



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

Software Development Risk Aspects and Success Frequency on Spiral and Agile Model

M. Soumya Krishnan

Assistant Professor, Department of Computer Science & IT, Amrita School of Arts and Sciences,
Amrita Vishwa Vidyapeetham, Kochi Campus, Kerala, India

ABSTRACT: Software companies face a lot of difficulties in choosing a correct development model when projects have to confront a lot of risk factors ranging from low to high. When software development becomes complex in nature due to risk, companies usual practice is to switch over to the only risk detection traditional model – the Spiral. Spiral model is considered to be the best conventional model for risk analysis. But in modern developmental style, all the development cycles around a single trend – the Agile. The main objective of this paper is to make a study on these two models and to get a conclusion on the impact of risk assessment and its success factors on complex software developmental projects.

KEYWORDS: Risk Factors, Risk Assessment, Spiral Model, Agile model, Software Development, SDLC.

I. INTRODUCTION

In this fast growing modern Information Technology world, software development has become a great challenge for the developers as well as the organization. The main challenge is related to various risk factors associated with the project. The identification, assessment, management and mitigation have really paved a challenging situation for the software companies. Most of the software development projects found to use more resources, more time and more budget than the specified and reach at a lesser quality and functionality product, giving rise to a dissatisfied customer.

On analysis, it is found that, all these problems occurred due to the lack of risk assessment during the inception of the project and the various activities during SDLC. Project risks are those that could prevent the achievement of the objectives stated by the Project manager and the Project team. Risks usually arise from the relationships among the technology and people. Inefficient assessment and poor management of risk may create a huge negative impact on the business growth of the company itself. Through this paper, I have tried to choke out some of the various generic risk factors and have studied on the impact range of risks on various software developmental issues by taking into account, the two main giant SDLC models – the traditional Spiral Model and the latest new generation lightweight model - the Agile.

II. RELATED WORK

Till today, much number of studies are found in the area of software development methodologies by comparing the different models as well as studies based on the risk factors and their effect on software developments in various perspectives. But certainly, these studies are not complete in the extent of finding the exact risk factors and how the models succeed from those effects. In [1], the author has indicated about the risk factors and its assessment techniques using various criteria. [2] emphasises on the concept of risk assessment and management experienced in Agile Methodologies. Here, it is clearly shown about the five different steps in easily managing those risks. This was the main source of my studies in this area. In [3], Boehm tries to explain the major steps and techniques involved in software risk management. [4] gave an insight about the software process models and their comparison based on many features. From this paper, we get a good idea about the overall nature of each software process model. [5] emphasises on the overall idea about agile methodology and how it helped IT managers to improve their productivity. In [6], the author helped in reviewing about the leading software process model and explained about the risk states and their management methods. Thus, they have tried to reveal the concept that some of the development methodologies



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

inherently possessed risk management capability and are inevitable in nature. [7] refers to the comparison between the five different software models by stating their advantages and disadvantages very clearly. And finally, in [8], the author has discussed about the linear sequential model and its risk factors throughout its development phases. All these together helped me to gain an insight about risk and their effect on various software models.

III. OVERVIEW OF RISK ASSESSMENT ON SDLC MODELS

Risk assessment and management has become a vital part in software development. There occurred a huge transformation from the traditional risk analysis model – the spiral to the latest trendy model – the Agile. This change has created new challenges in the field of risk analysis and their management.

In this paper, I have made a comparative study on different aspects of risks and the efficiency of the two models while dealing and assessing these risk factors. For this study, I have chosen the two famous models – Spiral and Agile. Let us first have an overlook on these models and their general characteristics and then apply them on the generic risk factors and find out the efficiency of these models to cope up with these risks during the normal software development life cycle.

SPIRAL MODEL

The Spiral Life Cycle Model is a type of iterative software development model which is generally implemented in high risk projects. It was first proposed by Barry Boehm in 1986 through his article – “*A Spiral Model of Software Development and Enhancement*” [3]. In this approach, we combine the characteristics of both, waterfall model and prototype model. In Spiral model we can arrange all the developmental activities in the form of a loop or spiral.

