

**Perl Diver**

A long time ago in a galaxy far,  
far away....

A long time ago in a galaxy far,  
far away....

(Actually in 2019  
at TPC in Riga)

---

# Measuring the Quality of Your Perl Code

— Dave Cross —  
@davorg / @perlhacks

---

## Measure THEN Improve

In order to improve something you need to have a baseline to measure your improvements against

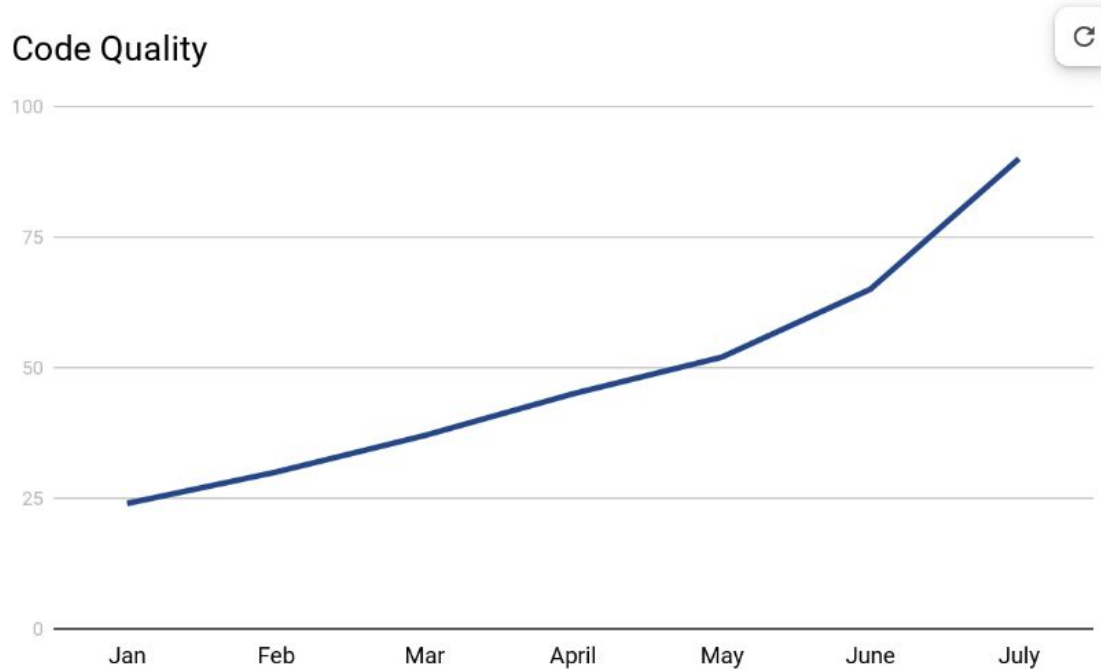
---

## Measuring the Quality of Your Perl Code

A set of values you can track  
over time to demonstrate  
improvements

---

# Measuring the Quality of Your Perl Code



## Measuring the Quality of Your Perl Code

There are no numbers that  
directly and accurately  
measure the quality of code

---



## Four Proxies

- Test passes
  - Test coverage
  - Perl critic
  - Cyclomatic complexity
-

# Continuous Integration

- Travis CI
  - Tests
- Travis CI + Coveralls.io
  - Test coverage
- Kritika
  - Perl critic
  - Cyclomatic complexity

- ~~Travis CI~~
  - Tests
- ~~Travis CI~~ + Coveralls.io
  - Test coverage
- Kritika
  - Perl critic
  - Cyclomatic complexity

- ~~Travis CI~~
  - Tests
- ~~Travis CI~~ + Coveralls.io
  - Test coverage
- ~~Kritika~~
  - Perl critic
  - Cyclomatic complexity

- ~~Travis CI~~

- Tests

- ~~Travis CI~~ + **Coveralls.io**

- Test coverage

- ~~Kritika~~

- Perl critic
  - Cyclomatic complexity

**Hold that  
thought**

A long time ago in a galaxy far,  
far away....



A long time ago in a galaxy far,  
far away....

