Impact of the Availability of Chat-GPT on Software Development

Alexander Quispe[1,2] Rodrigo Grijalba[1]

[1]Pontificia Universidad Católica del Perú, Departamento de Economía

[2]Banco Mundial

December 1, 2023

Table of Contents

Introduction

- Utility of LLMs in software development: models can be trained with sample text that contains code.
- LLMs then can generate responses to programming-related queries through patterns found in sample text.
- Most important example: Chat-GPT, which became the fastest application to reach 100 million users earlier this year.[1]
- Research question: How important has Chat-GPT been for software development? In our case, specifically: what has been the impact of Chat-GPT availability on the number of pushes to GitHub?

[1] Reuters, 2023

Hypothesis

- Chat-GPT can assist in the problem-solving and implementation of solutions for software development. This would make an individual developer faster/more productive at constructing the code for the project they're working on
- On the other hand, access to Chat-GPT was restricted in several countries.[2]
- Therefor, we would expect to see larger growth in pushes in countries that had access to Chat-GPT, compared to those who did not.

[2] OpenAI, 2023

Potential of ChatGPT in Software Development

Abu Jaber, Beganovic & Abd Almisreb (2023):

- Potential applications of ChatGPT in software development: troubleshooting, bug repair, optimizing programs and numerical algorithms, inclusion as integrated controller in Generalized Intelligence.
- Creation of software solutions and architectures through evaluation of multiple responses and a dialogical process
- Importance of Prompt Engineering in using ChatGPT for software architectural design.

Potential of ChatGPT in Software Development (cont.)

Rahmaniar (2023)

- Potential productivity gains in SD through programming assistance, training, code review, and client interaction
- ChatGPT's natural language processing, generative capacity, dialogical interactivity, and Open Source contributions.
- Potential downsides: code incompleteness, security vulnerabilities, malware creation, ethical implications related to inherent biases.
- Possible future improvements: enhanced performance and real-time code optimization.

Tests of ChatGPT's Capabilities

Ahmad, Waseem, Liang, Fehmideh, Aktar & Mikkonen (2023)

- Test on human-bot collaboration for outlining software architecture, starting with an architecture story describing software conditions.
- A novice software architect engages in a dialogical process with the chatbot to analyze, synthesize, and evaluate potential architectural solutions.

Sobania, Hanna, Briesch & Petke (2023)

- Test of ChatGPT's code repair capabilities with QuickBugs database: ChatGPT's performance comparable to specialized LLMs and better than a standard, NN-based Automated Repair Program (about 50%).
- Dialogue queries for clarification increase success rate to 78%.
- Potential training sample bias may be a limitation (Zhang, et al.; 2023).

Tests of ChatGPT's Capabilities (cont.)

Zhang, Zhang, Zhai, Fang, Yu, Sun & Chen (2023)

- Assessment of ChatGPT's code repair capabilities using a new database, EvalGPTFix, created from programming contest problems on AtCode to address potential training sample biases.
- With simple prompt: 109 out of 151 programs repaired
- Total with modified prompts that target specifics about each problem: 143 out of 151 programs repaired

ChatGPT’s Impact on Software Development

Gallea (2023)

- Contrasts the Stack Overflow pages for Python and R: Python is more popular and got more training material for ChatGPT, while ChatGPT showed less efficiency in answering R-related queries.
- Findings suggest a decrease in quantity, increase in quality (average score), and a decrease in the proportion of resolved questions for Python compared to R.

Saguu & Ante (2023)

- Study on returns of artificial intelligence-related crypto-assets using daily price data from Coingecko and CoinMarketCap; launch of ChatGPT marks the start of the post-treatment period.
- DiD and SDID methods: launch of ChatGPT positively impacted the return of AI-related assets, compared to non-AI related assets.

ChatGPT’s Impact on Software Development (cont.)

Demirci, Hannane & Xinrong (2023)

- Study of the influence of ChatGPT on demand for freelance services by assigning an AI exposure index to each service type and using the Google Search Volume Index for adjustment.
- DiD method: negative impact of ChatGPT on service postings in areas with high AI exposure, as compared to low-exposure areas.

Del Rio-Chanona, Laurentsyeva & Wachs (2023)

- Impact evaluation of ChatGPT on question and answer platforms: Stack Overflow vs. math-focused Math Exchange and Stack Overflow’s Russian and Chinese analogues, where ChatGPT access is restricted.
- DiD: decrease in the weekly number of posts, questions, and weekday posts on Stack Overflow compared to the other platforms.

