SYNOPSYS®

Silicon to Software™

Software Security Initiative – The Basics

And how to measure your success

Eli Erlikhman, Managing Principal

October 15, 2019

Software Security – What Comes to Mind First?

Is Software Security?

- Penetration Testing
- Static Analysis / Code Reviews
- Testing
- Patching
- Fixing Bugs

- This is not software security
- This is bug squashing
- Will it ever stop?

So What is Software Security?

Famous Hacks Through a Different Lens

- Deployment Hardening
- Security Architecture

- OSS Governance
- Vulnerability Management

- Vendor Management
- Secure SDLC

About Synopsys Software Integrity Group

- 1,500+ people organization focused on software security solutions
- Acquired Cigital in 2016
- Published software security book in 1998?
- Preaching “shift left” from 2008
- Measured 100s of software security programs
- Built successful programs for some of the largest companies in the world

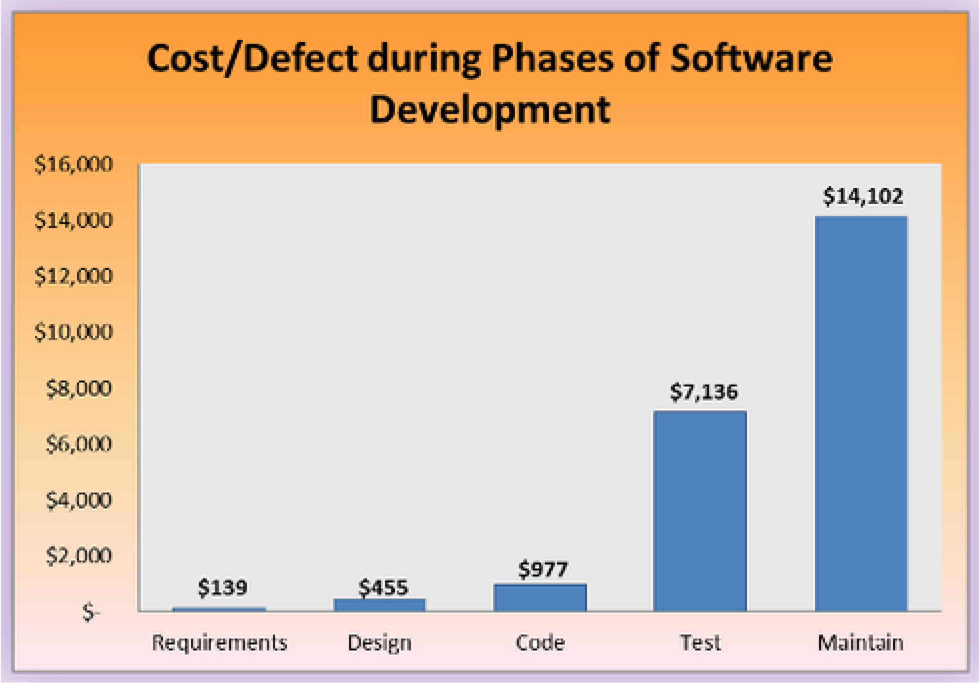
Presenter

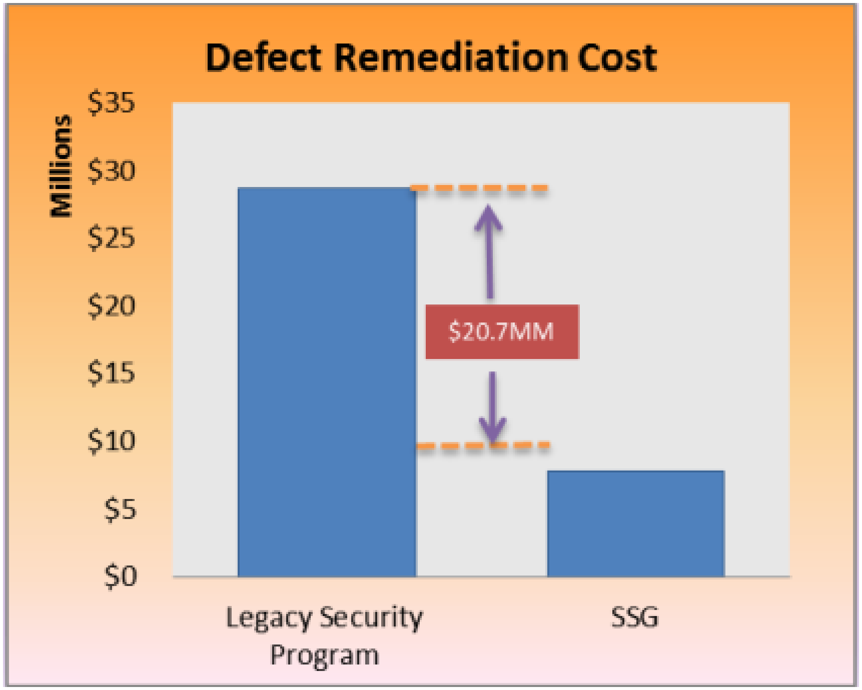
- Managing Principal for Canada
- Former embedded developer. Go pointer arithmetics, jump tables, and bit manipulation!
- Certified BSIMM expert
- Spends weekends at swimming meets

