

May' 24

Webinar Recap

Perfana ADAPT

The Future of Continuous Performance Engineering



Daniel Moll,
Performance Engineer,
Co-Founder | Perfana



Scott Moore,
Performance Engineer,
Scott Moore Consulting

What is Continuous Performance Engineering?

It is not just running a lot more tests.

It is **a holistic approach** and a practice of integrating performance into every area of your development cycle in an **automated** and **continuous** way.

Continuous Performance Engineering (CPE) can be daunting because you **have to consider many things**. That's why there is a need for tooling to help teams to implement CPE successfully.



The key elements of CPE:

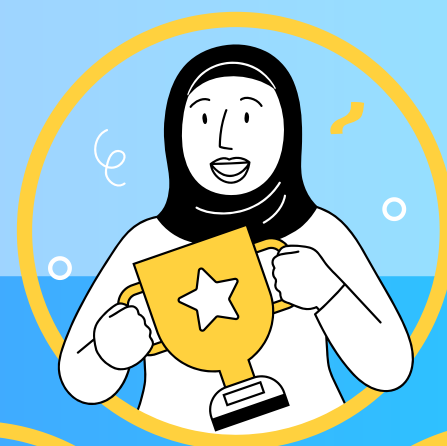
- **Shift-left** performance testing.
- **Incorporate** performance testing into **CI/CD** to give teams immediate feedback on how their code changes impact performance.
- **Glass box** approach:

The analysis includes resource metrics and other signals from System Under Test. This provides **actionable insights** and makes addressing performance issues more **effective**.



Teams that benefit from CPE the most

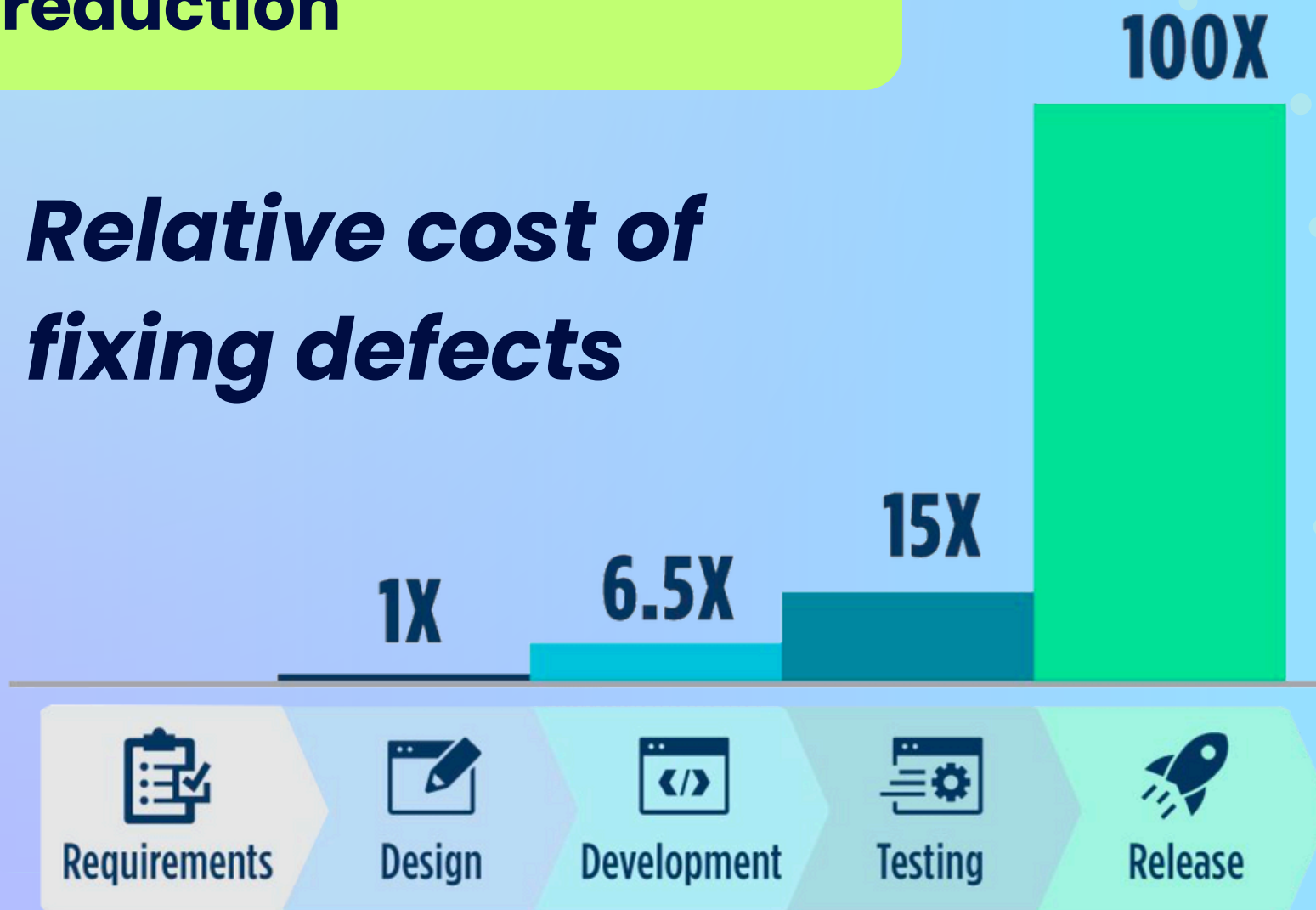
- Teams that want to **release frequently**
- Teams that have **implemented CI/CD**
- Teams that are **already** performance testing and acknowledge its value
- Teams that can't afford to rely on observability only



