

Categories describing the impacts of AI and their definitions

1. Object-related – Refer to the (usually complex) performance output the ai provides. Classifying AI technologies by the output they generate entails the difficulty that diverse AI functionalities interact to produce a certain technical result. These are usually hard to understand and to differentiate for outsiders not involved in the development of the software and coding process.
2. Function related – These classifications of AI in software engineering refer to the technologies of AI applied in the software engineering process.
3. Process-related – These classifications of AI in software engineering refer to the stages of software development and analyze to what extent AI can support software engineering at the respective stage. Suggest to classify AI applications according to three stages of the software engineering life cycle: the software development, testing and deployment, and maintenance phase.

(Maybe use Fig. 1 in PowerPoint)

Study relies on two data sources – a systematic review and qualitative expert interviews

Give some background on the 5 engineers interviewed.

AI in software project planning

This is the stage where software developers and clients come together to determine the project objectives and customer requirements. Scheduling and planning is paramount here in order to ensure technical effectiveness and economic efficiency of software projects. The analytics of cost, duration and quality under certain constraints has grown over the years and increased complexity. Duration and cost of projects are contradicting goals at first sight and human planners struggle to harmonize both objectives. AI tools are useful to support this process. Task assignment in this phase is an issue for optimal task, time and budget allocation. AI is useful to support software project management at the stage of task assignment and human resource allocation. Use of ant colony optimization and Bayesian Network algorithms.

AI at the stage of problem analysis

This is the stage where the problem set is defined in terms of software tools and development requirements by the software development team. AI is used here to predict project success and risk factors in software development.

AI at the stage of software design

This is the stage where the software project is clearly structured and development tasks are assigned. In the future, AI may be given simple inputs (the problem set) and develop own program codes that are also self-improving. Neural networks are currently self-enhancing, but they still operate on man-defined routines in the phase of software design. These tools can be used to answer specific components and problem within a problem, but the work strategy and software still has to be designed by a human engineer.

AI at the stage of software implementation

This is the stage where the actual coding gets done. Neural networks have been developed to assist software coding. Processing natural language into software code is a capability of which has been researched since the 1980s. Currently, AI can generate prototype codes from human language and then are adjusted by human engineers to be usable. In the future, it is possible that AI could produce coherent codes and possibly even implement the code into existing routines self-reliantly. It's possible that the code it generated might not be readable by humans.

AI at the stage of software testing and integration

This is the stage where the developer and client test the functionality of the software product in practice, identify and analyze errors and tailor the product to practice requirements. AI uses strategies of pattern recognition and machine learning to support software testing and integration. Automated AI based testing and integration functions today are self-improving and use dynamically changing routines, but to date, human engineers are required to define the testing process and requirements to the program, while the test implementation can be done by the machine. A survey among 328 experts came to the conclusion that about 35% assume that a complete substitution of human programmers by machines in the testing phase will never be possible. AI does however currently drastically reduce the amount of time, manpower and cost to perform the testing process, documentation and evaluation of the tests. Time to market and development costs are reduced. Human control and intervention to date remains necessary to prevent erroneous testing routines and to critically reflect the validity and reliability of test results.

AI at the stage of software maintenance

This is the stage where the software company assists the customer in product application, provides regular upgrade and makes further adjustments upon client requirement. This stage has some of the greatest success in AI implementations because it's dealing with already completed code. It's not having to build from scratch but analyzing something already created. Humans are still needed to make useful decisions with information gathered.