iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** IV **Month of publication:** April 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 3 Issue IV, April 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

A Survey of Artificial Intelligence in Software Engineering

Rajdeep Kaur Aulakh
Department of Computer Science and Engineering

Abstract: Software engineering are the principles which are used in the development of the software. Software development is a very long, complex process and it's a human oriented activity. This paper discusses the concept of Artificial Intelligence and how it can be used in software engineering i.e. how artificial intelligence is related to software engineering? Various Artificial Intelligence techniques are discussed in this paper, which can be used to improve many of the software development activities, with which quality of the software can be increased.
Keywords: Software, Software development, Software Engineering, Artificial intelligence, artificial techniques

I. INTRODUCTION

When you have started reading this research paper, these three questions might have stricken your mind.What is Software engineering?What is Artificial Intelligence?How Artificial Intelligence is related to Software engineering?*Software Engineering* is the act of adopting engineering principles that can be used for efficient and effective development of high quality and mostly very large software system. The goal is to support software engineer and managers in order to develop better software faster with intelligent tools and method. In this act principles of analysis and synthesis observed. *Analysis* is the process of decomposing something into components or modules with a view to understand the individual component. *Synthesis*, the reverse of analysis, is the process of putting together of small building blocks to build a large structure. (Emrich M.L, et al, 1988).

Thus any problem solving techniques must have two parts:-

A. Analyzing the problem to determine its nature.
B. Synthesizing a solution based on the analysis.

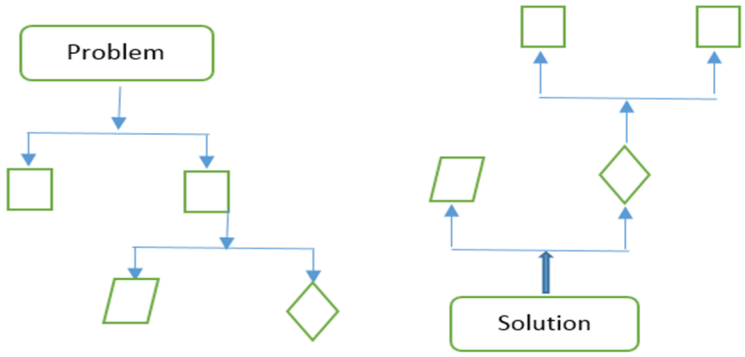


Figure :(1a)An analysis of the problem to determine.(1b) A synthesis of a solution based on the analysis.

Artificial intelligence is that field of computer usage which attempt to construct computational mechanism foractivities that are considered to require intelligence when performed by humans. Artificial intelligence focus on creating machine that can engage in behave that humans considered intelligent. Artificial Intelligence is concerned with the study andCreation of computer systems that exhibit some form of intelligence and attempts to apply such knowledge to the design of computer based systems that can understand a natural language or understanding of natural intelligence. One element in AI methodology is that progress is sought by building system that perform Synthesis before Analysis. (Wachsmuth, 2000**)**

©IJRASET 2015: All Rights are Reserved

A. Artificial Intelligence In Software Engineering

Software engineering is the introduction of formal engineering principle to the creation and production of the software. Software engineering basically deals with the development of the software. Development of a software is a long process which goes through various phases and requires an executable code. Human beings have been writing a code for the development of the software for a long time and no machine can yet do better.AI is the process of creating an intelligent machines that perform tasks done by human beings. Artificial intelligence technique can be used to improve many of the software development activities. Hence, there is significance potential for improving all phases of SDLC (Software development life cycle).

B. Related Work

According to (Katz and Zimmerman-Gal**,** 1981), an advisory system is developed to assist human programmers in selecting data structures. This system is currently in use in an educational setting.

The XPLAIN system incorporated techniques for explaining the behavior of a program by examining the expert knowledge from which the program was derived. The XPLAIN experiments were done in the domain of digitalis therapy (W. Swartout, 1983).

The DESIGNER project used psychological modeling and protocol analysis techniques to characterize the behavior of human algorithm designers. This characterization was the basis of attempts to automate the design of certain algorithms [D. Steier and E. Kant, 1985].

RML is a formalism for expressing software requirements using knowledge representation techniques. It has been used to describe the requirements for several small information systems. (Borgida *et al,* 1985).

According to (Jonathan**,** 2002**),** automated programming is a solution to phase independence, which results in reusable code. Some of the techniques and tools that have been used in an automated programming environment are:

Language Feature: Language feature is a technique which uses the concept of late binding. It means making the data structure flexible. Another important language feature is the object oriented programming which is a package of data and procedures in an object.

