



Demystifying AI in Software Development

How AI is Transforming the Industry?

WHITEPAPER

Abstract

AI is quickly becoming prevalent in different industries and software development is part of this change. This white paper mainly aims to analyze the potential and role of AI in software development and to reveal its significance.

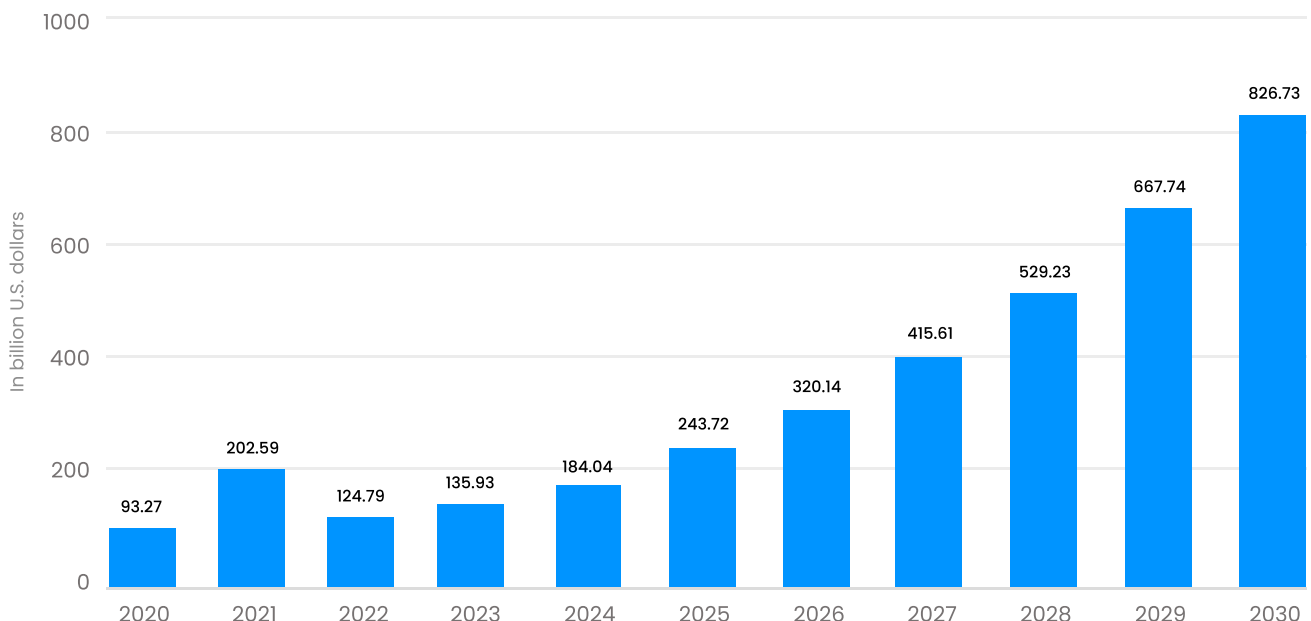
We will also describe concrete use cases in more detail, such as code generation, testing, and development process enhancement. In addition, we demystify the use of AI-powered tools & techniques in SDLC. This paper is an attempt to provide an insight into - how AI is really taking over software development, and what benefits one can reap by using them.

Introduction

Humans and machines are two crucial elements that are known for driving technological advancements in modern society. Among a lot of tech advancements, artificial intelligence is definitely a true player. Even though AI has been around for over a decade, it continues to be in high demand.

In 2024, the AI market size will hit 184 billion U.S. and in 2030, it is expected to value **826 billion U.S. dollars.**

Artificial intelligence (AI) market size worldwide from 2020 to 2030
(In billion U.S. dollars)



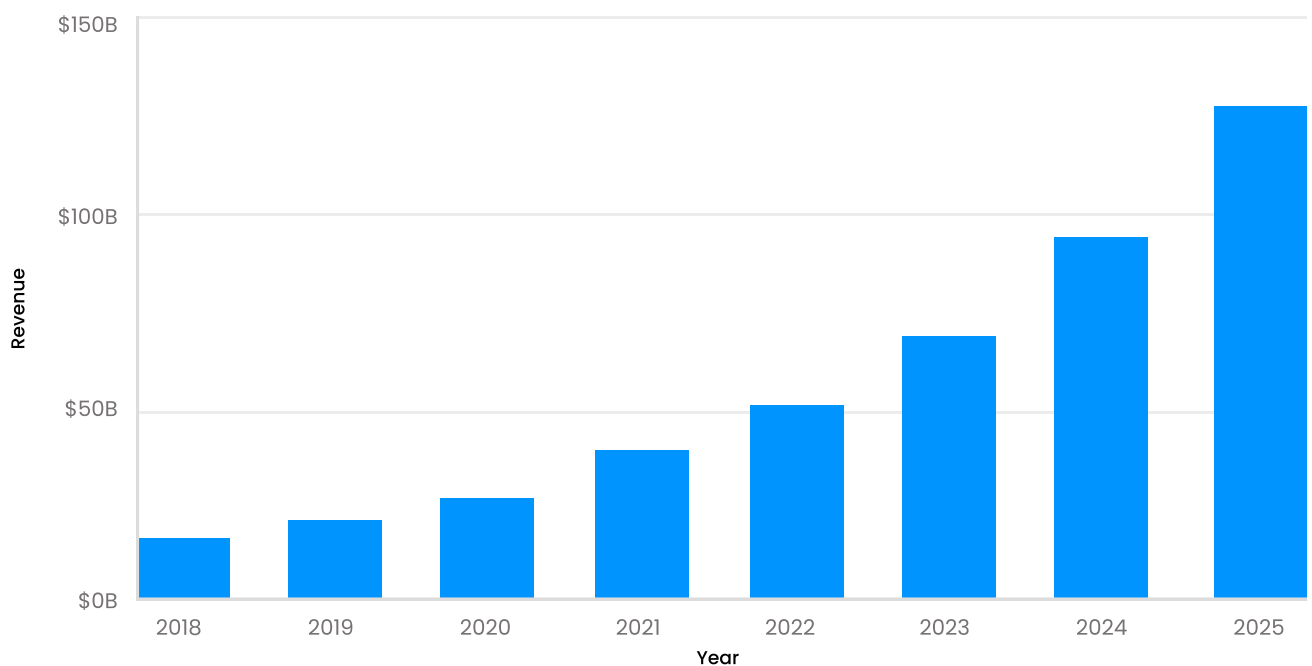
The graph of AI use is on the rise. Startups to large enterprises, are increasingly adopting AI technology to improve their operations and gain a competitive edge in the market.

