SOFTWARE ENGINEERING & REQUIREMENTS

SIMRAN BAGWAN [1], PRAJAKTA GAWAND[2], DEEPIKA THAKUR[3], NEHA SALUNKHE[4], SHOBHANA GAIKWAD[5]

[1] *Student, Computer Technology, Bharati Vidyapeeth Institute of Technology, Maharashtra, India*

[2] *Student, Computer Technology, Bharati Vidyapeeth Institute of Technology, Maharashtra, India*

[3] *Student, Computer Technology, Bharati Vidyapeeth Institute of Technology, Maharashtra, India*

[4] *Student, Computer Technology, Bharati Vidyapeeth Institute of Technology, Maharashtra, India*

[5] *Professor, Computer Technology, Bharati Vidyapeeth Institute of Technology, Maharashtra, India*

ABSTRACT

Software Engineering is like a driver who drives the computer. In this topic, we will see how Software engineering work and also its requirements. The software is a mediator that communicates with computer hardware and users. All our day to day activities like banking transactions, shopping, bill payments, entertainments like playing games, movie etc is done by the software.

Keywords: *driver, mediator, hardware, users, technology*.

1. INTRODUCTION

The software is the data structures that enable the program to adequately manipulate information. The software has introduced the concept of artificial intelligence which is a booming technology in the market. many scientists are trying to develop artificial intelligence in machines like robot or computers. In short, software is an agent which produces the information in required form that could be audio, video, textual, graphical or even encrypted format.

As a product, it delivers the computing potential across a network of hardware.

The maintenance should be feasible for the software to involve to meet changing requirements. This is a software product has good reusability if the different modules of the product can easily be reused to develop new products.

2. LITERATURE REVIEW:

The Scientists Naur and Randell who coined the software engineering concept in 1969[1]. The meeting of many Computer Scientists and captains of industry convened to form a “software crisis” [2]. Although there are difficulties to find a road map to guide the industry towards well they agreed to believe that their arguments are on the firm ground or footing [3].

Since the term software engineering was coined some progress has been made toward the development of a profession of software engineering [4]. The body of Knowledge is particularly noteworthy, with the Trial Version of guide to the Software Engineering Body of Knowledge (SWEBOK) being issued in 1999 [5].

3. METHODOLOGY:

Software engineering is a layered technology which is based on quality. The quality refers to the software quality which is committed to the client by the software development organization. As the software engineering processes, methods and the tools used for assuring the quality management in the software. The process, methods and even the tools become standards due to the layered approach. Due to this development, software engineering process has become more and more mature. The different layers of software engineering are Quality focus, process, method, and tools.

The quality focus is a very important aspect of the software aspect of software engineering. The foundation layer of the layered approach is processed. There are specific processes which takes place while developing software. The process helps to produce a rational and integrated software product. The process is a software development framework, which defines “Key Process Area” (KPA).

The key process areas are the main areas which become the basis for management control of software projects**.** **Methods** are a collection of the tasks that entails requirements analysis, design, program construction, testing, and support. Software engineering processes are based on the basic principles of these methods.

Tools are the next layer which is a very important part of Software engineering which could be fully automated or computerized or tools provide automated or semi-automated which supports the process and the methods.

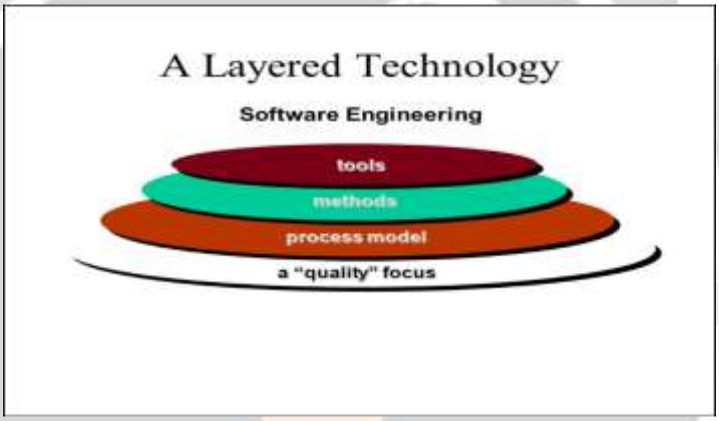

Fig 1.1 Software Engineering a layered approach

Types of Software:

1. **System software**
2. **Application software**
3. **Engineering/Scientific software**
4. **Embedded software**
5. **Product-line software**
6. **Web applications**
7. **Artificial intelligence software**

1. **System software:**
   System software is an important category of software which mainly deals with hardware and user. An operating system, compilers, interpreter, editors, file managers are important examples of system software.