(Actually in 2023  
at TPC in Toronto)

# GitHub Actions for Perl development



Dave Cross  
@davorg  
dave@perlhacks.com

# GITHUB ACTIONS ESSENTIALS

Automate, Integrate, Deploy:  
Unlocking the Power of GitHub  
Actions

Written by  
DAVE CROSS

# GITHUB PAGES ESSENTIALS

Creating, Customizing, and  
Publishing Your Web Projects

Written by  
DAVE CROSS

# GITHUB ACTIONS ESSENTIALS

**Automate, Integrate, Deploy:  
Unlocking the Power of GitHub  
Actions**

Written by  
**DAVE CROSS**



# GITHUB PAGES ESSENTIALS

**Creating, Customizing, and  
Publishing Your Web Projects**

Written by  
**DAVE CROSS**

# GitHub Actions + GitHub Pages

- That's where the magic happens
- GitHub Actions runs a CI process
  - Writes data to a database
  - Generates website from database
- GitHub Pages hosts your website



**Some revision**

## Some revision

- What is CI?
- What is GitHub Actions?
- What is a build radiator?
- What is GitHub Pages?





# What is CI?

- Continuous Integration
- Frequent Code Integration
- Automated Testing
- Fast Feedback
- Collaborative Development
- Early Detection of Issues



**And it all  
happens  
automatically**

# CI systems

- Jenkins
- Travis-CI
- Circle CI
- GitHub Actions



# What is GitHub Actions?

- CI system built into GitHub
- Run any code you want
- In response to a number of triggers



# Trigger events

- Time and date
- Commits
- Pull requests
- Issues
- etc...



# Run where?

- GitHub runners
- Linux/Mac/Windows
- Docker images
- Self-hosted runners



# Run what?

- Pretty much anything
- You have a Docker container
- Code in your repo
- Install other tools
- Tests/linters/etc.
- Static site generator



# What is a build radiator?

- A display of useful data about a project
- Quality proxies
- Displayed on a screen near the team
- Team can see their progress
- Management loves this stuff





# What is GitHub Pages?

- Web pages stored in GitHub repos
- Web pages hosted on GitHub servers
- Static web pages
  - HTML/CSS/Javascript/Images
  - No backend



## Static site :- (

- GitHub Pages sites can only contain static assets
- No backend server available
- Not necessarily a restriction



## Semi-static site

- Use GitHub Actions to rebuild GitHub Pages sites
- On a schedule
- On a push
- etc...



# Semi-static example

- "Planet" site
  - Planet Perl - [perl.theplanetarium.org](http://perl.theplanetarium.org)
- RSS/Atom aggregator
- Rebuild on a schedule
- Rebuild on template change
- Rebuild on code change



**Putting it all  
together**

# The beginning of a plan

- We can use GitHub Actions to run CI
- In a specific manner
- Generate data
- Store that data in a database



# The beginning of a plan

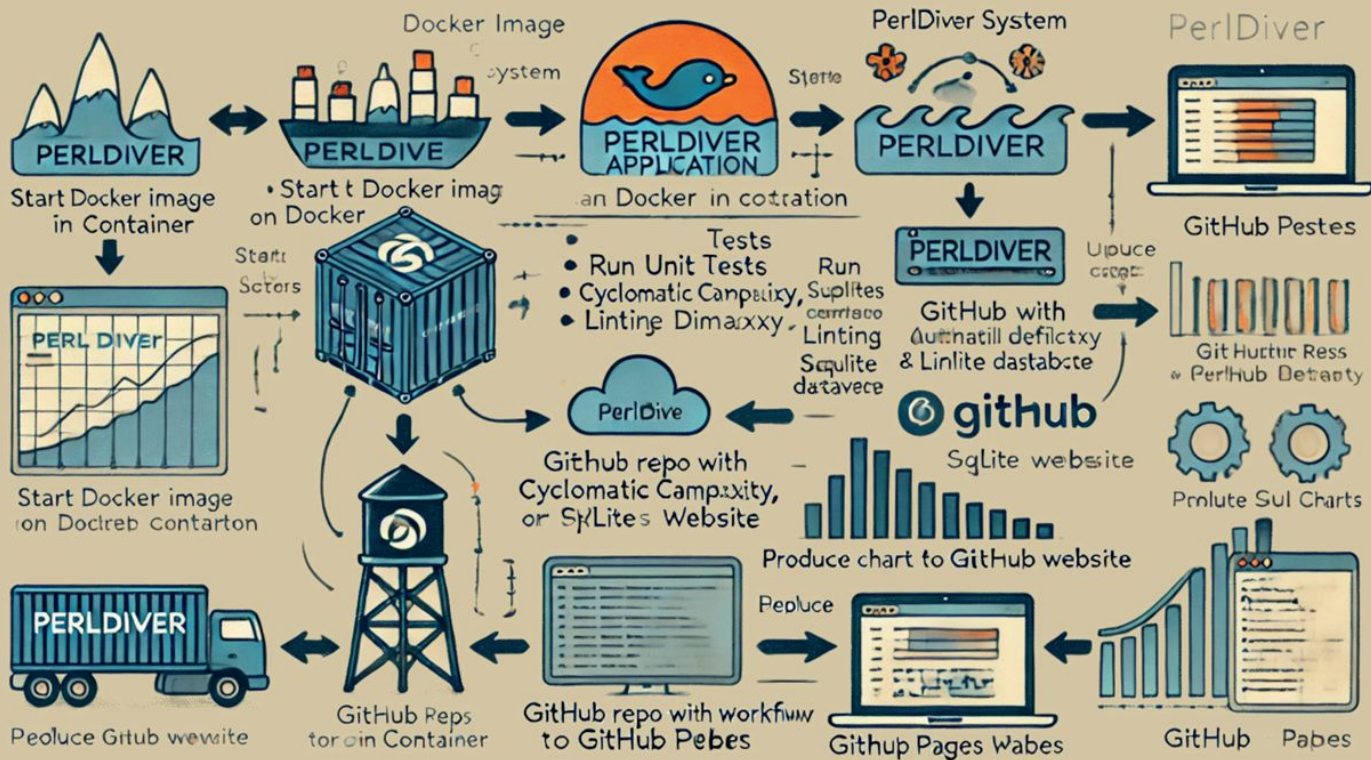
- We can use GitHub Actions generate a web site
- Use the data in the database
- Generate pages
- Publish to GitHub Pages