Meta Programming: This concept is developed in natural language processing and, it uses automated parser generators and interpreters to generate executable lisp codes.

Program Browsers: Program browsers looks at different portions of the code to make changes, thus obviating the need for ordinary text ordinary.

Automated Data structuring: It is a concept in which we map the high level specification of data structures into the implementation structure.

II. ASPECTS OF SOFTWARE ENGINEERING AND ARTIFICIAL INTELLIGENCE

Software engineering categorizes two kind of knowledge programming knowledge (for e.g. - data structure construction, control structure, programming language syntax and how to combine and choose them) and domain knowledge (e.g. - concepts, theories and equations characterizing the particular domain).a domain has unique properties which must be mapped to the programming language in which soft. Is being developed, thus it require a conversion from one knowledge to another (i.e. from domain knowledge to programming knowledge or from domain to programming language).In this we require some method from these types of conversion, which require some techniques in AI.

The traditional view of software development process begins at the requirements specification and ends at testing the software. At each of the stages, different kinds of knowledge (design knowledge at design stage and programming and domain knowledge at the coding stage) are required. At each of the two stages: design and coding exist a cycle: error recognition and correction. Experience shows that errors can occur at any stage of development. Error due to coding may occur because of faulty design.

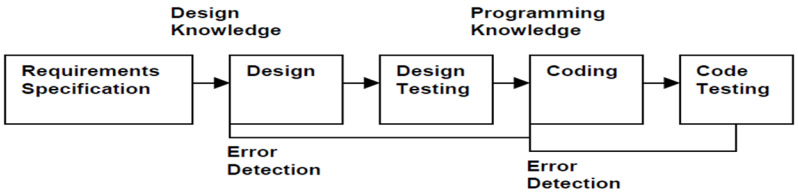


Figure 2. The traditional software development process

©IJRASET 2015: All Rights are Reserved

A basic problem of software engineering is the long delay between the requirements specification and the delivery of a product. This long development cycle causes requirements to change before product arrival. There is the problem of phase independence of requirements, design and codes in software development process. Phase independence means if we have made any decision any one level of the software development process, become fixed for the next level. Thus the coding team is forced torecode whenever a change occur in the requirement.

The term *Artificial intelligence* (AI) refers to the intelligence of machines, and the fields of scientific and engineering research that strive to develop that intelligence. The design of AI systems focuses on the creation of intelligent entities that are capable of perceiving their environment and reacting to various stimuli.

According to Wachsmuth it is not the aim of AI to build intelligent machines having understood natural intelligence but to understand natural intelligence by building intelligent machines" A further sub-division adapted from *Richter and abecker* of AI into sub-fields, methods and techniques is shown in figure 3.

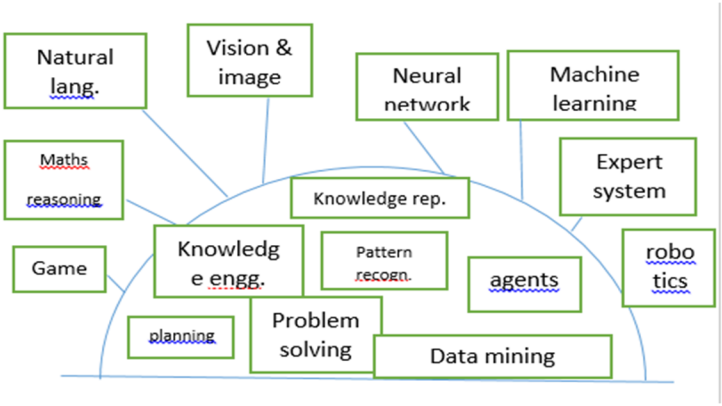


Figure 3. AI fields, methods and techniques.

A. Intersections between AI and SE

Today, methods and techniques from both disciplines support the practice and research in the respectively other research area.Figure 4 represent the current research areas in artificial intelligence, software engineering as well as common areas between them.The common research area between artificial intelligence are- ambient intelligence, computational intelligence agents and knowledge based software engineering.

