritance)	<ul style="list-style-type: none"> ▪ Provide abstraction ▪ Extensibility ▪ Reusability ▪ Eliminates redundant code
Messaging	A count of the number of public methods that is available as services to other classes. This is a measure of the services that a class provides.	<ul style="list-style-type: none"> ▪ Functionality ▪ Testability ▪ Effectiveness ▪ Reusability

5. OBJECT ORIENTED METRICS

Object orientation has several advantages such as lower development effort and lower development time and better testability, flexibility and maintainability etc. It is based on objects that constitute a system. It is more close to the real world. Object oriented approach supported some very useful design principles like maintainability, testability, reusability and quality. Each of the design properties identified in this paper signify an attribute or characteristics of a design that is satisfactorily well defined to be objectively assessed by using one or more well defined design metrics during the design stage. A review of existing design metrics [19] revealed that there are numerous metrics that can be modified and used in the estimation of some design properties, such as abstraction, inheritance and messaging. After a thorough review of some of the existing object-oriented metrics those might be applied in the design phase of object-oriented

software systems, a set of metrics listed in [17] has been elected for measuring flexibility at design phase.

6. RELATIONSHIP OF FLEXIBILITY WITH DESIGN PROPERTIES

Many experts tried to incorporate a variety of views as to how design properties may influence design quality attributes and establish a strong relationship between object oriented design properties and quality attribute flexibility. An extensive review of object oriented design and development was conducted in order to develop a basis for relating design properties to quality attribute flexibility [4]. It was observed that each of these properties have positive or negative impact on the factors that influence testability of object oriented design. After an in-depth evaluation of available literature on the topic [5] [9] [10] [11] [12] [15] [17], we established a correlation among object oriented design properties and flexibility as shown in Figure 1.

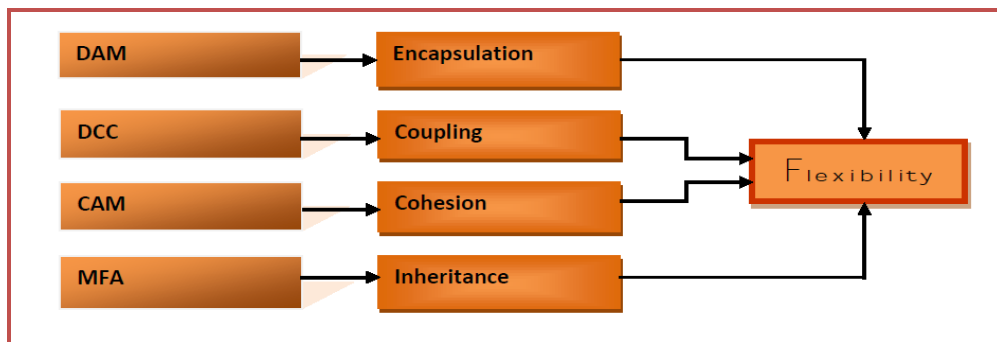


Figure 1: Correlation among Object Oriented Design Properties and Flexibility

Table 3: Metrics Description

Metric	Description
DAM	Data Access Metrics
DCC	Direct Class Coupling
CAM	Cohesion Among Methods of Class
MFA	Measure of Functional Abstraction

7. FLEXIBILITY MEASUREMENT MODEL

As discussed in the previous section, we have established the correlation among flexibility, object oriented design properties and object oriented metrics as depicted in Figure 1. With this correlation, this paper proposes a model for flexibility measurement. The proposed model has been developed using multiple linear regression procedure. This regression procedure has been widely accepted [8, 10, 26]. With the established correlation and linear regression we can say that

$$\text{Flexibility} = \beta + A1 \times \text{Encapsulation} + A2 \times \text{Coupling} + A3 \times \text{Cohesion} + A4 \times \text{Inheritance} \quad \text{Eq. (1)}$$

