

Ship Secure Code ... Always

How to continuously ship secure code



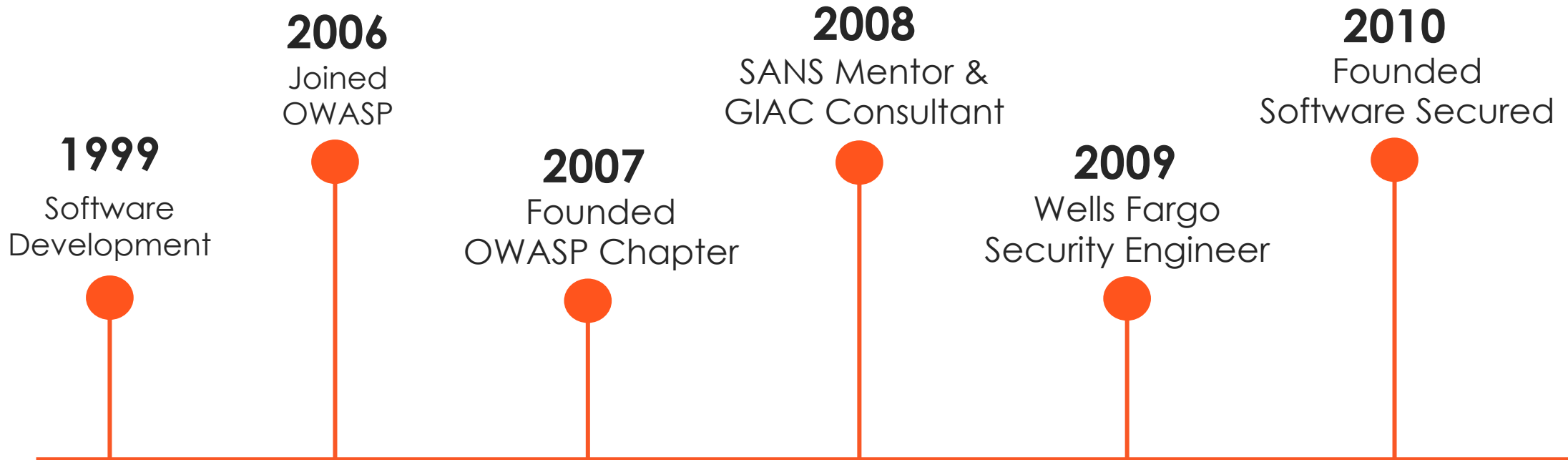
Sherif Koussa



@skoussa

ABOUT ME

SOFTWARE SECURED



Certifications: GSSP-Java, GSSP-NET, GWAPT

SOFTWARE SECURED

Penetration Testing as a Service company based out of
Ottawa, Canada.



How can we continuously ship secure code

**But first...a quick look on how
did it all start...**

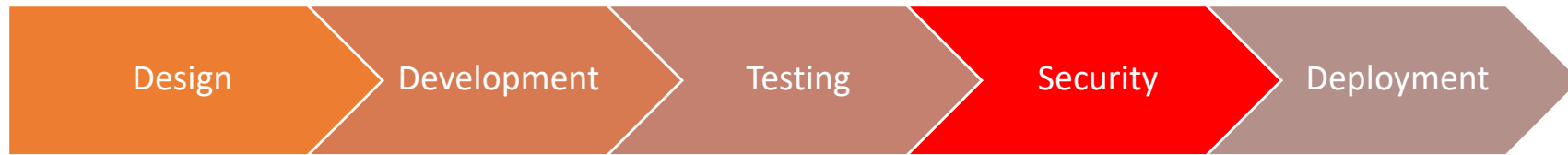
Waterfall SDLC



18 Months

A blue double-headed arrow spans the width of the four phases above it, indicating the total duration of the project cycle.

