

A Metamorphosis Approach to Software Engineering

^[1] M.Rithvik, ^[2] Merugu Naresh Babu, ^[3] Ch.Ambedkhar, ^[4] Narala Sudhakar Reddy, ^[5] Puli Nageswara Rao
^[1, 2, 3, 4, 5] Assistant Professor, SRK INSTITUTE OF TECHNOLOGY

Abstract:- The most important industry that is going to play a key role now a days is the software industry because it's the only industry which have high return values .For any industry to have a good survival applications plays a key role because an industry growth can be estimated by its outcomes those are nothing but products and services. Software Engineering is the subject that deals with various methodologies in order to ensemble software. This Paper represents a metamorphosis approach to software because when we perform Engineering we also need to know about reengineering sometimes many interventions may come up with reengineering process .Here we represent such type of reengineering approach with the help of a demo application.

Keywords— Metamorphosis,Engineering,reengineering,emerging,trends.

I. INTRODUCTION

Software engineering is a field of engineering, for designing and writing programs for computers or other electronic devices. A software engineer, or programmer, writes software (or changes existing software) and compiles software using methods that improve it. Better quality software is easier to use. Better documentation helps other people understand and maintain it and add new features. Becoming a software engineer requires experience and practice writing code. Software engineering may be very difficult work. It may be repeated during the software release life cycle.

Software engineering can broadly be split into the following steps:

Requirements say what the software should do.

Software design is usually done on paper. It says what the different parts of the software are, and how they talk to each other.

After the design phase is done, each component (part) of the software is coded. Code is what tells the computer exactly what to do at each step.

Testing is done to see if the components meet the requirements and that the system as a whole meet the requirements.

Part or all of this process can be repeated if software bugs are found or new requirements are needed.

Software Development Life Cycle, SDLC for short, is a well-defined, structured sequence of stages in software engineering to develop the intended software product.

SDLC Activities

SDLC provides a series of steps to be followed to design and develop a software product efficiently. SDLC framework includes the following steps:



1) Communication

This is the first step where the user initiates the request for a desired software product. He contacts the service provider and tries to negotiate the terms. He submits his request to the service providing organization in writing

2) Requirement Gathering

This step onwards the software development team works to carry on the project. The team holds discussions with various stakeholders from problem domain and tries to bring out as much information as possible on their requirements.

The requirements are contemplated and segregated into user requirements, system requirements and functional requirements. The requirements are collected using a number of practices as given -

- studying the existing or obsolete system and software,
- conducting interviews of users and developers,
- referring to the database or
- collecting answers from the questionnaires.

3) Feasibility Study

After requirement gathering, the team comes up with a rough plan of software process. At this step the team analyzes if a software can be made to fulfill all requirements of the user and if there is any possibility of software being no more useful. It is found out, if the project is financially, practically and technologically feasible for the organization to take up. There are many algorithms available, which help the developers to conclude the feasibility of a software project.

4) System Analysis

At this step the developers decide a roadmap of their plan and try to bring up the best software model suitable for the project. System analysis includes Understanding of software product limitations, learning system related problems or changes to be done in existing systems beforehand, identifying and addressing the impact of project on organization and personnel etc. The project team analyzes the scope of the project and plans the schedule and resources accordingly.

5) Software Design

Next step is to bring down whole knowledge of requirements and analysis on the desk and design the software product. The inputs from users and information gathered in requirement gathering phase are the inputs of this step. The output of this step comes in the form of two designs; logical design and physical design. Engineers produce meta-data and data dictionaries, logical diagrams, data-flow diagrams and in some cases pseudo codes.

6) Coding

This step is also known as programming phase. The implementation of software design starts in terms of writing program code in the suitable programming language and developing error-free executable programs efficiently.

7) Testing

An estimate says that 50% of whole software development process should be tested. Errors may ruin the software from critical level to its own removal. Software testing is done while coding by the developers and thorough testing is conducted by testing experts at various levels of code such as module testing, program testing, product testing, in-

house testing and testing the product at user's end. Early discovery of errors and their remedy is the key to reliable software.

