*International Journal of Engineering Inventions*
*e-ISSN: 2278-7461, p-ISSN: 2319-6491*
*Volume 11, Issue 7 [July. 2022] PP: 17-19*

# Impact of Artificial Intelligence in Software Engineering

**Dibakar Goswami[1], Aditi Paul[2], Arittya Dutta[3], Debrupa Pal[4]**
*[123]Student, Computer Application Department, Narula Institute of Technology, Nilgunj Road, Agarpara, Kolkata 700109*
*[4]Assistant Professor, Computer Application Department, Narula Institute of Technology,Nilgunj Road, Agarpara 700109*

**ABSTRACT:**

This paper surveys the application of artificial intelligence approaches to the software engineering processes. These approaches can have a major impact on reducing the time to market and improving the quality of software systems in general. Existing survey papers are driven by the AI techniques used, or are focused on specific software engineering processes. This paper relates AI techniques to software engineering processes specified by the IEEE 12207 standard of software engineering. The paper is driven by the activities and tasks specified in the standard for each software engineering process. The paper brings the state of the art of AI techniques closer to the software engineer, and highlights the open research problems for the research community

Date of Submission: 22-06-2022 Date of Acceptance: 04-07-2022

## I. INTRODUCTION

The software intensive systems we develop these days are becoming much more complex in terms of the number of functional and non-functional requirements they need to support. The impact of low quality can also have a catastrophic impact on the mission of these systems in many critical applications. Moreover, the cost of software development dominates the total cost of such systems. Research in applying artificial intelligence techniques to Software Engineering have grown tremendously in the last two decades producing a large number of projects and publications. A number of conferences and journals are dedicated to publish the research in this field. The AI techniques are proposed in order to reduce the time to market and enhance the quality of software systems. Yet many of these AI techniques remain largely used by the research community and with little impact on the processes and tools used by the practicing software engineer.
The recent survey papers published in this field are mainly targeted to the research community. They are driven by the specific AI techniques used rather than the software engineering activities supported. They are also focused on a specific software engineering process such as software design [1]

## II. THE DEVELOPMENT PROCESS

The system and software architectures play a major role in driving the management activities during the development and maintenance of software systems. The standard provides a flow of the development process activities and associated documents. The system architecture design activity which produces the Software architecture and requirements allocation description (SARAD) document, establishes the top-level architecture of the system and defines the software and hardware items of the system.
Requirements are first expressed in natural language within a set of documents. These documents usually represent the unresolved views of a group of individuals and will, in most cases be fragmentary, inconsistent, contradictory, not prioritized and often be overstated, beyond actual needs. [2].
Requirements are first expressed in natural language within a set of documents. The main activities of this phase
are requirements elicitation, gathering and analysis and their transformation into a less ambiguous representation [3].
Problems arising during this phase can be summarized as follows:
● Requirements are ambiguous [4]
● Requirements are incomplete, vague and imprecise [5]
● Requirements are conflicting [6]
● Requirements are volatile [7]

Techniques learned from AI research make advanced programming much simpler, especially with regard to information flow and control as a result of advances in knowledge representation. In the following we focus on the AI techniques used in supporting the tasks of coding and testing.
Constraint programming is another AI technique that is applied in software engineering. Constraint programming has been, for example, used to design the PTIDEJ system (Pattern Trace Identification, Detection and Enhancement in Java. PTIDEJ is an automated system designed to identify micro-architectures looking like design patterns in object-oriented source code.

