

# Dashboard Lab04 - Visualização de dados

Pontifícia Universidade Católica de Minas Gerais - Engenharia de software (Noite)

Alunos: Henrique Lobo, Estevão Rodrigues

Professor: Danilo Quadros

Diciplina: Laboratório de experimentação de software

O objeto de estudo deste trabalho são os repositórios de código aberto do GitHub.

A metodologia de coleta visa criar dois grupos distintos para comparação, com base na frequência de suas releases:

- **Rapid Release Cycle (RRC):** Re却itórios com ciclos de release curtos, definidos como um intervalo entre 5 e 35 dias.
- **Slow Release:** Re却itórios com ciclos de release longos, definidos como um intervalo superior a 60 dias.

## Questões de pesquisa

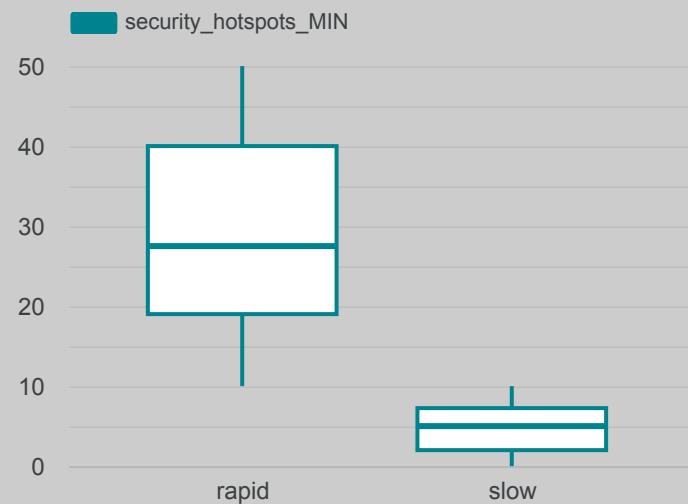
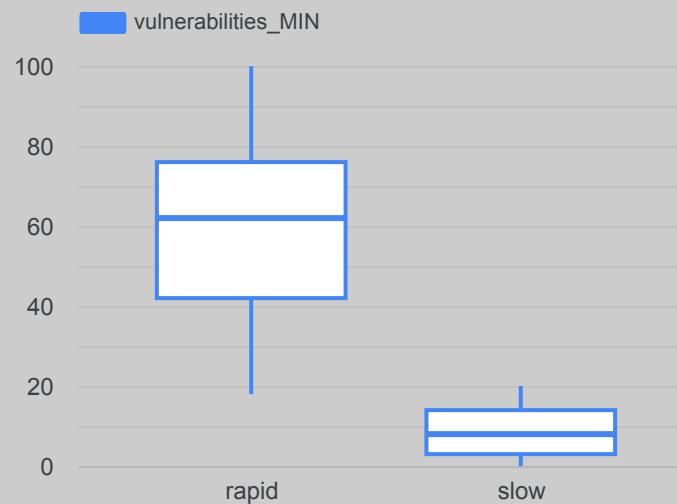
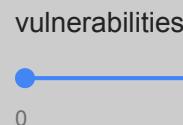
category

### RQ1: O RRC torna as releases mais vulneráveis?

Métricas: vulnerabilities, security hotspots

#### Resposta:

Para ambas as métricas de segurança analisadas, tanto a tendência central (mediana) quanto a dispersão (variabilidade) são significativamente maiores no grupo Rapid Release Cycle (RRC). Isso sugere uma forte correlação entre a velocidade do ciclo de release e o aumento da vulnerabilidade e do risco de segurança.

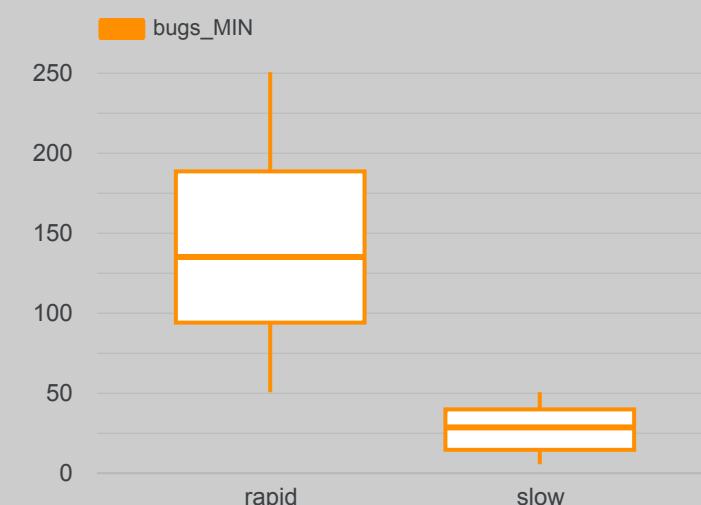


### RQ2: Erros são mais comuns em releases de RRC?

Métrica: bugs

#### Resposta:

Com base na análise da distribuição de bugs, os dados mostram que o Rapid Release Cycle (RRC) está fortemente associado a uma contagem de erros muito maior em comparação com o Slow Release Cycle (SR).



### RQ3: O retrabalho é maior em sistemas que utilizam RRC, do que em sistemas que utilizam releases lentas?

Métricas: technical debt, duplicated lines density

#### Resposta:

Ambos os indicadores de qualidade de código mostram que o Rapid Release Cycle (RRC) está associado a uma qualidade de código significativamente inferior (maior dívida técnica e maior duplicação) em comparação com o SR. Essa falta de qualidade se traduz diretamente em um maior volume de retrabalho futuro para os sistemas RRC.