8) Integration

Software may need to be integrated with the libraries, databases and other program(s). This stage of SDLC is involved in the integration of software with outer world entities.

9) Implementation

This means installing the software on user machines. At times, software needs post-installation configurations at user end. Software is tested for portability and adaptability and integration related issues are solved during implementation.

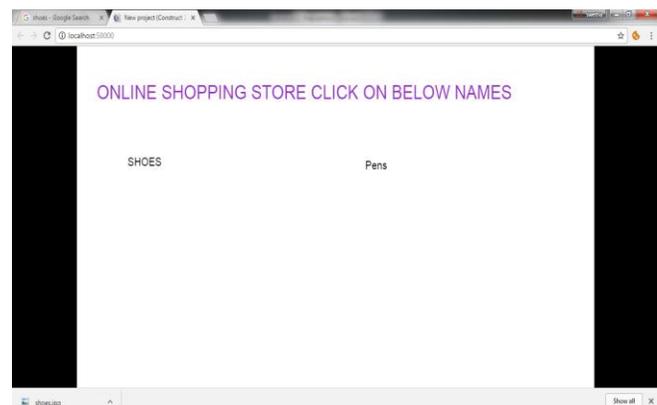
10) Operation and Maintenance

This phase confirms the software operation in terms of more efficiency and less errors. If required, the users are trained on, or aided with the documentation on how to operate the software and how to keep the software operational. The software is maintained timely by updating the code according to the changes taking place in user end environment or technology. This phase may face challenges from hidden bugs and real-world unidentified problems.

11) Disposition

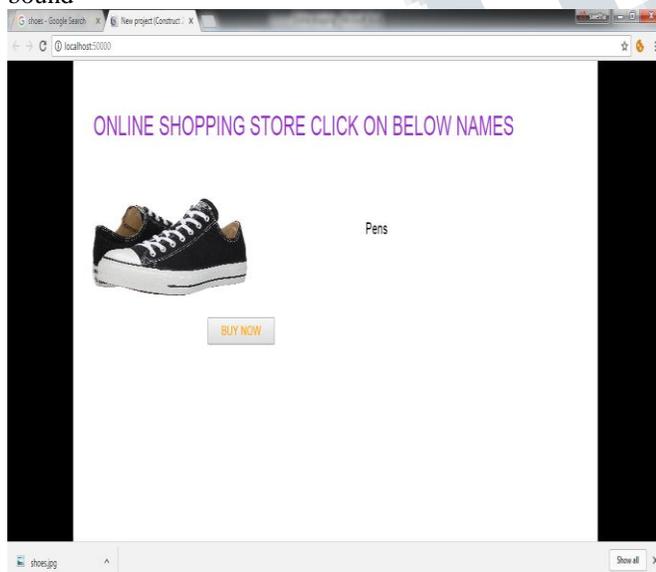
As time elapses, the software may decline on the performance front. It may go completely obsolete or may need intense upgradation. Hence a pressing need to eliminate a major portion of the system arises. This phase includes archiving data and required software components, closing down the system, planning disposition activity and terminating system at appropriate end-of-system time

