# INTEGRATING CMMI MATURITY LEVEL-3 IN TRADITIONAL SOFTWARE DEVELOPMENT PROCESS

Dr. Reena Dadhich[1] and Ujala Chauhan[2]

[1]Associate Professor, Govt. Engineering College, Ajmer (RTU, Kota)
reena.dadhich@gmail.com

[2]Govt. Engineering College, Ajmer (RTU, Kota)
ujala.chauhan@gmail.com

## ABSTRACT

*CMMI defines the practices that are specially implemented by software development businesses to achieve success. Practices includes topics that direct about eliciting and managing requirements, decision making, measuring performance, planning work, handling risks, and more. In this paper we will discuss Capability Maturity Model Integration (CMMI) software process improvement maturity model and the process areas at various levels of CMMI in brief. The main emphasis of the paper is to discuss about the Risk Management (RSKM) which is one of process area at CMMI level-3. The purpose of Risk Management (RSKM) processes is to identify potential problems before they occur so that risk-handling activities can be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives. The main aim of the paper is to analyse the effect of integrating the CMMI maturity level-3(process area -RSKM) with the traditional software development process. It represents an attempt to organize the sources of software development risk around the principal aspects of the software development cycle.*

## KEYWORDS

*CMMI, Software Process Model, Risk Management, Integrated model, Sources of risk*

## 1. INTRODUCTION

"Capability Maturity Model Integration (CMMI) is a process improvement maturity model for the development of products and services [1]". This model consists of transcending disciplines by offering the best practices through pointing out development and maintenance programmers covering the whole life cycle of the product from the very early phase (conceptualization) to the very end (delivery and maintenance)[6]. Therefore this system is recognized as a reference model that covers those development and maintenance activities. The model allows two representation-approach types of the so called Quality Management: continuous representation and staged representation. In few words, the continuous representation aims [5] the performance improvement of one organization process area in which it is expected the growth of diverse sectors, but in a way always lined up to the organization strategic objectives. The staged representation approach [4] focuses the process improvement in a systemic and structured way, aiming to reach a stage that allows the generation of a framework for the next stage.

The staged representation presents [4] five maturity levels [6]: **Initial level** (Maturity Level 1), **Managed level** (Maturity Level 2), **Defined level** (Maturity Level 3), **Quantitative Managed** level (Maturity Level 4), **Optimized level** (Maturity Level 5). Each of the Maturity level has its own various process areas as given in [6]. In section II of the paper we will discuss more about the process area of level-3 (i.e. Risk Management) and in section III we will integrate the Risk

DOI : 10.5121/ijsea.2012.3102

Management with SDLC. In section IV elaborate about the various sources of software development risks in SDLC.

## 2. Related Work

Since 1991, CMMs have been developed for variety of disciplines, some of the most popular are models for systems engineering, software engineering, software acquisition, workforce management and development, and process development (IPPD). The related work on the topic was appeared in [19, 20, and 21]. The detailed work on maturity models to measure the organizational maturity was published by CMU/SEI [5] in 2006.

## 3. Risk Management (RSKM)

Risk management is a continuous process for identifying potential/estimated problems before they occur so that risk handling activities can be planned and invokes as per need. Risk management should address issues that could endanger achievement of critical objectives. A continuous risk management approach effectively anticipates and mitigates risks that can have a critical impact on a project. Klein (1999) gives different types of risks [12] will affect budget, user satisfactions, and system performance. There are four major reasons for implementing the software risk management as given by Boehm [7]:

- Avoiding software project disasters, including run away budgets and schedules, defect-ridden software products, and operational failures.
- Avoiding rework caused by erroneous, missing, or ambiguous requirements, design or code, which typically consumes 40-50% of the total cost of software development.
- Avoiding overkill with detection and prevention techniques in areas of minimal or no risk.
- Stimulating a win-win software solution where the customer receives the product they need and the vendor makes the profits they expect.