Each loop in a spiral indicates a development phase (and we can have any number of loops according to the size and nature of the project). Each loop has four quadrants: First quadrant is used to determine the objectives, alternatives and constraints. Here, we try to understand the product intentions, changes in design and constraints enforced due of cost, technology, and schedule. Second quadrant indicates the Risk analysis and evaluation of alternatives. Here we try to find which other methods can be implemented in order to fulfil the identified constraints. Here, operational and technical issues are given importance. Risk identification and mitigation is given primary focus in this phase. And based on these factors future action are determined. Third quadrant indicates the execution of that phase of development. Now, we develop the planned product. Testing is also given equal importance. In order to do software development, waterfall or incremental approach can be considered. Fourth quadrant helps planning the next phase. Here we evaluate the progress and take decisions by considering all constraints. During this phase, the developer has the privilege to decide whether to continue with the project or to stop working with the project. If solvable issues are identified, it could be resolved and necessary steps to proceed further can be planned. Consequent loops of spiral model involve similar phases. Here, analysis and engineering efforts are applied. Huge, expensive or complicated projects use this type of life cycle. At any point of time, if one feels that the risk involved in the project is large and unmanageable than expected, there is a provision to abort it. Assessments at different phases can be done by an internal person or by an external client.

Spiral – A Meta Model

Spiral model is also called as meta-model because in a way it combines the properties of other models of SDLC. Mostly waterfall and prototype models are taken into consideration. Here we do software development systematically over the loops (following the waterfall approach) and at the same time we make a prototype and show it to user after completion of various phases (as that of prototype model). In this way, we are able to assess and reduce risks as well as follow a systematic approach.

The risk driven approach of the spiral model helps the model to accommodate any type of specification oriented, model based or any other transformation oriented approach to the development of software. Similarly, risk management factors helps us to determine the amount of time and effort that is committed to other project activities like planning, change management, quality factors, formal technical reviews, testing etc.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

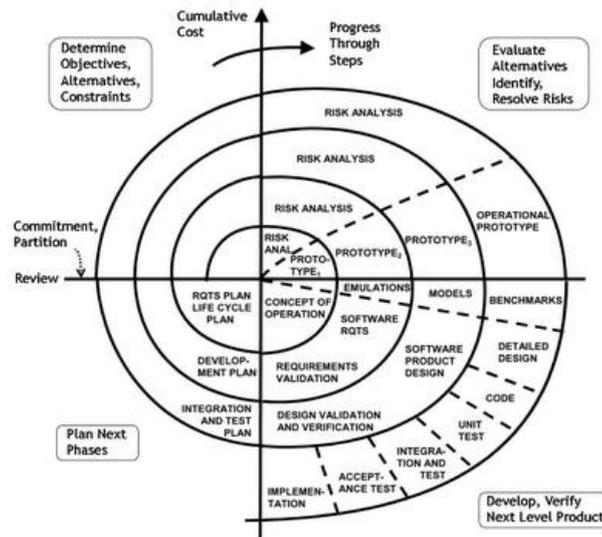


Figure 1 – Spiral Model

The highpoint of this model is that each of its cycle is completed through a fruitful interaction between the customers and the developers related with the project. Though this model has many positive point associated, as usual, the model has many negative side also. Let us summarize the merits and de-merits of this model as follows:-

Advantages

- Spiral Life Cycle Model is one among the most flexible SDLC models. Development phases can be determined by the project manager, according to the complexity, size and type of the project.
- Supervising the Project is very easy and active. Each phase, or each loop, requires a review from concerned authority. This makes the model more apparent.
- Risk management is one of the major characteristics of the model, which makes it more reliable compared to other models.
- Changes can be announced later in the life cycle.
- Project guesstimates in terms of schedule, cost, quality etc. becomes more realistic as the project moves forward and loops around the spiral and gets completed.
- It is most suitable for high risk projects, where business needs may be unstable and non-predictive in nature.
- A highly personalized product can be developed using this.

