Real-Life Experiment:

The Benefits and Opportunities of AI in Software Development

White Paper

Table of Content

Introduction

Starting a new era where human creativity and artificial intelligence work together, AI brings significant improvements in how we build, design, and use software. By employing this cutting-edge technology, engineers can deliver software products faster, making the entire process more efficient.

With AI solutions, it is possible to automate a number of manual, routine, and creative tasks, resulting in time and cost savings for both IT companies and their clients. The rise of generative artificial intelligence (GenAI) is shifting us from traditional ways of developing software towards a new AI-driven approach.

Since GenAI is at the peak of its hype, we at Intetics decided to find out whether the integration of AI tools into our software development process indeed can add that much value to us and our customers.

In this regard, we decided to launch an experiment where we asked our team consisting of three full stack developers, an automation quality assurance engineer, a business analyst, a UI/UX designer, and a project manager to start utilizing AI-powered assistants. The primary objective was to measure efficiency gain (or decrease) compared to the expectations on given tasks.

Based on a few of Intetics's projects, we will delve into how generative AI impacts the stages of the software development life cycle (SDLC), from app design to testing and deployment, and examine the potential of AI. In addition, you will discover whether AI is indeed so useful for IT experts or perhaps it is worth waiting until more advanced AI assistants come to the market.

Table of Content

1. Planning and Analysis

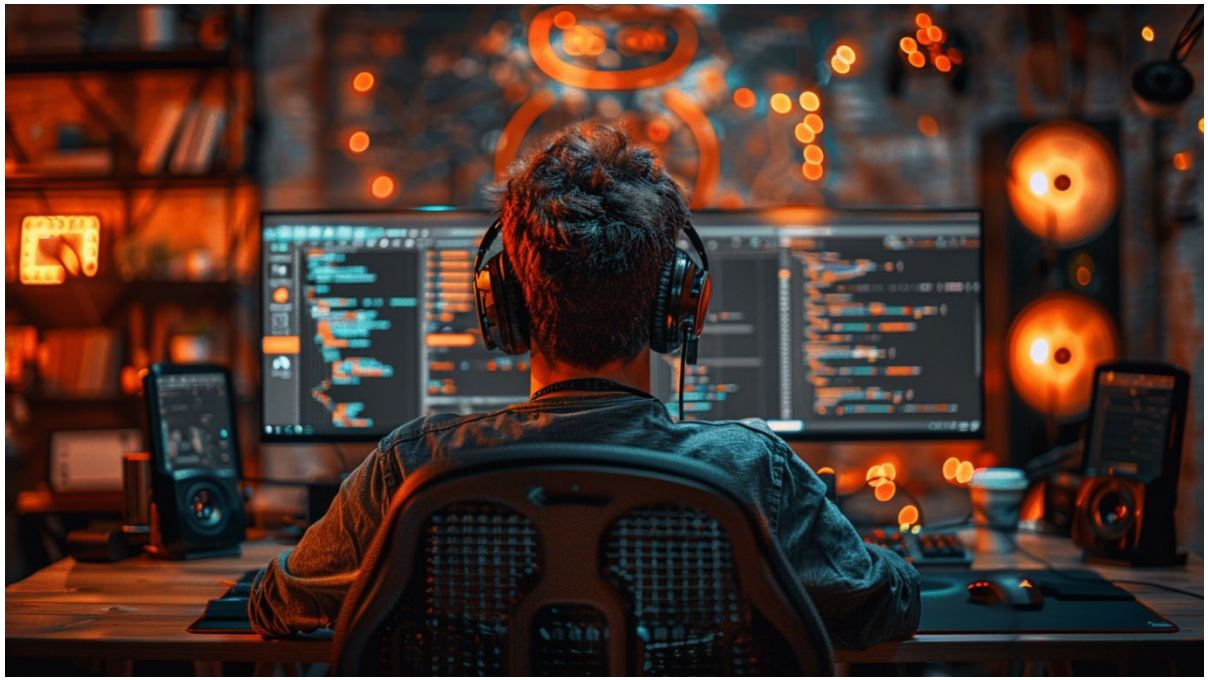
> ***AI helps generate basic project timeline and estimate in seconds***
>
> Dmitriy K., Project Manager at Intetics

Employing generative AI, it is possible to automate a variety of text-based tasks. For example, some AI assistants can search, retrieve, collect, and organize data while helping in content generation. As every software project generally starts with technical requirements, we expected to get a significant productivity boost at this stage.

