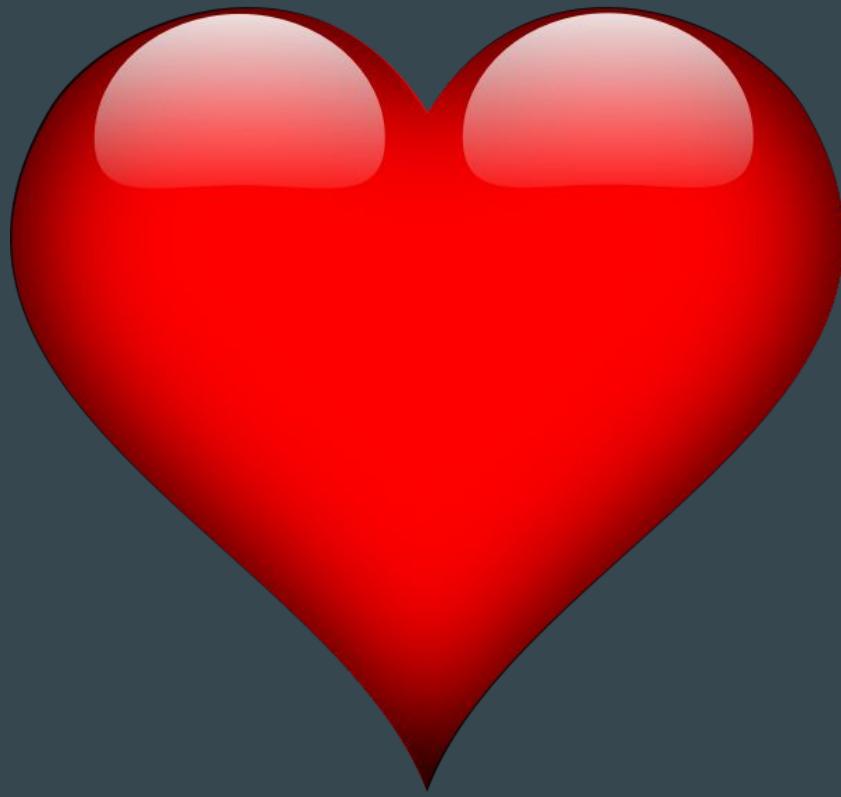




METRICS THAT MATTER

• • •

Empowering Your QA Function With Data



Action

Taribo West

Profile

Injuries & Bans

Contract

Transfer

History

Born 26.3.74 (Age 27). Nigerian (25 caps).

Acceleration	15	Flair	11	Set Pieces	4
Aggression	20	Handling	1	Stamina	19
Agility	9	Heading	15	Strength	20
Anticipation	11	Influence	14	Tackling	16
Balance	18	Jumping	17	Teamwork	15
Bravery	20	Long Shots	11	Technique	7
Creativity	11	Marking	17	Work Rate	18
Crossing	11	Off The Ball	12	Preferred Foot	Right
Decisions	11	Pace	16	Form	-
Determination	19	Passing	11	Morale	Ok
Dribbling	13	Positioning	14	Condition	100%
Finishing	11	Reflexes	3		

◀ ▶ Apps Gls Asts MoM Pass Tck Drb Sh Tar Av R

Non Competitive
League
Cup
Continental
International
Senior Club

Defender (Left/Centre)

IYKYK...



We Need Metrics!

Me

Uncharacteristically wearing a
shirt & windswept tie.

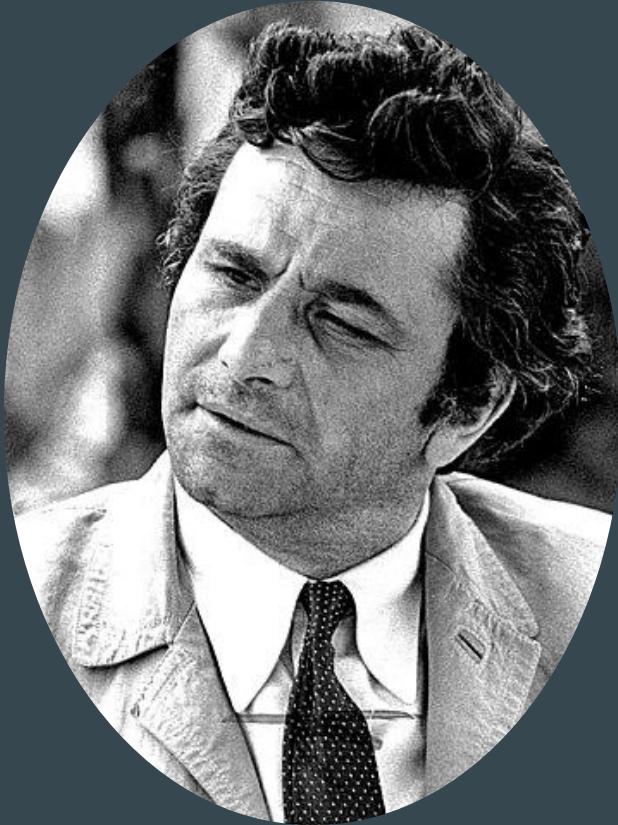
(Must have been attending
a funeral that day.)