We used SPSS to calculate the coefficients and the final flexibility model that we arrived at is

$$\text{Flexibility} = 1.051 + 2.320 \times \text{Encapsulation} + 0.160 \times \text{Coupling} - 2.283 \times \text{Cohesion} + 11.572 \times \text{Inheritance} \quad \text{Eq. (2)}$$

Table 4: Coefficients

Coefficients^a						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1.051	10.446		0.101	0.926
	Encapsulation	2.320	10.433	0.275	0.222	0.838
	Coupling	0.160	0.752	0.147	0.212	0.845
	Cohesion	-2.283	3.298	-0.846	-0.692	0.539
	Inheritance	11.572	5.869	0.810	1.972	0.143

a. Dependent Variable: Flexibility

The Coefficients part of the output gives us the values that we need in order to write the regression equation (1). The Standardized Beta Coefficients give a measure of the contribution of each variable to the flexibility model. A big value designates that a unit change in this predictor variable has a large effect on the criterion variable. The t and Sig (p) values give a rough indication of the impact of each predictor variable – a big absolute T value and small p value suggests that a predictor variable is having a large impact on the criterion variable. The experimental evaluation of flexibility is very encouraging to obtain testability index of software design for low cost testing and maintenance.

The data used for developing flexibility model is taken from [21] that have been collected through large commercial object oriented systems. The index value of the expert's rating of flexibility for these systems is known and is termed as 'Known Value' in this research paper. For experimental validation this 'Known Value' will be compared with the calculated value; values will be calculated with the help of the proposed model.

The descriptive statistics of the output gives the mean, standard deviation, and observation count (N) for each of the dependent and independent variables and is shown in Table 5.

Table 5: Descriptive Statistics

	Mean	Std. Deviation	N
Flexibility	7.1938	2.88881	8
Encapsulation	0.9013	0.34203	8
Coupling	3.0775	2.65602	8
Cohesion	0.7663	1.07103	8
Inheritance	0.4588	0.20223	8

The Model Summary table of the output is most useful when performing multiple regression. Capital R is the multiple correlation coefficients that tell us how strongly the multiple independent variables are related to the dependent variable. R square is very supportive as it gives us the coefficient of determination. The Model Summary is shown in Table 6.

Table 6: Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.903 ^a	0.816	0.571	1.89306
a. Predictors: (Constant), Inheritance, Cohesion, Coupling, Encapsulation				

8. Empirical Validation

The empirical validation is an essential stage of planned research. Empirical validation is the correct approach and practice to justify the model acceptance. Keeping view of this fact, practical validation of the Flexibility measurement model has been performed using sample tryouts. In order to validate proposed flexibility measurement model the data has been taken from [21].

During experiments, flexibility value of the projects has been calculated using the developed model, followed by the calculation of flexibility rating. These calculated ratings are then compared with the known rating given by experts with the help of Charles Speraman's Coefficient of Correlation.

Table 7: Known and Calculated Flexibility Rating

Projects	Known Flexibility Value	Known Flexibility Rating	Calculated Flexibility Value	Calculated Flexibility Rating
P1	8.22	11	8.36	11
P2	12.64	14	9.29	14
P3	12.94	15	9.13	13
P4	0.84	1	7.09	8
P5	1.5	2	6.53	7
P6	3.0	3	6.19	5
P7	7.2	9	4.79	3
P8	6.2	6	5.83	4
P9	4.2	4	10.79	15
P10	4.7	5	7.24	9
P11	8.2	10	8.67	12
P12	9.5	13	7.83	10
P13	8.5	12	3.56	2
P14	7.2	8	6.43	6
P15	6.7	7	1.40	1

Table 8: Computed Ranking, Actual Ranking and their Relation

Projects → Flexibility Ranking ↓	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P1 0	P1 1	P1 2	P1 3	P1 4	P1 5
Computed Ranking	1 1	14	13	8	7	5	3	4	15	9	12	10	2	6	1
Known Ranking	1 1	14	15	1	2	3	9	6	4	5	10	13	12	8	7
$\sum d^2$	0	0	4	49	25	4	36	4	12 1	16	4	9	10 0	4	36
r_s	1	1	.9 9	.9 1	.9 6	.9 9	.9 4	.99	.7 8	.97	.99	.98	.82	.99	.94
$r_s > .6536$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

- $r_s > .6536$ means significant results.
- Flexibility measurement model had statistically significant correlations with 15 of 15 projects.

