

Utilização de métricas do GitHub para aumentar a produtividade de times de desenvolvimento de software

Danielly Novaes Lima^{1*}; Abdoulaye Aboubacari Mohamed²

¹ Bacharel em engenharia química. Rua dona maria de angelis, 278, casa A – Afogados; 5085080, Recife, Pernambuco, Brasil.

² Universidade Federal de Viçosa. Doutor em Economia Aplicada. Avenida Peter Henry Rolfs, s/n-Campus Universitário, Viçosa, Minas Gerais, 36570-000, Viçosa, Minas Gerais, Brasil

*autor correspondente: daniinovaes@gmail.com

Utilização de métricas do GitHub para aumentar a produtividade de times de desenvolvimento de software

Resumo

O GitHub como maior host de código-fonte que usa versionamento e é utilizado de forma colaborativa por milhões de desenvolvedores ao redor do mundo. Considerando a temática da análise de métricas de projetos de desenvolvimento de software, a disponibilidade de uma API abrangente fez com que o GitHub se tornasse uma ferramenta de pesquisa de engenharia de software e colaboração online. Com o objetivo de compreender os padrões e práticas que permeiam o desenvolvimento de software nesse contexto, o presente estudo se propõe a investigar o uso de métricas relacionadas a pull requests, tempo de revisão, frequência de commits, taxa de erros, estabilidade, diversidade de contribuidores em repositórios hospedados no GitHub da openAI. Além disso, também se busca analisar a relação entre essas métricas e fatores como o tempo de correção de bugs e inclusão de novas features. Para tanto, será empregada uma metodologia de pesquisa que compreende a coleta de dados, análise exploratória e a aplicação de técnicas de mineração de dados para identificar padrões e relações entre as variáveis investigadas. Espera-se que os resultados obtidos possam contribuir para uma melhor compreensão do processo de desenvolvimento de software colaborativo e para a definição de práticas que possam melhorar a eficiência e qualidade desses projetos. **Palavras-chave:** GitHub, métricas de software, mineração de dados, estatística descritiva, gestão de projetos.

Using GitHub metrics to boost the productivity of software development teams.

Abstract

GitHub, as the largest source code hosting platform that employs version control and is collaboratively used by millions of developers worldwide. Considering the theme of analyzing metrics in software development projects, the availability of a comprehensive API has turned GitHub into a tool for software engineering research and online collaboration. With the aim of understanding the patterns and practices that pervade software development in this context, the present study seeks to investigate the use of metrics related to pull requests, review time, commit frequency, error rate, stability, and contributor diversity in repositories hosted on GitHub by OpenAI. Additionally, it also aims to analyze the relationship between these metrics and factors such as bug resolution time and the inclusion of new features. To achieve this, a research methodology will be employed, encompassing data collection, exploratory analysis, and the application of data mining techniques to identify patterns and relationships among the variables under investigation. It is expected that the obtained results can contribute to a better understanding of collaborative software development processes and the definition of practices that can enhance the efficiency and quality of these projects.

Keywords ou Palabras Clave: GitHub, software metrics, data mining, descriptive statistics, project management.

Introdução

Criar um produto complexo é um grande peso para as empresas. Sempre que as pessoas estão envolvidas em um esforço criativo e complexo, os métodos tradicionais de gerenciamento não funcionam como deveriam. Sabendo que o sucesso de uma empresa,

principalmente no setor de tecnologia, depende majoritariamente de um método de gerenciamento dinâmico que mede a eficiência e a produtividade das equipes de desenvolvimento (Sutherland, 2014). De acordo com Ciric et al., 2018, o ambiente de negócios atual é caracterizado como volátil, incerto, complexo e ambíguo (VUCA), marcado por alta concorrência global, rápidos avanços tecnológicos, dinamismo das oportunidades e a complexidade dos processos de negócio. Os times de tecnologia enfrentam constantes desafios, como a complexidade crescente dos sistemas, a integração de diferentes tecnologias, escopo extenso, mudanças nos requisitos e nas prioridades do negócio, documentação insuficiente, distância geográfica do time, a escassez de talentos e a gestão eficiente (Nerur et al., 2005). A pressão por inovação e a rápida evolução das tecnologias exigem dos times a habilidade de aprender e se adaptar de maneira muito dinâmica.

Com cada vez mais dados disponíveis, o data-driven decision-making (DDD) surge como uma importante aliada na otimização de processos e na tomada de decisões em diversos setores, incluindo o desenvolvimento de software. Estatisticamente, quanto mais uma empresa é orientada por dados, mais produtiva ela é, e as diferenças não são pequenas, a utilização do DDD está associada a um aumento de 4% a 6% na produtividade (Provost & Fawcett, 2013).

E com a crescente do uso de dados, a ciência de dados que é uma disciplina multidisciplinar tem um papel fundamental na tomada de decisões informadas. Por meio do uso de técnicas estatísticas, de mineração de dados e de aprendizado de máquina, a análise dos dados permite a identificação de padrões e a extração de insights valiosos a partir de grandes volumes de informações (Kelleher & Tierney, 2018). No livro *Perspectives on Data Science for Software Engineering* de Menzies et al. (2016) demonstra como a análise de dados é uma ferramenta valiosa para melhorar a tomada de decisão no desenvolvimento de software, permitindo que as organizações identifiquem e priorizem os problemas (gargalos), melhorem o processo de desenvolvimento e avaliem o impacto das mudanças no software.

Nesse contexto, o GitHub é um exemplo icônico de plataforma de trabalho baseado em conhecimento. Ela tem como objetivo a hospedagem de código-fonte, amplamente utilizada no desenvolvimento de software, facilitando o versionamento, a colaboração e o compartilhamento de projetos (Dabbish et al., 2012). A análise de métricas de repositórios de código-fonte, pode oferecer insights significativos para a gestão de projetos e o aprimoramento da produtividade das equipes, auxiliando na identificação de possíveis problemas e na tomada de decisões em relação a futuras melhorias (Gousios, Pinzger & Van Deursen, 2014).

Diante desse contexto, a avaliação da produtividade de times de desenvolvimento por meio da análise de métricas no GitHub pode ser um caminho para identificar gargalos e

oportunidades de melhoria no processo de desenvolvimento de software. O objetivo deste estudo é analisar os dez repositórios mais populares da openAI no GitHub, utilizando técnicas de ciência de dados, para identificar padrões e tendências, com foco em entender onde os esforços de desenvolvimento estão concentrados, identificar possíveis gargalos e melhorias no processo e também avaliar a efetividade das práticas de desenvolvimento adotadas. A investigação dos fatores que influenciam a produtividade e a qualidade dos projetos de software pode auxiliar as empresas a tomar decisões mais embasadas dos gestores e líderes de equipe, impulsionando a eficiência e a eficácia de suas equipes, sendo assim, espera-se que os resultados possam contribuir para a compreensão dos processos de desenvolvimento colaborativo de software e apontar possíveis caminhos para aprimorá-los.

Material e Métodos

Para a realização deste estudo sobre a utilização de métricas do GitHub da openAI para aumentar a produtividade de times de desenvolvimento de software, foram adotados os procedimentos ilustrados na Figura 1:



Figura 1. Fluxo do processo da análise desta pesquisa de estudo

Fonte: Resultados originais da pesquisa

1. Seleção da população estudada: foram analisados os dez repositórios destinados a produtos mais populares. A amostragem foi realizada de forma não probabilística por conveniência, selecionando-se os repositórios que atendiam aos critérios técnicos, de popularidade e relevância para o estudo.

2. Coleta de dados: os dados foram obtidos por meio de scraping utilizando a API do GitHub e o software JupyterLab.

3. Extração, transformação e carga: para a análise dos dados, foram utilizadas técnicas de ciência de dados, incluindo a limpeza e pré-processamento dos dados, análise exploratória dos dados, modelagem e mineração de dados, e a visualização dos resultados. A análise dos dados foi realizada com o uso da linguagem de programação Python e das bibliotecas Requests, Json, Time, DuckDB, WordCloud, Pandas, NumPy, Scikit-learn, Seaborn e Matplotlib.