What is the added value of CPE?

- **Cost reduction**

Relative cost of fixing defects



Source: IBM Systems Sciences Institute

It is a lot **cheaper** to fix issues early on.

If CI/CD pipeline is set correctly most of them can be caught in the development phase.

- **Faster Time to Market**

Defects caught earlier can be fixed **faster**.

Challenges & Solutions

1 False Positives & Variability



This is when the feedback loop raises a flag for detecting **regression** while **code changes** introduced by the team **did not cause the problem**.

It leads to:

- > **Wasting** resources on non-issues
- > **Delay** of release
- > **Undermines** trust in the testing process



The main reason for false positives is the **variability** of test results. It's crucial to ensure you're not comparing apples to oranges.

Challenges & Solutions

1 False Positives & Variability



Factors impacting variability:

- **Hot vs cold caches**, affecting response times and other metrics.
- The **order** and **randomness** of test data used by your load test script.
- **Cloud factor**.
- **External noise** (e.g. deployment settings, infrastructure settings, application properties).

Solution

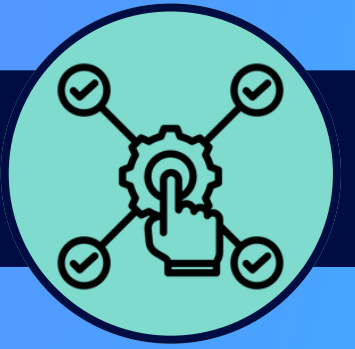
- **Limit the test scope** to the System Under Test (isolate SUT by stubbing calls to backends).
- Create **the same starting conditions** for each test run. Perfana offers plugins to help with this.
- Capture and store **metadata** test environment.
- Use **multiple** test runs as the **baseline**.



Challenges & Solutions

2

Analysis At Scale



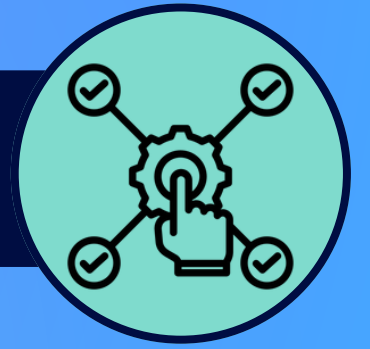
There is **so much test data** that you have to pull, process, and compile for the wanted results. **Analyzing the test results** and doing root cause analysis **at scale** becomes impossible without automation.



Challenges & Solutions

2

Analysis At Scale



- Multiple SUT's x Multiple metrics x Multiple builds/day.
- Defining **correct thresholds** for performance and resource metrics is challenging.
- **Issues** with explicit **static comparing thresholds**.

Solution

- Framework for **automated analysis** of performance test results. Process massive data in seconds without errors and inconsistencies.
- **Anomaly detection** on all metrics captured during tests (e.g. Perfana ADAPT addresses the issue of the static thresholds).
- **Augmented** root cause analysis in case of detected regression.



Challenges & Solutions

3

Team Involvement



Involving the team in continuous performance testing can be **challenging**. Teams are forced to choose between **development speed** or having **performance properly tested** every release.



Challenges & Solutions

3 Team Involvement



- **Cognitive load** / DevOps burnout.
- Struggle to get **priority** from product owners.
- High entry **barriers**. Performance testing is daunting if a team has to start from scratch.

Solution

- **Perfana** allows teams to have both **Speed & Performance**.
- Make onboarding as **frictionless** as possible.
- Allow teams to use the **preferred load test tool**.
- Involve teams in **root cause analysis**.
- **Minimize** false positives!





Perfana ADAPT

- Automated
- Deviation
- Analysis for
- Performance
- Testing

Our **anomaly detection** algorithm, ADAPT, uses baseline data from earlier test runs. It **processes all metrics** captured during a performance test, including the load test tool and SUT monitoring. ADAPT **flags anomalies** compared to earlier test runs or builds.

