Taking the Measure of Cybersecurity

Bill Nichols Ph.D.

Principal Research Engineer

$\frac{22}{2}$ 22 2022

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

In this presentation I will provide an overview of the meaning and purpose of measurement and how to measurement can apply to cybersecurity. Measurement has a scientific definition and an engineering implementation but is used primarily to make economic decisions. Nonetheless, problems in measurement result from several causes include the following: a poorly defined concepts, ill-defined objectives, lack of context, failure to connect the measures to outcomes, inattention to the quality aspect of the measure. These are especially problematic in cybersecurity because the relationship between measurements and outcomes can change for unexpected reasons. Many of these problems can be addressed using structured frameworks that recognize these sometimes-competing aspects of measurement. Some examples of measurements derived using disciplined frameworks will demonstrate how science supports engineering and both support economic decisions. The emphasis will be on metrics for making economic decisions. This presentation will conclude by noting that some emerging trends in software engineering such as increased reliance upon automated tools, use of “big data”, cloud computing, and end-to-end digital engineering models will profoundly influence future measurement. Each of these brings new challenges, but also promises more rigorous definition and documentation.

Cybersecurity State of the Art, a Personal View

Why is Cybersecurity so Hard?

Instead, we play Whac-A-Mole, reacting to each incident and hope for the best.

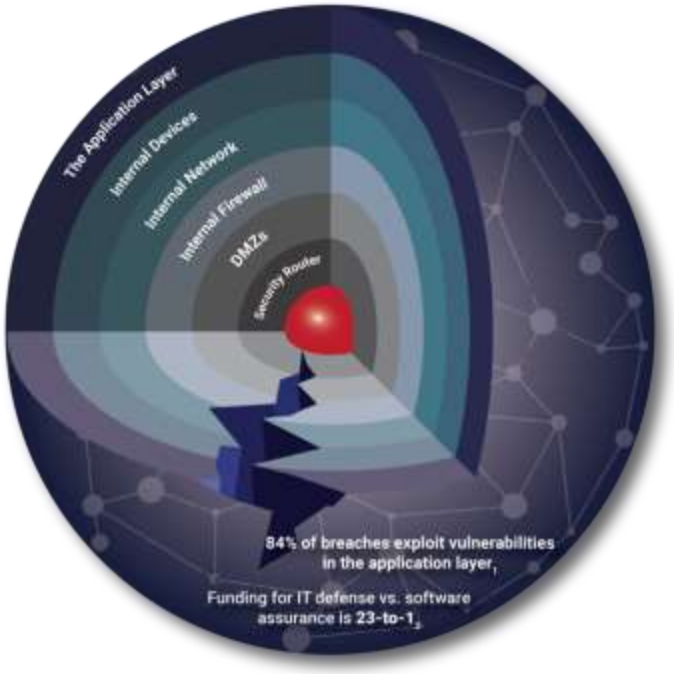
One reason is

Cybersecurity is a **system** property

Cybersecurity does not reside in any of the components but is influenced by the properties and interactions of many parts.

Missing is the ability to perform formal analysis of a system's numerous parameters and how they contribute to the properties such as Confidentiality, Integrity, and Availability

Effective Security Requires a Holistic Approach



The Application Layer is the new perimeter exploited by 84% of breaches

Security must be Engineered into the Lifecycle of Applications changing the way we build and buy technology

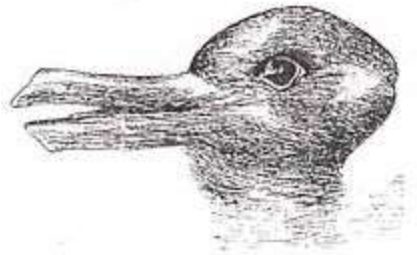
- "76 percent of U.S. developers use no secure application program process"[4]
- "More than 40 percent of software developers globally say that security isn't a top priority for them"[4]
- 2017 less than 5% of DevOps initiatives have achieved the level of security automation required to be considered fully DevSecOps.[3]

I've given you a few metrics here! But what do they mean?

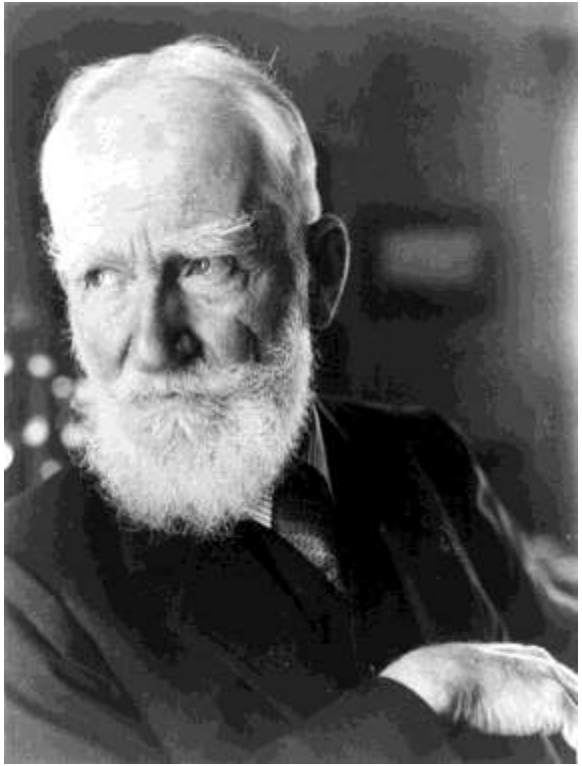
1. Clark, Tim, *Most cyber Attacks Occur from this Common Vulnerability*, Forbes. 03-10-2015
2. Feiman, Joseph, *Maverick Research: Stop Protecting Your Apps; It's Time for Apps to Protect Themselves*, Gartner. 09-25-2014. G00269825
3. *Horvath, Mark, Neil MacDonald, Ayal Tirosh: Integrating Security Into the DevSecOps Toolchain, Gartner. 11-16-2017. G00334254*
4. Microsoft[1]– http://visualstudiomagazine.com/articles/2013/07/16/majority-of-us-devs-dont-practice-secure-coding.aspx

George Bernard Shaw on the **Illusion** of Communication

The single biggest problem in communication is the illusion that it has happened



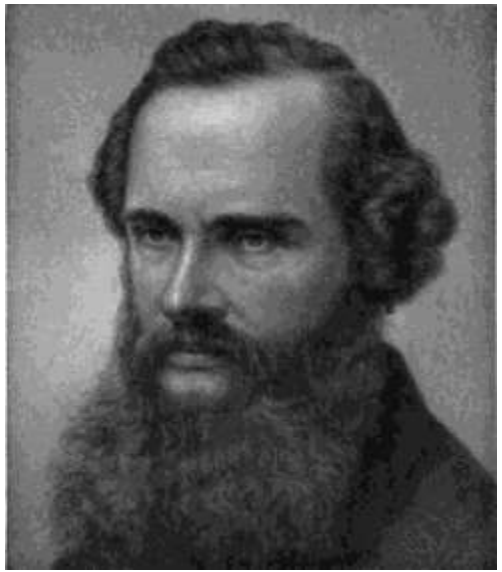
Is this a picture of a Duck or a rabbit?



Lord Kelvin, the Essential Understanding from Measurement

. . when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind: it may be the beginning of knowledge, but you have scarcely, in your thoughts advanced to the stage of science. . .

William Thompson, Lord Kelvin



Measurement helps reduce vagueness and ambiguity

James Clerk Maxwell, the Problem of Communicating

It has been felt that experimental investigations were carried on at a disadvantage in Cambridge because the apparatus had to be constructed in London. The experimenter had only occasional opportunities of seeing the instrument maker, and was perhaps not fully acquainted with the resources of the workshop, so that his instructions were imperfectly understood by the workman . . .”

What do the measures mean?

How were they taken?
In what environment?
With what equipment?

Threats to Measurement

Poorly defined concepts

Ill-defined objectives, what is the purpose of the measurement?

Lack of context or environment

Failure to connect the measures to outcomes with a robust model

Imprecision over precisely what measurements inform a metric

Inattention to the quality aspect of the measure (compare, evaluate?)

Measurement affecting the system behavior

What are metrics and why do we need them?

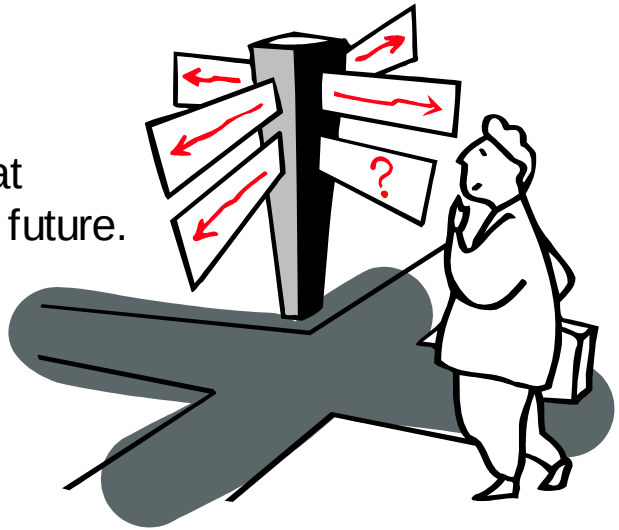
Metrics include information of system properties or performance.

Metrics inform decisions.

Metrics can be used to understand what happened or what might happen in the future.

Metrics help to determine

- If the process is stable
- If the process is capable
- If goals are being met
- How alternative processes, tools, or products compare, and
- How to manage change



Measurement Principles

Measurement monitors current state and enables improvement but measuring your product or process will not sustain or improve it. You must make changes to achieve lasting effects.

To be useful, measurements must be:

- Accessible/available
- Related to business goals
- Provide value greater than the cost of measurement

Measurements should be:

- Gathered for a specific purpose
- Explicitly defined
- Properly managed
- Properly used to make decisions

Outline

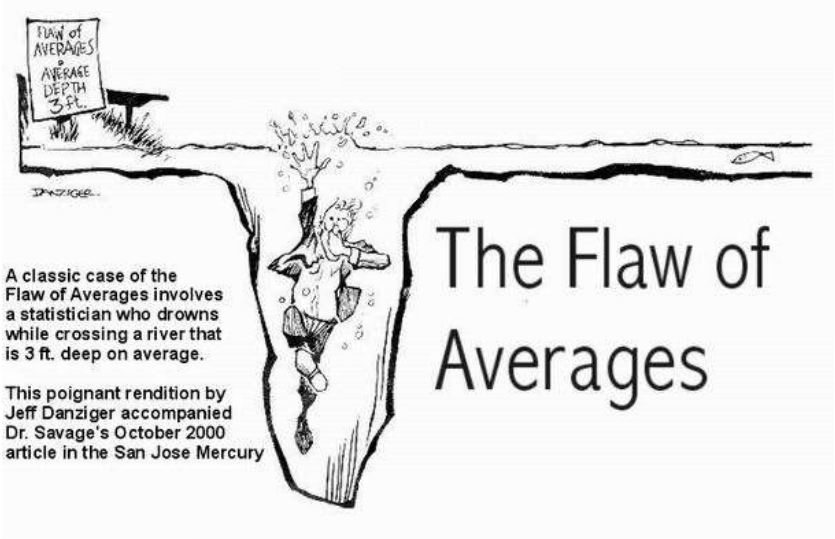
The Theory of Measurement

Take an example drawn from Cem Kaner's "Software Engineering Metrics: What do They Measure and How Do We Know?"

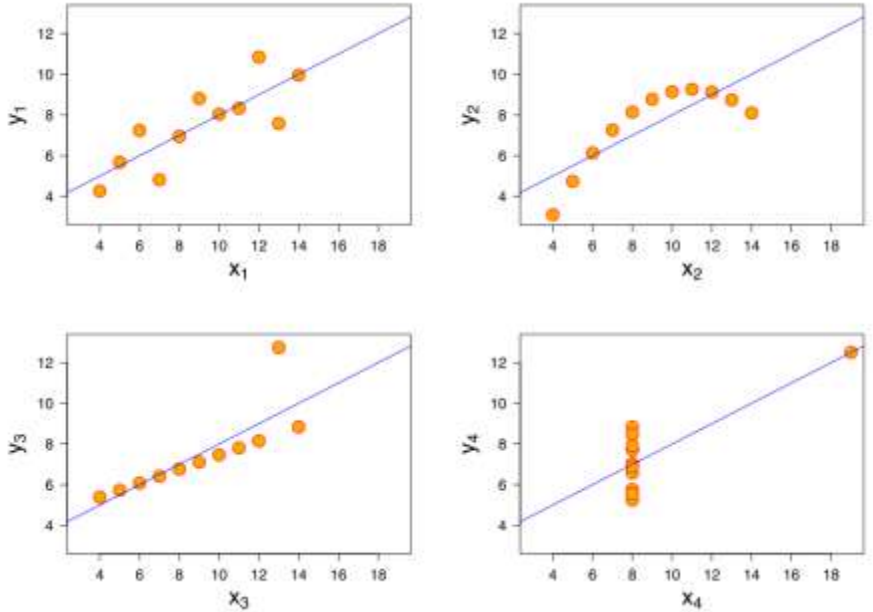
Kaner, Dr. Cem (2004), *Software Engineer Metrics: What do they measure and how do we know?*, CiteSeerX 10.1.1.1.2542

Mean Time to Incident

- How is the time measured?
- Why the mean?
- Are there sub-populations of user profiles?
- Under what conditions?
- Are they independent?



Beware the “Flaw of Averages”

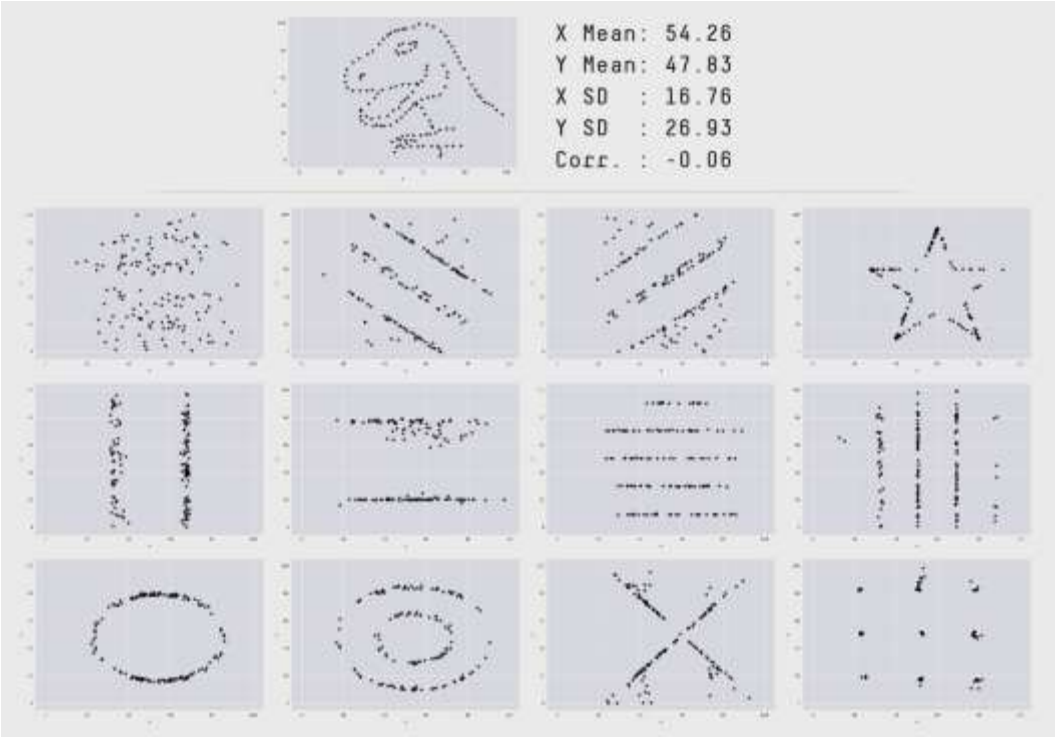


Each of the above distributions have both the same mean and same variance.

