SYNOPSYS®
Silicon to Software™

Software Security — Embracing Velocity

Shifting Left in a Circular World

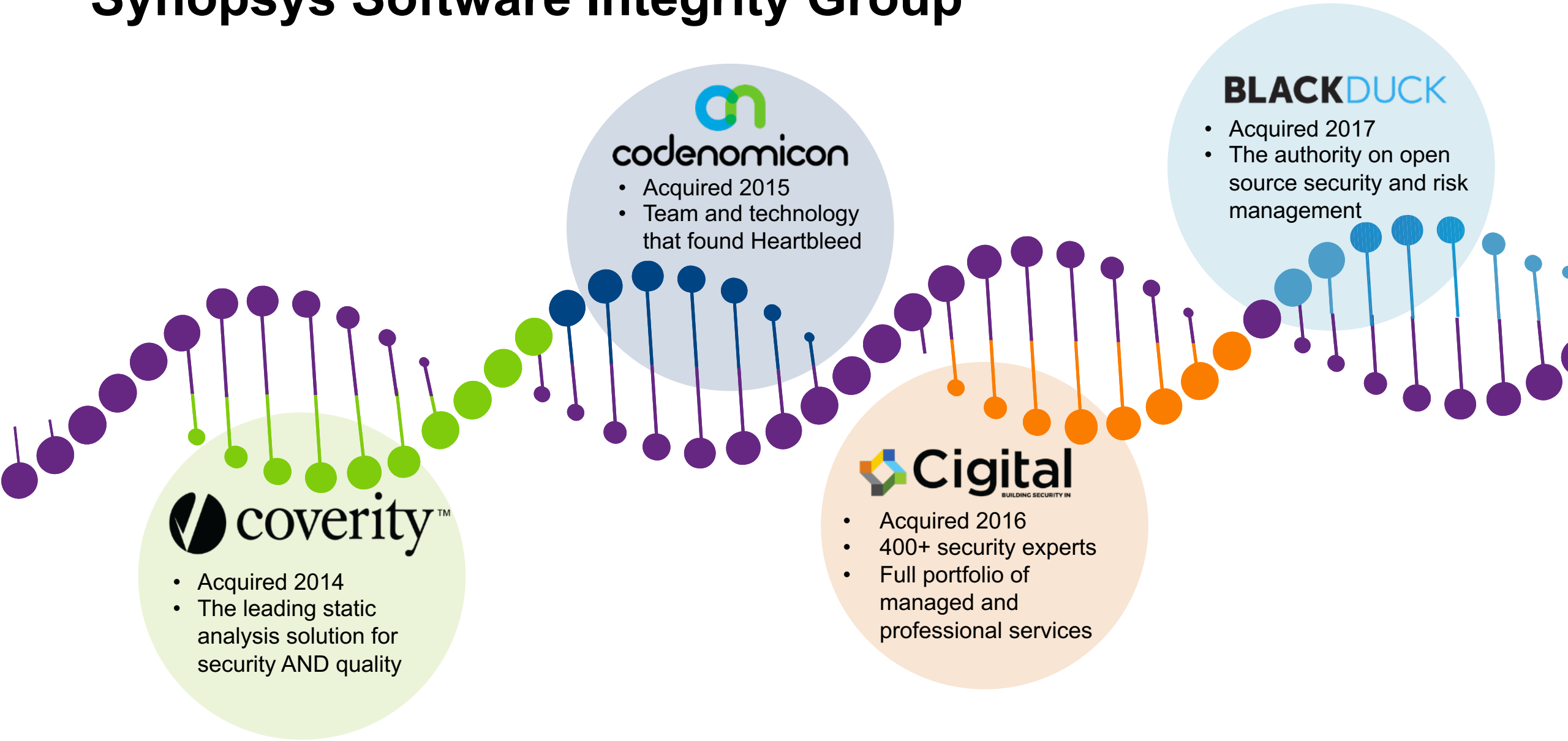
Nick Murison
Head of Software Security Services, Nordics

Introduction

- Head of Synopsys Software Security Services for Nordics
- Consultant ~14 years
  - Penetration testing
  - Incident response
  - Training
  - Software security: code, design, risk
  - Company-level initiatives to improve software security
- BSc in Computer Science
- MSc in Information Security

nick.murison@synopsys.com
@nickmurison

Synopsys Software Integrity Group



codenomicon
- Acquired 2015
- Team and technology that found Heartbleed

BLACKDUCK
- Acquired 2017
- The authority on open source security and risk management