II. EXPERIMENT CONDUCTED

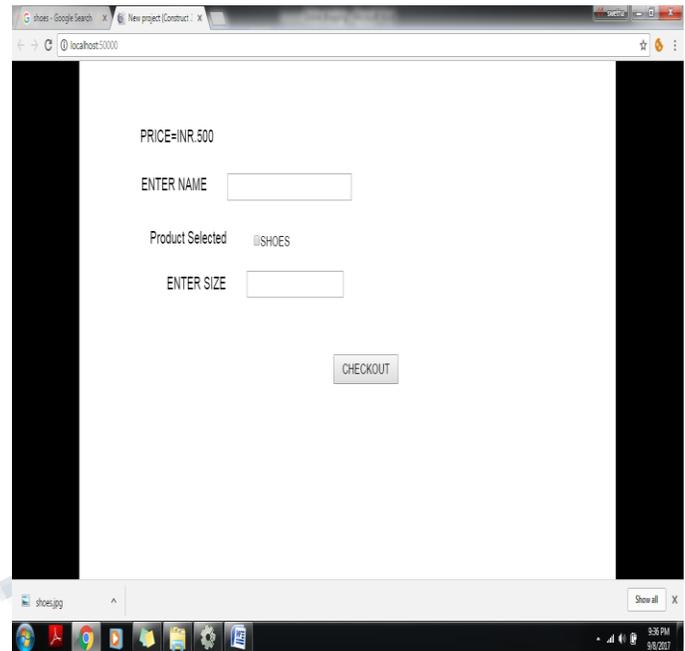


This is the sample application that shows the metamorphosis approach of a software. In order to develop an application we need to start with the SDLC as mentioned above strategy the

primary thing we need to do is requirement gathering and this is done only by identifying the current software scenarios. For Example let us consider the development of amazon or flipkart app we can know the product details but we cannot enhance or check the working condition of products. That means here We can identify a problem definition and this definition can be known to us only if we are willing to find out the current software applications which is nothing but we are going back we are identifying the requirements with the current issues in the present software that is already in use. So here we have identified a small issue in the Amazon or Flipkart app that is going to become our problem requirement. The requirement is by clicking on the good names relevant images must be displayed and that is presented in the above figure So the primary step of Software Development was done first because we have gathered the specific requirements and then the analysis phase must be started followed by the Waterfall Model because now a days waterfall model is going back because in order to reevaluate the latest tools and technologies new tools and new technological enhancements are evolving in the field of software thereby gaining our work to make lesser in the production time bound



By just clicking on the good relevant image can be displayed and this is the demo screen shot we can enhance this further by adding a video of the relevant good/product in order to identify the services and conditions of the good/service that is brought by the customer



The 3rd step in this scenario is a crucial step that is the customer selection which is nothing but the design process in our software Engineering and this can be enriched by using the latest technologies now a days. The Customer Enters the whole data of this selection and we can add the recent blockchain technology over here in order to make this faster. With a blockchain, many people can write entries into a record of information, and a community of users can control how the record of information is amended and updated. Likewise, Wikipedia entries are not the product of a single publisher. No one person controls the information. Descending to ground level, however, the differences that make blockchain technology unique become more clear. While both run on distributed networks (the internet), Wikipedia is built into the World Wide Web (WWW) using a client-server network model. A user (client) with permissions associated with its account is able to change Wikipedia entries stored on a centralized server. Blockchain's conceptual framework and underlying code is useful for a variety of financial processes because of the potential it has to give companies a secure, digital alternative to banking processes that are typically Bureaucratic. Here Comes our greatest challenge to integrate two technologies at one extent in the design process because blockchain technology is costliest technology and we need to design the code that is to be integrated with two technologies that is a great challenging work. Thereby we can understand that several challenges are going to get occurred while doing a design of a software. For the

customer to give a quality product various phases has to be undergone by the software developer and that is the challenge in the field of software



The Final Output suggests that the shopping of the customer is successful and this output gives us that the shopping of customer is successful.

III. CONCLUSION

We would like to suggest that rather than going to all the phases in software engineering it is easy to go for few phases. The 1st phase of requirements gathering is common after that move for the design phase then we can include the analysis phase. The coding phase of the software can also be made easy with the innovations of various tools like Construct2d and Scratch. We may have many more tools that require only the logic to build an application. The final step is integration of various technologies at a step to make an intervention analysis of solving a problem.

REFERENCES

- [1]. SOFTWARE ENGINEERING TUTORIAL
- [2]. Amazon App
- [3]. Blockchain Technology
- [4]. Construct2d
- [5]. Scratch.