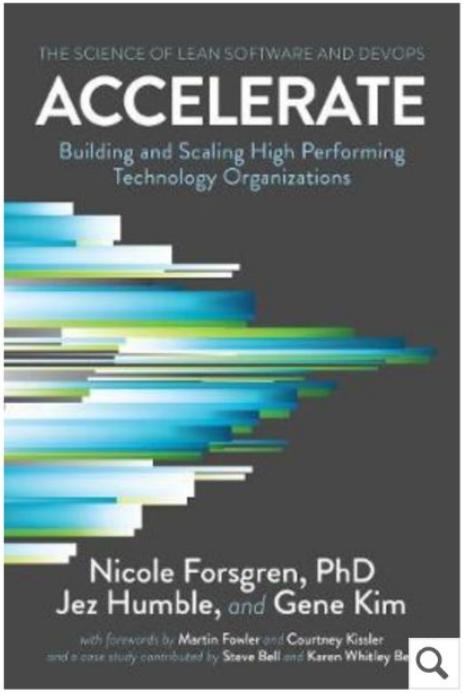


VS





Do We Actually Need Test Metrics?



- Greater Profit/Turnover
- Increased Market Share
- Improved Team Morale and Satisfaction

Available to buy in all good bookstores.

+ Some rubbish ones as well!

The Four Key Metrics

Accelerate by Nicole Forsgren, PhD, Jez Humble, and Gene Kim



1

LEAD TIME

Lead time is the time it takes to go from a customer making a request to the request being satisfied. Shorter lead times enable faster feedback.

DEPLOYMENT FREQUENCY

Deployment frequency is a proxy metric for batch size; the more frequently you deploy the smaller the size of the batch. Small batch sizes reduce cycle times, reduce risk and overhead, improve efficiency, increase motivation and urgency, and reduce costs and schedule growth.

2

3

MEAN TIME TO RESTORE

Reliability is traditionally measured as time between failures, but in a modern software organization failure is inevitable. Thus, reliability is measured by how long it takes to restore service when a failure occurs.

CHANGE FAIL PERCENTAGE

This metric looks at the percentage of changes made to production that fail; the same as percent complete and accurate in Lean product delivery.

4

ELITE PERFORMERS

Comparing the elite group against the low performers, we find that elite performers have...



208
TIMES MORE

frequent code deployments

106
TIMES FASTER

lead time from
commit to deploy



2,604
TIMES FASTER

time to recover from incidents

7
TIMES LOWER

change failure rate
(changes are $\frac{1}{7}$ as likely to fail)

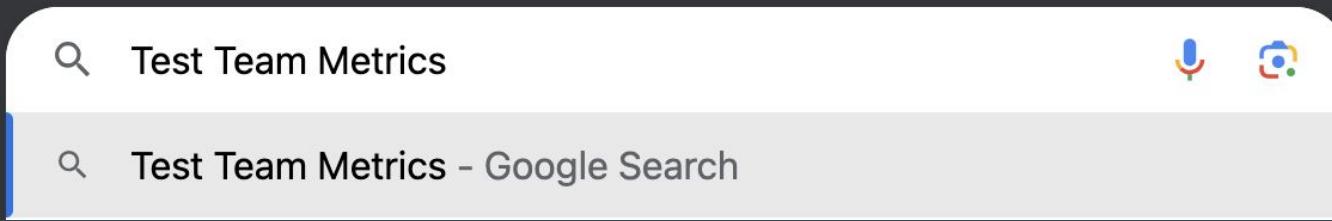


Throughput

Stability



Google



The Days Before ChatGPT...

Common Testing Metrics

- Test cases executed
- % Tests Automated
- Automated Test Coverage
- Number Of Defects Raised (by each team member)
- Test Cycle Time
- Defects Per Requirement
- Average Bugs Per Test
- Defect Resolution Time
- Avg Time To Create Tests
- ...

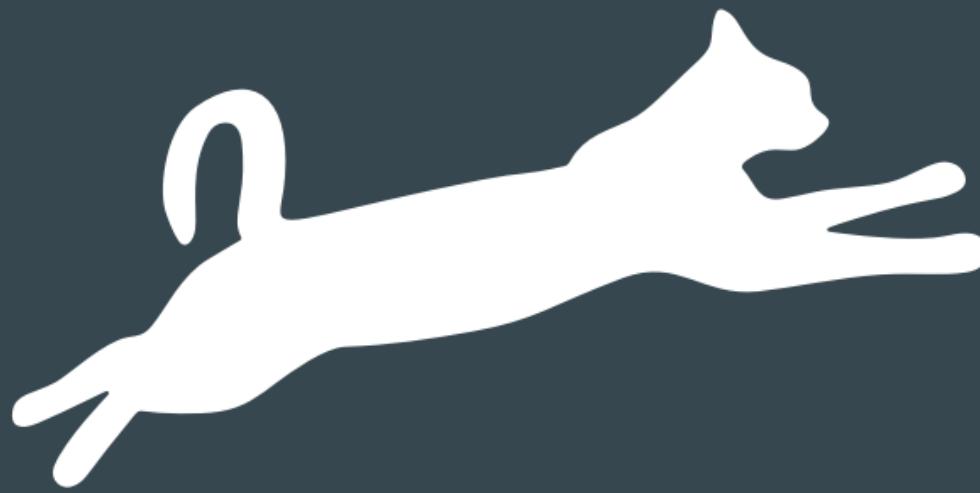


73%

This product is 73% good







T

I'm the principal test engineer of a software testing function. What are some metrics I could use to track my teams progress?