Use the full distribution of data to capture not only context, but also information not included in summary statistics.

Datasaurus example

https://www.autodeskresearch.com/publications/samestats



Time Between Incidents, How is the Time Measured?

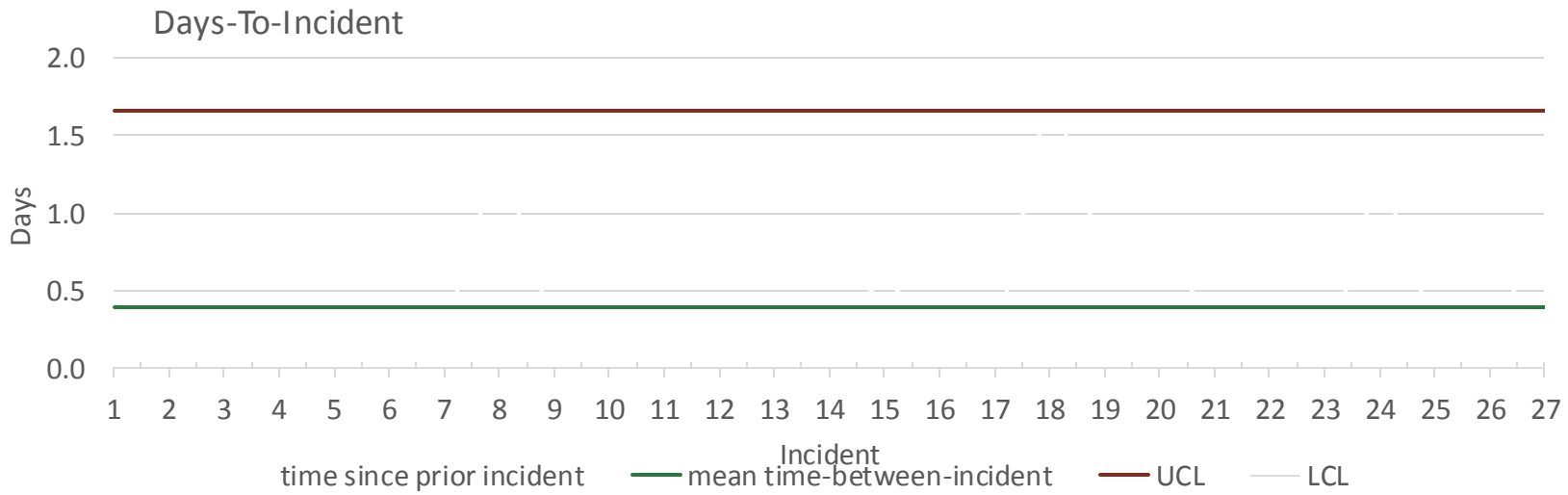
Wall clock/calendar time?

CPU time?

User Time?

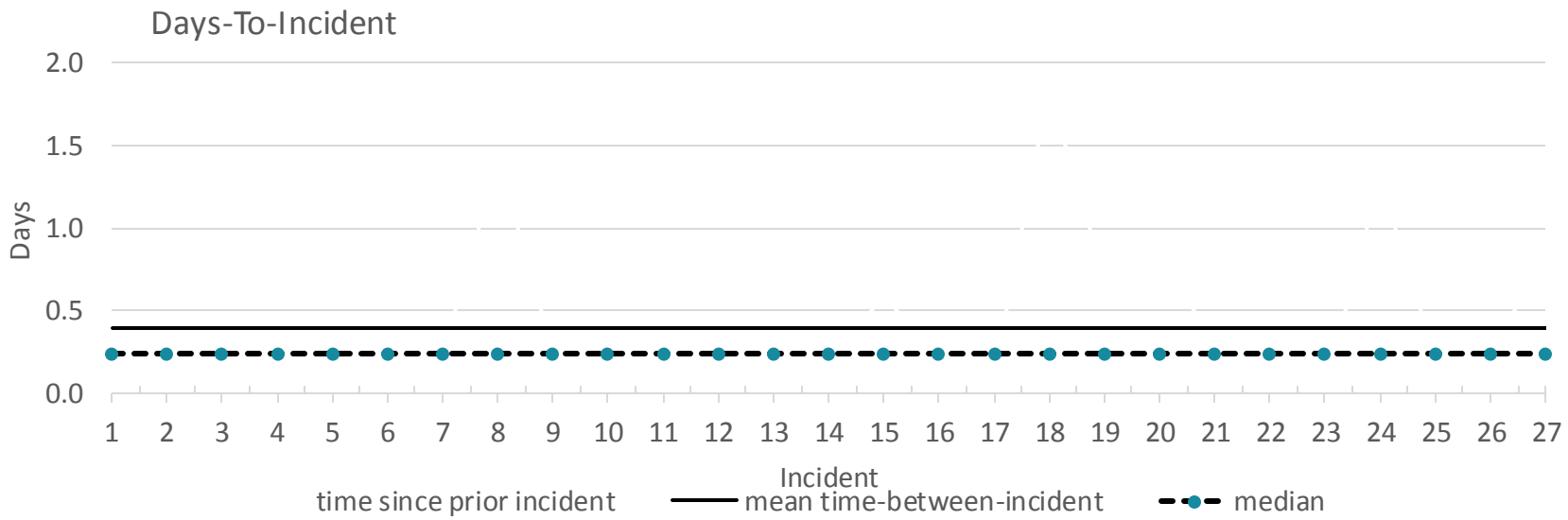
Do we count incidents or measure time between them?

How does the choice affect the interpretation?



Why the mean?

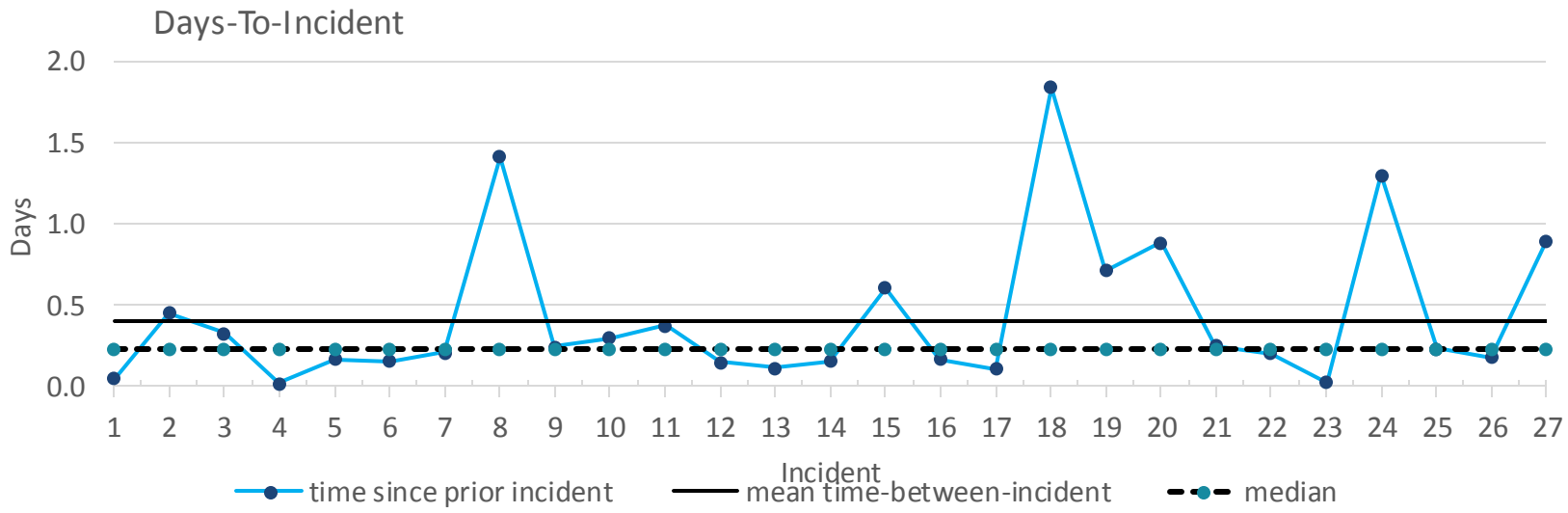
Why not median or mode? Does the distribution matter?



Look at Individual Points

Does it matter that these include two different distributions?

If we use MTTI in a regression, it may be critical!



Why the Mean?

In product reliability, the mean time to failure is used as an easy to measure indicator that relates to **down time**.

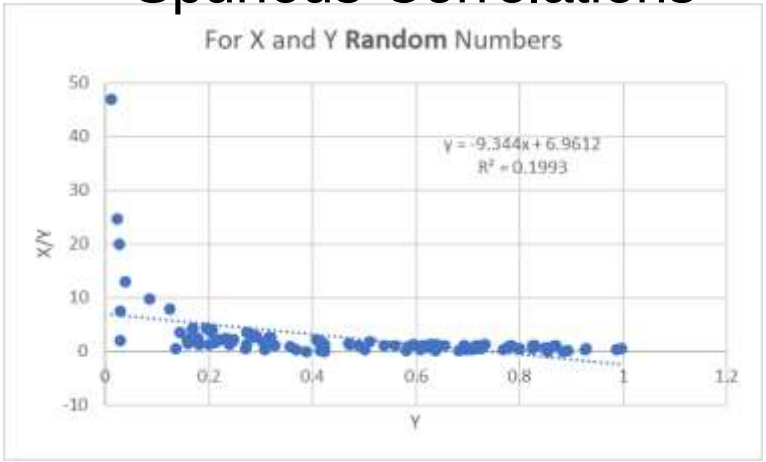
Assume each failure requires an average fix time.

A mean time to failure and a number of failures thus provides a good quick approximation of down time.

But, you cannot reliably

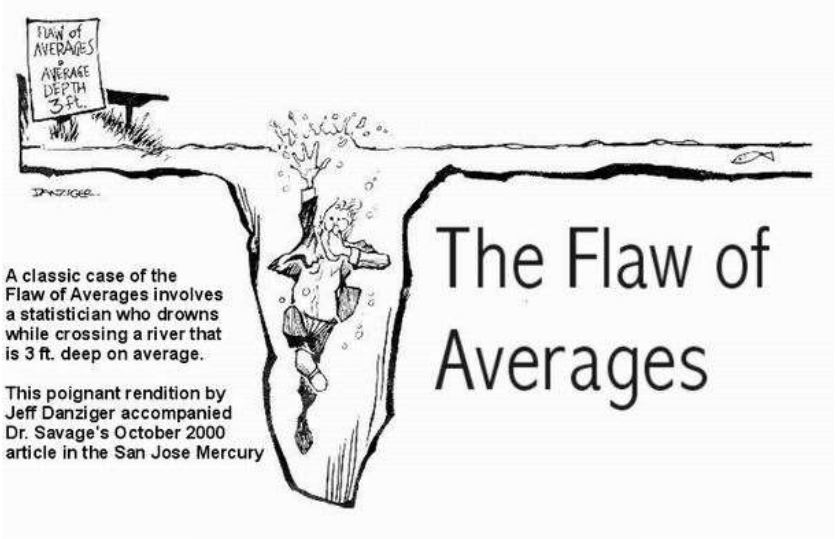
- use this derived value in a regression
- Use the MTTF as part of another derived calculation

Spurious Correlations



Mean Time to Incident

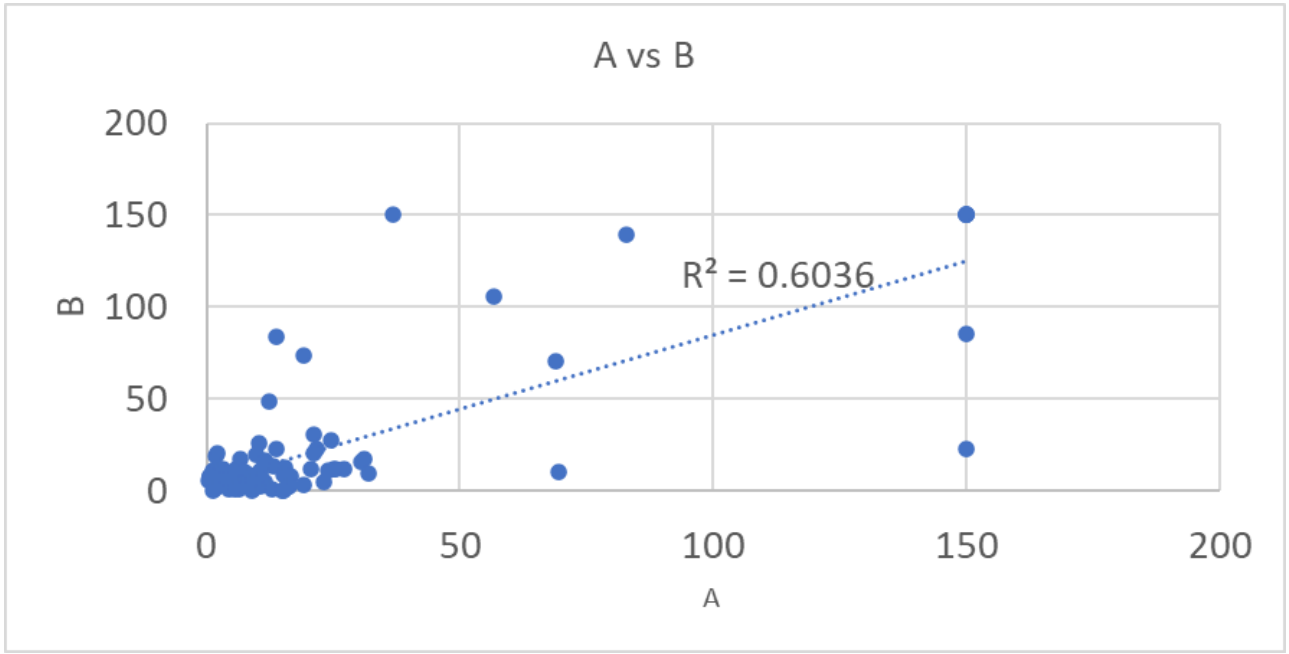
- How is the time measured?
- Why the mean?
- Are there sub-populations of user profiles?
- Under what conditions?
- Are they independent?



