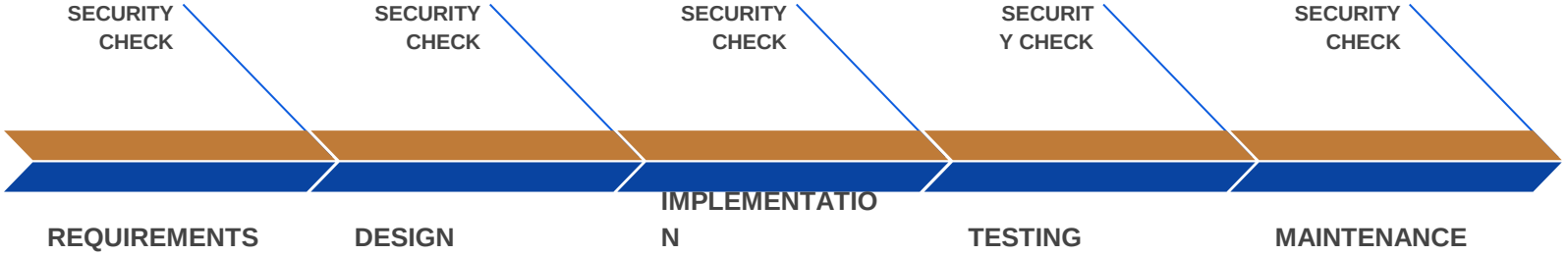
92 Years to DevOps

A Motorola Solutions Story

Adam Lewis
Chief Security Architect

MOTOROLA SOLUTIONS

In the Beginning ...