Tracking the progress of a software testing function is crucial for ensuring the quality and efficiency of the testing process. Here are some key metrics you can consider using to monitor your team's progress:



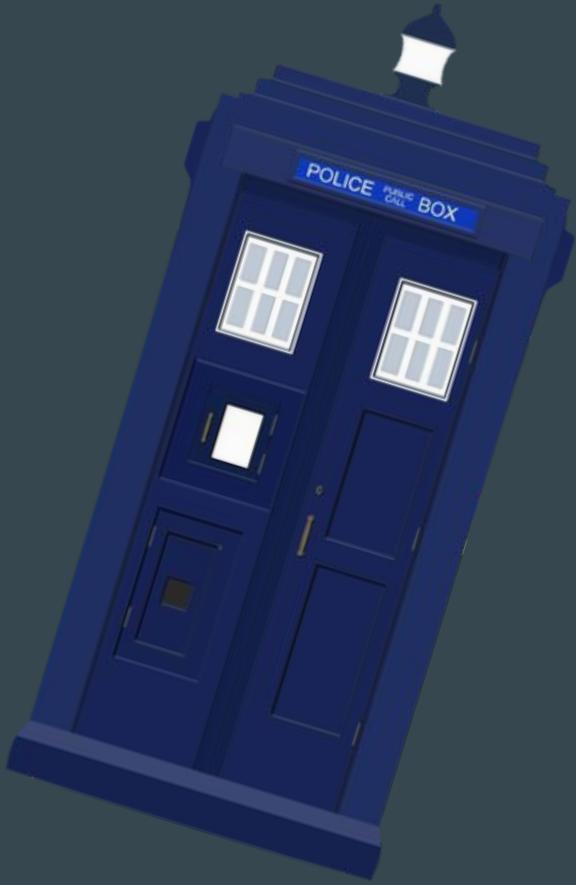
1. **Test Coverage:** This metric measures the percentage of the codebase that has been tested. It helps identify areas that might need more attention in terms of testing.
2. **Defect Detection Rate:** This metric measures the number of defects found per unit of time, such as per week or per sprint. It provides insights into how effective your testing efforts are at identifying issues.
3. **Defect Leakage:** This metric tracks the number of defects that escape the testing phase and are discovered by users or in production. It indicates the effectiveness of your testing in catching critical issues.
4. **Test Execution Time:** Measure the time it takes to run a test suite. This can help identify performance bottlenecks and inefficiencies in your testing process.
5. **Test Case Execution Status:** Track the status of test cases (e.g., passed, failed, blocked). This gives an overview of the health of your test suite and the progress of testing efforts.
6. **Test Case Pass Rate:** Calculate the percentage of test cases that pass successfully. This metric indicates the overall quality of the software under test.
7. **Regression Test Coverage:** Measure the percentage of tests focused on regression testing. This ensures that new changes don't break existing functionality.
8. **Automation Coverage:** Track the percentage of test cases that are automated versus manually executed. Automation can improve efficiency and consistency.

View Document

ChatGPT... is marginally better



This is going to require some more thought...



We're going to go on a bit of a
journey...