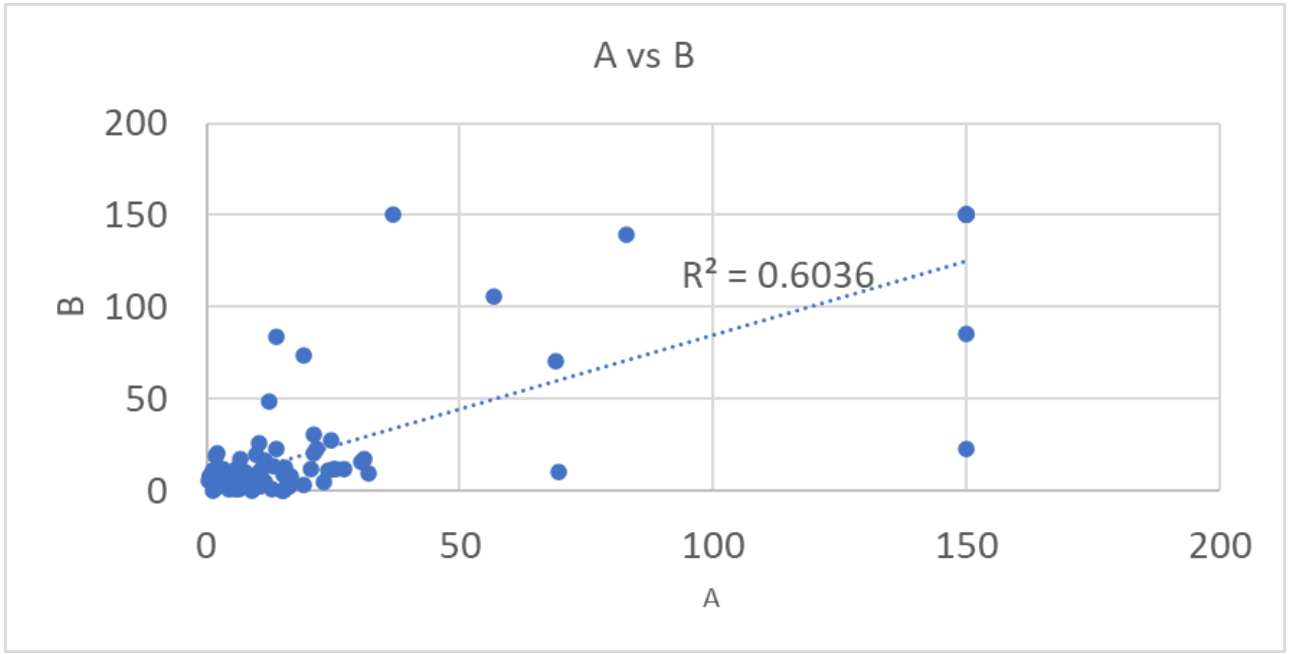
Points

1) Even a simple measure require careful definition
2) Human in the loop brings a large variety of additional factors
3) Easy to create, not so easy to interpret
4) Interpretation depends upon what you want to know

Caution! Does the Following Show Correlation?



Does the Following Show Correlation?



X=Rand()

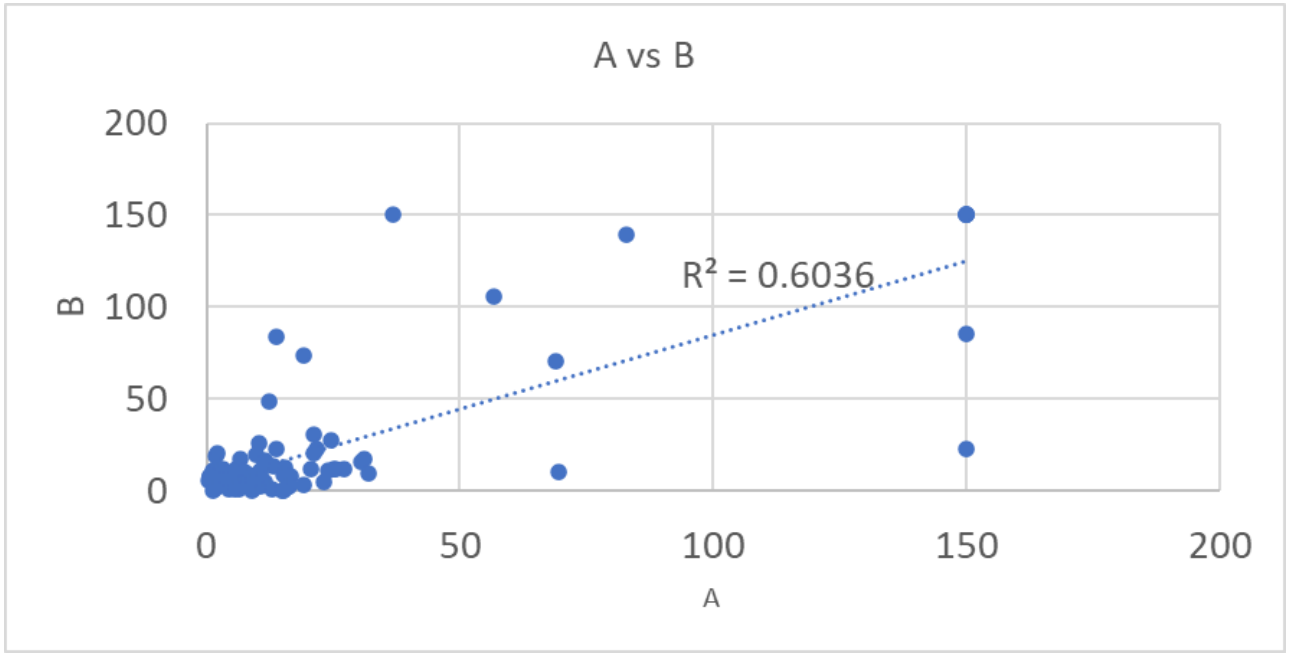
Y=Rand()

Z=Rand()

A = X/Z

B=Y/Z

Does the Following Show Correlation?



X=Rand()

Y=Rand()

Z=Rand()

A = X/Z

B=Y/Z

share a common denominator

This happens frequently with metrics!

Validity Criteria of Metrics (IEEE 1061)

1) *Correlation.* The metric should be linearly related to the quality factor as measured by the statistical correlation between the metric and the corresponding quality factor.

2) *Consistency.* Let F be the quality factor variable and Y be the output of the metrics function, M: F->Y. M must be a monotonic function. That is, if $f_1 > f_2 > f_3$, then we must obtain $y_1 > y_2 > y_3$.

3) *Tracking.* For metrics function, M: F->Y. As F changes from f_1 to f_2 in real time, M(f) should change promptly from y_1 to y_2.

Validity Criteria of Metrics (IEEE 1061)

4) *Predictability*. For metrics function, M: F->Y. If we know the value of Y at some point in time, we should be able to predict the value of F.

5) *Discriminative power*. "A metric shall be able to discriminate between high-quality software components (e.g. high MTTF) and low-quality software components (e.g. low MTTF). The set of metric values associated with the former should be significantly higher (or lower) than those associated with the latter.

6) *Reliability*. "A metric shall demonstrate the correlation, tracking, consistency, predictability, and discriminative power properties for at least P% of the application of the metric."

ISO/IEC/IEEE 15939 : 2017(E): ISO/IEC/IEEE International Standard - Systems and Software Engineering-Measurement Process.

decision criteria - thresholds, targets, or patterns used to determine the need for action or further investigation, or to describe the level of confidence in a given result

entity object that is to be characterized by measuring its attributes

Information need insight necessary to manage objectives, goals, risks and problems

measurable concept abstract relationship between attributes of entities and information needs

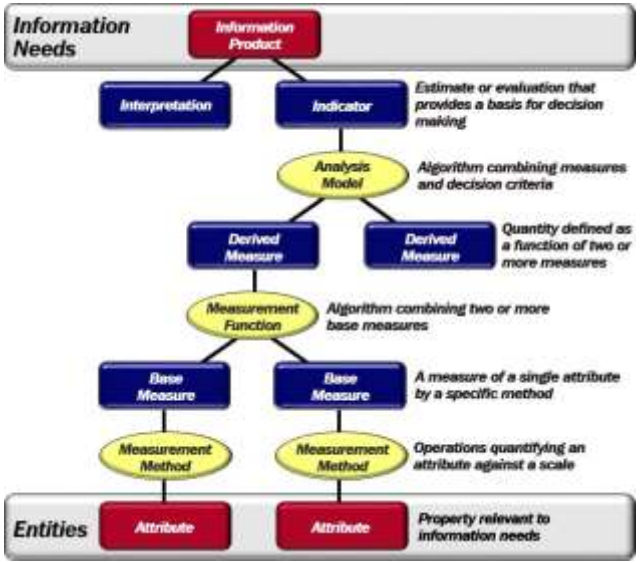
indicator measure that provides an estimate or evaluation of specified attributes derived from a model with respect to defined information needs

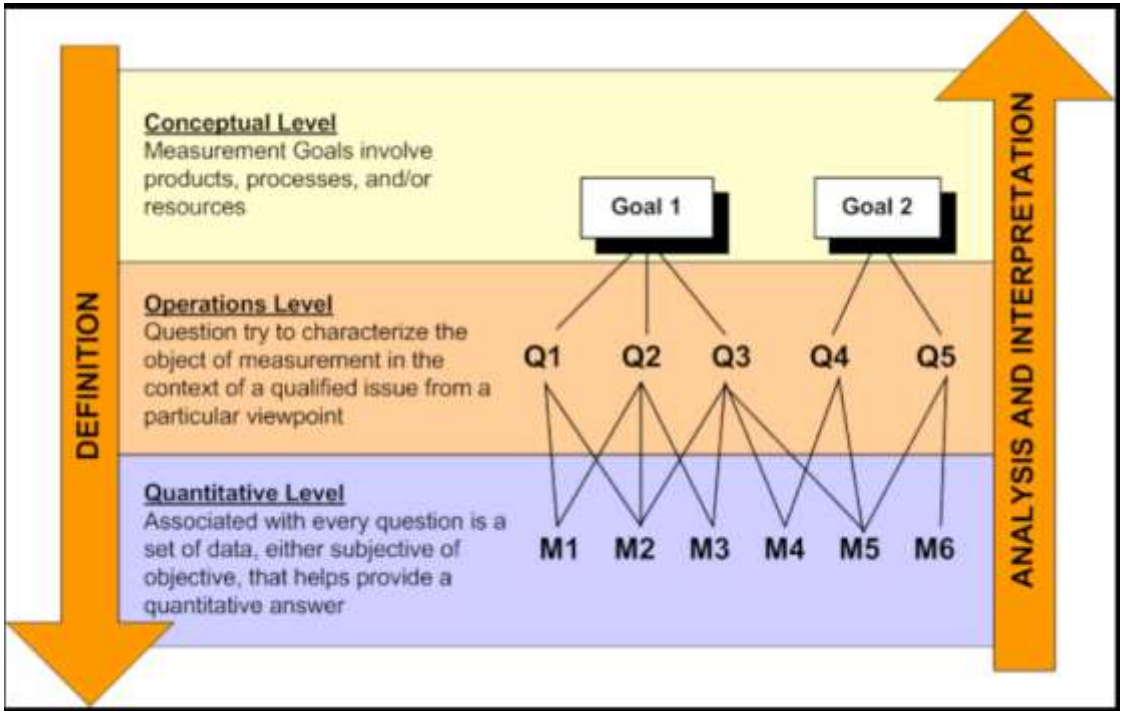
attribute property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means

base measure defined in terms of an attribute and the method for quantifying it

derived measure that is defined as a function of two or more values of base measures

IEEE Elements, and GQM (Basili)

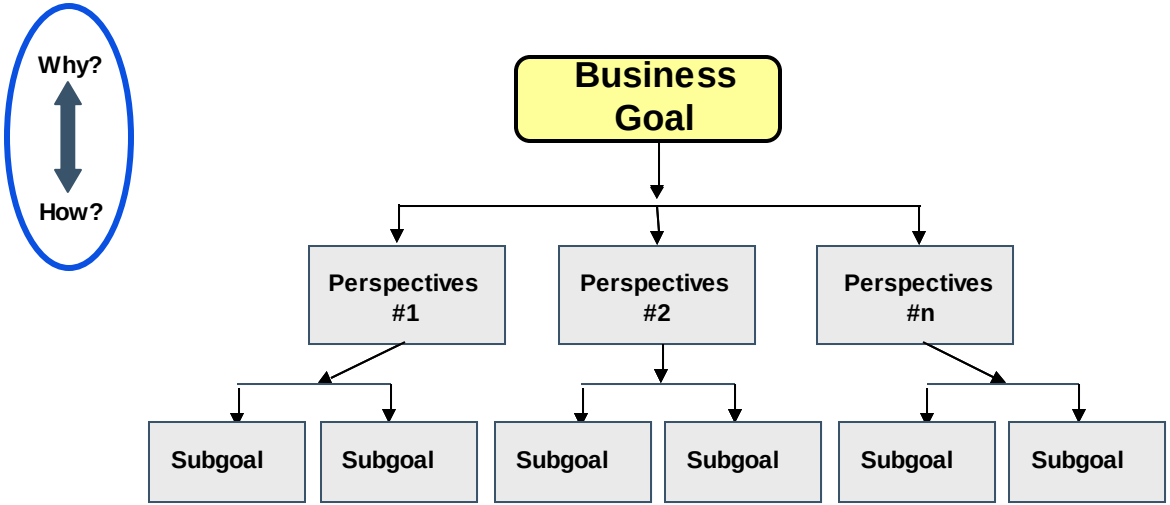




Subdividing the Goal

Now that we have a business goal, what do we do with it?

- **Indicators** address the **questions** about the goals



What does Security mean to a

- Developer
- Requirements engineer
- CEO
- User
- Tester
- Regulator

What do you need to know?

From each perspective, what can be done to support the business goal?

How does each process step contribute to the business goal?

Operationalize Goals

Components of an Operationalized Goal

Dimension	Definition
Object of study (item of interest)	What will be analyzed.
Purpose	Why the object will be analyzed.
Quality focus	The property/attribute of the object that will be analyzed.
Viewpoint or Perspective	Who uses the data collected. Who is interested in the results.
Context	In which environment. Under what constraints

Ref.: Solingen & Berghou

Goals

A conceptual objective for some object (e.g. a product, process, resource, or customer) defined from some point of view.

“Cybersecurity” is too abstract. Use the following template to make goal more specific.

Field		Examples
Object of study	For the	deployment process, operations
Purpose	I want to	characterize, understand, evaluate, predict, improve
Issue of focus	the	availability, security, privacy, timeliness
Stakeholder	for (whom)	developer, customer, manager, architect
context factors	when	(other important factors that may affect outcomes)

For the “web portal,” I want to “improve” the “security” for “users” when they enter sensitive data.

Template (Worksheet)

Goal or Subgoal: _____

Object of interest: _____

Purpose:

_____ *the* _____ *in order to* _____ *it.*

Quality Focus & Perspective:

Examine the _____

from the point of view of (the) _____ .

