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. 1, Issue 9, November 2013**

# Analytical Behavior of Software Professional moral code and Issues in Agile Methodology Development

P.Thenmozhi[1], P.Balasubramanie[2]

Assistant Professor, Dept. of Computer Science, Kongu Arts and Science College, Tamilnadu, India[1]

Professor, Dept. of Computer Science & Engineering, Kongu Engineering College, Tamilnadu India[2]

***Abstract*:** Software Engineering and software development life cycle (SDLC) are terminologies in the world of software of development. This involves team playing and each one's individual ability is counted , communication gap between developers, customers and stakeholders, lack of adequate skills, may lead to error prone development. Here we try to analysis the coherence of the individual's ability and team spirit using factors like technical ability, attitude, environmental, etc. Agile methodologies place vital role in fast planed software development for customer requirement is taken into accountable.

**Keywords**: Agile Methodology, Human Factor, Software Engineering, Software professional, Risk Analysis.

## I. INTRODUCTION

Software Engineering is a systematic approach to develop, operate, maintain and upgrade the software, it is a discipline which provides tools and techniques to develop quality software in an orderly fashion and provides an engineering approach that is used to build software for end users using well defined, consistent and cost effective processes.

The Software Development Life Cycle (SDLC) is the sequence of different activities that takes place during the development phase. Any software development consists of five phases namely Requirement Analysis, Design, Implementation, Testing and Maintenance [12].The SDLC begins with the identification of the requirements of a software and ends with the valid justification for the development of software against that requirement. SDLC is the period of time that starts, when a software product is conceived and ends when the product sets development. Traditionally, the models used for SDLC have been insisting on sequential development, progressing through a number of well defined phases activities of cost estimation, risk analysis and user requirements [8].

A Software professional is always responsible for successful development of software. Identifying human resources is more difficult than identifying the software or hardware assets. Concerned people with sufficient knowledge and experience should be assigned to a particular task. Engineers and scientists share a basic drive to accomplish what they want to so as to establish their own unique identity to have a rich in the profession. It is therefore necessary to match profession with their work assignment. Enhanced technical issues and impressive interpersonal skills are very much involved in it. The development teams face many challenges and hardships as they are pressurized by several factors during the software development phase.

## II. MOTIVATION AND TECHNICAL COMPETENCE OF SOFTWARE PROFESSIONAL

Professional skill and attitude varies from person to person. It is therefore necessary to identify the talent and attitude of an individual that can be appropriately applied to the different phases of the software development [11]. The best result

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. 1, Issue 9, November 2013

can then be obtained only if different tasks are allotted among different software professionals according to their nature and ability. Hence it becomes necessary to identify the unique skill of an individual and utility the same to the software development process, thus reducing the level of bugs.

One of the aims of software development is to create artifacts of quality software. The quality of a software product has many aspects, one of which is correctness, which refers to the absence of defects and errors can be identified and removed amicably [3]. The most damaging errors are those, which are not identified during the requirement phase, development phase and remain when the system becomes active. The purposes of analysis are to identify the initial defects, reduce the development time, provide better customer service, build better applications and check whether the user requirements are satisfied or not.

The customer and the develops needs to work together with one to one conversational environment. It is really a complicated task to organize such an environment, which management delay the quick and clear development as per customer requirement. This imbalance can be overloaded by Agile methodology which characters are quick and fast development pattern, short iteration and focusing on the completion of working code.

### *A. Identification scrutiny of Technical Professionals for Agile methodology*

Every field of specialization has a unique set of talents that is responsible for its success. The Professional can generally succeed as long as the talents are reasonably consistent with the requirement of the work. The talent pool is the most important asset of an organization. Creative ideas originate from it and solve the key problems to produce the most successful product. The following discussion is required for making a talent technical professional for agile methodology.
Trained engineers know how to estimate work plan and based on the work to gather data on the sequence order to meet their planned cost and schedules with better plan in future.

Most promising professionals offer present their work to senior managers, customer or outside professional groups to audit it, this activity build the self –confidante and motivate then to structure their work more logically.

### *B. Agile Project Management framework activities*

Agile Project Management framework consists of five phases namely Envision, Speculate, Explore, adapt and close, which gives brief activities in process of agile given in the Table 1.1 below.