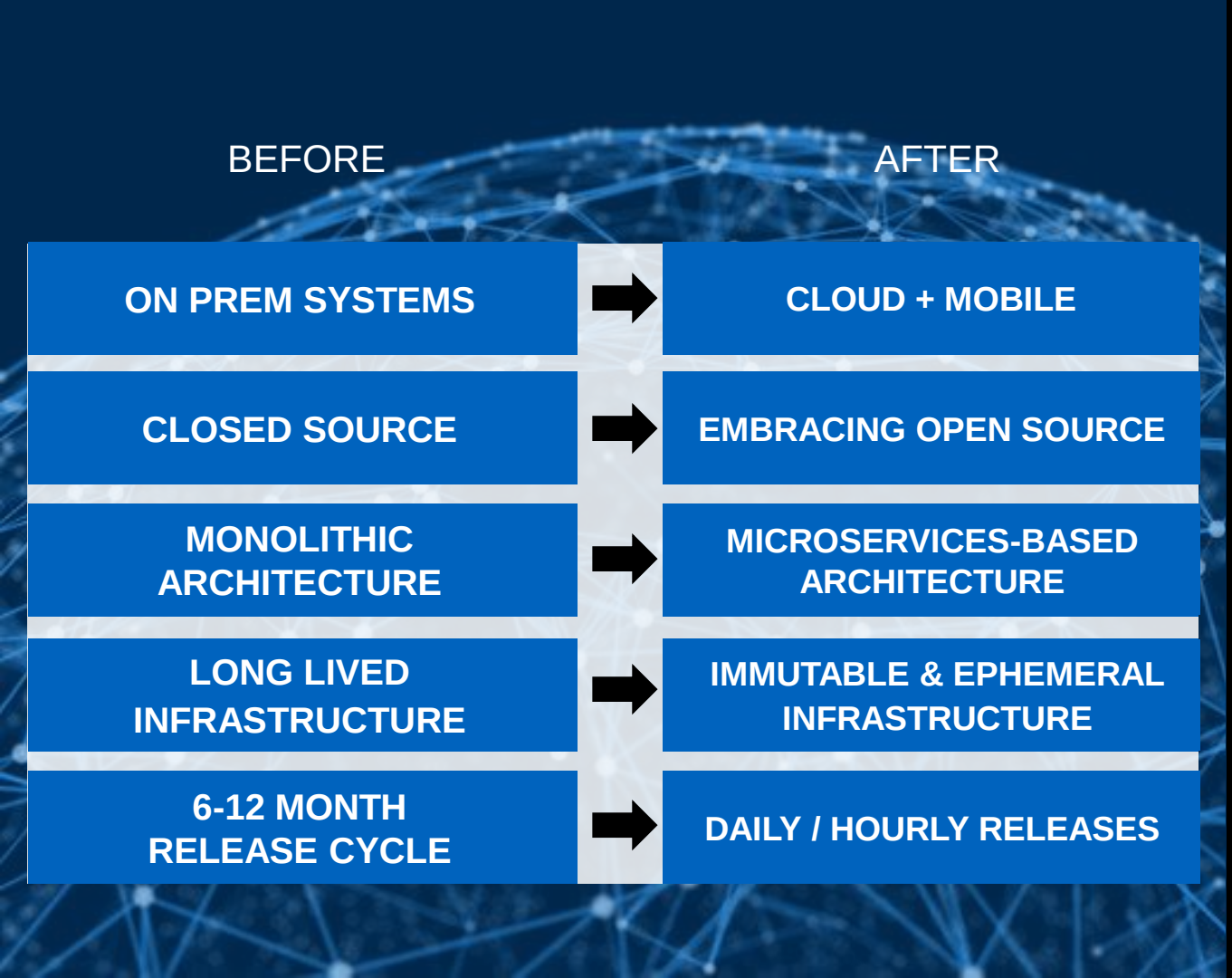
DEVELOPMENT

- Waterfall-style development approaches
- Monolithic applications & systems
- Infrequent product releases: 6-12 months
- Checkpoints / gates between phases

CYBERSECURITY

- Manual security checkpoints at each phase
- One time assessments
- Exclusive access to security tools
- Sign off before release

Then Something Happened.



Not Your Father's Oldsmobile ...

Security needed to change.

(and so did development)

The Great Chicago Fire of 1871

Lessons Learned and applicability to security in the age of DevOps

CULTURE SHIFT IN HOW WE APPROACH SECURITY

The Businesses

- Development teams = first line of defense
- Business owns the risk (at the right level)

Cybersecurity Organization

- Cyber org = second line of defense
- Enable business owners to make risk-based decisions for their products & customers
- Enable developers to own the security of their code

Hiring more security professionals is *not* the solution

DEVELOPER

- Security Champion program
- One champ per scrum team
- Focused training on Threat modelling, secure coding
- Access to security scanning tools for quick feedback
- Make it fun! - devs embrace it -> easiest change but requires investment

PRODUCT MANAGER

Quantify risk impact
- Product & customer context

Risk-based backlog prioritization
- Draw the "line" to recommend what should be fixed

Trained in security processes
- Pre - & post-release security
- Incident response & recovery

BUSINESS OWNER

Leadership support
- Drive cyber culture from the top
- Hold product teams accountable

Accountability
- Risk owner of severe product & business risks
- Accept known risks and treatment plans

Trained in security processes
- Crisis management

NICE
NATIONAL INITIATIVE FOR CYBERSECURITY EDUCATION

CYBERSECURITY TEAM

POLICY | TRAINING | ENABLEMENT

From the team of ‘no’ to the team of ‘how’

- Empower developers to build secure code
- Empower product owners to prioritize their risk management activities
- Empower business owners to accept risk (at the right level)
- Provide Training & Awareness
- Provide Policy, Standards, Guidelines, and Best Practices
- Provide Threat Intelligence, Pen tests, AppSec toolchain

Learn to Let go

- Recognize what can be taught or delegated
- Remove ourselves from critical paths
- Leave business decisions to the risk owners (NO SIGN-OFF)

