



Análise de Métricas Estáticas entre releases em repositórios de linguagem Python no GitHub





Equipe

André Murilo Neves Vasconcelos
Leonardo Antunes Barreto Noman
Lorrayne Reis Silva
Paulo Henrique Cota Starling
Victor Augusto dos Santos





Sumário

1. INTRODUÇÃO
2. TRABALHOS RELACIONADOS
3. METODOLOGIA
4. RESULTADOS
5. DISCUSSÃO
6. AMEAÇAS À VALIDADE
7. CONCLUSÃO



PUC Minas

A decorative graphic on the left side of the slide. It features a large, light blue hexagon in the center. Surrounding it are several smaller hexagons in various shades of blue and teal. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and several smaller nodes connected by lines.

1

INTRODUÇÃO



Objetivo Geral

Utilização da ferramenta Radon para explorar métricas de qualidade, complexidade e manutenibilidade das evoluções de releases de projetos em Python do Github.





Justificativa

- ◇ Questionamento existente na comunidade científica
- ◇ Alto número de releases promulga um sistema :
 - Easy maintenance
 - Less changes
- ◇ Ambos temas estão relacionados a qualidade



A decorative graphic on the left side of the slide. It features a large cyan hexagon with the number '2' inside. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and cyan. Some of these hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and connecting lines.

2

TRABALHOS RELACIONADOS



Understanding metrics based in detectable smells in Python software : A comparative study

- ◇ Code smells effect : maintenance ?
- ◇ Related : falhas + modularidade
- ◇ Semelhança : code smells + metrics + maintenance





Using Metrics to evaluate software system maintainability

- ◇ Metrics analysis : decisions and necessidades
- ◇ Qualidade de componentes
- ◇ Demanda de testes
- ◇ reengenharia
- ◇ RQ3 : advanced state = less maintainability ?
- ◇ + metrics





Cyclomatic complexity density and software maintenance

- ◇ Cyclomatic complexity = more or less maintenance?
- ◇ more cc = more maintenance
- ◇ Same metric : CC related to maintenance ?



A decorative graphic on the left side of the slide. It features a large cyan hexagon with the number '2' inside. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and cyan. Some of these hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and radiating lines.

2

METODOLOGIA



Hipóteses nulas e alternativas

RQ1 - A cada release do software python, existe o aumento de métricas brutas?

Hipótese nula: As métricas brutas (LOC, LLOC e Comentários) permanecem as mesmas durante as dez primeiras releases dos repositórios coletados

Hipótese alternativa: As métricas brutas (LOC, LLOC e Comentários) aumentam durante as dez primeiras releases dos repositórios coletados.





Hipóteses nulas e alternativas

RQ2 - A cada release, os sistemas se tornam mais fáceis de serem lidos e menos arriscados de serem modificados?

Hipótese nula: A cada release os sistemas mantêm o índice de complexidade ciclomática e não se tornam mais fáceis de serem lidos ou menos arriscados de serem modificados.

Hipótese alternativa: A cada release os sistemas aumentam o índice de complexidade ciclomática e não se tornam mais complexos de serem lidos e mais arriscados de serem modificados.





Hipóteses nulas e alternativas

RQ3 - A criação de releases diminui a manutenibilidade do sistema?

Hipótese nula: A cada release os sistemas mantêm a manutenibilidade do sistema.

Hipótese alternativa: A cada release os sistemas ficam mais difíceis e complexos de serem modificados e de receberem manutenção.





Métricas Coletadas

Research Question 1:

- ◇ LOC - número total de linhas de código
- ◇ LLOC - Número de linhas lógicas de código
- ◇ Comentários

Research Question 2:

- ◇ Complexidade ciclomática

Construct	Effect on CC
if	+1
elif	+1
else	+0
for	+1
while	+1
except	+1
finally	+0
with	+1
assert	+1
Comprehension	+1
Boolean Operator	+1




Métricas Coletadas

Research Question 3:

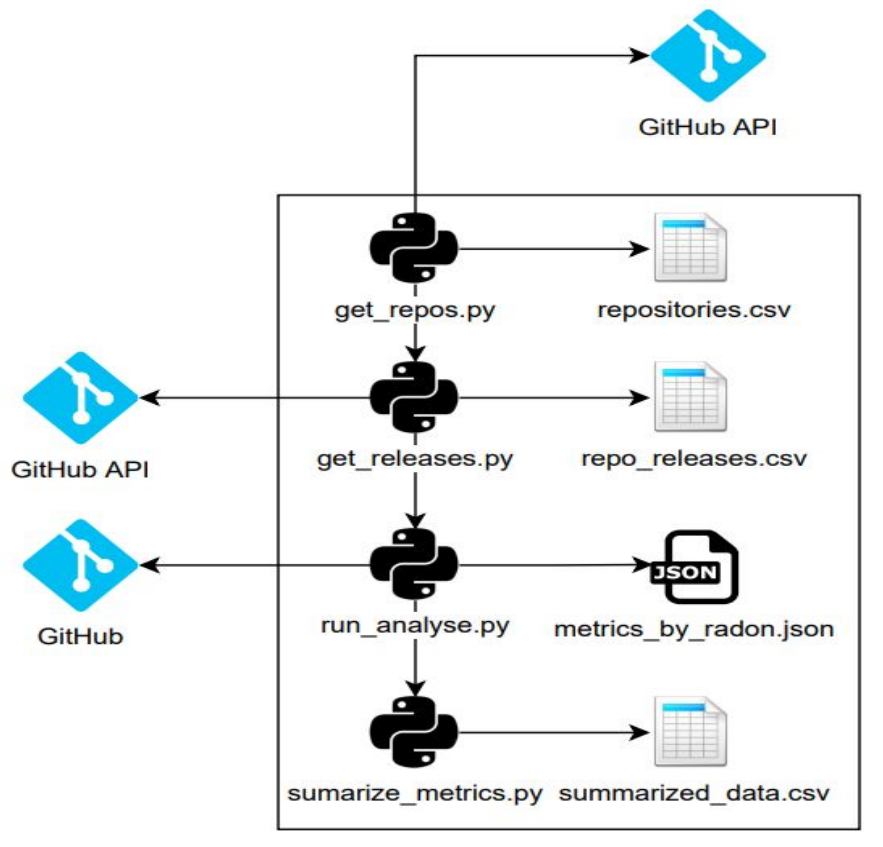
◇ Maintainability Index

$$MI = \max \left[0, 100 \frac{171 - 5.2 \ln V - 0.23G - 16.2 \ln L + 50 \sin(\sqrt{2.4C})}{171} \right]$$

Where:

- V is the Halstead Volume (see below);
 - G is the total Cyclomatic Complexity;
 - L is the number of Source Lines of Code (SLOC);
 - C is the percent of comment lines (important: converted to radians).
- 