Extração de métricas: foram coletadas diversas métricas dos repositórios, como: o número de estrelas, forks, releases, contribuidores, frequência de commits, mudanças no código, Ciclo médio da pull requests, frequência das pull requests, taxa de fechamento da pull request, taxa e tempo de resolução das issues, frequência das issues e, além de informações sobre as linguagens de programação utilizadas e muitas outras que podem ser extraídas analisando os dados do projeto.

4. Identificação de padrões e tendências: As métricas extraídas foram utilizadas para identificar padrões e tendências nos processos de desenvolvimento de software, bem como para avaliar a efetividade das práticas de desenvolvimento adotadas.

5. Apresentação dos resultados: os resultados foram apresentados por meio de tabelas e gráficos que mostram as principais métricas dos repositórios analisados, bem como as correlações entre as métricas. A análise dos resultados permitiu identificar os principais pontos de melhoria nos processos de desenvolvimento de software e fornecer insights valiosos para a gestão de projetos.

De acordo com Gil (2002), uma abordagem exploratória se concentra em elaborar hipóteses, validar instrumentos e proporcionar maior familiaridade com o problema, assim como este projeto de pesquisa. Quanto ao procedimento técnico, este estudo seguiu uma abordagem quantitativa e foi realizado como um estudo de caso. Para a análise foram aplicadas técnicas de mineração de dados, técnicas de estatística descritiva, transformação de dados e storytelling de dados. Cabe ressaltar que todos os procedimentos adotados neste estudo foram baseados em metodologias e técnicas previamente descritas na literatura, tendo sido realizadas adaptações e modificações para atender às especificidades do estudo em questão.

Toda informação encontrada foi originalmente obtida através da API do GitHub e foram utilizadas diversas bibliotecas que desempenharam papéis essenciais na coleta, análise e visualização dos dados. A biblioteca "requests" foi empregada para acessar informações na API. Para a manipulação de dados, "pandas" e "numpy" foram fundamentais. As visualizações foram geradas com o auxílio do "matplotlib" e "plotly.graph_objs", proporcionando gráficos informativos e interativos. Além disso, a biblioteca "xlsxwriter" viabilizou a geração de planilhas para documentação, produção das

tabelas e análise de algumas métricas. A persistência de dados foi realizada com o auxílio da biblioteca "duckdb", que proporcionou o armazenamento eficiente dos dados em um banco de dados. Por fim, o uso de "concurrent.futures" permitiu a execução paralela de tarefas, otimizando o processamento de informações em larga escala.

Para desenvolvimento desta pesquisa, se fez necessário o uso da linguagem de programação Python, que é uma linguagem de programação funcional, orientada à objetos e voltada à manipulação, análise e visualização de dados. A versão do Python utilizado foi "Python 3".

Resultados e Discussão

Dados dos Repositórios Selecionados:

A OpenAI mantém um total de 153 repositórios em seu portfólio, dos quais selecionamos 10 para análise. A seleção foi baseada principalmente na métrica de estrelas que cada repositório recebeu. Cada um desses repositórios possui suas peculiaridades e um histórico único de atividades ao longo dos anos. A Tabela 1 abaixo oferece uma visão resumida desses repositórios:

Tabela 1. Repositórios selecionados

Repositório	Criação	Estrela	Fork	Branch	Contribuidor	Commit		
						Total	Média	2023
whisper	2022	45324	5198	5	59	123	61	66
gym	2016	32789	8606	124	399	1757	220	2
chatgpt-retrieval-plugin	2023	19676	3606	5	35	99	99	99
CLIP	2020	17891	2555	2	20	55	14	3
baselines	2017	14811	4805	36	117	347	50	0
evals	2023	12000	2267	55	418	604	604	604
openai-python	2020	11307	1704	7	80	182	46	98
DALL-E	2021	10469	1915	1	1	3	1	0
shap-e	2023	10421	779	1	7	13	13	13
triton	2014	8354	904	61	178	1338	134	749

Fonte: Resultados originais da pesquisa

Ao analisarmos os repositórios selecionados, notamos que alguns deles, como "openai-cookbook," "gpt-3," "gpt-2" e "spinningup," foram excluídos da análise por servirem

como exemplos ou guidelines, em vez disso, incluímos "DALL-E," "sharp-e" e "triton." Os números que coletamos fornecem alguns insights valiosos:

- **Alta Quantidade de Contribuidores:** A presença de um grande número de contribuidores em um repositório pode indicar sua vitalidade e capacidade de atrair colaboradores interessados no projeto. Um exemplo notável é o repositório "evals," que, apesar de ser relativamente novo, tem atraído um grande número de contribuidores.
- **Número de Branches:** O número de branches em um repositório pode ser um indicador da complexidade e do volume de trabalho envolvido. Um grande número de branches pode sugerir desafios na manutenção do código e na gestão do projeto, como é o caso do "gym."
- **Número de Commits:** A quantidade de commits reflete a intensidade da atividade e da colaboração em um projeto. Como observamos o repositório triton um alto número de commits ao longo do tempo sugere que o projeto está ativo e continuamente mantido.

Ciclo de Tempo de Pull Requests:

O Ciclo de Tempo é uma métrica crucial para identificar ineficiências no processo de desenvolvimento. Times de elite frequentemente têm um ciclo de entrega de código de menos de 1 dia, enquanto times de alta performance o mantêm abaixo de 7 dias. Essa métrica é um indicador de potenciais problemas que precisam ser abordados.

A análise do ciclo de tempo das Pull Requests revelou algumas descobertas notáveis, conforme mostrado na Tabela 2:

Tabela 2. Ciclo de tempo da Pull Request.

Ano	PRs	Desenvolvimento	Revisão	Integração	Média do ciclo
2021	501	20 days 07:33:42	60 days 19:27:15	6 days 10:56:18	37 days 02:01:03
2022	877	8 days 03:07:54	10 days 04:09:18	3 days 17:00:46	10 days 18:37:57
2023	2349	13 days 03:38:03	7 days 23:50:40	2 days 02:56:40	19 days 22:52:51

Fonte: Resultados originais da pesquisa

Em análise mais profunda de 2023, observamos o ciclo de tempo se distribui como demonstrado na Tabela 3:

Tabela 3. Ciclo de tempo das pull requests de 2023 para cada repositório.

Repositório	Desenvolvimento	Revisão	Integração	Média do ciclo
whisper	11 days 08:59:06	15 days 17:54:16	6 days 20:51:37	13 days 11:31:56
gym	21 days 09:43:55	89 days 01:14:49	0 days 00:01:26	45 days 07:02:57
chatgpt-retrieval-plugin	5 days 04:24:02	3 days 10:54:11	1 days 04:13:51	7 days 19:48:02
CLIP	-	-	0 days 00:00:00	8 days 08:54:10
evals	22 days 06:47:10	10 days 15:13:43	1 days 17:42:52	31 days 04:04:14
openai-python	6 days 08:50:12	4 days 00:53:52	2 days 00:14:47	16 days 15:52:08
DALL-E	-	-	0 days 00:00:00	18 days 06:21:28
shap-e	0 days 18:06:53	0 days 19:49:04	0 days 00:06:48	17 days 01:47:43
triton	1 days 13:53:33	4 days 15:27:52	2 days 08:38:04	16 days 22:03:05

Fonte: Resultados originais da pesquisa

Alguns insights importantes incluem:

- **Eficiência na Revisão de Código:** Houve um aumento significativo na eficiência da revisão de código ao longo dos anos, com uma impressionante taxa de revisão de 66.02% em 2023, em comparação com 16.8% em 2021. Isso indica melhorias na qualidade e colaboração da equipe.
- **Redução no Tempo Médio de Ciclo:** O tempo médio de ciclo de PRs diminuiu de 37 dias em 2021 para 19 dias em 2023, embora em 2022 tenha atingido uma marca notável de 10 dias. No entanto, o aumento em 265% das PRs de 2022 para 2023 sugerindo desafios na gestão de projetos, especialmente para os repositórios "gym" e "evals."
- **Variação no Tempo de Revisão ao Fechamento:** Embora tenha havido variações no tempo médio de revisão ao fechamento ao longo dos anos, observamos uma tendência positiva, com uma redução drástica em 2023. Isso pode refletir dinâmicas internas de revisão.
- **Falta de Utilização do Ciclo de Tempo:** Cerca de 46.7% das PRs têm datas de abertura, fechamento e merge, mas a falta de dados pode afetar a análise, especialmente em 2023 com "DALL-E" e "CLIP."

