# Process Redesign to Recover Deficits and Difficulties of Software Lifecycle in Micro Enterprises

**Partha Protim Roy**
**Student Id: 012152012**

A Thesis

in

the Department

of

Computer Science and Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Science in Computer Science and Engineering

United International University

Dhaka, Bangladesh

June 2020

# Approval Certificate

This thesis titled " **Process Redesign to Recover Deficits and Difficulties of Software Lifecycle in Micro Enterprises**” submitted by **Partha Protim Roy**, Student ID: **012152012**, has been accepted as Satisfactory in fulfillment of the requirement for the degree of Master of Science in Computer Science and Engineering on 2nd June 2020.
**Board of Examiners**

1.

______________________________ Supervisor

Dr. Hasan Sarwar
Professor
Department of Computer Science and Engineering
United International University
Dhaka, Bangladesh

2.

______________________________ Head Examiner

Mr. Mohammad Moniruzzaman
Assistant Professor
Department of Computer Science and Engineering
United International University
Dhaka, Bangladesh

3.

______________________________ Examiner-I

Mr. Suman Ahmmed
Assistant Professor
Department of Computer Science and Engineering
United International University
Dhaka, Bangladesh

4.

______________________________ Examiner-II

Mr. Md. Benzir Ahmed
Assistant Professor
Department of Computer Science and Engineering
United International University
Dhaka, Bangladesh

5.

______________________________ Ex-Officio

Dr. Mohammad Nurul Huda
Director, Master of Science in Computer Science and Engineering
Professor, Department of Computer Science and Engineering
United International University
Dhaka, Bangladesh

# Declaration

This is to certify that the work entitled “**Process redesign to recover deficits and difficulties of software lifecycle in micro-enterprises** " is the outcome of the research carried out by me under the supervision of Prof. Dr. Hasan Sarwar.

______________________________________

Partha Protim Roy
012152012
Master of Science in Computer Science and Engineering
United International University
Dhaka, Bangladesh

In my capacity as supervisor of the candidate’s thesis, I certify that the above statements are true to the best of my knowledge.

______________________________________

Dr. Hasan Sarwar
Professor
Department of Computer Science and Engineering
United International University
Dhaka, Bangladesh

# Abstract

The concept of a startup is growing around the world where innovation is the major driving force. Managing such a small startup company from both a technical and administrative perspective is a challenging issue. Though several approaches for software development, like the waterfall model, agile model, SCRUM model are available, it is difficult to implement such a model with full features due to the limited number of resources, lack of experienced resources, and lack of enough financial capability. The scopes of software process improvement are even narrower in a situation where they hardly manage or adapt a methodology. As a result, small companies suffer and work without a good quality framework model. Companies struggle to produce quality software. In this work, we have suggested an integrated software engineering methodology, in which, the process improvement framework is a built-in phenomenon. Our system is dubbed as a 3D model, where the Ds stand for Design, Develop and Deploy. The Design indicates the design of the product making as well as the process design. Develop and Deploy is also hinting to the product and process. The process improvement is managed through observation and management, which are fed back into the system through proper documentation. We did some client surveys, organization surveys, and observed work patterns of a group of human resources for a period. We measured some of the parameters before and after the implementation of the 3D model. We found that the implementation of the 3D model gives an improved process and product development environment with no extra manpower or extra costing.

# Acknowledgment

First and foremost, I would like to thank God for the blessing of making me able to accomplish the thesis with success. Then I would be delighted to thank my professor Prof. Dr. Hasan Sarwar whose invaluable supervision helped me anticipate this thesis and make it an extraordinary success. His suggestions and right directions have steered to the execution of the thesis.

Then I would like to thank my parents, sibling, best friend, and friends for providing me with their unconditional support and constant encouragement throughout the process of researching and writing this thesis.

# Table of Contents

List of Tables…………………………………………………………………………… vii

List of Figures…………………………………………………………………………. viii

1. Introduction…………………………………………………………………….. 1

2. Background and Literature Review.................................................................... 2

2.1 Software Engineering Methodology ………………………...………………… 2

2.2 Software Process Improvement Framework……………………………………… 2

3. Methodology ......................................................................................................... 4

3.1 Company Survey………………….…………………………………………… 4

3.2 Client Survey…………………………………………………………………... 4

3.3 Developer Inspection…………………………………………………………4

3.4 Survey Result………………………………………………………………... 5

4. Result…………………………………………………………………………… 6

4.1 3D Model………………………………………………………….……………. 6

4.2 Block Diagram of 3D Model…………………………………………………… 7

4.2.1 Design…………………………………………………………………… 8

4.2.2 Development….…………………………………………………………. 8

4.2.3 Deploy...…………………………………………………………….. 9

4.2.4 Observation...…………………………………………………………… 9

4.2.5 Documentation..………………………………………………………… 10

4.2.6 Management.…………………………………………………………... 10

4.2.7 Justification of name 3D…………………………………………………… 11

4.3 Relations of Development and Process Improvement Life Cycle………………. 13

5. Case Study ……………………………………………………………………….. 14

5.1 Developer Satisfaction…………………………………………………………... 15

5.2 Rework Level……………………………………………………………………… 16

5.3 Client Satisfaction……………………………………………………………….. 18

5.4 Process Maturity…………………………………………………………………… 19

5.4.1 Maturity in Behavior………………………………………………………… 19

5.4.2 Knowledge State of Team Members……………………………………… 19

5.5 Summary of Survey Results with 3D Model…..…………………………………… 21

5.6 Observation, Management, and Documentation………………………………… 22

5.6.1 Business Monitor…………………………………………………………….… 23

5.6.2 Development Monitor……………………………………………………... 24

6. Discussion and Conclusion………………………………………………………….. 25

6.1 Comparative Study with SPI Model……………..……………………………… 26

6.2 A Comparative Study of the 3D Model…………………………………………. 27

6.3 Flow Chart………………………………………………………………………… 28

7. Future Work……………………………………………………………………………. 29

8. References…………………………………………………………………………... 30

9. Appendix A…………………………………………………………………………. 31

10. Appendix B………………………………………………………………………… 43

## LIST OF TABLES

Table 1: The Results of Top Questions of 5 Micro Company Survey ............................... 5

Table 2: Improvement of the 4(four) Parameters After the Use of the 3D Model ........... 21

Table 3: A Comparative Study with CMMI and other SPI Approaches.......................... 26

Table 4: A Comparative Study of the 3D Model with Waterfall, Spiral, Incremental Approach ........................................................................................................ 27

# LIST OF FIGURES

Figure 1: A Software Life Cycle with Integrated SPI ...................................................... 6

Figure 2: Block Diagram of the Integrated Software Model ........................................... 12

Figure 3: A Block Diagram Depicting the Relations of Development and Process Improvement Life Cycle ........................................................................................ 13

Figure 4: Use of Observation, Management, and Documentation to Improve the Product ........................................................................................................................ 28

# Chapter 1

## Introduction

The concept of a startup is growing around the world. Innovation is the major driving force for emerging economies and more startups are coming with new innovations. Innovative ideas are generating new companies. As a result, many new small software companies around the globe are emerging. Recent statistics suggest small companies account for 93% in Europe, 56% in the US, 66% globally [1]. These companies employ around 15-30 resources. In spite of the fact of being small, many of the companies are developing a large ERP system for their clients. They are also providing support services for those clients. In the meantime, they trying to maintain communication as they know communication and cooperation in the ERP system are important factors for improving performance in post-implementation [9].

Clients always expect the best service while the companies struggle to provide the best service as the system is sold at a cheaper price and not possible to maintain different specialized departments by the software companies. Many of the times, a single resource acts as a developer or requirement analyst or even a tester or trainer. In such an overlapping job responsibility, maintaining a strict workflow like Analysis, Design, Development, Testing, and deployment for a large project is complicated. This situation gets worsened while a lot of change requirements are generated even before the real use of the software.

We all know that companies have to follow a specific methodology for the development of software. These methodologies are known as Software Development Model. Analyzing the actual context, it is hard to follow the methodologies with their small budget, a small number of employees, and the complexity of works. There are some frameworks besides these methodologies which are used to speed up the company's process and improve its approach. We call those frameworks the Software Process Improvement framework. Software development methodology, as well as SPI frameworks, become more difficult to implement in companies.

# Chapter 2

# Background and Literature Review

## 2.1 Software Engineering Methodology

As we know, there are certain methodologies that can be used for software development purposes which are known as the waterfall model, agile model, SCRUM model. Small companies try to follow such models or a combination of these models. Every model has its own requirements. Waterfall demands that the client requirement should be collected fully before the design process begins. Small companies are not always capable to extract real requirements from clients due to their inexperience in system building or inexperienced resource persons. Especially in a culture where clients are not aware of the importance of the requirement collection process and always explain their requirement after the product comes for deployment. Agile models demand serious domain knowledge from the team leader. Since in agile models, developers collect requirements in iterative style and develop an iterative partial version of the product, lack of domain experience from analysts' part may guide the project in a wrong direction, which reveals at such a stage of the project we're starting from scratch is not at all possible.