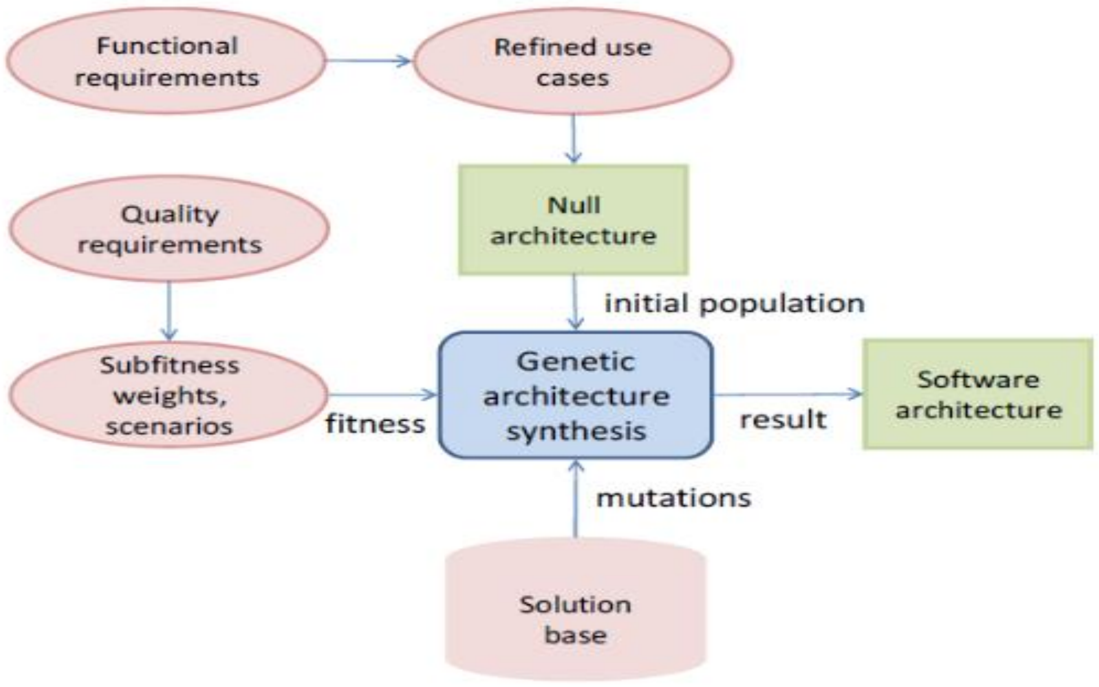

**Fig. 1 Evolutionary Architecture of Software Engineering**

Software testing remains an expensive task in the development process and one of the main challenges concerns its possible automation. AI techniques can play a vital role in this regard. There are many other ways how AI techniques can support the testing process. One of the earliest studies to suggest adoption of a knowledge-based system for testing.
There are many other ways how AI techniques can support the testing process. One of the earliest studies to suggest adoption of a knowledge-based system for testing.
A more active area of research since the mid-1990s has been the use of AI planning for testing. An AI planner could generate test cases, consisting of a sequence of commands by representing commands as operators, providing initial states, and setting the goal as testing for correct system behaviour

## III. CONCLUSION

Industrial AI can help in achieving the 3W's in smart manufacturing: Work Reduction,Waste Reduction, and Worry-Free Manufacturing. 'Worry' is an invisible concern with today'smanufacturing systems that could be due to things like product's bad quality, customerdissatisfaction or business decline. For handling these challenges, advanced AI tools must beutilized through a systematic approach. Work and waste reduction also can be achieved throughidentifying visible aspects of the problems and addressing their future concerns via theutilization of adaptive AI modules.
In this paper, we surveyed promising research work on applying AI techniques to solve some of the most important problems facing the software engineer. We surveyed research in the development activities ofrequirements engineering, software architecture design, and coding and testing processes. We summarized the most important open problems in these active research areas.

## REFRENCES

[1]. OutiRäihä, A survey on search-based software design,"Computer Science Review, 4 ( 2 0 1 0 ) 203 – 249.

[2]. J. (1993). READS: a requirement engineering tool. Proceedings of IEEE International Symposium on Requirements Engineering, (pp. 94–97), San Diego.

[3]. R. R. (2003). The requirements Engineering Handbook. Norwood, MA: Artech House Inc

[4]. S. G. (1986). The analysis of natural language requirements documents. PhD Thesis, University of Liverpool, UK.

[5]. Y., Xia, F., Zhang, W., Xiao, X., Li, Y., & Li, X. (2008). Towards Semantic Requirement Engineering. IEEE International Workshop on Semantic Computing and Systems (pp.67-71).

[6]. J., & Liu, F. X. (1995). A Formal Approach to the Analysis of Priorities of Imprecise Conflicting Requirements. In Proceedings of the 7th international Conference on Tools with Artificial intelligence. Herndon, V A, USA

[7]. F. (1994). From English to Formal Specifications. PhD Thesis, University of Salford, UK