Pinned Tweet

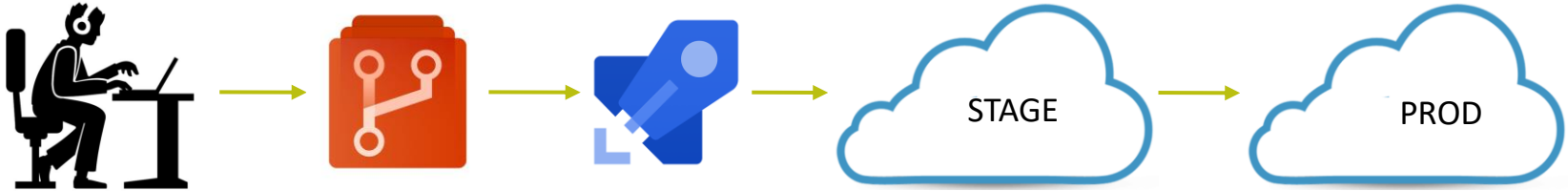
Jim Manico @manicode · Dec 18, 2017

From my experience all software developers are now security engineers wether they know it, admit to it or do it. Your code is now the security of the org you work for. #GoldenAgeOfDefense

20 209 403

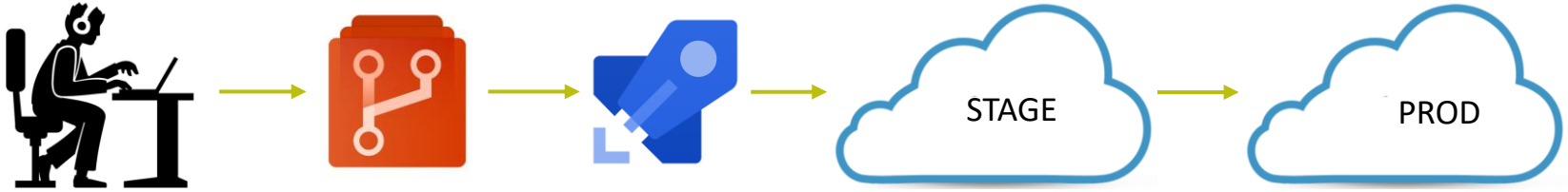
INTEGRATED SECURITY

CONTINUOUS DEVELOPMENT + CONTINUOUS SECURITY



INTEGRATED SECURITY

CONTINUOUS DEVELOPMENT + CONTINUOUS SECURITY

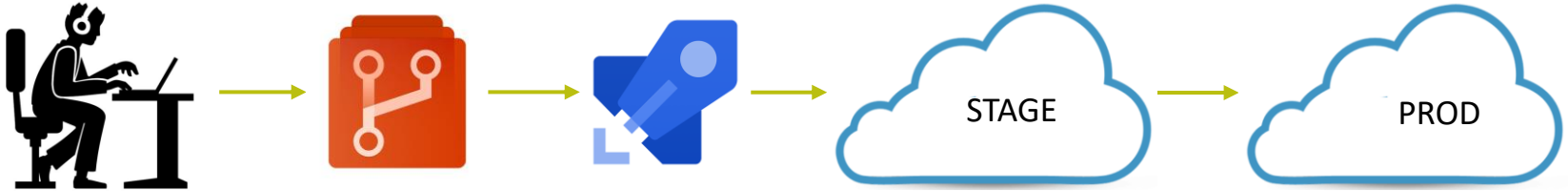


Security activities before code is checked into version control

DEVELOPER TRAINING
THREAT MODELING
SECURITY AND PRIVACY STORIES
IDE SECURITY PLUGINS
PRE-COMMIT SECURITY HOOKS
SECURE CODING STANDARDS

INTEGRATED SECURITY

CONTINUOUS DEVELOPMENT + CONTINUOUS SECURITY



Security activities before code is checked into version control

DEVELOPER TRAINING
THREAT MODELING
SECURITY AND PRIVACY STORIES
IDE SECURITY PLUGINS
PRE-COMMIT SECURITY HOOKS
SECURE CODING STANDARDS

Fast, automated security checks during the build and Continuous Integration steps.

