

# Diagnóstico e Auditoria do Pipeline de Integração Contínua do Projeto Crawn4AI

## Integrantes

- Carlos Daniel Lima de Gois
- David Silva Santana
- Felipe Osni Santos Moura
- João Pedro Cardoso Arruda
- Nicolas Matheus Ferreira de Jesus
- Samuel Bastos Borges Pinho
- Vinícius Vasconi Villas Boas Micska
- Vitor Leonardo Sena de Lima





# Introdução

- Evolução contínua de sistemas de software
- Qualidade e segurança
- Reduzam riscos de regressão
- Facilitem a manutenção
- Crawl4AI
- Mecanismos automatizados de verificação
- Impactos dessas práticas na confiabilidade do projeto e na experiência de novos contribuidores





# Github actions





# Github actions

Enhance your workflow with extensions

Tools from the community and partners to simplify tasks and automate processes

type:actions

Featured

Models

Apps

Actions

All actions

AI Assisted

API management

Backup Utilities

Chat

Code quality

Code review

Code Scanning Ready

Code search

Container CI

Continuous integration

Dependency management

Deployment

Deployment Protection Rules

Desktop tools

Game CI

Actions

Automate your workflow from idea to production

Filter: All

By: All creators

Sort: Popularity

TruffleHog OSS

Find and verify leaked credentials in your source code

Action

Metrics embed

An infographics generator with 40+ plugins and 300+ options to display stats about your GitHub account

Action

yq - portable yaml processor

create, read, update, delete, merge, validate and do more with yaml

Action

Super-Linter

Super-linter is a ready-to-run collection of linters and code analyzers, to help validate your source code

Action

Gosec Security Checker

Runs the gosec security checker

Action

Rebuild Armbian and Kernel

Support Amlogic, Rockchip and Allwinner boxes

Action

Checkout

Checkout a Git repository at a particular version

Action

OpenCommit — improve commits wit...

Replaces lame commit messages with meaningful AI-generated messages when you push to remote

Action

SSH Remote Commands

Executing remote ssh commands

Action

generate-snake-game-from-github-...

Generates a snake game from a github user contributions grid. Output the animation as gif or svg

Action

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Skip this and [set up a workflow yourself](#) →

Search workflows

Suggested for this repository

Simple workflow

By GitHub

Start with a file with the minimum necessary structure.

Configure

Deployment

Deploy Node.js to Azure Web App

By Microsoft Azure

Build a Node.js project and deploy it to an Azure Web App.

Deploy to Amazon ECS

By Amazon Web Services

Deploy a container to an Amazon ECS service powered by AWS Fargate or Amazon EC2.

Build and Deploy to GKE

By Google Cloud

Build a docker container, publish to Google Container Registry, and deploy to GKE.



# Por que YAML?

XML	JSON	YAML
<pre>&lt;Servers&gt;   &lt;Server&gt;     &lt;name&gt;Server1&lt;/name&gt;     &lt;owner&gt;John&lt;/owner&gt;     &lt;created&gt;123456&lt;/created&gt;     &lt;status&gt;active&lt;/status&gt;   &lt;/Server&gt; &lt;/Servers&gt;</pre>	<pre>{   Servers: [     {       name: Server1,       owner: John,       created: 123456,       status: active     }   ] }</pre>	<pre>Servers: -   name: Server1     owner: John     created: 123456     status: active</pre>