The case for software security program – Our Client's Experience

Based on real life data as provided by our client

$21M in cost savings





https://www.darkreading.com/perimeter/the-economics-of-software-security-what-car-makers-can-teach-enterprises-/a/d-id/1329083

So What is a Software Security Initiative (SSI)?

- The term Software Security Initiative (SSI) describes a firm’s end-to-end effort to organize and coordinate their effort of improving the security of their software (e.g. from requirements definition through operations and incident handling).

- The single most important activity than a firm does in software is formalizing their efforts to organize, execute and evolve an SSI.

- To have an SSI you need to build enterprise-wide capabilities across:

People	Empower people, not just to developers, to do software security.
Governance & Processes	Define why, what, and how. Don’t forget to measure your KPIs and KRIs.
Verification & Defect Discovery	Verify that you are doing the right things. Look for defects, not just bugs.

Common SSI Problems – What We See In The Wild

You don't actually have an SSI

- We've seen organizations with dozens of software security activities in motion without anyone connecting the dots, establishing consistent process, cementing policy, and integrating into development life-cycle

Activity engagement processes make life-cycle impossible

- Some organizations have built years and years worth of engagement processes, workflows, and team charters that focus on justifying the groups existence vs solving the greater problem
- When these activity engagement models are laced into your overall SDLC, execution from the development teams becomes infeasible

Common SSI Problems – What We See In The Wild

Bolted on security programs

- Software security programs extrinsic to development processes are ineffective
- Chasing development around with your security program DOES NOT WORK
- Agile is not an excuse, a lot of security integrations work better in Agile methodologies. All methodologies have secure variants

Missing metrics, or ridiculous metrics

- A smiley face or green light tells you anything about your security program
- A lot of metrics are over-composed, flawed, and not indicative of program adoption or performance (let alone risk)
- Unable to determine where the best money spent is, because you can't measure impact on your risk

Common SSI Problems – What We See In The Wild

The wrong kind of pentesting

- Compliance driven pentesting program
- Pentesting is an important part of the program, it’s not a program
- Non-security people think they’re magic
  - Misconception: Find all issues
  - Misconception: easy to execute
- In many cases developers tries to defeat the example exploit vs fix the software the right way
- Trend to do “blackbox” testing to “simulate attackers”; reality this approach just reduces testing effectiveness

Static program built on top of immature build process

- Most static solutions require software build (there are some exceptions that trade coverage for simplicity)
- One-off integration of a static technology into a bunch of diverse and immature build processes is difficult work, costly, and scales poorly
- Many tools are not suitable for IDE integration

Common SSI Problems – What We See In The Wild

Over emphasis on bugs vs flaws

- Bug – mistake during software development. Flaw – mistake during design and architecture phase
- Flaws are at least half of the problem
- Flaws are harder to find and harder to fix after implementation
- Finding flaws is more difficult

Superficial remediation process

- “We fixed it.” vs “We fixed all occurrences of the issue and improved the process to avoid that issue in the future.”
- Most results from pentest & static are fixed individually
- A found bug is worth MORE to us than we’re usually getting out of it

Common SSI Problems – What We See In The Wild

Threat intelligence Vacuums

- Tell us about your threat intelligence operations:
  - “Our Threat Intel team gets data from a wide array of commercial sources, private industry alliances, and law enforcement partnerships.”
- How do you pass on the relevant data to your software developers?
  - “Why do they need to know?”

You’ve embraced the tech stack rainbow

- “Our shadow IT guy Tim says he can write the app in a week using perl/cgi.”
- Business moves quickly, and to keep up with the pace of invocation you have embraced a lot of 1 time use tech stacks to “Get the job done.”
- Each tech stack brings it’s own set of security issues and amplifies the amount of work that needs to be done to secure it

Building Successful SSI – SSI Cultures

- Two main cultures to building SSI
  - SSI was purposely started in a central corporate group (e.g., under a CISO) and focused on compliance, testing, and risk management (Governance-led culture)
  - SSI was started by engineering leadership (e.g., senior application architects), but the organization rather quickly created a centralized group (e.g., under a CTO) to set some development process, create and manage security standards, and ensure the silos of engineering, testing, and operations are aware of and adhere to security expectations (Engineering-led culture)

