

GENERATIVE AI'S POTENTIAL FOR INDUSTRIALIZING AND SCALING SOFTWARE DELIVERY

In this paper we assess the trustworthiness of AI-assisted software development, using the example of an information broker for an instant payments provider.

With its unprecedented ability to parse natural language and process extremely large and diverse datasets, generative AI promises to revolutionize the way businesses operate. But how reliable are the outcomes?

Capco is currently conducting a study to evaluate the potential of using generative AI to create assets required as part of the software delivery process. The study uses a real-life financial institution's setup to assess the quality and benefits of AI-generated content for the creation of an information brokerage service, based on an instant payments mechanism we have previously developed for a bank in response to regulatory change.

This analysis presents the promising early findings of our ongoing study, along with our recommendations for CIOs on how to take advantage of the new AI models to improve quality, timeliness and cost-effectiveness of software delivery.

EXPERIMENT OVERVIEW INFORMATION BROKER FOR A FINANCIAL SERVICES INSTITUTION

Objective

Our ongoing study aims to demonstrate the potential of generative AI to help CIOs transform their technology operating models across the software delivery process to:

- Improve quality of delivery
- Compress delivery timelines
- Keep operational costs under control
- Ensure regulatory compliance.

Workstreams and evaluation criteria

For the initial body of work, we focused on six common activities (disciplines) executed during a typical software delivery phase (Agile program increment), including user stories, requirements definition, architecture and design, user experience (UX), writing code, testing and DevOps.

For each of these activities we used tools developed by OpenAI (the creator of ChatGPT and the DALL-E image/art creation AI system) to generate assets that are typically created as part of the delivery lifecycle for both greenfield and brownfield software development projects. We then evaluated these outputs qualitatively against three dimensions – quality of product, time savings and resource/cost savings.

Six common disciplines of the software delivery process

- Writing requirements/ user stories
- Architecture and design
- Defining user experience
- Writing code
- Assisting with test activities
- Assisting with DevOps activities

