RSAConference2019
San Francisco | March 4–8 | Moscone Center

SESSION ID: ASD-T08

Building security processes for developers and DevOps teams.

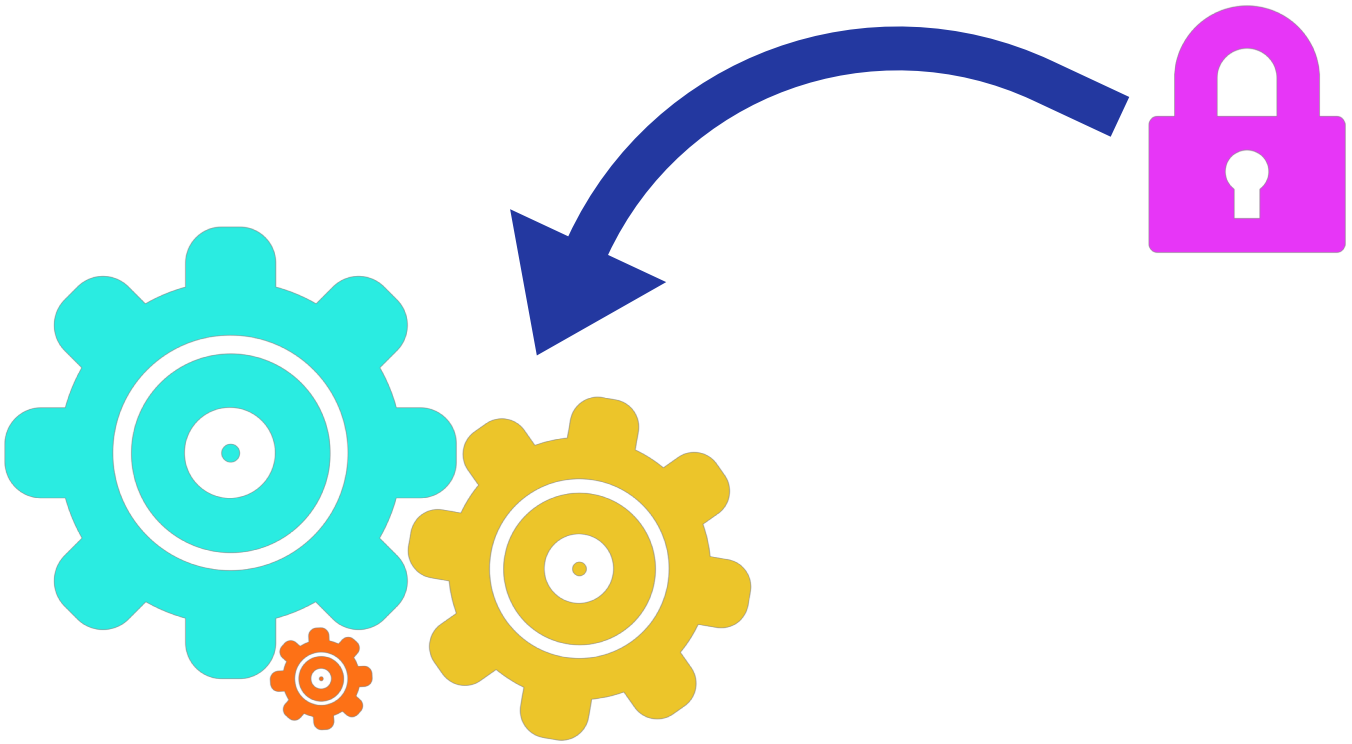
Ainsley K. Braun
Co-founder & CEO
Tinfoil Security, Inc.
@tinfoilsecurity

RSAConference2019

Take away: Automate the easy stuff, focus on the hard stuff

What is Security for DevOps?

- Security for DevOps (DevSec, SecDevOps, and others) is simply adding security into the software creation process



Why Does it Matter?

- Security teams are overwhelmed and can't keep up

Why Does it Matter?

- Development teams are held up, wanting to produce faster

Development

Why Does it Matter?

- Ops teams are working in a silo, security is left to last-minute

Why Does it Matter?

- Teams aren't communicating, and sometimes even feel blocked by one another

Why Does it Matter?

CISO

Losing good talent

Timeline slowed

CIO / CTO

Security

Unable to work on big picture

Frustrated by security team

Engineering & Ops

Why Does it Matter?

CISO

Losing good talent

Timeline slowed

CIO / CTO

Complete development

Reports vulnerabilities

Huh?