Keshlaf and Hashim (2000) have developed models for tools [14] to aid the software risk management process. The process of RSKM [11] can be performed as follows in various steps (fig. 1) according to [26, 27, 28]:

### 3.1. Determine Risk Sources and Categories

Determining risk sources provides a basis for systematically examining the changing situations over time to uncover risks that can effect a project  or organization. There are many sources of risks, both internal and external to a project given in [26]. Risk sources identify the origin of risks. The sources of risks as discussed in [27] may include uncertain requirements, unprecedented efforts (i.e. estimates unavailable), infeasible design, competing quality attribute requirements that affect solution selection and design, unavailable technology, unrealistic schedule estimates or allocation, inadequate staffing and skills, cost or funding issues, uncertain or inadequate subcontractor capability, uncertain or inadequate supplier capability, inadequate communication with actual or potential customers or with their representatives, disruptions to the continuity of operations, regulatory constraints (e.g. security, safety, environment).

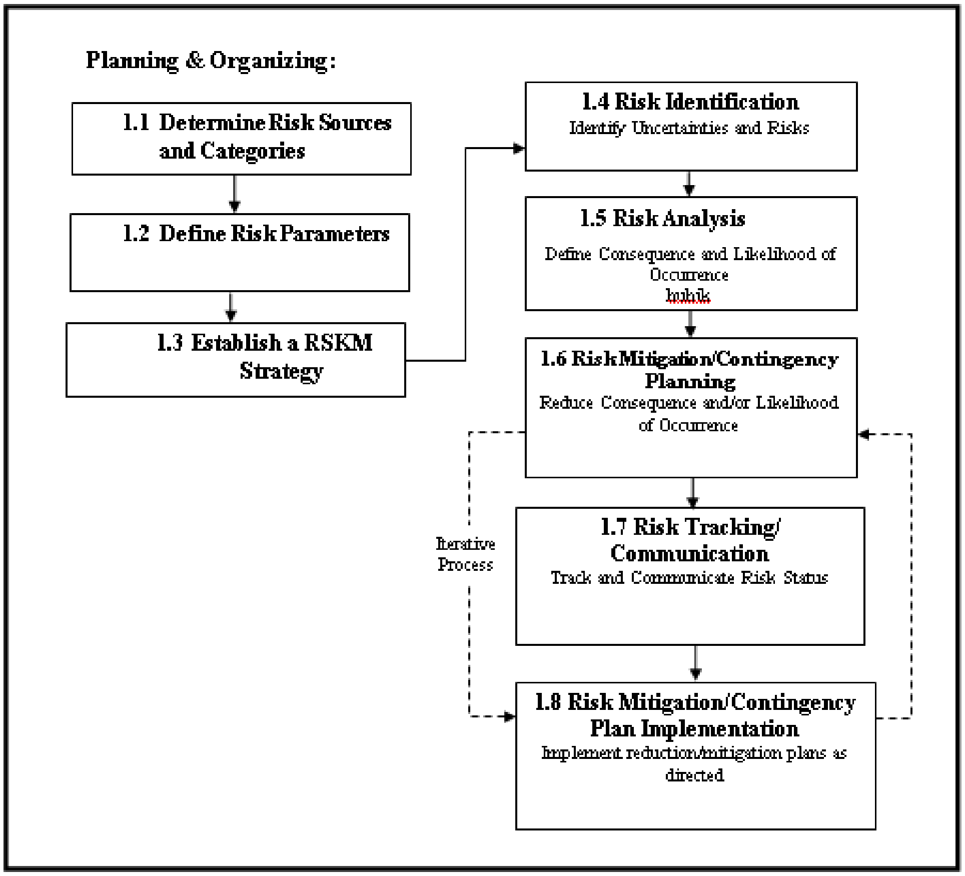
### 3.2. Define Risk Parameters

Risk parameters help in analyzing, categorizing and controlling the risk management effort. There are various parameters for evaluating, categorizing, and prioritizing risks like Risk likelihood (i.e., probability of risk occurrence), Risk consequence (i.e., impact and severity of risk occurrence), Thresholds to trigger management activities.

### 3.3. Establish a RSKM Strategy

After defining risk sources, categories and risk parameters, the risk management strategy should be defined which includes the sources, categories, and parameters from tasks 1 and 2; risk handling options (accept, avoid, share, mitigate) and mitigation techniques as discussed in [27]; thresholds and triggers; methods, measures, and tools. After defining the RSKM strategy, it is necessary to review it with relevant stakeholders to promote commitment and common understanding.