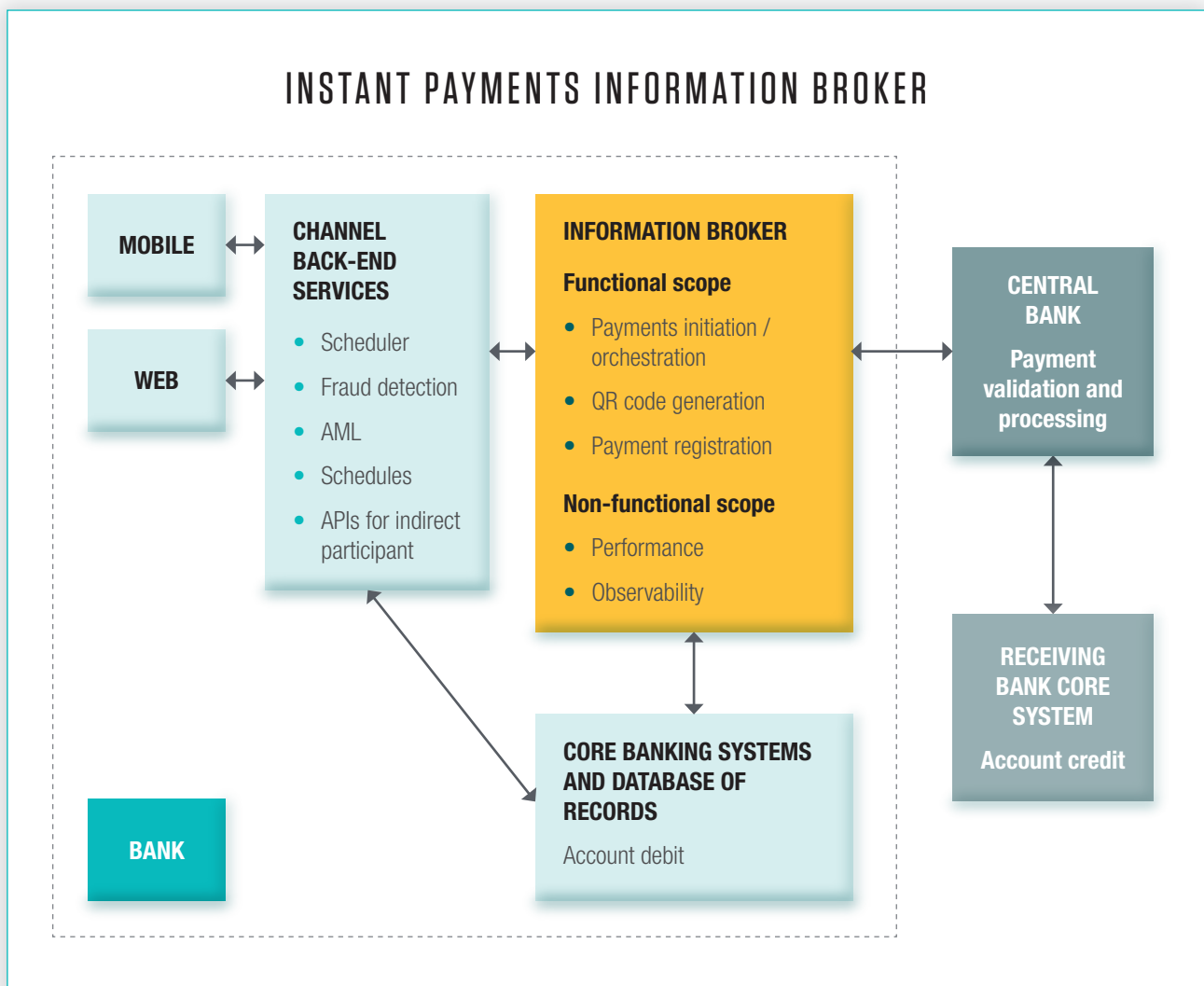


Evaluation criteria

- Quality of output
- Time required to generate
- Cost compression in terms of human input

Basis

For our experiment, we used a case for creating a service stack that acts as an information broker for a financial services institution facilitating instant online payments (based on a real-life instant payments system we have previously designed and implemented at a major bank). The purpose of the information broker is to facilitate information exchange between the financial institution's core banking and digital channel systems and the Central Bank, to allow end-users – the bank's customers – to make instant payments.



Sample prompts

For each task in the creation of the information broker service, we used natural language prompts, discovering in the process that the more precise the prompt, the more accurate the answer.

DISCIPLINE	SAMPLE PROMPTS USED
Writing requirements	<ul style="list-style-type: none">• Create a user story for key payments using email as identification.
Architecture & design	<ul style="list-style-type: none">• I want to create middleware as a service for an instant payments service. Please create for me a high-level architecture at L1 for the solution ecosystems.• Can you describe a sequence diagram at this most granular level that encompasses an instant payment transaction?
Defining user experience	<ul style="list-style-type: none">• Can you create for me a user experience optimized for mobile around conducting an instant payment?• Can you create basic screen mock-ups of the above user experience?
Writing code	<ul style="list-style-type: none">• Create a microservice that will use Java with Spring Boot (to a detailed spec including goals, access, encryption and decryption algorithms).
Assisting with test activities	<ul style="list-style-type: none">• List the combinations of input data that you see as sufficient to test the given parameters (e.g. 'A specific amount a customer can transfer') using equivalence partitioning and threshold value analysis.• Now describe these test cases using Gherkin (behavior driven development method).
Assisting with DevOps	<ul style="list-style-type: none">• Create a TerraForm script for the creation of a new development treadmill using a docker container (to a detailed criteria).

HOW AI PERFORMED

We found that the biggest benefit of using generative AI for all six activities was around the realization of time savings, with code-writing benefitting the most collectively across all three evaluation criteria.

On the scale of one to five stars, our findings are summarized below.

QUALITATIVE EVALUATION OF AI-GENERATED ASSETS AND OUTPUTS			
Software delivery process discipline	Quality of output	Time savings	Cost compression
Requirements/ user stories	★★★★☆	★★★★☆	★★★☆☆
Architecture and design	★★★★☆	★★★★☆	★★★☆☆
UX	★★★☆☆	★★★★☆	★★★☆☆
Code	★★★★☆	★★★★★	★★★★☆
Testing	★★★★☆	★★★☆☆	★★★☆☆
DevOps	★★★★☆	★★★★☆	★★★☆☆

RATINGS EXPLAINED

Rating	Quality of output	Time savings	Cost compression
★☆☆☆☆	AI outputs provide misleading or incorrect information that requires significant human input to review and analyse	5% reduction of time on asset / output generation	10% reduction of cost associated with human labour
★★★★★	AI outputs can be fully trusted and used without any human oversight or review	80% reduction of time on asset / output generation	50% reduction of cost associated with human labour

INTERPRETING RESULTS

BENEFITS OF USING AI FOR SOFTWARE DELIVERY

The OpenAI suite of tools particularly impressed us with how it helped improve the quality of requirements, user stories and code. The tool also dramatically reduced the time needed to generate user stories and code as well the time required to execute routine DevOps tasks.

Even in those areas where the AI performance was inferior, the work products the AI models generated were still useful in terms of performing sanity checks against the completeness of human generated assets. Having the machine generated assets helped shift-left defect discovery to much earlier in the delivery lifecycle which, when applied at scale, will reduce churn and rework, yielding tangible IT delivery cost savings.

