**Ways of Applying Artificial Intelligence in Software Engineering**

**Theme:** This paper tells us that there have been lot of trials done to combine Ai with Software engineering and even applying Artificial intelligence concepts in Software engineering methods. But that has always been a reason of conflicts between the researches and the scientists.

Some of the researcher have tried and excelled in their work and various techniques to apply AI in SE proved beneficial. One of them discussed in this paper.

**AI-SEAL taxonomy**. AI in SE Application Levels (AI-SEAL) taxonomy that categorises application according to their point of AI application, the type of AI technology used and the automation level allowed.

**Conflicts:** However, there has been recent criticism that many of these approaches to building more intelligent software are too far from human-level intelligence and, thus, are not likely to be enough. Instead the critics argue that we actually need algorithms that build and extend causal models, can learn from very few examples:  
(one- or few-shot learning), and can reason symbolically with the patterns and knowledge they extract from sensors.

**Other Taxonomies:** There are several taxonomies proposed for SE in all of its diﬀerent knowledge areas (e.g., requirements and testing), but very few are created systematically. **Usman et al (It was best of all)**. performed a systematic mapping on the use of taxonomies in software engineering, and proposed a method to develop such taxonomies, which we used when defining AI-SEAL.  
**There are four phases** in the Usman et al. method: i) planning, ii) identification and extraction, iii) design and construction, and iv) validation. During planning we decided to be inclusive of all  
knowledge areas within software engineering (e.g., testing, requirements, processes) and rather seek a more fundamental aspect of how AI is applied.

There are **three major levels of this facet**,

two that are relevant before deployment of the software system **(process and product**) while the third is post-deployment representing the **runtime** application of AI in a software system.

The **process level** indicates that the AI is applied in the software development process and does not necessarily aﬀect, directly, the source code that will be deployed. An example would be test analytics, which could be used to optimise testing, but it does not by itself directly alter the code.

In contrast, the **product level** indicates that the AI directly aﬀects the source code. A concrete  
example would be automated program repair, which manipulates the code directly to automatically fix defects.

The **runtime level** represents AI applications that aﬀect the deployed software system during runtime. The canonical example would be autonomous and self-adaptive software systems in which some AI technology is learning and changing the system itself in a feedback loop. A more mundane, but recent example, would be the online learning of more optimal data structures and database indices based on the actual data stored during operation, in line with recent results from Google.

**CONCLUDING REMARKS**In this paper we propose the AI-SEAL taxonomy to help researchersand practitioners to classify diﬀerent AI applications in softwareengineering. The taxonomy has three facets allowing its users toclassify, the point of application (process, product and runtime)the type of AI technology (based, initially, on the five tribes proposed by Domingo’s