Journal of Basic and Applied Engineering Research
Print ISSN: 2350-0077; Online ISSN: 2350-0255; Volume 1, Number 10; October, 2014 pp. 28-43
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/jbaer.html

Advance Software Engineering and Software Metrics

Sumit Kumar Mishra

Deptt. of Computer Science, Babu Banarasi Das University (BBDU) Lucknow

Abstract: **Software engineering is important branch of computer science engineering. This paper provide a precise summery of a survey of advance software engineering and its metrics. Present day Software engineering is the most important technology with the rapid development. Software is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. The goal of software engineering is to provide models and processes that lead to production of well documented maintainable software in a manner that is predictable. The term software metrics designates that a unit of measurement of software product. In this paper on the basis of current researches on advance software engineering and preexisting software metrics is described. This paper includes description about system engineering, business process engineering and agile software engineering concepts also.**

Keywords: **Agile Menifesto, Agile Software Engineering, Business Process Engineering, Requirements Engineering, System Engineering, Web Engineering, Component Based Software Engineering Software Metrics.**

1. INTRODUCTION

1.1 SOFTWARE ENGINEERING

The goal of software engineering is to provide models and processes that lead to production of well documented maintainable software in a manner that is predictable.

The principle aim of software engineering is to adopt the systematic and organized approach for developing the software product. Software process is the part of software engineering. Software process deals with more technical and management issues.

SOFTWARE LIFE CYCLE MODEL

Software development life cycle(SDLC)model is a diagrammatic representation of various activities required to make a good software product. The software life cycle also specifies the order in which each phase of software must be executed.

2. REQUIREMENT ENGINEERING

Requirement engineering is a communication process between software customers users and software developers. It is not simply a technical process but it is influenced by users, dislikes political and organizational issues, in the software context Requirement engineering provides appropriate mechanism what the customer wants analyzing their needs .

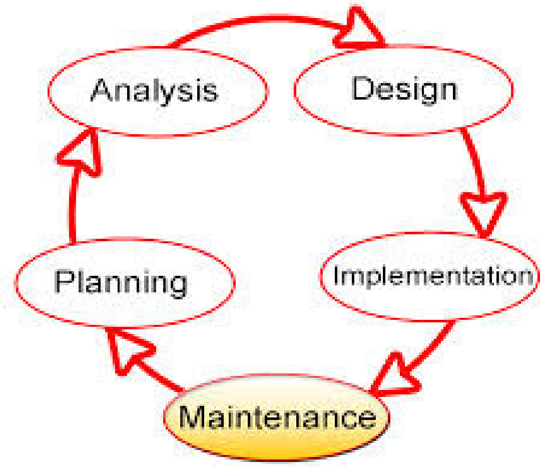


Fig. 1. Software Life Cycle Model

2.1 MISCONCEPTS OF REQUIREMENT ENGINEERING

- Any subject matter expert can become requirements engineer after the week or two of training.
- Process that work for small no. of requirements will scales.
- Nonfunctional and functional requirement can be elicited speared team and process.[1]

3. BUSINESS PROCESS ENGINEERING

The Goal is to define architecture that will enable a business to use information effectively.

Business process engineering is an approach for creating and overall plan for implementing the computing architecture (not only the specification of computing architecture but software architecture that populate the unique configuration of heterogeneous computing resources is required).

System engineering in context of business enterprises called the business process engineering.

3.1 SYSTEM ENGINEERING

It is collection of interrelated component working together towards some common objectives.

A system may include software, electronic hardware, mechanical and electrical component & its operated by people.

System engineering is concern is all aspect of computer based system development including hardware, software, process engineering and software engineering is a part of this process.

The main step involved in system engineering

Design → Implementation → Deployment → Operating the system

4. COMPONENT BASED SOFTWARE ENGINEERING(C.B.S.E.)

Component based software engineering emerged in the late 1990 as a reuse based approach to software systems development.

Component based software engineering is the process of define, implementing and integrating or composing loosely coupled independent component into systems. It is important for development approach because in current scenario software systems are becoming larger and more complex and customers are demanding more dependable software that is developed more quickly.

4.1 Component Based Software Engineering Process

The C.B.S.E. follows the two parallel engineering activities.

DOMAIN ENGINEERING

Domain engineering explores the application domain with specific intent of finding behavioral, functional and data component candidates for reuses.[2]

It provides three major activities-

I. Domain analysis phase
II. Domain construction phase
IIII. Domain dissemination

COMPONENT BASED DEVELOPMENT

It obtains requirements from the customers and select appropriate architectural style to be built.

It includes 4 major activities –

I. Component qualification
II. Component adoption
III. Component compesition
IV. Component update

5. WEB ENGINEERING

Web engineering systems and applications now deliver a complex array of functionality to a large number of diverse groups of users.

Web engineering is a most important way of developing and organizing knowledge about web application development and applying that knowledge to develop its applications. It also manages complexity of web applications.[3]

6. AGILE SOFTWARE ENGINEERING

Agile can be formally defined as an iterative and incremental approach to software development. Which is performed in a highly collaborative manner by self-organizing teams within an effective governance framework, with “Just enough” ceremony that produces high quality software in a cost effective and timely manner which meets the changing needs of stakeholders. Agile methodology are often seen as providing ways to avoid overheads typically perceived as being imposed by traditional software development environments.