Fig.1 Risk Management context diagram


Planning & Organizing:
1.1 Determine Risk Sources and Categories
1.2 Define Risk Parameters
1.3 Establish a RSKM Strategy
1.4 Risk Identification
Identify Uncertainties and Risks
1.5 Risk Analysis
Define Consequence and Likelihood of Occurrence
luhik
1.6 Risk Mitigation/Contingency Planning
Reduce Consequence and/or Likelihood of Occurrence
Iterative Process
1.7 Risk Tracking/ Communication
Track and Communicate Risk Status
1.8 Risk Mitigation/Contingency Plan Implementation
Implement reduction/mitigation plans as directed

### 3.4. Risk Identification

The detailed risk identification process as discussed in [32] explains about identifying potential issues, hazards, threats, and vulnerabilities that could negatively affect work efforts or plans is the basis for sound and successful risk management. Risks should be identified and described understandably on time before they can be analyzed and managed properly. Risks are documented in a concise statement that includes the context, conditions, and consequences of risk occurrence described in [28]. Many methods are used for identifying risks. Typical identification methods include the following:

- Examine each element of the project work breakdown structure.
- Conduct a **risk assessment** using risk taxonomy.
- Interview subject matter experts.

- Review risk management efforts from similar products.
- Examine lessons learned documents or databases.
- Examine design specifications and agreement requirements

### 3.5. Risk Analysis

During risk analysis each identified risk is being evaluated and categorized using predefined risk categories and parameters, and determining its relative priority.

### 3.6. Mitigation/Contingency Planning

Risk mitigation planning is the activity that identifies, evaluates, and selects options to set risk at acceptable levels within project constraints and objectives. This can also include contingency plans to deal with the impact of selected risks that may occur despite of attempts to mitigate them, or avoidance plans to circumvent a risk before it can be realized as given in [27].

### 3.7. Risk Tracking/Communication

Risk tracking is the activity of systematically evaluating the status of risks. It feeds information back into the other risk activities of identification, analysis, handling (e.g., mitigation/contingency planning), and mitigation/contingency plan implementation, and also assists in tracking risk dependencies. Risks are updated in the Actions/Issue/Risk Log and reference (f), and are tracked to closure. The risk log shall include risk name and description, likelihood, consequence, priority, and mitigation/contingency plans, as well as any metrics defined for tracking the risk and risk dependencies. The risk tracking find out the answers of the question: “How are things going?” Another objective of risk tracking is to communicate risks and risk status to all affected stakeholders, including management, to establish a clear understanding and support for the project risk management strategy; to manage stakeholder expectations; and to effectively manage risks as discussed in [27].

### 3.8. Risk Mitigation/Contingency Plan Implementation

According to [27] the appropriate members of the project take the lead and implement mitigation/contingency plans as directed by the Project Manager. Selected risk-handling options are invoked, and a schedule is developed for each risk handling activity. Resources are committed to ensure risk mitigation activities those can be carried out successfully, and performance measures are collected based on the risk mitigation activities. The intent is to ensure successful risk mitigation took place and to answer the question: “How can the planned risk mitigation be implemented?” Contingency plans are implemented for selected critical risks when the predefined trigger is reached and the risk is realized.

## 4. RISK MANAGEMENT AND SDLC

Most of the good organizations are implementing RSKM process area for minimizing negative impact and to fulfil the need for sound basis in decision making. Effective risk management must be totally integrated into the SDLC [8]. Traditional SDLC have five phases: initiation, development or acquisition, implementation, operation or maintenance, and disposal. According to Micheal [29], IT system may occupy several of these phases at the same time. However, the risk management methodology is the same regardless of the SDLC phase for which the ssessment is being conducted. Risk management is an iterative process that can be performed during each major phase of the SDLC. Table 1 describes the characteristics of each SDLC phase and indicates how risk management can be performed in support of each phase.