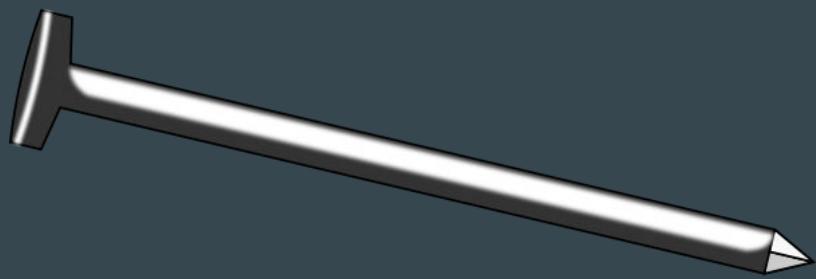
Through time space & reality



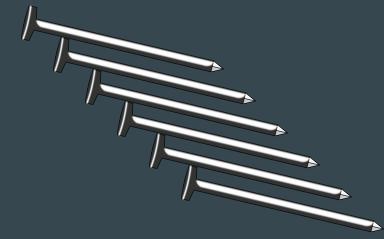
To Soviet Union in the time of Lenin





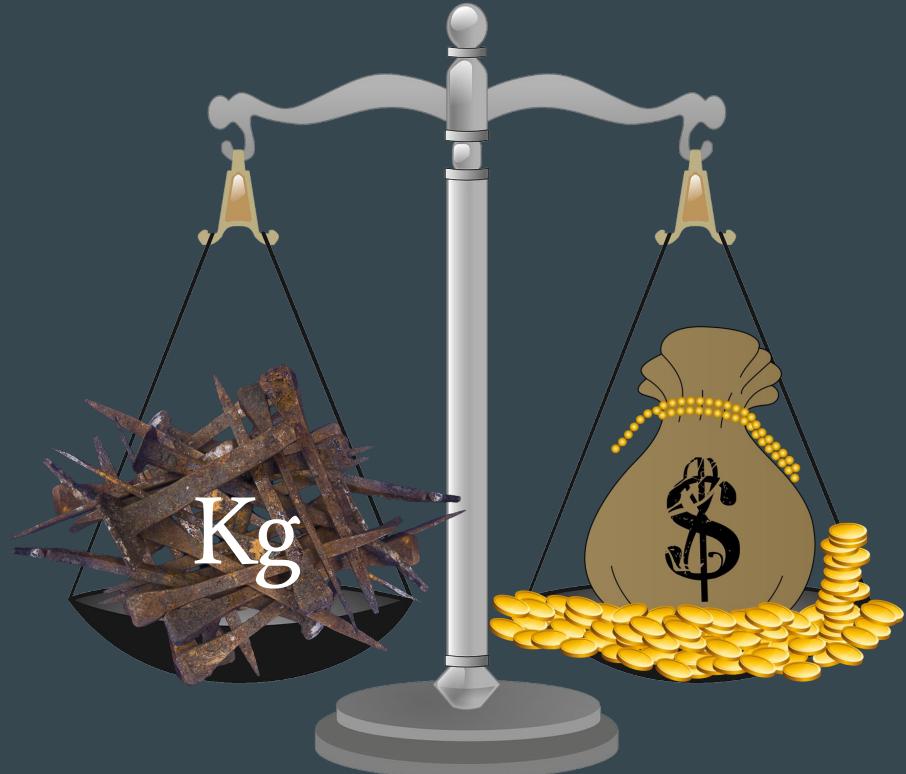


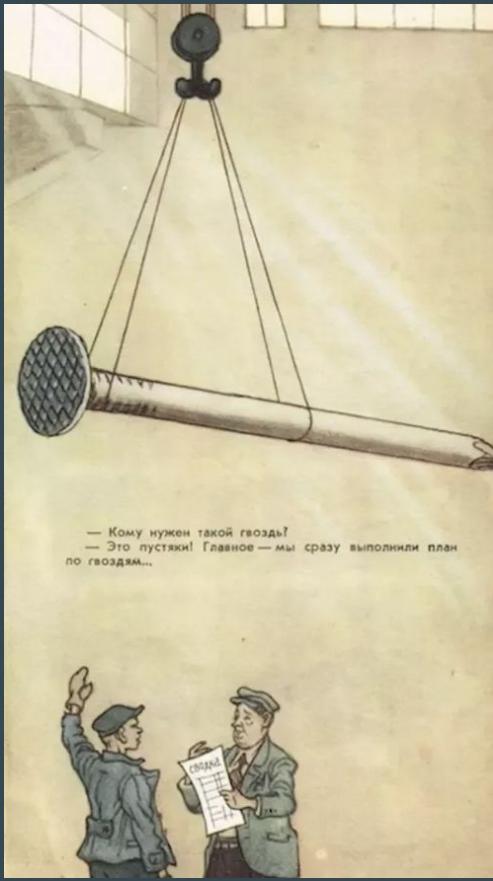
>>











"Who needs such a nail?"

"It doesn't matter, what's important is that we fulfilled the plan for nails."

“When a measure
becomes a target, it ceases
to be a good measure”

GOAL



MEASURE AS TARGET



UNINTENDED ACTIONS



Fewer venomous cobras → Bounty for dead cobras ← People breeding cobras (*Cobra Effect*)



Efficient shoe production → Shoe production numbers ← Produce only left footed shoes for speed

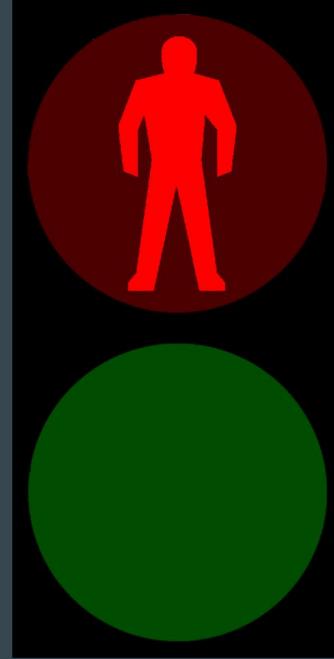


Increase widget sales → Number of sales calls ← Make shorter calls with no sales

“When a measure becomes a target, it ceases to be a good measure.” - Marilyn Strathern

“The more any quantitative social indicator is used for social decision-making, the more subject it will be to corruption pressures and the more apt it will be to distort and corrupt the social processes it is intended to monitor.”





Reduce patients overall wait time... seems noble enough



#Patient Hold Time



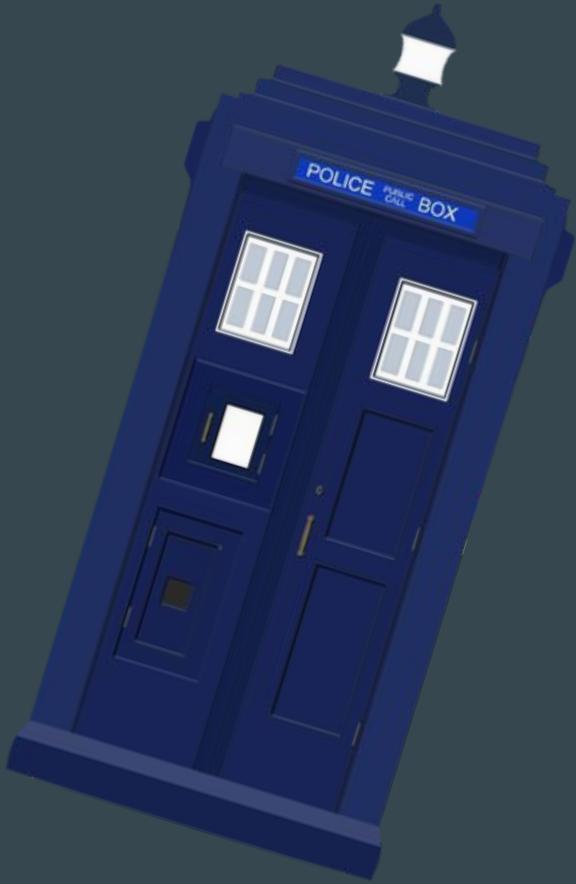
0%

Hold Time

Huge Bonuses for the implementing manager

Huge Wait Time





Hope someone remembers where we parked the TARDIS..

