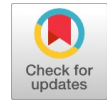


A Framework to Measure Level of Changeability & Reusability by using K means Classifier

Sonal Gahlot, Rajender Singh Chhillar



Abstract: *Component's reusability shows the level to which the component can be reused in alternate or as same component constructed applications. Software component's intricacy is measuring the internals of the modules to achieve the quality maintained using the modules. On the basis of metrics which contribute towards the reusability and complexity factors, the training samples are prepared with the help of metric plug in. This research work is centered around constructing a structure which calculates level of variability and reusability with K means Classifier, the planned method computes the reusability and intricacy of the components (samples) and every of the sample falls in a classification or a cluster like high to low reusability cluster. The proposed technique automatically calculates the reusability & complexity of the testing samples given as input to the classifier with the help of enhanced K-means classifier.*

Index Terms: *Component Based Software Engineering, Source Line of Coding, Lines of Coding, Rate of Component Customizability, Component Based Software Development*

I. INTRODUCTION

Creating financially savvy and quality items is a basic and testing bit of programming advancement. Segment based programming improvement can push engineers to effectively create programming inside the time and spending requirements. The hypothesis of (CBSE) segment put together programming building is based with respect to the improvement of free and inexactly coupled parts of the framework, by maintaining a strategic distance from disconnected reliance among framework segments. [1]. the segment based programming advancement is getting recognized in the business as another feasible programming improvement worldview. (CBSE) Component-based programming designing has been abridged by two improvement forms: the advancement of parts for use and the improvement of segment based programming frameworks (CBSS) with reuse by consolidating segments that have been sent independently [2]. In a profound sense, a product segment is depicted as a unit of synthesis. A client of a part can trade it autonomously as an item code without source codes. As the quantity of parts open available expands, it is ending up progressively fundamental to devise programming measurements to qualify the various attributes of segments.

Programming measurements are proposed to gauge programming quality attributes quantitatively. Among a few quality attributes, the reusability is basically basic when reusing segments. It is important to gauge the reusability of segments remembering the reuse of segments effectively [3].

1.1 Role of Component in Reusability

Segments are the accumulations of various prearranged tools which are utilized as the extra sheet which is to construct utilization of individual's implements.

1.1.1 Evolution of Components

In 1990's, customizing structures for e-business trades were worked of parts over different stages, programming language and system conventions. No instruments were available for perceiving "bugs" or to choose the efficiencies in the undertakings (question organized tasks). Fundamentally the missteps are printed using the yield explanation and the quantity of lines, number of bundles and most basic thoughts of article situated projects with the unequivocal ventures Those activities which are used to quantify the OOPS projects seem, by all accounts, to be authoritative and they are executed with the generous number of coding. In 1997, Intermetrics, a product designing association, proposed to build up an investigating device for Component-based programming structures. It just perceives the mix-ups and yields the blunder free program. In any case, finding the shortcoming won't give the treatment. As such the issue to find the adequacy of the program is settled with the assistance of the part transferred devices which gainfully gives the diagrams to the results and measures the program. The head favored point of view of segment based program is that the customer can pick their very own apparatus to survey their program as demonstrated by their necessities. In this, the customer can in like manner watch the associations from where the apparatuses are accessible, that is the customer can moreover download or see increasingly about the instrument from the immediate association.

1.1.2 Component Integration

Every one of the implements won't rally the client requisite standards. Some might be valuable for creating report and some might be utilized only to evaluate the program. Some device may assess the most seasoned technique for estimating the program with some confined parameters which are not in ebb and flow utilize and which might be proficient to gauge. All the recorded devices will quantify the normal parameter such SLOC (Source Line of Coding) and LOC (Lines of Coding). The require relies upon the developer or the client who apply the segment measurements. Coordinating the different devices has numerous focal points and furthermore a few drawbacks. Those are recorded.

Manuscript published on 30 September 2019.

*Correspondence Author(s)

Sonal Gahlot, Computer Science & Applications, Maharshi Dayanand University, Rohrak, India

Rajender Singh Chhillar, Computer Science & Applications, Maharshi Dayanand University, Rohrak, India

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Favorable circumstances on joining parts

1. Adaptable
2. Easy to use
3. Similarity
4. Comparability

Drawbacks on merging modules

1. Misunderstanding may occur nearly which tool to be used
2. Specific may not encounter the consumer essential
3. A limited tools are just examination versions
4. Every tool which is verified may have some shared criteria which are not supportive [4].