A software process is defined as a set of methods, practices, activities and transformations that are used to acquire and affirm software and its related products [4].The appearance of agile methods has been the most noticeable change to software process development in the last fifteen years. Many reviews, studies, and surveys have been conducted on agile methods. [5].

6.1 Agile Menifesto

The manifesto reads as follows (Agile Alliance, 2001):

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value [6]

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes tackle change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.

6.2 WHAT IS AGILE?

The Agile method has the highest priority of satisfying customer needs through early and continuous software development. The main focus of agile is it welcomes the changing requirements even late the development .The delivery of the software is very fast possibly with the much shorter time scales.

Agile software is not only meeting with the customer requirement but it is meeting the changing requirement right up to the level of product development.

Alistair Cockburn is one of the initiators of the agile movement in software development, he defines agile as "agile implies being effective and maneuverable. An agile process is both light and sufficient. The lightness is a mean of staying maneuverable. The sufficiency is a matter of staying in the game" [7]

Barry Boehm described agile methods as "an outgrowth of rapid prototyping and rapid development experience as well as the resurgence of a philosophy that programming is a craft rather than an industrial process" [8]

6.3 AGILE DEVEOPMENT METHODOLOGIES

In this paper we provide an overview of Some basic and main methodologies:

- Scrum
- Feature Driven Development
- Extreme Programming

6.3.1 SCRUM

Scrum is the approach to software development. Rather than a full process or methodology. It is a framework. Scrum is unique because it introduced the idea of "empirical process control."In scrum, projects are divided into succinct work cadences, Known as sprints, which are typically one week, two weeks, or three weeks in duration. At the end of each sprint, team members meet to assess the progress of a project and plan its next steps.

Some basic point is given below which helps to understand this methodology :

THE TEAM

1. Teams are self organizing.
2. Membership should change only between sprints.
3. The daily scrum meeting.
4. Helps to avoid other unnecessary meetings.
5. Team presents what it accomplished during the sprint.
6. Everyone answers 3 questions
   I. What will you do today?
   II. Is anything in your way?
   III. What did you do yesterday?

7. Whole team participates.

6.3.2 Features Driven Deveopment

The FDD process enforces rigorous guidelines in order to find defects in the system. It also enforces coding standards and encourages regular builds on a daily or weekly basis in order to add freshly designed features to the base system. Since the features are developed in parallel, it is mandatory to have a configuration management system that allows calmly integrating of the changes that are made to the system. In FDD approach there is a tracking and papering mechanism that is used to show the project status based on the number of features that have been implemented as well as the overall progress of the design, coding, and testing activities[9]

6.3.3 Extreme Programming

Software development a human activity and human decisions should be important. Extreme programming (XP) is one of the first agile processes that have been proposed. In general, XP consists of a set of individual practices that when put together yield a successful software Practice. Further the focus of XP is on the business aspect of a project resulting in increased productivity. There are 5 values of XP which are given below:

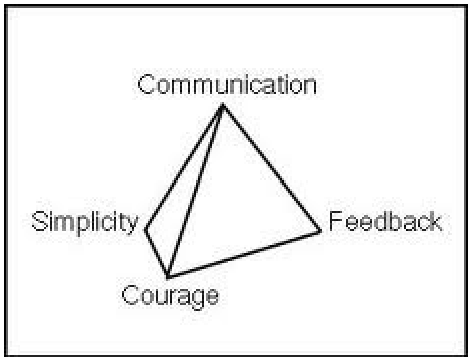
a. Communication
b. Respect
c. Courage
d. Value in time
e. Feedback

7.0 SOFTWARE COMPLEXITY

Software complexity has been shown to be one of the major contributing factors to the cost of developing and maintaining software.[10]

Software play an important role in ensuring the desired quality and have widely been applied to practical software project.

According to IEEE[11] defines software complexity as “the degree to which a system or component has a design or implementation that is difficult to understand and verify”.



Values of X.P.

According to **CURTIS[12]** he categorized the concept of software complexity into algorithmic and psychological complexity. Algorithmic complexity characterizes the run time performance of an algorithm.

Psychological complexity affects the performance of programmers trying to understand or modify a code module.

7. SOFTWARE METRICS

The term software metrics designates that a unit of measurement of software product. It is the degree to which a system, component, or process possesses a given attribute.

A. LINE OF CODE(L.O.C.)

A Line Of Code is any line of program text that is not a comment or blank line, regardless of the number of statements or fragments of the statements on the line. This was the first measurement attempted.

Advantages

a. Many existing methods use LOC as a key input.

b. A large body of literature and data based on LOC already exists.

Disadvantages

a. This measure is dependent upon the programming language.

b. It does not accommodate non procedural languages.

c. In early stage of development it is difficult to estimate LOC.

B. FUNCTION POINT

This is simplest techniques called Function Point analysis developed by ALAN ALBERCHT at IBM . In this techniques we measures functionality from the user points of view. The main function point principle is that a system is decomposed in functional units.