AI empowers machines to exhibit intelligent behavior—such as learning, perceiving, and decision-making. Businesses use all of these AI's capabilities in various ways. One such is software development.

Today, AI has reached a stage where it acts as a powerful ally in writing, debugging, and even optimizing code. This is not an illusion, it is a new reality. If you are not aware of it, it's high time to learn things about AI in software development.

Currently, the worldwide yearly earnings from AI software are now more than \$50 billion.

Global artificial intelligence software market revenue



© Exploding Topics

- Moreover, 80% of companies are investing in AI, including in software development.
- Approximately 20% of companies have incorporated AI in some offerings or processes.
- 93% of developers want to learn more about deploying AI & ML models in their applications.
- AI technology in software development can reduce operational costs by 62%.
- 50% of organizations plan to increase their spending on AI and machine learning for software development in the future.
- 21% of tech executives said their organizations are embedding AI in software development to improve efficiency.

On a serious note, unlike the science fiction scenarios of self-writing software, AI today is more like a collection of powerful tools that can assist developers.

Therefore, whether you're a developer, a tech enthusiast, or someone curious about the future of AI in software development, this whitepaper is an eye-opener. Sportingly, you will get everything around this from the beginning.



What is AI?

AI is an umbrella term for computer programs. It has been commonly known as a branch of computer science that deals with machines that imitate human intelligence such as thinking and learning capabilities. However, a more practical definition describes AI as a collection of computer programs that can:



Analyze data

AI has the potential to analyze large sets of data more effectively. In general, humans may take much longer to spot those trends in large databases.



Make decisions

AI can draw or arrive at decisions/ conclusions if decisions or recommendations have to be made based on the data analysis.



Adapt and learn

We can train AI systems to become better as they are fed with more information.

Different types of AI relevant to software development

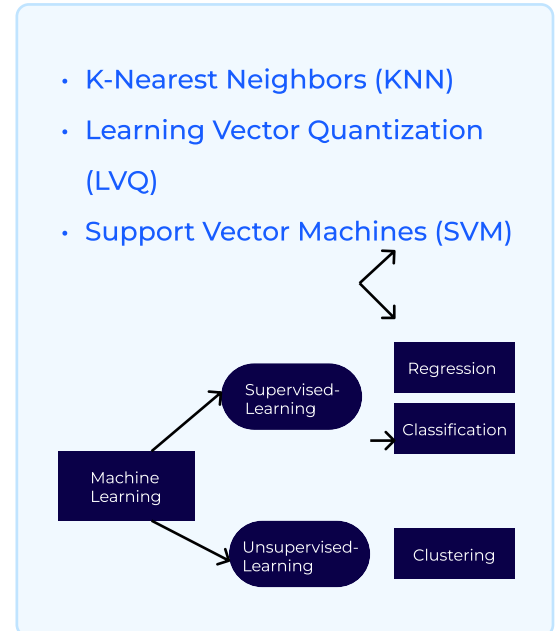
1. Machine Learning

First off, machine learning is a subset of AI. Popularly denoted as ML (an abbreviation used for machine learning). It allows computers to learn without being explicitly programmed. Machine learning algorithms are trained on data, and they can then use that data to make predictions or decisions.

Supervised, Unsupervised, Semi-Supervised Learning, and Reinforcement Learning are their four different types. In software development, ML is used in situations such as code, bug, exception, error prediction, and recommendation systems for software.