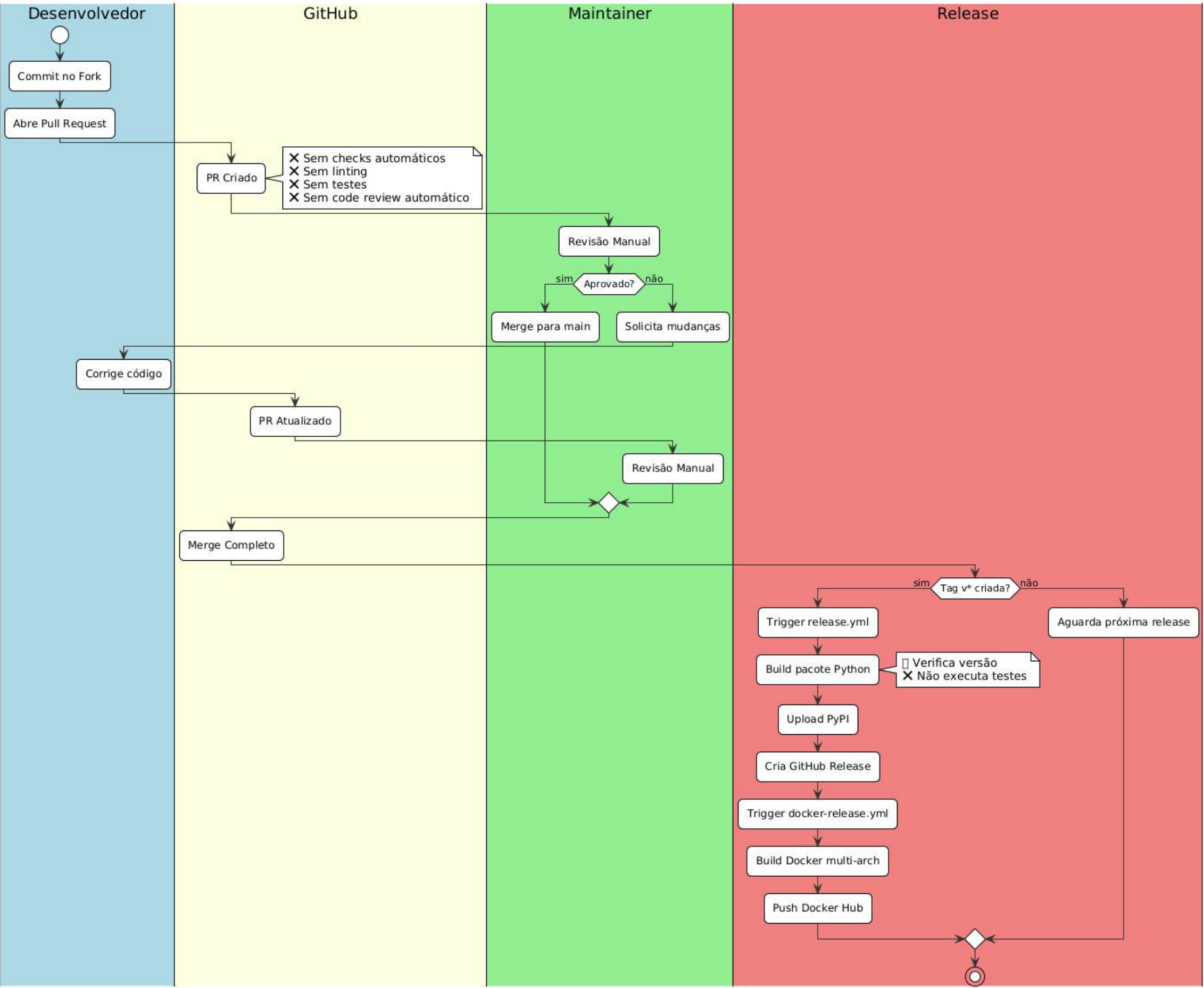


# Mapeamento do fluxo atual & Arquitetura do Github Actions

Como tudo está conectado e build/entrega de imagens Docker



Fluxo Atual do Projeto Crawl4AI



**Início do Fluxo:** O desenvolvedor abre um Pull Request (PR) a partir de um fork.

**Revisão Manual:** A aprovação e o merge dependem exclusivamente da revisão humana do mantenedor.

**Entrega Python:** O envio de uma tag de versão dispara o release.yml para build e upload no PyPI.