Script





RESULTADOS



PUC Minas

Caracterização do Dataset

13,00

Máximo Age

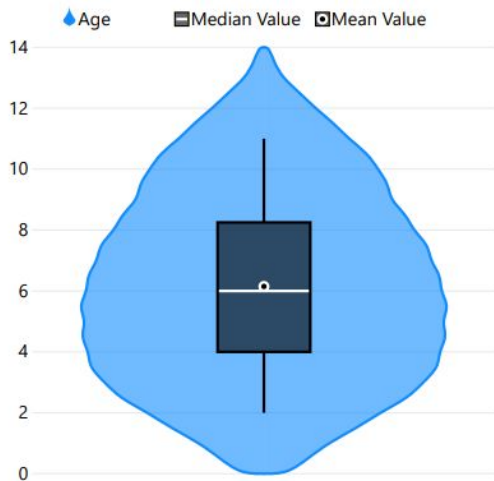
6,15

Média de Age

6,00

Mediana de Age

Idade



1.224,00

Máximo de Num_of_Releases

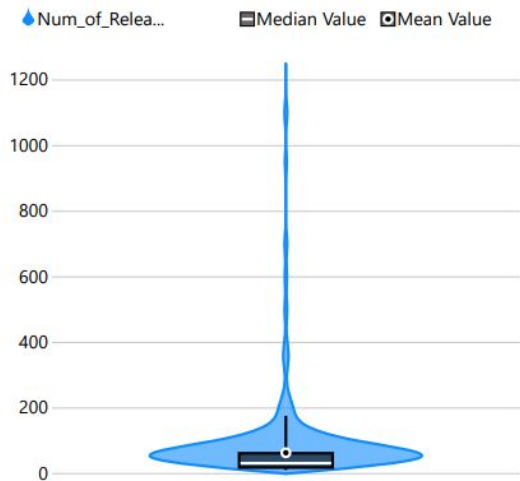
10,00

Mínimo de Num_of_Releases

32,00

Mediana de Num_of_Releases

Número de Release



109.590,00

Máximo de Stargazers

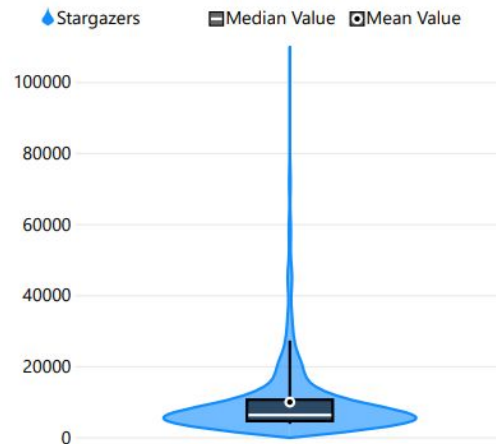
3.803,00

Mínimo de Stargazers

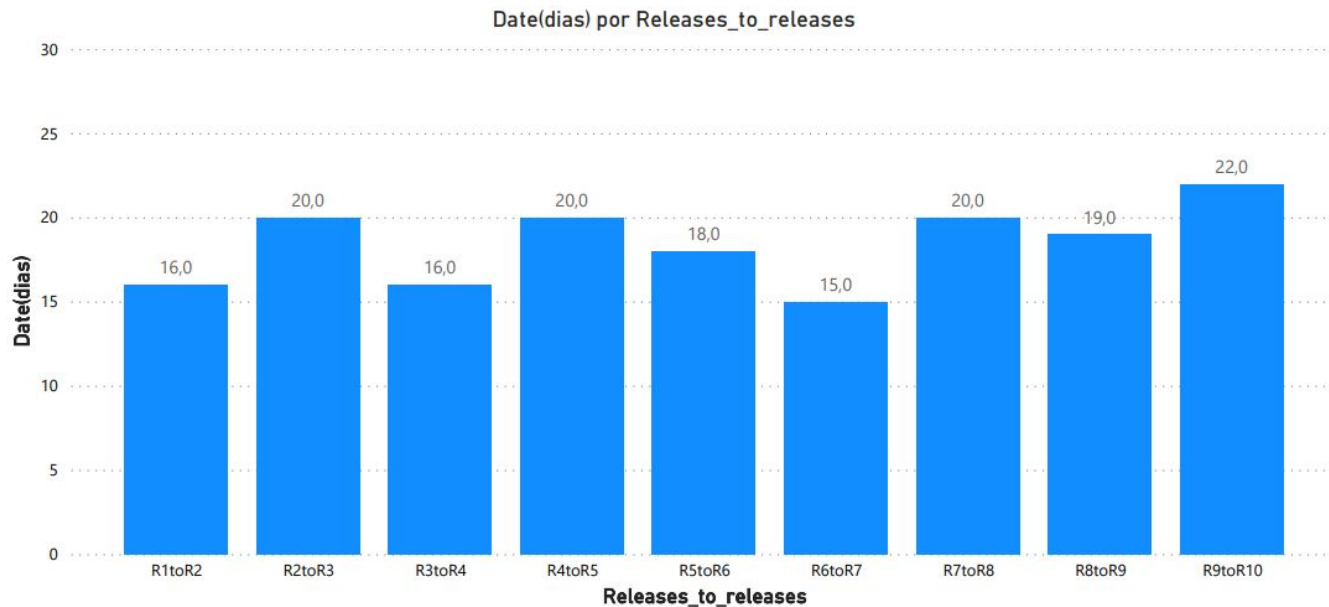
6.411,00

Mediana de Stargazers

Estrelas



Caracterização do Dataset



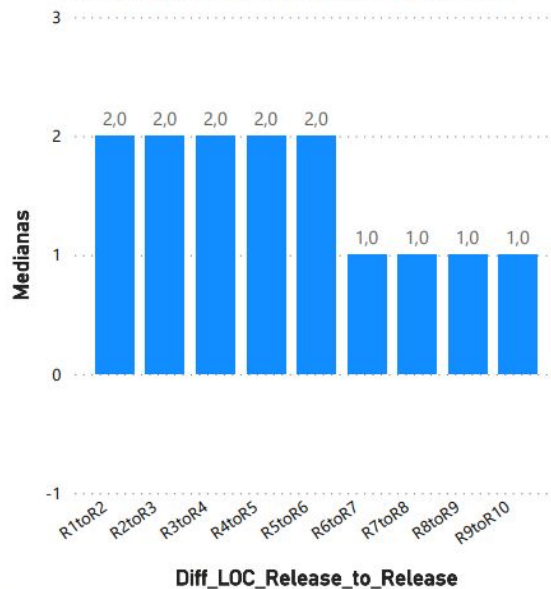


PUC Minas

A cada release do software python, existe o aumento de métrica brutas ?