Compreensão dos Commits, Issues e PRs:

A análise de commits, issues e PRs é essencial no desenvolvimento de software, pois oferece insights valiosos sobre a dinâmica da equipe, evolução do projeto e alocação de recursos.

(a) Dinâmica dos Repositórios

Durante a análise, observa-se uma oscilação marcante nos repositórios, um refletindo o aumento da maturidade da OpenAI de 2016 a 2023. Em nossa análise aos 10 repositórios selecionados, notamos um espectro amplo de atividade de commit, com picos em lançamento de projetos e uma crescente geral após o lançamento do chatGPT.

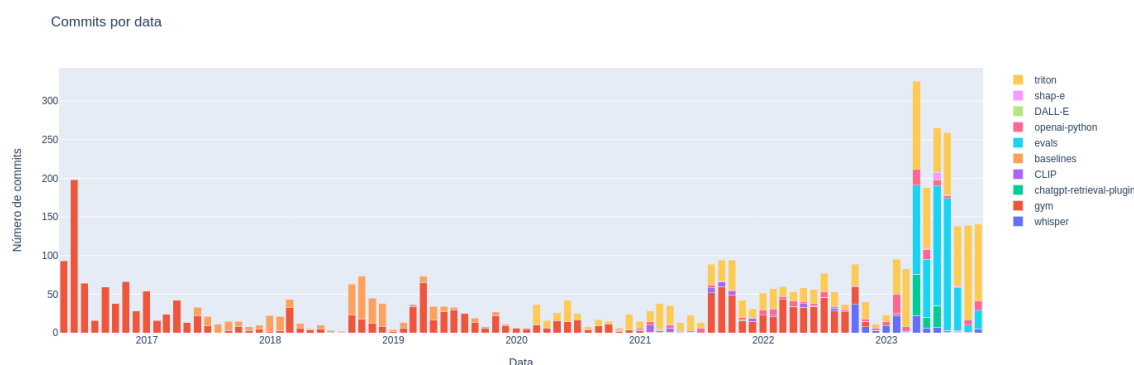


Figura 2. Fluxo de comits realizados por repositório

Fonte: Resultados originais da pesquisa

Como mostrado na Figura 2 alguns repositórios, como o como 'gym', resistiram ao teste do tempo com sua constância, embora com uma leve redução gradual com o tempo. Em contraste, 'evals', o mais jovem de nossa amostra, emergiu como líder em atividade em 2023. Entretanto, 'DALL-E' e 'gpt-2' testemunharam períodos de inatividade relativa. Essa diversidade de dinâmicas reflete nuances como a maturidade do projeto, o tamanho da equipe, a complexidade da empreitada e a natureza das contribuições.

(b) Commits e sua Ligação com Pull Requests

Uma descoberta-chave foi a irregularidade no uso de Pull Requests (PRs). Surpreendentemente, cerca de 26,6% dos commits surgiram sem uma PR correspondente. Isso levanta questões sobre a utilização plena dos benefícios das PRs, que são essenciais para a revisão do código e colaboração.

Já no restante dos commits, que estão vinculados a uma PR, apenas 29.3% possuem somente um commit, enquanto apenas 0.7% das PR possuem cinco ou mais commits. Além disso, nota-se que a maioria das PRs relacionadas a commits tinha em média apenas 2,15 commits, sugerindo um foco predominante em alterações pequenas e incrementais. Isso aponta para a necessidade de aprimorar a gestão de tarefas mais substanciais e uma baixíssima proporção de PRs contendo cinco ou mais commits.

Foi observado uma tendência crescente de vincular commits a Pull Requests (PRs). Em 2021, essa proporção era relativamente baixa, de 60% em 2021 mas aumentou consistentemente ao longo dos anos chegando em 94% em 2023. Assim a média de

commits por PRs desceu para 1,26 em 2023 com somente 14% de commits únicos e 0,2% de PRs com 5 ou mais commits. Isso reflete um foco crescente na revisão e colaboração no processo de desenvolvimento e sugere que há espaço para melhorar a gestão de tarefas de maior porte.

(c) Desafios da Expansão e a Sobrecarga de Trabalho

O aumento significativo no volume de commits e PRs reflete, em primeira instância, o crescimento das atividades de desenvolvimento. Contudo, apesar desse crescimento quantitativo das PRs em 2023, a proporção de PRs em que permaneceram em aberto superou consideravelmente o período de 2022, como mostrado na Figura 2 e na tabela 4.

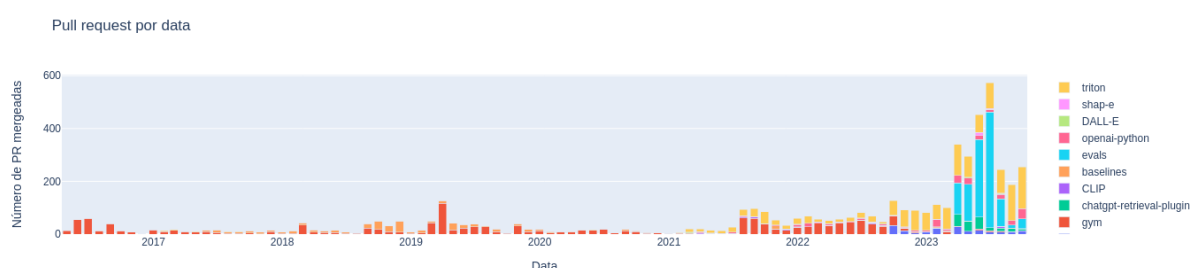


Figura 3. Fluxo de PRs mergeadas por repositório

Fonte: Resultados originais da pesquisa

Tabela 4. Análise das PRs para os últimos anos.

	2021	2022	2023
Commits no período	551	614	1634
PRs abertas no período	513	891	2509
PRs fechadas no período	501	877	2349
Razão PRs fechadas e abertas	0.99	0.99	1.06
PRs fechadas após período	9.2%	4.8%	7.5%
PRs abertas até o momento	47	43	189

Fonte: Resultados originais da pesquisa

Esse fenômeno suscita questionamentos sobre a gestão da crescente carga de trabalho ou mesmo sobre a complexidade das tarefas necessárias para a sua conclusão.

Em 2021, 2022 e 2023, a razão entre PRs fechadas e abertas esteve acima de 1, sinalizando um alto desempenho operacional na qual as PRs eram tratadas quase no mesmo ritmo em que eram criadas. E, apesar da crescente carga de trabalho, a porcentagem de PRs fechadas tardiamente caiu drasticamente de 2021 para 2022, indicando que a equipe está buscando maneiras de se tornar ainda mais eficiente.

(d) Sazonalidade na Produtividade:

Com a exporação dos dados, é possível notar um fenômeno interessante relacionado à sazonalidade que parece estar intimamente ligado ao aumento da carga de trabalho.

O padrão se manifesta na relação entre as Pull Requests (PRs) fechadas e as PRs abertas, evidenciado na Tabela 5, Tabela 6 e Tabela 7, e geralmente coincide com momentos de queda na produtividade.