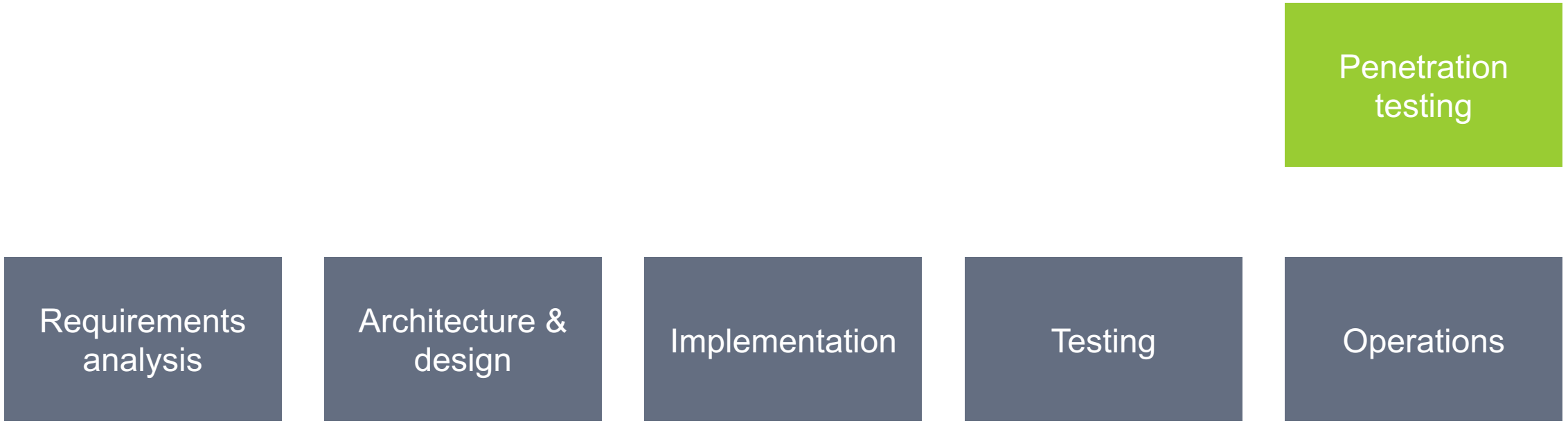
coverity™
- Acquired 2014
- The leading static analysis solution for security AND quality

Cigital
BUILDING SECURITY IN
- Acquired 2016
- 400+ security experts
- Full portfolio of managed and professional services

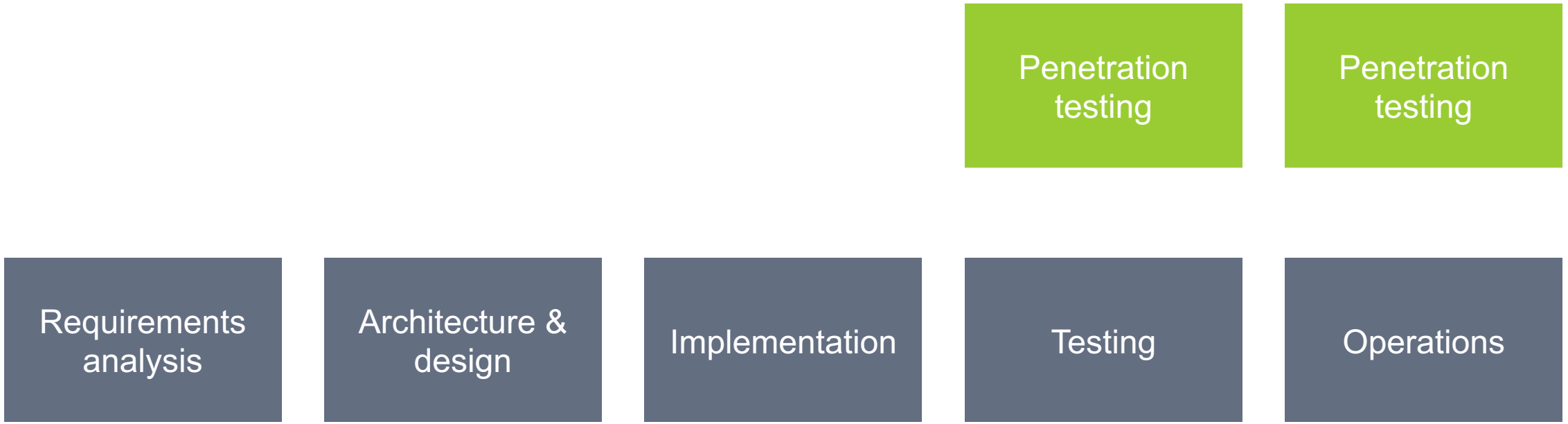
In the beginning…

“Where do we add security?”