Table-1. Integration of Risk Management into the SDLC

| SDLC Phases | Phase Characteristics | Support from Risk Management Activities |
|---|---|---|
| Phase 1—Initiation | The need for an IT system is expressed and the purpose and scope of the IT system is documented | Identified risks are used to support the development of the system requirements, including security requirements, and a security concept of operations (strategy) |
| Phase 2—Development | The IT system is designed, purchased, programmed, developed, or otherwise constructed | The risks identified during this phase can be used to support the security analyses of the IT system that may lead to architecture and design tradeoffs during system development |
| Phase 3—Implementation | The system security features should be configure, enabled, tested and verified. | According to Samuel [29] the risk management process supports the assessment of the system implementation against its requirements and within its modelled operational environment. Decisions regarding risks identified must be made prior to system operation |
| Phase 4—Operation or Maintenance | The system performs its functions. Typically the system is being modified on an ongoing basis through the addition of hardware and software and by changes to organizational processes, policies, and procedures | Risk management activities are performed for periodic system reauthorization (or reaccreditation) or whenever major changes are made to an IT system in its operational, production environment (e.g., new system interfaces) |
| Phase 5—Disposal | During the last phase the activities like disposition of information, hardware and software take place. These may include moving, archiving, discarding, or destroying information and sanitizing the hardware and software as appeared in [30] and explained by Mary Summer in [31]. | Risk management activities are performed for system components that will be disposed of or replaced to ensure that the hardware and software are properly disposed of, that residual data is appropriately handled, and that system migration is conducted in a secure and systematic manner |

## 5. THE SOURCES OF RISK IN SOFTWARE DEVELOPMENT LIFE CYCLE

Kendall et al. In [9] proposed the development cycle which encompasses the activities that are associated with the development of production-worthy code: requirements gathering, code design, the formulation of specifications, project planning, implementation, and testing. There are various sources responsible to incur risks at various phases of SDLC as listed in Table 2.

Table-2. The Sources of Software Development Risk in the Respect of Software Development Cycle

| Software Development Cycle | Software Development Risk |
|---|---|
| **Requirement** | Predictability<br>Evolvability<br>Completeness<br>Clarity<br>Accuracy<br>Precedence<br>Execution Performance Expectations<br>Proportionality |
| **Design** | Difficulty<br>Modularity<br>Usability<br>Maintainability<br>Portability<br>Reliability |
| **Implementation** | Specification<br>Project plan<br>Scale of effort |
| **Testing** | Verification<br>Unit testing<br>Integrated testing<br>Interoperability testing<br>Validation |

### 5.1. Requirements Risks

Risk attributes as appeared in [9] of the requirements risk element effects both the quality of the software requirements specification and also the difficulty of implementing software that satisfies the requirements.

A lack of ***predictability*** in requirements is often a consequence of the evolutionary nature of the requirements. As such there is an inherent unpredictability about the requirements that must be addressed in the budgets and schedules of the project.

The failure to recognize and adequately address the continuous evolution of requirements, that is ***evolvability*** is an especially important source of risk in long-lived scientific and engineering projects.

***Incomplete*** set of requirements fail to describe either the full intent or true intent (or both) of the customer. The principal consequence of this source of risk is that scope cannot be aligned with schedule and budget (resources).

***Clarity*** here is synonymous with understandability. Understandability is especially important when high-level goals are expressed by a customer who expects the developers to translate them into actionable requirements or a complete specification.

***Accuracy*** refers to the expectation that the aggregate requirements. If the requirements do not capture customer expectations, customer commitment to the project may be jeopardized.

Any software development project that posits capabilities that have not been demonstrated in existing software or that are beyond the experience of the project team or institution—that is, for which there is no ***precedent***—may be vulnerable to this source of risk.
If ***execution performance*** is a major driver of the code development project, then these expectations must be addressed in the requirements, design, specifications, and testing of the application.

***Proportionality*** refers to the possibility that the requirements may be disproportionate to the solution, that is, that the problem is over-specified. For example, too many and too specific nonessential requirements can preclude feasible solutions. This source of risk is not confined to technical requirements; they often enter through management mandates that impact the function of the development team.

## 5.2. Design Risks [9]

Design encompasses those steps through which requirements are translated into an actionable development plan. The existence of functional or performance requirements or expectations that are believed at the outset to be ***Difficult*** should be viewed as a potential source of risk.

***Modularity*** refers to the extent to which the code has been created using components or units that function independently from the other components. Software that has many direct interrelationships between different parts of the code is said to be less modular.