Table 1.1 Agile Framework phase activities

| Agile framework phase | Description Agile framework |
|---|---|
| Envision phase | Identify the product vision and Scope of the project and talent of the team. |
| Speculate phase | Planning the release of the project through Vision, Milestones and other additional features. |
| Explore phase | Consecutively check the test features with a timeframe, and reduce the risk factor in the project. |
| Adapt phase | Analysis the delivered result, current situation, and the team's performance with necessary requirement. |
| Close phase | Conclude the project along with key factors. |

The various related activities of agile framework associated with technical and human intelligence take part in task to success which key of talented team in Envision phase, additional features in Speculate phase as like customer

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. 1, Issue 9, November 2013

requirement even later stage which adapt feedback and change, Reduce the risk factor in Explore phase are software code made by developer, team's performance in Adapt phase using Analysis Of Current Situation and Conclude the project along with key factors in Close phase if concludes by team of high officals from each part of view the human involvement is more dictorial for development than using automatation tools.

*C Problem in change and rework of software development*

Software needs to be modified during the development and operation phase. Requirement can be complex during the development phase. Requirement may not be known the functionality interfaces, performance, constraints, environment and communication. Requirement may not be well Visualized also. For large project, the requirement is "assembled" from inputs provided after several changes (e.g. Re origination). Finally the change to requirements leads to rework the projects, the rework costs can be a high as 30-40% of the development costs. The software succeed Meets the needs of the people to use the End user's point of view without error prone and user requirement get translated into a well designed software system from Developer's point of view. Techniques for the software development is to satisfy themselves that they have made some progress in the right direction for success. We highlight the talent professional for agility team, recognition and selection of technical talent professional for agile process through set theory with relational based and encourage to devise tactics that can help project teams more flexible, adaptable and dynamic.

## III. CODE ANALYSIS FOR SOFTWARE PROFESSIONAL

Code analysis is intended to identify the perspective of developer knowledge. This is done through the comparative study on the programming skills of different developer in a particular scenario. Every developer has his own technique to develop a program. The style of program writing of a programmer is entirely different from others. Hence this study promotes an idea of choosing the best developer among a crew of developer based on three levels of program writing. They are:

1. Program involves simple programming with limited number of variables. Example: Simple mathematical calculations.
2. Program consists of variables at multilevel systems. Example: Use of Inheritance to access the variable from base class.
3. Program involves complex data variables in the system which depicts the importance of the variable and its scope within the system. Example: Functions used for online quiz system.

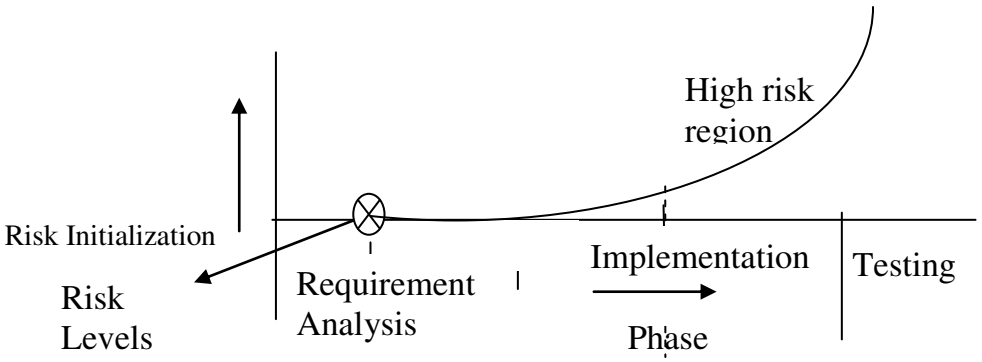


Figure 2.2 Risk Levels During the Development Phases

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. 1, Issue 9, November 2013

The calculation metrics such as Line of Code (LOC), Program Volume, Compilation Detail and Error are used for analyzing the skill levels of a programmer. The origin risk attack is at the analysis phase and goes through requirement design, implementation and ends in the testing phase. Figure 2.2 shows the areas of a project where the project manager has high level of risks in the requirement and implementation coding phase. As long as the risks are determined and securely mitigated, the quality of risk will slowly move on to its predicted path. If risks are not identified and not securely mitigated, the bugs will rapidly increase in the implementation phase and finally the software project might fail to satisfy the requirements of the customer.