We're headed to WW2 USA in 1943

Specifically the “Center for Naval Analyses”

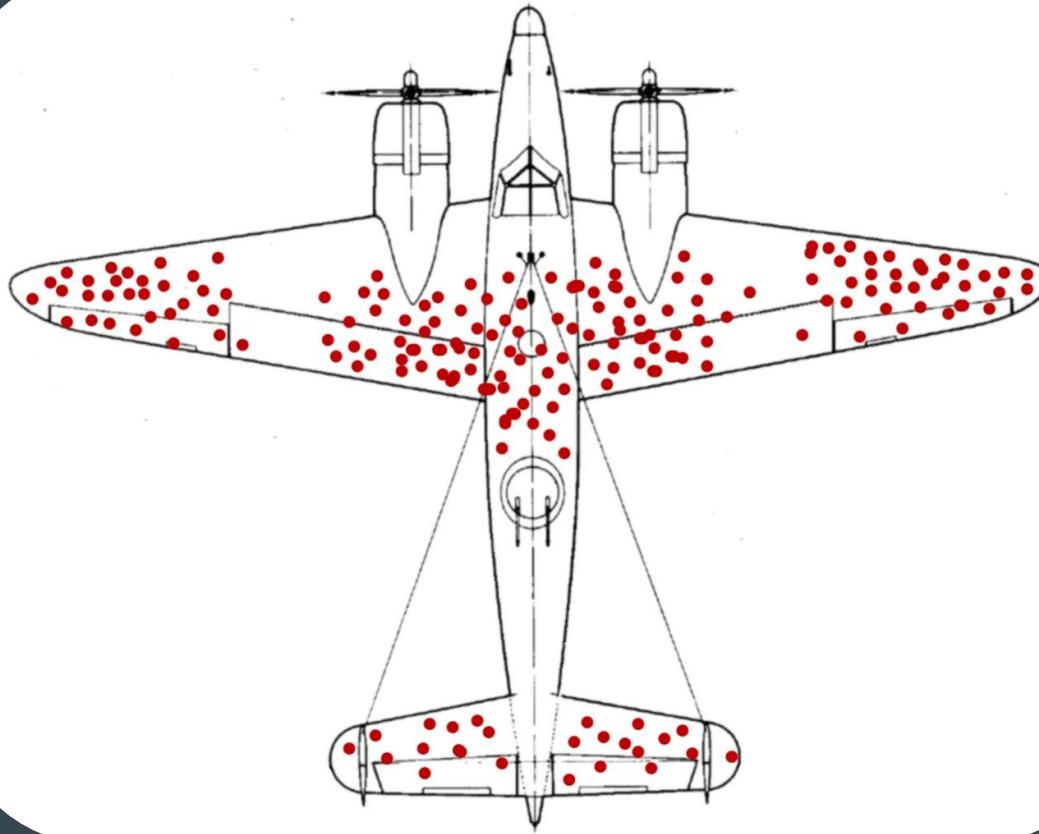


Image depicting “damage done to returned aircraft after missions”

Abraham Wald

Born Hungary 1902
Died December 1950



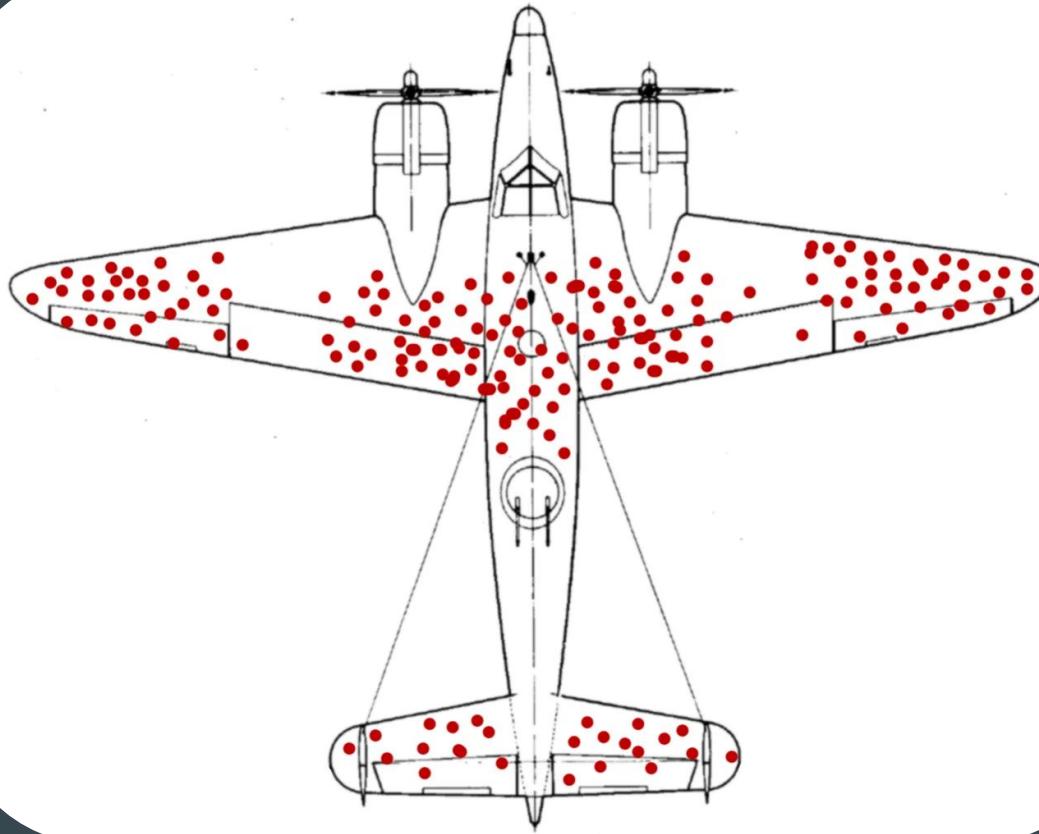


Image depicting “damage done to returned aircraft after missions”