A lack of ***usability*** grids that is difficult to set up. Scientific code developers, while they may be experts in their scientific domains, may not be experts in usability, human factors, or even the use of their own codes by others. The failures may result from undefined or un-enforced standards, or from neglecting to analyze the system from the perspective of future ***maintainability***. The codes are used only once, or only with one type of computer, the majority of these codes outlast the generation of computers that they are first installed on, or are required to run on multiple hardware platforms from the beginning that is incurred ***portability***. ***Reliability*** refers to the ability of the software to be used in a production setting.

## 5.3. Implementation Risks

***Specifications*** are typically the output of the design step; they describe how the requirements are to be met in the code to be developed and drive the planning process. A ***project plan*** translates the specifications into a plan of action with a schedule, resources, and budget. In the Constructive Cost Model (COCOMO) estimation models, "***scale of effort***" is the most important factor contributing to a project's duration and cost.

### 5.4. Testing and Evaluation Risks

***Verification*** refers to ensuring that the code solves the equations of the model correctly. ***Validation*** refers to determining whether the mathematical model instantiated in the code faithfully mimics the intended physical behaviour.

## 6. CONCLUSIONS

As given in [29] risk management should be conducted and integrated in the traditional SDLC for IT systems, not because it is required by law or regulation, but because it is a good practice and it supports the organization's business objectives or mission. There should be a specific schedule for assessing and mitigating mission risks, but the periodically performed process should also be flexible enough to allow changes where warranted, such as major changes to the IT system and processing environment due to changes resulting from policies and new technologies. Risk management allows the IT managers to balance the operational and economic costs of protective measures and achieve gains in mission capability by protecting the IT systems and data that support their organizations missions. The objective of performing risk management is to enable the organization to accomplish its missions by better securing the IT systems that store, process, or transmit organizational information, by enabling management to make well-informed risk management decisions to justify the expenditures that are part of an IT budget, by assisting management in authorizing (or accrediting) the IT systems on the basis of the supporting documentation resulting from the performance of risk management. We can elaborate rest of the process areas at each level of CMMI and it fulfils the specific and generic goal of this specific area then we can achieve the higher level of CMMI.

## ACKNOWLEDGEMENTS

I Ujala Cahuahan would like to thanks to **Dr. Reena Dadhich**, Associate Professor and Head, Department of MCA, Govt. Engineering College Ajmer, who gave me the opportunity to work on this topic, and inspired me to carry forward this work as a challenge.

## REFERENCES

[1] CMMI Product Team, (2010) CMMI® for Services, Version 1.3 Software Engineering Institute, Carnegie Mellon University.

.

[2] CMMI Product Team (2010) CMMI® for Development, Version 1.3, Software Engineering Institute, Carnegie Mellon University.

[3] CMMI Product Team (2010) CMMI® for Acquisition, Version 1.3 Improving processes for acquiring better products and services, Software Engineering Process Management Program, Carnegie Mellon University.

[4] CMMI Product Development Team, (SEI 2002a) CMMI for Systems Engineering/Software Engineering/Integrate Product and Process Development/Supplier Sourcing, Version 1.1 Staged Representation Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

[5] CMMI Product Development Team, (SEI 2002b) CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1 Continuous Representation (CMU/SEI-2002-TR-011, ESCTR- 2002-011). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

[6] CMMI Product Development Team, (2006) CMMI® for Development, Version 1.2 Pittsburgh, A: Software Engineering Institute, Carnegie Mellon University.

[7] Barry W. Boehm, Tutorial, (1989), "Software Risk Management, Les Alamitos", CA, IEEE Computer Society.

[8] Gary Stoneburner, Alice Goguen, and Alexis Feringa, ( July 2002) " Risk Management Guide for Information Technology Systems" Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg.

[9] Richard P. Kendall ,Douglass , E. Post , Jeffrey C. Carver ,Dale B. Henderson , David A. Fisher, (2006), A Proposed Taxonomy for Software Development Risks for High-Performance Computing (HPC) Scientific/Engineering Applications, Software Engineering Institute, Carnegie Mellon University.

[10] Linda Westfall and The Westfall Team, (2001) "Software Risk Management", Copyright © 2001 The Westfall Team. All Rights Reserved.

[11] Ray C. Williams, (14 July 2006) , "The CMMI RSKM Process Area as a Risk Management Standard", Sixteenth Annual International Symposium of the International Council on Systems Engineering (INCOSE).