Even though a software developing team could identify the risks from design phase, it has to overcome many risks that occur at coding phase also. Professionals, who lack in knowledge in a particular area, tend to increase risk in the product. The primary goal of a software development team is then to develop a code and document the development that should meet the project's requirement. The primary issue is that the software must be maintainable and reusable.

## IV. PROBLEMS WITH AGILE METHODS

t is difficult to keep the interest of developers who are involved in the process without motivating and awareness about the technology.

- Team members may be unsuited to the intense involvement that characterizes agile methods.
- Prioritizing changes can be difficult when there are multiple stakeholders.
- Maintaining simplicity requires extra work.
- Contracts may be a problem as with other approaches to iterative development.
- Fast testing lead to low risk coverage

## V. LITERATURE SURVEY: TWO LEVELS OF SOFTWARE PROFESSIONALS

Software comprising of several modules are generally developed by a team of professionals in a software company. Each module is developed by an individual or a group of Professionals. It is well known that a risk in the software depends on various factors such as skill requirements, scheduling and cost. Among these factors, skill of an individual of a particular language in which the product has to be developed will give a greater impact on risk in that software. Developers are those who are experts in a particular language and they may not be well-versed in other languages. In Contrast, those who are not experts in a particular language but they are moderately fair in many languages.

In such a way, the fundamental concept of set theory is that a membership belonging to a set or any object enables to a member or an element of that set [16]. The object in sets may be anything, say numbers, people, rivers, cars, or mountains. If an object x is a member of a set A, and represented $x \epsilon A$, which may be understood as "x belongs to A" in other word "x is an element of the set A", Symbolically, this relation is denoted by $A \subseteq B$, or equivalently $B \subseteq A$. Alternatively , $A \subseteq B \Leftrightarrow (x)\ (x \in A \rightarrow x \in B) \Leftrightarrow B \subseteq A$. By using Venn diagram of inclusion set theory i.e .,$n(A \cup B) = n(A) + n(B) - n(A \cap B)$.where, $n(A \cup B)$ refers to the strength or risk free code developed from union. Hence this phenomenon has shown the pair programming concept of Agile Methodology. The above set theory conclude that project people should be two sense, to getting the right person and that right person should be in perfect person in Behavioral and technical aspect.

In Agile programming, the effective pair programming practice concept validate through a new approach in grouping engineering level students and experience were conducted and its outcome of an efficient pair programming to share their knowledge, communication skill and latest trends of field from high level students to low level students i.e., technical transfer and similar practice was transfer from education environment to industrial environment[14]. In industrial environment, fresher learn the technical from senior programmer and update the knowledge and team work sprit for involvement of structure of pair programming.

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. 1, Issue 9, November 2013

Two type of metrics founded by [9] were as first metrics found the percentage of pair programming and solo programming is defined % pp $=E_P / E_T$ ,where as $E_P$ is pp effort (in seconds)spent in the method during an observation period, $E_T$ is the total effort (in seconds) spent in the method during the same observation period and another metrics to find the defect density of quality of code between the ration of defect per line of software codes, which these two type of the ability and technical skill of the pair programming were working together for a task. Laboratory Experiment of [7] shows the three primary aspect of pair programming measure are speed of development, quality of software design and defect rate. The measure outcome of human resource of involvement of project team work in pair programming concept through completion time, individual technical ability, code rates by independent raters, interaction relation among programmer and behavior of individual. Based on above observation, it's equally important that the software professional team with talent and technical ability is mandate for every activity in each phase of software development life cycle to develop the good quality products.

## VI. TRAITS OF SOFTWARE PROFESSIONAL VS RISK IMPACT ON AGILE FRAME WORK

Table :1.2 Few Risk Impact of Agile Manifesto

| Ineffective problem | Description | Risk impact type |
|---|---|---|
| Teamwork Obligations | • Effective team work result from combined effort of all the team members<br>• Communication between team members<br>• Making and meeting commitments<br>• Participation in team activities | Personnel and staffing risk |
| Development of Manager Goal | • Fully utilize the team member skills and ability<br>• Better on how well it was performed<br>• Better on helpfulness and Produce a Superior Product<br>• SRS and SDS<br>• Product met its function and operational objectives support | Requirement risk |
| Failure to compromise or cooperate | • Not willing or Able to work cooperatively with the team | Managing risk |
| Procrastination and lack of confidence | • Not set deadlines or establish goal and milestones. | Requirement risk |
| Function creep | • No clear dividing line between the function that stem from interpretation of the requirement. | Technical risk |
| Ineffective peer evaluation | • Grading in team course is not entirely fair.<br>• Competition among team members and can reduce the willingnesss of team members to fully cooperate. | Managing risk |