**Entrega Docker:** A publicação da release aciona o docker-release.yml para build e push das imagens no Docker Hub.

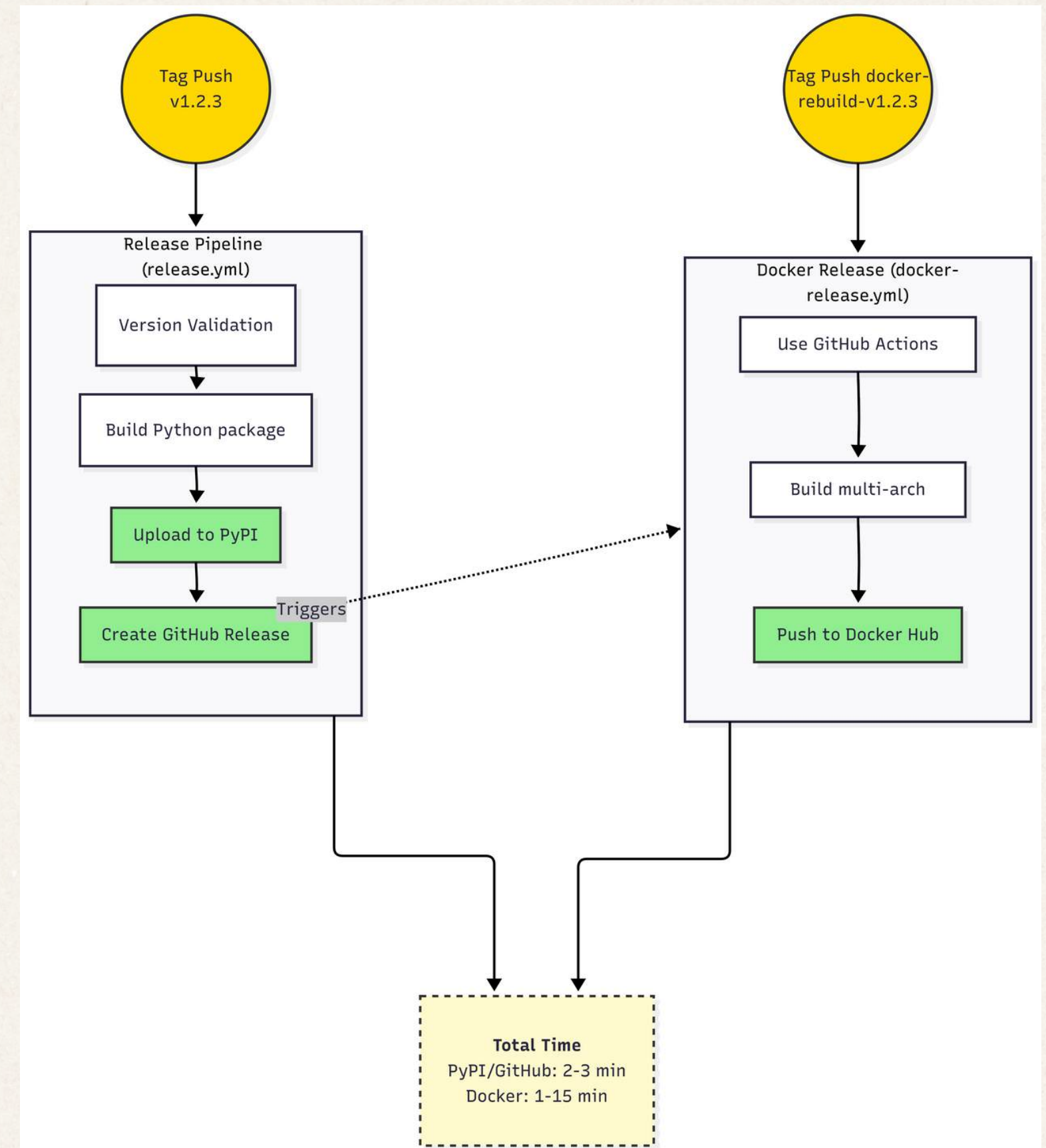
Cor	Significado
Desenvolvedor	
GitHub Actions	
Maintainer	
Pipeline Release	



# Workflow de releases no GitHub Actions

O repositório utiliza de uma arquitetura de split releases pipeline, onde o processo de releases é dividido em dois workflows independentes:

- Release Pipeline
- Docker Release



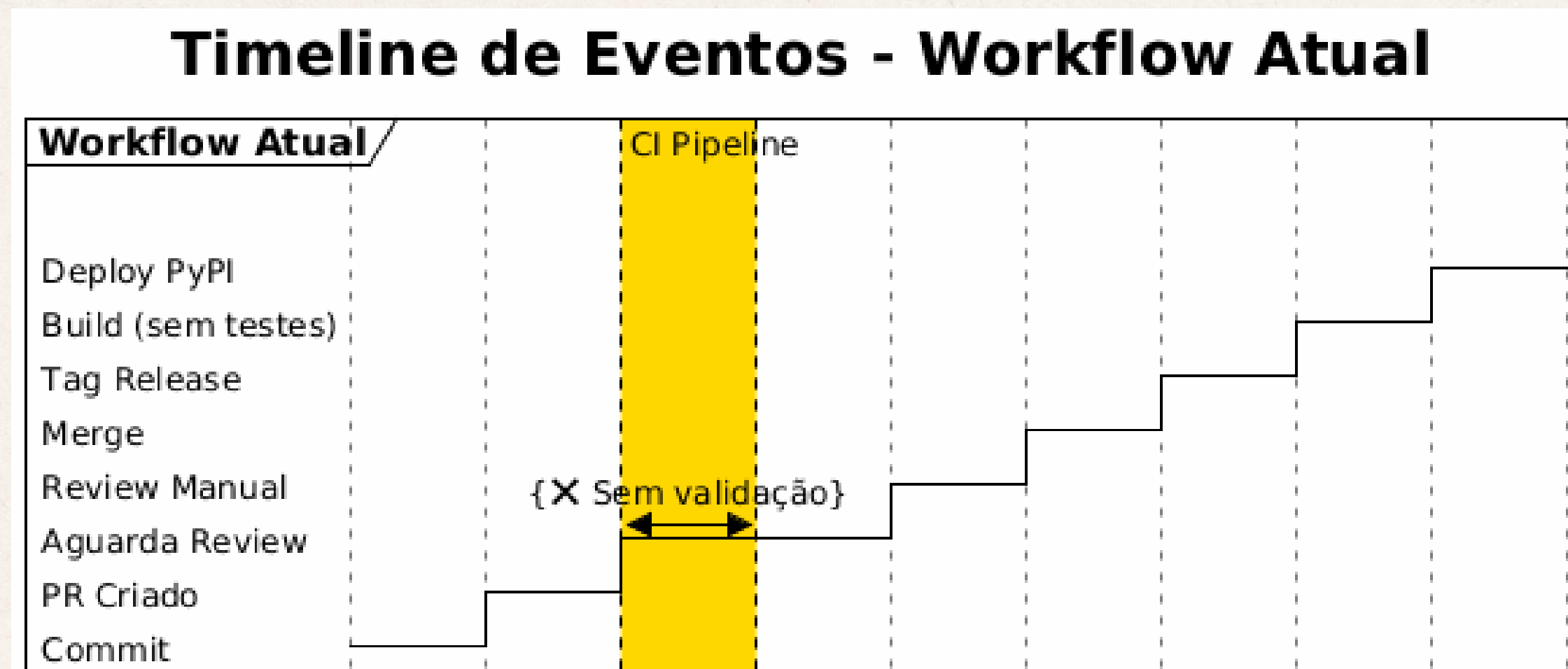


# **Gargalos e Ineficiências do Processo Atual**





# O Fluxo Atual e Seus Pontos de Fricção



O mapeamento revela um processo funcional mas altamente manual. Existem áreas críticas onde a velocidade é sacrificada:

- Gargalo de aprovação humana
- Validação tardia



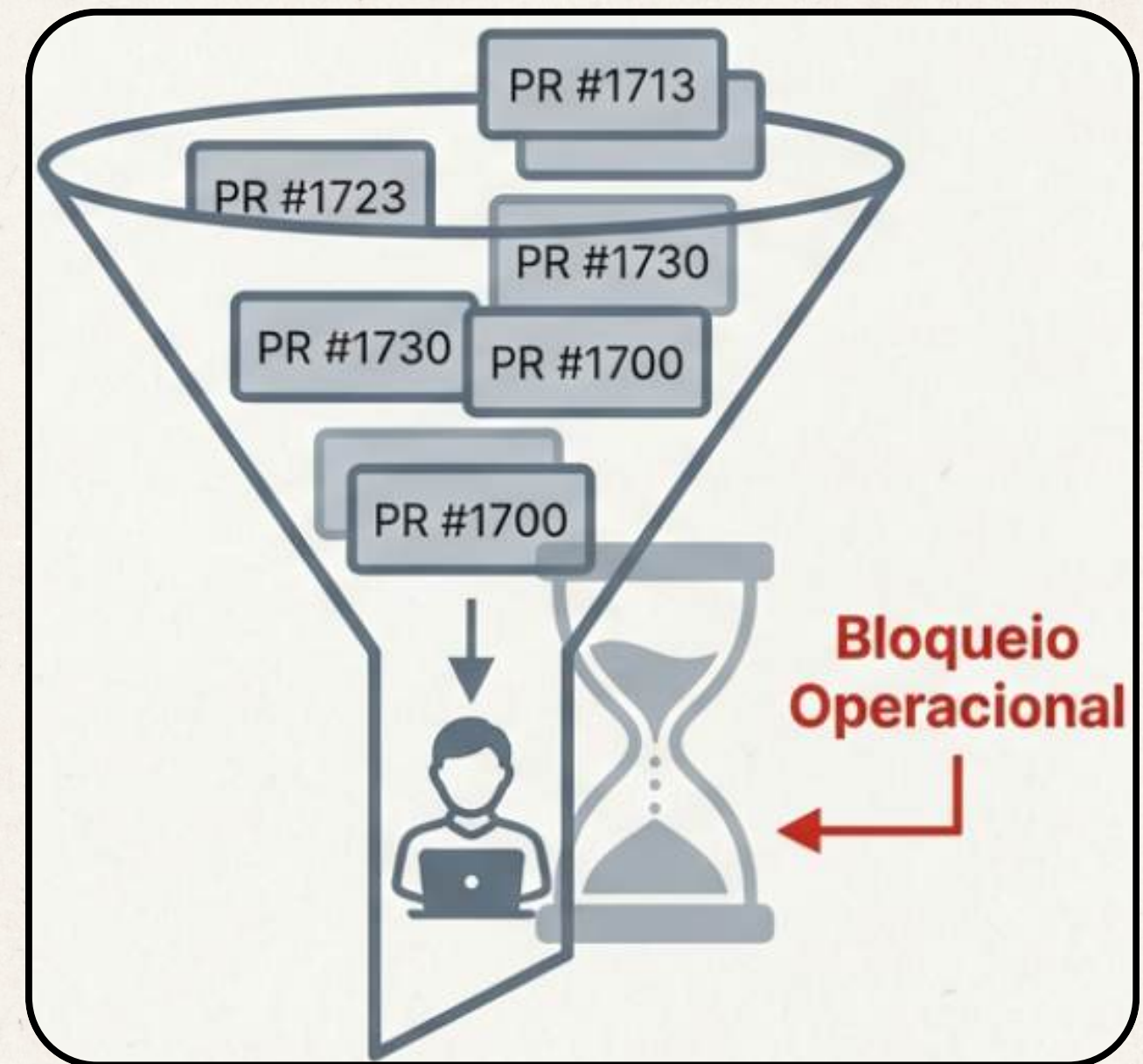
# O Gargalo da Validação Humana

## O Problema: Dependência Crítica

Não existem "Quality Gates" automáticos. Todo Pull Request, inclusive de IA ou documentação, exige tempo e atenção total dos maintainers para ser validado.