Based on our experiment so far, the benefits of using Generative AI across the software delivery process are as follows:

- The biggest benefit is clearly in time savings in generating work products.
- Code generation benefits most strongly across each dimension – quality, time savings and human input/resource reduction.
- The average score across all three dimensions is sufficiently high to warrant investing time and effort in using generative AI models towards industrializing the software development lifecycle, particularly as AI models are becoming increasingly more sophisticated with each market release.
- Extrapolating against these outcomes, we foresee a future with a far leaner workforce focused on guiding AI tools and validating work products generated by AI.

OUR ONGOING RESEARCH

This study represents our first steps in understanding how generative AI may help shape the future of technology development. Encouraged by the initial results, we plan to focus on two key areas as immediate next steps:

Legacy modernization

Perform a deep-dive on how Generative AI can help with legacy transformation – particularly in joining the power of the new technology with our methodology for optimizing technology stack when ‘slicing’ monolithic applications into microservices (described in our paper [Decomposing the Monolith: Optimization and Automation](#)). Using the two concepts together will allow for full end-to-end automation of the legacy modernization process.

Wider IT projects

Thinking beyond the software delivery process, our experimental efforts will extend to proof-of-concepts (POCs) to validate the AI models’ capability to accelerate and mature the implementation of supporting project sub-disciplines that bring tangible business value. Specifically, opportunities for investigation include:

- Analysing and understanding natural language queries from IT teams, enabling them to quickly find solutions to issues through training the models on corporate log data.
- Developing a knowledge management system to enable IT teams to quickly find and access relevant information.
- Investigating the effectiveness of AI models in identifying and mitigating project risks, such as schedule delays, quality issues, and security concerns.

Finally, we plan to continue with the AI-assisted software delivery process study as new versions of the AI tool suite and/or competing products are released in the market, particularly as aspects of artificial general intelligence (AGI) mature and AI tools become better at performing concept abstraction and making inferences based on incomplete or ambiguous information.

OUR RECOMMENDATIONS FOR CIOs

Today’s software delivery is largely manual, however emerging technologies such as generative AI make automation achievable and scalable. In addition, AI’s capabilities are increasing exponentially, which will give its application unlimited possibilities.

We offer three key recommendations for CIOs looking to increase the level of automation of software delivery processes, thus accelerating time-to-market and saving costs without compromising on quality and compliance:

- 1. Systematic – and realistic – transformation:** Create a systematic ‘change roadmap’, starting from human-driven software delivery and leading to automated, AI-driven software delivery. Identify easy win/no regret points and also the biggest pain points and efficiency ‘drains’. Those would be the projects to prioritize for AI-assisted development, as they will have the biggest impact. Once the early pilots take off, the resource savings generated as a result could be re-invested into the next wave, meaning that the program can become partially self-funded after the first round.

2. Emerging skillsets: Increased automation will transform the workforce composition, so mapping out your future workforce and investing in relevant skills training is important. These will include prompt engineering (effectively communicating with AI models), training generative models on proprietary corporate information, and monitoring the AI-generated outputs.

3. Local instances of ChatGPT/Codex: Today's generative AI models are trained using the wealth of knowledge available on the internet. While this is a lot of information, firms would benefit further from having proprietary data available for educating AI. This will provide highly specific and valuable insights, enabling artificial intelligence to generate smarter answers. It will also minimize the need for using cloud-based versions of the AI tool suite, reducing the risk of data leaks and non-compliance.

CONCLUSION

Transforming and automating the software delivery process using AI-assisted approaches will help CIOs achieve results that were previously not possible, specifically in the areas of:

- De-risking legacy modernization initiatives
- Enabling smart enterprise data management (i.e. exploiting transactional, analytical and operational data towards better decision making while safeguarding corporate reputation)
- Facilitating faster and more efficient digital transformation.

Given the rapid pace of development of generative AI models, their potential across IT and business areas promises to be

a game-changer for many industries, and the results of early pilots are encouraging.

Even firms that are in the early stages of modernizing their legacy systems will benefit from initiating POCs using generative AI to be better prepared for when new and more powerful models become available.

Contact us to find out more and discuss your ideas and visions for finding practical solutions to accelerate software delivery, while improving quality and saving time and resources.

AUTHORS

Gerhardt Scriven, Managing Principal

Diego Sarai, Managing Principal

Diogo Santos, Principal Consultant

Julio Lima, Principal Consultant

Graziella Bonizi, Principal Consultant

CONTACT

Alessandro Corsi, Partner

E Alessandro.corsi@capco.com

WWW.CAPCO.COM