Tabela 5. Análise das PRS por mês de 2021

	2021											
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Commits no período	28	38	35	13	23	13	89	94	94	42	31	51
PRs abertas no período	14	20	21	24	27	21	55	95	80	68	24	64
PRs fechadas no período	7	18	18	12	23	16	97	95	84	44	31	56
Razão PRs fechadas e abertas	0.50	0.90	0.86	0.50	0.85	0.76	1.76	1.00	1.05	0.65	1.29	0.88
PRs fechadas após período	64%	20%	19%	54%	19%	29%	13%	13%	9%	41%	29%	27%
PRs abertas após 2021	1	1	1	3	0	3	0	1	1	17	2	17

Fonte: Resultados originais da pesquisa

Tabela 6. Análise das PRS por mês de 2022

	2022											
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Commits no período	57	60	53	58	56	77	53	37	89	40	11	23
PRs abertas no período	56	63	52	60	59	98	64	42	126	93	96	82
PRs fechadas no período	67	59	55	59	61	85	64	47	120	88	95	77
Razão PRs fechadas e abertas	1.20	0.94	1.06	0.98	1.03	0.87	1.00	1.12	0.95	0.94	0.99	0.94
PRs fechadas após período	16%	25%	19%	17%	14%	21%	20%	14%	13%	15%	13%	21%
PRs abertas após 2022	0	0	1	1	1	7	1	1	6	5	4	17

Fonte: Resultados originais da pesquisa

Tabela 7. Análise das PRS por mês de 2023

	2023								
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set
Commits no periodo	95	83	326	188	265	259	138	139	141
PRs abertas no periodo	106	103	706	451	313	315	171	181	163
PRs fechadas no periodo	106	99	333	275	431	550	183	159	213
Razão PRs fechadas e abertas	1.00	0.96	0.47	0.61	1.38	1.75	1.07	0.88	1.31
PRs fechadas após periodo	21%	17%	55%	57%	40%	27%	27%	22%	12%
PRs abertas após 2022	6	1	35	35	33	25	21	13	20

Fonte: Resultados originais da pesquisa

A observação interessante é que a proporção de PRs fechadas no ano subsequente (ou seja, aquelas fechadas após o ano em que foram abertas) diminuiu de 9,2% em 2021 para 4,8% em 2022. Esse dado pode ser interpretado como um indicativo positivo de que a equipe está refinando a eficiência e a velocidade do seu processo de revisão.

Em 2023, notamos picos de sazonalidade a partir do mês de março, estendendo-se até setembro, afetando a eficiência, apesar do crescimento da produtividade nos repositórios a partir de maio. No entanto, é crucial destacar que os dados não sugerem uma sazonalidade com um padrão definido relacionado a eventos como férias ou o final do ano. A complexidade desse fenômeno requer uma análise mais profunda para identificar as causas subjacentes.

(e) Eventos Excepcionais

Ao explorar em detalhes o tamanho das Pull Requests (PRs), revelamos informações cruciais para aprimorar o processo de desenvolvimento de software. Essa análise não só nos possibilita quantificar o alcance das alterações, mas também destaca potenciais problemas relacionados ao desempenho e à qualidade do software.

É fundamental reconhecer que o tamanho das PRs, embora seja um indicador útil, não oferece uma visão completa da complexidade do trabalho realizado. É bem sabido que a complexidade pode residir em alterações que, apesar de modestas em termos de linhas de código, têm um impacto significativo. Adicionalmente, diferentes equipes e projetos têm a liberdade de adotar padrões diversos para o tamanho das PRs, o que torna comparar essas métricas um desafio intrincado. Portanto, na Figura 4 abaixo, apresenta-se o volume de mudanças ao longo dos anos.

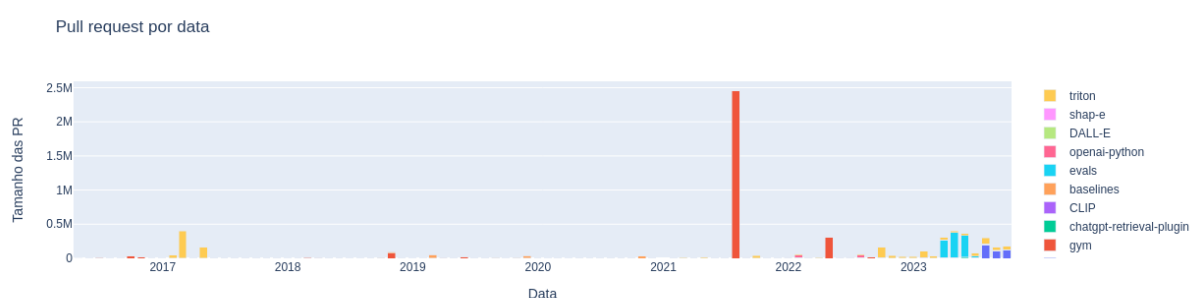


Figura 4. Fluxo de mudanças de código por repositório.

Fonte: Resultados originais da pesquisa

Ao longo do período de análise mais profunda foi realizada nos anos de 2021, 2022 e 2023, mostrado na Figura 5, e identificamos eventos excepcionais que merecem especial destaque. Por exemplo, em julho de 2021, registramos um notável aumento nas mudanças

de código no repositório "gym," impulsionado por 34 PRs. Situação semelhante ocorreu em abril de 2022, novamente no repositório "gym," desta vez com 3 PRs. Em agosto de 2022, testemunhamos um pico significativo de atividade no repositório "triton," que manteve um nível substancial de engajamento até setembro de 2023. Além disso, durante 2023, identificamos picos de atividade notáveis, ocorrendo de fevereiro a abril no repositório "evals" e de junho a setembro no repositório "whisper".



Figura 5. Fluxo de mudanças de código nos anos de 2021 a 2023

Fonte: Resultados originais da pesquisa

Para uma análise mais aprofundada dos repositórios, a Tabela 8 exibe as 20 PRs mergeadas com as maiores mudanças de código:

Tabela 8. Top 20 PRs com maiores mudanças de código (continua)

Repositório	Numero da PR	Data do merge	Mudanças de código
triton	2016	2023-07-31	18224
triton	1146	2023-02-04	19455
triton	2396	2023-09-26	21648
triton	2036	2023-08-07	30729
triton	268	2021-09-09	31596
triton	711	2022-09-26	32000

Tabela 8. Top 20 PRs com maiores mudanças de código (conclusão)

Repositório	Numero da PR	Data do merge	Mudanças de código
openai-python	67	2022-01-25	33614
triton	1921	2023-07-11	40756
triton	11	2017-01-21	44852
evals	206	2023-03-21	52084
evals	196	2023-03-15	52114
evals	203	2023-03-16	52130
triton	1004	2023-01-08	76497
evals	191	2023-03-27	90813
whisper	1661	2023-09-15	115992
triton	18	2017-02-08	119012
triton	20	2017-04-05	164691
triton	19	2017-02-16	276893
evals	590	2023-05-25	287102
evals	717	2023-04-19	329025

Fonte: Resultados originais da pesquisa

É imperativo destacar eventos como esses, pois funcionam como indicadores de alerta para questões mais profundas que requerem investigação meticulosa. Por exemplo, em 19 de abril de 2023, um dia se destacou devido à considerável quantidade de mudanças no repositório "evals," onde foram modificados 212 arquivos, adicionando 329,025 linhas de código, juntamente com uma extensa quantidade de samples e código.

Além disso, é relevante notar que 13 das 20 maiores alterações ocorreram em 2023. Uma análise mais aprofundada revela que, dessas alterações, 60% delas estão concentradas no repositório "triton," enquanto 30% estão no repositório "evals." Esses números merecem atenção especial, pois um aumento significativo na atividade de código pode sobrecarregar o processo de revisão e colaboração. Essa situação levanta importantes questionamentos sobre a gestão e revisão das mudanças.