As mentioned above, Charles Speraman’s Coefficient of Correlation (rank relation) r_s was used to check the significance of correlation between calculated values of flexibility using model and it’s ‘Known Values’. Rank correlation is the process of determining the degree of correlation between two variables. The ‘ r_s ’ was calculated using the method given as under: Speraman’s Coefficient of Correlation

$$r_s = 1 - \frac{6\sum d^2}{n(n^2 - 1)} \quad -1.0 \leq r_s \leq +1.0$$

‘d’ = difference between ‘Calculated values’ and ‘Known values’ of Flexibility.
 n = number of projects (n=15) used in the experiment.

The correlation values between flexibility through model and known ranking are shown in table above. Pairs of these values with correlation values r_s above $[\pm.6536]$ are checked in Table 8. The correlations are up to standard with high degree of confidence, i.e. up to 99%. Therefore we can conclude without any loss of generality that flexibility measurement model measures are really reliable and significant.

9. Conclusion

Flexibility is one of the most significant factors for measuring testability of object oriented software design. Study developed a flexibility measurement model that establishes the correlation among flexibility, object oriented design properties and object oriented metrics. This paper shows the significance of flexibility as a key factor of testability and its relationship with various object oriented design properties. Flexibility measurement model in design phase has been developed and validated theoretically as well as empirically using experimental try-out. For experimental validation several large commercial projects has been used. The applied validation on the flexibility model concludes that developed model is highly reliable, up to standard and

significant. Finally paper concludes there is a high correlation between flexibility and testability, in addition statistical results shows that flexibility model is statistically very much significant and reliable.