**The Devil is in  
the detail**



**I described the  
architecture to  
ChatGPT**

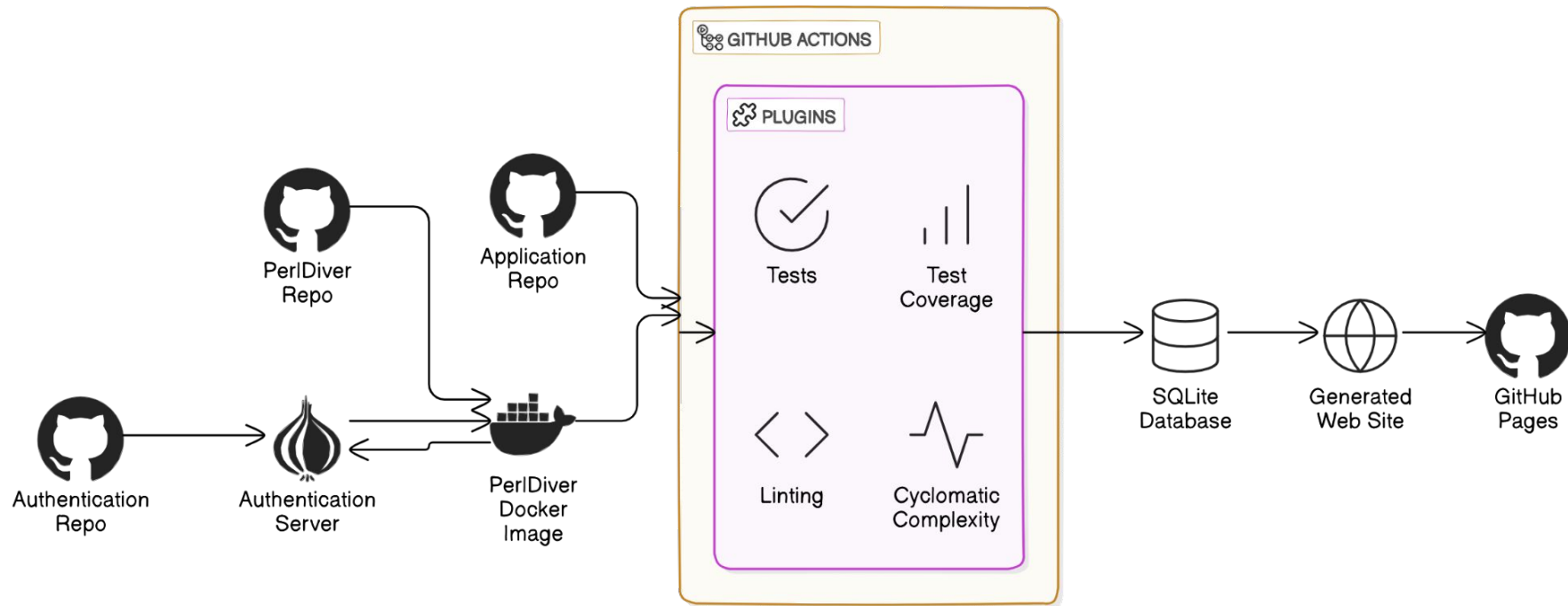


## Try again

- I asked Eraser.io
  - AI to draw architectural diagrams
- It did better



## PerlDiver CI Architecture



## Three GitHub repos

- Perl Diver code
  - Used to build a Docker container
- Perl Diver Authentication server
- Your code