(f) Priorização das Issues

As issues desempenham um papel crucial em qualquer projeto de desenvolvimento de software. Elas servem como um registro completo de tudo o que precisa ser feito, está sendo feito ou já foi concluído. Isso abrange desde relatórios de bugs até solicitações de novos recursos e tarefas gerais. Portanto, o gerenciamento inteligente de issues é crítico para a organização, o acompanhamento e o sucesso do projeto.

Tabela 9: Descritivo das issues por repositório em 2021

Repositório	Issues 2021			
	Criadas	Fechadas	Abertas e fechadas	Abertas
gym	402	526	360	3
CLIP	202	156	156	42
baselines	34	9	7	27
openai-python	61	44	44	1
DALL-E	48	26	26	22
triton	346	299	296	12

Fonte: Resultados originais da pesquisa

Tabela 10: Descritivo das issues por repositório em 2022

Repositório	Issues 2022			
	Criadas	Fechadas	Abertas e fechadas	Abertas
whisper	70	59	59	3
gym	620	695	590	26
CLIP	105	48	44	61
baselines	17	3	3	14
openai-python	102	97	81	9
DALL-E	39	4	4	35
triton	589	482	478	47

Fonte: Resultados originais da pesquisa

Tabela 11: Descritivo das issues por repositório em 2023

Repositório	Issues 2023			
	Criadas	Fechadas	Abertas e fechadas	Abertas
whisper	104	85	77	27
gym	70	32	28	42
chatgpt-retrieval-plugin	351	156	156	195
CLIP	81	20	20	61
baselines	10	2	2	8
evals	1314	1214	1214	100
openai-python	455	323	311	144
DALL-E	17	2	2	15
shap-e	111	36	36	75
triton	1329	1159	1061	268

Fonte: Resultados originais da pesquisa

Dada a natureza sempre mutável e complexa dos projetos de software, a equipe deve dominar a arte de lidar eficazmente com as issues à medida que elas surgem. Nas Tabelas 9, Tabela 10 e Tabela 11, mergulhamos nos dados da OpenAI para obter insights sobre como nossa equipe tem gerenciado essas questões e identificar áreas onde precisamos aprimorar nossa abordagem.

No centro dessa análise, estão os repositórios individuais. Surpreendentemente, ainda há issues abertas dos anos de 2021 e 2022, com o repositório "CLIP" chamando atenção por manter o maior número de issues antigas em aberto. Além disso, observamos o repositório "triton" acumulando muitas issues abertas nos últimos três anos, com um pico notável em 2023 detalhado na Figura 6. Ainda na na Figura 6 observamos outro repositório com alto fluxo de issues, mesmo sendo mais recente, o "evals" ocupa o segundo lugar em termos de criação de issues. Isso destaca como até mesmo projetos jovens podem causar um impacto significativo na comunidade OpenAI.

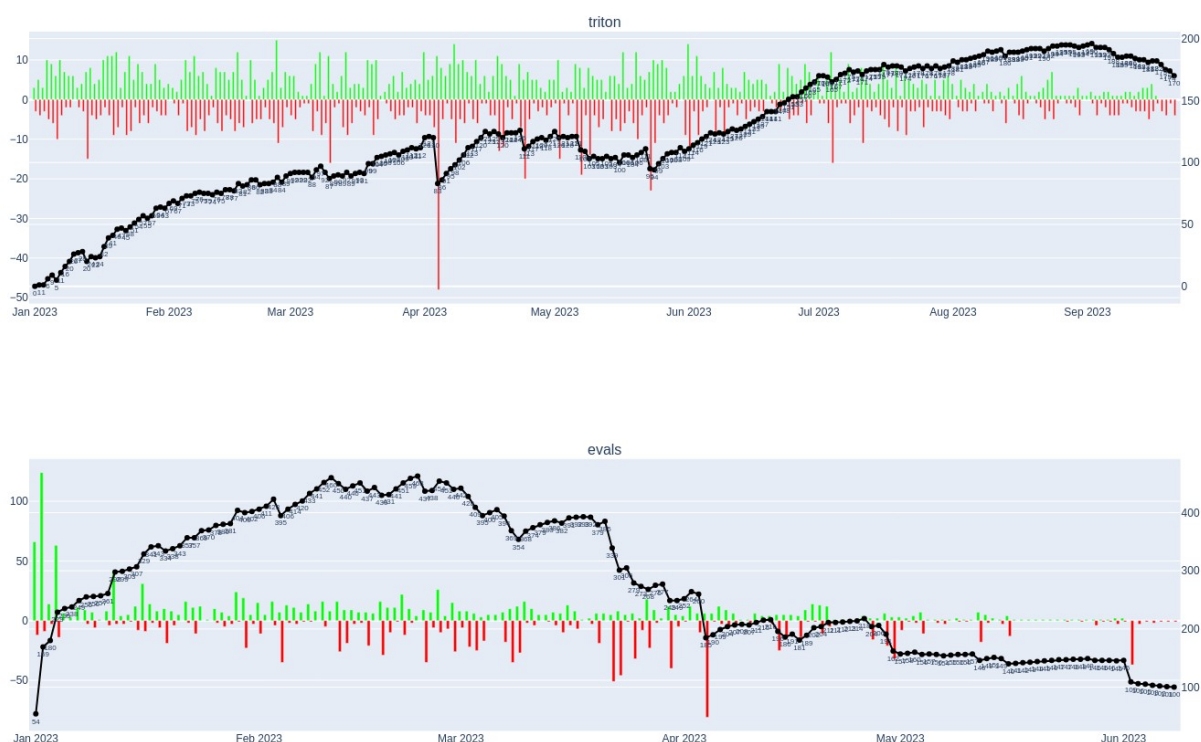


Figura 6. Evolução do backlog de issues dos repositórios mais movimentados de 2023

Fonte: Resultados originais da pesquisa

Ao examinarmos os números gerais para as issues abertas nos anos de 2021, 2022 e 2023, identificamos uma tendência interessante, conforme mostrado na Tabela 12. A média do tempo necessário para fechar uma issue está diminuindo, passando de 48 dias em

2021 para 17 dias em 2023. Isso sinaliza que nossa equipe está se tornando mais eficiente na resolução de issues, resultando em entregas mais rápidas.

Tabela 12: Descritivo das issues dos ultimos anos

	2021	2022	2023	8/2023
Issues abertos no periodo	1093	1542	3842	288
Total de issues fechados	1060	1388	3029	195
Taxa de resolução total	97%	90%	79%	68%
Média de tempo de resolução total	48 days 22:49:53	18 days 07:19:22	17 days 21:00:4	6 days 11:10:35
Desvio da média de tempo de resolução total	110 days 16:57:56	51 days 16:53:17	28 days 17:15:22	11 days 07:38:52
Issues fechados (abertas no período)	889	1259	2907	166
Taxa de resolução do período	81%	82%	76%	58%
Issues que continuam abertas	107	195	935	84
Issues fechadas tardiamente	9%	6%	0%	13%

Fonte: Resultados originais da pesquisa

Além disso, o desvio padrão da média também está diminuindo, indicando uma maior consistência na resolução de issues. No entanto, é crucial observar que ainda temos um grande número de issues abertas aguardando resolução, o que sugere a necessidade contínua de melhorias nessa área.

Na Figura 7 nota-se o aumento no número de issues abertas por ano, especialmente em 2023, refletindo um maior envolvimento da comunidade desde o lançamento do chatGPT. No entanto, essa tendência de crescimento também destaca a importância de um gerenciamento eficaz das issues, dada a crescente complexidade do projeto.