Disadvantages

- Cost involved in this model is usually unpredictable high.
- It is a complicated method especially for projects with a clear specification (SRS).
- Skills required, for evaluating and reviewing project from time to time, need high expertise.
- Procedures and protocols should be followed properly to effectively implement this model. By following such activity through-out the period of project is really a threatening aspect.
- Due to various customizations acceptable from the customer at any time, using the same prototype in other projects, will invariably affect in future
- It is not a suggested model for low risk projects.
- Meeting economic and planning requirements is dangerous if this way of development process is practised.
- The amount of documentation required in intermediary stages of the project, makes the management complex.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

AGILE MODEL

Agile model is the modern trend in software development methodology used to quicken the operational product/system delivery within a scheduled time frame using set of values that include adaptability, transparency, simplicity and unity in effective and light weighted manner. Agile model is extensively practiced model which has a frequent approach in simplifying the process by dividing it into short and light weighted phases such as Requirements, Specification, Architecture, Design, Implementation, Testing, Deployment and Maintenance that results in effective software systems. There are many specific agile development methods. Most encourage the development, team spirit, association, and process compliance throughout the life-cycle of the project.

Agile methods break tasks into small increments with minimal planning and do not directly involve long range planning. Iterations are very short time frames which are also called timeboxes that typically range from one to four weeks. Each of the iteration involves a cross functional team working in all tasks like *project planning, requirement analysis, system and architectural design, coding, unit level testing, and finally the acceptance testing*. At the end of every iteration, a working product is validated to the stakeholders. This allows to minimize the overall risk and allows the project to adapt to the changes quickly. Much number of iterations might be required to release a product. A model showing the agile life cycle along with its risk analysis phase is pictured in *Figure 2*.

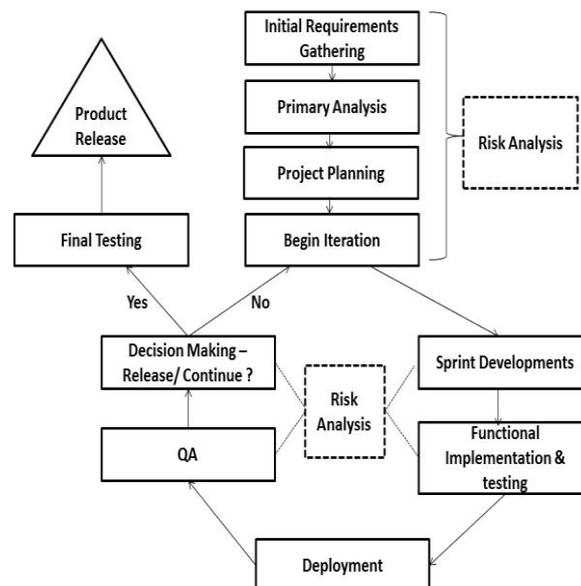


Figure 2 – Agile Model

Some of the well-established and used agile software development methods include:

- Crystal Clear
- Dynamic Systems Development Method or (DSDM)
- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Kanban (development)
- Lean software development
- Scrum

Advantages

- Customer satisfaction is guaranteed by releasing rapid, continuous delivery of the required software.
- People and interactions are accentuated rather than process and tools. Customers/users, developers and testers constantly interact with each other.
- Working software is delivered regularly (weeks rather than months).



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

- Face-to-face conversation is the best adapted form of communication.
- Close and daily interaction and cooperation between business people and developers are found.
- Continuous attention to technical perfection and good design is taken into consideration.
- Regular adaptation to changing environments is accepted.
- Welcomes changes in requirements even at the later stages.

Disadvantages

- In case of larger software projects, it is really difficult to evaluate the effort required at the starting of the software development lifecycle.
- There occurs a lack of prominence on necessary designing and documentation.
- The project can easily change track if the customer is not clear about the final outcome that they require.
- Only senior programmers are proficient in taking the proper decisions required during the development process. Hence it has no place for newbie or junior level programmers, unless combined with experienced resources.