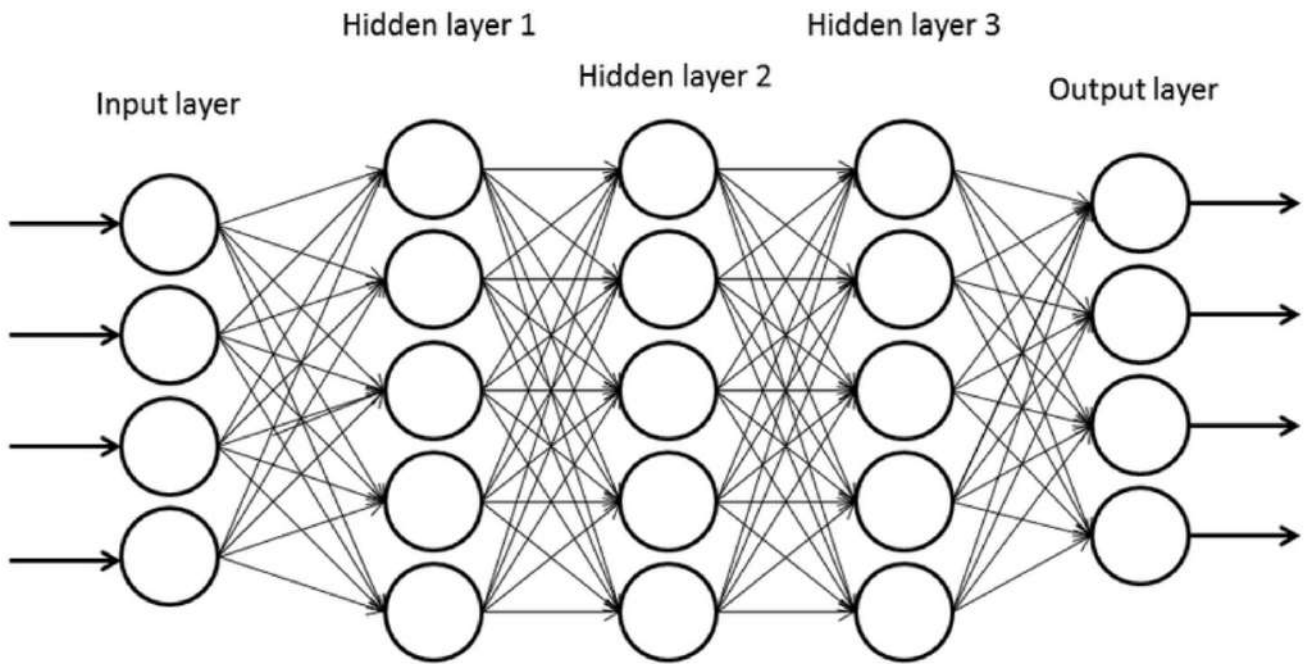
Moving further, Besides, some of the popular machine learning algorithms are-

- Linear Regression.
- Logistic Regression.
- Linear Discriminant Analysis.
- Classification and Regression Trees.
- Naive Bayes.



2. Deep Learning

Deep learning is a type of supervised learning that uses artificial neural networks. Neural networks are inspired by the structure of the human brain, and they are able to learn complex patterns from data. Some of the activities in software development that can be done by deep learning are image recognition for user interface design, code generator, and code quality prediction.



3. Natural Language Processing (NLP)

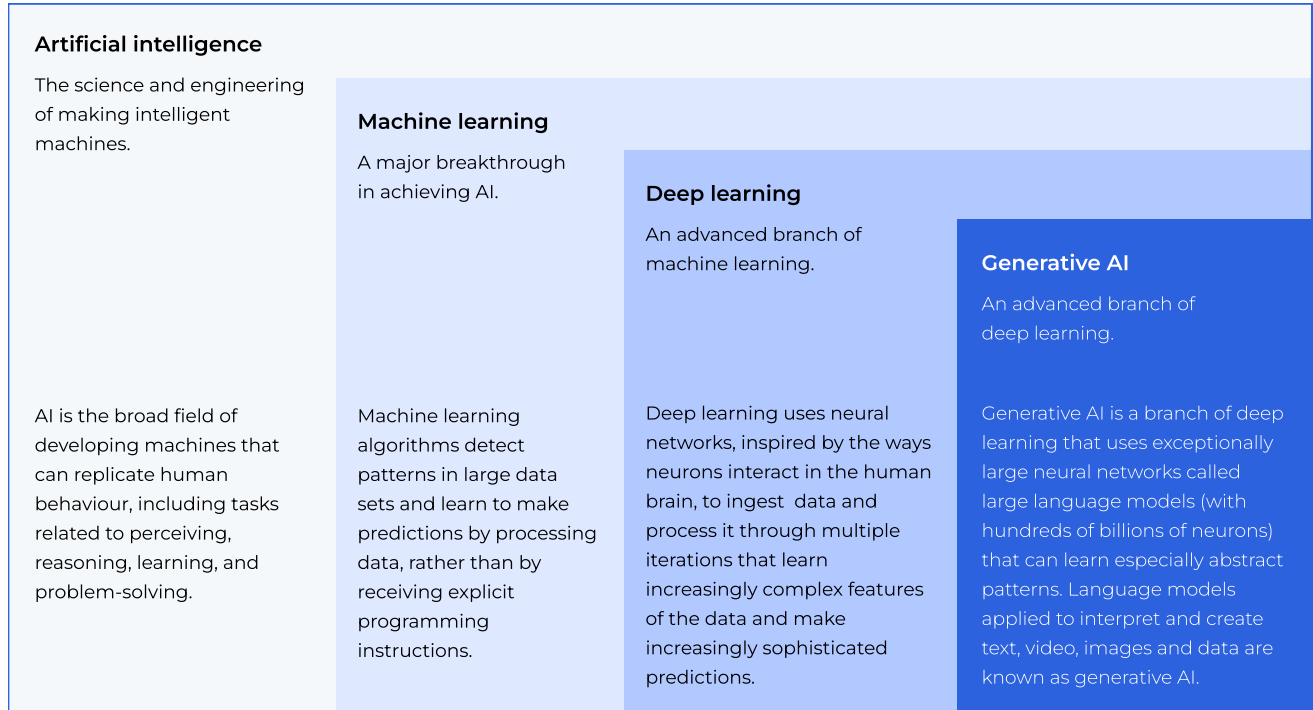
NLP is a field of artificial intelligence that deals with the interaction between computers and human language. Natural Language Processing supports computers to understand the meaning of text and speech, and to generate text and speech that is natural-sounding. In accordance with applying NLP techniques in software development, it is possible to automatically translate comments to documentation, code from commentaries, and enhance code search.