1.2 Reusability Computation Metrics In evaluating reusability creators depended on accessible surely understood metrics. Some researchers provide straight metrics for reusability which show measurable or semantic value showing level of reusability. Few researchers provide unintended metrics, which provide an incentive for specific further excellence trademark or estimated aspect, which further profoundly impacts reusability. Assessing reusability in such case involves understanding of given qualities.

1.2.1 Black box measurements

Discovery metric part general reusability (COR) consolidates measurements for flexibility, understandability and compactness. Understandability is estimated by (Existence of meta-data) EMI metric, flexibility by (Rate of Component Customizability) metric RCC and versatility by SCCr (Self-Completeness of Component's Return esteem).

1.2.2 White box/Glass box measurements

On the off chance that we are proprietors of segment, or we can accomplish part's source code some other way, we will have countless/glass box measurements for estimating reusability available to us. When managing reusability coupling, union and legacy are considered. Other measurements in view of fluffy approach are coupling, volume, unpredictability, consistency, reuse recurrence [5].

1.3 Benefits of Software Reusability the fundamental advantages of programming reusability are as per the following:

- Enhanced quality
- Limited expenses
- Expanded profitability
- Unwavering quality
- Interoperability

1.4 Different Types of Reuse

Ad-hoc Reuse: Ad-hoc reuse is favored when reuse happen inside tasks. Bidirectional trade is been done between the application gatherings.

Vault Based Reuse: Repository Based Reuse is done when segment archive is utilized and can be gotten to by assorted application gatherings. It depends on amount in light of the fact that any number of segments can be put into the capacity region and there is no influence over their quality and handiness. Here the storehouse is the trade medium between the applications bunches [6].

II. RELATED WORK

Adnan Khan et al. [2009] Component-based advancement allows us to create and coordinate item parts which encourage programming reusability, high caliber and speculation for analysis. CBSE creates utilization of methodologies which are relayed on design description languages, programming and object oriented design. These policies support in the improvement of together

domain-specific and nonspecific programming items. Reusability system accelerates programming advancement by utilizing beforehand created segments, in this way programming improvement price tag and interval is generously lessened. In this paper, creator exhibits a CBSE system for programming reusability.

Majdi Abdellatif et al. [2013] a (CBSS) is a product framework that is created by joining parts that have been conveyed freely. Over the most recent couple of years, numerous scientists have proposed measurements to assess CBSS qualities. However, the functional utilization of these measurements can be perplexing. For instance, a portion of the measurements have ideas that both cover or are not all around characterized, which could thwart their usage. The point of this investigation is to comprehend, arrange and inspect existing exploration in part based measurements, concentrating on methodologies and components that are utilized to gauge the nature of CBSS and its segments from a segment purchaser's perspective. This paper shows a composed mapping investigation of a few measurements that were proposed to gauge the nature of CBSS and its segments. Creator discovered 17 propositions that could be connected to appraise CBSSs, while 14 recommendations could be connected to assess singular segments in segregation. Different components of the product parts that were estimated are evaluated and talked about. Just a couple of the anticipated measurements are soundly characterized. The quality appraisal of the essential examinations identified numerous confinements and recommended rules for potential outcomes for enhancing and expanding the acknowledgment of measurements. Notwithstanding, it remains a test to portray and assess a CBSS and its segments quantitatively. Thus, much exertion must be improved to achieve an assessment approach later on.

Hironori Washizaki et al. [2003] in segment based programming advancement, it is important to gauge the reusability of segments keeping in mind the end goal to understand the reuse of parts productively. There are some item measurements for estimating the reusability of Object-Oriented programming. Be that as it may, in application growth with reuse, it is hard to utilize ordinary measurements in light of the fact that the source codes of segments can't be delivered, and these measurements require investigation of source codes. In this paper, creator propose a measurements suite for estimating the reusability of such discovery segments in view of constrained data that can be gotten from the outside of parts without any source codes. Creator characterizes five measurements for estimating a part's versatility, understandability and movability, with certainty interims that were set by factual investigation of various JavaBeans segments. In addition, creator gives reusability metric by consolidating these measurements in light of reusability demonstrate. Because of estimation tests, it is discovered that our measurements can successfully perceive discovery segments with high reusability.

P. Edith Linda et al. [2011] the principle point of this paper is to incorporate the distinctive question arranged metric devices and makes them available as a solitary extra.