Forced to help

repeat

Security

Unable to work on big picture

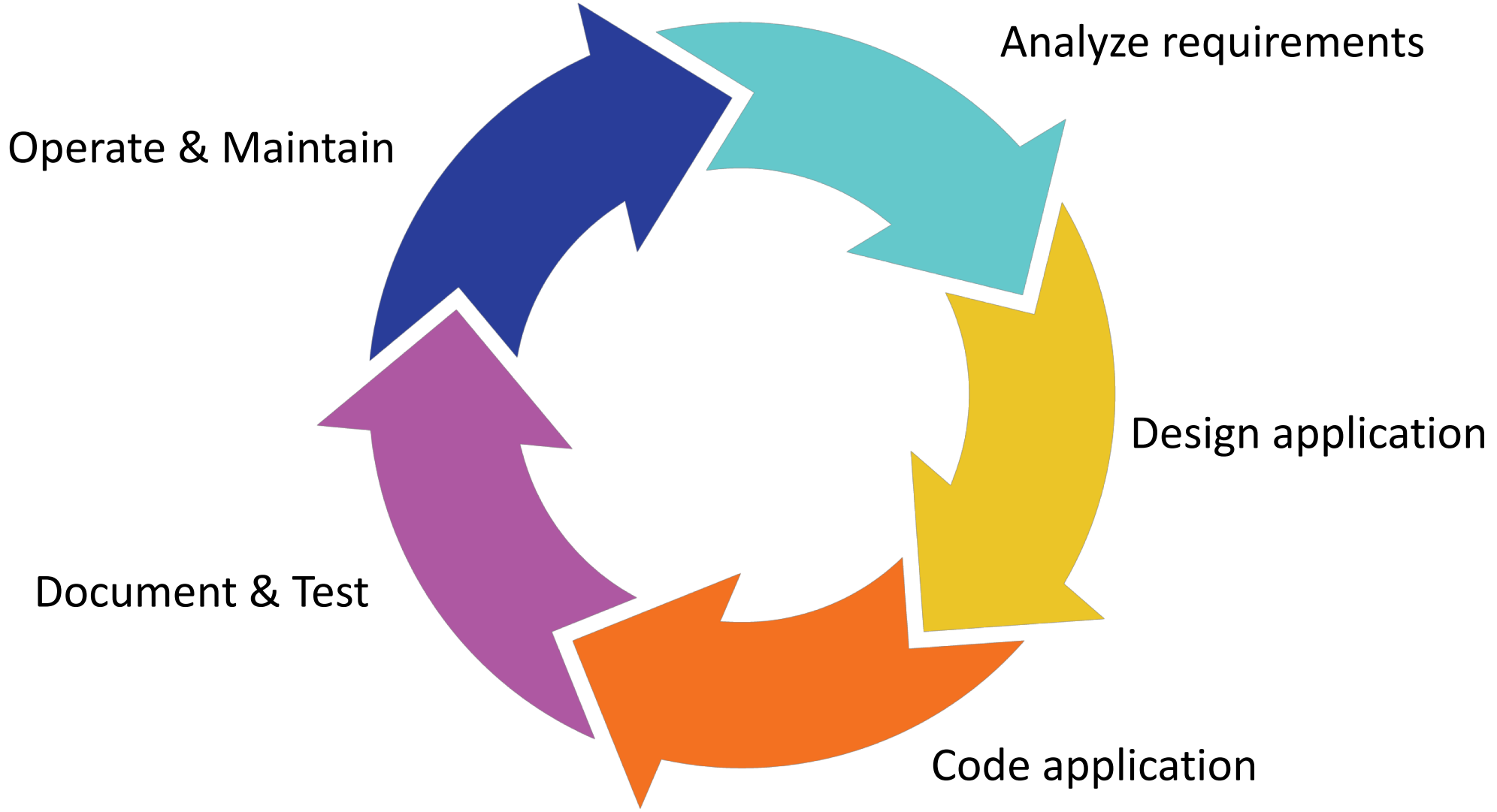
Frustrated by security team

Engineering & Ops

Formalizing Your Software Creation Process

- The SDLC
  - Software / System
  - Development
  - Life
  - Cycle

What is the SDLC?

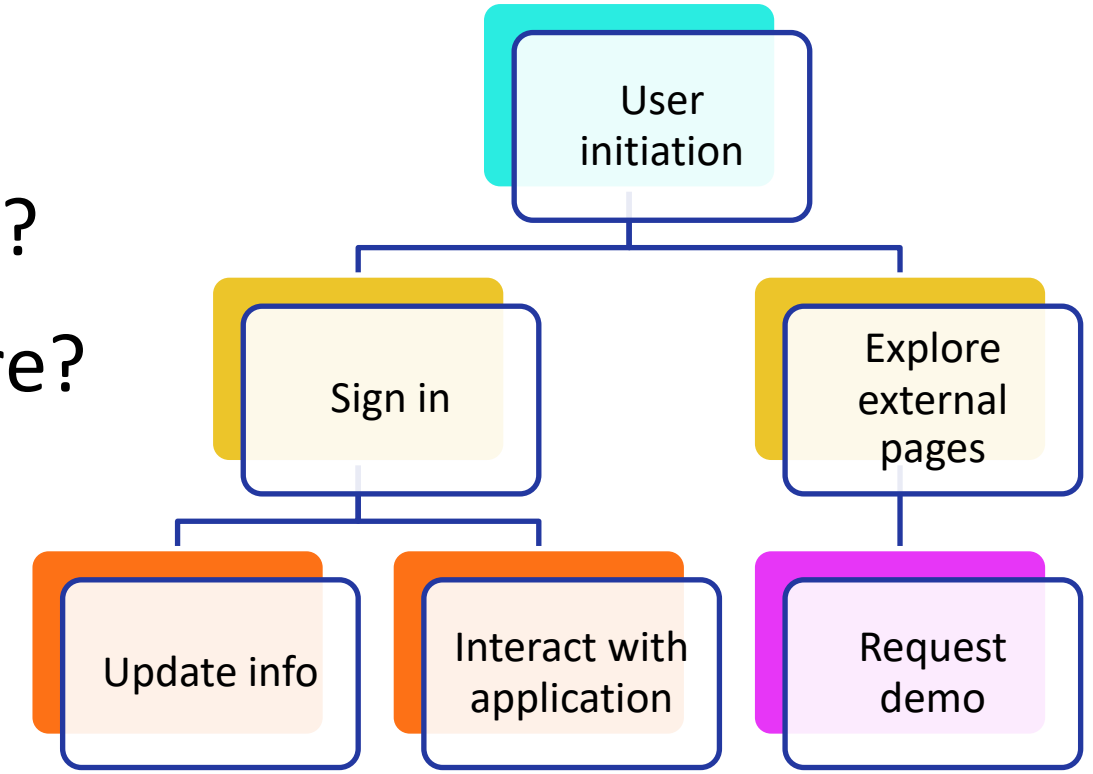


SDLC – Analyze Phase

- Who are the stakeholders and what do they want?
- What should the system be doing?
- Prototype a UI or UX Flow
- Does anything like this already exist?