STATIC CODE ANALYSIS
SECURITY UNIT TESTS
SOFTWARE COMPOSITION ANALYSIS
CONTAINER SECURITY
CONTAINER HARDENING
CONTAINER IMAGE SIGNING

JIRA

STATIC CODE ANALYSIS

CONTINUOUS DEVELOPMENT + CONTINUOUS SECURITY

- White box testing (has full access to code)
- Looks for weaknesses in proprietary code (such as SQLi) that could (potentially) result in an exploitable vulnerability
- Finds bugs earlier in the SDLC where they are cheaper to fix
- Lacks ability to discover runtime and environment issues

😢 It's still not fast enough, and needs to catch up to the velocity of development

😢 Can produce too many findings to be actionable, and also false positives

😢

```
var config = require('../config.json');
var mysql = require('mysql');
var pool = mysql.createPool(config.connectionString);
var uuid = require('uuid');

module.exports = {
  addPoll: function(id, question, file, name, email, callback) {
    pool.getConnection(function(err, connection) {
      connection.query('INSERT INTO quiz (_id, question, file,
          CreatorName, CreatorEmail, admin, YesCount, NoCount)
          values (?, ?, ?, ?, ?, ?, ?, ?)', [id, question, file,
          name, email, uuid(), 0, 0], function(err, rows) {
        connection.release();
        err ? callback(err) : callback(null, rows);
        return
      });
    });
  },
  getPoll: function(_id, callback) {
    pool.getConnection(function(err, connection) {
      if (_id) {
        connection.query('SELECT * FROM quiz WHERE _id =
            "'+_id+'"', function(err, rows) {
          connection.release();
          err ? callback(err) : callback(null, rows);
          return
        });
      } else {
        connection.query('SELECT * FROM quiz', function(err,
            rows) {
          connection.release();
          err ? callback(err) : callback(null, rows);
          return
        });
      }
    });
  },
  removePoll: function(_id, callback) {
    pool.getConnection(function(err, connection) {
      connection.query('DELETE FROM quiz WHERE _id = ?', [_i
          function(err, rows) {
        connection.release();
```

Open Source

It’s a Supply Chain Issue - What’s in YOUR code?

80%

The average percentage of an application comprised of open source software components

40m+

Developers on GitHub, including 10m new users in 2019. Over 44m new code repos created in 2019.

1,568

The number of open source components pulled in to create a “Hello World” program in a popular web framework (1.1m lines of code)

SOFTWARE COMPOSITION ANALYSIS

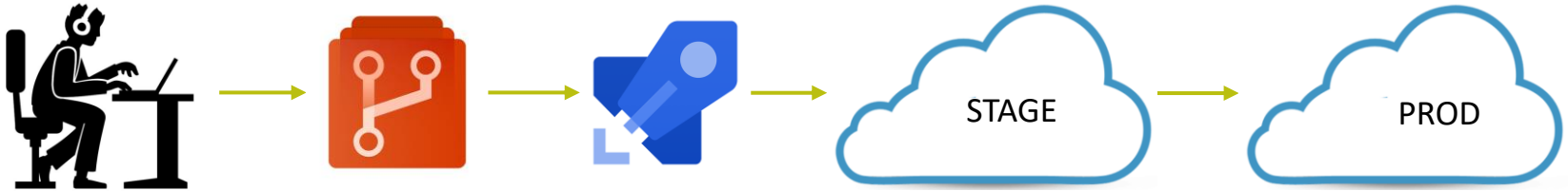
WHAT'S IN YOUR CODE?

- Enumerates the open source packages used by code and builds a Software Bill of Materials (SBOM)
- Cross-references the inventory in the SBOM with KNOWN vulnerabilities from the Mitre National Vulnerability Database (NVD)
- Also produces licensing findings (addresses the legal risk of OSS)

Software Component Transparency

INTEGRATED SECURITY

CONTINUOUS DEVELOPMENT + CONTINUOUS SECURITY



Security activities before code is checked into version control

DEVELOPER TRAINING
THREAT MODELING
SECURITY AND PRIVACY STORIES
IDE SECURITY PLUGINS
PRE-COMMIT SECURITY HOOKS
SECURE CODING STANDARDS

Fast, automated security checks during the build and Continuous Integration steps.

STATIC CODE ANALYSIS
SECURITY UNIT TESTS
SOFTWARE COMPOSITION ANALYSIS

CONTAINER SECURITY
CONTAINER HARDENING
CONTAINER IMAGE SIGNING

JIRA

Automated security acceptance, function testing, and deep out-of-band scanning during Continuous Delivery.