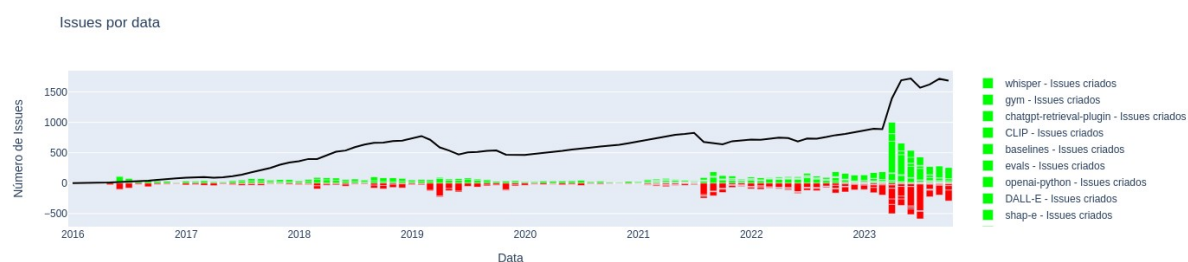


Figura 7. Evolução do backlog de issues

Fonte: Resultados originais da pesquisa

Também é evidente o aumento na quantidade de issues fechadas, indicando um esforço da comunidade para solucionar problemas, otimizar o código e introduzir novas

funcionalidades. No entanto, o ritmo de fechamento de issues ainda não acompanha o ritmo de abertura, resultando em um desequilíbrio crescente. Em 2021, a taxa de resolução era de 97%, mas caiu para 79% em 2023. Isso pode sugerir que os projetos estão se tornando mais complexos, a equipe pode estar tendo dificuldades para acompanhar o crescimento das issues ou pode estar priorizando a resolução de outras questões. É relevante notar que a porcentagem de issues fechadas tardiamente diminuiu, indicando que os contribuidores estão priorizando issues mais recentes.

Da mesma forma, a porcentagem de issues que permaneceram abertas é um indicador importante, observe que a diferença entre o número de issues abertas e fechadas pode indicar a presença de um backlog de tarefas acumulado de 1685 issues, sugerindo gargalos no processo e a necessidade de melhorias.

Por fim, a queda na taxa de resolução de issues em 2023 é notável. Enquanto em 2021 e 2022 a taxa estava acima de 90%, em 2023 caiu para cerca de 79%. Isso pode sugerir desafios na manutenção da eficiência, apesar do aumento da carga de trabalho. É importante explorar estratégias para melhorar essa taxa e garantir que a qualidade do produto não seja comprometida pelo aumento da carga de trabalho.

Também é relevante considerar a possibilidade de um alto volume de fechamentos antes do final do ano, embora essa tendência não tenha sido observada nos dados apresentados na Tabela 13:

Tabela 13. Descritivo das issues dos 5 ultimos meses de 2022

	8/2022	9/2022	10/2022	11/2022	12/2022
Issues abertos no periodo	100	193	163	139	139
Total de issues fechados	73	162	144	110	105
Taxa de resolução total	73%	84%	88%	79%	76%
Média de tempo de resolução total	32 days 23:05:18	10 days 14:44:12	14 days 17:02:36	14 days 18:09:04	9 days 14:11:28
Desvio da média de tempo de resolução total	62 days 03:39:54	32 days 23:42:51	42 days 23:07:06	46 days 02:00:19	25 days 02:15:36
Issues fechados (abertas no período)	53	141	116	96	88
Taxa de resolução do período					
Issues que continuam abertas	21	22	25	20	24
Issues fechadas tardiamente	26%	16%	14%	17%	19%

Fonte: Resultados originais da pesquisa

Essas análises fornecem informações valiosas sobre o gerenciamento de issues, destacando áreas para melhoria à medida que o projeto continua a evoluir.

Work Breakdown

A prática de Work Breakdown para mudanças de código é uma ferramenta poderosa que nos permite compreender a natureza das alterações, identificar tendências de trabalho, avaliar a produtividade da equipe e fornecer estimativas precisas para tarefas específicas. Esta análise divide as mudanças de código em três categorias: New Code (novas funcionalidades), Refactoring (melhorias no código existente sem alterar a funcionalidade) e Rework (correção de erros). Essa abordagem fornece uma visão clara da alocação de esforços, permitindo uma gestão mais eficaz do tempo e dos recursos.

Enfrentamos um desafio ao aplicar essa técnica aos repositórios da OpenAI: a falta de um padrão convencional nas descrições dos commits. Para solucionar essa questão, estabelecemos a seguinte regra: commits que contêm apenas adições de código são classificados como New Code, commits com a palavra "refactoring" ou seus derivados na descrição, que também incluem alterações de conteúdo (adições e remoções), são categorizados como Refactoring, enquanto todas as outras mudanças são consideradas Rework. A adoção de um padrão como o "Conventional Commits" poderia aprimorar ainda mais essa segregação, estabelecendo uma linguagem comum e facilitando a análise de dados, melhorando a comunicação e permitindo previsões mais precisas para futuras mudanças. Isso, por sua vez, aprimoraria a confiabilidade e a eficiência do processo de Work Breakdown.

Nossa análise detalhada, que desagregou o trabalho em categorias de New Code, Refactoring e Rework, trouxe à tona os seguintes resultados e insights:

- **Distribuição do Trabalho:** Como mostrado na Figura 8, a maior parte do esforço está relacionada à adição de novo código, representando aproximadamente 49.5% do trabalho total. Em segundo lugar, encontramos o retrabalho, com 47.4%, enquanto a refatoração compõe uma parcela menor, representando 3.1% do esforço.

Alterações por tipo



Figura 8. Work breakdown dos repositórios selecionados

Fonte: Resultados originais da pesquisa

- **Tendências ao Longo dos Anos:** As tendências variáveis ao longo dos anos, expressas na Figura 8, indicam a importância de equilibrar os esforços entre a correção de erros, a adição de novas funcionalidades e a melhoria do código existente. Em 2021, o retrabalho representou uma parcela significativa, atingindo 58.4% do trabalho, com apenas 41.3% para novos códigos e uma mínima porção de 0.3% para refatoração. No ano seguinte, 2022, houve um aumento considerável no retrabalho, reduzindo a adição de novos códigos para cerca de 24%. A refatoração manteve-se relativamente constante, com um leve aumento para 4.1%. Em contrapartida, em 2023, observamos uma inversão no cenário, onde a adição de novos códigos dominou com 67.2% do trabalho, enquanto o retrabalho diminuiu para 31.3%. A refatoração, mais uma vez, permaneceu em um nível muito baixo, representando apenas 1.5%.

Alterações por tipo nos anos

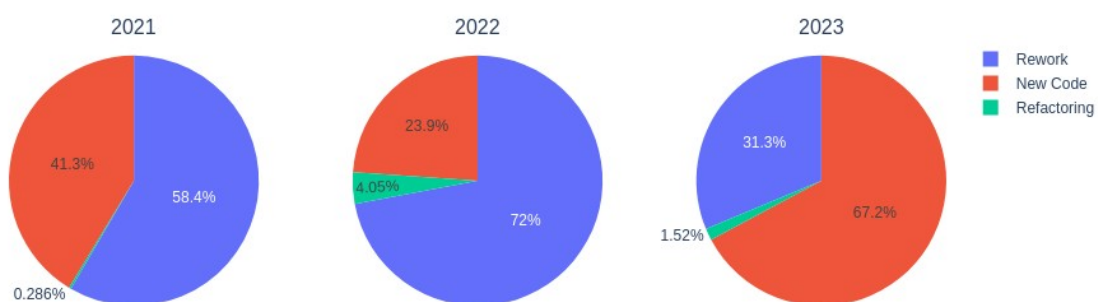


Figura 8. Work breakdown dos repositórios selecionados nos últimos anos

Fonte: Resultados originais da pesquisa

Essas tendências levantam questões sobre o equilíbrio dos esforços de desenvolvimento. Embora o retrabalho seja uma parte essencial do desenvolvimento de software, sua predominância em 2022 sugere que recursos significativos foram alocados para correções ou repetição de trabalho. Por outro lado, a proporção relativamente baixa de refatoração indica que talvez não estejamos dedicando tempo suficiente para melhorar a qualidade do código existente. Isso é fundamental para garantir a manutenibilidade do código e reduzir a necessidade de retrabalho no futuro. Essas observações destacam a importância de uma abordagem equilibrada para o desenvolvimento de software, que não apenas adicione novas funcionalidades e corrija erros, mas também dedique tempo para aprimorar a qualidade do código existente. Ao fazê-lo, podemos aprimorar a eficiência de nosso processo de desenvolvimento e garantir a qualidade contínua de nossos produtos.