The success or failure of a project is sildom due to various technical and magement issues, these two issues are important factor for success or failure of project and the problem issues may be real or imagine. Software professional worry about the imaginary problem and difficulties because that may not be real, the team responds to this pressure by alternative in shortcut using methods, technical, languages, tools etc. Each software professionals respond for every

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. 1, Issue 9, November 2013

alternate in a team, a team should be constructive manner to solve the technical and managing issues based on behavior of teamwork and its relationship with risk categories [3], each category risks addressed individually for avoidance and mitigation strategies which is given Table 1.2.

In Agile processing models, Extreme Programming, Scrum and pair programming are associated terms for software development methodology which improve the quality of various activities and support the customer requirement changes in the product and these manifesto use made by team sprit with soft relationship among the members which is proven earlier. It shows various issues involved about the technical ability and human factors that performance attitude in solving the problem in software development that similarly table 1.3 describes about the requirement of talent pool for software profession takes part in development stages and the ability of the requirement of human behavior and technical characteristics to match the selection of agile team.

Table 1.3 Human Vs Technical Relational Behavior Perspective of Agile team

| Gentral Human Behavior Activies | Technical Behaviour Activies for agile taem. |
|---|---|
| Skill | Ability ,talent ,cleverness, Proficiency ,Expertise. |
| Attitude | Approach ,outlook, manner, stance. |
| Ethical | Moral ,principled, right, fair. |
| Intellectual discipline | Thinker ,brain, scholar, rational, logical. |
| leadership | Management, control, guidance, headship, direction. |
| Innovation | Novelty ,modernism, modernization, improvement. |
| Management | Administration ,supervision, running, managing. |
| Assessment | Appraisal, evaluation, estimation, measurement, review, consideration ,opinion. |
| Ability(decision making + fuzzy problem) | Aptitude ,capability,capacity. |
| Mutual respect | Shared ,joint,common. |
| Self organization | Personality ,nature,identify,association,group,union. |
| Competence | Capability ,fitness,aptitude. |
| Common focus | Ordinary,widespread,frequent, familiar,regular. |

## VII. CONCLUSION

Levels of software professional analytics are used to understand the various skills of programmers in code development. It also represents the growth of system with various levels of programmers involved in the development of a system. This depicts the efficiency of risk levels when used with different people for development, it shows that the risk level gets reduced when probability people are combined. It satisfies the set theory relationship based on personnel behaviors. It also predicts the goal of a system when they comprise both the professionals for effective achievement of the goal and it order to make a system efficient. The reduction of mismatch ability between the software professional people used in development stages which would be effective during development time, on effective memory utilization and on-time delivery of the products. It will be useful in Scrum, which ignores risk managing completely and a great deal to explicit risk management becomes unnecessary when a project uses agile approach.

## REFERENCES

1. Brain Fitzgerald, Klaas-Janstol.,Ryan O'Sullivan and Donal O'Brien, "Scaling Agile methods to Regulated environments: An Industry Case Study", Proceeding on 35th International Conference on Software Engineering (ICSE 2013), pp. 863-892, 2013.
2. Charette R.N, 'Software Engineering Risk Analysis and Management", McGraw Hill, New York, pp. 12-20, 1989.
3. Debbie Tesch., Timothy.J.Kloppenborg and Mark.N.Froclick, "IT project risk factors: The Project Management Professional Perspective", Journal of Computer Information System, Vol 4, Iss 2, pp 61-69, 2007.
4. Dietmar Winkler, Martin Kitzler, Christoph Steindl, and Stefan Biffl, "Investigating the Impact of Experience and Solo/Pair Programming on Coding Efficiency: Results and Experiences from Coding Contests", Lecture Notes in Business Information Processing ,Vol. 149, pp 106-120, 2013.