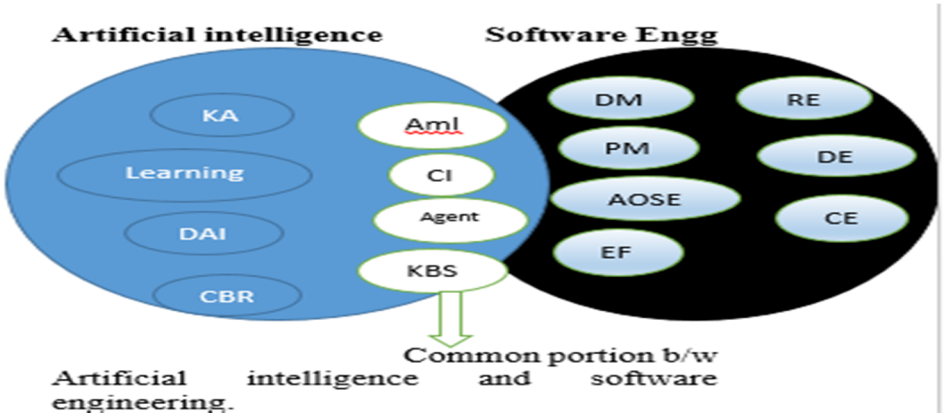


Figure 4. Research Areas in AI and SE and common between the two.

*Ambient intelligence*help us making a sensitive, adaptive and reactive systems that are informed about the user's needs, habits and emotions in order to support them in their daily work (*Da Costa and Punie Y, 2003*).There are several AI research areas for the development of smart algorithms for Aml applications (*Verhaegh et al, 2004*) e.g., user profiling, context awareness, scene understanding (*Aarts E. H. L., 2003*), or planning and negotiation tasks. Research from the SE side is concerned with model-driven development for mobile computing (*Da Costa and Punie Y, 2003*), the verification of mobile code (*Roman et al., 2002*), the

©IJRASET 2015: All Rights are Reserved

specification of adaptive systems,or the design of embedded systems*(Basten et al,2003*).

*Computational intelligence*has many techniques like neural networks, evolutionary algorithms, or fuzzy systems are increasing applied and adapted and adapted for specific SE problems. They are used to estimate the progress of projects to support software project management (Brandt M and Nick M, 2001), for the discovery of defect modules to ensure software quality, or to plan software twsting and verification activities to minimize the effort for quality assurance (khoshgoftaar T.M., 2003).

Software Agents are typically small intelligent systems that cooperate to reach a common goal. These agents are a relatively new area where research from KI and SE intersects. From the AI side the focus in this field lies on even more intelligent and autonomous systems to solve more complex problems using communication languages between agents in SE agents are seen as systems that need more or less specialized formal methods for their development,verification,validation, and maintenance

III. ARTIFICIAL INTELLIGENCE TECHNIQUES

Expert system development: -Expert system use knowledge rather than data to control the solution process. Knowledge engineers build systems by eliciting knowledge from experts, coding, that knowledge in an appropriate form, validating the knowledge, and ultimately constructing a system using a variety of building tools.

The main phases the expert system development processes are:-

A. *Planning*
B. *Knowledge acquisition and analysis*
C. *Knowledge design*
D. *Code*
E. *Knowledge verification*
F. *System evaluation*

Planning phase involves feasibility assessment, resource allocation, and task phasing and scheduling Requirements analysis. Knowledge acquisition is the most important stage in the development of ES. During this stage the knowledge engineer works with the domain expert to acquire, organize and analyze the domain knowledge for the ES. The goal of knowledge analysis is to analyze and structure the knowledge.

Gained during the knowledge acquisition phase. After knowledge analysis is done, we enter the knowledge design phase. At the end of design phase, we have Knowledge definition, detailed design, and decision of how to represent knowledge decision of a development tool. Consider whether It supports your planned strategy, internal fact structure, Mock interface. Coding this phase occupies the least time in the Expert System Development Life Cycle. It involves coding, preparing test cases, commenting code, developing User's manual and installation guide.

Knowledge-based techniques in AI can be used to modify traditional software development approach.one strategy is to automatically translate from the requirement specification to program testing. The user only provide the requirement and machine build using does the translation into the codes. (Engr.FarahNaazRaza, 2009)

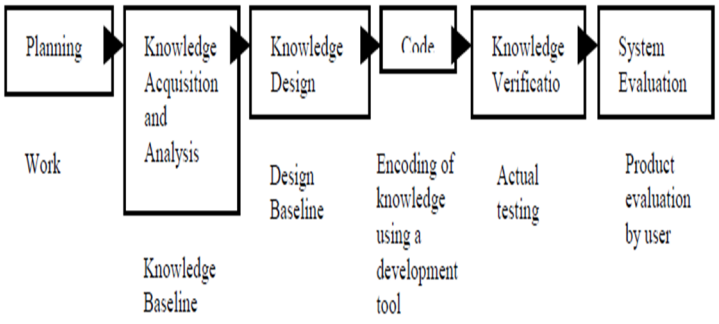


Figure 5 Expert System development

Richter defines three different levels as essential for describing knowledge base system. The cognitive layer (human oriented,

©IJRASET 2015: All Rights are Reserved

rational, and informal), the representation layer (formal, logical) and implementation layer (machine oriented, data structure and programs).