As a result of the experiment, managers and business analysts revealed that GenAI was particularly helpful with:

- **Architecture ideation (20%-30% faster compared to the traditional approach).** AI allows developers to model and test multiple options simultaneously.
- **Evaluation of tradeoffs.** You can ask AI to evaluate all possible tradeoffs when preparing a technical specification.
- **Adaptability to requirements changes.** If a customer decides to change the requirements, for instance, feature delivery priorities, AI can estimate the impact of those changes on time effort and other project aspects.
- **Novel designs.** Based on initial design inputs, GenAI can help craft application prototypes that can be further used for early idea validation and customer feedback. Additionally, AI can simulate different user interactions and system behaviors to test design feasibility before actual software development starts.
- **Innovative brainstorming.** GenAI can provide ideas and frameworks to spur creative thinking, suggesting new approaches and features that might not have been considered otherwise. What's more, AI can provide existing design patterns and reusable components that align with new requirements, fostering innovative solutions built on proven methods.

Table of Content



As our experiment showed, AI assistants can bring **30%-40% time reductions** on selected tasks during the planning phase. However, it is worth noting that GenAI often struggles with correct interpretation of diagrams and infographics when used as input data. Meanwhile, since AI enables teams to boost productivity and creative thinking, we estimate the efficiency of AI assistants for this project stage as "High".

Table of Content

2. Solution Design

A design phase is another phase in software development where generative AI marks its transformative impact. Demonstrating a **10%-20% improvement in efficiency** for architectural and technical design tasks, AI-driven solutions are reshaping how teams approach this critical stage.

GenAI can automatically generate a software requirements document, outline a solution architecture, and even provide rough estimates on efforts needed for each step. Although, it is worth noting that top-of-the-line tools such as ChatGPT, Gemini, or Claude can only prepare text descriptions for a software product. Currently, they are not capable of creating complex technical graphics.

One important advantage of using GenAI in design, as described by Ben Evans, is its capability of providing "unlimited interns." This means AI comes with numerous junior helpers. Although those helpers are not very professional, they are tirelessly productive. Therefore, AI can swiftly generate thousands of feature variations in minutes. An architect can then select the three most promising options and proceed to their refinement.

Furthermore, the introduction of AI into the design phase enables a more iterative and dynamic design process. AI tools can quickly adapt designs based on user feedback or requirements changes, reducing the time necessary for revisions and adjustments. This agility enhances the team's ability to innovate and experiment, fostering a design environment that is both efficient and flexible.

3. Implementation (Code Writing)

The role of generative AI in the implementation stage may significantly vary depending on the project stage. If the team builds a feature from scratch, **the productivity boost might reach 50%-60%**.

In case software engineers spend most of their time on debugging, productivity gains will be much more modest, 5%-10%. However, all participants agree that AI assistants help automate various tasks, relieving developer time for other important activities.

Considering SDLC's strategies, it is possible to integrate GenAI into development environments in two primary ways, each having its advantages and disadvantages:

- **Using AI-powered tools**
  Embedded AI, such as GitHub Copilot, can be used in development environments like IDEs. This direct integration enhances efficiency as AI has instant access to the codebase, enabling it to provide contextually relevant suggestions specifically tailored to ongoing tasks. This setup also allows for the optimization of the AI's performance to match various demands and workflow, which can significantly increase productivity.
- **Using AI as a separate solution**
  Standalone AI solutions like ChatGPT operate independently on any development platform. Thanks to this flexibility, ChatGPT can be used across a variety of projects and environments despite the limitations of a single system. Basically, it is easier to scale standalone tools that can also be integrated with a broader range of services. Additionally, they offer a wider number of features and are accessible to a larger audience, which is beneficial for users who may not have access to specific solutions but still need advanced AI capabilities.

To choose between an embedded AI and a standalone AI solution, it is advisable to consider specific team structure and overall needs. For example, GitHub Copilot might be more suitable for engineers seeking an integrated experience with real-time, relevant contributions to their coding tasks.

In contrast, ChatGPT will be an ideal option for IT experts aiming to get a scalable and versatile tool that can support a broader scope of applications and environments. At Intetics, we use both approaches based on employee role, tasks, and project stage.

AI Opportunities & Limitations for Developers

What GenAI can do	What GenAI cannot do
✓ **Autocomplete Code** AI can suggest code completions based on context, which speeds up the coding process.	✕ **Fully Replace Developers** AI cannot fully understand business logic and specific project requirements without human involvement.
✓ **Detect and Fix Errors** AI is capable of analyzing code for common errors and advising corrections.	✕ **Solve Complex Logical Problems** AI may struggle with understanding context or making strategic decisions that are not obvious from the data.
✓ **Refactor Code** AI can suggest optimizations and simplifications to improve the code structure.	✕ **Ensure Complete Code Security** AI might miss complex security vulnerabilities that require an IT expert's insight.
✓ **Generate Code on Demand** AI can generate code templates or entire functions based on user specifications.	✕ **Operate Without Initial Data:** AI requires initial setup and configuration to perform tasks correctly and provide relevant results.
✓ **Provide Training and Tips** AI can offer educational materials and recommendations, for instance, on best software development practices.	