Environment & Constraints:

_____ , _____ , _____ , _____ ,

_____ , _____ , _____ , _____

Indicator - Challenges and a Solution

Many of the potential benefits that an organization can derive from a sound measurement program are often not achieved due to inconsistent construction and interpretation of indicators derived from measurement data.

The indicator template is a

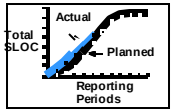
- tool an organization can use to direct its data collection and measurement and analysis processes
- comprehensive template that provides guidance for the development and precise description of an indicator

Data and Infrastructure Assessment

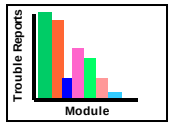
What data supports which indicators?

Where does that data come from?

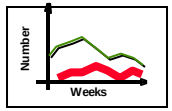
Success



Progress



Analysis



Required Data Elements

- Size
- Effort
- Weakness
- Vulnerability
- Incidents
- .....

Cross Reference Matrix

	Indicators a	b	c	d	e	Availability	Source
Size	x					+	QA
Effort	x					0	
Weakness		x	x		x	-	CM
Vulnerability		x		x			QA
Incidents			x				
.....		x		x	x		

Definition Checklist

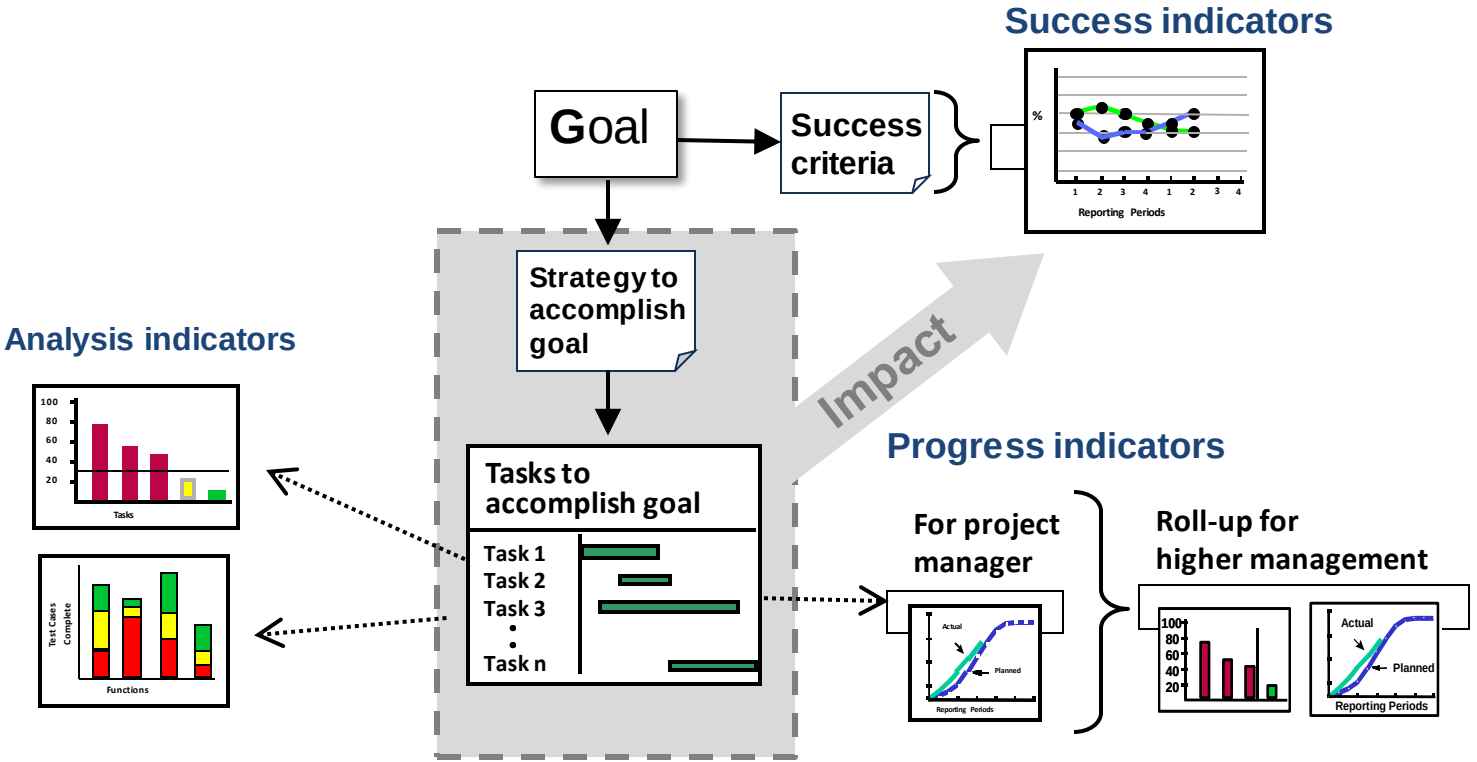
Success

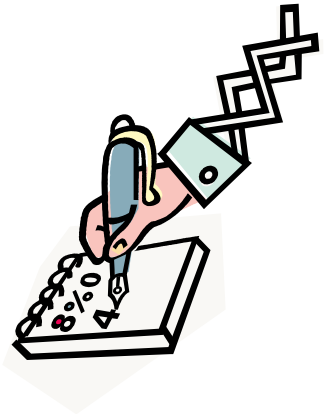
Progress

Analysis

What Kinds of Data Will We Need?

The SEI's Goal-Question-Indicator-Method





MEMO

Two Old SEI Proverbs

If all you want is a number, any number will do.

If you don't know what your numbers represent, no number will help.

Capers' Jones

If can only measure one thing, it doesn't matter. You will fail"

Bill's Corollary

If you can only measure one thing what would it be?

"how fast I can get out of here"

Plan Your Measures: Goal, Question, Metric

Goal: Before measuring, clarify the concept.

- What is your goal or objective?
- What do you want to know?

Question: Operationalize the concept with a model.

- What do you need to know?
- How much information do you need?
- How accurate and precise must the data be?
- How much is that information worth?

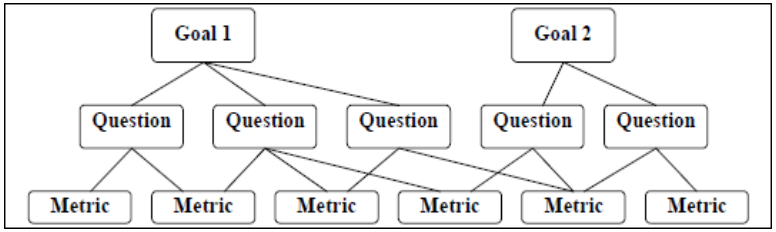
Metric: Quantify the questions

- What can you measure to answer each question quantitatively?

Goal-Question-Metric (GQIM)[1] Method

GQM is a systematic approach for defining metrics for evaluating organizational objectives with three analysis steps: conceptual, operational, quantitative.

…there are often many observable characteristics that are potential candidate metrics, but only a few that support the organization's goals.
– V. Basili



Conceptual - *Goal*

- What are the goals that the organization wants to achieve?
- What is purpose and issue(s) that the goals are addressing?
- What is the object of the study (product, process, resource)?
- What stakeholder perspectives are involved?

Operational - *Question*

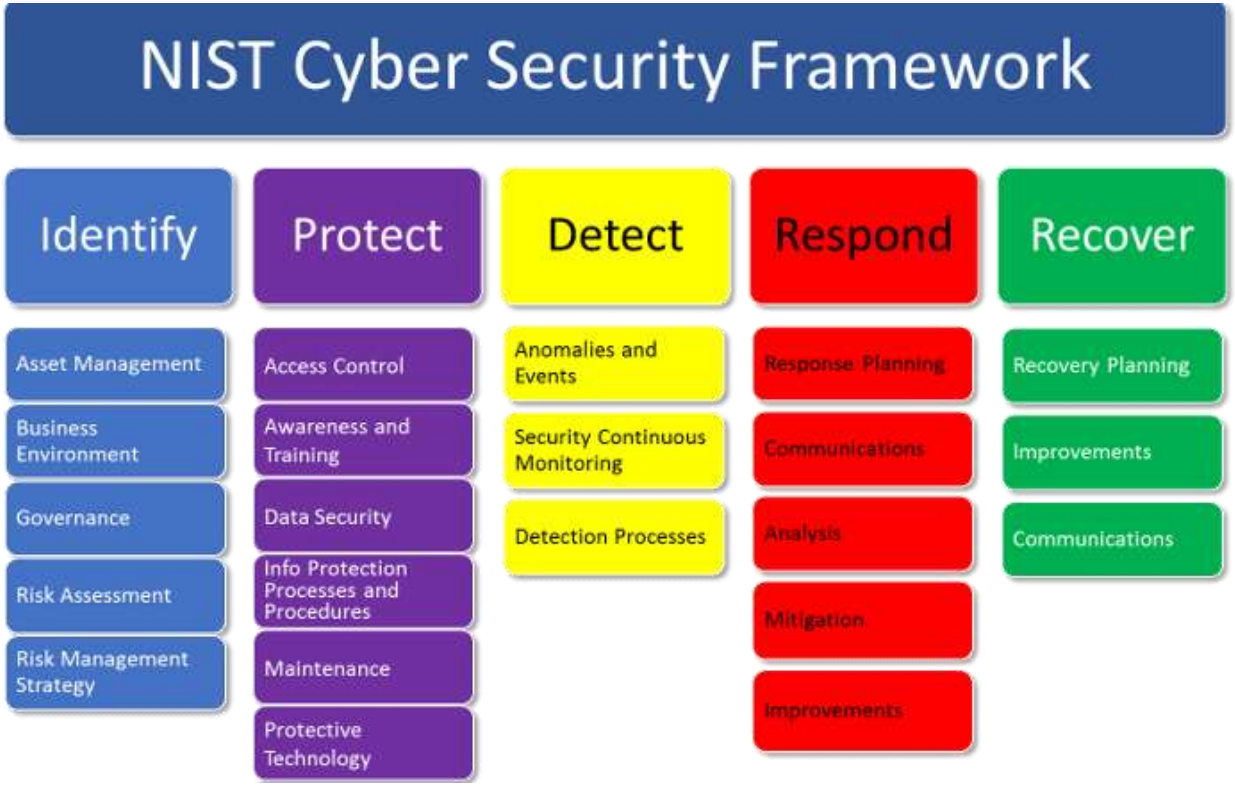
- What questions need to be answered to determine goal satisfaction? (Questions should try to characterize the object of measurement (product, process, re-source) with respect to a selected quality issue and viewpoint.)

Quantitative - *Metric*

- What metrics and metadata provide contextually valid data to answer the question(s)? What data sources are available? How will the reliability of the data be assessed? Are there objective and direct measures that can be used or are subjective measures necessary?

1. GQM was developed by Dr. V. Basili, U. of Maryland [ref].

What is Cybersecurity?



Security and Software Assurance: Definitions

Security

- A condition that results from the establishment and maintenance of protective measures that enable an organization to perform its mission or critical functions despite risks posed by threats to its use of systems. Protective measures may involve a combination of deterrence, avoidance, prevention, detection, recovery, and correction that should form part of the organization's risk management approach. [NIST 800-53, Rev 5]

Cybersecurity

- Prevention of damage to, protection of, and restoration of computers, electronic communications systems, electronic communications services, wire communication, and electronic communication, including information contained therein, to ensure its availability, integrity, authentication, confidentiality, and nonrepudiation. [NIST 800-53, Rev 5]
- the state of being protected against the criminal or unauthorized use of electronic data, or the measures taken to achieve this.

Security and Software Assurance: Definitions

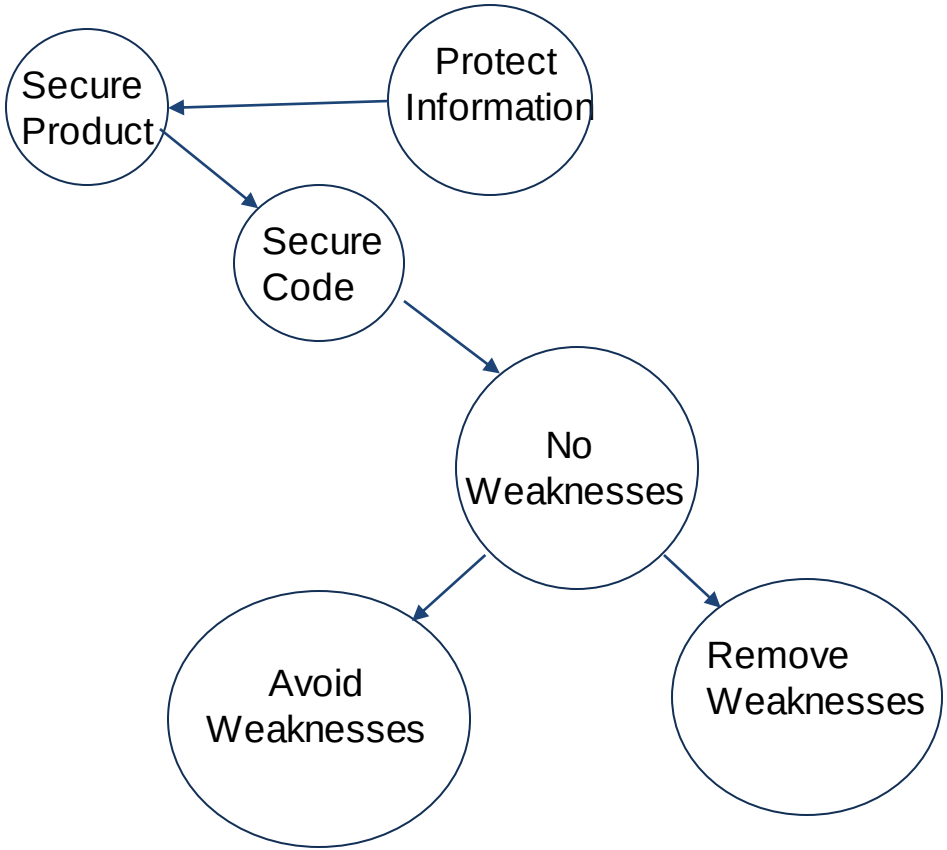
Software Assurance, n the level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software, throughout the lifecycle DoDI 5200.44

Software Assurance, v The planned and systematic set of activities that ensure that software life cycle processes and products conform to requirements, standards, and procedures. NASA-STD 8739.8