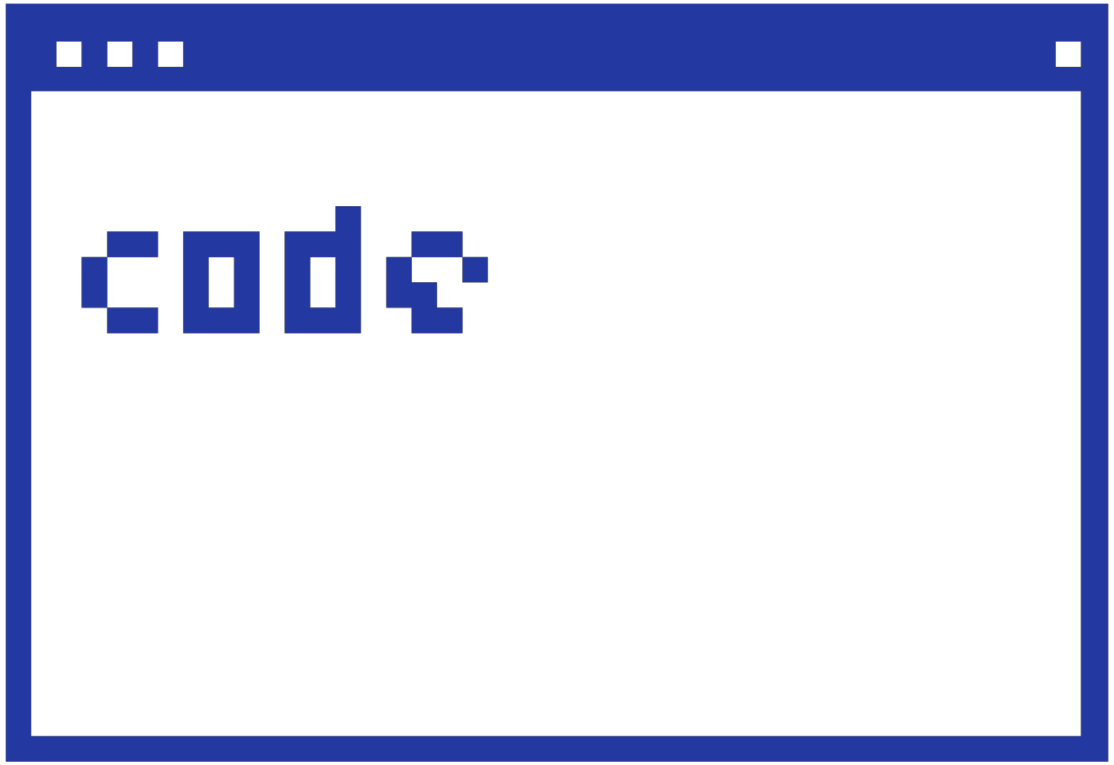
SDLC – Design Phase

- Are there any risks with the system?
- What is the UX / UI Design?
- What is the flow for the application?
- How can it be extended in the future?
- How should it perform?



SDLC – Code Phase

- Writing code for functionality and writing tests



SDLC – Document & Test Phase

- Document how it works
- Training others
- Run your tests

Testing Your Applications

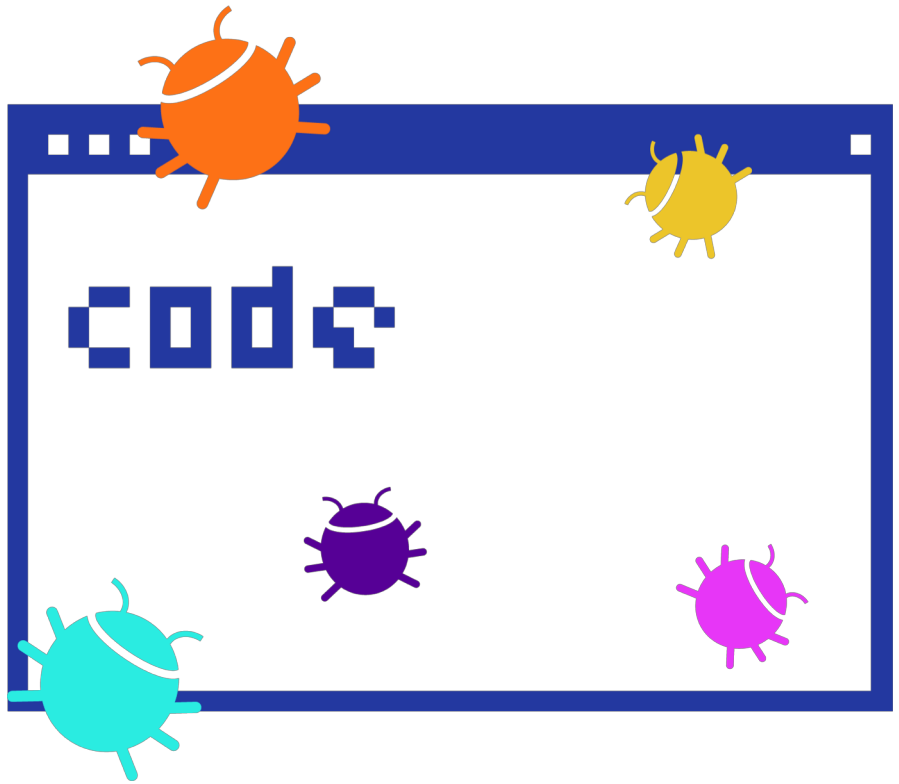
- Integration tests
- Unit tests
- Regression tests