4. Generative AI

Generative AI, or gen AI, is a type of AI that creates content when given a prompt. A tool like ChatGPT is one popular tool based on Gen AI that produces text when the prompt is given. Plus, they learn from existing software and create new code snippets, translate code between languages, or even create unit tests.

Artificial intelligence is a machine's ability to perform some cognitive functions we usually associate with human minds.

The evolution of artificial intelligence



© McKinsey

5. Robotics

Although not directly involved in writing code, robotics can play a supporting role in software development. For instance, robotic process automation (RPA) can automate repetitive tasks associated with software deployment and testing. Thus, developers and testers are free from more complex work.

6. Machine Vision

This technology allows computers to analyze and understand visual information. In software development, machine vision can be used for tasks like automated visual regression testing (comparing UI screenshots for changes), analyzing code diagrams and flowcharts, and even generating code based on visual mockups.

The role of AI in various software development stages

1. AI in software project planning

AI especially disrupts the pre-development phase of the software development life cycle. Its uses change the way in which planning and requirements analysis is being done. Since AI has the ability to identify patterns and potential trends from historical data, it leads to a less subjective and more objective approach to creating project specifications.

Managing requirements—figuring out what users need from software, checking those needs, and keeping track of them—are some of the big things that are easily be managed by integrating AI in the planning phase of software development.

Some companies have started using AI digital assistants that can read requirements documents, find unclear or inconsistent parts, and suggest ways to improve them. Such tools are either trained or using advanced algorithms like- ant colony optimization (ACO) and Bayesian networks.

Businesses using these tools have reported cutting their requirements review time by more than half. AI helps optimize key project factors like cost, duration, and quality. In short, AI tools offer invaluable insights into effective project planning strategies, ultimately leading to improved project outcomes.

2. AI for Problem Analysis in Software Development

AI embedded with its smart algorithms has the power to predict the success and risks of software projects. Often, hybrid technology road mapping techniques that use artificial intelligence are used in evaluating new technologies.

AI can integrate both qualitative (expert opinions) and quantitative data (project history) to assess project feasibility.

Currently, the task of structuring problem sets is primarily done by humans. Machines just follow set patterns and use probabilities to handle uncertainties.

But if we start using AI in our software problem analyzing stage, we can make the process of finding and understanding problems much faster and easier. This will help us move projects forward more quickly because we will have a clearer and more accurate idea of what we want to achieve.

3. AI at the Stage of Software Design

AI tools have changed how we design user interfaces. Now, it is equally important as it helps in improving user experiences. These tools use different algorithms that can create design prototypes automatically. Not just automation AI-driven tools in software development significantly cut down on the resources and time traditionally required to design software.

4. Coding with AI

As developers write code, AI tools can suggest how to complete lines of code, which can cut the number of keystrokes needed by up to half. Some of these tools even provide a ranked list of useful code snippets. They work similarly to Gmail's Smart Compose, which suggests words or phrases as you type an email.