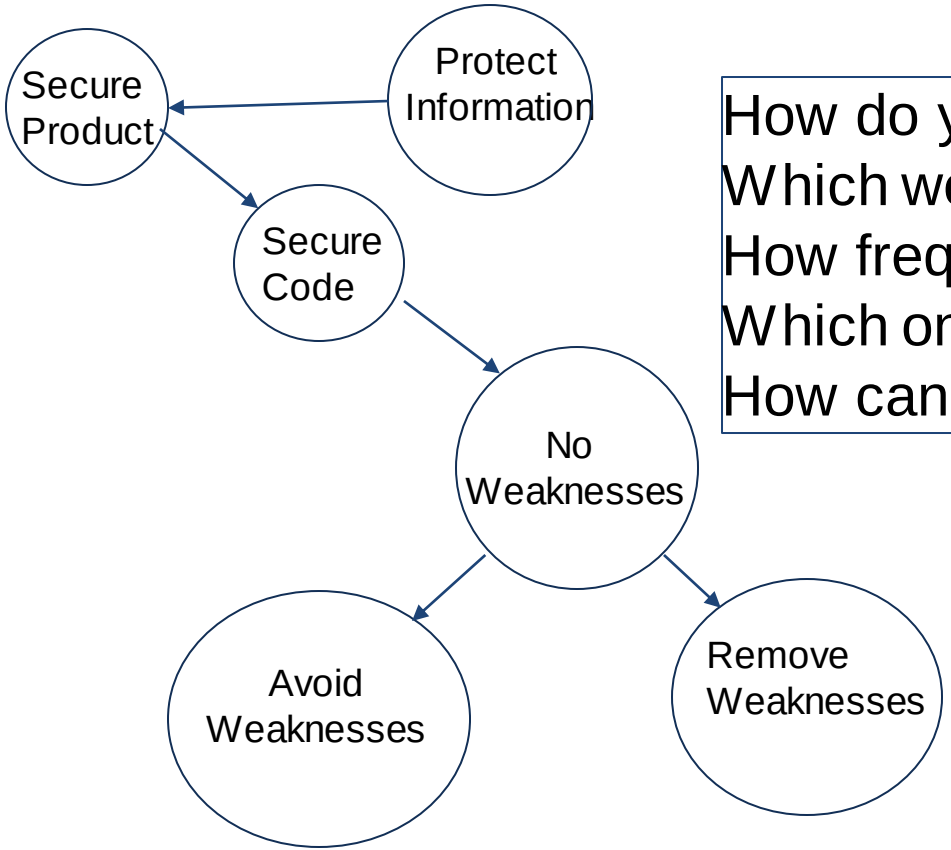
Some Cybersecurity Metrics

How to choose

Goals Hierarchy



Goals Hierarchy

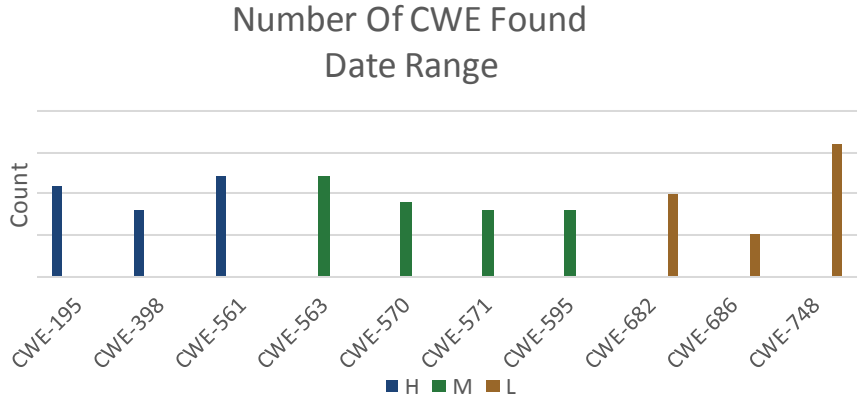


How do you detect weaknesses?
Which weaknesses are found?
How frequently?
Which ones matter?
How can they be addressed?

Which CWE Weaknesses have we Found

Goal As a software development lead, I want to know what kinds of weakness are injected into our system so that we can improve our development and evaluation process to find and remove those weaknesses.

Visualization, a frequency histogram of discovered CWE.



Include all weaknesses identified in completed products. The weaknesses need to be identified by the Common Weakness Enumeration and sorted by severity and adjusted severity.

Common Weaknesses

Purpose:

Compare how frequently different **types** of CWE identified through static analysis, dynamic analysis, inspections, or test in order to assess needs for

- Training
- Process changes
- New tools

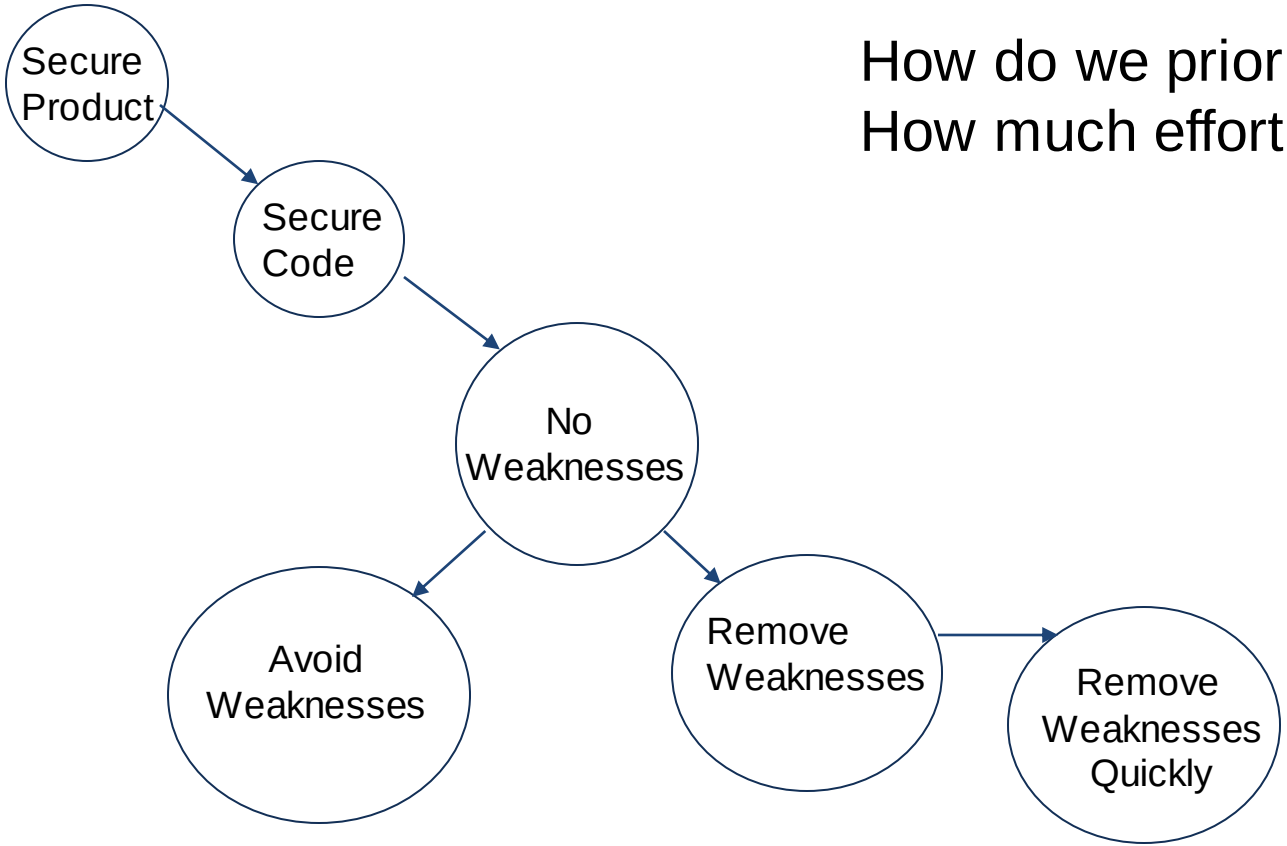
Question:

Which weaknesses are both common and high severity?

Follow up Question:

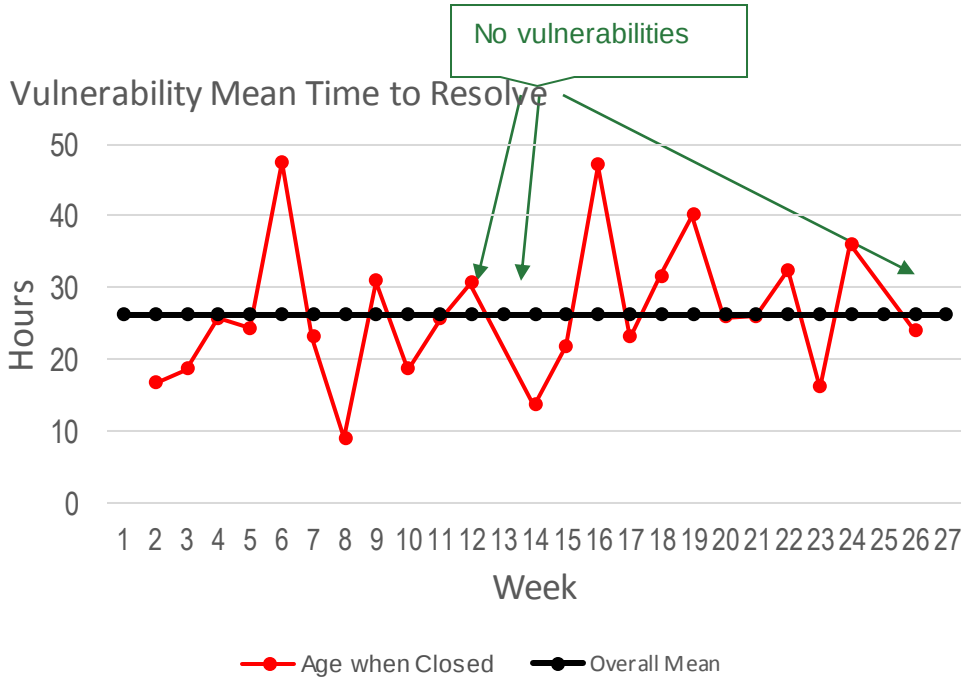
- Is this weakness from requirements, design or coding?
- Did we find them all?

Goals Hierarchy



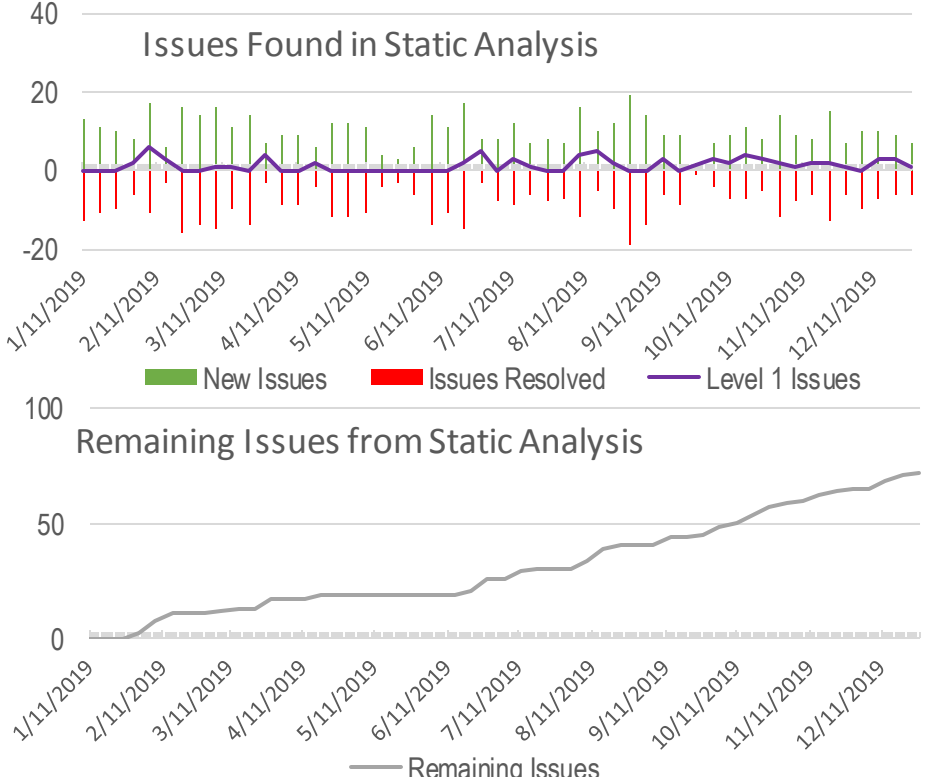
How do we prioritize?
How much effort to remove?

I want vulnerabilities to be removed as quickly as practicable



Under normal operating conditions

Static Analysis Finds and Build Up



The Remaining Issues chart shows that total issues are building up.

Vulnerability Fix Time,

Question: How long does it take to fix vulnerabilities?

- When did the clock start? Time of discovery taken from trouble ticket.
- When did it finish? Time that fix was deployed

Related questions

How frequent are vulnerability discoveries?

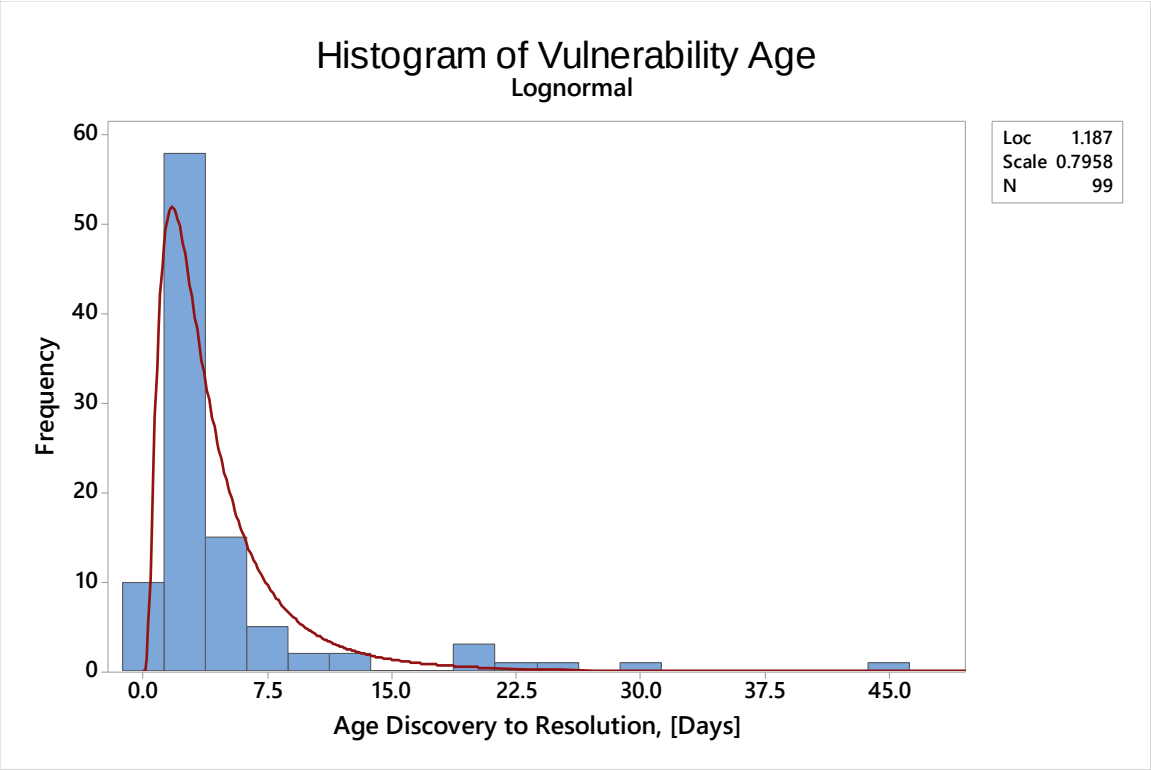
How quickly could a vulnerability be fixed and deployed?