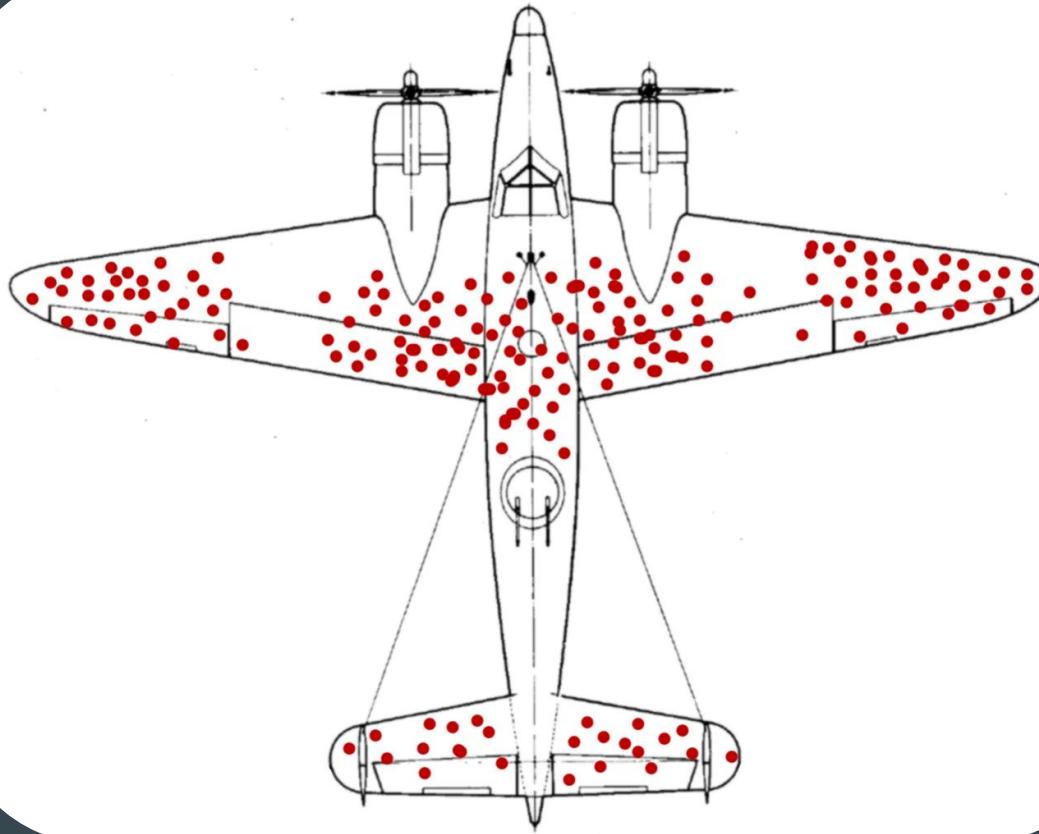


Image depicting “damage done to returned aircraft after missions”

Survivorship bias or survival bias is the logical error of concentrating on entities that passed a selection process while overlooking those that did not.

Common Testing Metrics

- Test cases executed
- % Tests Automated
- Automated Test Coverage
- Number Of Defects Raised (by each team member)
- Test Cycle Time
- Defects Per Requirement
- Average Bugs Per Test
- Defect Resolution Time
- Avg Time To Create Tests
- ...







Just another busy day in the office for me...



Remember.. This was pre-ChatGPT

To enable teams to constantly deliver working software that meets customer requirements by providing fast feedback and seeking prevention of defects rather than defect detection.

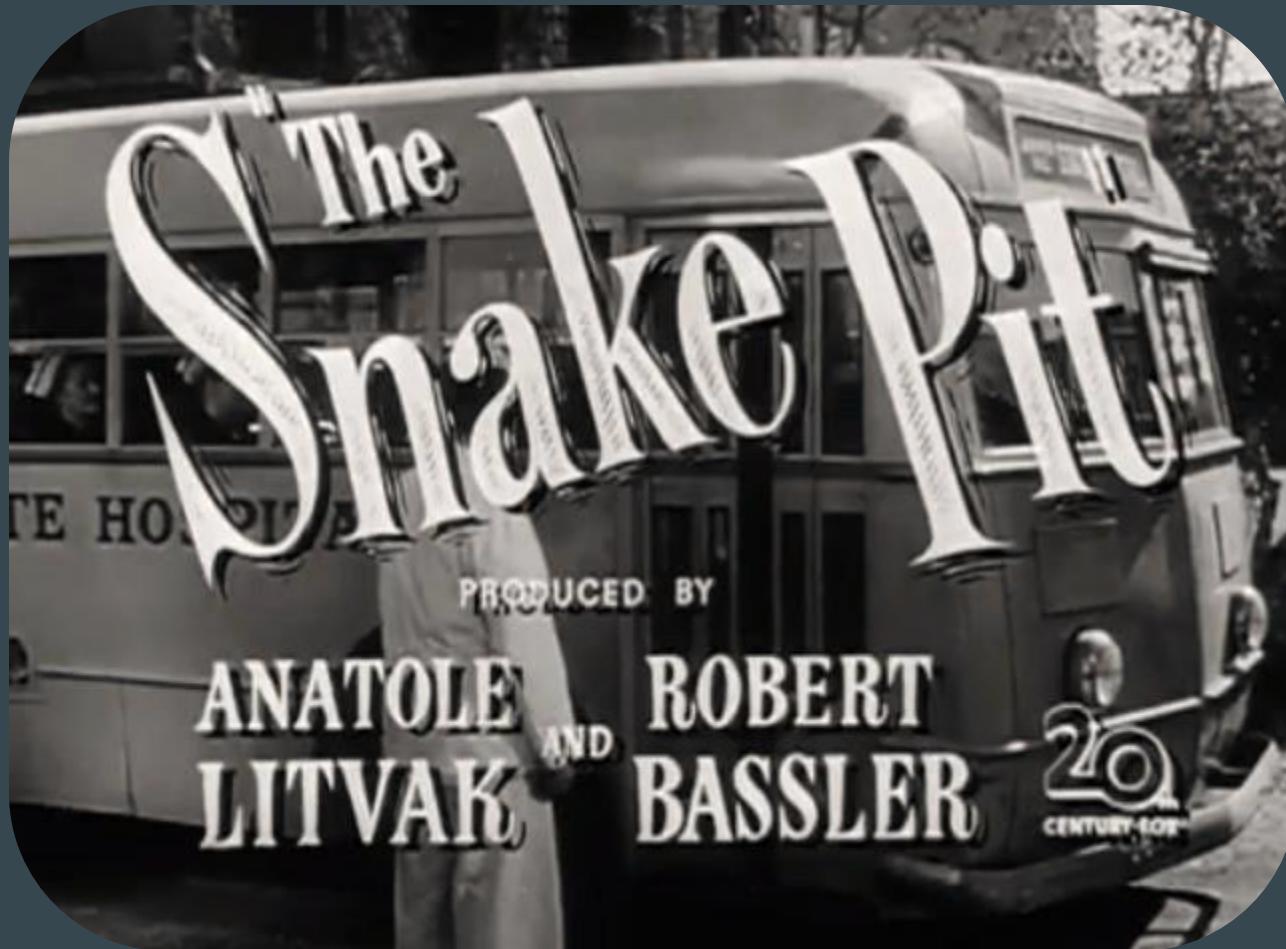
To enable teams to constantly deliver working software that meets customer requirements by providing fast feedback and seeking prevention of defects rather than defect detection.