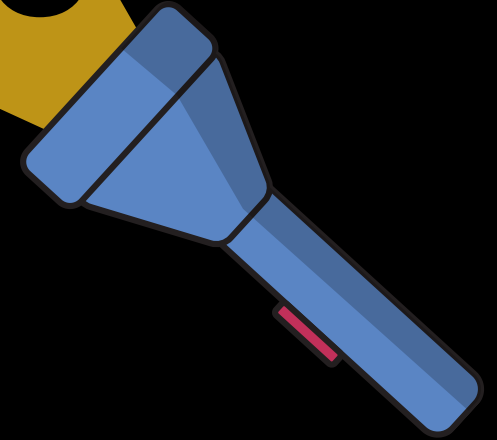
## A Consequência

A velocidade do desenvolvimento é limitada pela disponibilidade humana, criando uma fila de espera nos PRs





# O Desenvolvedor Fica no Escuro



## O Problema: Falta de feedback

Ao submeter o código, não há feedback instantâneo de linter ou testes unitários no PR.

## A Consequência

O revisor é obrigado a apontar erros triviais que uma máquina faria em segundos



# Ruído no Repositório: Gestão Manual de Duplicidade

## Evidência de Ineficiência

Foram identificados múltiplos PRs resolvendo a mesma issue simultaneamente

## O Risco

Sem triagem automática, maintainers gastam tempo revisando código redundante. Isso gera confusão no histórico e frustração para os contribuidores que veem seu trabalho descartado

🔗 123 Open ✓ 286 Closed	
<div>🔗 #1721 Fix Redis task data issue</div> <div>#1721 opened last week ago • Review required</div>	} <b>Esforço Duplicado</b>
<div>🔗 #1703 Resolve Redis data problem</div> <div>#1703 opened last week ago • Review required</div>	
<div>🔗 #1698 Fix: Redis data TTL</div> <div>#1698 opened last week ago • Review required</div>	
#1724 opened 2 weeks ago by git-pranavbabu • Review required	



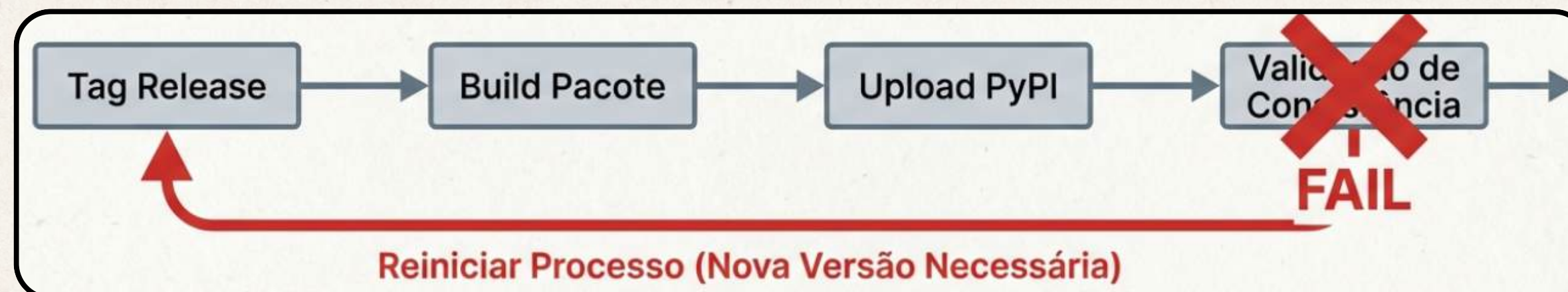
# Versões “Queimadas” por conta da Validação Tardia

## O Bloqueio Técnico

O PyPI é imutável. Se um erro for detectado durante o upload, a versão (ex: v1.2.3) não pode ser corrigida.

## Impacto Operacional

É necessário descartar a versão atual e reiniciar todo o processo com uma nova numeração, gerando ‘lixo’ no histórico e desperdício de tempo de deploy.