What is the longest time a vulnerability stays in the system

Related Plots

Vulnerability fix time CDF

Alternate indicator, example of metric reuse



Is the Product or Network Being Attacked?

As a SecOps, I want to know if we are being attacked so that I can plan resources or prepare for an adversarial attack measures such as blocking traffic.

Questions include :

- Are there an unusual number of failed access attempts or rules violations
- Are there spikes in traffic from suspicious locations

Data sources include file server logs, DNS logs, and failed login attempts.

Interpretation of Application or Network Usage and Traffic

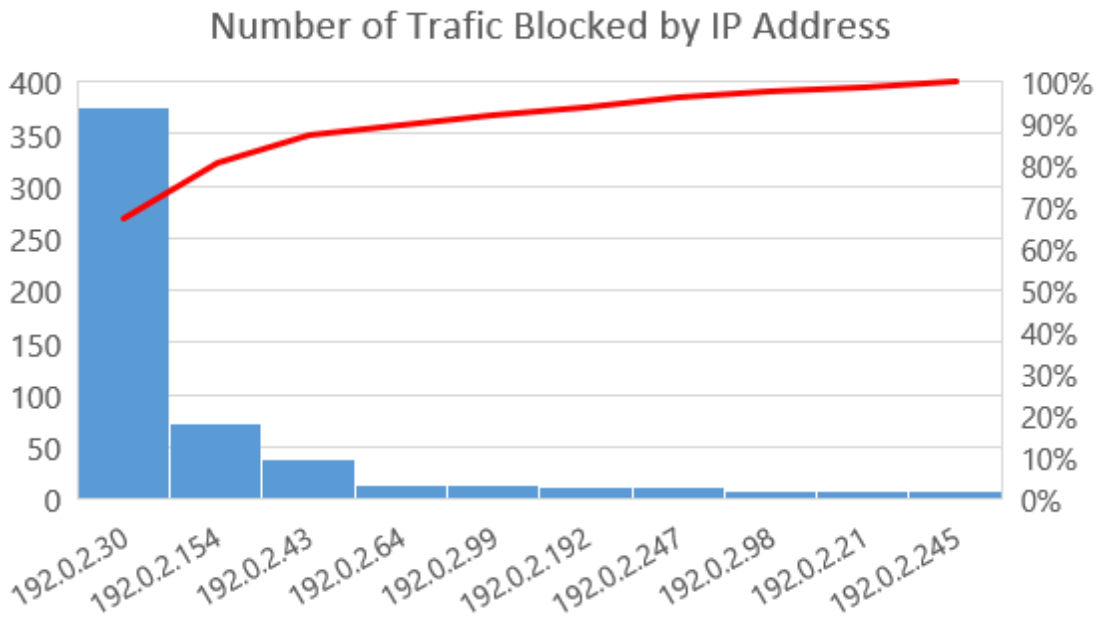
Direct Measure(s): Instances, requests, and memory consumption.

Change in application usage may indicate

- A need for additional infrastructure
- A need for product changes or a new product
- Adversarial probing or a denial-of-service attack

Presentation: Tool specific, typically run charts with baselines, averages, and action thresholds and pareto sorts

Pareto of Blocked Traffic by IP address



Goal Question Indicator Metrics, A Structured Framework

This with IEEE leads into GQIM

Characteristics of a Good Metric

A DevSecOps metric must be:

Observable: A metric that can't be measured is useless.

Actionable: It should suggest the need for corrective actions or improvements to workflows, policies, incentives, tools, etc.

Relevant: It must be related to a business goal.

Traceable: It should be possible to causally trace the metric to root causes.

Reliable: It should produce similar results under similar conditions and resist manipulation.

Automatable: Metrics collection should be built into the system to avoid manual work, errors, and delay.

Future Trends

Cloud

DevSecOps

Structured Methods

Future Trends

DevSecOps

- Automated Measurement
- Prescriptive process and environments

Structured Approaches ,

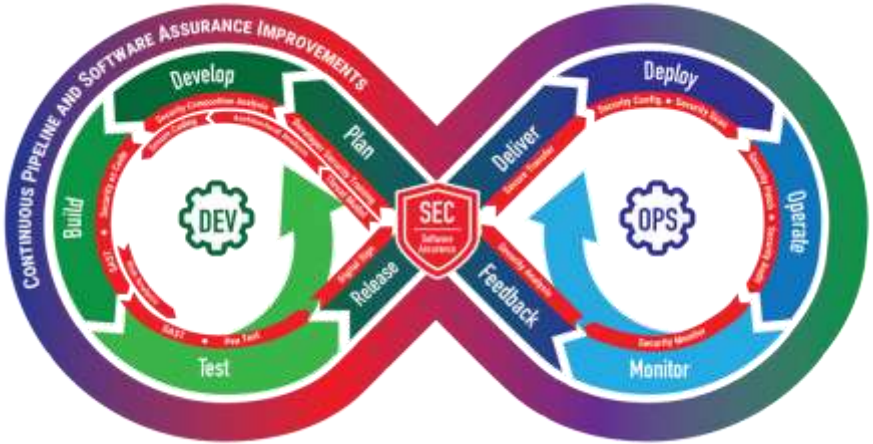
- Platform independent and dependent models
- formal definition

Cloud

- Measurement as a service

Predictions are hard
Especially about the future

DevSecOps: a Complex Socio-Technical Information System



DSO is an approach that integrates development (Dev), security (Sec), and delivery/operations (Ops) of software systems to reduce the time required to move from need to capability and provide CI/CD with high software quality [1].

The DSO Continuous integration (CI) and Continuous Development (CD) pipeline is a **socio-technical system made up of both a collection of software tools and processes** [2].

DSO CI/CD is **not a system to be built or acquired**, it is a personal and organizational **mindset** defining processes for the rapid development, fielding, and operations of software and software-based systems **utilizing automation where feasible** in order to achieve the desired throughput of new features and capabilities.

[1] Guide to Implementing DevSecOps for a System of Systems in Highly Regulated Environments, CMU/SEI-2020-TR-002
[2] Len Bass, Ingo Weber, and Liming Zhu. 2015. DevOps: A Software Architect's Perspective (1st ed.). Addison-Wesley Professional.

DevSecOps Driven by Automation

Continuous Integration (CI): The process (automated) in which developers build, test, and validate new code.

Continuous Delivery (CD): The process (automated) of creating releasable artifacts.

Infrastructure as Code: The scripting and/or virtualization of infrastructure that replicates the operational environment and optimizes computing resources.

Automation of test: The scripts that carry out both functional and interface testing.

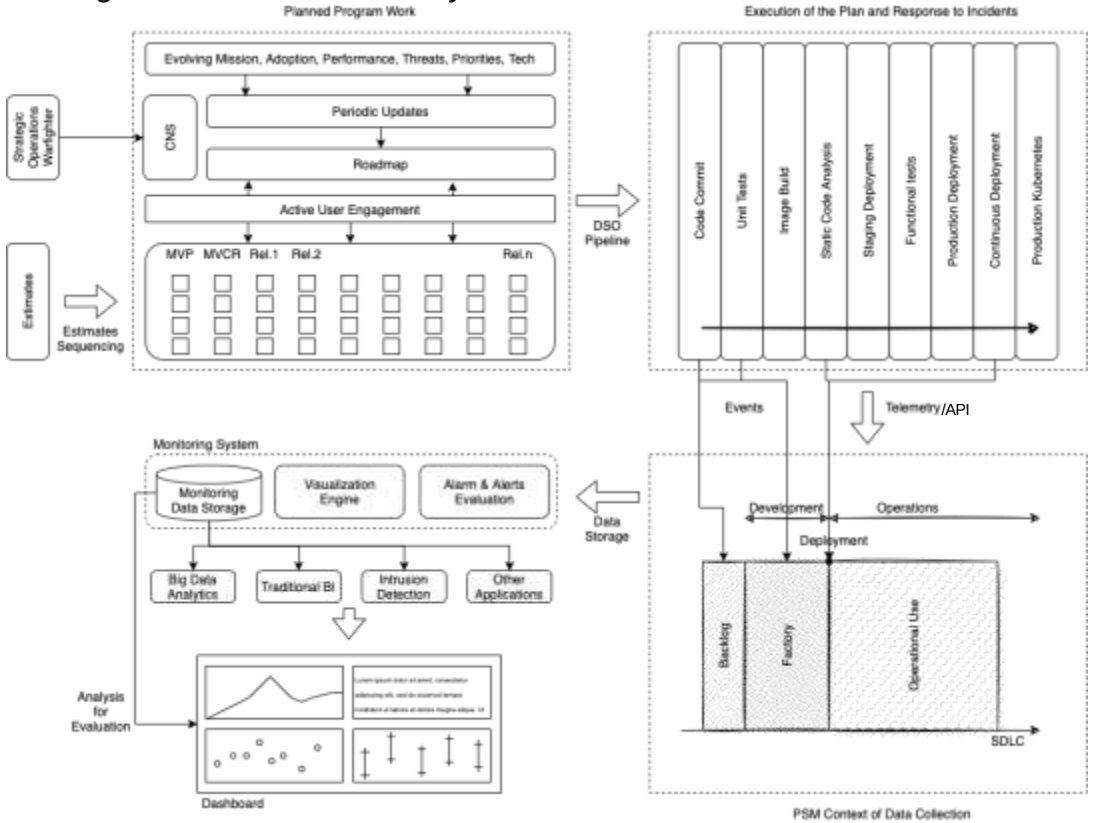
Automation of security: Scripts perform security checks

Monitoring and measurement: Constant measurement of the environment to proactively address issues

How do we get the data? Data Collection Context

Managed with Jira, Gitlab, Rally

Factory Pipelines



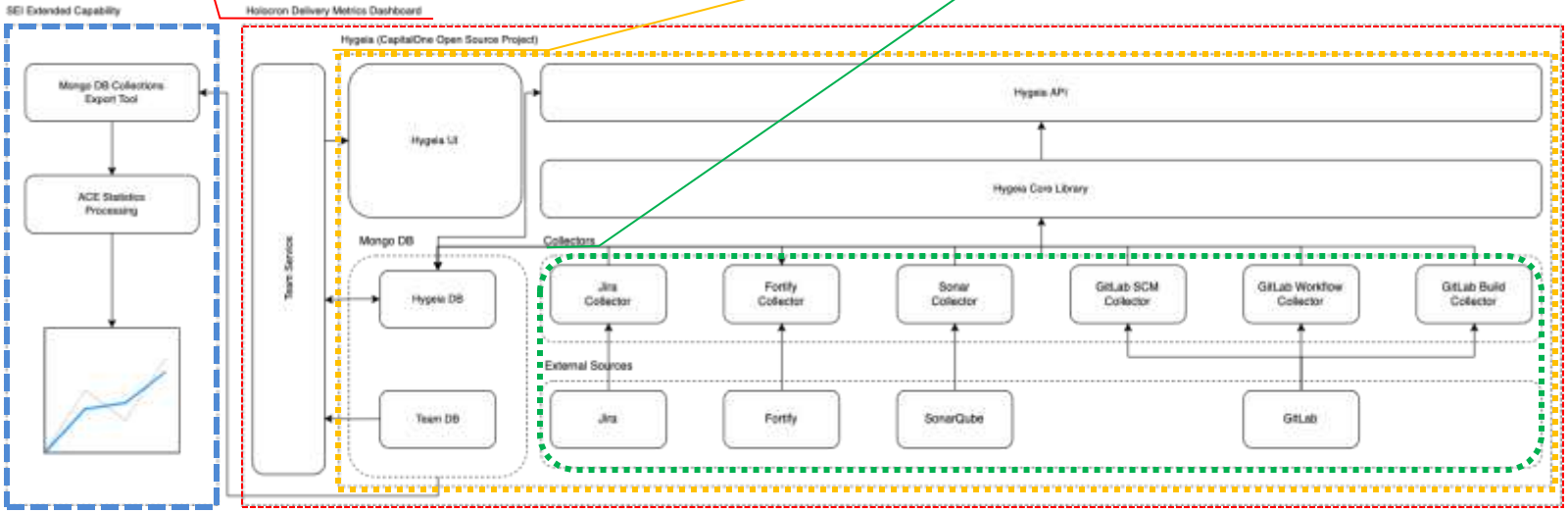
Data is collected and **Transformed** for storage

The **Warehouse** loads the data and provides the interface for analysis and dashboards

Technical Approach: Using Hygeia/Holocron

Apply on local development pipeline (instrumenting a local research project)

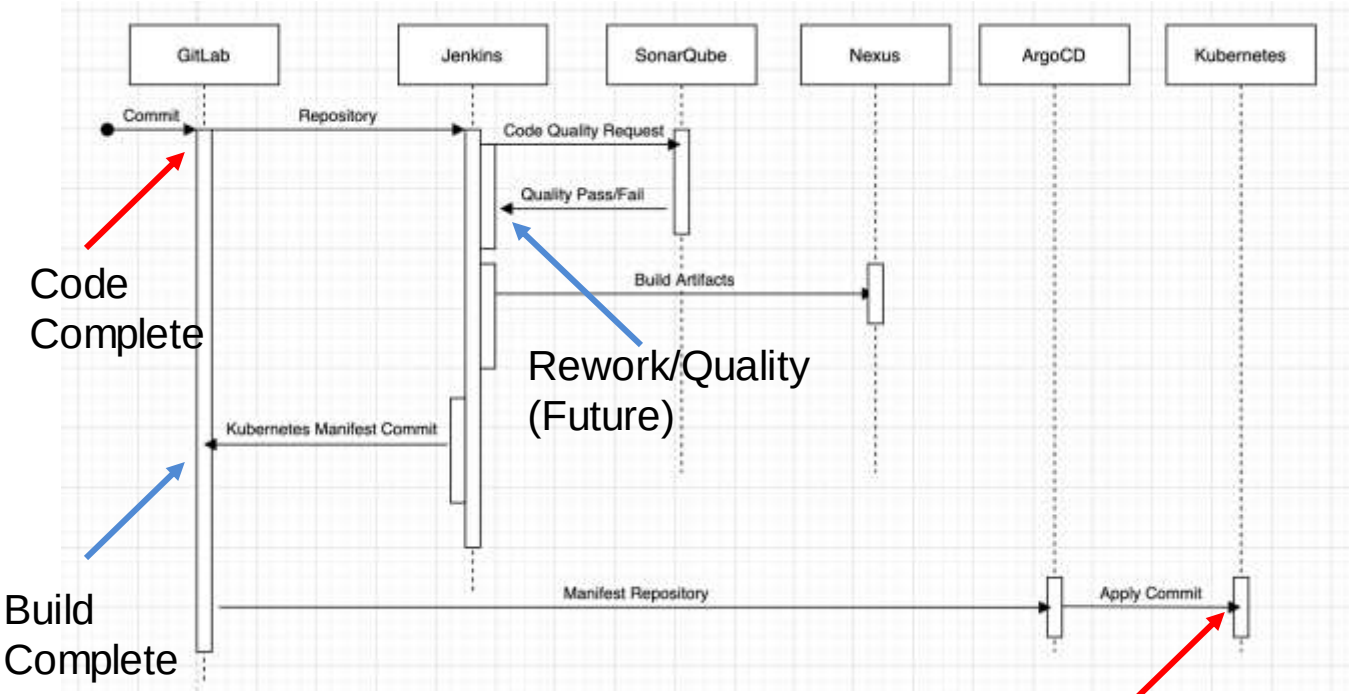
Holocron provided by Platform One, uses Hygeia Collectors