Testing Your Applications

- Integration tests
- Unit tests
- Regression tests
- Security tests ← Starting your security for DevOps

SDLC – Operate & Maintain Phase

- Application is now running in production
- Fix any new bugs
- Discuss retrospectives
  - How did it go?
  - What could be changed for next time?

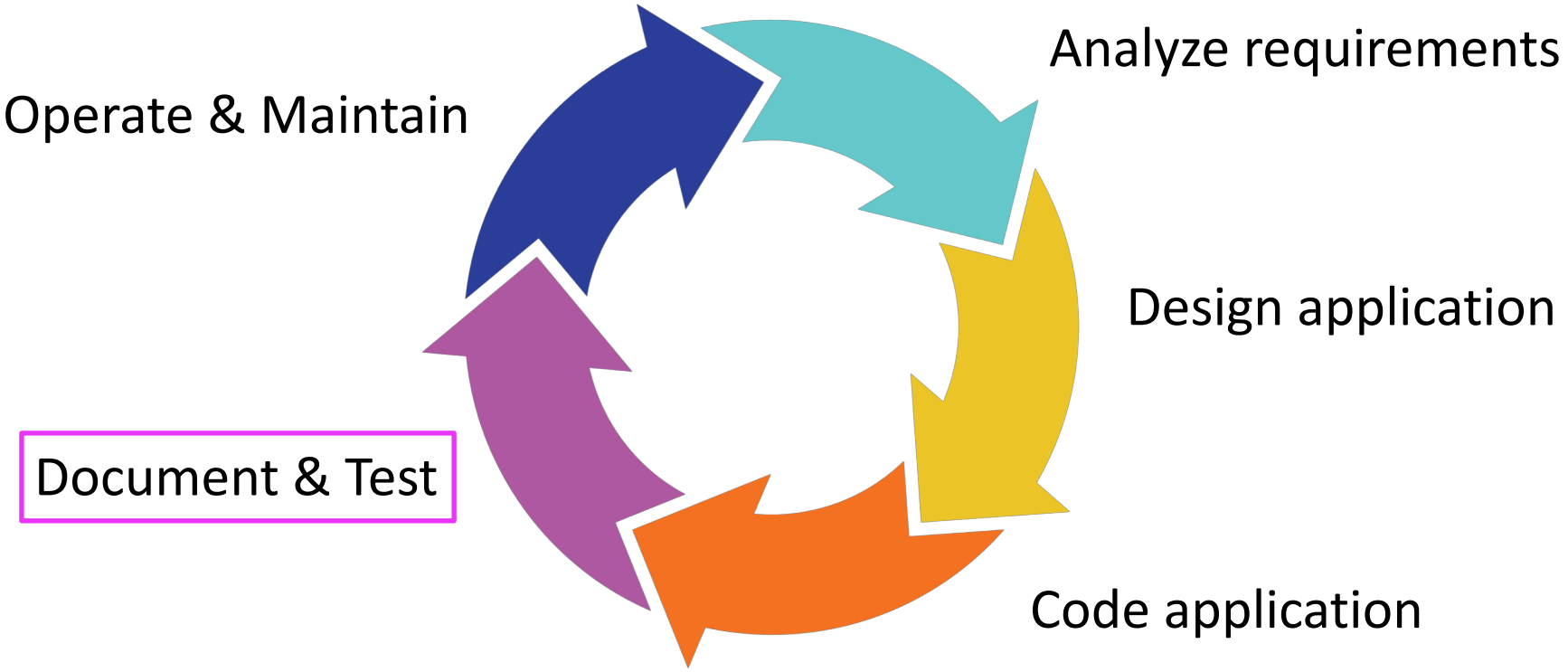


SDLC Methodologies

- Waterfall
- Prototype
- Agile
- Iterative
- V Model
- Incremental
- Rapid Application Development RAD
- Spiral

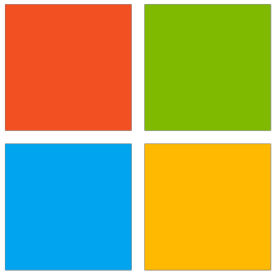
Comparing Methodologies

- Methodologies are common in one way: the testing phase
  - This is where security ideally enters the development process



Where did Security for DevOps Begin?