IV. GENERIC RISK FACTORS ASSESSED DURING SDLC

Risk is always defined as the term which creates the potential future harm that may arise due to some present actions. Risk management is the most important issue involved in the software project development. Risk management consists of the different processes, methodologies and tools that are used to deal with various risk factors in the SDLC process of software development. It is also defined as the activity that identifies a risk, assesses the risk and defines the strategies to minimize or mitigate the risk. Risk Management can also be defined as the practice of systematically deciding cost effective approaches for minimizing the outcome of threat and maximizes the opportunities in the future.

Some of the basics factors regarding software risk are:-

- Risks are influencing aspects that might harmfully affect the outcome of a project.
- Risk is the direct consequence of uncertainty. If there is no uncertainty, it is never called a risk – it is known as certainty.
- Risk analysis is used to help a team to understand the different forms of uncertainty which could affect the outcome of the project.
- Risk management (sometimes called Risk Mitigation) is the method that the team puts into place to anticipate, and mitigate the effects of risk to a project.

The important aspect to remember is that even in simple projects, things are subjected to go wrong, and that you need to make plans to minimize the impact of those events when they occur. The major activities or steps in dealing with risk are:

- Risk Identification
- Risk Classification
- Risk Quantification
- Risk Planning & taking Action
- Risk Repetition based on situations.

In this paper, I have tried to classify the risk that is probable to occur during the development of software. And then to make a comparative study on the two above mentioned models and to find out the real effect of risk assessment and the impact created on these models.

Some of the major potential risk categories related with the software development, which I have chosen for study includes:-

- Mission and Goal of Organization
- Project Management Capability
- Decision Teamsters
- Organizational Management
- Customer/User Relationship
- Project Constraints

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

- Software Content
- Deployment Factors
- Development Activities
- Development Environment
- Project Management
- Project Team
- Technology Issues
- Maintenance Factors

These above mentioned risk types have many number of sub category risk too. Let us go through each and every factor and classify the risk as *low*, *medium* or *high* based upon their impacts on software development. And then compare the Spiral and Agile model to see their assessment ability to deal with the so called potential risk situations. The risk assessment models, methods and techniques are widely used to control risk in a software development. Risk assessment produces either a positive or negative outcome. Bad outcomes are risks and good outcomes are possibilities to produce good quality software. Now, let us see how we can make this risk assessment using our traditional Spiral Model and the new generation Agile Model. And find out the efficiency of these two models in Risk Assessment capacity and their management methodologies.

V. RISK ASSESSMENT CAPABILITY OF SPIRAL AND AGILE MODELS – A COMPARATIVE STUDY

According to Barry Boehm and Richard Turner, Risk analysis can be used to choose between adaptive (*agile – Value-driven*) and predictive (Spiral - *Plan-driven*) methods and have concluded that each side of the range has its own *associated features*, as shown in *Table 1*:

Models	AGILE	SPIRAL
1	Smaller Criticality	Higher Criticality
2	Senior Developers are involved	Junior Developers are also involved
3	Requirements Change frequently	Requirements do not change often
4	Small number of developers involved	Large number of developers involved
5	Principles that respond to change	Philosophy that demands order

Table 1: Aptness of Agile and Spiral Models

Table 1 describes clearly the difference in the risk assessment capability of spiral and Agile Model. It stresses mainly on five different points regarding criticality, class of developers, requirement change, number of developers involved as well as the principles and philosophies followed. Overall, we can find that Agile Model supports the overall issues and gives a lesser risk probability during the development life cycle.

Now, let us have a comparative check of the various risk factors and its associated effect on both models which are depicted in *Table 2*. The risk effect is graded as low, medium and high on the two great risk analysis models.