External Sources
- Jira
- Gitlab
- Sonarqube
- Fortify

How do know work is DONE? Look inside the Pipeline

Extracting metrics https://youtu.be/u96OFTXgr0g



Date is collected from key events

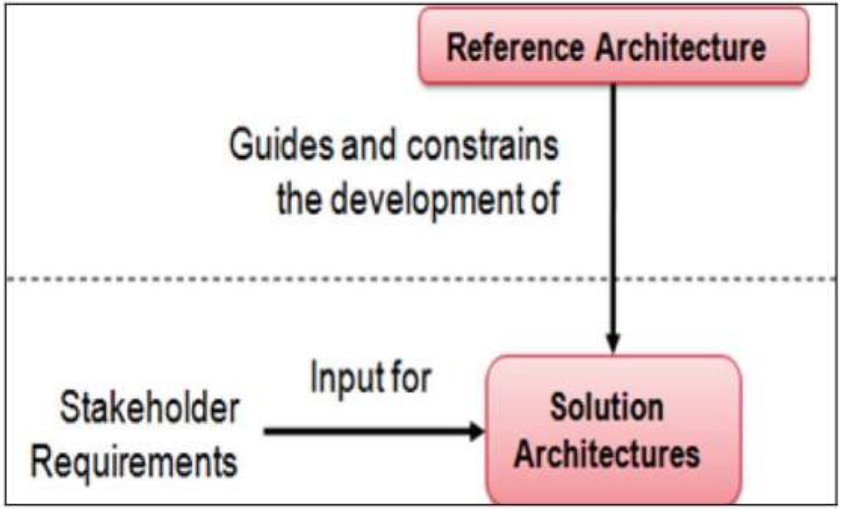
The data specification is on the following slide

- Lead times,
- Estimated Dates
- Actual Times

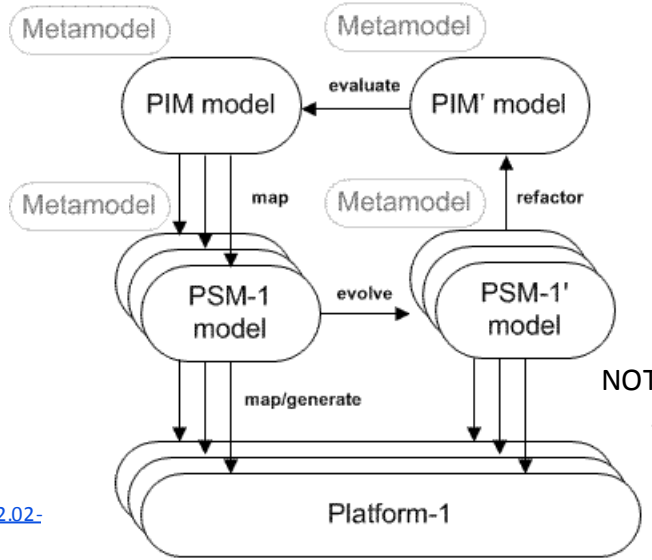
We can **precisely** define the measures and metrics

Reference Architecture/Platform Independent Model (PIM)

A **Reference Architecture** is an authoritative source of information about a specific subject area that guides and constrains the instantiations of multiple architectures and solutions [3].



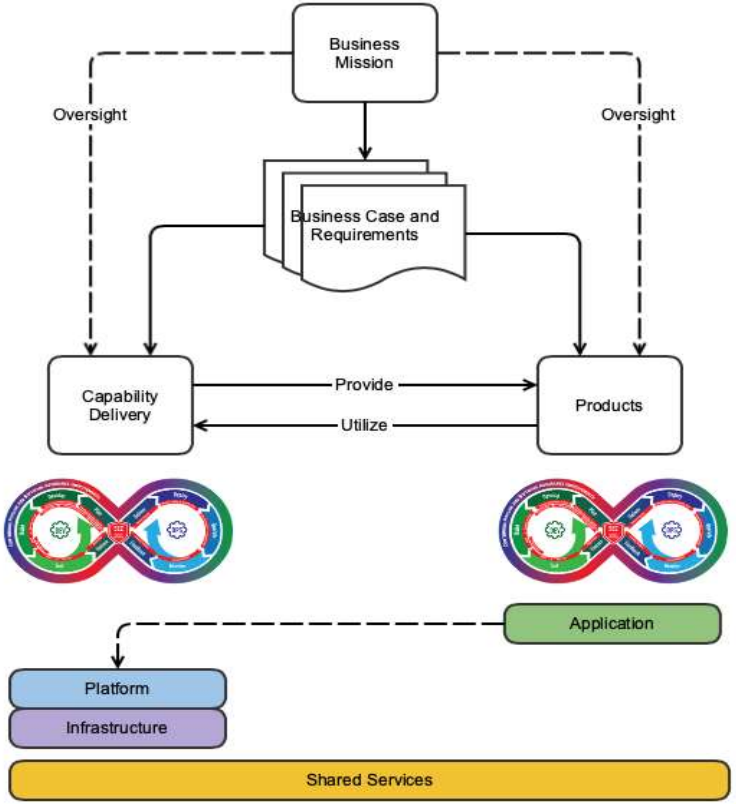
A PIM is a general and reusable model of a solution to a commonly occurring problem in software engineering within a given context, and is independent of the specific technological platform used to implement it.



NOTE: PSM = Platform Specific Model

[3] DoD Reference Architecture Description, Background - DODAF - DOD Architecture Framework Version 2.02 - DOD Deputy Chief Information Officer (defense.gov)

Challenge for DSO: cybersecurity of pipeline and product



Managing and monitoring all of the various parts to ensure the product is built with sufficient cybersecurity and the pipeline is maintained to operate with sufficient cybersecurity is complex.

Cybersecurity demands effective governance to address:

- What trust relations will be acceptable, and how will they be managed?
- What flow control and monitoring are in place to establish that the pipeline is working properly? Are these sufficient for the level of cybersecurity required?
- What compliance mandates are required? How are they addressed by the pipeline? Is this sufficient?

Cloud

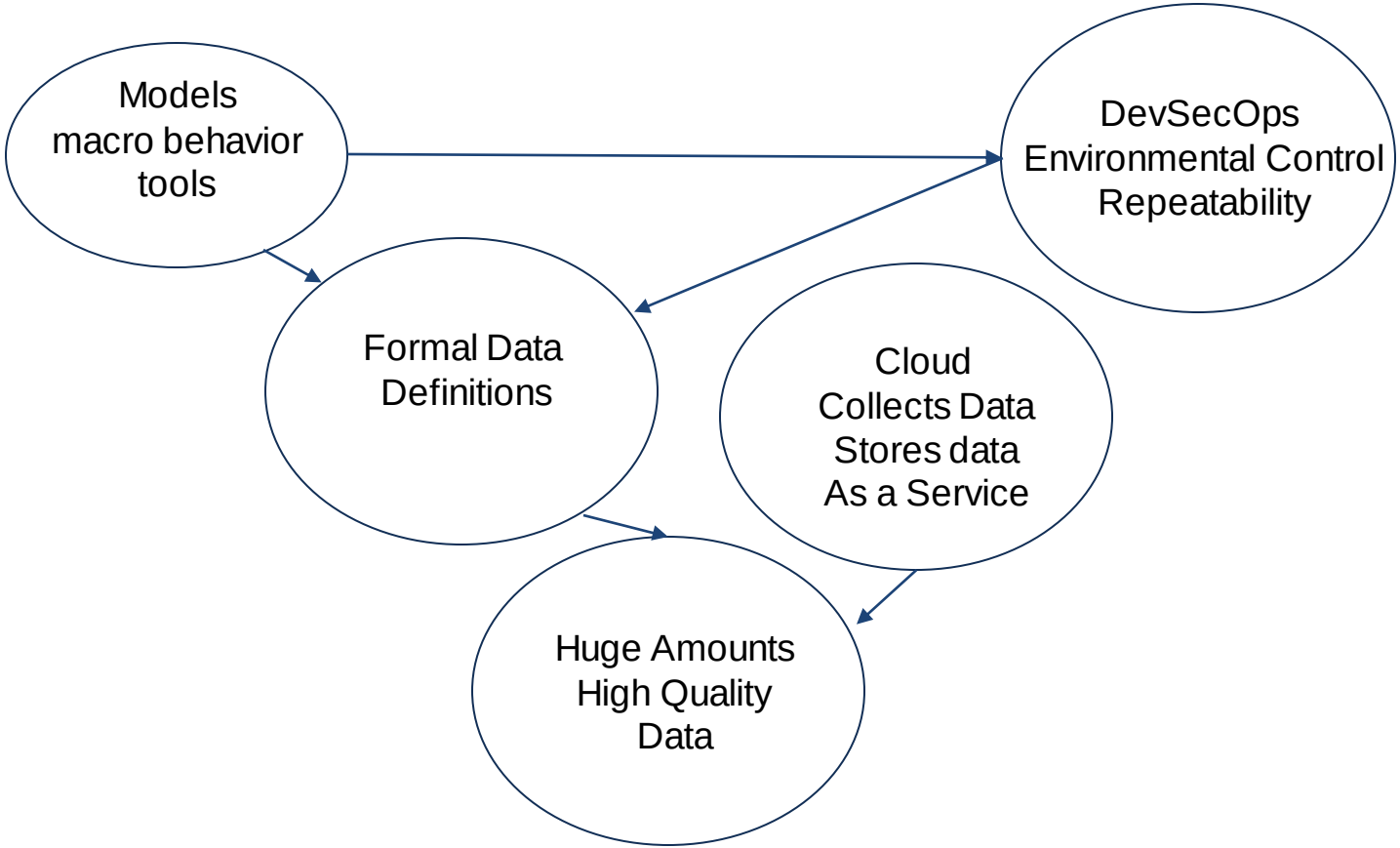
Risks and Challenges

- Reduced visibility and control
- On-Demand service simplifies unauthorize use
- Internet-accessible Management APIs can be compromised
- Separation among Multiple tenants fails
- Data deletion is incomplete

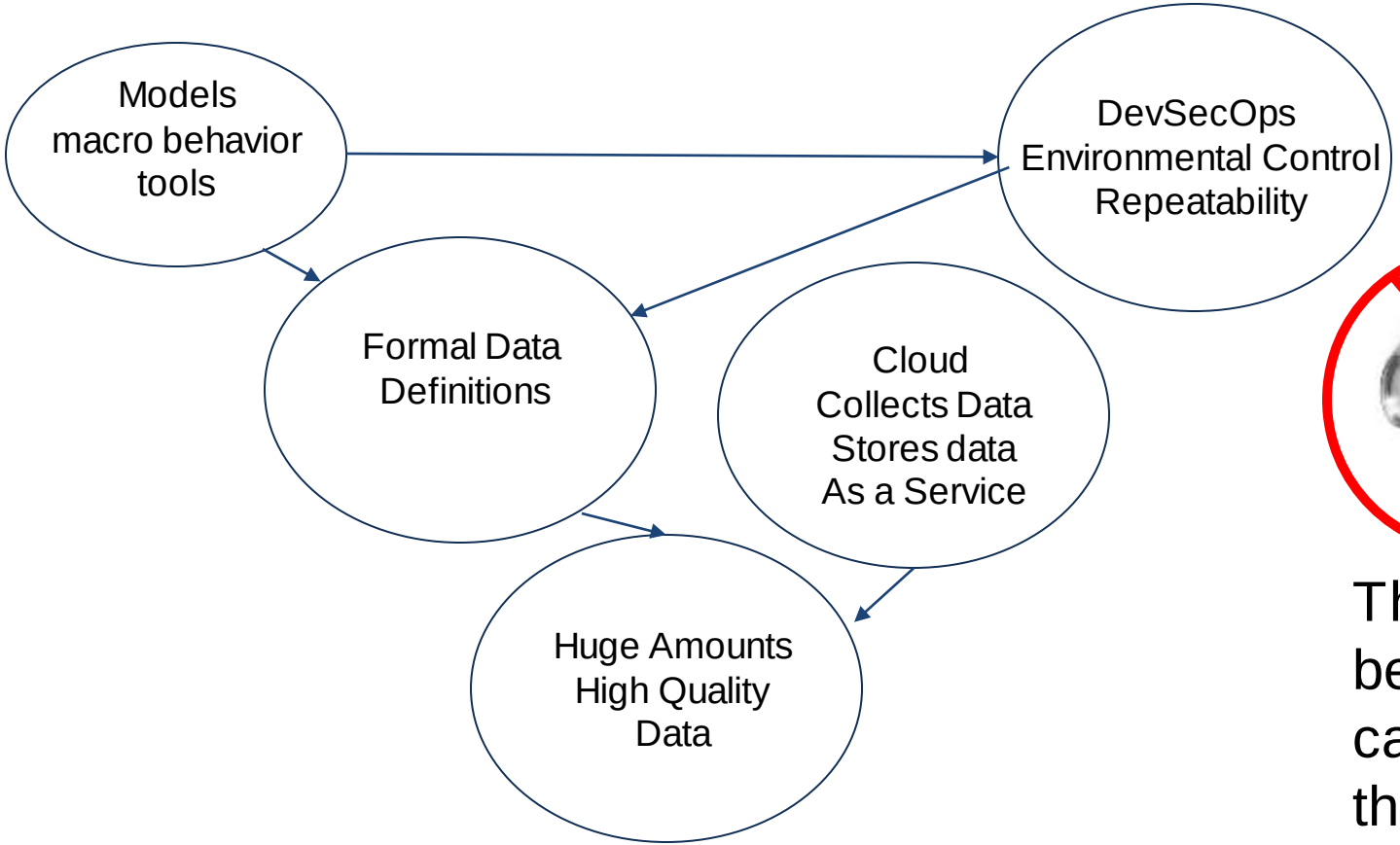
Big change,

- Measurement as a service

Summary of the Future



Summary of the Future



The risk is becoming careless with the analysis

Conclusion

Art for Art's Sake

Measure with Purpose

Contact Information

William R. (Bill) Nichols , Ph.D.

wrn@sei.cmu.edu

Web Resources

https://sei.cmu.edu/

Example of Complexity: Tool Management -2

Each tool type requires specific technical skills that must be drawn from the integrated capabilities (Dev, Sec & Ops) and work together in the process flow.