To enable teams to constantly deliver working software that meets customer requirements by providing fast feedback and seeking prevention of defects rather than defect detection.

To enable teams to constantly deliver working software that meets customer requirements by providing fast feedback and seeking prevention of defects rather than defect detection.

Metric Type	Supporting Metrics
Goal	<ul style="list-style-type: none"> # Releases deployed Mean time between incidents 7 day rolling bug count System uptime % of fixes vs new functionality (git log) NFR fails on prod
Features of the function	<ul style="list-style-type: none"> Mean time to resolution (between failed builds) Mean time to resolution (fixing prod issue) Build Times (By build type) + monthly trend Mean Job time(which jobs in build are longest running) Deployment frequency Re-run passes (flaky test indicator, when a failing build is re-triggered and passing without code change) Story run time RFT errors %
Informative	<ul style="list-style-type: none"> % Broken builds Bugs by component Setup:Assert ratio?







Diversify Metrics to avoid context loss



1. Are we moving in the right direction?
2. Are these still the right metrics?

Review Regularly

	2.160	27,000	2,180	2,170	5,690	4,500	5,420	0.000	0.000	0.000	0.000
5.340		1,225	5,350	5,690	584,484	0.450	0.450	0.000	0.000	0.000	0.000
0.450			0	0.000	0.410	92,484	2,600	0.000	0.000	0.000	0.000
2.600		30,393	2,440	2,750	56,512	1,600	0.000	0.000	0.000	0.000	0.000
1.600		5,000	1,600	1,830	128,544	2,290	0.000	0.000	0.000	0.000	0.000
73,778		2,300	2,310	3,100	874,820	3,090	0.000	0.000	0.000	0.000	0.000
0.951		0	0.000	0	0.000	0	1,450	0.000	0.000	0.000	0.000

Visualise For Everyone!



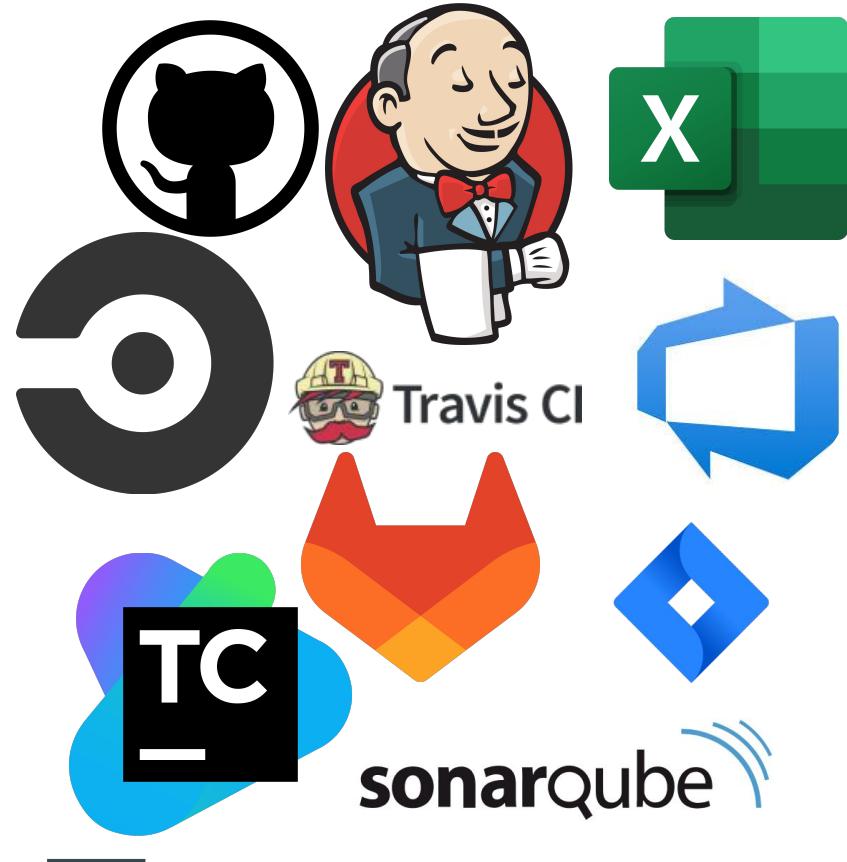
Focus on the trends, rather than the absolute values

You've selected your
metrics: now what?