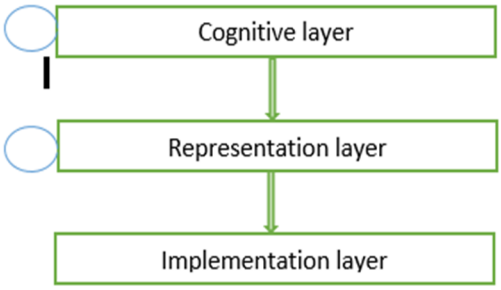


Figure 6. The three levels of knowledge based systems

The AI technique that handles phase independence problem is automated programming which results in reusable code. Thus, when a change is made in the design, that part of the design that does not change remains unaffected. Thus, automated tools for system redesign and reconfiguration resulting from a change in the requirements will serve a useful purpose.

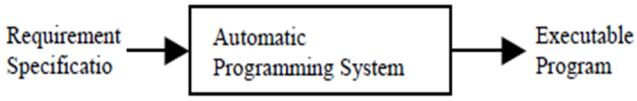


Figure 7 Automatic Programming System (APS)

A. Neural Networks

It has been widely and successfully used for problems that require classification given some predictive input features. They therefore seem ideal for situations in software engineering where one needs to predict outcomes, suchas the risks associated with modules in software maintenance (khoshgoftaar andlanning, 1995), software risks analysis (Neumann, 2002) and for predicting faults using object oriented metrices (Thwin and Quah, 2002).

B. Genetic Algorithms

In the areas of software development, Shan et al, 2002 utilize Genetic programming to evolve functions for estimating software effort. Two target grammars were adopted for the functions that allowed use of a range of mathematical functions (e.g., exp, log, sqrt) as well as a conditional expressions.

Fuzzy logic is another AI technique that is applied in software testing to manage the uncertainty involved in this phase of software development (Nand et al, 2007).