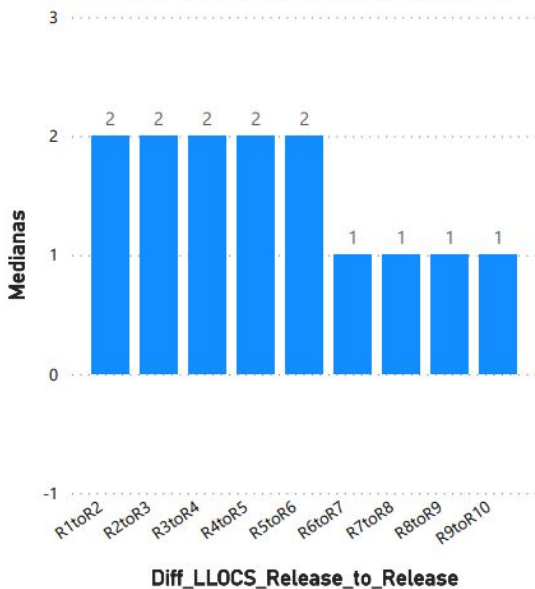
LOCS

Medianas por Diff_LOC_Release_to_Release



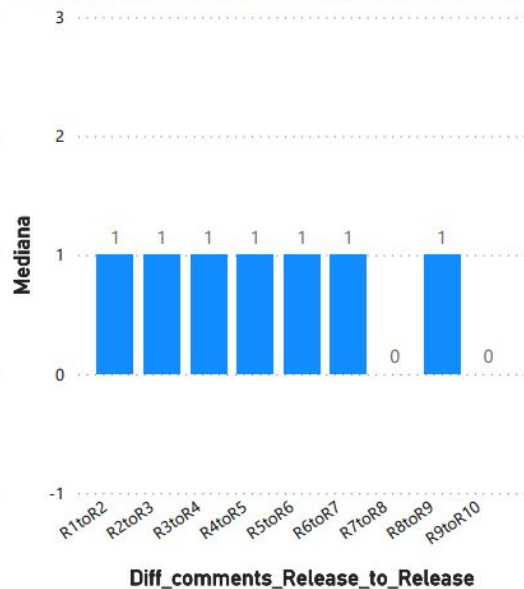
LLOCS

Medianas por Diff_LLOCS_Release_to_Release



COMMENTS

Mediana por Diff_comments_Release_to_Release



A cada release, os sistemas se tornam mais fáceis de serem lidos e menos arriscados de serem modificados?

50.705,00

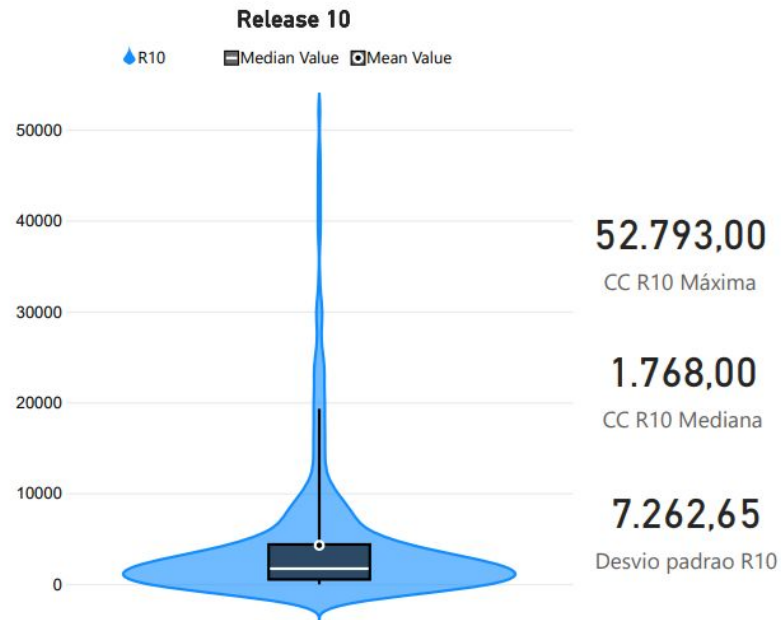
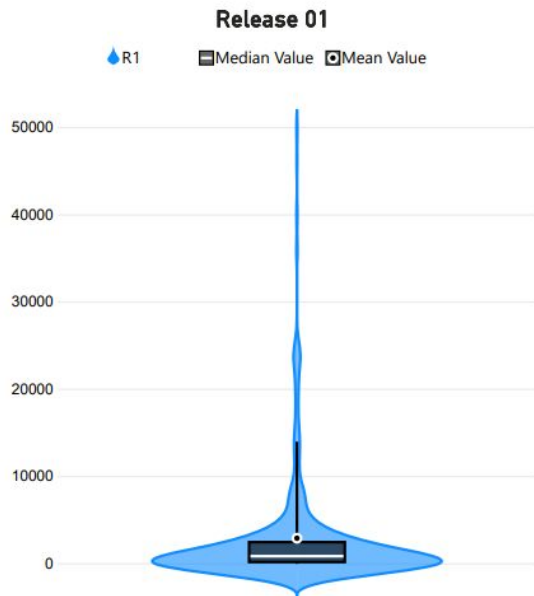
CC R1 Máxima