REFERENCES

- [1] K.K. Aggarwal, Yogesh Singh. New Age International, Jan 1, 2005 - Software engineering.
- [2] Nazir, Mohd, and Raees A. Khan. "Testability Estimation Model (TEMOOD)." *Advances in Computer Science and Information Technology*. Computer Science and Information Technology. Springer Berlin Heidelberg, 2012. 178-187.
- [3] Jibitesh Mishra, Ashok Mohanty "Software Engineering. Page No.21, Pearson, 2012. www.pearsoned.co.in/jibiteshmishra
- [4] Nazir, Mohd, and Raees A. Khan. "Software Design Testability Factors: A New Perspective." *Proceedings, 3rd National Conference: INDIACom-2009, Bharti Vidya Peeth Institute of Computer Application and Management, New Delhi, Feb. 2009.*
- [5] Lee, Ming-Chang. "Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance." *British Journal of Applied Science & Technology* 4.21 (2014).
- [6] Poston, Robin, Jignya Patel, and Jasbir Dhaliwal. "A software testing assessment to manage project testability." (2012).
- [7] S. Mouchawrab, L. C. Briand and Y. Labiche, "A Measurement Framework for Object-Oriented Software Testability," *Information and Software Technology*, Vol. 47, No. 1, 2005, pp. 979-997
- [8] Jungmayr, Stefan. "Testability measurement and software dependencies." *Proceedings of the 12th International Workshop on Software Measurement*. 2002.
- [9] J Voas and Miller , "Improving the software development process using testability research", *Proceedings of the 3rd international symposium on software Reliability Engineering* , p. 114--121, October, 1992, RTP, NC, Publisher: IEEE Computer Society.
- [10] IEEE Press, "IEEE Standard Glossary of Software Engineering Technology," ANSI/IEEE Standard 610.12-1990, 1990.
- [11] ISO, "International standard ISO/IEC 9126. Information technology: Software product evaluation: Quality characteristics and guidelines for their use." 1991.
- [12] Jianping, Fu, Liu Bin, and Lu Minyan. "Present and future of software testability analysis." *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*. Vol. 15. 2010.
- [13] IEEE Standard Glossary of Software Engineering Terminology, ANSI/IEEE Standard 610.12- 1990, IEEE Press, New York, 1990.
- [14] Abdullah, Dr, Reena Srivastava, and M. H. Khan. "Testability Estimation of Object Oriented Design: A Revisit." *International Journal of Advanced Research in Computer and Communication Engineering* .Vol. 2, Issue 8, pages 3086-3090, August 2013
- [15] Abdullah, Dr, Reena Srivastava, and M. H. Khan."Modifiability: A Key Factor To Testability", *International Journal of Advanced Information Science and Technology*, Vol.26, No.26, Pages 62-71 June 2014.
- [16] Antoniol, Giulio, et al. "Design-code flexibility for object-oriented systems." *Annals of Software Engineering* 9.1-2 (2000): 35-58.
- [17] J. Gao, S. Ming-Chih, "A Component Testability Model for Verification and Measurement", In *Proceedings of the 29th Annual International Computer Software and Applications Conference*, pages 211–218. IEEE Computer Society (2005).
- [18] G. Jimenez, S. Taj, and J. Weaver, "Design for Testability" in the *Proceedings of the 9th Annual NCIA Conference*, 2005.
- [19] R. V. Binder, "Design for Testability in Object-Oriented Systems," *Communications of the ACM*, Vol. 37, No. 9, 1994, pp. 87-101
- [20] Marinescu, Radu, and Daniel Ratiu. "Quantifying the quality of object-oriented design: The factor-strategy model." *Reverse Engineering*, 2004. *Proceedings. 11th Working Conference on*. IEEE, 2004.
- [21] Bansiya, Jagdish, and Carl G. Davis. "A hierarchical model for object-oriented design quality assessment." *Software Engineering, IEEE Transactions on* 28.1 (2002): 4-17.

- [22] M. Bruntink and A. V. Deursen, "Predicting Class Testability Using Object-Oriented Metrics," Proceedings of IEEE International Workshop on Source Code Analysis and Manipulation, Chicago, 15-16 September 2004, pp. 136-145.
- [23] ISO 9001:2005, Quality management system Fundamentals and vocabulary; 2005.
- [24] Mulo, E. "Design for Testability in Software Systems." Master's Thesis (2007).
- [25] Gupta, S. P., and M. P. Gupta. Business statistics. Sultan Chand & Sons, 1983.
- [26] Abdullah, Dr, Reena Srivastava, and M. H. Khan. "Testability Measurement Framework: Design Phase Perspective.", International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 11, Pages 8573-8576 November 2014.

Authors

Abdullah received the MCA degree from Uttar Pradesh Technical University, Lucknow, in 2006. He is currently working as an Associate Professor, in the Department of Computer Application, at Institute of Environment and Management, Lucknow. His research interests include Software testability, Software Quality Estimation. He has written various books and study materials for (North Orissa University) Orissa, (Suresh Gyan Vihar University, Jaipur) Rajasthan, (Bharati Vidyapeeth University, Pune) Maharashtra, NAAC Re-Accredited "A" Grade University.



Dr. Reena Srivastava is currently working as Dean, School of Computer Applications at BBD University, Lucknow. She received her Ph.D. degree from MNNIT Allahabad, India. Her research area includes Multi-Relational Classification, Privacy Preserving Data Mining and Software Engineering.



Dr. M. H. Khan, Associate Professor, Department of Computer Science and Engineering at IET Lucknow UP. He obtained his MCA degree from Aligarh Muslim University (A Central University) in 1989. Later he did his PhD from Lucknow University. He has around 25 years rich teaching experience at UG and PG level. His area of research is Software Engineering. Dr. Khan published numerous articles, several papers in the National and International Journals and conference proceedings.