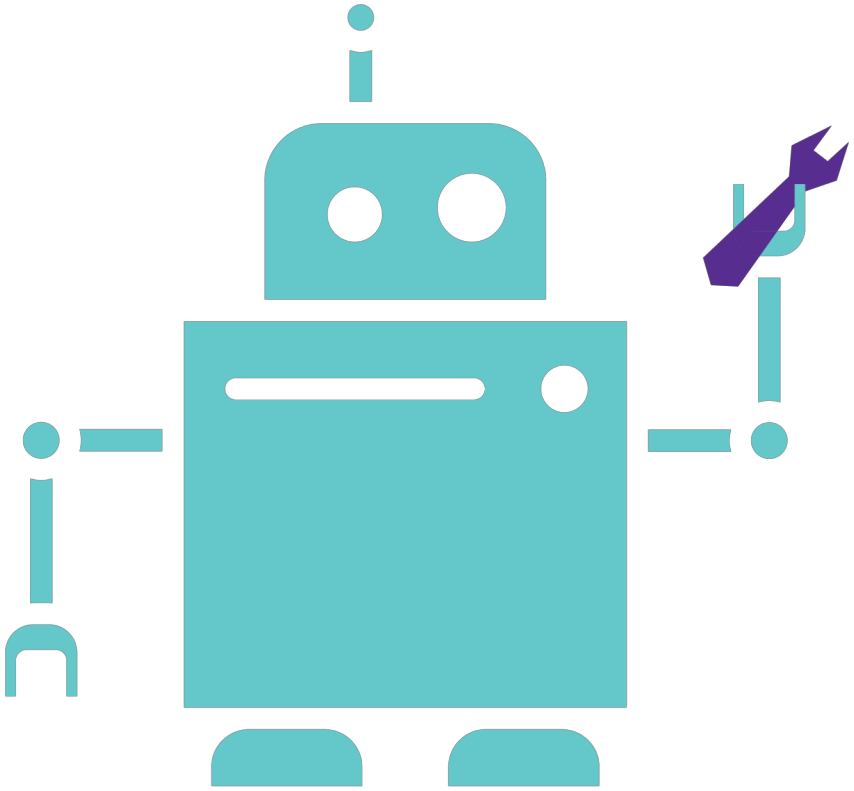
- Microsoft Secure Development Lifecycle
  - A more formal approach to the SDLC with Security in mind
  - Microsoft learned the value of thinking about security early
  - Threat Modeling: can begin earlier than the testing phase



Microsoft

Where to Begin

- There are minimal automation requirements for your DevOps to build security in
  - You can't just have a QA team

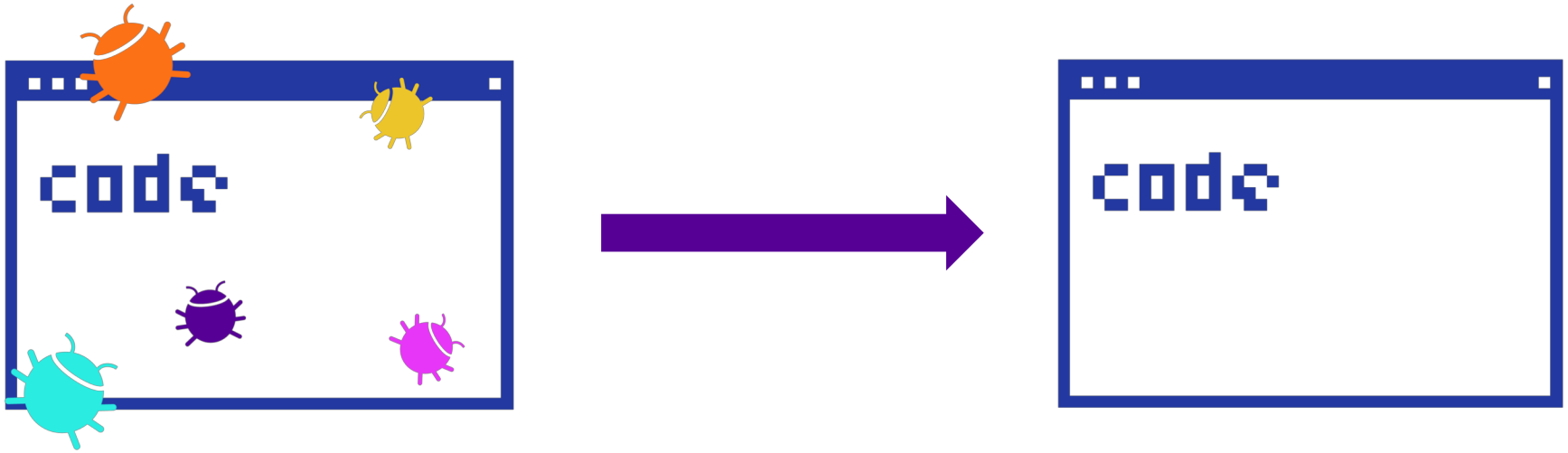


RSAConference2019

Automate the easy stuff, focus on the hard stuff

Where to Begin

- CI (Continuous Integration)
  - Get features merged into your mainline or master branch frequently
  - Typically also paired with CD (Continuous Deployment)
  - Allows mainline to be built and deployed at any time
    - You MUST have great, automated testing for this to work!