## 2.2 Software Process Improvement Framework

There are also some frameworks known as the Software Process Improvement framework. These frameworks are specially meant to initiate a quality practice in any kind of software development methodology. Of them, CMMI, ISO, and 6 Sigma are commonly known and widely practiced industry standards. Implementation of these quality frameworks requires certain specifications of the company format and development process. Small companies cannot afford such process standards as these require the implementation of certain engineering practices. As a result, small companies suffer without a good quality framework model. In such a scenario, companies are struggling to produce quality software with minimal cost and effort.

In most of the works of SPI for large or small industries, SPI is an added process on top of the regular engineering methodology. One approach suggests improving the global level of software practices by introducing some high-level goals and operational goals. [2-3] In another work, SPI has been tried by aligning the software processes with reference models like CMMI-DEV and ISO/IEC 15504 [4].  The software process matrix has been developed along with some practices [5]. MESOPYME is another model that tries to create a method to assess and improve software processes. [6]. PRISM is another model for implementing process improvement. Here, a separate team is employed that finds plans to improve the current process and comes up with a revised process model. All these efforts indicate a common pattern of process improvement approach. The pattern is that process itself is in its place. Process improvement is a separate approach where surveys are conducted, and teams come up with an improvement plan in the process.

# Chapter 3

# Methodology

As an exploratory survey, we conducted some surveys on a few micro-companies. We carried the survey into three main parts. In the first part, we took a detailed interview with software companies and about their processes. In the second part, we collected some clients' feedback about the software company and its services. Finally, in the third part, we analyzed and planned a work process at a micro software company. Survey briefs are given below.

## 3.1 Company Survey

There were questions in 9 different categories. Survey questions were related to employee information, requirement analysis, project design, development process, deployment, testing, service, support, and client.

## 3.2 Client Survey

There were 21 questions in 6 categories. The categories were requirements, product assessment, communication, support, software training, and general assessment. Questions were more targeted to get client feedback on the above issues.

## 3.3 Developer Inspection

A questionnaire was created that was circulated among the developers. The questionnaire was given each day on a regular basis for a certain time period. Developers used to fill up the form at the end of each day. The filling pattern, the data entered into the form suggests the work pattern of the employees.

## 3.4 Survey Result

We prepared a set of questions to use as a basis of software companies' life cycle and their process. The questions were designed to capture information on their current situation.

| **Categories** | **Questions** | **Results** | | | | |
|---|---|---|---|---|---|---|
| | | **Company 1** | **Company 2** | **Company 3** | **Company 4** | **Company 5** |
| **Information about the Company** | Use of Software life cycle | No | Private | No | No | Private |
| | Causes of hampering their business goal | Time to finish a Task is high | Client Changes, Management | Not focus on sale and marketing | Over talking | Fixed Client |
| **Employee information** | Number of employees | >20 | <20 | <20 | <10 | <10 |
| | Multitasking | Yes | Yes | Yes | No | No |
| | Over timing | Yes | No | Yes | No | Yes |
| **Proposal and requirement analysis** | Analyzing cost, risk, resource and estimated delivery time | Other Ways | Manual analysis of resource and requirements | Manual analysis of resource and requirements | Manual analysis of resource and requirements | Manual analysis of resource and requirements |
| | Time spending to collect requirements | >90 days | <7 days | <7 days | <90 days | <7 days |
| | Causes of changing requirements frequently by clients | *For their Mistake<br>*Other reasons | *Lack of Clients domain knowledge<br>* For their mistake | *Other reasons | *Lack of Clients domain knowledge | *Lack of Clients domain knowledge |
| **Software project design** | Using the framework | Yes | No | No | Yes | Yes |
| **Development process** | Maintaining the process for development | No | No | No | No | No |
| | Time of development | Depend on Client’s Expectation | Depend on Client’s Expectation | Depend on Client’s Expectation | Depend on all resource | Depend on Client’s Expectation |
| | Reason for not using a popular or conventional SDLC | Popular SDLC will not work | Lack of resource, Not matching with a custom strategy | Deduction of energy, Waste of time and effort | Lack of domain knowledge | Not appropriate for them |
| **Deployment** | Deployment team (without developers) | Yes | Yes | No | No | No |
| | The training process for trainer | Ownself, Other ways | N/A | Internal discussion | Official guide | Ownself |
| **Software testing** | Software testing team and No. | No | Yes, <3 | Yes, <5 | No | No |
| | Number of errors after delivery | <10 | <10 | <5 | <5 | <5 |
| **Service and support** | Support team | Yes | No | Yes | Yes | No |
| **Client** | Follow-up clients after completing the project | Yes | No | Yes | No | Yes |
| | The unexpected situation from clients | Client rejects the solution and starts to give new requirements | Instant budget and Giving a call to the client for payment | Emergency requirement during Holiday | Giving a call to the client for payment | Giving a call to the client for payment |

Table 1: The Results of Top Questions of 5 Micro Company Survey

# Chapter 4

# Results

## 4.1 3D Model

We have come up with an integrated approach. The software engineering methodology has been re-engineered in a way that encompasses the process improvement framework and product development framework in one structure as shown in Fig.1. Against the usual practice of finding the scopes where the process can be improved by a separate set of activities, the following diagram shows ways where the process itself is insisted to get improved by following a secondary circle. Basically, the main cycle of software development is presented as DESIGN, DEVELOP, DEPLOY in these 3 stages.

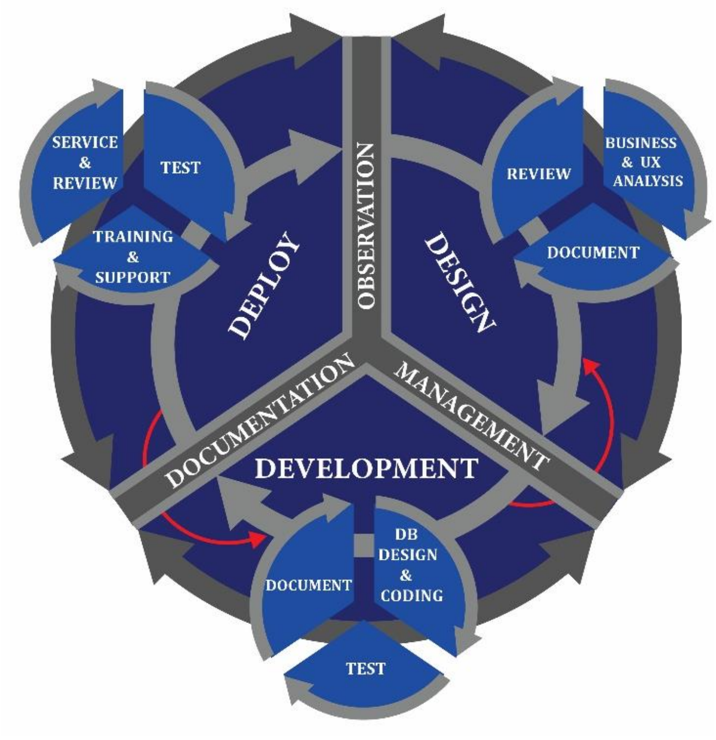


Figure 1: A Software Life Cycle with Integrated SPI

Secondary circles for Design are Business & UX Analysis, Document, and Review. Now while Design is concerned, we will focus on the Design of the process as well. And the same secondary cycle will work here for process improvement. Especially the part of Review and document corresponds to the process and the process improvement together. The next stage is Develop, which has the corresponding secondary cycle of DB Design & Coding, Test, Document. Here especially Testing and Document checking can be used as part of process improvement as well. The third D is Deploy, As the cycle is concerned, the deploy stage is used to transfer the program to the user. The secondary circle may be used for performance improvement. It is not essential that all the secondary client-side are to be followed with equal importance in the 3 stages. The level of emphasis will be determined by another circular (outer) top-view operation consists of 3 activities, which are Observation, Management, Documentation. Apart from the developers and support personnel of them, a team of 1 or more members is appointed as a monitor of a project. Monitor's job responsibility is to observe the work progress, find out the bottlenecks, discuss with the management of both the developer organization and the client organization and improve or emphasize the processes mentioned at the secondary circles.

## 4.2 Block Diagram of 3D Model

A stepwise typical work routine for the 3 (three) stages of development is given below. These steps have been established as the most optimized best practice which might be followed strictly in all types of projects. However, it is also a guideline as well. The selection of steps depends on the Management team and the type of project involved. Basically, it can be designed as follow.

### 4.2.1 Design

The design includes taking several decisions. Requirement collection is the first part of the design process. Here design refers to the activities of designing the present flow and the proposed flow of the system. The designated person will have to write the document and get this document reviewed by the stakeholders. This was requirement will be fixed in this stage.

After the requirement is reviewed and finalized, the design team will develop specific use cases and related functionalities for all the features. If necessary, they will communicate the stakeholder for further clarification of the requirements.

The generated document of use cases, functionalities, and process flow will be handed over to the development team where they will design the database, the User interface, and User experience. They will send it to the upper management and the SRS team for further feedback and review. Once these designs are approved, the documentation will be sent to the Development team.

### 4.2.2 Development

The development team will implement the database, required front end and back end programming. Once the development is done, it will be reviewed in a joint meeting at the presence of the developer and the team who did the design part. There will be several revisions through a joint team meeting where development output will be presented and explained to the design team. Upon each meeting, the developers’ team will modify the stated modifications or corrections. The developers will create documentation mainly in the coding part.

The design team will test the product thoroughly and give documented feedback to the development team. This review will also have one or more cycles until the expected outcome is achieved.

