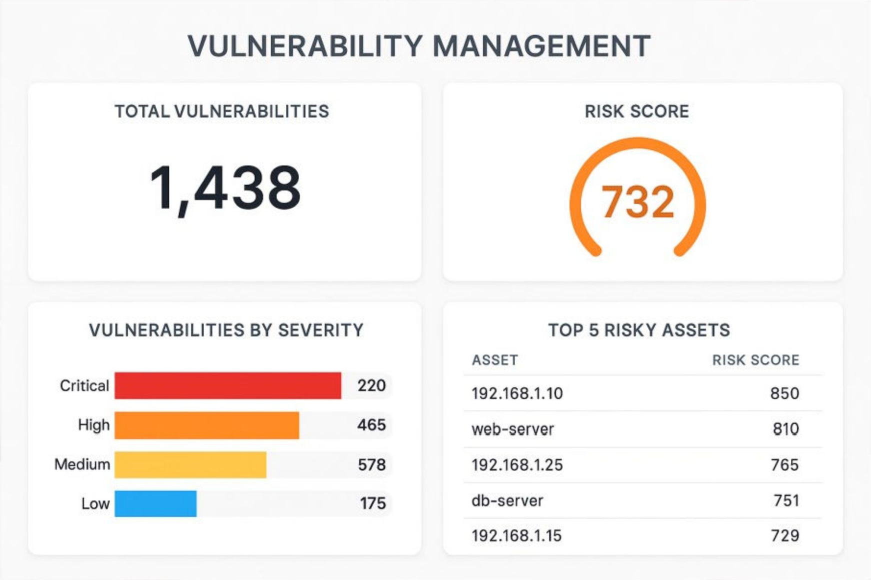
WE BELIEVE
IN A SIMPLE AND
SAFE DIGITAL
FUTURE

CODIFIC

Your security dashboard is lying to you: The science of metrics

Aram Hovsepyan

Vulnerability dashboard: are these good metrics?



VULNERABILITY MANAGEMENT

TOTAL VULNERABILITIES

1,438

RISK SCORE

732

VULNERABILITIES BY SEVERITY

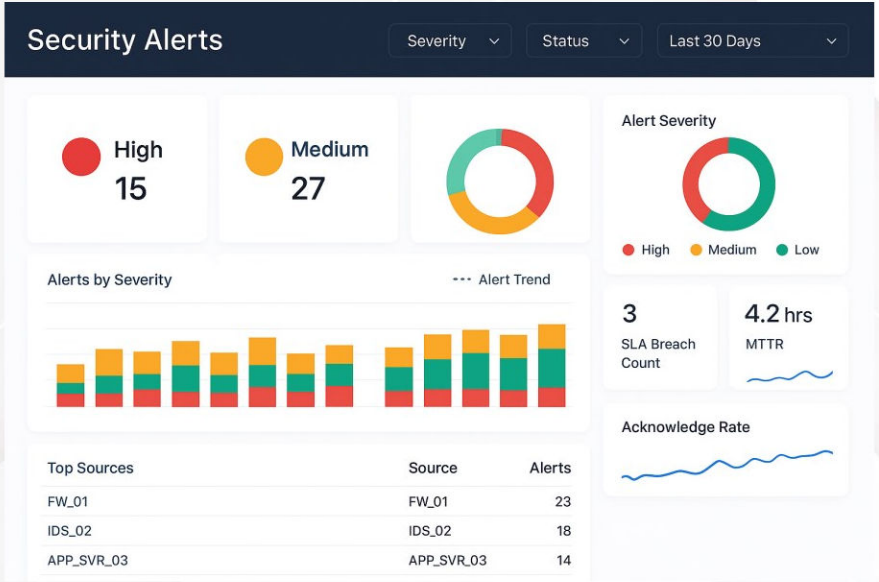
Critical	220
High	465
Medium	578
Low	175

TOP 5 RISKY ASSETS

ASSET	RISK SCORE
192.168.1.10	850
web-server	810
192.168.1.25	765
db-server	751
192.168.1.15	729

- Total vulnerabilities
- Risk score

SIEM dashboard: is this a good metric?