Análise Mensal por Repositório

No contexto do desenvolvimento de software, a análise mensal desempenha um papel crucial para acompanhar o progresso, identificar tendências e tomar decisões informadas. Nesta análise o foco foi o mês de agosto de 2023.

Esta análise aborda vários aspectos, desde atividades em repositórios e tempo de resolução de issues até a distribuição de mudanças no código e eventos excepcionais. Ao compreender essas métricas e tendências, podemos obter insights valiosos para melhorar a eficiência, qualidade e direção dos esforços de desenvolvimento de software.

- (a) Total de Commits:** Durante o mês de agosto de 2023, a equipe realizou um total de 139 commits nos os repositórios selecionados da OpenAI. Isso representa um uma constante em relação a julho de 2023, quando houve 138 commits.
- (b) Total de Pull Requests:** Foram abertas 181 pull requests e fechadas 159 no mês de agosto. Os repositórios mais ativos em termos de pull requests foram "triton" seguido por "openai-python" e "evals".
- (c) Tempo de Resolução de Issues:** A média de tempo necessário para fechar uma issue aberta em agosto de 2023 foi de 6 days 11:10:35 com destaque para o whisper com um tempo de 23 days 06:00:00, devemos ter me mente que ainda temos 29% das issues abertas e que isso elevará o tempo de resolução. Comparando com o ano anterior temos um tempo de resolução de 32 days 23:05:18 com 21% de issues abertas até o momenton da análise. Vemos que esse tempo tem uma tendencia de melhorar em comparação ao ano anterior, mas ainda está longe de ter um tempo que é de até 5 dias para issues de alta e média prioridade.
- (d) Taxa de Resolução Total:** A taxa de resolução total foi de 68%, indicando uma baixa eficiência na resolução de issues e quando comparado ao ano anterior houve uma queda de 3% na eficiência.
- (e) Work breakdown:** Em agosto de 2023, observamos a distribuição mostrada na Figura 8 para asas mudanças de código. Comparando com agosto de 2022, notamos que o retrabalho, foi majoritariamente o que foi feito em agosto de ambos os anos. Isso pode indicar uma necessidade de melhorar a qualidade das entregas.

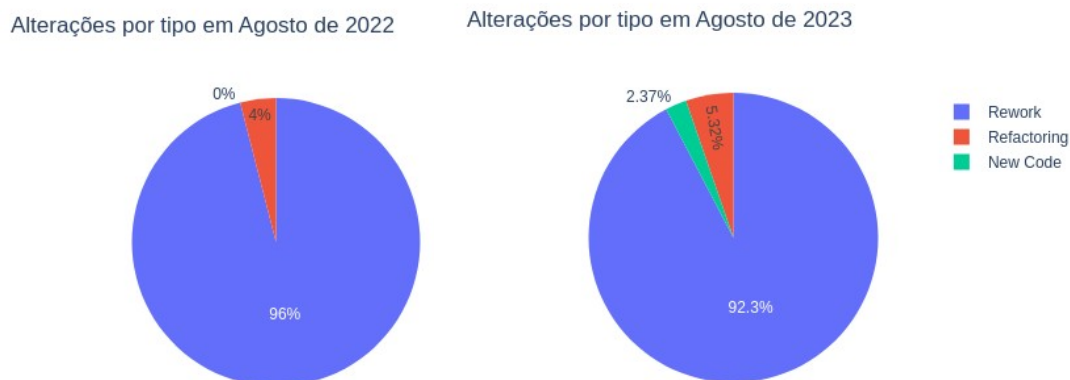


Figura 9. Work breakdown dos repositórios selecionados em agosto de 2023

Fonte: Resultados originais da pesquisa

A Figura 10 evidencia como a distribuição do workbreakdown se deu da seguinte forma nos repositórios:

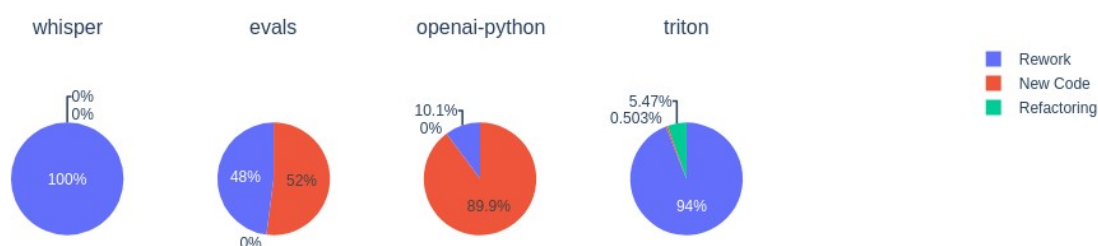


Figura 10. Work breakdown detalhado de agosto de 2023

Fonte: Resultados originais da pesquisa

- (f) **Eventos Excepcionais:** A Figura 11 coloca em evidência o dia 7 de agosto de 2023, com um pico significativo de mudanças no repositório "whisper," com 1 pull requests e mais de 100.000 linhas de código alteradas em 20 arquivos. Isso pode ser um indicativo de uma grande atualização ou correção sendo realizada.

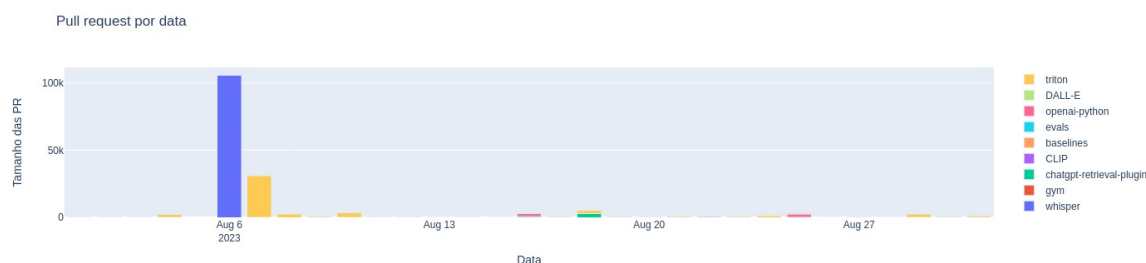


Figura 11. Work breakdown dos repositórios selecionados

Fonte: Resultados originais da pesquisa

- (g) **Priorização das Issues:** Durante agosto de 2023, foram criadas 288 novas issues nos repositórios selecionados, com destaque para "triton" com 159 issues abertas. Um total de 195 issues foram fechadas no mesmo período. Isso resultou em um aumento no backlog de issues abertas.

Análise dos contribuidores

Uma análise detalhada dos repositórios da OpenAI revela um cenário interessante quando se trata de contribuições. Houveram 3 estágios até chegarmos no ambiente atual:

- **Concentração Inicial (2016-2018):** Nos primeiros anos de existência dos repositórios da OpenAI, a contribuição estava altamente concentrada entre um pequeno número de principais contribuidores. Isso sugere que um grupo seleto de desenvolvedores desempenhou um papel fundamental no estabelecimento e desenvolvimento dos projetos.
- **Diversificação (2019-2020):** Entre 2019 e 2020, observamos uma tendência de diversificação na distribuição da contribuição. O número de principais contribuidores aumentou, indicando uma maior participação da comunidade de código aberto nos projetos da OpenAI.
- **Estabilidade (2021-2023):** Nos últimos anos, de 2021 a 2023, a distribuição da contribuição parece ter se estabilizado, com um grupo relativamente consistente de principais contribuidores mantendo uma parte significativa dos commits. No entanto, a diversificação observada nos anos anteriores também continua a desempenhar um papel importante.