879,00

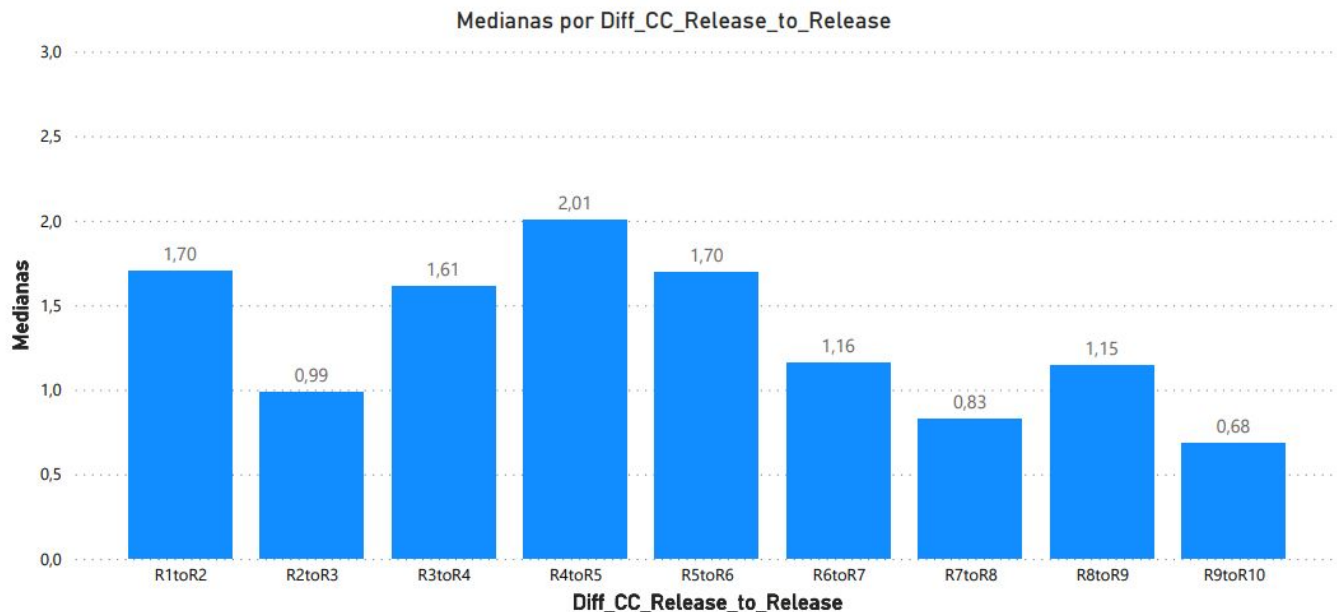
CC R1 Mediana

6.256,94

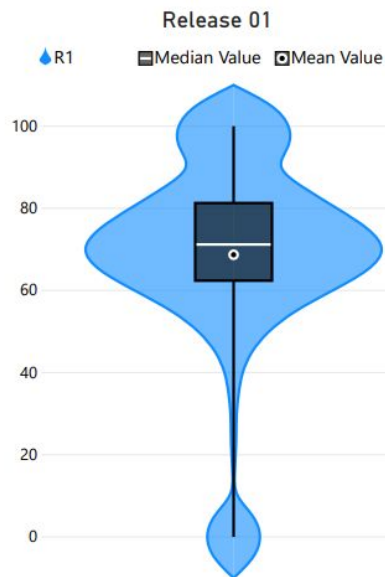
Desvio padrao R1



A cada release, os sistemas se tornam mais fáceis de serem lidos e menos arriscados de serem modificados?