High 15

Medium 27

3 SLA Breach Count

4.2 hrs MTTR

Top Sources	Source	Alerts
FW_01	FW_01	23
IDS_02	IDS_02	18
APP_SVR_03	APP_SVR_03	14

- Number of security alerts

Outline: how to fail spectacularly with metrics

- Measure and report useless things
- Design bad metrics
- Make sure your metrics math makes no sense
- Dress up bad data as good decisions

Aram Hovsepyan

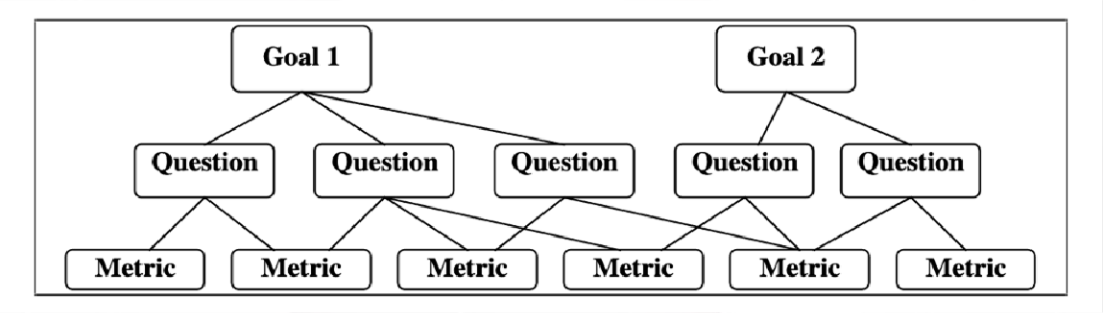
- PhD in AppSec
- CEO @ Codific
- OWASP SAMM Core team member

https://www.linkedin.com/in/aramhovsep
https://appsecscience.com

Metrics purpose: Are these good metrics?

- M1: Number of exploitable vulnerabilities in production
- M2: Impact score of each exploitable vulnerability
- M3: Likelihood score of each exploitable vulnerability

Goal-Question-Metric framework*



Conceptual level

Operational level

Quantitative level

- G: Improve the time to fix for high-risk vulnerabilities in production
- Q1: How many high-risk vulnerabilities are in production?
- Q2: What is the current mean time to fix?
- Q3: Is the time to fix improving?
- M1: Number of exploitable vulnerabilities in production
- M2: Impact score of each vulnerability
- M3: Likelihood score of each vulnerability
- M4: Time to deploy of each vulnerability fix

* The Goal Question Metric Approach, V. Basili et al

Goal-Question-Metric exception

- Qualitative analysis
  - Explore and understand complex phenomena
  - Generate and refine goals

Metrics design: Which metric is better?

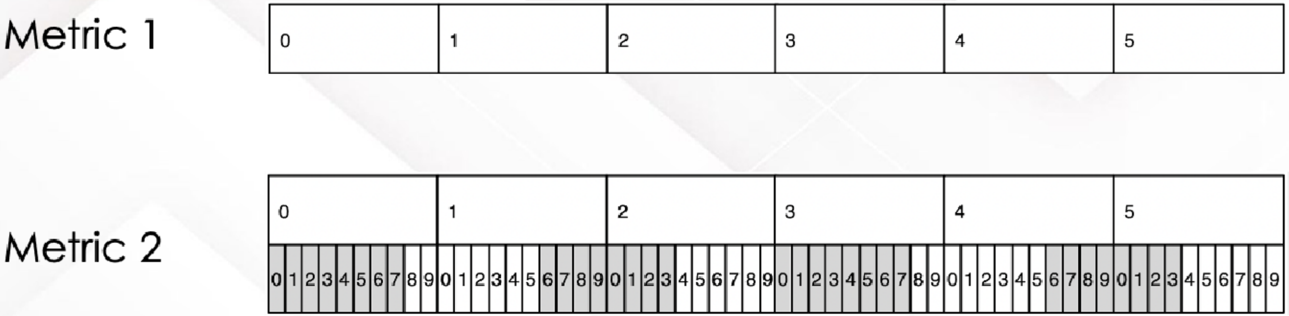
- G: Improve security awareness of developers
- Q: What is the current security awareness of developers
- M: Secure code training test results
  - M1: Pass / fail
  - M2: 0 to 10 score
  - M3: 0 to 100000 score

Metrics design: Which metric is better?