5. AI in Testing and Quality Assurance

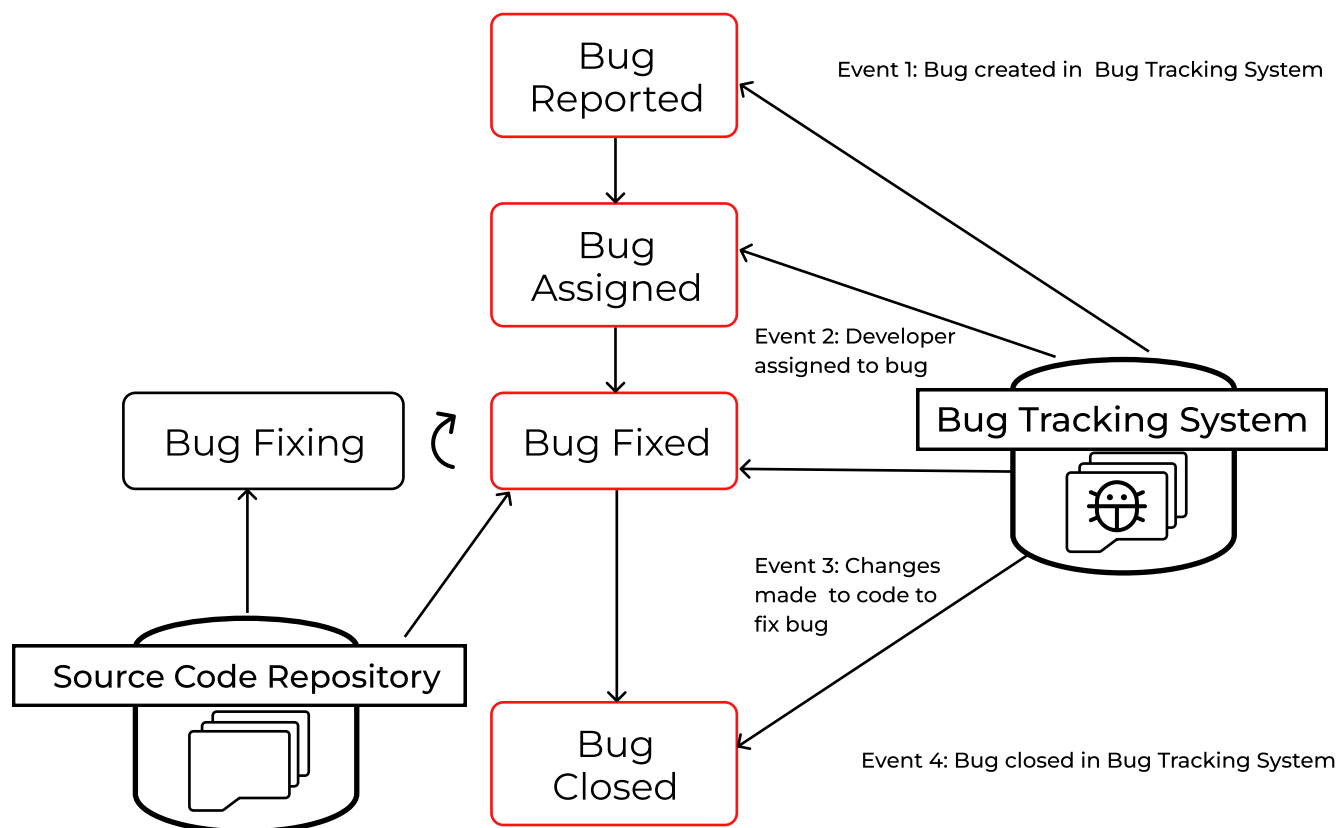
Sometimes, software issues only show up after the software is actually running in the environment it's designed for. However, AI tools are now helping predict these issues before they happen. AI-driven testing tools employ sophisticated algorithms to generate comprehensive test cases.

With their help, you can automate critical testing processes and elevate the accuracy of quality assessments.

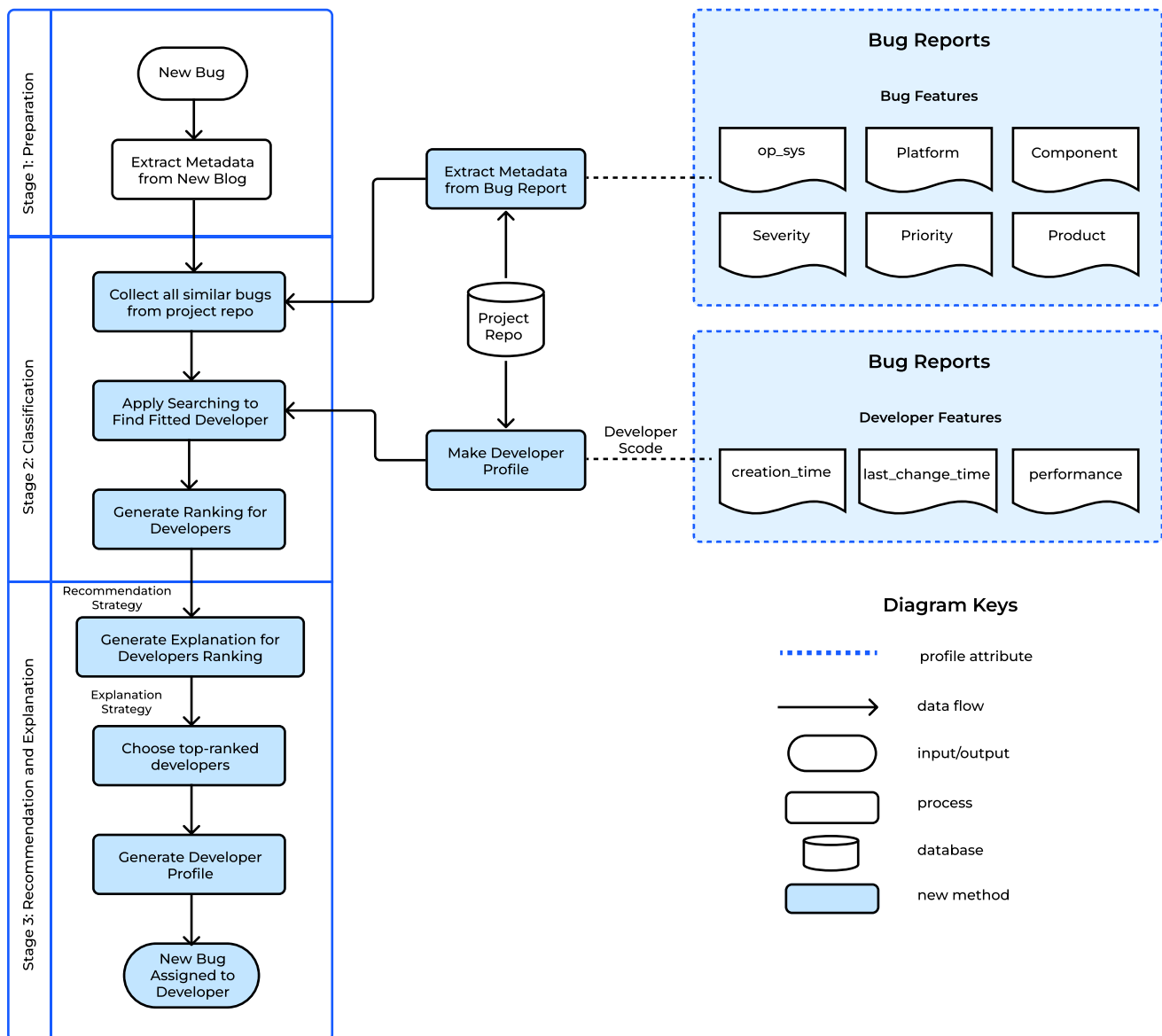
AI-powered code-review tools can automatically find bugs and suggest fixes by understanding what the code is supposed to do and spotting common mistakes. For example, Facebook uses a tool that predicts bugs and offers solutions with an 80% accuracy rate.