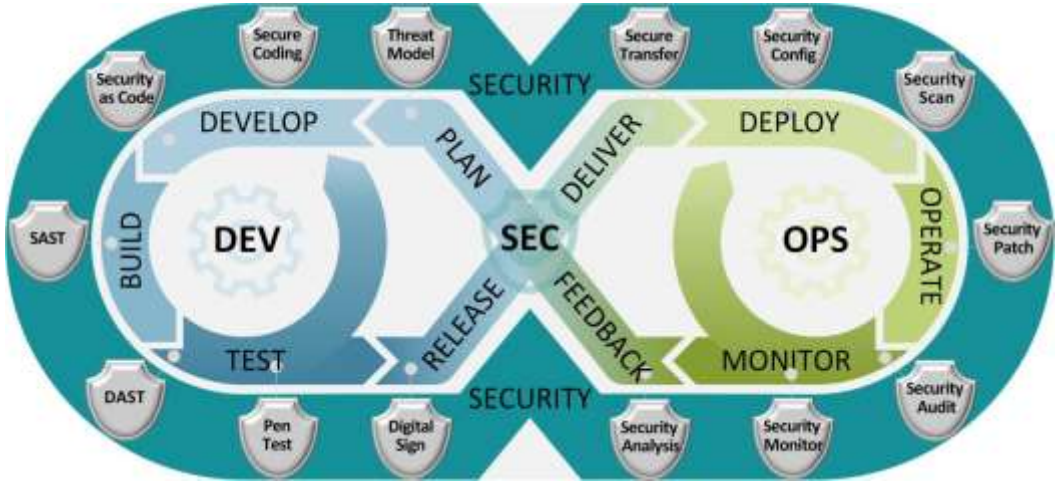
- Product build should move through the security activities as part of the pipeline flow.
- Security considerations can be in the control gates for the pipeline flow.

Process Type	Process	Security Activities
Dev	Plan	Threat Model
	Code	Secure Coding
	Build	SAST, Security as Code
	Test	DAST, Pen Test
	Release	Digital Sign
Ops	Deliver	Secure Transfer
	Deploy	Security Configuration and Scan
	Operate	Security Patch and Audit
	Monitor	Security Monitor
	Feedback	Security Analysis

Pipeline flow does not address security for the pipeline's capabilities

- Pipeline security must be integrated into the roles and responsibilities of those that administer and support these capabilities.
- Pipeline administrators should perform similar processes and use similar tools, but they are applied to different content.

DevSecOps Measurement Informs Decisions



Source::**DoD Enterprise DevSecOps Reference Design**

What to do next?

Is the code ready for test?

Does the code need more test?

Example of Complexity: Tool Management -3

A range of processes can be allocated to various pipeline administrative roles.

Each process focuses on a different component of the pipeline, but all processes are needed to keep the pipeline functioning effectively.

Due to this complexity and repurposing of existing administrative resources without integrated oversight, infrastructure services and development tool types are increasingly the target of attacks.

Manual validation is impossible – too many combinations

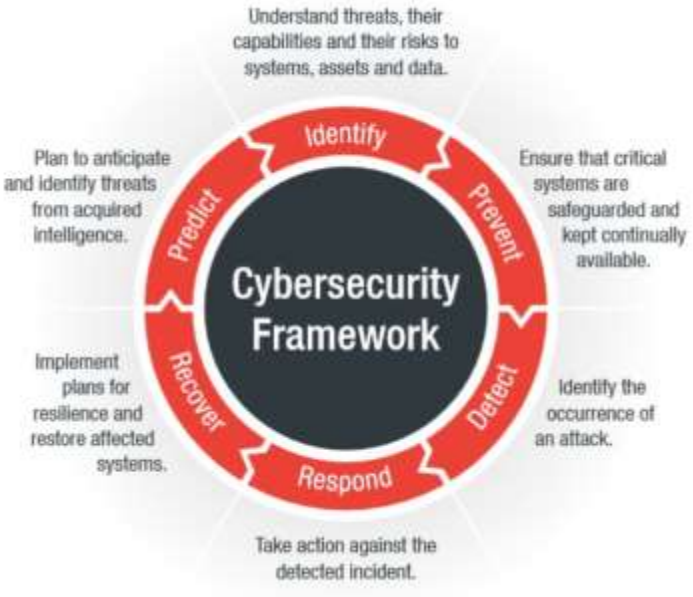
Operational Process	Component	Role
Add Hardware	Host System	infra
Code Software	Source Control System Issue Tracking System IdAM Communication System Code Review System	dev
Configure Infrastructure	Host System	infra
Decommission Hardware	Host System	infra
Deploy Application	Any	ops
Disaster Recovery	Any	all
Install Software	Any	admin
Manage Incidents	Monitoring System	admin
Manage Users	IdAM System	admin
Monitor Infrastructure	Monitoring System	infra
Operate Solutions	Any	ops
Patch Infrastructure	Host System	infra
Patch Software	Any	admin
Perform Backup	Any	admin
Review Logs	Monitoring System	ops
Test Applications	Any	dev

Dysfunctional Behavior

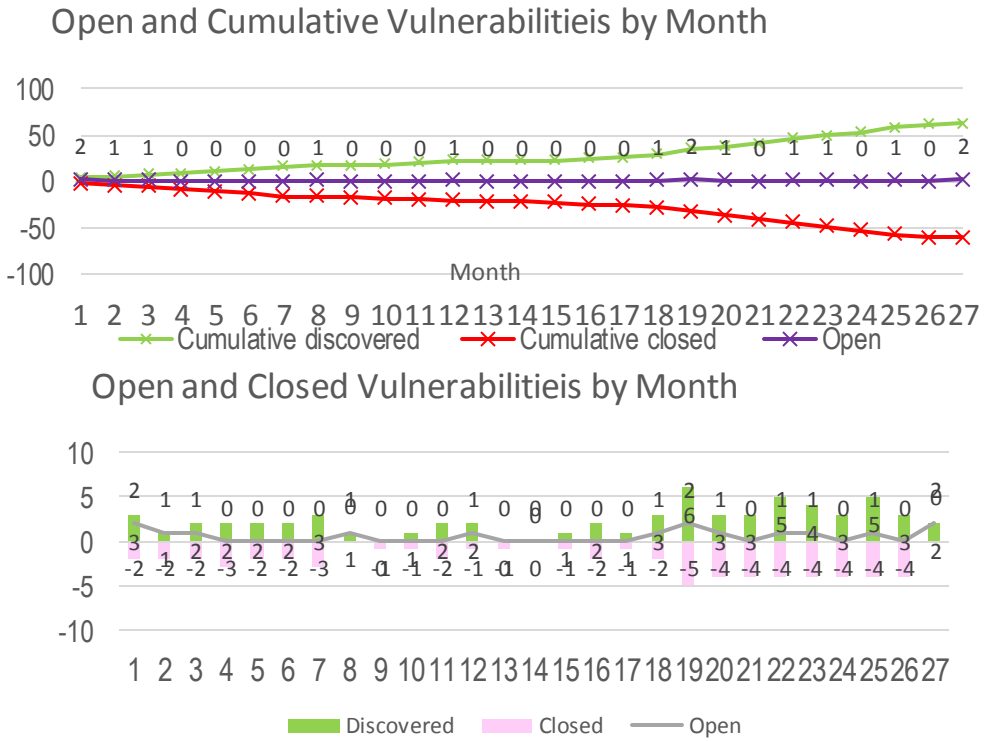
Austin Measuring and Managing Performance in Organizations

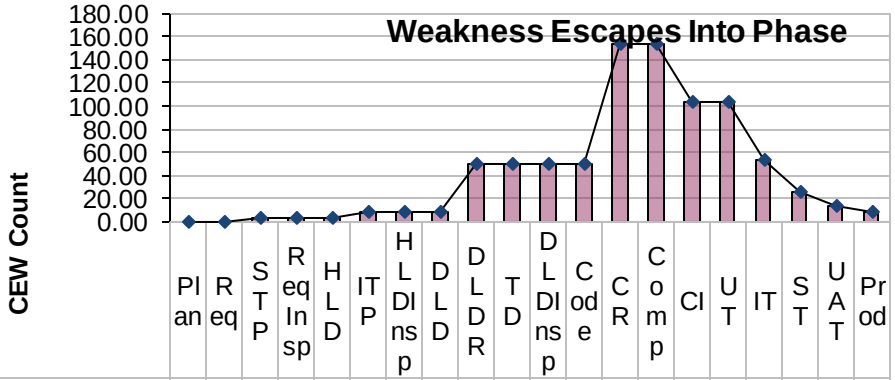
- **motivational versus information measurement**

Deming strongly opposed performance measurement, merit ratings, management by objectives, etc.



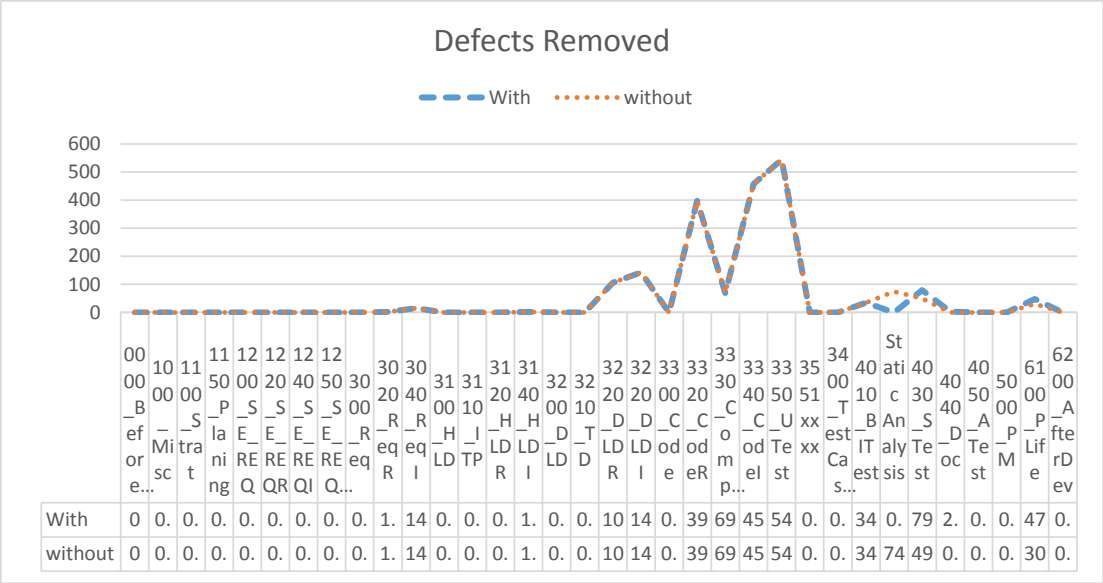
Open and Closed Vulnerabilities



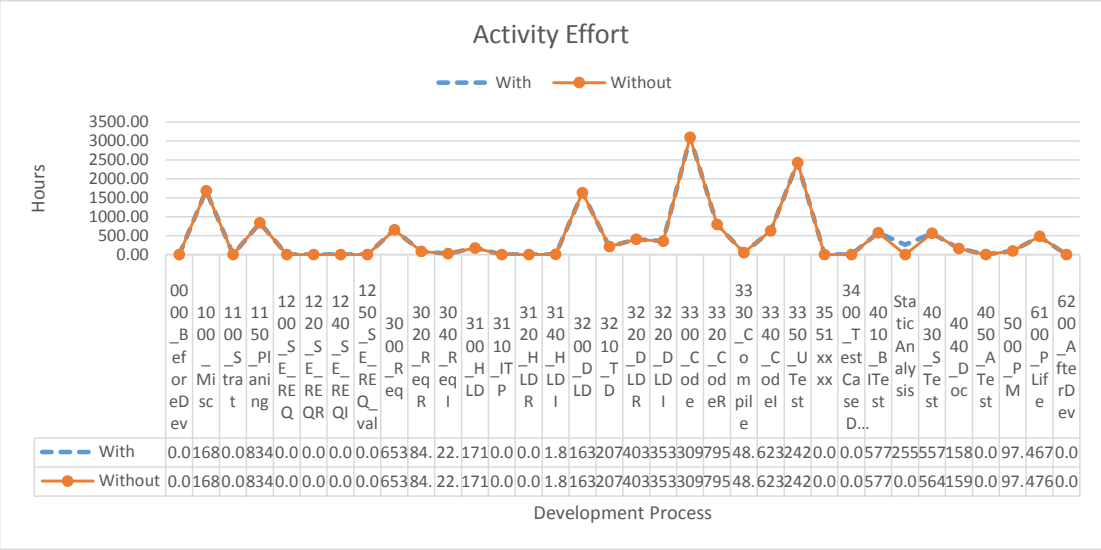


	Plan	Req	STP	Req Insp	HLD	ITP	HLD Insp	DLD	DLDR	TD	DLD Insp	Code	CR	Comp	CI	UT	IT	ST	UAT	Prod
Baseline Defects Entered	0.0	0.0	2.6	2.6	2.6	7.8	7.8	7.8	49.4	49.4	49.4	49.4	153	153	103	103	52.2	26.1	13.0	8.5
Defects Entry	0.0	0.0	2.6	2.6	2.6	7.8	7.8	7.8	49.4	49.4	49.4	49.4	153	153	103	103	52.2	26.1	13.0	8.5

Effort Cost With and Without Weakness Analysis



	0000_Before...	1000_Misc	1100_Strat	1150_Planning	1200_SE_REQ	1220_SE_REQR	1240_SE_REQI	1250_SE_REQ...	3000_Req	3020_ReqR	3040_ReqI	3100_HLD	3110_ITP	3120_HLDR	3140_HLDI	3200_DLD	3210_TD	3220_DLDR	3220_DLDI	3300_Code	3320_CodeR	3330_Comp...	3340_CodeI	3350_UTest	3551xxxx	4000_TestCas...	4010_BITest	Static Analysis	4030_STest	4040_Doc	4050_ATest	5000_PM	6100_PLife	6200_AfterDev
With	0	0.	0.	0.	0.	0.	0.	0.	0.	1.	14	0.	0.	0.	1.	0.	0.	10	14	0.	39	69	45	54	0.	0.	34	0.	79	2.	0.	0.	47	0.
without	0	0.	0.	0.	0.	0.	0.	0.	0.	1.	14	0.	0.	0.	1.	0.	0.	10	14	0.	39	69	45	54	0.	0.	34	74	49	0.	0.	0.	30	0.



Evidence for Software Code Analysis Effectivness

Composing Effective Software Security Assurance Workflows

Empirical studies of 3 substantial development efforts suggest SCA effective.

Vulnerabilities are somewhat rare (estimated at 1%-5% of defects) few modest size components have multiple vulnerabilities.

These tools are likely to find only a fraction of these.

A rising tide lifts all boats. SCA is a cost effective way to improve quality and security.

Static Code Analysis (SCA) and Static Binary Analysis are **cost effective tools.**

- Using **SCA** will both reduce escaped defects, **and** shorten development time to modestly reduce development costs.
- Static Binary Analysis (**SBA**) will take slightly longer but improve quality.
- It is **cost effective** to integrate SCA / SBA into your development and builds

These tools are not silver bullets, but nonetheless, bullets.