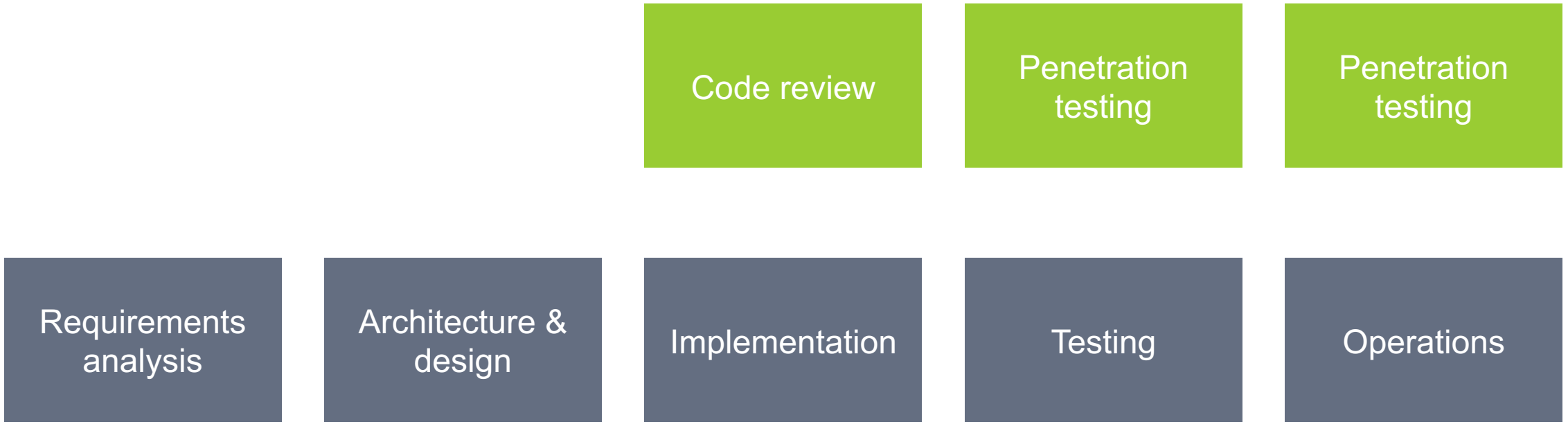
Classic SDLC, classic security



“Let’s do More Security”



“Let’s do Even More Security”



“Let’s do All The Security Things”

Risk analysis Abuse cases Security reqs	Risk analysis Threat modeling	Code review	Risk analysis Penetration testing	Security ops Penetration testing
Requirements analysis	Architecture & design	Implementation	Testing	Operations

Grossly Simplified Archetypes

Risk Focused

- Often mature programmes
- Have the resources to adapt to rapid release development
- Fluent in business risk
- E.g. banks

“DevOps-first”

- Lean
- Focus on feature velocity through high level of automation
- Used to iterating both software and processes
- E.g. start-ups

Engineering Focused

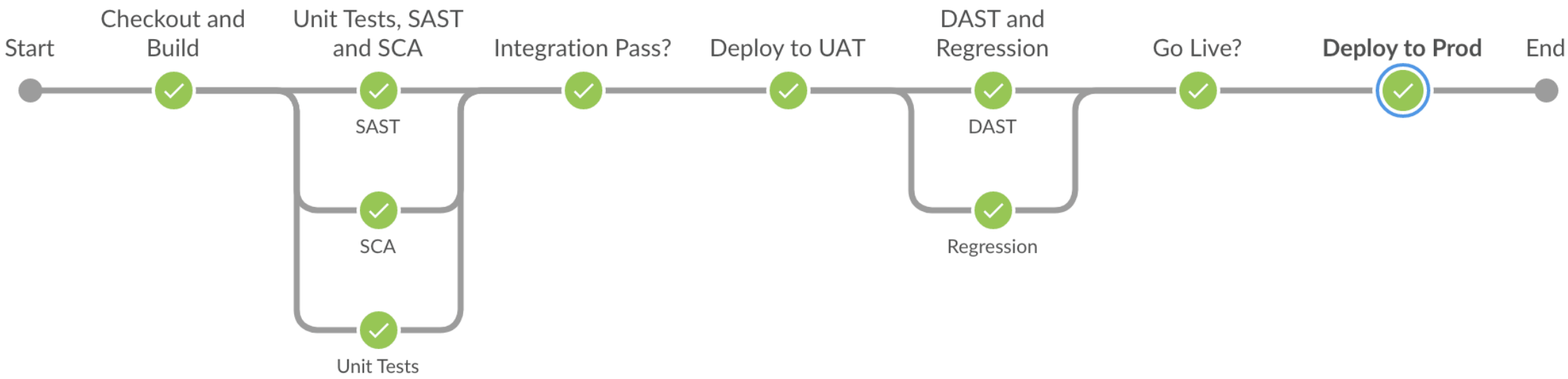
- Often “traditional” approach
- Agile = “doing the same things faster”
- Preference for technology-based solutions to security problems
- E.g. enterprise technology firms

If we're not speaking the **language** of development,

and we're not using tools and processes that **align** with the developers' world,

we're not doing **software security**.