### 4.2.3 Deploy

Finally, the deployment will be done. Deployment personnel engaged in this process transfer the knowledge on the usage of the developed software.

The deployment team does the field level testing of the software in association with the stakeholder with real data.

Deployment personnel consults with the stakeholder, get their reactions and feedback, and transmits the information back to the design team. The Design team again follows all the steps from start to this point for the implementation of the required changes or modifications.

On top of that, there are 3 more operations that make this model different is the presence of Management, Observation, and Documentation. These are the 3 processes or activities that run independently of the above 3 activities of Design, Development, and Deploy. Though The extract from Design, Development and Deploy processes contribute to the Observation, Documentation, and Management process. These 3 processes are described below.

### 4.2.4 Observation

The clear meaning of observation is to closely monitor something or someone. Here, to play this role some person is engaged, whose job is to observe all the people and processes that are involved in the Design, Develop, and deploy activities. It is observed that each of the activities or some of the sample activities are happening in the right order or right direction. The success of Software development is mostly dependent on the successful understanding of requirements and the successful design of the product with perfect deployment at the right time and order. These are very much dependent on human behavior and interaction. For example, collecting requirements from a wrong stakeholder will create an unwanted final product. The responsibility of the observer is to check if the design team is communicating with the right stakeholder. Again, even if the right stakeholder is involved, the mode of communication might not be perfectly suited to the stakeholder for them to express fully until the end. Eventually, it becomes to bring the partial or incomplete requirement which ends up with lots of change requirements after the final delivery. The role of the observer is to dynamically observe every part of the activities and find out if there is any gap in the process that might accumulate into a faulty product.

### 4.2.5 Documentation

This part is clear. The observer is bound to document the issues it finds in the observation process. Again, it is understood that documentation is the only of exchanging information among the stakeholders. So, this documentation moves to the upper management team.

### 4.2.6 Management

The upper management of both the end meaning the supplier and the stakeholder is made aware of the documentation of observation and make necessary changes in the process to align the processes toward generating the right result or product. Management not always come in front or get into action. It depends on the Observer team whose produced document creates such an environment.

This is how the 3 basic processes of Design, Development, and Deployment is improved or substantiated by the Observation, Documentation, and Management processes.

### 4.2.7 Justification of the name 3D

The justification of the name 3D comes from the 2 facts as below.

There are 3 Ds like Design, Development, and Deployment. All operations have been phased in these 3 stages.

Secondly, the third eye of Observer and Management is there to improve the processes happening in the base activities. This third eye is working over the Design, Development, and Deployment process, which is synonymous with process improvement that happens in the SPI framework in traditional scenarios.

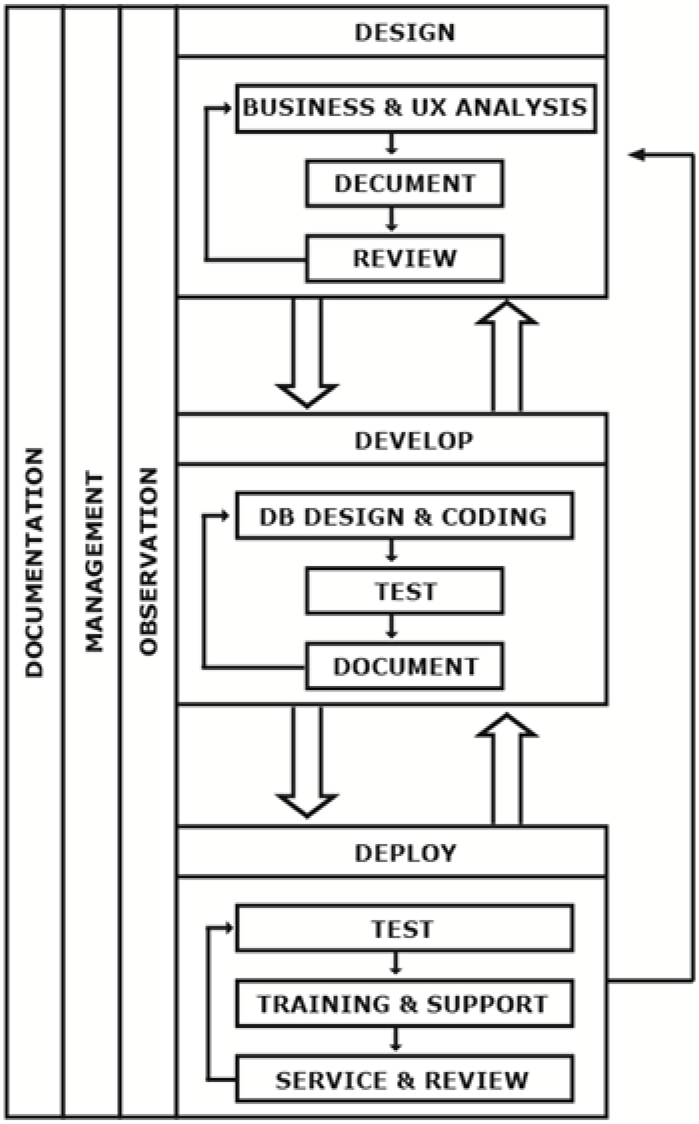


Figure 2: Block Diagram of the Integrated Software Model

## 4.3 Relations of Development and Process Improvement Life Cycle

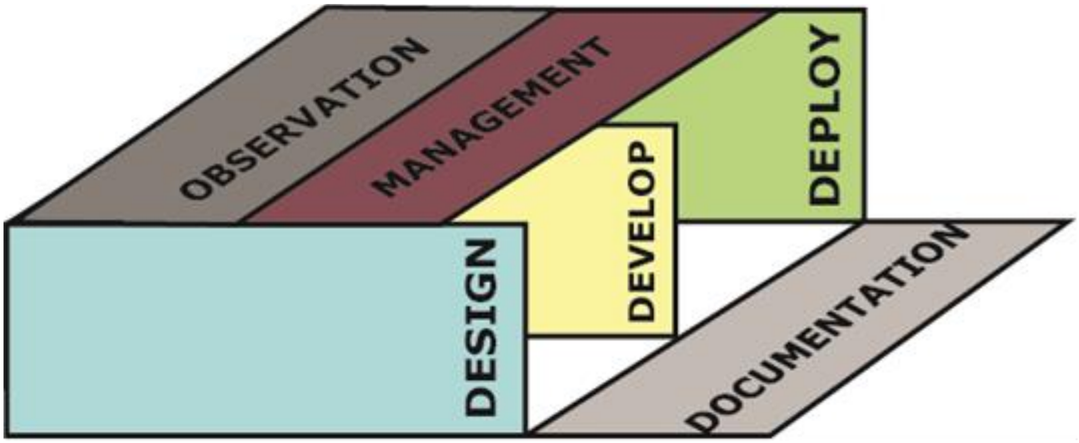


Figure 3: A Block Diagram Depicting the Relations of Development and Process Improvement Life Cycle

Figure 2 shows the block diagram of the Software life cycle phases along with the SPI operations. Observation, management, and documentation are the 3 operations that are viewing the complete development cycle. This phenomenon is better understood in Figure 3. Observation is done as a top view for DESIGN, DEVELOP, and DEPLOY stages. The parameters that are observed are eventually the level of engagement of the developers, quality of the produced software, or client's reaction to using the product. Based on the observed value, the Management changes the work pattern of the 3 phases, do the required documentation, and provide intermittent training or counseling to the respective teams for better outputs.

# Chapter 5

## Case Study

We chose a project as our case study. The project was struggling to be succeeded. So, we decided to apply this new model and see if there is some progress in the implementation of the project. It was a challenge for us to design a study with the following objective.

1. Assessment of the present scenario for 3 months
2. Identify some observable parameters to assess the present scenario
3. Implementation of some of the features of the 3D model in the present scenario for 2 months
4. Measure the values of those observational parameters after the implementation of the 3D model for a month

After lots of observation of software development scenarios in the present context, we have identified the following parameters.

1. Developer Satisfaction
2. Rework Level
3. Client Satisfaction
4. Company’s process maturity

## 5.1 Developer Satisfaction

Developer satisfaction is a parameter that usually does not follow mathematics. However, when we go and talk to a developer, his/her satisfaction level can easily be understood. No person can compromise his/her satisfaction level when expressing so. For example, a question like “How much you feel comfortable with the present state of the process of the company? And Are you satisfied with it?” is good enough to understand the easiness a developer is facing there. When a developer knows that he is working on such a requirement which is not clear and there are a lot of change requirements that will be coming, as usual, destroys the developer’s spirit of delivering a quality product. So, we have chosen developer satisfaction as an important parameter to understand the present scenario.

Measurement of developer satisfaction before and after the Model implementation was done as below.

We did a survey twice, namely before and after the implementation of the 3D model. The survey consisted of a few questions that indicate the comfort zone of an engineer involved in the development process. For example, if the requirement specification was not documented, the developer took more time to fix his work after the first delivery. It took more time to even complete a single task in different cycles resulting wastage of time. A developer caught in such a situation always expresses a lower satisfaction level. Initially, we found a developer satisfaction of 46%, which is equivalent to below average. This value is the average of the perceived easiness of several developers’ comfort levels. Afterward, a portion of the 3D model was implemented which brought the following change in the process.