- Different SSI cultures require different approaches to building SSIs

- In both cases, building SSI has 3 main phases:
  - Emerging
  - Maturing
  - Optimizing

Building Successful SSI – Getting Started in Governance-led culture

1. **Leadership** - Put someone in charge of software security, and provide the resources he or she will need to succeed

2. **Inventory software** - Know what you have, where it is, and when it changes

3. **Select in-scope software** - Decide what you're going to focus on first

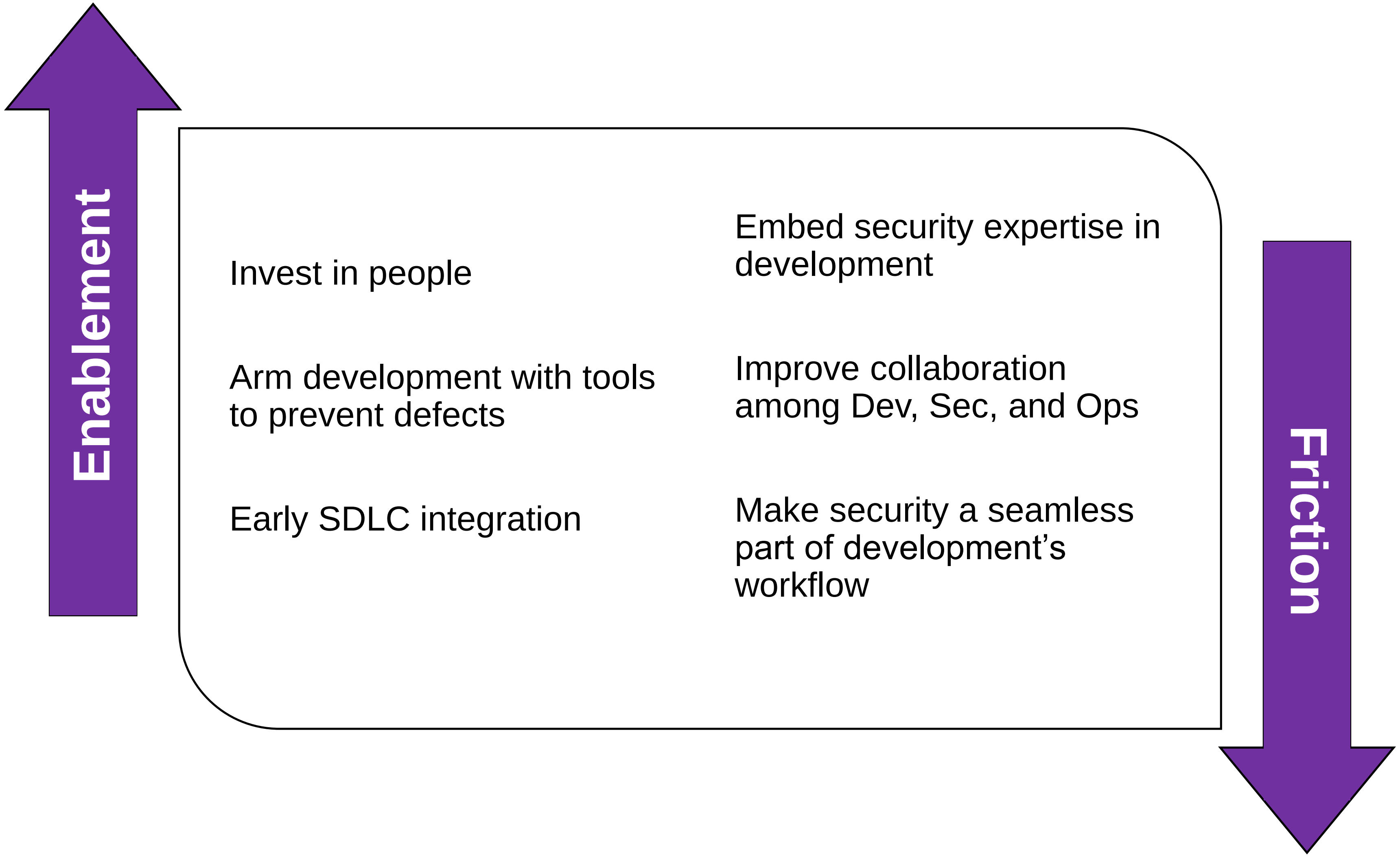
4. **Ensure host and network security basic** - Don't put good software on bad systems or in poorly constructed networks (cloud or otherwise)

5. **Do defect discovery** - Determine the issues in today's production software and plan for tomorrow

6. **Select security controls** - Start with controls that establish some risk management to prevent recurrence of issues you're seeing today

7. **Repeat** - Expand the team, improve the inventory, automate the basics, do more prevention, and then repeat again