# Additions to your repo

- Perl Diver needs to add things to your repo
- GitHub Actions workflow definition
- Perl Diver database (SQLite)
- GitHub Pages configuration



# GitHub Action workflow definition

- YAML file
- Defines when the workflow is triggered
- And what the workflow does



# GitHub Action workflow definition

```
name: Perl Diver
on:
  push:
  workflow_dispatch:

jobs:
  perl-diver:
    runs-on: ubuntu-latest
    container: davorg/perl-diver:latest
```





# GitHub Action workflow definition

steps:

- name: Checkout repo  
uses: uses: actions/checkout@v4
- name: Run Perl Diver  
run: perl-diver \${{ github.repository }} \${{ secrets.api\_key }}
- name: Commit database  
run: #standard stuff
- name: Update pages artifact  
uses: actions/upload-pages-artifact@v3  
with:  
path: docs/



# GitHub Action workflow definition

```
deploy:
  needs: build
  permissions:
    pages: write
    id-token: write
  environment:
    name: github-pages
    url: ${ steps.deployment.outputs.page_url }
  runs-on: ubuntu-latest
  steps:
    - name: Deploy to GitHub Pages
      id: deployment
      uses: actions/deploy-pages@v4
```



# Database

- Small SQLite database
- Stored in **db** directory
- Contains data from previous runs
- "No user-serviceable parts"



# Website

- Written to the **docs** directory on the container
- Uploaded to GitHub Pages server
- Not checked into repo
- Access at `username.github.io/reponame`
- Some (one-off) setup required



# Website

- What does the website look like Dave?
- I'm glad you asked
- I am not a web designer
  - But Bootstrap and GitHub Copilot help!