We found that there was a lack of cooperation and understanding between the requirement collection team and the development team. Precisely, the relation between these two teams was not friendly or understanding. It was always thought by the development team that the poor product quality was due to poor requirement collection. On the other side, the requirement team felt that the other team is not performing the work well. It was observed that the transmission of knowledge from the requirement team to the development team was simply through verbal communication and on an ad-hoc basis. As a result, developers were never informed about the full requirement. We tried to improve the process here. We implemented documented knowledge sharing to a limited scale between the two teams. Of course, it was not possible to implement all the features of the 3D model in a short span of time. But a simpler implementation of these features resulted in a situation where the developer feels more confident about the requirement and therefore less time was wasted due to confusion or indecision. In table I, we can see that the implementation of the 3D approach improved the developer satisfaction level from 0.46 (below average) to 0.63 (average). These numbers are projected as 0 – not evaluated, .2 – Low, .4 – below average, .6 – average, .8 – good, 1 – excellent.

## 5.2 Rework Level

This parameter defines the number of reworks. Here, Rework is defined as a number which indicates how many times a single task is sent back to the developers for further fixing of bugs or change requirement. In the traditional approach, usually after the collection of user requirements, the requirement was transmitted to the developer almost instantly as it is. The developer used to do the development accordingly. When the output was sent to the user, in most of the cases, it was found the user disagrees to accept the system and comes up with a lot of new requirements in the flow of the product. As a result, rework must be done on that item and again new output was given. Unfortunately, this time again, we find the same things happening. It is found that a simple interface takes 3-4 weeks of developer time while before the start of work, it was anticipated to be a 1 week of work only. Drastically the output of the developers was affected due to this rework. Se we started to count the number of reworks being done for a few tasks and found that at worst case, 20 times a task is bounced back for correction. There were quite simple and silly corrections like a spelling mistake or data alignment in this count. So, we decided to reduce the number of times a task is being given to a developer. To reduce the number of reworks, we applied the Observation activities on the whole process and found out the following.

1. When a user is providing a requirement, he is never able to see the system as a whole, he only considers the local part of the problem without considering its effects on the big picture. As a result, the firsthand requirement specification is a local problem rather than addressing the real whole problem. As soon the developer provides the output, the user immediately goes to the next step of the bigger process and finds that the current solution does not properly fit within the whole picture. Immediately the user starts to change the requirements and subsequently another new system is started to build.

2. When a user receives a product, the user's reaction time is very fast, meaning that the user quickly starts to comment on the new system. After 2-5 comments, they make, they stop there and request to make the necessary changes. The developer also thinks that if he/she can fix the issues fast, they will be able to close the product development. But unfortunately, as soon the second delivery goes, users resume to find faults at the point where they stopped. So again, the developer needs to open the code and start fixing the next bunch of fixes. And it goes for several times.

After observing this scenario, the implementation of the 3D model was done as below.

1. The collection of requirements from the user was taken in the Design phase as stated earlier. The requirements were documented and sent back to the user for more time requesting a lot of clarifications. In this way, the requirements were made stabilized considering the big picture of the system. Afterward, direct Requirements documents were not transmitted to the developers. Another set of documentation stating the flow of the User Interface and the flow of the process with all functional specification was given to the developer. It was found that this time the first output of the developer was not fully rejected by the user.

2. Secondly after receiving the first output, when the user gave the first feedbacks of change requirement and bug fixes, we documented their requirement and sent back it to the user again for further feedback to be added or corrected in the document. It was found that the number of times this feedback was sent to the user, they changed the

content of the document several times. So before reaching the developer, several revisions were done through the communication between Deploy team and the user. As a result, the second delivery was not attacked with so many more changes like before.

Finally, it reduced the rework level as shown in the table. The table shows that just only implementing the above 2 steps, we found that in the worst cases, rework level, meaning the re-engagement on the same code by the developers reduced from 20 to 5. This was quite a remarkable development.

## 5.3 Client Satisfaction

Client satisfaction has been understood through a survey of questionnaires. The questionnaire asked some questions on the following items

- requirement
- product
- communication
- training
- overall assessment

It was found that the overall perceived satisfaction was not up to standard. Clients were somewhat dissatisfied with the providers in the above-stated cases. Clients were initially disturbed for providing repetitive requirement specifications due to poor data collection approach. Besides, the lack of proper review and coordination, resulting in missing deadlines generated a certain level of mistrust among the customers. Unorganized support maintenance service created irritation among the clients resulting in dissatisfaction. As the few updates in the process have been implemented mentioned above, a certain level of satisfaction among the customers or users was achieved. The table shows that client satisfaction grew from .6 to .8.

These numbers are projected as (0 – not evaluated, .2 – Low, .4 – below average, .6 – average, .8 – good, 1 – excellent)

## 5.4 Process Maturity

The 3D model has already been defined earlier. It created a process where the 3D refers to more specific Design, Development, and Deployment. On top of that, the upper-level activity of Observation, Management, and Documentation allowed establishing a more mature process. In this project, it is found that a mature process requires the maturity of the stakeholders as well.

### 5.4.1 Maturity in Behavior

The blame game for missing deadlines was a common issue in every task. It has been tackled by introducing the updated history of all types of events. It has been shown that both the parties failed in providing proper data or output. If the users fail to get the expected result, they get frustrated. We understood that process maturity just does not mean keeping deadline or a fault-free or a bug-free software. Maturity is beyond that. A module may fail to meet the deadline and may require 3 to 4 times more time. However, a responsible and reasonable explanation causing the delay gives hope to the client without making them disturbed and irritated. Maturity is not a concern only with the client or user.

### 5.4.2 Knowledge State of Team Members

We also identified that process maturity also requires to improve the knowledge state among the team members. Knowledge must be categorically shared among the developers, designers, and deploy representatives in a collaborative and coordinated way. Earlier, before the implementation of the 3D model, collaboration, and cooperation among the team members of both the developer side and client-side was a nightmare. However, encouraging proper knowledge sharing attitude and a respectable atmosphere reduced the gap of collaboration and cooperation in terms of knowledge transfer. Having developed proper knowledge before doing any sorts of activities ensures appropriate output for that part.

Maturity in the process was established simply by introducing the following practices

1. More Email communication with the users has been performed. As soon as an issue was raised during development or other operation, email was sent to the users with a copy to respective management.

2. Prompt reply of emails from users was made. Earlier there were delays in replying to emails.

3. Increased online communication among the team members of a project was ensured. Not only sharing of knowledge but also sharing of work status on a regular basis, distribution of tasks by informing all members of the team over communication medium was a great relief to synchronize the development activity and at the same time to keep the team on the same platform.

4. Sharing of all types of client information through a communication medium to all the team members brought all of them on the same page.

5. A group meeting during the time of crisis helped all the team members act accordingly were in an earlier situation it was difficult to motivate a team member to solve a certain scenario. If a single person is considered responsible, he/she feels alone and sometimes tries to escape the situation. In these cases, through group communication, an immediate small team was formed to solve the crisis.

It was kind of a perceptive measure of process maturity with an aim to smoothen the Implementation of the 3D process. Increased and effective communication made the process more matured and we found that the process maturity developed to .4 from .2. This number was achieved just from a simple interview with the users and the development teams. These numbers are projected as (0 – not evaluated, .2 – Low, .4 – below average, .6 – average, .8 – good, 1 – excellent)

## 5.5 Summary of Survey Results with 3D Model

Upon using the above-mentioned approach in a project, we found some improvement in the 4 parameters stated above, namely developer satisfaction, rework level, client satisfaction, and company's process maturity. All the surveys/interviews for each parameter were performed 2 times. One survey was performed while the teams were just following the traditional process. The next survey was performed after the implementation of the 3D model. In the case of every parameter, we found some improvement in the satisfaction levels or maturity levels or rework levels. The following table combines the values found. The levels of the values are 0 – not evaluated, .2 – Low, .4 – below average, .6 – average, .8 – good, 1 – excellent.

| Parameter | The value found before the application of the approach | The value found after the application of the approach |
|---|---|---|
| Developer Satisfaction | 0.46 (Below Average) | 0.63 (Average) |
| Rework Level | 20 (approx.) | 5 (approx.) |
| Client Satisfaction | 0.60 (Average) | 0.80 (Good) |
| Company's process maturity | 0.20 (Low) | 0.40 (Below Average) |

Table 2: Improvement of the 4(four) Parameters After the Use of the 3D Model

## 5.6 Observation, Management, and Documentation

In this article, we will explain the role of Observation, Management, and documentation in the whole process. These activities came to LIVE when a situation arose where it was felt that the project is struggling. As mentioned earlier, these operations have been created with the idea to improve the software development process by seeing it as a whole picture rather fix the problem in a localized area. The findings related to the process improvement activities are already defined in the above chapters. A summary is given below.