The initial segment of this paper investigated five different instruments and they are moved into one to make utilization of those apparatuses in productive way.

Marko Mijac et al. [2015] Reusing programming resources have heaps of points of interest and has been fundamental component of all product advancement approaches. Segment based programming improvement has been primarily motivated by reuse. With a specific end goal to reuse programming segment, the part must be proposed and worked for reusability. Since reusability is impacted by various distinctive components, there are diverse methodologies and measurements used to quantify reusability. In this paper creator led broad writing audit with a specific end goal to perceive reusability measurements and variables affecting reusability. Whole of 39 papers presenting reusability measurements were begun and broke down. Creator recognized 36 distinct variables affecting reusability, in excess of 20 white box/glass box measurements and 12 discovery segment measurements.

Swati Thakral et al. [2014] the paper exhibits a writing audit of differing programming reusability ideas. It gives a brief and precise audit of accessible reusability metric anticipated by various scientists in various diaries and gatherings. The Objective is to assemble helpful data on programming segment reusability and the variables depend on which reusability of the part is exceptionally needy. As a result of writing survey we make that reusability is very subject to adaptability, interface difficulty, documentation quality and conveyability.

V. PrasannaVenkatesan et al. [2009] Software measurements can give a computerized approach to programming professionals to assess the nature of their product. The prior in the product improvement lifecycle this data is exhibited the more profitable it is, as changes are considerably more costly to make later on in the lifecycle. To the extent the Component-Based Software Engineering is based, the measurements can enable gauge, to design and recognize regions to create quality, diminish costs, upgrade venture administration and encourage chance administration. At last the achievements of the CBSE tasks can be assess from the measurements. In this paper creator characterize seventeen measurements for seven part characters. It comprises three utilitarian characters in particular the reasonableness, exactness and many-sided quality and four non-practical characters particularly the ease of use, viability, reusability and execution. The measurements are landed at, transferred on a metric model. The measurements are then tried with a contextual analysis.

William B. Frakes et al. [2005] This paper quickly outlines programming reuse investigate, talks about principle examine commitments and unsolved issues, gives pointers to fundamental productions, and presents four papers browsed The Eighth International Conference on Software Reuse (ICSR8).

Adnan Khan et al. [2014] Component-based improvement enables us to create and incorporate item parts which encourage programming reusability, lifted quality and speculation for testing. Segment Based Software Engineering makes utilization of methodologies which are transferred on design definition dialects, question situated propose and programming design. These systems bolster in the advancement of both space particular and nonspecific programming items. Reusability approach accelerates programming improvement by utilizing already created parts,

in this way programming advancement cost and time is considerably decreased. In this paper, creator introduces a Component-Based Software Engineering structure for programming reusability.

AnshulKalia et al. [2014] the reusable programming segments can be characterized in some ways. The reusable programming segments have an alternate usefulness that does not influence the usefulness of different parts. It has likewise been indicated definitely that for what the part reuse remains for and for what the segment reuse does not remains for. It is basic to portray the parts for enhanced reuse. Portrayal depicts the highlights and attributes of segments. Particular segments indicate unmistakable attributes in assorted areas of their utilization and in various working conditions. The parts can be grouped based on properties it have, that encourages with the better use, enhanced recovery, improved understanding and better listing. Through part order one gets the confirmation of choosing right segment and it proposes different manners by which a segment can be reused. The paper clarifies for the expect of portrayal which at last is reflected from the above expressed certainty. It likewise gives the criteria to describe the reusable segments. The standard is set down while remembering the general particulars and formal determinations. These stipulations in a single manner decide the practicality of a reusable part at the underlying level. The normal and formal determinations portray the interior and outside qualities which play a noteworthy undertaking in the distinguishing proof, choice, reception and execution of segments in specific application advancement. Details will tell the genuine idea of a reusable part those frameworks the required segments. It likewise examines the weight of portrayal that it puts on the reuse of reusable parts.