Automated Deviation Analysis for Performance Testing **BETA**

Regression Improvements No significant differences No comparison possible Filtered All

ADAPT fails test runs when it detects regression

RED metrics

> Duration Golden signals Regression detected for 12 metrics

USE metrics

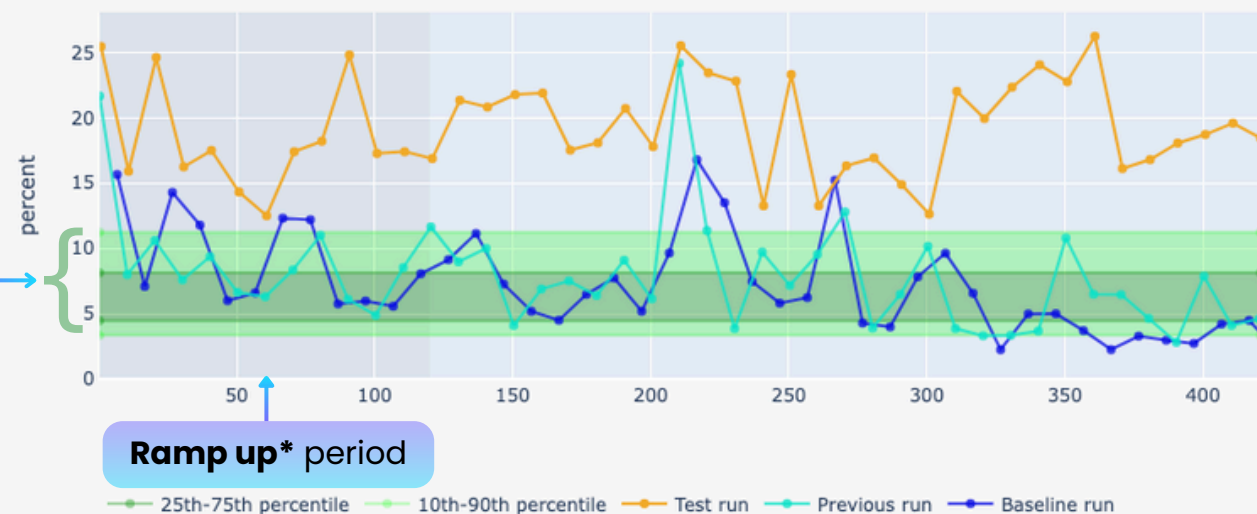
Usage Regression detected for 2 metrics

ML- calculated score

Dashboard	Panel	Metric	Score
JFR Exporter	CPU	max_jvmUser	100
Kubernetes Namespace Resources	Pods CPU usage as percentage of limit	afterburner-db-mysql-0	85

ADAPT has determined that the test run has regressed from the accepted range

max_jvmUser



Accepted range learned from previous tests

Details

Metric	Test	Observed difference
Median	19.59 %	+ 13.78 (237%)
Mean	19.49 %	+ 12.70 (187%)
Minimum	12.63 %	+ 11.33 (877%)
Maximum	26.23 %	+ 1.98 (8%)

Thresholds

Threshold	Threshold value	Observed difference	Result
Percent	15%	+ 237%	❌
Interquartile Range	2	+ 3.76	❌
Absolute	Not set	+ 13.78	-

Conclusion

2 / 2 checks found a difference: ['pct', 'iqr']

*Excluded from analysis



Perfana ADAPT

- Automated
- Deviation
- Analysis for
- Performance
- Testing

- ADAPT **detects regressions** without a threshold being manually configured. ADAPT learns from previous test runs and sets **automatic thresholds!**
- ADAPT **reduces false positives** by dealing with variability.
- ADAPT analyzes **hundreds of metrics** in seconds and presents the detected regressions. Now you can use all your metrics in the analysis and find regressions in anything you measure **automatically.**
- ADAPT saves a lot of manual, and often repetitive, work! Engineers can **focus on solving the problems**, not finding them.

Summary

- CPE is tedious (e.g., dashboards configuration, trending and root cause analysis). **Automation** is the key!
- **The value** of any load testing tool comes in the **analysis**.



Scott Moore: “*Perfana really brings value to whether it’s an open-source load generation tool like Jmeter or a commercial one, adding the value of a deep analysis*”.

- **Perfana** allows **CPE implementation** including shift-left performance testing, empowerment of developers to take ownership, easy onboarding and faster release cycles. This ultimately leads to **a better user experience** and **the risk reduction** of the application you are rolling out in your business.



- For more insights and practical use case, please feel free to reach out to us:

 info@perfana.io

- Start your 30-day free trial:
perfana.io/free-trial

perfana.io