Initially, the project was built in a short time constraint and every module and component was developed on an ad hoc basis. Some of the preconditional tasks were not performed in time. As a result, the dependent tasks or activities were creating a lot of bugs and errors producing a wrong result and serious dissatisfaction among the users. The project was almost on the verge of failure. The top management of the user/client side sat several times with the software service provider and the project was almost on the verge of collapse. The project at that moment was on LIVE and there were lots of inconsistencies in data and operation. The developers were asked to fix all the errors that were popping up and the level of untrusted on service providers, especially to handle and level the situation, was growing. In such a scenario, we opted to apply the 3D Model to come out of this situation. We took refuge using the observation phase; a specific role was created as a monitor cum manager. The monitor's job was to review the Design, Develop, Deploy process as mentioned in the 3D model. There was another role created as a business monitor (works at the ground level). Monitors found that issues raised by the clients are repeating again and again even after fixing anomalies. The monitor verified the code, the business flow, and found that some serious issues remain with the implementation of business flow. The issues that were found during the observation and the way things were solved is given below

1. While there was anarchy in the implementation of the software project, the monitors arranged several meetings with the top management of the clients and the top management of the software development team. They found that there was a lack of understanding and a gap in knowing the real fact of the software and its effects. There were many claims from many quarters of the users, of which some were baseless.

2. Monitors also found that some departments of the user side were not cooperating to implement the software to make it successful.

3. The serious gaps monitor found that the most important activity which is the precondition of all the projects has not been confirmed before the implementation of the project. It created a lot of errors in the following activities. As soon they discovered such wrong implementation, immediately they moved the focus of the development team to put their all efforts in completing the preconditioned activity and thereby reduced the error of dependent activities.

4. The monitors found another issue. Due to hasty development, requirements were collected mostly verbally from different stakeholders, and naturally, most of them were incomplete and incorrect that led to chaos. Here also the monitors implemented a semi-systematic approach to collect information and transmit to the development team.

A brief description of the responsibilities of the 2 types of monitors are given below:

### 5.6.1 Business Monitor

This monitor observed all the bugs in the business process, analyzed the issues, and designed the perfect solution for implementation. The solutions were of 2 types.

1. It was found that the existing business flow is not suitable for the users' expected requirements. So, a new flow was designed and got approved by the user, which were then implemented by the development team.

2. In cases, where there was no change in the business flow required, this monitor, in cooperation with the development monitor participated in meetings with the development team and directed the development team to change the code according to the business flow.

### 5.6.2 Development Monitor

This monitor mostly worked in-house at the development place. His job was to inspect the algorithm and coding pattern and then instruct the developer to modify the code accordingly to inhibit future errors.

# Chapter 6

## Discussion and Conclusion

We implemented some of the features of the 3D model into the case study. Still, we experienced some improvement in the output. It is understood that a team working in an unstructured way will find it hard to adopt a new approach to a systematic process. So, our goal was not to put a rigid order in the development process. The team we worked with was not used to follow a very formal process of development. We also did not put them in pressure to follow the 3D model. Rather the implementation of this model was very flexible.

After the implementation of the model, we found that developers have gained incremental satisfaction through this approach. Previously, their satisfaction level was 0.46 (Below Average) which later increased to 0.63 (Average). One of the reasons for this achievement was to reduce the rework level. Rework level has dropped from 20 (approx.) repetitions to 5 (approx.) repetitions only. We also found that the progress of software development largely depends on the mental state of the client. It is the responsibility of the companies to give them peace. Through the implementation of this process, success has been achieved in this area as well. As an outcome, the client satisfaction level has been raised from 0.60 (Average) to 0.80 (Good). The final and most important outcome related to a more mature process was realized through this model. The company has increased their process maturity from 0.20 (Low) to 0.40 (Below Average).

In comparison to the different approaches of the software engineering life cycle, our 3D model reduces the number of steps of the software development life cycle. Compared to 5 steps of Waterfall model – Requirement, Design, Construction, Testing, Release; 4 phases of RUP approach – Inception, Elaboration, Construction, Transition; or 4 segments of Spiral model – Objectives, Risk Resolution, Develop, Plan; our model has only 3 steps, namely DESIGN, DEVELOP, and DEPLOY.

As a result, many time-consuming actions have been reduced and from specification to user acceptance, time spent was also reduced by a significant amount.

A comparison with some SPI models is shown in table 3 below. Another comparison with Waterfall, Spiral, and Incremental Model is shown in table 4.

## 6.1 Comparative Study with Software Process Improvement (SPI) Model

| Model | Characteristics |
|---|---|
| CMMI | A complex model works in parallel with the process model to improve Process |
| Other SPI approaches | Assessment setup is done, then SPI planning is done, afterward SPI implementation is done [8]. |
| 3D approach | SPI is not a separate activity. It is integrated with the Process Model and continuously adjusting the development process. |

Table 3: A Comparative Study with CMMI and other SPI Approaches

The 3D methodology is effective, while we used the 3 functions of the outer circle to be simultaneously implemented, namely, Observation, Management, and Documentation showed in Fig.1. The observation function was focused to observe the overall system with a critical analysis and continuous suggestions to the middle management or monitor to introduce changes in various activities of the 3D process. It is not mandatory that every use case of the development will follow strict cycle times of all the primary and secondary circles. It is the management decision whether for each scenario we should go through the steps of all primary and secondary cycles. The management, by observation, can dynamically decide to skip some secondary cycles while puts emphasis on other secondary cycles. In this way, the total cost of implementation of the project is reduced. The practice here was to introduce change management based on the observation. Finally, a documentation of the whole process is emphasized on other secondary cycles. In this way, the total cost of implementation of the project is reduced.

## 6.2 A Comparative Study of the 3D Model with Other Models

| Model/<br>Feature | Waterfall | Spiral | Incremental/<br>Iterative | 3D |
|---|---|---|---|---|
| Specification of All the Requirements in the beginning | Yes | Not all and Frequently Changed | Not all and Frequently Changed | Not all and Frequently Changed |
| Long term project | Inappropriate | Appropriate | Appropriate | Appropriate |
| Complex Project | Inappropriate | Appropriate | Appropriate | Appropriate |
| Frequently Changed Requirements | Inappropriate | Appropriate | Appropriate | Appropriate |
| Cost | Not costly | Costly | Costly | Not costly |
| Cost estimation | Easy to estimate | Difficult | Difficult | Medium |
| Flexibility | Not | Less flexible | Flexible | Flexible |
| Simplicity | Simple | Intermediate | Intermediate | Simple |
| Supporting high-risk projects | Inappropriate | Appropriate | Appropriate | Appropriate |
| Guarantee of Success | Less | High | High | High |
| Customer Involvement | Low | Low, After Each Iteration | High, After Each Iteration | High, After Each Iteration |
| Testing | Late | At the end of each phase | After every Iteration | After every Iteration |
| Maintenance | Least maintainable | Yes | Maintainable | Maintainable |
| Ease of Implementation | Easy | Complex | Easy | Easy |

Table 4: A Comparative Study of the 3D Model with Waterfall, Spiral, Incremental Approach

## 6.3 Flow Chart

The following flow chart in Fig. 4 shows the upper 3 activities namely Observation, Management, and Documentation. We can see that when an issue is found, it might get solved in one path. However, if similar kind of issues starts recurring, then the alternate path is activated. Here, at the Observation phase, the root cause of this issue is found. If it is found in the code, management takes steps to correct the code in a different way. If it is found that certain changes in the business process are required, Management takes certain action to change the business process to keep it aligned logically with the software system. Finally, documentation is made on the whole activity.

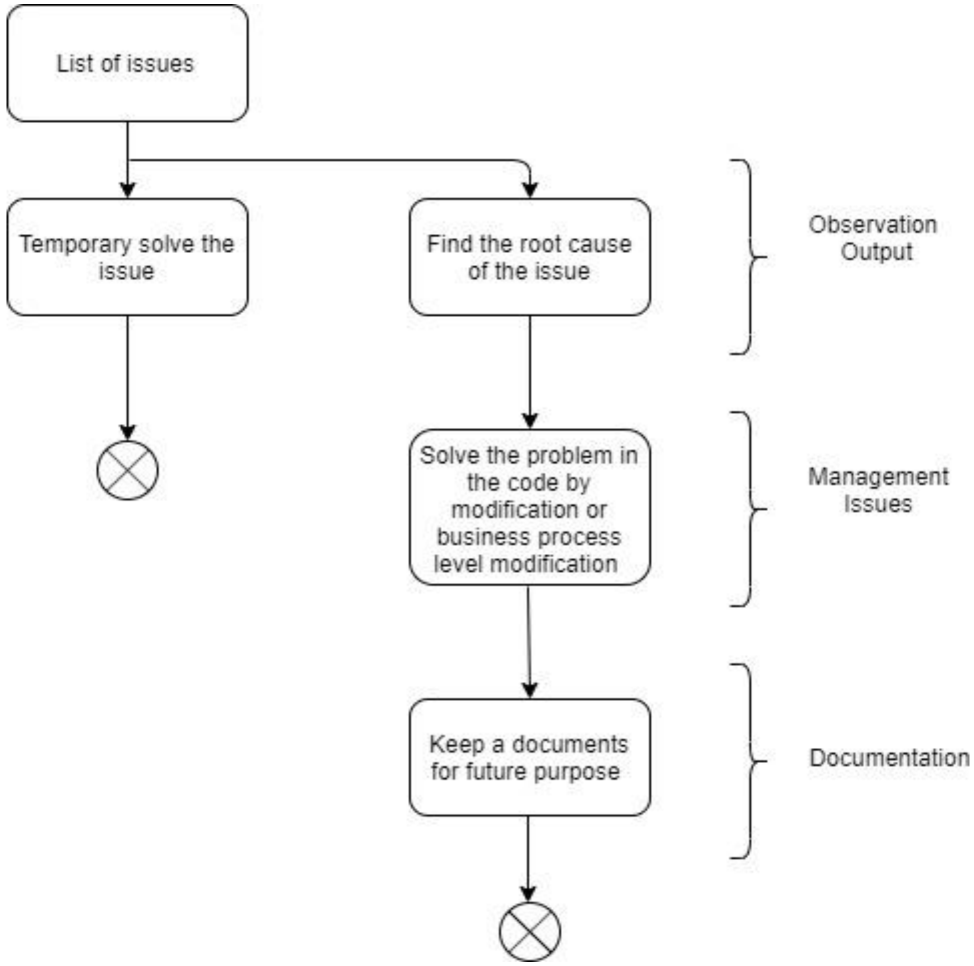

List of issues
Temporary solve the issue
Find the root cause of the issue
Observation Output
Solve the problem in the code by modification or business process level modification
Management Issues
Keep a documents for future purpose
Documentation

Figure 4: Use of Observation, Management, and Documentation to Improve the Product

# Chapter 7

## Future Work

A more detailed analysis of this model needs to be done considering more observed parameters from the developer side, client-side, and management side.

