

# From Neglected to Community-Driven

Reviving Percona Toolkit

 **Sveta Smirnova**  
Principal Support  
Engineering  
Coordinator

# Percona Toolkit History

- Maatkit, Aspersa by Baron Schwartz
- Percona Toolkit
  - 2008 Baron Schwartz
  - 2008-2016 Daniel Nichter
  - 2016 Frank Cizmich
  - 2016-2022 Carlos Salguero
  - 2022- Sveta Smirnova

# Percona Toolkit History

- Maatkit, Aspersa by Baron Schwartz
- Percona Toolkit

-2008 Unicorn

2008-2018 Mature, actively developed product

2018-2022 Loved by customers and services while **abandoned** by Engineering

2022- Trying to improve

# Written in **Perl**, **Bash**, and **Go**

<code>align</code>	<code>galera-log-explainer</code>	<code>online-schema-change</code>	<code>secure-collect</code>
<code>archiver</code>	<code>heartbeat</code>	<code>pg-summary</code>	<code>stalk</code>
<code>config-diff</code>	<code>index-usage</code>	<code>pmp</code>	<code>summary</code>
<code>deadlock-logger</code>	<code>ioprofile</code>	<code>query-digest</code>	<code>table-checksum</code>
<code>diskstats</code>	<code>k8s-debug-collector</code>	<code>replica-find</code>	<code>table-sync</code>
<code>duplicate-key-checker</code>	<code>kill</code>	<code>replica-restart</code>	<code>table-usage</code>
<code>eustack-resolver</code>	<code>mext</code>	<code>show-grants</code>	<code>upgrade</code>
<code>fifo-split</code>	<code>mongodb-index-check</code>	<code>sift</code>	<code>variable-advisor</code>
<code>find</code>	<code>mongodb-query-digest</code>	<code>slave-delay</code>	<code>visual-explain</code>
<code>fingerprint</code>	<code>mongodb-summary</code>	<code>slave-find</code>	
<code>fk-error-logger</code>	<code>mysql-summary</code>	<code>slave-restart</code>	

# Language Issue

- Perl is not popular language anymore
- All other Percona products moved to other languages
- Only Perl developer for Percona Toolkit was reporting to PMM team lead

# The Outcome

- Only Perl developer in Percona
  - Worked on the Percona Toolkit alone
  - Forced to work on PMM most of the time
  - Did not have resources to improve the product
- Review process was broken
- QA stopped after MySQL 8.0 support was added
- Services needs were not heard
- Customer's requests were implemented after PMM

# My Original Plans in 2022

- Work on Percona Toolkit together with Carlos
- Sveta
  - Update tools to support modern databases features
  - Implement features for the Support team

# My Original Plans in 2022

- Work on Percona Toolkit together with Carlos
- Carlos
  - Continue working on customer's escalations and whatever bugs and features he or his manager wants

# My Original Plans in 2022

- Work on Percona Toolkit together with Carlos
- Together
  - Restore review and QA processes
  - Review and accept some community contributions

# My Original Plans in 2022

- Work on Percona Toolkit together with Carlos
- **In late 2022 Carlos left Percona**

# Actual State in Late 2022

- Tools are outdated and do not support modern database features
- Features, necessary for the Support team, are not implemented
- No resources to fix escalated customer bugs timely
- Go tools are hardly useful
- Documentation for Go tools is not accurate
- Community contributions wait the answer for ages
- No QA process
- Perl and Go code are built separately
- **Only developer working on the product in free time again**

# My Plan

- Do work myself
- Encourage others to help

# Keep Good Quality Code

- No regression bugs
- Follow style of existing tools
- Document new additions or interface change

# Working with Contributions



# Auto-generated Code

- Fat Perl modules
- Command options
- User manual
- **Perl and Bash tools only**

# Quick Fixes

- Minimal contribution guide
  - Code layout
  - Differences between Perl, Bash and Go tools
  - Build instructions
- Consistency for Go tools user docs

# Regression Perl Tests: Stage 1

- Fixed suite
- Runs on my machine
- Many tests fail for various reason
  - Broken code
  - Broken test
  - MySQL version compatibility
- **Accepted that not all tests are perfect**