Comparing the usual bug-fixing model vs. machine learning bug-fixing technique.

General Bug Fixing Model



Assign Bug based on Developer Recommendation (ABDR) training model



(ABDR) training model, is a technique, developers use to train a system that learns from past developer work to recommend the best person for a new bug.

Basically, it uses a supervised machine learning technique so that it is easy to extract things like what the developer knows (developer similarity), how long it typically takes them to fix something (bug handling time), and how successful they've been in the past (effectiveness). This helps match the right developer to the right bug.

Looking at data from previous code releases and application logs, AI can speed up the process of finding and fixing problems if they do occur. Ubisoft, a video game company, says machine learning helps it find 70% of bugs before testing even begins.

6. Deployment and Maintenance Made Smarter with AI

Introducing AI into the deployment and maintenance phases of software development is a major step towards creating more efficient and reliable systems. AI-driven tools help make these stages smoother, ensuring that software moves from development to deployment seamlessly and stays reliable after launch.

One of the key benefits of AI is predictive analytics, which can foresee issues before they happen. This allows teams to take action early, saving time and resources. For example, predictive maintenance algorithms can analyze data to predict system failures. This means maintenance can be done ahead of time, reducing downtime and extending the life of the application.

Automated deployment tools also play a significant role by simplifying the release process. They ensure software updates happen smoothly and without human errors. This not only makes operations more efficient but also enhances the user experience by keeping applications reliable and up-to-date with minimal disruptions.

Common AI Tools & Frameworks for Software Development

List of the most popular AI tools and frameworks used in software development:

1. TensorFlow



Mention as - An end-to-end platform for machine learning.

TensorFlow is a powerful open-source framework that is being developed by one of the world's top brands- Google.

The main purpose is to serve numerical computation and large-scale machine learning. It offers a flexible architecture that allows developers to build and deploy models on various platforms, including CPUs, GPUs, and mobile devices.

TensorFlow is known for its various usefulness, one of which is- it supports various deep learning architectures and functionalities. Besides, it has the potential; to handle large datasets and complex models efficiently. Widely, it is used to build AI-based applications and perform advanced research.

2. PyTorch



PyTorch is a machine learning library that was originally developed by Meta AI but now it is owned by the Linux foundation. Although, it is based on the Torch library. The applications of PyTorch are computer vision and natural language processing.

It is known for its dynamic computational graph and ease of use. Its strength is that it uses Python syntax which makes it familiar to developers with Python experience.

3. Scikit-Learn



Scikit-Learn open-source library focuses on traditional machine learning algorithms and statistical modeling. It offers a wide range of algorithms for tasks like classification, regression, clustering, and dimensionality reduction.

It is easy to learn and use. Beginners can get started with this in a more efficient way. Besides, it integrates well with other Python libraries like NumPy and Pandas.

4. Keras



Keras is a high-level API designed to simplify the process of building and training deep learning models. It provides a Python interface for artificial neural networks. It runs smoothly on both GPU and CPU. Besides, its user-friendly design makes it easy to learn and easy to use. Keras allows for the rapid prototyping and deployment of models.

5. Hugging Face Transformers



Hugging Face Transformers is an open-source library that provides pre-trained models for natural language processing (NLP) tasks. It offers state-of-the-art models for tasks like text classification, sentiment analysis, question answering, and machine translation. Besides, it also provides APIs to download to further tune them to maximize performance. Hugging Face Transformers support framework interoperability between TensorFlow, PyTorch, and JAX

6. Tabnine



Tabnine is an AI code assistant that personalizes suggestions based on your coding style and protects your code privacy. It offers features like code generation, explanation, and testing to help developers be more productive. Tabnine is secure, with models never trained on your code, and offers enterprise-grade security. You can try it for free or talk to an expert for more information.

7. Stepsize AI



Stepsize AI is an AI-powered tool that automates project reporting for Agile software development teams using Jira or Linear. It generates insightful reports with commentary, helping project managers and product owners understand progress, identify risks, and stay aligned on goals. In essence, Stepsize AI eliminates manual setup by providing automated weekly updates based on your issue tracker activity.