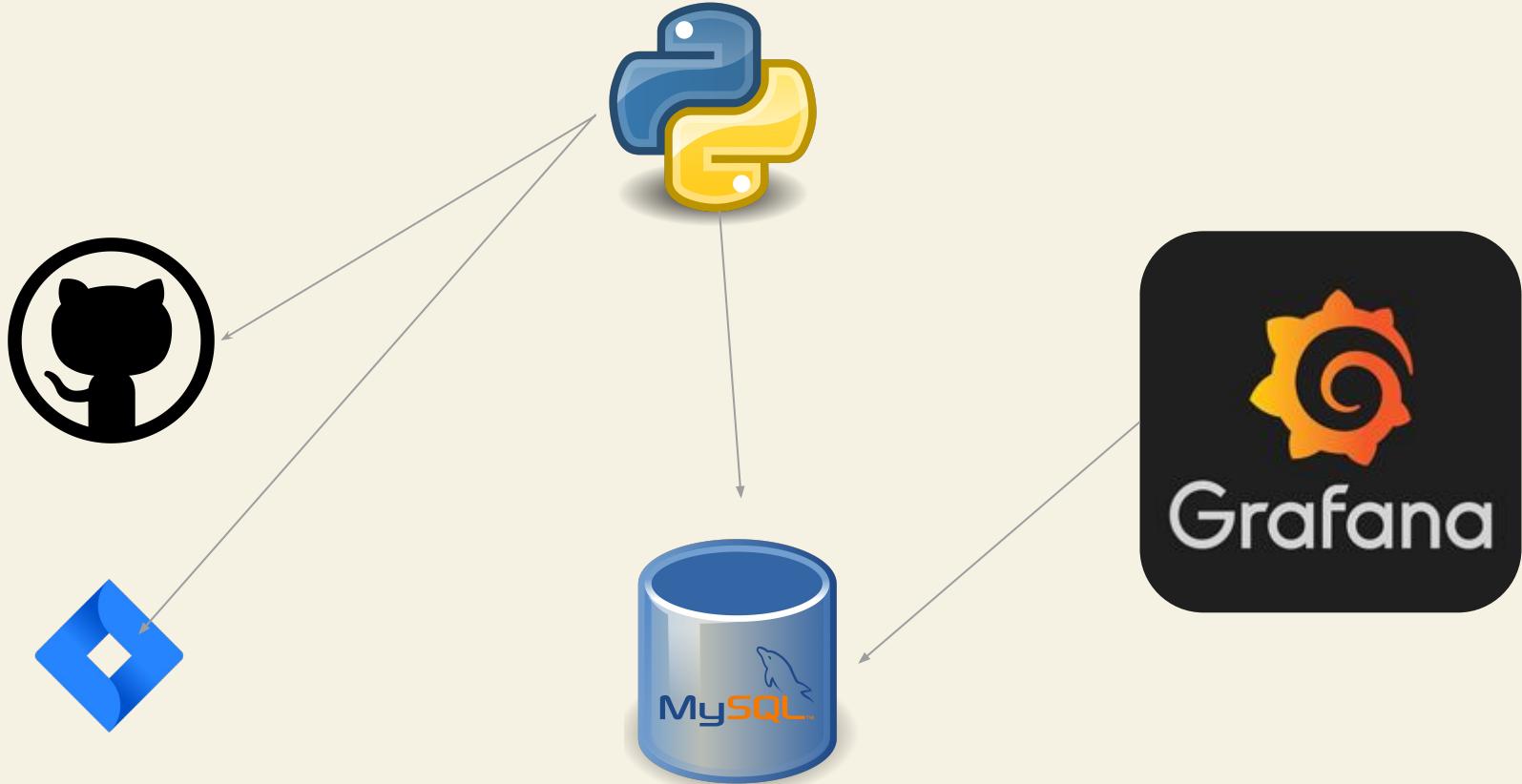


MADE IN CHINA
100% COTTON
ONE SIZE FITS ALL
HAND WASH ONLY

- % Broken builds
- Bugs by component
- Setup:Assert ratio?
- Mean time to resolution (between failed builds)
- Mean time to resolution (fixing prod issue)
- Build Times (By build type) + monthly trend
- Mean Job time (which jobs in build are longest running)
- Deployment frequency
- Re-run passes (flaky test indicator, when a failing build is re-triggered and passing without code change)
- Story run time
- RFT errors %
- Mean time between incidents
- 7 day rolling bug count
- System uptime
- % of fixes vs new functionality (git log)
- NFR fails on prod







Sample Architecture

Metrics that Matter: Empowering Your QA Function for Success



1. Measure what matters, not just what's easy
2. Align metrics with goals and features of your function
3. Diversify & Assess your metrics regularly to ensure you're avoiding Goodhart's law & other pitfalls
4. Visualise your metrics to integrate them into your culture

Thank you for listening!

Sample Repo: <https://github.com/TomMcC89/GrafanaDemo>