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. 1, Issue 9, November 2013**

5. Gaurav Kumar.,and Pradeep Kumar Bhatia, "Impact of Agile Methodology on software development process", International Journal of Computer Technology and Electronics Engineering(IJCTEE), Vol. 2, Iss.4, pp 46-49, 2013.
6. Han Van Loon , "A Management Methodology to Reduce Risk and Improve Quality", IT Professional, Vol. 9, No.6, pp. 30-35, 2007.
7. Janusz Gorski and Jakub Miler, "Risk Identification Pattern for Software Projects", Journal of Foundations of Computing and Decision Science, Vol.29, Iss 2, pp.115-131, 2004.
8. Naresh Kumar, A. S. Zadgaonkar, and Abhinav Shukla, "Evolving a New Software Development Life Cycle Model SDLC-2013 with Client Satisfaction", International Journal of Soft Computing and Engineering (IJSCE), Vol. 3, Iss 1, pp 216-221, 2013.
9. Nattakarn Phaphoom., Alberto Sillitti.,and Giancarlo Succi, "Pair Programming and Software Defects--A Large, Industrial Case Study", IEEE Transaction on Software Engineering, vol. 39, Iss. 7, pp. 930-953, 2013.
10. Orit Hazzan and Jim Tomayko, "Teaching human aspects of software engineering", Proceedings of the 27th international conference on Software engineering, pp.647-648, 2005.
11. Padberg, F. and , "Analyzing the cost and benefit of pair programming", Proceedings of Ninth International Software Metrics Symposium, 3-5 Sept. 2003, pp. 166 – 177, 2003.
12. Pressman R.S, "Software Engineering: A Practitioner's Approach", Fifth Edition, McGraw-Hill International Edition, pp. 20-30, 2000.
13. Singaravel G., Palanisamy V. and Krishnan A, "Ethics Understanding Of Software Professional In Risk Reducing Reusability Coding Using Inclusion Set Theory", International Journal of Computer Science and Information Security (IJCSIS), Vol. 6, Iss. 2, pp. 189-196, 2009.
14. Sukhpal Singh and Inderveer Chana, "Enabling Reusability in Agile Software Development", Journal of Computer Application, Vol.50, Iss.13, 2012.
15. Tockey Steve, "Recommended Skills and Knowledge for Software Engineers", Proceedings of the 12th Conference on Software Engineering Education and Training, pp. 168, 1999.
16. Tremblay J.P. and Manohar R, "Discrete Mathematical Structures with application to Computer Science", McGraw Hill, pp. 104-121, 1989.
17. Venkata Vinod Kumar Padmanabhuni, Hari Praveen Tadiparthi, Muralidhar Yanamadala ,and Sagar Madina, "Effective Pair Programming Practice- An Experimental Study", Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, Iss. 4, pp.471-479, 2012.
18. Watts S.Humphrey, 'Introduction to the Team Software Process", Pearson Education, pp. 250-260, 2009.

## BIOGRAPHY

**P.Thenmozhi** has completed her M.Phil degree in Mother Teresa women's University kodaikanal in 2004.She has completed 12 years of service in teaching. Currently she is Assistant Professor, Department of Computer Science, Kongu Arts and Science college, Tamilnadu, INDIA. She has guided 3 M.Phil students and presented 7 papers in various conferences.

**P.Balasubramanie** was qualified in the Junior Research Fellowship (CSIR) in the year 1990 and Awarded Ph.D degree in the year 1996 by Anna University, Chennai. He has completed 18 years of his dedicated service in Engineering Colleges. Currently he is working as a professor in the department of Computer Science & Engineering in Kongu Engineering College, Perundurai, Erode. He is the approved supervisor of Anna University and guided 20 research scholars. He is guiding 12 research scholars. He has published 136 research articles in reputed international journals. He has published 10 books with the reputed publishers. He is the recipient of the Cognizant Technology Solutions(CTS) award 2008, P.R. Dass Memorial Award, 2011, and Best Academic researcher Award 2012 by Association of Scientists, Developers and Faculties (ASDF). He has completed one AICTE Research Promotion Scheme(RPS) as a principal investigator. He has organized more than 20 seminars/workshops/SDPs/STTPs Sponsored by number of funding agencies for the benefit of faculty members and research scholars. He is actively involved in consultancy(software development) work and completed number of works. His specialization includes optimization Techniques, Data mining, and Networking.