8. MutableAI



MutableAI is an AI-powered tool designed to create and maintain up-to-date code documentation. It automatically generates wiki-style articles directly from your codebase, eliminating the need for manual documentation creation.

Ask questions about your codebase in a natural language format, and receive AI-powered answers. A non-technical mode is also available for those less familiar with code.

Besides, you can revise sections of the documentation or the entire article manually or with the help of AI suggestions. MutableAI offers a free plan for open-source projects, with paid plans providing features like advanced generation, higher quality models, and priority support.

9. CodiumAI



CodiumAI offers two AI-powered developer tools to improve code quality and efficiency: This tool integrates with your IDE (Integrated Development Environment) and helps you write better code by suggesting meaningful tests and improvements as you code.

Analyzing code, comments, and specifications to identify potential issues. Besides, CodiumAI helps you understand how your code behaves and how changes might affect other parts of the codebase. Saving you time by generating tests and catching potential problems early on.

This tool helps development teams review pull requests more efficiently by providing AI-powered analysis and feedback. It works seamlessly with various Git platforms like GitLab, GitHub, etc.

10. Grit.io



Grit.io is an AI-powered tool that automates software maintenance tasks like code migrations and dependency upgrades. It uses machine learning to analyze code and suggest improvements, helping developers save time and focus on new features. Grit integrates with popular platforms like GitHub and VS Code, and offers customization options for developers to define their own coding styles.

Benefits of AI in Software Development

The integration of AI into software development offers a multitude of advantages, revolutionizing the industry and empowering businesses to build better products faster. Here's a closer look at some key benefits:

1. Increased Productivity and Efficiency

AI steps up by automating tedious and repetitive tasks that often bog down developers. This can include code generation for boilerplate functionalities, automated testing, and data migration. Furthermore, by freeing developers from these chores, AI allows them to focus on more complex and creative aspects of software development.

In addition, AI-powered tools can analyze code for potential errors and vulnerabilities, helping developers write cleaner and more robust code. This not only reduces the risk of bugs but also improves maintainability and future development efforts.

Ultimately, by automating tasks and improving code quality, AI leads to faster development lifecycles. This translates to quicker product launches and faster time-to-market, giving businesses a competitive edge.

2. Enhanced User Experience (UX)

AI can look at information about users and how they use things to change the way a software program works for them. This can involve changing how things look on screen, suggesting features they might like, and showing them information they'd find interesting. In turn, this keeps people engaged, makes them happier with the program, and makes them more likely to keep using it.

Besides, using AI can be used to design interfaces that are easier to understand and use. For example, AI-powered chatbots can provide immediate help to people using the program, and something called natural language processing (NLP) can let people talk to the software using regular words instead of having to learn special codes. These advancements make software applications more accessible and user-friendly, and overall, a better experience for people using them.

3. Innovation and New Product Development

AI acts as a catalyst for innovation, opening doors to entirely new features and functionalities that were previously unimaginable. For example, AI-powered chatbots or AI assistants can be integrated into software, creating a new level of user interaction where users can have natural conversations with their software.

Furthermore, AI can accelerate the development process by analyzing vast amounts of data to identify trends and opportunities. This data can inspire the creation of entirely new software solutions and product categories that we haven't even thought of yet.

Additionally, AI can streamline the process of experimentation by allowing developers to rapidly test and iterate on new software concepts. With AI's help, developers can experiment and refine ideas more efficiently, leading to the creation of groundbreaking new products.

4. Data-Driven Decisions

Making smarter decisions with data: AI can analyze software development data to identify trends and patterns that humans might easily miss. This allows developers to make data-driven decisions about features, functionalities, and how to best use their resources.

In addition, AI can analyze software performance data to predict potential problems before they happen.

This proactive approach can prevent costly downtime and ensure smooth software operation. By analyzing project data with AI, development teams can make informed decisions about assigning tasks to developers, ensuring that everyone's skills and expertise are being used effectively.



Challenges with AI in Software Development

1. Skill Demands Escalate

One of the challenges of implementing AI in software development is the skill gap. No doubt AI has a lot to offer, but to actually make use of that we need highly skilled software engineers. There is a need for talented developers who understand both AI and software development to truly unlock its benefits.

You can't just throw AI at a problem and expect magic to take place. Besides, for a result, it takes precise instruction and expertise to get the most out of this powerful technology.

The truth is - that the number of AI developers for now is less. In the study, around the world, there are only about **150,000** machine learning engineers. This is a very small number compared to the 29 million software engineers globally.

2. Increased Hardware Requirements

Leveraging advanced AI tools necessitates substantial computational resources. Correspondingly, software development firms utilizing AI need to invest in high-performance hardware. This includes powerful computers with extensive storage capacity and robust energy sources.

Unfortunately, this high-performance hardware can come with a hefty price tag. Companies that have limited budgets find this as a big challenge.

3. Rising Licensing Costs

Access to cutting-edge AI and machine learning technologies often comes with significant licensing fees. While these tools offer considerable advantages by streamlining development processes and fostering innovation, the initial and ongoing financial investment can be substantial. However, it's worth noting that, despite the high costs, these AI-enhanced tools typically offer a strong return on investment due to the efficiencies they introduce.