Advantages

Function point approach is independent of the language, tools, or methodologies used for implementation.

Disadvantages

a. This method is more suitable for business systems and can be development for the domain.

b. Many aspects of this method are not validated.

c. The function point has no significant meaning .it is just a numerical value.

C. DATA STRUCTURE METRICS

A Count of the amount of data input to processed in, and output from software is called a data structure metrics. There are various methods for measuring data structures.

I. The amount of data

II. The usage of data within module.

THE AMOUNT OF DATA

Most of compilers and assemblers have an option to generate a cross reference list, including the lien where a certain variable is declared or used. Such list is useful in debugging and maintenance and also can help to determine the amount of data in the program.

THE USAGE OF DATA WITH IN MODULE

In order to characterize the intra module data usage, we may use the metrics live variables and variable spans which are given below-

LIVE VARIABLE

A variable is live from the beginning of a procedure to the end of the procedure.

(or)

A variable is live from its first to its last references within a procedure.

VARIABLE SPANS

A Metric that captures some of the essence of how often a variable is used in a program is called span. This metric is the no. of statements between two successive references of the same variables.

Object-Oriented Metrics

Object-oriented software development requires a diverse approach from more traditional functional decomposition and dataflow development methods. But the OOAD software development life cycle is not easier than the typical procedural approach. Therefore, it is necessary to provide dependable guidelines that one may follow to help ensure good OO programming practices and write reliable code. Object-Oriented programming metrics is an aspect to be considered. Metrics should be a set of standards against which one can measure the effectiveness of Object-Oriented Analysis techniques in the design of a system. [13]

Merits and demerits of OO metrics are: [14]

1. The OO metrics come out to be able to differentiate simple from complex OO projects.
2. The OO metrics do not support studies outside of the OO paradigm.
3. The OO metrics have not yet been applied to testing.
4. The OO metrics have not yet been applied to maintenance.
5. The OO metrics have no conversion rules to lines of code metrics.
6. The OO metrics have no conversion rules to function point metrics.
7. The OO metrics lack automation.
8. The OO metrics are difficult to enumerate.
9. Software project estimation tools do not support the OO metrics.

8. CONCLUSION

Today, software engineering is the most important technology with rapid development. National economy is also dependent upon this technology.

In this paper we describe some most important topic of advance software engineering i.e. Web engineering, business process engineering, and agile methodology, component based software engineering, and software metrics. Software metrics research has helped to build up an empirical basis of software engineering.

This paper described a new type of model for software development i.e. agile software development. In this paper, we presented our analysis of agile software processes. We also describe what are the problems faced during implementation of agile software development. The objective is to help software engineers to understand the key characteristics of these processes and therefore select the most suitable process with respect to the type of software projects they develop.

REFREENCES

[1] Software & Systems Requirements Engineering In Practice.[BRIAN BERENBACH, DANIELJ.PAULISH, JUERGEN KAZMEIER, ARNOLD RUDORFER]

[2] Abhikrit Narwal “EMPIRICAL EVALUATION OF METRICS FOR COMPONENT BASED S/W SYSTEM”, International journal of latest research in science and technology ISSN(online)2278-5299, Vol 1, issue4:page no. 373-378, November-December(2012).

[3] Sun Murugenson, Athula Ginge “WEB ENGINEERING”.

[4] G.CUGOLA AND C.Ghezze;”Software processes a retrospective and a path to the future”, In Proc.of the Software Process Implement and Practice Conference, 1998, PP 101-123.

[5] Kieron Conboy and Sharan Coyle, ”People overprocess;Key Challenges in agile development “IEEE Software.

[6] Highsmith, ”Agile S/w development:the business of innovation”, IEEE Software 2001.

[7] Agile Documentation :A Pattern Guide to Producing Light Weight Documents for Software Projects.

[8] Agile Process in Software Engineering and Extreme Programming :9th International Conference, XP 2008, Linerick Irland, June 10-14, 2008;Proceedings.

[9] Qasaimeh M.;”Comparing Agile Software Processes Develpoment Project Requirements IEEE Computer Society, 2008.

[10] R.B. Grady, Practical, ”s/w metrics for project management & process Improvement”Hewlett Packard Professional Books, Prentce Hall, New Jersey, 1992.

[11] IEEE computer society “IEEE standard Glossary of software Engineering Terminology.”IEEE std.1990.

[12] Bill Curtis, ”Measurement and experimentation in software engineering” Proceeding of the IEEE vol.68 no.9pp 1144-1157, 1980.

[13] Software quality metrics for object oriented system Environments”, June 1995. National Aeronautics and space Administration.Goddard space flight center, Greenbelt maryland

[14] Capers jones, chief Scientist Emeritus, ”Strengths and Weaknesses of software metrics”, version 5, march 22, 2006.

ABOUT THE AUTHOR:

SUMIT KUMAR MISHRA received his B.Tech degree from G.B.T.U. in 2013. Currently he is pursuing M.Tech in Software engineering from Babu Banarasi Das University, Lucknow Uttar Pradesh, India.