Making CI and CD Successful

- CI / CD need infrastructure
  - If you're going to deploy multiple times a day, even to staging, you need infrastructure to automatically run tests and be assured of correctness

CI / CD Platforms

- A place to run your code and verify tests pass


Travis CI

TeamCity

PHABRICATOR


CODESHIP

go 15.2

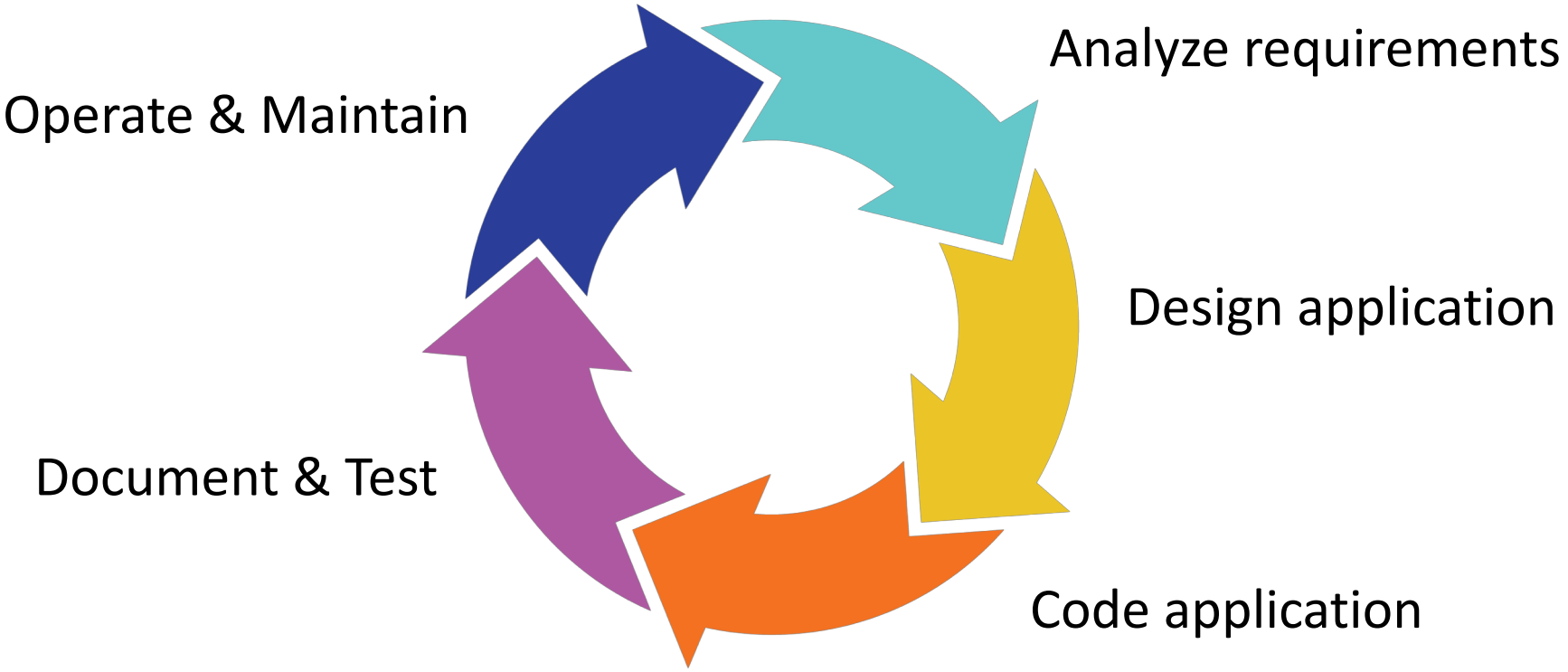
circleci


Jenkins

Visual Studio Team Services

Build an Organization That Supports SecDevOps

- Define a pipeline for your process so you can track code as it moves through the system



Current Organization Pitfalls

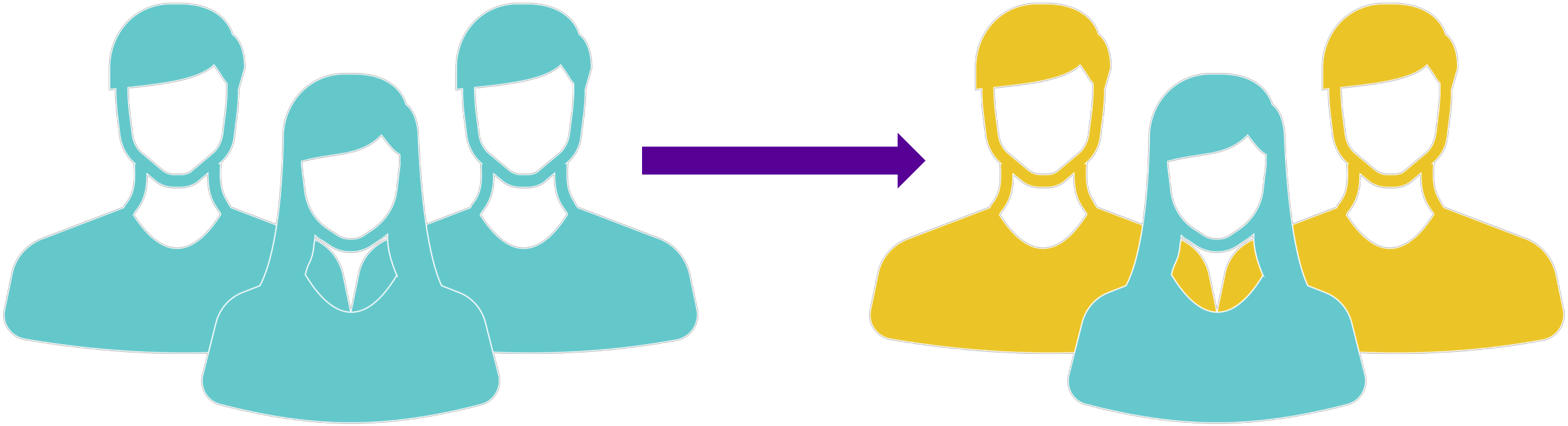
- Most organizations have kept security as the final check, but that has drawbacks
  - Security mitigations can make it hard to find the root of a vulnerability
  - Without automation, manual processes are slow and never create optimal results

Solution 1: Embedded Security

- Security engineers embedded across each team in the organization

Solution 2: Training the Breaking Mindset

- Recurring training and proper tooling



Our goal: Collaboration and Automation.