RISK CATEGORY	RISK FACTORS	Effect on SPIRAL	Effect on AGILE
Mission and Goal of Organization	Project Fit to Provider / Customer Organization	High	Low
	Customer Insight	High	Medium
	Work Flow	Medium	High
Project	Conflict in Goals	Low	Low



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

Management	Resource Conflict	Medium	Medium
	Customer Conflict	Low	Low
	PM Experience	Low	Medium
	Project Definition	High	Low
Decision Teamsters	Political Issues	Low	Low
	Deadlines	High	Low
	Technology Used	High	Low
Organizational Management	Stability	Low	Low
	Roles & Responsibilities	Medium	Low
	Policies & Standards	Medium	Low
	Support from Management	Low	Low
	Managerial Involvement	Medium	Low
	Project Objectives	High	Medium
Customer/ User Relationship	Customer Involvement	Medium	Low
	Customer Experience	High	High
	Customer Acceptance	Medium	Medium
	Customer Training Needs	Medium	Medium
	Customer Validation	Medium	Low
Project Constraints	Project Size	Low	High
	Hardware Limitations	Medium	Low
	Reusable Constituents	High	Low
	Budget size	High	Low
	Budget Constraints	High	Medium
	Cost Regulation	High	Low
	Delivery Issues	High	Low
	Development Calendar	High	Low
Software Content	Requirements Permanence	High	Medium
	Requirements Completeness & Clarity	High	Medium
	Testability Matters	Medium	Medium
	Design issues	High	Medium
	Implementation complications	Medium	Medium
	System Reliance	Medium	Low
Deployment factors	User Service Impact	High	Medium
	Team Involvement	Medium	Low
	External Hardware / Software Interfaces	High	Medium
Development Activities	Alternate Analysis	High	Medium
	QA Approach	Low	Low

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

	Documentation Preparation	Low	Low
	Use of specified engineering mechanisms	Low	Low
	Initial Defect Tracking	Medium	Low
Development Environment	Hardware Platform	Medium	Medium
	Availability of Tools	Medium	Medium
	Vendor Support	Medium	Medium
Project Management	Project Manager Approach	Medium	Low
	PM Interaction	Medium	Low
	PM Expertise & Knowledge	Medium	Low
	Attitude of PM	Medium	Low
	Support from PM	Medium	Low
Project Team	Availability of Team Members	Medium	Low
	Team Skills	Medium	Low
	Domain Experience	Medium	Medium
	Familiarity with Hardware & Software	Medium	Medium
	Process Experience	Low	Low
	Team Training	Medium	Medium
	Team Spirit & Attitude	Medium	Low
	Team Productivity	Medium	Low
Technology Issues	Domain Expertise	Medium	Low
	Technology compliance with Project	Medium	Low
	Team Experience with Project Technology	High	Medium
Maintenance factors	Obtainability of Technology Expertise	High	Medium
	Design Complexity	Medium	Medium
	Support Personnel	Medium	Low
	Vendor Support	Low	Low

Table 2 –Risk factors and its effect on Spiral and Agile Models

Table 2 expresses some of the major risk categories and their related risk factors based on which I have made an assessment on the impact of risk and how the two models - Spiral and Agile deal with the situation. These are the data which was collected from various software development companies and had done a quantitative study based on these facts.

Based on these facts, a quantitative means of analysis has been carried out as shown in Table 3. This exactly shows the risk rate percentage on each model. From this Table 3, it is clear that rate of high risk is very less for Agile and Medium type risk is more for Spiral Model.

International Journal of Innovative Research in Computer and Communication Engineering

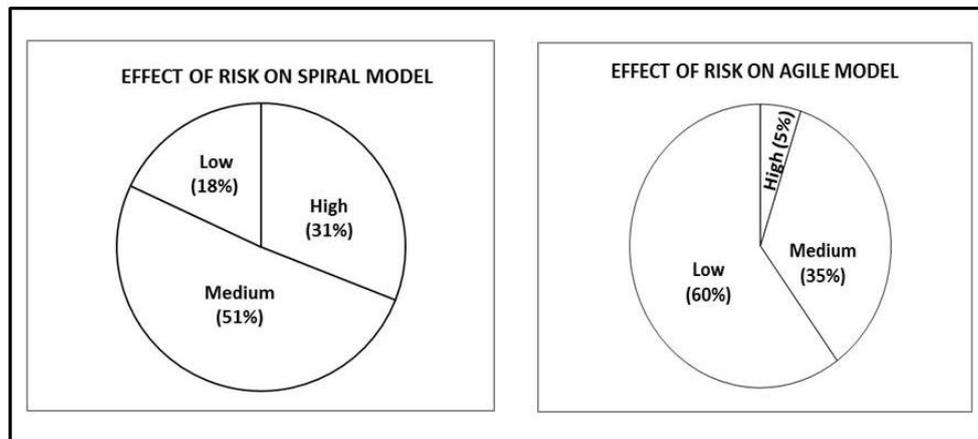
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