Marcus Kessel et al. [2015] Pragmatic programming reuse, in which available programming segments are intrusively adjusted for use in new tasks, includes three noteworthy exercises – choice, adjustment and reconciliation. The vast majority of the scholarly research into commonsense research to date has worried on the second of these exercises, adjustment, particularly the meaning of reuse plans and confirmation of obtrusive changes, despite the fact that the gathering movement is ostensibly the most critical and exertion concentrated of the three exercises. There is therefore a fabulous arrangement of extension for enhancing the level of help gave by programming web indexes and suggestion devices to sober minded re clients of programming parts. Test-driven web indexes are especially encouraging in such manner since they secure the natural capacity to "assess" parts from the viewpoint of clients' reuse situations. In this paper creator talk about a portion of the principle issues engaged with enhancing the determination hold up for sober minded reuse gave by test-driven web crawlers, depict some new measurements that can help address these issues.

Singh Ajmer et al. [2017] demonstrates that different Evolutionary Algorithms (EA) like Artificial Immune System (AIS), Particle Swarm Optimization (PSO), Simulated toughening, Artificial Bee Colony (ABC), Cuckoo Search Algorithm (CSA),

Ant settlement advancement (ACO) are being functionalized in the field of Software Engineering to acquire ideal arrangements.

Bhandari, Parul, and Ajmer Singh [2017] report an audit of coupling based experiment determination in model-based testing likewise the product under test this article situated one. We present an investigation of different procedures and instruments that specialists are utilizing in this field. Here we can utilize coupling estimation measurements to choose the experiments utilizing CKJM apparatus

Sharma, Sonia, and Ajmer Singh [2016] to cover the most extreme number of issues in similarly less time, it is basic to rank the experiments. To determine the time imperative experiment positioning glitches Ant Colony advancement (ACO) is an upgraded methodology. This paper exhibits a survey on experiment prioritization from a given test suite utilizing ACO.

Singh, Ajmer, Rajesh Bhatia, and Anita Singhrova [2018] present the examination of AI strategies in issue expectation

III. PROPOSED WORK

A huge number of segments are required to get the last total programming venture. The greater part of the segments of are created such that these segments are reusable and can be utilized as a part of any another undertaking if necessary and on the off chance that one segment isn't working appropriately it can be traded with other segment by connecting to and out with different segments, so support can be made simpler. There is a critical need to see how these product parts can be executed as fitting and play gadgets and reusability of programming segments can be comprehend in some unmistakable structure. consequently, in this examination work we are tackling this issue by building a structure which assesses level of reusability by utilizing grouping techniques (machine learning) , which would be most appropriate for our concern definition and better from past research works.

- Recognize parameters which affect the reusability of software components.
- By means of machine learning algorithm to improve programmed valuation of reusability

Selecting projects for developing representative datasets

For every software examination the elementary step is to generate the database. At this stage we will improve the representative datasets so that we could assess metrics and then unpredictability could be measured.

Downloading the projects from git repository

In this stage we are receiving open source GIT local repository and the developments after downloading getting warehoused in Eclipse.

Github open source project repository

Git supplies and thinks about information much differently than these different frameworks like the significant contrast amongst Git and some other repository Subversion and friends included is the way Git considers its information. Theoretically, most different frameworks store data as a as a list of document based changes. These frameworks CVS, Subversion, Perforce, Bazaar, et cetera think about the data they stay as an arrangement of documents and the changes made to each record after some time, as represented in Figure 1

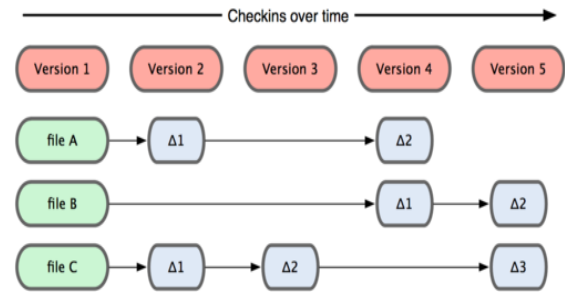


Figure 1: Checkins over time

Git doesn't take of or store its information thusly. Rather, Git contemplates about its information more like an established of previews of a small scale file system. Each interval you commit, or spare the state of your project in Git, it mainly takes a photo of what every one of your records look like right then and there and stores a situation to that snapshot. To be proficient, if archives have not changed, Git doesn't store the document once more—only a link to the past identical file it has just stored

These are the following total metrics we have selected
Metric Name Comparative Cost of Metric Inference for Reusability

LCOM	raises	raises
WMC	raises	drops
NOC	raises	raises
DIT	raises	raises
CPD	raises	drops

Metrics 1.3.6 plugin for NetBeans 8.0.2 is applied to gather values for the succeeding metrics: LCOM.WMC, NOC, DIT and CPD

LCOM Metric: If value of LCOM metric rises then the reusability of the module is high and module is less complex.