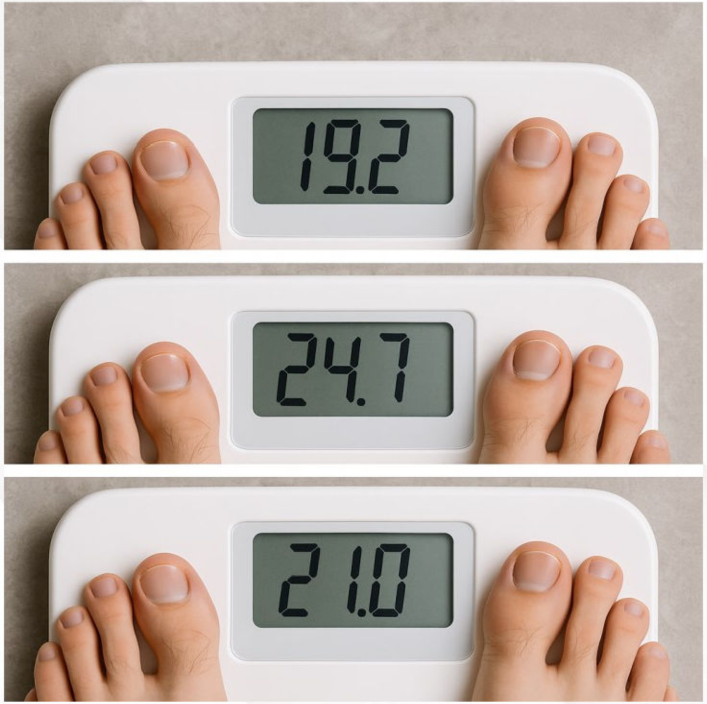
- G: Reduce the risk of getting breached
- Q: What is the likelihood of getting exploited due to a known vulnerability in production
- M: Vulnerability exploitation likelihood
    - M1: Yes/No based on KEV (list of exploits in the wild)
    - M2: 0.0 to 1 based on EPSS (prediction model)

Metric precision: smallest unit of measurement

- Secure code training test results
  - M1: Pass or fail
  - M2: Score from 0 to 10



Metric reliability: consistency of measurements



Metric reliability: consistency of measurements

- Vulnerability exploitation likelihood
  - M1: Known Exploited Vulnerabilities (KEV) score
  - M2: Exploit Prediction Scoring System (EPSS) score

Precision vs reliability

- Secure coding test results
  - M1: Pass / fail
  - M2: 0 to 10 score
  - M3: 0 to 100 score
- Five developers with similar levels of security expertise
  - M1: everyone passes
  - M2: 2 devs score 4 out of 10 | 3 devs score 9 out of 10
  - M3: everyone scores 30 out of 100
- Which is the best metric?

Metric validity

- Can **Number of Lines of Code** serve as a good metric to assess security vulnerabilities?

Content validity

- **How much of the outcome does the metric cover**
- Security awareness of employees
    - M1: Number of hours of training completed
    - M2: Number of top security risks seen during training

Criterion validity

- **How well the metric correlates with the outcome**
- Security awareness of employees
  - M1: Percentage of top security risks seen during the training
  - M2: Security awareness test score right after the training
  - M3: Security awareness test score a year after the training

Criterion validity revisited

- Five devs with similar levels of **strong** security expertise
    - M1: everyone passes
    - M2: 2 devs score 4 out of 10 | 3 devs score 9 out of 10
    - M3: everyone scores 30 out of 100
- Which is the best metric?

Construct validity

- **How well the metric correlates with the concept**
- How well are we prepared for an actual cyberattack?
  - M1: Security awareness test score a year after the training
  - M2: Attack simulation exercise scores
  - M3: Real attack

Metric math: Measurement scales

- Which of the following statements is true?
  - SQL injection is worse than XSS
  - 1 HIGH vulnerability is better than 10 MEDIUM
  - CVSS 10.0 is twice more severe than 5.0
    - AV x AC x PR x UI x C x I x A => 0..10
  - EPSS 1.0 is twice more likely than 0.5
    - Likelihood of an exploit expressed as a percentage => 0..1

Measurement scales

Scale	Examples	Mode	Median	Average
Nominal	Vulnerability types	✅	❌	❌
Ordinal	Severity levels CVSS scores	✅	✅	❌
Ratio	EPSS scores Risk value in $$$	✅	✅	✅