CISO

Retains talent

Projects completed faster

CIO / CTO

Builds securely

Security

Can focus on big picture

Unblocked by security team

Engineering & Ops

Why (Some) Security Has to Be Automated

- Computer's aren't as ingenious as humans – humans are needed for the hard stuff.
- Humans have limited time and energy, so automate the easy stuff to allow them to focus on the hard stuff.

RSAConference2019

PAUSE: What's easy? What's hard?

Easy: not needing domain knowledge

- Things that are similar across products, teams, technologies.
- Example: SQL injection on all of the search forms on a product

Hard: needing to understand the product

- Domain knowledge and business logic is needed from the people who build and use the product.
- Example: Second order SQL injection on an order form that requires knowledge about the shopping cart system

RSAConference2019

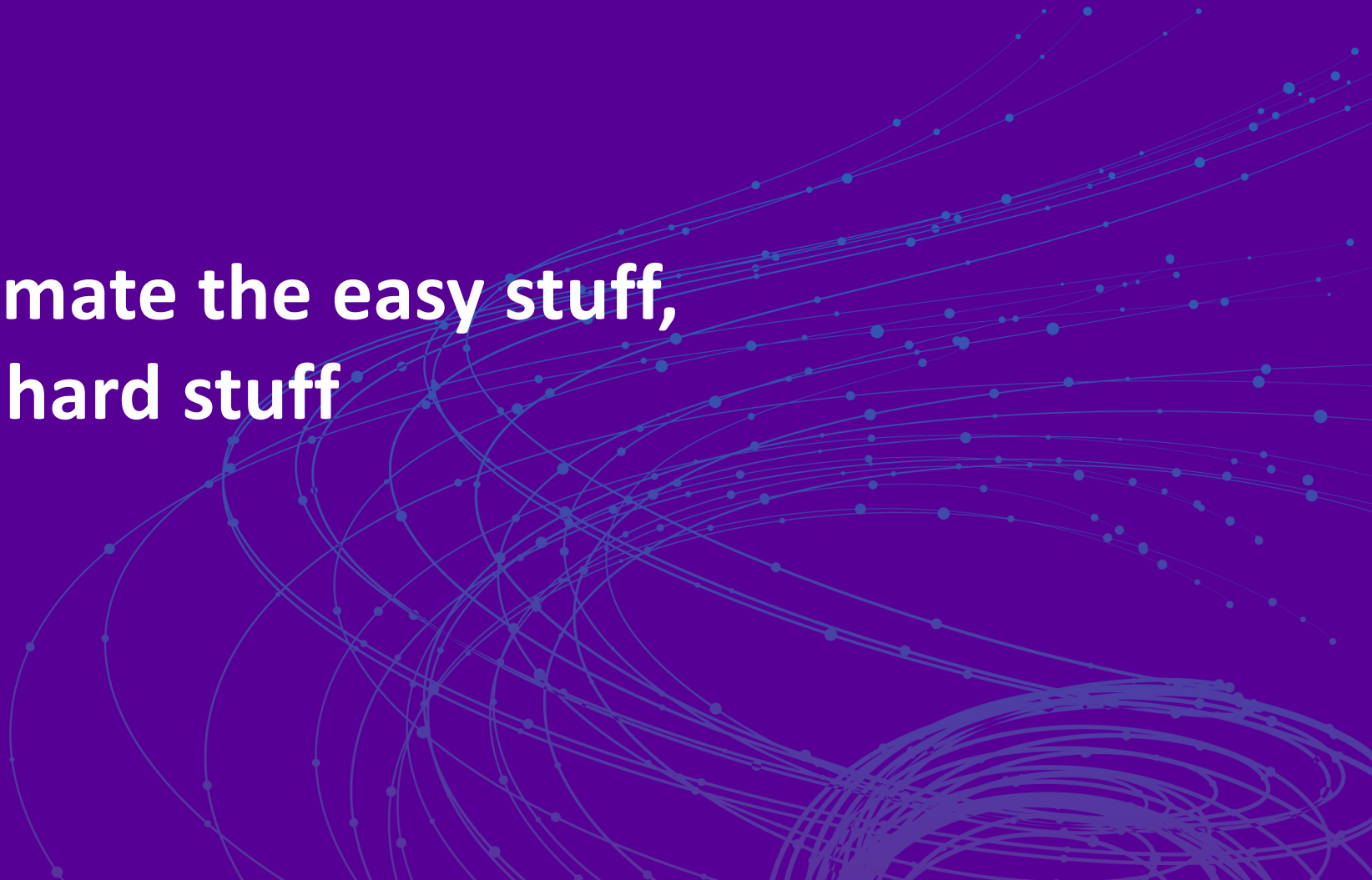
Automate the easy stuff, focus on the hard stuff