WMC Metric: If the value of WMC metric raises then reusability of segment drops.

NOC Metric: If the value of NOC metric raises then that module is strongly reusable

DIT Metric: If the value of DIT metric raises then reusability of component is high and also quality is high

CPD Metric: if the value of CPD metric raises then reusability of component drops

IV. RESULTS

Table 1: Dataset used for Training (Kmeans)

Cluster	LCOM	WMC	NOC	DIT	Reusability
1	0.303	34	1	2	Low Reusability
2	0.265	6.511	47	2.383	Highly Reusable
3	0.599	11.474	38	4.211	Highly Reusable & Less Complex Component
4	0	2.667	9	1.222	Low Reusable & Highly Complex Component
5	0.333	18	2	7	Highly Reusable & Highly Complex Component

Table 2: Test Samples

LCOM	WMC	NOC	DIT
0.467	13.455	22	3.864
0.14	8.703	12.333	2.324
0.482	13.75	20	2.85
0.41	5.143	14	2.714
0.353	10	4	1

Table 3: Results of Kmeans Classifier

Mem ber of Cluste r	LCO M	W MC	NO C	DIT	Reusability
3	0.467	13.455	22	3.864	Highly Reusable & Less Complex Component
4	0.14	8.703	12.333	2.324	Low Reusable & Highly Complex Component
4	0.482	13.75	20	2.85	Low Reusable & Highly Complex Component
4	0.41	5.143	14	2.714	Low Reusable & Highly Complex Component
4	0.353	10	4	1	Low Reusable & Highly Complex Component

At this step we downloaded 10 projects from internet which will help in developing the representative datasets so that we could evaluate metrics and their reusability could be measured. The following 10 projects are used i.e. first 5 for training and another 5 for testing. Using Eclipse, we run each of the below java projects and collect the various metric values like WMC, NOC, DIT LCOM etc. When we run the algo, we find the expected values called predicted values corresponding to our actual values for each observation of every project. K-means is unsupervised learning procedure to solve the bundling problem. The technique follows a very simple and basic style to characterize a prearranged informational group in a precise amount of bunches (assume k bunches). The fundamental thought is to distinguish k centroids, unique for every cluster. The better choice is to put the centroids as conceivable as much distant from every other. The next phase is to yield all point from a prearranged informational collection and connecting it to the nearby centroid. When every one of the focuses is viewed as, at that point the initial assembly is completed. Now k new centroids are re-computed of the groups formed by the past phase. When k new centroids are created another connection is established among related informational group attentions to the nearby fresh centroid. Now a circle has been shaped. Within circle the k centroids change their area in a disciplined way again n again until the point when no more changes are done. After the arrangement of over 8 group, we give the following 8 projects metric qualities to our classifier and after performing out the entirety of its count i.e. comparing at the metric values of each cluster with each other, it assigns out every one of the task to that cluster which have comparable characteristics as that of the project.

V. CONCLUSION

In this research, we have proposed the technique which automatically calculates the reusability & complexity of the components or projects with the help of enhanced K-means classifier. On the basis of metrics which contribute towards the reusability and complexity factors, the training samples form 5 clusters and when the testing samples go through or given as input to the classifier, they all lies in different cluster according to their matching features with their corresponding cluster as shown in table 4.3. In this paper, the planned method calculates the reusability and intricacy of the components (samples) and each of the samples falls in a category or a cluster like high to low reusability cluster but the models now taken in this research are small. So in future, this approach or technique can be enhanced or extended for better performance of evaluation for large databases or samples.