Building Successful SSI - Two Levers to Transform Software Security Culture



Invest in people

Arm development with tools to prevent defects

Early SDLC integration

Embed security expertise in development

Improve collaboration among Dev, Sec, and Ops

Make security a seamless part of development’s workflow

Building Successful SS – Measure your success

I often say that 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 meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of *science*, whatever the matter may be. –Lord Kelvin

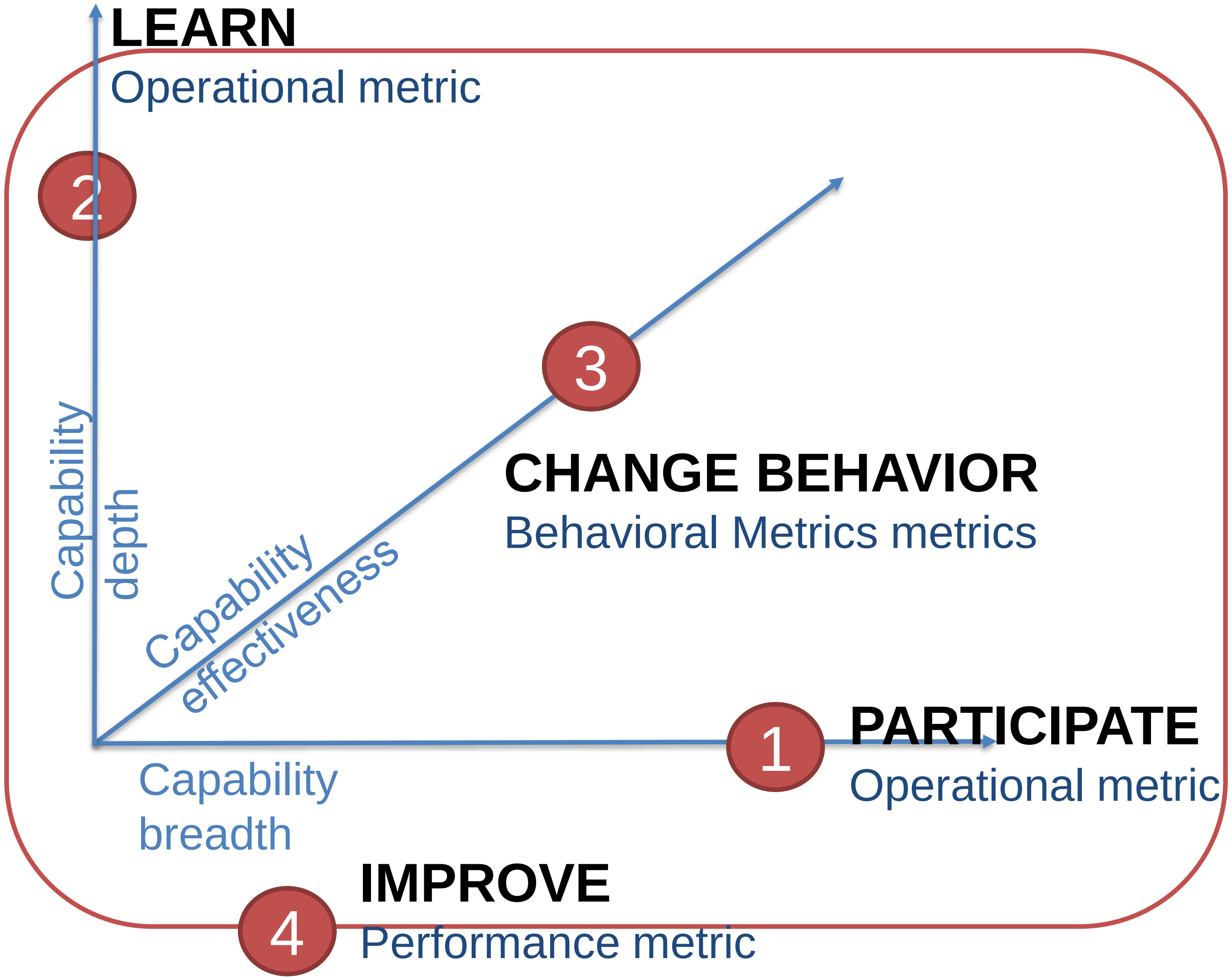
Critical Characteristics of a Metric

- Each metric must answer a question from business or risk or security or some other stakeholder

- The question must be in response to a goal that one of the stakeholders is trying to achieve

- If you don't know what kind of trends you want this metric to exhibit over time, than you need rethink metric's question and goal

Building Successful SSI - What Kind of Metrics do You Need?



- Use operational and behavioral metrics to measure each capability coverage and its impact on stakeholders behavior
- Use performance metrics to measure overall impact on software security posture of the organization

Metrics Program must

“create feedback loop to SSDL enhancement”

Otherwise, what’s the point?