# References

[1] Laporte, C.Y., O’Connor, R.V., Fanmuy, G., “International Systems and Software Engineering Standards for Very Small Entities,” CrossTalk - The Journal of Defense Software Engineering Vol. 26(3), pp. 28–33, 2013.

[2] N. Habra, A. Renault, S. Alexandre, and M. Lopez, "OWPL Micro Assessment," in Software Quality Workshop, 24rd International Conference on Software Engineering ICSE, Orlando, Florida USA, 2002.

[3] Alexandre, S., Renault, A., and Habra, N., “OWPL: A Gradual Approach for Software Process Improvement in SMEs,” Proceedings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO- SEAA 2006), pp. 328–335, 2006.

[4] Jean Carlo R. Hauck, Christiane Gresse von Wangenheim, Richard H. de Souza1, and Marcello Thiry, “Process Reference Guides – Support for Improving Software Processes in Alignment with Reference Models and Standards,” Springer. EuroSPI, CCIS, vol. 16, pp. 70–81Heidelberg, 2008.

[5] Ita Richardson, “Software Process Matrix: A Small Company SPI Model,” Software Process: Improvement and Practice, Vol. 6, pp. 157–165, September 2001.

[6] Calvo-Manzano, J.A., Cervera, J., and San Feliu, T., “Software process improvement: MESOPYME model,” Journal of Computing and Information Technology (CIT), vol. 43: 159-165, 1997.

[7] Alshamrani A and Bahattab A, “A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model” International Journal of Computer Science Issues (IJCSI). 12, 106, 2015

[8] Deepti Mishra and Alok Mishra, “Software process improvement in SMEs: A comparative view” Computer Science and Information Systems · June (2009), 658.5:004.4, 10.2298/CSIS0901111M, 2009

[9] YungYu Lin, Yukari Nagai, TzuHang Chiang and HuaKo Chiang, “Design and Develop Artifact for Integrating with ERP and ECS Based on Design Science” ICISS 2020, March 19–22, Cambridge, United Kingdom, ACM ISBN 978-1-4503-7725-6/20/03, 2020

# Appendix A

The Process of Micro Companies Survey:

The micro-companies survey information is divided into nine categories (presenting only the most important questions), as follows:

1. Information about the Company
    - Use of Software life cycle
    - Causes of hampering their business goal
2. Employee information
    - Number of employees
    - Multitasking
    - Over timing
3. Proposal and requirement analysis
    - Analyzing cost, risk, resource and estimated delivery time
    - Time spending to collect requirements
    - Causes of changing requirements frequently by clients
4. Software project design
    - Using the framework
5. Development process
    - Maintaining the process for development
    - Time of development
    - Reason for not using a popular or conventional SDLC
6. Deployment
    - Deployment team (without developers)
    - The training process for trainer
7. Software testing
    - Software testing team and No.
    - Number of errors after delivery

8. Service and support
    - Support team
9. Client
    - Follow-up clients after completing the project
    - The unexpected situation from clients

# Survey about the process of Micro Enterprise

**Company Name:**

**Contact Person:**

## Information about the Company

When was the firm founded?
Answer:

Do you have the BASIS registration for this firm?

□ Yes □ No □ Processing

How many employees do you have in your firm?

□ Less than 5 □ Less than 10 □ Less than 20 □More

How many Full-time employees do you have in your firm?

□ Less than 5 □ Less than 10 □ Less than 20 □More

Do you have any branches of this firm?

□ Yes □ No

What type of software do you build in this firm?
□ Game □ Application □ Other □ User Less Than 1,000 □ User Less Than 10,000 □ More
□ Small □ Medium □ Large □ All type

Recently, is your company in profit or loss?

□ Profit □ Loss □ On breakeven

Do you use any common software life cycle? Or do you have any private software life cycle?

□ Yes □ Private □ No

In which stages do you maintain the documentation and how?

□ Requirement □ Development □ Testing □ Deployment □ Support □ Training □ Service □ None

Have you ever been failed to develop a project? Why?
□ Yes □ No
Answer: (If yes)

Which mistake of any stage does increase the cost of the project?
Answer:

What are the things you may consider that is hampering your profit or business goal?
Answer:

## Employee information

How do you appoint new employees?

□ Written Exam □ Viva □ Practical Test □ Reference □ Other

Does any person have to work on any other responsibilities that she or he has not been informed prior to joining or anyone have to do any other duties besides his own post specific duties?
□ Yes □ No

What steps you may consider improving your current working environment?
Answer:

Do the employees work overtime?

□ Yes □ No

Do the employees have any training system to learn new things?

□ Yes □ No

How do you measure the skills and improvement of your employees?

□ Practical Test □ Official Task □ Certification □ Other

## Proposal and Requirement Analysis

| |
|---|
| How do you approach your clients and initiate the development process?<br>□ Proposal submission □ Requirement analysis □ Depend on the situation |
| How do you calculate the cost of a project?<br>Answer: |
| How do you keep, store, manage, archive your requirements, and other project documents?<br>Answer: |
| How do you get requirements from clients (Prepared by them)?<br>□ Full documentation □ Orally □ Other ways □ Never |
| Approximately, how many days do you need to collect the whole requirement of Module or Project?<br>□ Less than 7 days □ Less Than 15 days □ Less Than 30 days □ Less than 90 days □ More |
| Before starting a new project, how you prefer analyzing cost, risk, resource, and estimated delivery date?<br>□ Software-based calculation □ Manual analysis on requirements and resources □ Other ways |
| How do you calculate the software delivery date?<br>□ Depend on Team □ Depend on All Resource □ Depend on Client's expectation |
| What are your conditions for the clients who have added some requirements after starting the work of a project?<br>□ Any penalty □ None<br>Answer: (What type penalty) |
| what are your conditions for the clients who have added some unexpected requirements after ending the work of a project?<br>□ Any penalty □ None<br>Answer: (What type penalty) |
| If the client makes a mistake by giving the requirement or if you make any mistake taking the requirement, what should be your decision?<br>Answer: |
| What do you think about why clients frequently change their requirements?<br>□ Lack of their Domain Knowledge □ Lack of your Domain Knowledge<br>□ For any mistakes on their side □ For any mistakes on your site □ Other |
| Why do you sometimes fail to collect proper requirements from clients?<br>□ Lack of their Domain Knowledge □ Lack of your Domain Knowledge<br>□ For any mistakes on their side □ For any mistakes on your site □ Other |

## Software project design

Before starting any new project, do you do a business analysis?

□ Yes □ No

What are the stages of creating a project architecture?
Answer:

For creating a project architecture do you use any framework?

□ Yes □ No

What type of problems do you face to create a project architecture?
Answer:

Do you use any software to manage the project?

□ Yes □ No

## Development process

What do you prefer to use for development tools?
Answer:

Do you follow any model or process for development?

□ Yes □ No

How do you distribute your works?
Answer:

How do you observe the improvement of the development process?
Answer:

How do you solve the difficulties of the development process?
Answer:

How do you decide the time of the development process?

□ Depend on Team □ Depend on All Resource □ Depend on Client's Desire

How do the developers improve their skills?

□ Learn from development □ Internal Office training □ Training from Institute □ Other ways

During development, if the client changes the existing requirement, how do you solve it?

□ Making extra charge □ Other ways (penalty). □ None

What are some of the reasons for not using a popular or conventional SDLC?
Answer:

## Deployment

How do you develop software (Life cycle)?
Answer **(Ex:** Requirement>Analysis>Project Architecture>Design>Coding>….**):**

Do you show them any demo before the software delivery?

□ Yes □ No

Do you have any deployment team or developers do the software deployment?

□ Yes □ No

What do you do if you fail the deployment?
Answer:

From your side or on behalf, who does lead the training process of software for clients?

□ Trainer □ Developer □ QA □ RA □ Other

If there is any other team for software training, how are they learning the training process?

□ Official Guide □ Own self □ Other

Have you ever been failed to deploy software? Why?
□ Yes □ No
Answer (If yes):

## Software testing

How important is software testing for you?

□ Recommended □ Important □ Not Important □ Useless

Do you have any other team for software testing? And how many people are involved in it?
□ Yes □ No
□ Less than 3 □ Less than 8 □ Less Than 15 □ More

Do the members of the testing team have knowledge of development?