4. Testing

By employing AI technology, quality assurance engineers can significantly improve test coverage and overall testing speed, this way reducing product release time. With GenAI, it is possible to enable automated test generation and produce more comprehensive tests faster compared to manual methods while ensuring more extensive coverage across the application.

In addition, AI can elaborate data-driven test cases that are not only precise but also tailored to the solution's specific use scenarios, **improving testing efficiency by up to 40%**. This approach allows teams to achieve deeper insights and more robust validations at a quicker pace, which is crucial in today's fast-moving tech environment.

AI-driven testing plays a pivotal role in optimizing software quality. By using AI for early error detection, potential issues can be identified and addressed sooner, minimizing the risk of bugs in the production. Powered by generative artificial intelligence, security testing helps identify and prevent potential vulnerabilities that might be overlooked by QA experts.

On top of that, the ability of GenAI to analyze and draw insights from large datasets contributes to more accurate and actionable testing outcomes. This allows teams to deliver software products of higher quality, providing a competitive edge in the market.

The adoption of AI in testing environments offers advantages such as cost savings and increased scalability. By automating software debugging, AI cuts down expenses associated with prolonged testing cycles and post-release fixes. AI can effortlessly analyze and manage large volumes of test data and adapt to complex test scenarios without additional resource allocation, leading to enhanced scalability.

Furthermore, AI automates a wide range of repetitive tasks, enabling IT experts to focus on more critical aspects of software development and testing. This not only boosts productivity but also improves team collaboration, creating a more dynamic and responsive testing environment.

Table of Content

AI Opportunities & Limitations in Testing

AI opportunities for software testing	AI limitations for testing
✓ Creation of checklist/test coverage. Using AI, engineers can cut the time to analyze feature requirements and brainstorm full test coverage with team members. However, it is necessary to re-check AI test coverage.	✗ Creation of test coverage for a big feature. It's better to split into sub-tasks and create test coverage for each separately.
✓ Creation test-case. AI allows for reducing the time to write all steps manually.	✗ It is impossible to create full automation tests (in existed framework) or new framework from scratch for application. Can be used as assistant.
✓ Creation of a test scenario for automation test cases (not full auto-test implementation).	
✓ Verification of the existing test coverage.	
✓ Solving some specific automation testing tasks – it is easier and faster to get and debug ready solution from AI than searching in net.	
✓ Check and improve existing code in automation tests.	

5. Deployment and Maintenance

Employing artificial intelligence, software engineers can automate routine tasks and refine deployment methodologies, **increasing efficiency by up to 15%.** For instance, AI can speed up the configuration management process, making sure that each deployment environment is set up consistently without manual control.

This is particularly beneficial in complex systems where configuration discrepancies may lead to significant issues. For example, components may not behave as intended, leading to unpredictable system responses, crashes, freezes, or misconfigured access controls. On top of that, authentication mechanisms may leave a software system vulnerable to unauthorized access by hackers.

What's more, AI-driven tools like predictive analytics can forecast potential deployment failures by analyzing trends from past deployments, allowing teams to proactively adjust their strategies and avoid common problems.

Additionally, the role of AI in optimizing deployment workflows facilitates a more agile and responsive development environment. Using AI, it is possible to enable automated app scaling in cloud environments based on real-time traffic data, ensuring efficient resource allocation on demand.

This way, IT companies not only deliver more robust and scalable solutions but also reduce time to market. By accelerating mundane, yet important activities, AI lets software engineers focus on product enhancement and evolution, leading to faster, more reliable, and user-centric feature release cycles.

These specific use cases showcase how generative artificial intelligence transforms deployment and maintenance operations, helping organizations outrun rivals in highly competitive markets.

6. Conclusion

During our 3-month experiment, we **saved almost 200 hours of work for our clients**. We determined that AI-driven tools can bring value at each stage of the software development life cycle. While our team observed productivity gains for every project stage, we concluded that the outcome might vary dramatically, ranging from **5% to 50% in efficiency increase**.

Currently, Intetics is scaling GenAI assistant adoptions across the whole company to iterate faster, minimize bugs, and deliver more robust solutions. Our software engineers also outlined a roadmap for further AI integration, aiming to optimize every function within our software development process.

Generative artificial intelligence is transforming numerous industries, from IT, fintech, and insurance to real estate and logistics. After the experiment, we highly recommend that you integrate AI-powered tools to automate workflows, minimize human errors, and cut costs.


intetics
Where software concepts come alive™
INTETICS
MEANS YOUR
SUCCESS
Toll Free: +1 (877) SOFTDEV
US: +1 (239) 217-4907
DE: +49 (211) 3878-9350
UK: +44 (20) 3514-1416
www.intetics.com