A criação de releases diminui a manutenibilidade do sistema?



100,00

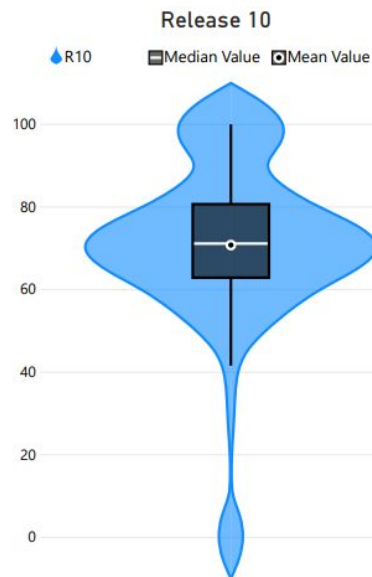
MI R1 Máxima

71,20

MI R1 Mediana

23,81

Desvio padrao MI R1



100,00

MI R10 Máxima

71,14

MI R10 Mediana

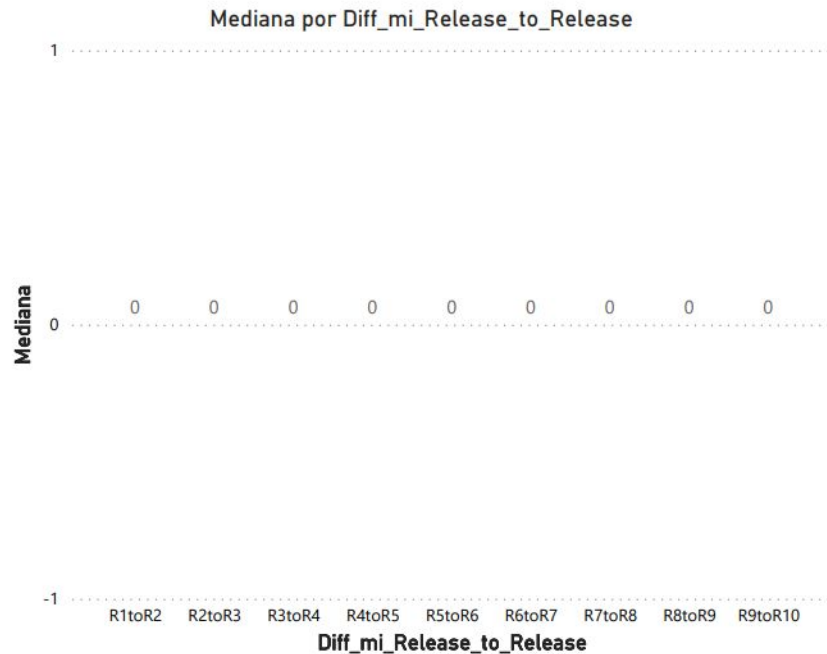
19,97

Desvio Padrao R10 mi



PUC Minas

A criação de releases diminui a manutenibilidade do sistema?



A decorative graphic on the left side of the slide. It features a large cyan hexagon in the center containing the white number '6'. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and cyan. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, a gear, and a speech bubble. There is also a small network diagram icon with a central node and five connecting lines.

6

Ameaças às Validades



Ameaças

- ◇ Validade de construção: interrupção no momento de coleta dos repositórios e análise de métricas pelo Radon.
- ◇ Validade interna: a utilização do radon como ferramenta de inspeção de código é fator não controlado no experimento durante a coleta das métricas.
- ◇ Validade externa: foram analisadas as 10 primeiras releases dos repositórios mais famosos em python. Generalização em relação às próximas releases.
- ◇ Validade de conclusão: análise dos resultados das medianas podem gerar conclusões errôneas.





Conclusão

Observações sobre as métricas coletadas das 10 primeiras releases:

- ◇ Os softwares aumentaram de tamanho a cada release
- ◇ Complexidade Ciclomática também aumentou
- ◇ Índice de manutenibilidade manteve na mediana entre as releases





Obrigado!

Perguntas?