Rate of Risk on Spiral and Agile Model			
Risk Effect→	HIGH	MEDIUM	LOW
Model↓			
SPIRAL	31%	51%	18%
AGILE	5%	35%	60%

Table 3 – Risk Rate Percentage on Spiral and Agile

To make it more clear a pie chart is drawn using these data as shown in Graph1.



Graph1 – Pie Chart showing the effect of risk on Agile and Spiral Model.

Thus, it was found that, depending upon many of the risk factors, Agile Model showed a good effect on minimizing the risk and produces a better success frequency result in developing software using this model. Many criteria and their sub criteria’s were considered during this relative study.

VI. CONCLUSION AND FUTURE WORK

From this comparative study, we can find that the level of risk is higher when dealing with Spiral Model or the development of software when compared with the latest lightweight methodology – the Agile. Though we have a notion that Agile methodologies are inefficient in large organizations and certain types of projects, it seems to be best for development of small and non-sequential projects based on the risk assessment done herewith. Many organizations believe that agile methodologies, adopt a hybrid approach that mixes elements of agile and plan-driven -Spiral approaches. Spiral Model is found to have higher order risk than Agile Model. Even then, we can conclude that based on the type of projects and its huge size, Spiral model also plays a very important role in software development and dealing with the risk factors associated with the software project development.

Even though risk level is lower for Agile model when compared to Spiral, under many conditions like Project Size, Customer Experience and nature of Work Flow, Spiral model is also taken into consideration for software development by various organization now a days. But from analysis done, it is estimated that, majority of the modern development companies support the Agile Model considering it as their core model for live, fast deliverable, software developmental model in recent years. Risk Assessment and Product development always go hand in hand and without proper risk assessment, never ever a quality product can be developed. Thus, we can conclude that depending upon the type, complexity and size of the projects, proper selection of software developmental model



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

helps the organization to assess the various risk factors and based on which the organization can release a quality product on time and within the given budget.

REFERENCES

1. Boehm Barry W, "Software Risk Assessment", IEEE Computer Society Press, Vol.15 (7), pp.902-916,1989.
2. Michael Lant, "Five Simple Steps to Agile Risk Management", <http://michaellant.com/2010/06/04/five-simple-steps-to-agile-risk-management/>, June 4, 2010
3. Boehm, B.W. –"Software risk management: principles and practices"-Software, IEEE (Volume:8 , Issue: 1), ISSN :0740-7459,
4. Sanajana Taya – "Comparative Analysis of Software Development Life" – ISSN:2229-4333 (print), SSN:0976-8491(online), vol.2ISSUE 4, Oct- Dec2011
5. V. Szalvay, "An Introduction to Agile Software Development," technical report, Danube Technology, 2004.
6. HaneenHijazi, " A Review of Risk Management in Different Software Development Methodologies " ,International Journal of Computer Applications (0975 – 8887) Volume 45– No.7, May 2012
7. Nabil Mohammed Ali Munassar and A. Govardhan," A Comparison Between Five Models Of Software Engineering", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 5, September 2010, ISSN (Online): 1694-0814
8. R. Dash and R. Dash, "Risk assessment techniques for software development," European Journal of Scientific Research, Vol 42, No. 4, 2010, pp. 629–636.

BIOGRAPHY



M. Soumya Krishnan, working as Assistant Professor in CS & IT Dept., Amrita School of Arts and Sciences, Amrita Vishwa Vidyapeetham, Kochi Campus, has over 12 years of experience in teaching. She has done MCA (B'Level) , M.Sc. CS andM.Phil.in Computer Science.Her areas of interest are Software Engineering, Data mining, Software Development Privacy and ITES.