□ Yes □ No

If you do not have a testing team, who does the software testing?
Answer:

What testing do you perform for software development?
Answer:

As well as the development, how much time do you fix for the testing?

□ One-fourth □ One-third □ Half □ Other □ None

What type of testing do you do?

□ Manual □ Automation

On average how many bugs usually your client report with in the first month of deployment?

□ Less than 10 □ Less than 100 □ More than 100

## Service & Support

Do you have any other team for support?

□ Yes □ No

Do you have any specific rules for support?

□ Yes □ No

How much success are you for giving instant support?

□ Satisfactory □ Average □ Not satisfactory

How much information on the project they have who work in the Support team?
Answer:

Do you control the software version?

□ Yes □ No

Recently, how successful you are for client service?

Answer:

Have you ever failed to provide client software support? Why?

□ Yes □ No

Answer (If yes):

## Client

Do you prepare yourself before meeting clients? And how?

□ Yes □ No

Answer (If yes):

During the project, how do you follow-up with your clients?

□ By visiting □ By phone calling □ Other ways

After completing the project, do you follow-up with your clients by yourself?

□ Yes □ No

What stages do you prefer for increasing the knowledge of the software domain of the clients?

Answer:

Do you prioritize your clients? And how.

□ Yes □ No

If a client wants immediate change in requirements or added features how do you respond to it?

Answer:

Most of the time, what type of unexpected situation you face from clients?

Answer:

Have you ever failed to maintain regular contact with a client? Why?

□ Yes □ No

Answer (If yes):

□□□ Write a summary of your company's current situation.

# Client Feedback Survey

| Company/Institution Name |
|---|
|  |

| Contact Person |
|---|
|  |

## Requirements:

How long did the analyst get to collect basic requirements?

☐ 5-10 days ☐ 10-20 days

☐ 20-30 days ☐ More

How long did you take to provide basic requirements?

☐ 5-10 days ☐ 10-20 days

☐ 20-30 days ☐ More

**How many people did come to understand and receive all requirements?**

☐ 1-2 ☐ 3-4 ☐ 5-6 ☐ More

Have you got any pre-plan copy of your requirements from us on which we will develop?

☐ Yes | ☐ No

How many times did you make changes in requirements during development?

☐ 1-2 times ☐ 2-5 times

☐ 5-10 times ☐ More

How many times did you make changes in requirements after development?

☐ 1-2 times ☐ 2-5 times

☐ 5-10 times ☐ More

## Product Assessment:

Was the job done accurately?

☐ Poor ☐ Fair ☐ Average ☐ Good ☐ Excellent

How long did ERP take to develop?

☐ 1-3 months ☐ 3-6 months

☐ 6-9 months ☐ More

How would you rate the quality of our product?

☐ Poor ☐ Fair ☐ Average ☐ Good ☐ Excellent

Are you happy with this product price?

☐ Extremely likely ☐ Somewhat likely ☐ Neither likely nor unlikely ☐ Somewhat unlikely ☐ Extremely unlikely

How did you compare the price?

☐ Comparing with National Market ☐ Comparing with International Market ☐ Other sources

## Communication:

How often does Company's person try to contact you?

☐ Once every 2 months ☐ 1-2 times per month

☐ 3-5 times per month ☐ Other

How are the Company person's management and assistance?

☐ Poor ☐ Fair ☐ Average ☐ Good ☐ Excellent

Would you like to contact us?

□ Yes | □ No

## Support:

How is support a person's punctuality?

□ Poor □ Fair □ Average □ Good □ Excellent

How long do they take to solve or manage emergency cases (like bugs, errors)?

□ instantly □ Hours

□ Day □ More

What is your current expectation from us?

______________________________________________

______________________________________________

______________________________________________

______________________________________________

## Software Training:

What is your opinion about our training process?

□ Poor □ Fair □ Average □ Good □ Excellent

What type of training process will you expect from us?

□ Temporary □ Module Base

□ Constantly □ Other

## General Assessment:

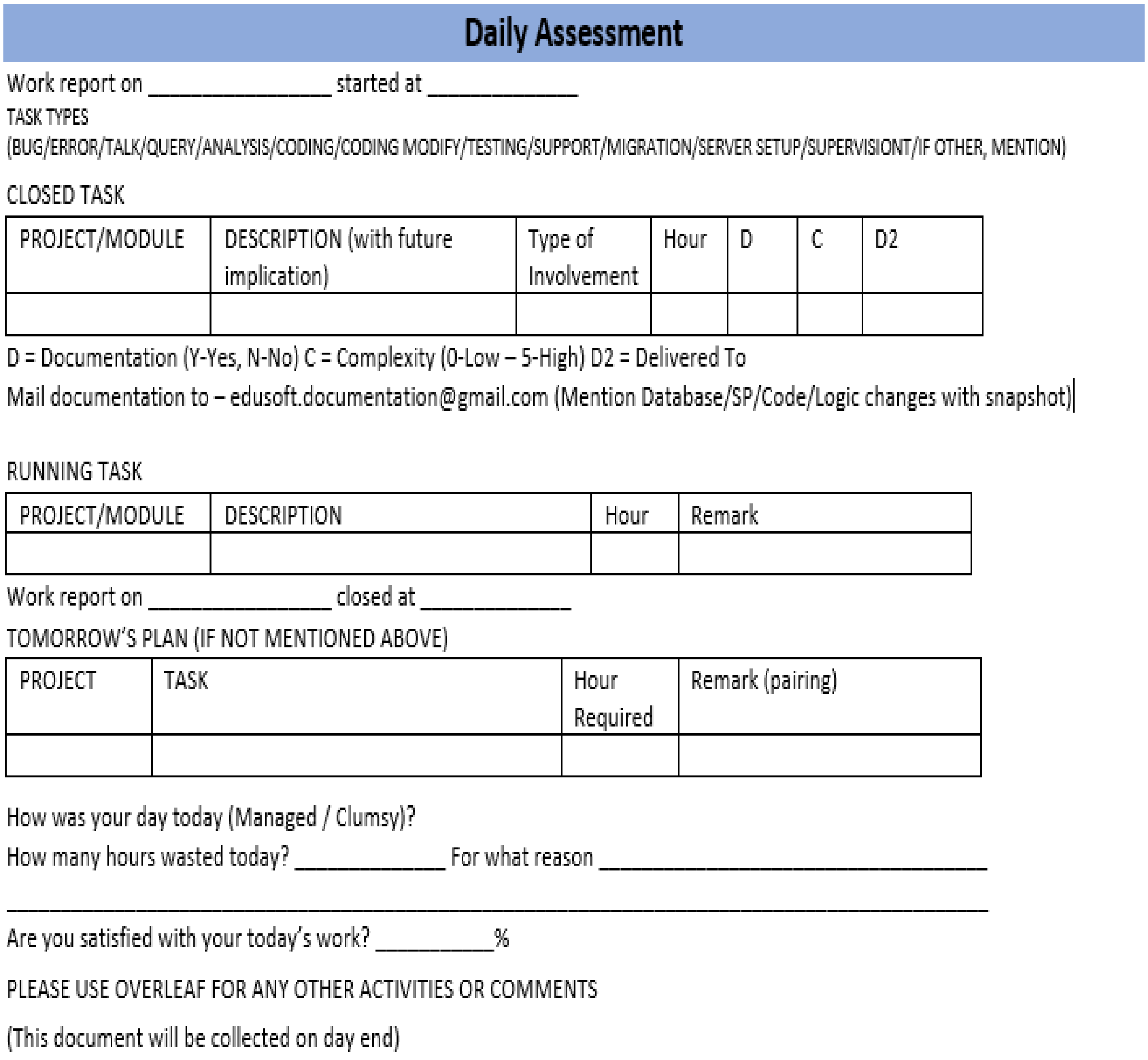
How will you rate the overall quality of our packing and moving?

□ Poor □ Fair □ Average □ Good □ Excellent

How would you like to recommend us to others?

☐ Yes | ☐ No

# Developer Daily Assessment Survey



## Daily Assessment

Work report on _______________ started at ____________

TASK TYPES

(BUG/ERROR/TALK/QUERY/ANALYSIS/CODING/CODING MODIFY/TESTING/SUPPORT/MIGRATION/SERVER SETUP/SUPERVISIONT/IF OTHER, MENTION)

CLOSED TASK

| PROJECT/MODULE | DESCRIPTION (with future implication) | Type of Involvement | Hour | D | C | D2 |
|---|---|---|---|---|---|---|
| | | | | | | |

D = Documentation (Y-Yes, N-No) C = Complexity (0-Low – 5-High) D2 = Delivered To

Mail documentation to – edusoft.documentation@gmail.com (Mention Database/SP/Code/Logic changes with snapshot)

RUNNING TASK

| PROJECT/MODULE | DESCRIPTION | Hour | Remark |
|---|---|---|---|
| | | | |

Work report on _______________ closed at ____________

TOMORROW'S PLAN (IF NOT MENTIONED ABOVE)

| PROJECT | TASK | Hour Required | Remark (pairing) |
|---|---|---|---|
| | | | |

How was your day today (Managed / Clumsy)?

