

Análise Comparativa de Repositórios com Releases Rápidos e Lentos: Implicações para a Qualidade do Código

Alfredo Luis, Bruno Evangelista, Estevão Rodrigues, Vinicius Salles

25 de Agosto de 2025

Release Rápido (RRC) e o desenvolvimento de Software

- Releases rápidos referem-se a ciclos de desenvolvimento de software com intervalos curtos (ex.: 5-35 dias entre releases)
- Desenvolvimentos baseando-se em RRC podem prejudicar a segurança de um sistema?

Motivação principal: Estudar implicações do uso do

- desenvolvimento ágil e engenharia de releases em projetos open-source.
- Conjunto de dados: 994 projetos do GitHub com mais de 2 milhões de issues, tempo médio de release de 22 dias.

Visão Geral do Conjunto de Dados Existente

- Base de dados encontrada em estudos anteriores.
- Coletado do GitHub usando API, focando em projetos populares (>50 estrelas, >50 forks) em 18 linguagens.
- Filtrado para releases rápidos: Tempo médio de release entre 5-35 dias, pelo menos 19 colaboradores e 19 releases distintos.
- Estatísticas: Média de 112 colaboradores, 48 releases distintos, 2365 issues por projeto.

Extensão para Releases Lentos

- Ampliar a pesquisa para repositórios de projetos que utilizam “Releases Lentos”
- Ampliação de escopo: Minerar um conjunto de dados comparável “SlowRelease” para projetos com ciclos mais longos (>60 dias entre releases).
- Entender as diferenças entre repositórios que utilizam estes dois métodos de desenvolvimento, principalmente no quesito de qualidade.

Goal: Avaliar os efeitos das RRCs na qualidade de sistemas open-source, comparando-os com sistemas lançados em releases lentas.

- **Q1:** O RRC torna as releases mais vulneráveis?

Métricas: vulnerabilities (Número de Vulnerabilidades), security_rating (Classificação de Segurança), security_hotspots (Pontos de Segurança Críticos)

- **Q2:** Erros são mais comuns em releases de RRC?

Métricas: bugs (Número de Bugs), reliability_rating (Classificação de Confiabilidade), code_smells (Número de Code Smells)

- **Q3:** O retrabalho é maior em sistemas que utilizam RRC, do que em sistemas que utilizam releases lentas?

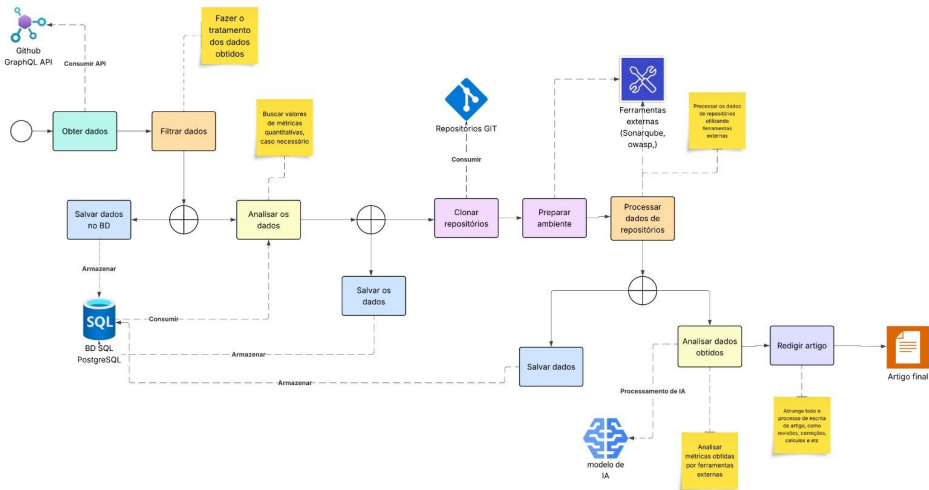
Métricas: technical_debt (Dívida Técnica), sqale_index (Índice de Dívida Técnica), duplicated_lines_density (Densidade de Linhas Duplicadas),

O que Podemos Encontrar

- Cenários onde as RRCs podem causar vulnerabilidades e baixa maturidade nas releases de sistemas.
- Como o open-source está reagindo a este modelo de desenvolvimento, e suas consequências.
- Como o retrabalho está relacionado às RRCs, e se aumenta ou não em sistemas que utilizam/não utilizam esta metodologia

- Desenvolvedores da comunidade open-source.
- Gestores de projetos open-source
- Gerentes de projeto que utilizam metodologias ágeis
- Desenvolvedores focados em segurança

Metodologia



Árvores de Artigos



Referências

- S. D. Joshi and S. Chimalakonda, "RapidRelease - A Dataset of Projects and Issues on Github with Rapid Releases," 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), Montreal, QC, Canada, 2019, pp. 587-591, doi: [10.1109/MSR.2019.00088](https://doi.org/10.1109/MSR.2019.00088).
- T. Jansen, Z. A. Khalil, E. Constantinou and T. Mens, "Does the duration of rapid release cycles affect the bug handling activity?," 2021 IEEE/ACM 4th International Workshop on Software Health in Projects, Ecosystems and Communities (SoHeal), Madrid, Spain, 2021, pp. 17-24, doi: [10.1109/SoHeal52568.2021.00009](https://doi.org/10.1109/SoHeal52568.2021.00009).
- D. A. da Costa, S. McIntosh, U. Kulesza and A. E. Hassan, "The Impact of Switching to a Rapid Release Cycle on the Integration Delay of Addressed Issues - [An Empirical Study of the Mozilla Firefox Project](#)," 2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR), Austin, TX, USA, 2016, pp. 374-385.
- M. V. Mäntylä, F. Khomh, B. Adams, E. Engström and K. Petersen, "On Rapid Releases and Software Testing," 2013 IEEE International Conference on Software Maintenance, Eindhoven, Netherlands, 2013, pp. 20-29, doi: [10.1109/ICSM.2013.13](https://doi.org/10.1109/ICSM.2013.13).