Developers are using Jenkins. Why can’t Security?



SYNOPSYS®

Iterate

Consider

- What security defects are we finding now?
- What is causing developers to deviate from what we want them to do?
- What have we learnt that we can reuse?
  - Abuse cases
  - Reusable design blueprints
  - Microservices that do Security Things

Build better

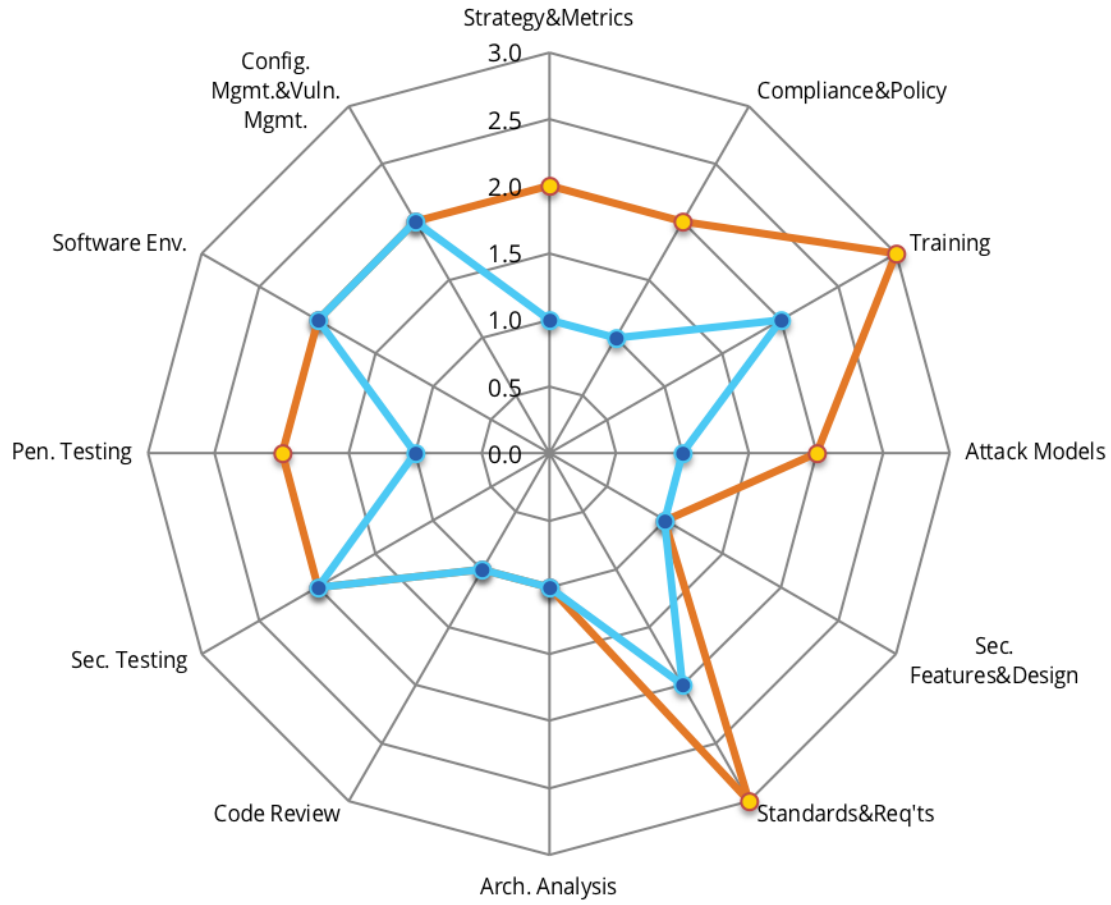
- Training
- Iterative threat models
- Reusable abuse cases
- Security design blueprints
- Security test cases
- Add / replace tools in automation
- Proactive security capabilities / activities
- And more

Measure (and remeasure)

- Use a common measuring stick
- There is no “perfect”
- There is only moving forward

BSIMM

www.bsimm.com



SYNOPSYS®

SYNOPSYS®
Silicon to Software™

Thank You

@nickmurison