How many hours wasted today? ____________ For what reason ________________________________

________________________________________________________________________________

Are you satisfied with your today's work? _________%

PLEASE USE OVERLEAF FOR ANY OTHER ACTIVITIES OR COMMENTS

(This document will be collected on day end)

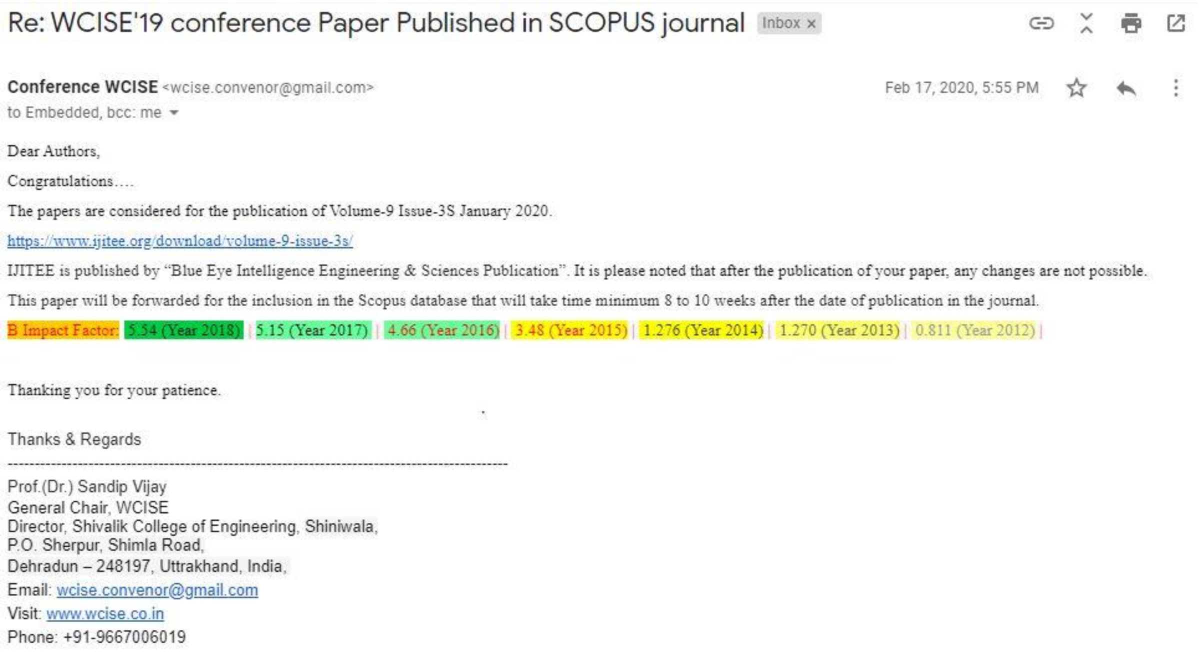
# Appendix B

# Journal Publication Proof

## Journal Web Link

https://www.ijitee.org/download/volume-9-issue-3s/

## Confirmation Mail



Re: WCISE'19 conference Paper Published in SCOPUS journal Inbox ×

**Conference WCISE** <wcise.convenor@gmail.com>
to Embedded, bcc: me

Feb 17, 2020, 5:55 PM

Dear Authors,

Congratulations….

The papers are considered for the publication of Volume-9 Issue-3S January 2020.

https://www.ijitee.org/download/volume-9-issue-3s/

IJITEE is published by "Blue Eye Intelligence Engineering & Sciences Publication". It is please noted that after the publication of your paper, any changes are not possible.

This paper will be forwarded for the inclusion in the Scopus database that will take time minimum 8 to 10 weeks after the date of publication in the journal.

B Impact Factor: 5.54 (Year 2018) | 5.15 (Year 2017) | 4.66 (Year 2016) | 3.48 (Year 2015) | 1.276 (Year 2014) | 1.270 (Year 2013) | 0.811 (Year 2012) |

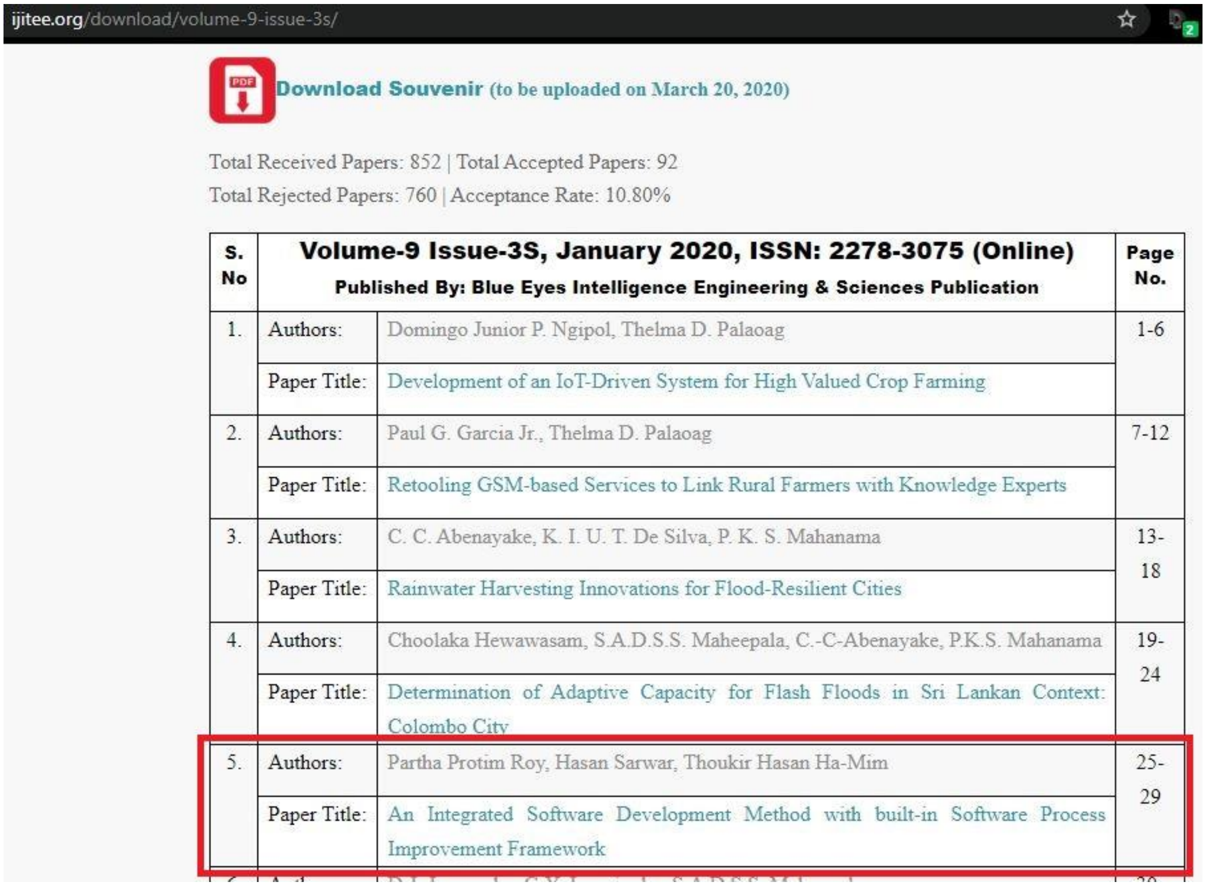
Thanking you for your patience.

Thanks & Regards

------------------------------------------------------------------------------------------------

Prof.(Dr.) Sandip Vijay
General Chair, WCISE
Director, Shivalik College of Engineering, Shiniwala,
P.O. Sherpur, Shimla Road,
Dehradun – 248197, Uttrakhand, India,
Email: wcise.convenor@gmail.com
Visit: www.wcise.co.in
Phone: +91-9667006019

## Journal Publication



ijitee.org/download/volume-9-issue-3s/

Download Souvenir (to be uploaded on March 20, 2020)

Total Received Papers: 852 | Total Accepted Papers: 92
Total Rejected Papers: 760 | Acceptance Rate: 10.80%

| S. No | Volume-9 Issue-3S, January 2020, ISSN: 2278-3075 (Online)<br>Published By: Blue Eyes Intelligence Engineering & Sciences Publication | | Page No. |
|---|---|---|---|
| 1. | Authors: | Domingo Junior P. Ngipol, Thelma D. Palaoag | 1-6 |
| | Paper Title: | Development of an IoT-Driven System for High Valued Crop Farming | |
| 2. | Authors: | Paul G. Garcia Jr., Thelma D. Palaoag | 7-12 |
| | Paper Title: | Retooling GSM-based Services to Link Rural Farmers with Knowledge Experts | |
| 3. | Authors: | C. C. Abenayake, K. I. U. T. De Silva, P. K. S. Mahanama | 13-18 |
| | Paper Title: | Rainwater Harvesting Innovations for Flood-Resilient Cities | |
| 4. | Authors: | Choolaka Hewawasam, S.A.D.S.S. Maheepala, C.-C-Abenayake, P.K.S. Mahanama | 19-24 |
| | Paper Title: | Determination of Adaptive Capacity for Flash Floods in Sri Lankan Context: Colombo City | |
| 5. | Authors: | Partha Protim Roy, Hasan Sarwar, Thoukir Hasan Ha-Mim | 25-29 |
| | Paper Title: | An Integrated Software Development Method with built-in Software Process Improvement Framework | |