[12] Jiang, J.J., Klein, G, (1999), "Risks to different aspects of system success", Information and Management Vol.36, No.5, pp263–272.

[13] Y.H. Kwak, J. Stoddard , (2004) "Project risk management: lessons learned from software development environment", School of Business and Public Management, The George Washington University, Washington, DC 20052, USA Technovation Vol.24, pp915–920.

[14] Keshlaf, A.A., Hashim, K., (2000) "A model and prototype tool to manage software risks" Proceedings of First Asia–Pacific Conference on Quality Software,, pp. 297–305.

[15] Klein, S.A., (1998) "Putting methodology in perspective from a project risk viewpoint", IEEE Power Engineering Society 1999 Winter Meeting, Vol. 1, pp. 362–365.

[16] Kwak, Y.H., Ibbs, C.W., (2000) "Calculating project management's return on investment", Project Management Journal Vol.31, No.2, pp38–47.

[17] Yacoub, S.M., Ammar, H.H, (2002) "A methodology for architecture level reliability risk analysis", IEEE Transactions on Software Engineering Vol.28 , No.6, pp529–547.

[18] Jiang, J.J., Klein, G., Discenza, (2001)" Information system success as impacted by risks and development strategies" IEEE Transactions on Engineering Management Vol.48, No.1, pp46–55.

[19] Pittisburgh, (July 1997) Draft Version 0.98 "Integrated product development Capability Maturity model" , PA: Enterprise Process Improvement Collaboration and Software Engineering Institute, Carnegie Mellon University.

[20] Washington, DC, EIA 1998 "Systems Engineering Capability Model (EIA/IS-731)" , Electronic Industries Alliance.

[21] SEI 1997b October 22, 1997 Version 2.0 (Draft C), "Software CMM ", Software Engineering Institute.

[22] Habib, M., Ahmed, S., Rehmat, A., Khan, M. J., & Shamail, S. (2008), "Blending Six Sigma and CMMI - an approach to accelerate process improvement in SMEs." IEEE International Multitopic Conference (pp. 386-391).

[23] Guzmán, J., Mitre, H., Amescua, A., & Velasco, M. (2010), "Integration of strategi management, process improvement and quantitative measurement for managing the competitiveness of software engineering organizations", Software Quality Journal, 18, 341-359.

[24] Hwang, S., & Yeom, H. (2009). "Analysis of Relationship among ISO/IEC 15504, CMMI and K-model" 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing (pp. 306-309).

[25] Kelemen, Z. D., Kusters, R., Trienekens, J., & Balla, K. (2009) "A Process Based Unification of Process-Oriented Software Quality Approaches", Fourth IEEE International Conference on Global Software Engineering (pp. 285-288).

[26] http://hci-itil.com/CMMI/references/sp_project_5_1_1_determine_risk_sources.html.

[27] http://www.docstoc.com/docs/43885029/Risk_Management_Process.

[28] http://www.mitre.org/work/sepo/toolkits/risk/compliance/files/RiskProcessGuidelines

[29] http://www.slideshare.net/Micheal22/risk-management-guide-for-information-technology-systems

[30] http://www.projectperfect.com.au/white-paper-enterprise-risk-management.php

[31] Mary summer, 2000, Risk factors in enterprise-wide/ERP projects , Mary Sumner. Journal of Information Technology. London: Dec 2000. Vol. 15, Iss. 4; p. 317.

[32] http://www.scribd.com/doc/40315956/CMMI-for-Development-Version-1-3.

## Authors-

**Ms. Ujala Chauhan** has done her B.E(CS) in 2007 from Govt. Engg. College, Bikaner(Raj.) India. At present she is doing her M.Tech(CS) from Rajasthan Technical University Kota, India. She has 3 years of teaching experience. She has presented papers in conferences.

**Dr. Reena Dadhich** is presently working as a Associate Professor and Head of the Department of Master of Computer Applications at Engineering College Ajmer, India. She received her Ph.D. (Computer Sc.) and M.Sc. (Computer Sc.) degree from Banasthali University, India. Her research interests are Algorithm Analysis & Design , Wireless Ad-Hoc Networks and Software Testing. She is involved in teaching since more than 12 years and in research since last four years. She is working as an Editorial Board Member/Reviewer/Committee member of various International Journals and Conferences. She has written many research papers and authored 2 books.