How to Have Vulnerabilities Fixed Faster and Earlier

- Good automated testing as a culture

RSAConference2019

Again… Automate the easy stuff, focus on the hard stuff



How to Have Vulnerabilities Fixed Faster and Earlier

- Good automated testing as a culture
  - Educate your developers

How to Have Vulnerabilities Fixed Faster and Earlier

- Good automated testing as a culture
  - Educate your developers
  - Build a culture of security awareness

How to Have Vulnerabilities Fixed Faster and Earlier

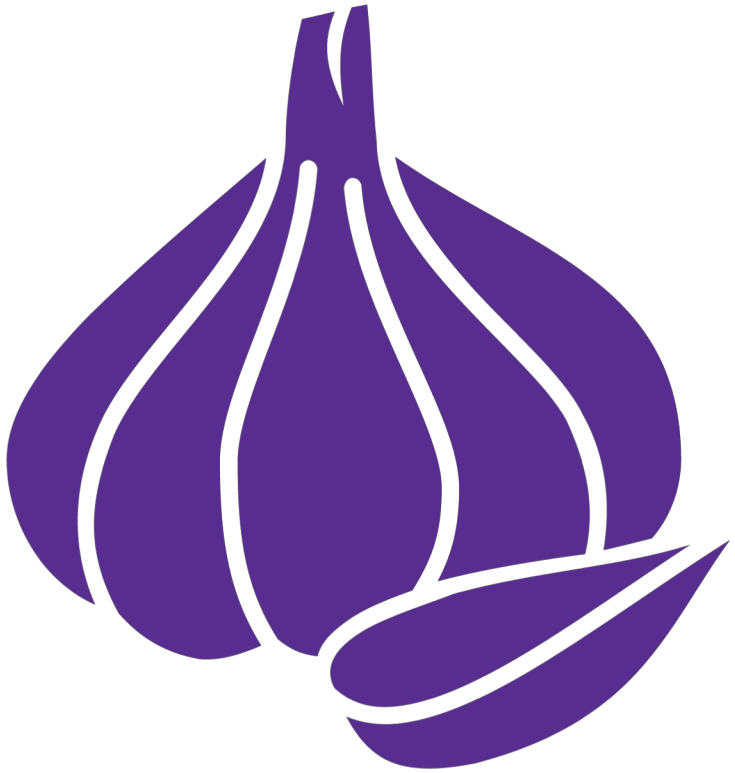
- Good automated testing as a culture
  - Educate your developers
  - Build a culture of security awareness
- Build security into the development processes

How to Have Vulnerabilities Fixed Faster and Earlier

- Good automated testing as a culture
  - Educate your developers
  - Build a culture of security awareness
- Build security into the development processes
- Simplify reporting – provide only required details, with suggested fixes

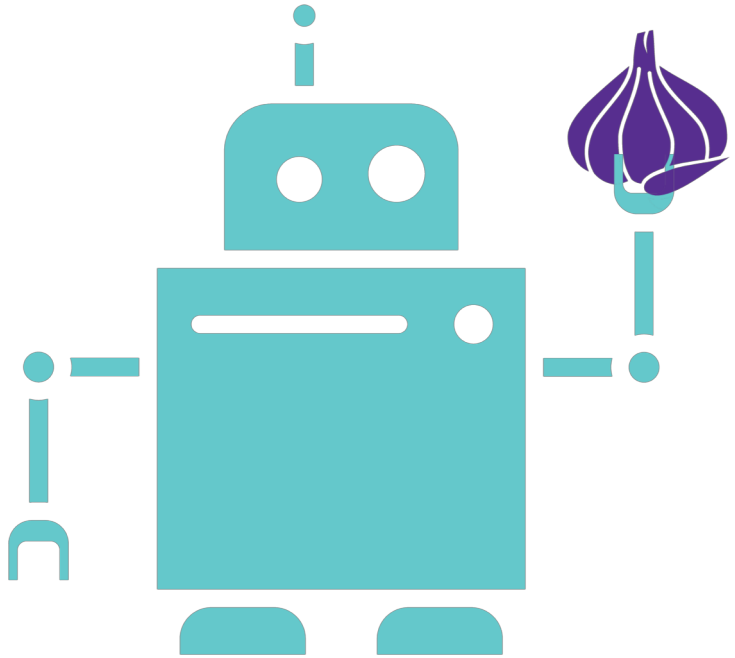
Where Do You Draw the Line?

- Let's be honest… security is like an onion. Luckily, you decide how many layers of security you want and need
  - How vulnerable of a target are you today?
  - How attractive of a target are you?
  - What's the potential loss?
    - Monetary
    - Reputation
    - Trust
    - PII / Compliance



Let's Start Building Security Into Development

- Next week you should:
  - Assess where you are _not_ automating, but could be
  - Decide on how you plan to restructure your organization with security embedded throughout
    - Integrate security into each team
    - Train the breaking mindset



Let's Start Building Security Into Development

- In 3 months you should:
  - Automate your deployments and implement a CI/CD process
  - Re-evaluate your security tools, focused on automation and integration
  - Have a security liaison spend time with each team to evaluate gaps

Let's Start Building Security Into Development

- In 6 months you should:
  - Automate and integrate your security
    - Pen tests, network tests, web tests, API tests
  - Implement your organizational changes
    - Embed security in each team
    - Train the breaking mindset

RSAConference2019

Automate the easy stuff, focus on the hard stuff

More questions? Find me at booth # 3216 (South Expo)

ceo@tinfoilsecurity.com