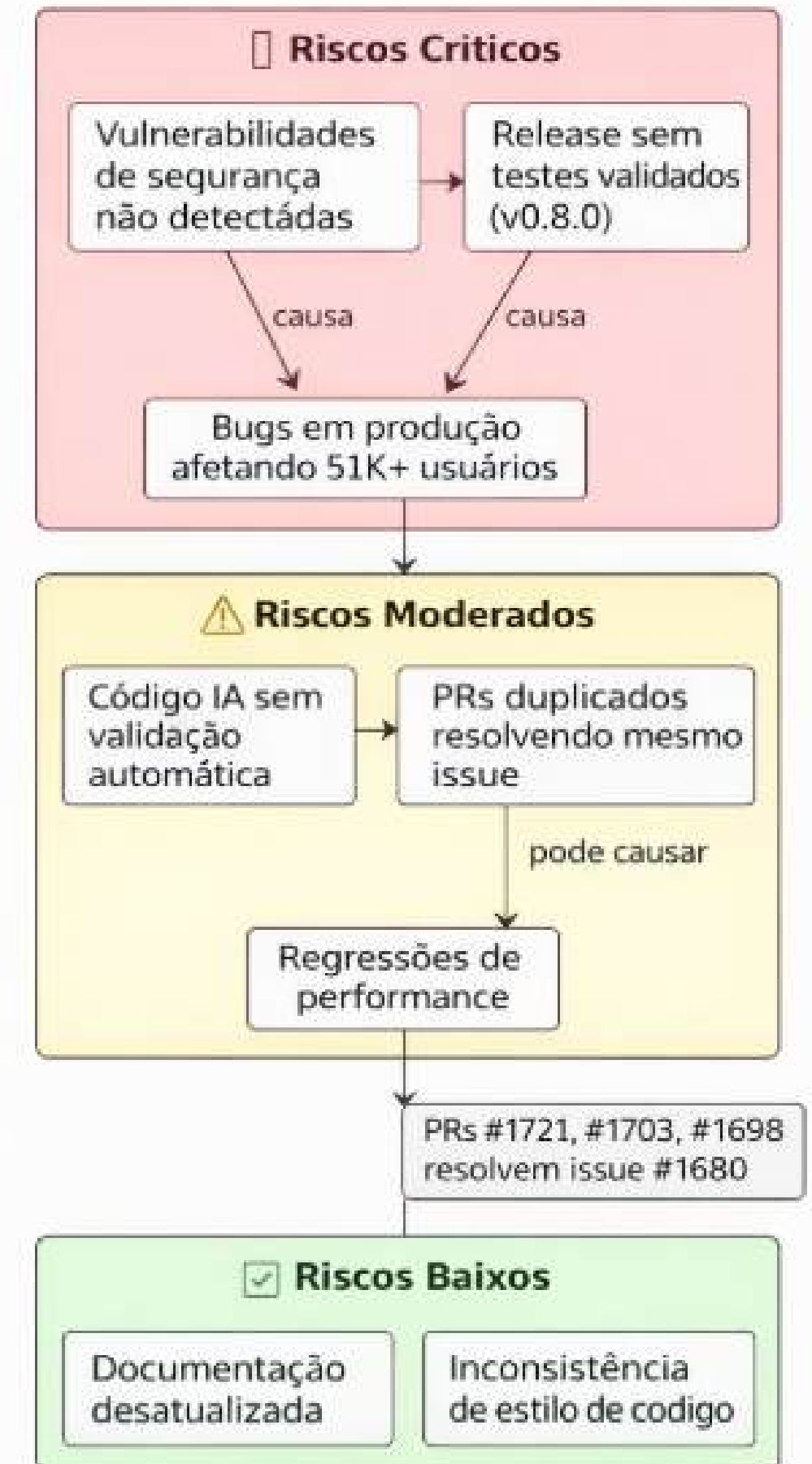




# Riscos Identificados

- Ausência de verificações automáticas:
  - Vulnerabilidades e bugs em produção;
  - Inconsistências de desenvolvimento;
  - Resolução repetida de problemas;
  - Ex.: PyPI (Python Package Index);
- Documentação e novos contribuidores.

## Matriz de Riscos – Ausência de CI/CD





**Obrigado pela atenção.**