Quando análise de dá para os 50 maiores contribuidores, observamos o papel fundamental que desempenharam na evolução desses projetos. A Tabela 14 apresenta como esses contribuidores se distribuem entre os repositórios:

Tabela 14. Top 50 Contribuidores nos repositórios selecionados (continua)

Repositório	Top 50 Contribuidor	Total Contribuidor
whisper	1	59
gym	21	399
chatgpt-retrieval-plugin	3	35
CLIP	1	20
baselines	6	117

Tabela 14. Top 50 Contribuidores nos repositórios selecionados (conclusão)

Repositório	Top 50 Contribuidor	Total Contribuidor
evals	3	418
openai-python	5	80
DALL-E	0	1
shap-e	1	7
triton	13	178

Fonte: Resultados originais da pesquisa

É notável que esse grupo seletivo de 50 usuários, que representa menos de 4% do total de contribuidores nos projetos selecionados da OpenAI, é responsável por quase 94% das mudanças no código. Eles demonstram um comprometimento excepcional com esses projetos e desempenham um papel vital em sua evolução.

Dentre esses contribuidores, destacam-se aqueles que têm participação significativa nos repositórios mais proeminentes. O "gym", por exemplo, é liderado por 21 dos 50 maiores contribuidores, representando a maior parte das mudanças. O "triton" também possui uma forte presença desses contribuidores destacados, com 13 deles liderando o desenvolvimento.

Essa concentração de contribuições não está restrita apenas aos commits, mas também influencia as pull requests. Embora esses 50 contribuidores desempenhem um papel significativo nas mudanças de código, eles representam apenas 45% dos commits realizados e menos de 35% das pull requests nos repositórios selecionados.

Uma observação interessante é que quatro dos 50 maiores contribuidores estão ativos em dois dos repositórios analisados, o que totaliza 54 contribuidores no grupo de análise. Esse resultado destaca a importância de entender e monitorar os padrões de contribuição desses usuários-chave, bem como explorar formas de incentivar e engajar outros colaboradores para garantir a continuidade e aprimoramento dos projetos.

Considerações Finais

Após uma análise profunda e abrangente das práticas de desenvolvimento de software na OpenAI, emergem conclusões cruciais que lançam luz sobre a natureza dinâmica e desafiadora desse empreendimento. As principais descobertas revelam áreas de otimização e considerações críticas que podem moldar o futuro da organização. Aqui, delineamos essas conclusões, enfatizando sua abordagem centrada no cliente e orientada para resultados.

1. **Foco na Eficácia da Resolução de Issues:** Ao examinar o tratamento de issues nos repositórios da OpenAI, é evidente que a eficácia na resolução desses problemas é um componente-chave para aprimorar. Embora tenha havido melhorias notáveis no tempo médio de resolução de issues, a taxa global de resolução diminuiu em 2023. Essa tendência levanta a bandeira vermelha de que, à medida que a organização cresce, pode enfrentar desafios em manter a eficiência na resolução de problemas. Para a OpenAI, este é um indicador crítico a ser monitorado e melhorado.
2. **Distribuição Inteligente do Trabalho:** A análise do trabalho desagregado em categorias de "New Code" (novo código), "Refactoring" (melhorias no código existente) e "Rework" (correção de erros) revelou insights valiosos. O foco significativo no retrabalho, especialmente em agosto de 2023, destaca uma necessidade urgente de elevar a qualidade das entregas iniciais. Reduzir a necessidade de correções posteriores não apenas melhora a eficiência, mas também impulsiona a satisfação do cliente.
3. **Contribuições-Chave e Diversificação:** Um achado notável é a concentração de contribuições em um pequeno grupo de principais colaboradores. Embora esses indivíduos desempenhem um papel fundamental, é essencial buscar uma diversificação mais ampla das contribuições. Engajar uma comunidade mais ampla de desenvolvedores é vital para garantir a continuidade e a evolução sustentável dos projetos.
4. **Qualidade Inicial como Pedra Angular:** A lição crítica aqui é a importância de entregar código de alta qualidade desde o início. Essa prática não só minimiza retrabalho, mas também aprimora a eficácia global, permitindo uma alocação mais equilibrada de esforços entre novas funcionalidades, correções e refatorações.
5. **Mudanças e Tendências ao Longo do Tempo:** As tendências observadas ao longo dos anos, com mudanças na distribuição do trabalho e na eficácia, destacam a natureza fluida do desenvolvimento de software. A capacidade de se adaptar a essas mudanças e aprender com elas é fundamental para a resiliência e o crescimento contínuo da OpenAI.
6. **Padronização e Comunicação:** A adoção de padrões de commits, como o "Conventional Commits", surge como uma recomendação valiosa. Essa prática não apenas facilita a análise de mudanças no código, mas também melhora a comunicação e a compreensão das alterações, impulsionando uma colaboração mais eficaz.

Em resumo, estas conclusões não apenas impactarão a OpenAI em seus esforços futuros, mas também servirão como um farol para outras organizações de desenvolvimento de software. À medida que navegamos pelas águas em constante evolução da indústria de tecnologia, a habilidade de se adaptar, aprimorar a eficiência e manter padrões excepcionais de qualidade permanece vital para o sucesso e a inovação. ência e manter altos padrões de qualidade permanece crítica para o sucesso e a inovação.

Referências

Ciric D., Lalic B., Gracanin D., Palcic I., Zivlak N. "Agile Project Management in New Product Development and Innovation Processes: Challenges and Benefits Beyond Software Domain," 2018 IEEE International Symposium on Innovation and Entrepreneurship (TEMS-ISIE), Beijing, China, 2018, pp. 1-9, doi: 10.1109/TEMS-ISIE.2018.8478461.

Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). Social coding in GitHub: transparency and collaboration in an open software repository. *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, 1277–1286.

Georgios Gousios, Martin Pinzger, and Arie van Deursen. 2014. An exploratory study of the pull-based software development model. In *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*. Association for Computing Machinery, New York, NY, USA, 345–355. <https://doi.org/10.1145/2568225.2568260>

Gil, A.C. 2002. Como elaborar projetos de pesquisa. 4ed. Editora Atlas S.A, São Paulo, SP, Brasil.

Kelleher, J. D.; Tierney, B. (2018). *Data science fundamentals*. Boca Raton: CRC Press.

Menzies, T., Williams, L., & Zimmermann, T. (Eds.). (2016). *Perspectives on Data Science for Software Engineering* (Vol. 1). Morgan Kaufmann Publishers.

Nerur S., Mahapatra R., Mangalaraj G. "Challenges of migrating to agile methodologies". *Commun. ACM* 48, 2005, 72–78. <https://doi.org/10.1145/1060710.1060712>

Provost, F., & Fawcett, T. (2013). *Data science for business: What you need to know about data mining and data-analytic thinking*. O'Reilly Media, Inc.

Sutherland, J. (2014). Scrum: The Art of Doing Twice the Work in Half the Time. Crown Business.

Apêndice

O código-fonte completo do projeto de análise realizado neste trabalho encontra-se disponível publicamente no repositório do GitHub. Ele pode ser acessado através do seguinte link:

https://github.com/Daniinovaes/Github_tcc_analysis

Neste repositório, você terá acesso ao código utilizado para coletar, analisar e visualizar os dados, bem como todas as informações necessárias para replicar o estudo e realizar análises adicionais, se desejado. Sinta-se à vontade para explorar e utilizar o código conforme necessário para seus próprios estudos e investigações.