SECURITY SCANNING (DAST)

JIRA

DYNAMIC SECURITY SCANNING

ATTACKING YOUR CODE

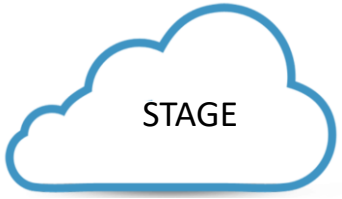
- Operates on running code
- Black box testing (outside-in), has no knowledge of the internal source code or frameworks.
- Unlike the others, DAST attempts actual exploits.

- Later in the SDLC and hence more costing to fix.


Limitations: mostly just for web pages. Even web services is lacking :-(

INTEGRATED SECURITY

CONTINUOUS DEVELOPMENT + CONTINUOUS SECURITY



Security activities before code is checked into version control

DEVELOPER TRAINING
THREAT MODELING
SECURITY AND PRIVACY STORIES
IDE SECURITY PLUGINS
PRE-COMMIT SECURITY HOOKS
SECURE CODING STANDARDS

Fast, automated security checks during the build and Continuous Integration steps.

STATIC CODE ANALYSIS
SECURITY UNIT TESTS
SOFTWARE COMPOSITION ANALYSIS

CONTAINER SECURITY
CONTAINER HARDENING
CONTAINER IMAGE SIGNING

JIRA

Automated security acceptance, function testing, and deep out-of-band scanning during Continuous Delivery.

SECURITY SCANNING (DAST)

JIRA

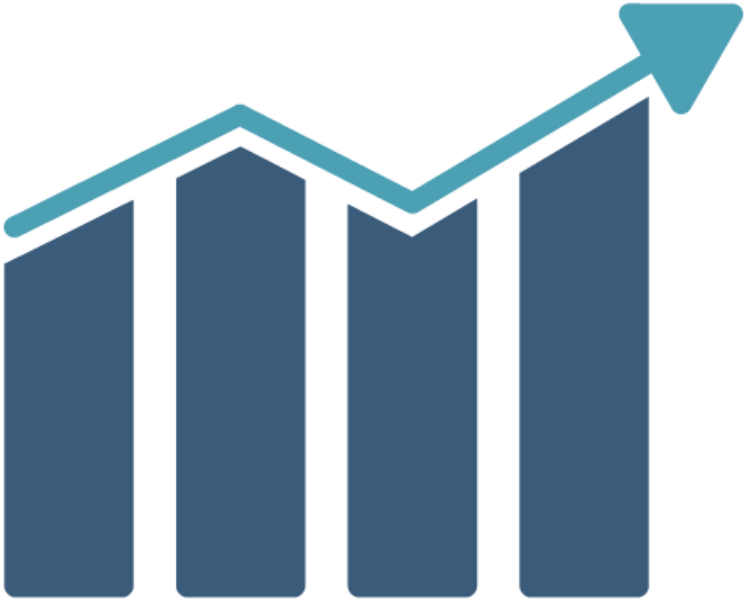
Vulnerability Management

IDENTIFY
TRIAGE
PRIORITIZE
VALIDATE
DEPLOY

JIRA

WHAT GETS MEASURED GETS DONE

- Is the rate at which security defects are introduced into applications trending down?
- Are security defects that are introduced getting caught earlier in the development cycle?
- Are the TOTAL number of security defects in production going down over time?
- What is the average time to remediate?



“Security has a problem with Measurability.” #AppSecDC 2019 security conference

THANK YOU

Adam Lewis
@lewiada

TAKEAWAYS

- **Development teams = first line of defense**
  - Software Engineer = Security Engineer
- **Security team = second line of defense**
  - Enable developers to own the security of their code (and security needs to slot into dev, not the other way around)
  - Enable business owners to make risk-based decisions for their products & customers (at the right level)
- **Pick one or two metrics, get good at them, and then pick a few more**
- **Get senior leadership buy-in!**