REFERENCES

- Adnan Khan, Khalid Khan, Muhammad Amir and M. N. A. Khan, "A Component-Based Framework for Software Reusability", International Journal of Software Engineering and Its Applications Vol. 8, No. 10 (2014), pp. 13-24.
- MajdiAbdellatif, Abu BakarMd Sultan, Abdul Azim Abdul Ghani1, Marzanah A. Jabar, "A mapping study to investigate component-based software system metrics", The Journal of Systems and Software, 2013, pp. 587-603.
- Hironori Washizaki, Hirokazu Yamamoto and Yoshiaki Fukazawa, "A Metrics Suite for Measuring Reusability of Software Components", Ninth International Software Metrics Symposium, IEEE, 2003, pp.1-13.
- P. Edith Linda, V. ManjuBashini, S. Gomathi, "Metrics for Component Based Measurement Tools", International Journal of Scientific & Engineering Research Volume 2, Issue 5, May-2011.
- Marko Mijac,ZlatkoStapic, "Reusability Metrics of Software Components: Survey", Research gate publications, September 23-25, 2015, pp.221-231
- Swati Thakral, ShraddhaSagar and Vinay, "Reusability in Component Based Software Development - A Review", World Applied Sciences Journal 31 (12): 2068-2072, 2014, pp.2068-2072.
- V. PrasannaVenkatesan, M. Krishnamoorthy, "A Metrics Suite for Measuring Software Components", Journal of Convergence Information Technology Volume 4, Number 2, June 2009, pp.1-16
- William B. Frakes and Kyo Kang, "Software Reuse Research: Status and Future", IEEE Transactions on Software Engineering, Vol. 31, No. 7, July 2005, pp.529-536
- Adnan Khan, Khalid Khan, Muhammad Amir and M. N. A. Khan, "A Component-Based Framework for Software Reusability", International Journal of Software Engineering and Its Applications Vol. 8, No. 10 (2014), pp. 13-24.
- AnshulKalia, SumeshSood, "Characterization of Reusable Software Components for Better Reuse", IJRET: International Journal of Research in Engineering and Technology, Volume: 03 Issue: 05, May-2014, pp.584-588.
- Marcus Kessel, Colin Atkinson, "Ranking Software Components for Pragmatic Reuse", 2015-16th International Workshop on Emerging Trends in Software Metrics, IEEE, pp.63-66
- A.Aloysius and K.Maheswaran, "A Review on Component Based Software Metrics", Intern. J. Fuzzy Mathematical Archive Vol. 7, No. 2, 2015, pp.185-194
- Singh, Ajmer. "Multi-objective test case minimization using evolutionary algorithms: A review." Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of. Vol. 1. IEEE, 2017
- Bhandari, Parul, and Ajmer Singh. "Review of object-oriented coupling based test case selection in model based testing." Intelligent Computing and Control Systems (ICICCS), 2017 International Conference on. IEEE, 2017

15. Sharma, Sonia, and Ajmer Singh. "Model-based test case prioritization using ACO: A review." *Parallel, Distributed and Grid Computing (PDGC)*, 2016 Fourth International Conference on. IEEE, 2016
16. Singh, Ajmer, Rajesh Bhatia, and Anita Singhrova. "Taxonomy of machine learning algorithms in software fault prediction using object oriented metrics." *Procedia Computer Science* 132 (2018): 993-1001

AUTHORS PROFILE



Sonal Gahlot joined Northcap University as Assistant Professor in July 2014 in Computer Science and Engineering Department. Currently working with DPG College of Engineering since april 2016. I have completed my B.E in Computer Science and Engineering from Gurgaon Institute of Technology and Management, Gurgaon in 2009, M.Tech in Computer Science and Technology from ITM University with First

Division in 2011, completed pre-Phd cousework from MDU, Rohtak with first division in August 2013 and persuing PhD from MDU, Rohtak. Areas of interest are Software Engineering, Component-Based Software Engineering, Data Structures, Computer Networks.



Rajender Singh Chhillar is a professor at the Department of Computer Science at Maharshi Dayanand University, Rohtak, Haryana, India. During his service in Maharshi Dayanand University, Rohtak. He served as the director at the University Institute of Engineering and Technology; a director at the Computer Centre. He served as the head at the Department of Computer Science, Chairman, a board of studies; a member, an executive and an academic

councils. His research interests include software engineering, software testing, software metrics, web metrics, bio metrics, data warehouse and data mining, computer networking, and software design. He has published more than 91 journal and 65 conference papers over the last several years and has also written two books in the fields of Software Engineering and Information Technology. He has visited many countries including France, Hong Kong, China, U.K, Dubai and Nepal. He also won the best paper award in International Conference ICCEE- 2013 held in Paris, France during October 12-13, 2013. Professor Chhillar is a director of board, CMAI Asia association, New Delhi and a senior member of IACSIT, Singapore and a member of Computer Society of India. Professor Chhillar has been serving as an editorial board member, guest editor and reviewer of multiple international journals, and serving as a program committee chair, keynote speaker and session chair of multiple international conferences. He also performs advisory work to Government agencies and academic bodies.