# First Contributions Review

- Read and commented them all
- **We had 5 years old contributions without any comment!**
- Accepted good quality ones with regression tests
- Asked contributors to fix others

# One Month Later

- Accepted contributions with suggested edits
- No answer *I cannot expect someone would be happy to work with us after 5 years of silence!*
  - Wrote tests for good quality PRs, then accepted
  - Improved small PRs suggesting useful changes, then accepted
- Scheduled work with bigger PRs that cannot be accepted as is
- Thanked all accepted contributors in the release notes and in the blog
- **Number of open contributions is manageable and we do not look like dumb people ignoring community**

# Regression Perl Tests: Stage 2

- Fixed all broken tests
- Tested with Percona Server for MySQL 5.7 and 8.0
- **All tests pass on my laptop**

# Contribution Acceptance Rules

- New PR must be reviewed by me as soon as possible
- All new code should have regression tests
- Utility `update-modules` should not break the fix
- Documentation updated when needed
- At least 2 engineers must review suggested change
- Existing regression tests should pass on my laptop
- Release notes must contain name of the contributor

# Regression Tests: Stage 3

- Perl regression suite is running on all supported machines prior release
- Work on Go tests sandbox and unification is in progress
- **We can work with contributions without fear to break something!**

# Help from Colleagues

- There are Perl experts in Percona
- Few Go engineers help with Go contributions
- Documentation team helps with release notes and user manual building
- QA team guides me on automation for pre-release tests
- Build team is building packages
- Services engineers contribute

# Contribution Acceptance State

- All contributions receive fast feedback
- Imperfect PRs are improved, then accepted
- Rejections happen as soon as possible and always with feedback
- I am happy to help with code improving
- Contributor name is published in the release notes

# Result of Changes

- We regularly receive new PRs
- Quality of PRs is improving
- Many bugs are fixed by community
- We have contributions for GitHub workflows that allow us to have better quality code automatically

# Outcome

- Feedback is important
- Not perfect code is OK
  - Contributor still did great work on investigating and designing the fix
  - Suggested fix interface in GitHub is a great tool for improving PRs
  - Rewriting submitted code returns in the future as better contributions
- Community comes with really great ideas
- **Sometimes someone you don't know fixes important bug you was going to spend days on!**

Thank you!



# Community Contributors

- Ernie Souhrada
- Kazuya Yokogawa
- Kushal Haldar
- monty solomon
- PinoCao
- Viktor Szepe
- Marcelo HP Ferreira
- Jason Ng
- Zongzhi Chen
- Yijian Zhang
- Jakob
- fraff
- Ilaria Migliozi
- Ayush Goyal
- Chrys Swingler
- ikruglov
- Satoshi MITANI
- Alexey Kopytov
- Fernando Ipar
- Alexey Stroganov
- GOKUL S
- Henning Pottker
- Perry Harrington
- Nilnandan Joshi
- Hartley McGuire
- Viktoras Agejevas
- Taehyung Lim
- Nyele
- Marek Knappe
- Maciej Dobrzanski
- Pawel Kudzia

# Percona Engineers

- Perl code review
  - Evgeniy Patlan
  - Saikumar VS
- Go code review
  - Artem Gavrilov
  - Jiri Ctvrtka
- Documentation
  - Alina Derkach
  - Alex Firu
  - Nastya Aleksandrova
- Build
  - Evgeniy Patlan
  - Vadim Yalovets
- QA
  - Eleonora Zinchenko
  - Hrvoje Matijakovic
  - Yash Panchal
- Contributing
  - Yoann La Cancellera
  - Agustin Gallego
  - Marco Tusa
  - Michael Patrick
  - Sami Ahlroos
  - Jobin Augustine
  - Roberto de Bem
  - Matthew Boehm
  - Iwo Panowicz
  - Jiri Ctvrtka

# More Information



Contributing How-To



Release Notes



[github.com/percona/percona-toolkit](https://github.com/percona/percona-toolkit)



 /svetsmirnova  
 /svetsmirnova

 /svetasmirnova  
 /SvetaSmirnova