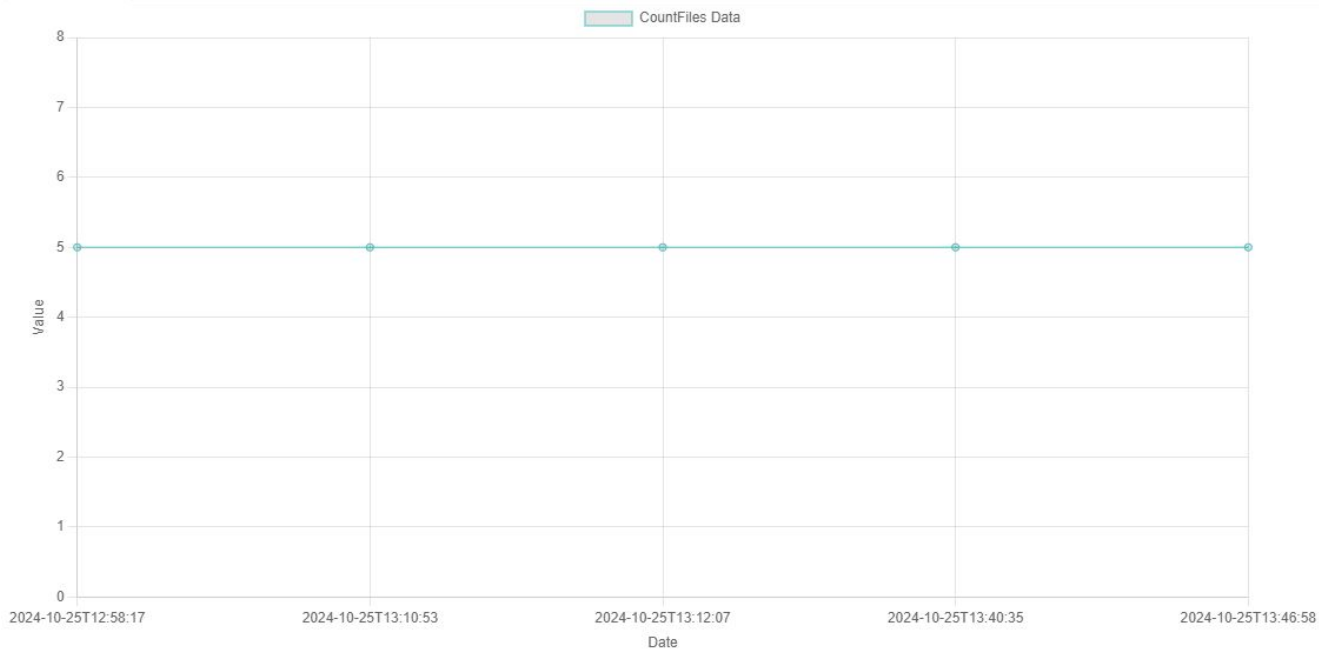


# davorg-cpan/array-compare

Most recent run: 2024-10-25T13:46:58

CountFiles

PodCoverage



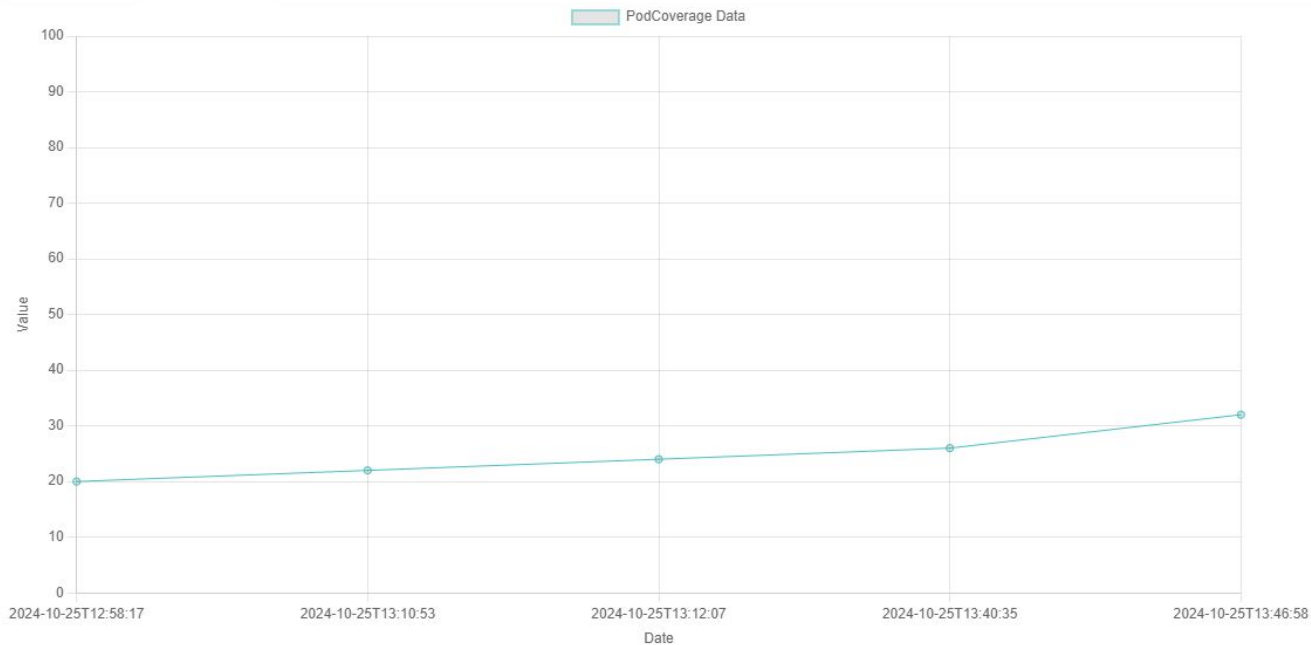


# davorg-cpan/array-compare

Most recent run: 2024-10-25T13:46:58

CountFiles

PodCoverage



# Graphs

- Using `chart.js`
- Default display for simple plugins
- Plugins can override any part of the chart
- Or completely do everything themselves





# Plugins

- Plugin architecture
- Standard plugins for the quality proxies
- Other people can write plugins
  - Once I've documented how that works
- Plugins store JSON in the database



# Conclusion

- Code measurement is cool
- Code measurement is useful
- Managers ❤️ this stuff
- PerlDiver makes it easy



# Contact Information

- Dave Cross - [dave@perldiver.dev](mailto:dave@perldiver.dev)
- [perldiver.dev](http://perldiver.dev)
- Take a postcard!



**Any questions?**