ChatGPT's Impact on Software Development (cont.)

Kreitmeir & Raschky (2023)

- Analysis on the productivity impact in software development using daily frequency user-level data from GitHub, comparing Italy - where ChatGPT is banned - with France and Austria.
- Impact on usage of Tor browser; Tor sistem can be used to bypass bans.
- DiD: significant negative impact on the probability of Italian users releasing new software each day, suggesting lower productivity as a result of the ban. No overall impact on Tor usage

Data

- Quarterly panel data on GitHub pushes per country taken from GitHub's Innovation Graph Initiative repository.[3]. This data spans 178 countries and 13 quarters (2020-Q1 - 2023-Q1)
- Data on ChatGPT availability from OpenAI's documentation (see footnote 3).
- We define the target variable as the number of pushes per 100 000 inhabitants in each country. We consider the start of the treatment to be Q4 of 2022

[3] GitHub, 2023

Methodology

We are looking to ascertain the existence and magnitude of causal effects of ChatGPT availability on volume of pushes to GitHub, and the data provide us a control and a treatment group of aggregate (country)-level measurements. Therefor, we consider the following causal inference methods:

- Difference in Differences (DiD): Simple implementation and interpretation, although it relies on the Parallel Trends Assumption (PTA) for the data, which can be hard to argue in favor of.
- Synthetic Control (SC): Formalizes the argument for the PTA, which reduces bias in estimation.
- Synthetic Difference in Differences (SDID): Adds a difference in the intercept as well as weights for pre-treatment periods for a more robust estimation.

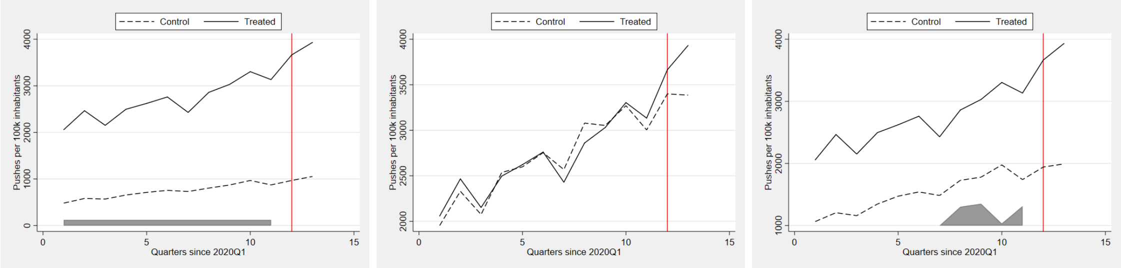
Main Results

	Coef.	s.e.	z	Prob $< \|z\|$	95% C.I.
DID	849.9***	139.06	6.11	0.00	[577.34, 1122.45]
SC	405.92	501.68	0.81	0.42	[-578, 1390]
SDID	568.7***	138.69	4.1	0.00	[296.88 , 840.52]

Note: * $p < 0.1$; ** $p < 0.05$; *** $p < 0.01$. Standard errors calculated through 100 bootstrap replications.

Table: Model results

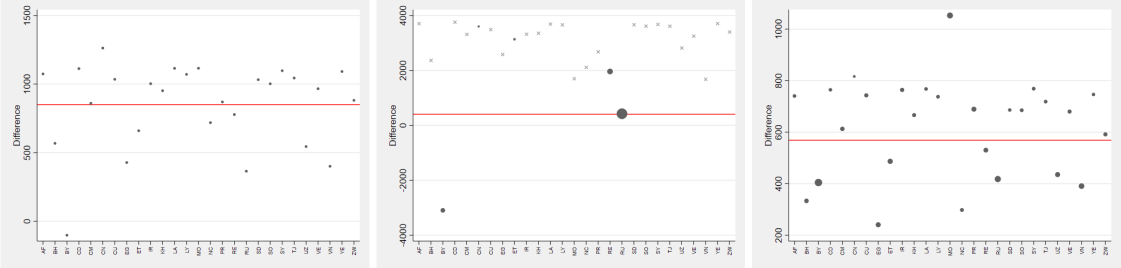
Estimated Trends



(a) Estimated DiD trends (b) Estimated SC trends (c) Estimated SDiD trends

Figure: Estimated Trends

Estimated Weights



(a) Estimated DiD weights (b) Estimated SC weights (c) Estimated SDiD weights

Figure: Estimated Weights