2. **Application software:**
   Application software is developed to do particular business activities. These are standard programs. They are used for business process or business management. It also supports business with real-time support. E.g. Point- of- sale transaction processing and real-time manufacturing process control.

3. **Engineering/Scientific software:**
   The Engineering/Scientific software has made a great process. This software is into space shuttle orbital and dynamics and molecular biology etc. They range from astronomy and volcanology.

4. **Embedded software:**
   Embedded system resides within a product or system. This type of the software used to control features and function for system as well as users.

5. **Product- line software:**
   Product- line mainly emphasize on a limited and esoteric marketplace or they can even address mass consumer markets.

6. **Web applications:**
   Web applications are a wide array of applications. These are set of files which are linked with the hypertext.

7. **Artificial intelligence software:**
   The artificial intelligence software uses a non-numerical algorithm. They are used for complex and highly analytical problems.

Requirement Engineering Tasks:

1) **Inception**: Inception means beginning. It is usually said that 'well beginning is half done'. But it is always problematic for the developer that, 'from where to start'. The requirements engineering itself is a 'communication intensive' activity.

2) **Elicitation:** Elicitation means, to draw out the truth or reply from anybody. In relation to requirement engineering, elicitation is a risk that helps the customer to define what is required.

3) **Elaboration:** Elaboration means 'to work out in detail.' The information that is received the beginning during inception and elicitation phases are expanded and medications are made during elaboration.

4) **Negotiation:** Negotiation means a discussion on financial and other commercial issues.

5) **Specification:** The specification is the final work product produced by requirement engineer. The specification plays a role for all software engineering activities.

6) **Validation:** All previous work completed will be just useless & meaningless if it is not validated against the customer's expectations.

Positive Side of Software Engineering:

- You will be able to find a lane that suits you with minimal effort. It's not difficult to find a niche that lets you work harder and make more money or work less and enjoy more time off.
- Remote work opportunities are a HUGE quality of life advantage. This cannot be overstated.
- You will have lots of opportunities to use your skill set outside the normal workplace if you so choose. It's nice to have a skill people value.

Negative Side of Software Engineering:

- Even five hours of sitting on a chair peering at software can feel grueling, never mind eight or more.
- You are likely to work in a heavily male-dominant environment.
- Some of the job is going to be boring, and involve things like getting bugs out of reams of code.

Conclusion:

1) In this topic, we have looked at some key concept, themes, and skills related to software development.
2) This topic will allow you to develop fundamental Knowledge, understanding analysis and synthesis skills that you need to develop fit-for-purpose software in organizational context by taking a practice-based approach based on an organization you are familiar with.
3) This topic will also give you an opportunity to investigate emerging trends in software development and carry out some independent that interest you.

Reference:

[1] Timothy C. Lethbridge & Robert Laganiere.

Object-Oriented Software Engineering: Practical Software Development using UML and Java (Second Edition).

[2] M.R.V. Chaudron. J.F.Groote, K.M. van Hee, C. Hemerik, L.J.A.M. Somers and T. Verhoeff. "*Software Engineering Reference Framework*".

[3]Jessica Keyes. *Software Engineering Handbook*. Auerbach Publications (CRC Press), 2003.

[4]Roger S. Pressman. *Software Engineering: A Practioner's Approach (Sixth Edition, International Edition)*. McGraw-Hill, 2005.

[5]Karl E.Wiegers. *Software Requirements*. Microsoft Press: 2nd edition (February 26, 2003). 1st edition:1999.

IJARIIE