4. Complications in Implementation

Access to cutting-edge AI and machine learning technologies often comes with significant licensing fees. While these tools offer considerable advantages by streamlining development processes and fostering innovation, the initial and ongoing financial investment can be substantial. However, it's worth noting that, despite the high costs, these AI-enhanced tools typically offer a strong return on investment due to the efficiencies they introduce.

5. Ethical and Privacy Concerns

As AI applications become more pervasive, ethical questions and privacy concerns grow. Ensuring AI systems are designed and operated ethically requires ongoing vigilance and potentially introduces complexities in development, such as the need for bias mitigation and the safeguarding of user data.



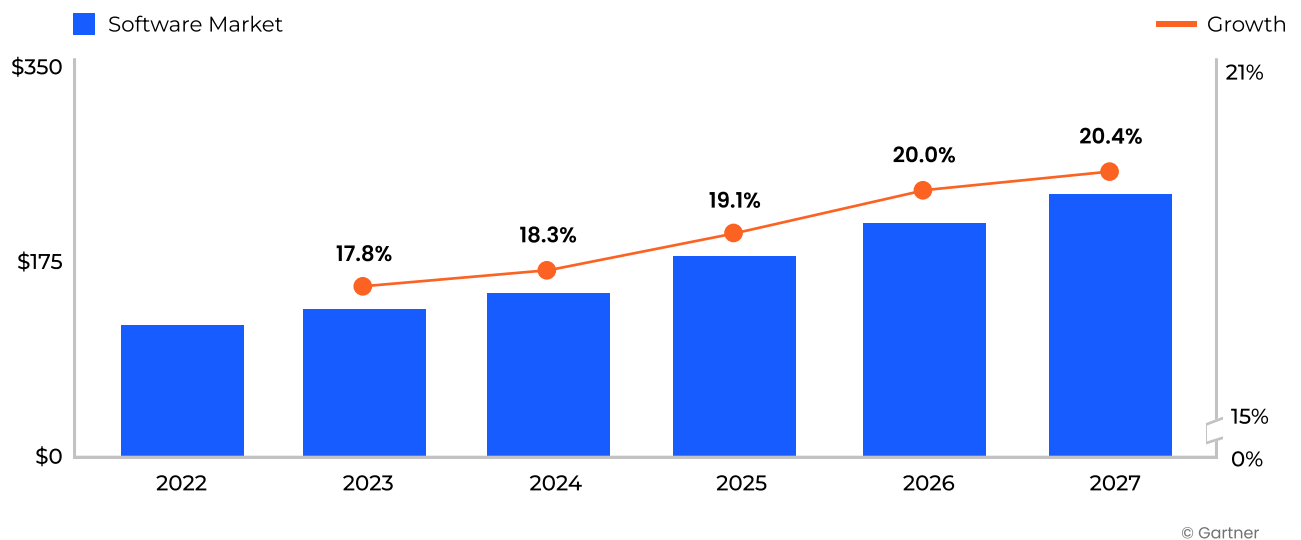
Future of AI in Software Development

A recent report by Gartner predicts that by 2025, a significant majority (over 70%) of organizations will integrate AI into their processes in various forms.

Correspondingly, as organizations gain more experience in experimenting with AI, their investment will continuously keep increasing.

In line with this trend, Gartner predicts that AI software will experience significant growth, reaching a market value of \$297 billion by 2027.

AI Software Forecast (\$m) and Growth(%)



As if everything goes as planned, AI in software development promises to make the future of technology even more exciting and advanced.

Predictive Analytics will change the way we plan our projects. AI coding assistants could well become more advanced. They may learn to code better than human professionals, and will ultimately code for us if we train our neural networks sufficiently.

In short, in the future, AI will take care of the boring stuff, like writing repetitive code and finding bugs. Above all, it can spot user trends that programmers might miss. That means, AI may come up with fresh ideas for developing innovative software solutions.

Understand this as -the future of AI in software development is about teamwork. It is not replacing developers, but changing the way they do their jobs. Even though AI is reliable for coding, software developers out there have to act as an AI coach who can choose the right AI tools and make sure everything works together.

Partner with ScalaCode to ensure the best support for the implementation of AI in software development.

**Get in
Touch →**



About ScalaCode™

ScalaCode is a globally acclaimed software development and IT consulting services provider with 13 years of experience. We help businesses of all sizes with custom software solutions that meet their specific requirements and drive business growth.



Headquarter & Delivery Center

India

Suite # 16, Ground Floor
Tower A, Stellar IT Park, C 25
Sec – 62, Noida, Uttar Pradesh

☎ +91 8527981600

✉ sales@scalacode.com

Our Global Presence

United States

P.O Box 1286,
Layton, Utah 84041

☎ +1 801 618 5791

☎ +1 409 995 3910

United Kingdom

Rowan House, Culmhead
Nr Taunton TA3 7DU
Somerset, UK

☎ +1 801 618 5791

Sweden

Fagerlidsvägen 15D
566 92 HABO

☎ +46 76 310 62 86

Canada

2433 Lakeshore Road, Burlington
Ontario Canada L7R 1B9

Australia

16-18 Beverley Avenue
Rosedale South Brisbane
Queensland 4123

☎ +61423661345