IV. CONCLUSION

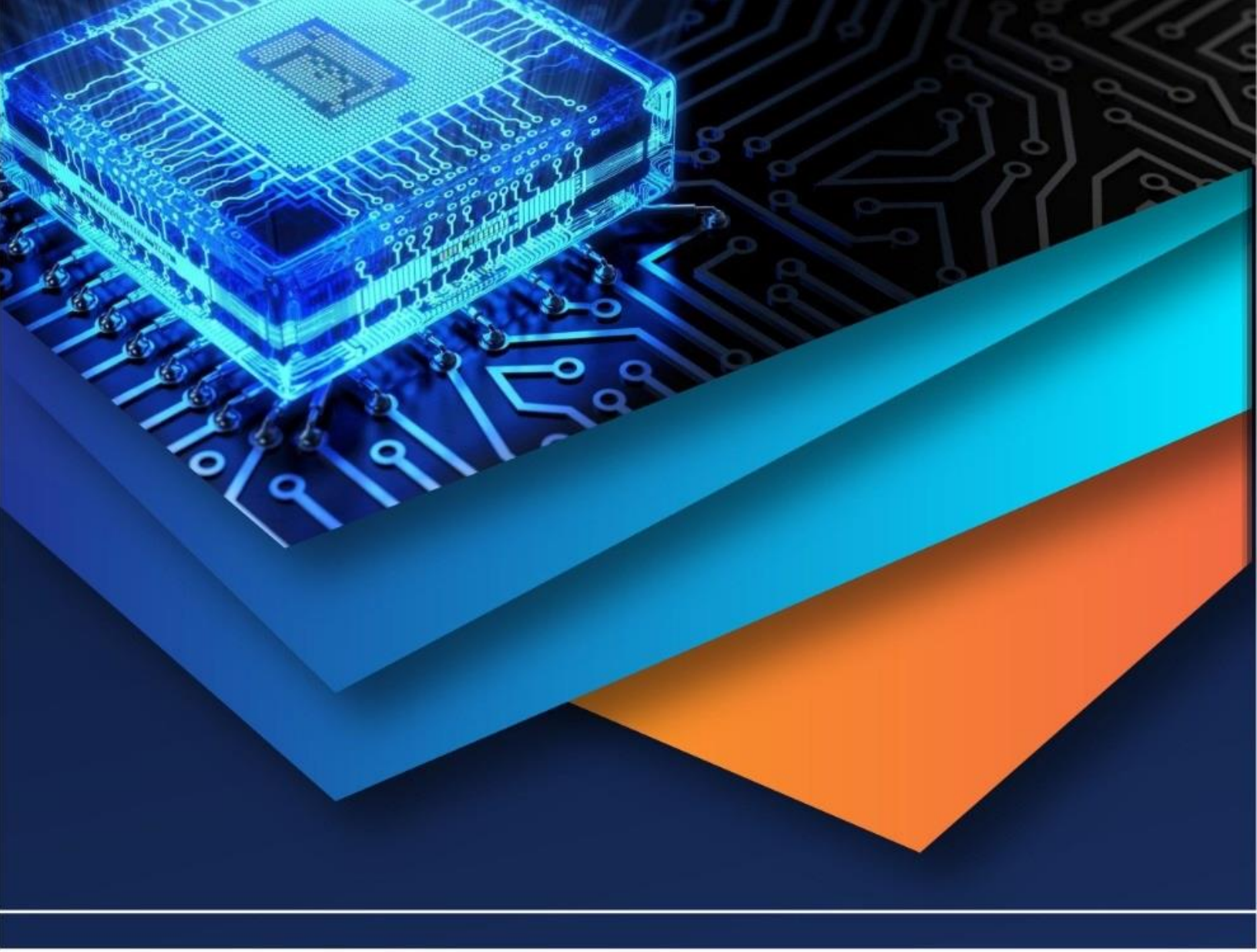
Software engineering help us to build a software product but by following the software engineering principles, the development of product takes very much time. The quality of the product can be increased by using AI techniques in the software development. By using AI based systems with the help of automated tool or automated programming tool we can eliminate risk assessment phase saving our time in software development, and building an effective product. Because of Artificial techniques in Software Engineering, we can reduce the development time in software development. Coding phase in software development process can be changed into Genetic Code, which is the main concern in the development of software.

©IJRASET 2015: All Rights are Reserved

REFERENCES

[1]M.L. Emrich, A. Robert Sadlowe, and F. Lloyd Arrowood (Editors), Expert Systems And Advanced Data Processing: Proceedings of the conference on Expert Systems Technology the ADP Environment (Elsevier-North Holland, New York, New York, USA) 1988.
[2]S. Katz and R. Zimmerman. An advisory system for developing data representations. In Seventh International Joint Conference on artificial Intelligence, pages 1030-1036, Vancouver, British Columbia, Canada, August 1981.
[3]W. Swartout. Explain: A system for creating and explaining expert consulting systems. Artificial Intelligence Journal, 21(3):285-325, September 1983.
[4]E. Kant. Understanding and automating algorithm design.IEEE Transactions on Software Engineering, 11(11):1361-1374, November 1985.
[5]A. Borgida, S. Greenspan, and J. Mylopoulos. Knowledge representation as the basis for requirements specifications.1EEE Computer, 18:82-91, April 1985.
[6]Jonathan OnowakpoGoddeyEbbahDepartment of Computer Science University of Ibadan,
Nigeria, 18 March-2002.
[7]M.L. Emrich, A. Robert Sadlowe, and F. Lloyd Arrowood (Editors), Expert Systems and Advanced Data Processing: Proceedings of the conference on Expert Systems Technology the ADP Environment (Elsevier-North Holland, New York, New York, USA)
[8]Shari Lawrence Pfleeger, Software Engineering: theory and Practice (Prentice Hall Publishers, Upper SaddleRiver, New Jersey, USA) 1998.
[9]Da Costa O and Punie Y., Science and Technology Roadmapping: Ambient Intelligence in Everyday Life (Ami@Life):Unpublished IPTS working paper, 2003.
[10]Verhaegh W.F.J., Arts E.H.L., and KorstJ.,Algorithms in ambient intelligence, vol.2. Boston: Dordrecht, ISDN: 140201757X, 2004.
[11]Roman G.C., Picco G.P., and Murphy A. L., "Software Engineering for Mobility: A Roadmap," presented at Future of Software Engineering Track of 22nd ICSE, Limerick, Ireland, 2000.

©IJRASET 2015: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)