✅ - meaningful operation

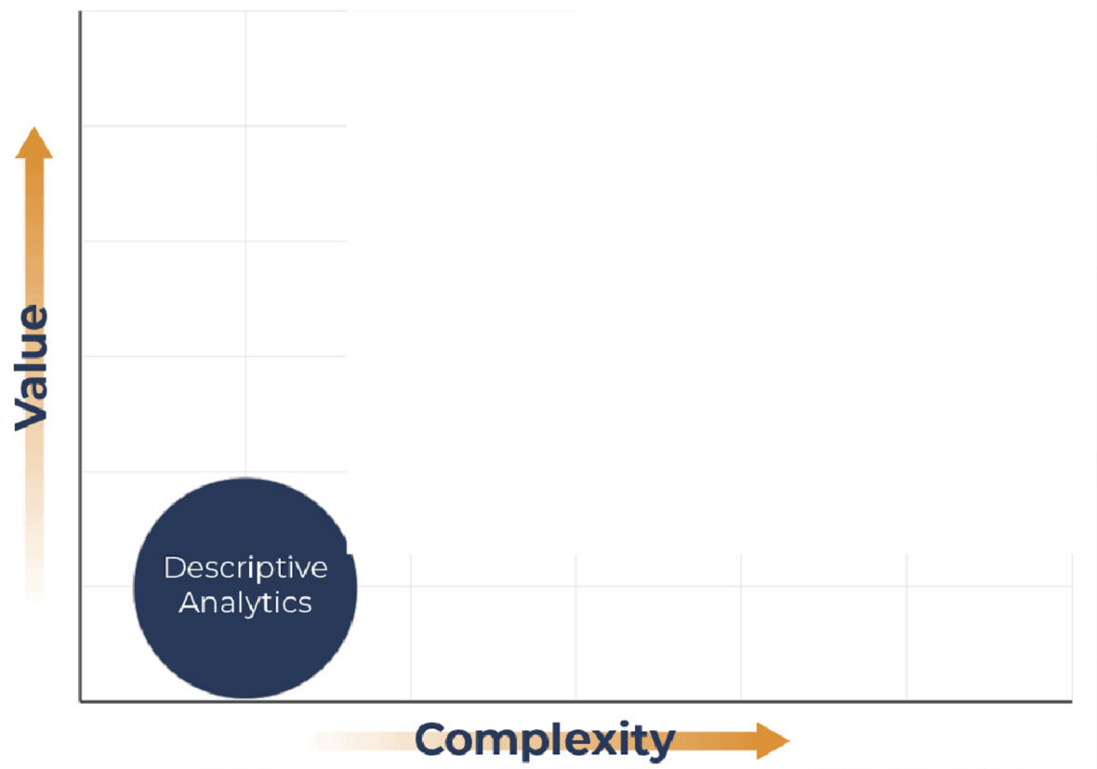
❌ - meaningless operation

ASPM Risk Score

- **Risk = CVSS × (1 + EPSS) × (1 + 0.5 × PROD + 0.5 × CLOUD)**
  - CVSS = impact score [0..10] *
  - EPSS = likelihood score [0..1]
  - PROD = is production [Yes/No]
  - CLOUD = is internet facing [Yes/No]
- **Asset Risk = Average(Risk of top 5% of vulnerabilities)**
- Meaningful?
- Userful?

* https://appsecscience.com/cve_video

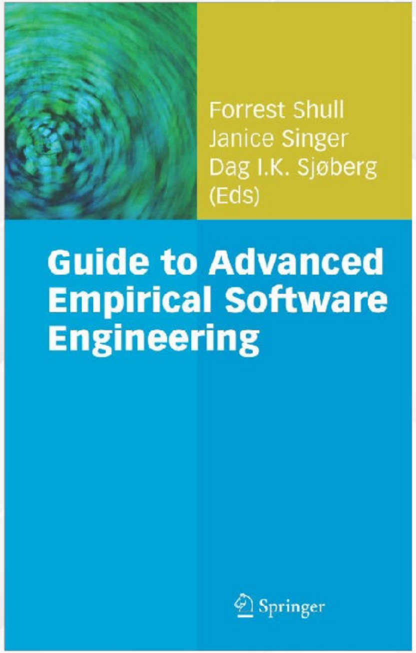
Data analysis techniques



Key takeaways: metrics that don’t lie

- Good metrics answer meaningful, not convenient questions
- Good metrics are precise, reliable, valid … and hard to find
- Bad averages create beautiful lies
- Great dashboards reveal truth, not dress up vanity metrics

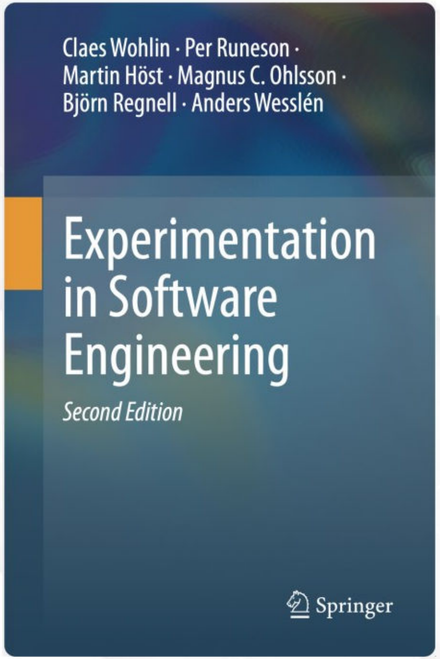
Thank you



Forrest Shull
Janice Singer
Dag I.K. Sjøberg
(Eds)

Guide to Advanced Empirical Software Engineering

Springer



Claes Wohlin · Per Runeson ·
Martin Höst · Magnus C. Ohlsson ·
Björn Regnell · Anders Wesslén

Experimentation in Software Engineering

Second Edition

Springer

https://appsecscience.com