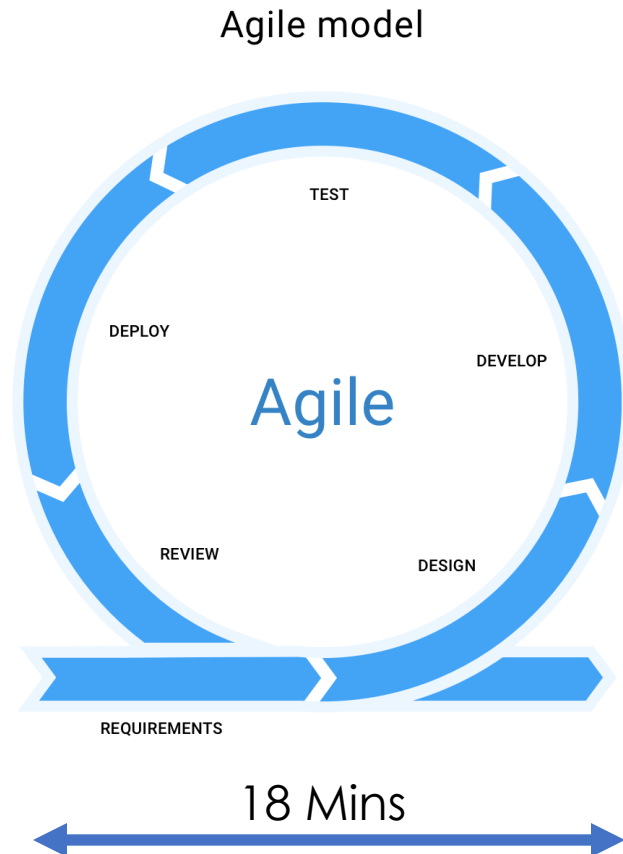
Waterfall Secure SDLC



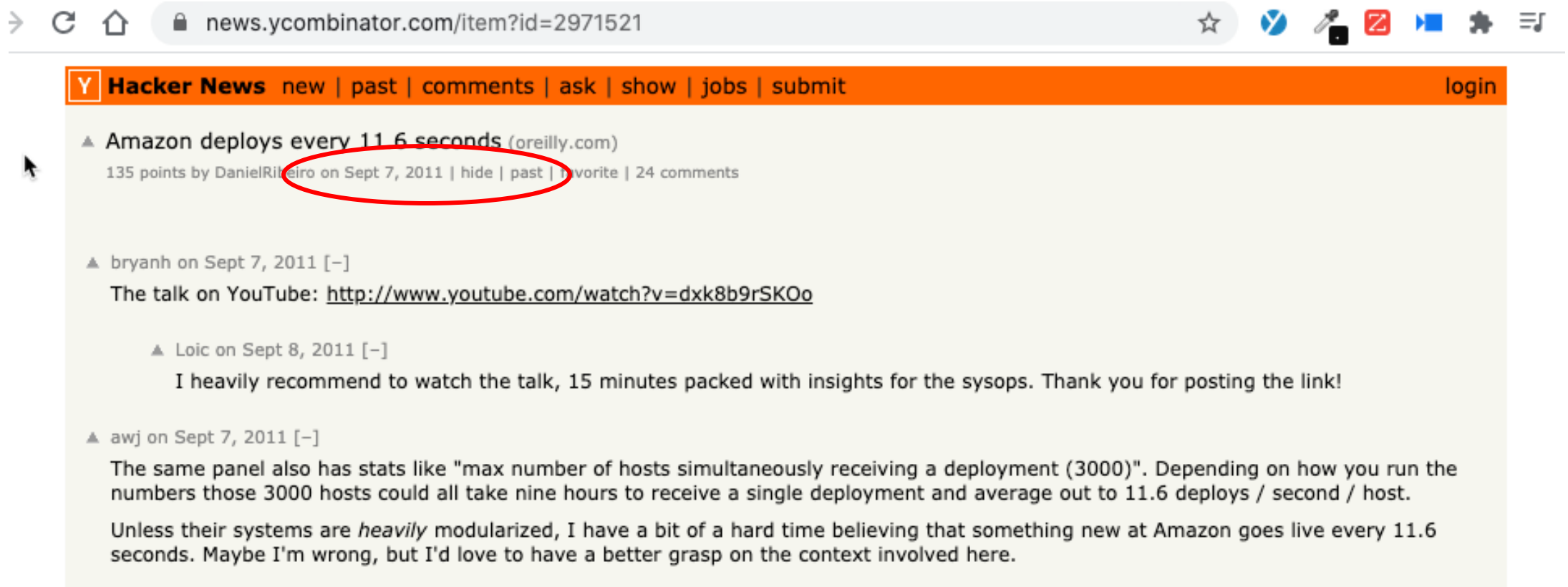
21 Months

A blue double-headed arrow spans the width of the diagram, indicating the total duration of the 21-month process.

But...the way we develop software completely changed



Software Development on Steroids



The screenshot shows a web browser window with the URL `news.ycombinator.com/item?id=2971521`. The browser's address bar and toolbar are visible at the top. Below the browser window is a screenshot of the Hacker News website. The page has an orange header with the 'Y' logo, the text 'Hacker News', and navigation links: 'new', 'past', 'comments', 'ask', 'show', 'jobs', 'submit', and a 'login' link on the right. The main content area is white and contains a list of items. The first item is titled 'Amazon deploys every 11.6 seconds (oreilly.com)' and has 135 points by Daniel Ribeiro, dated Sept 7, 2011. The text '11.6 seconds' in the title is circled in red. Below the title are links for 'hide', 'past', 'favorite', and '24 comments'. The second item is by 'bryanh' on Sept 7, 2011, with the text 'The talk on YouTube: <http://www.youtube.com/watch?v=dxk8b9rSKOo>'. The third item is by 'Loic' on Sept 8, 2011, with the text 'I heavily recommend to watch the talk, 15 minutes packed with insights for the sysops. Thank you for posting the link!'. The fourth item is by 'awj' on Sept 7, 2011, with the text 'The same panel also has stats like "max number of hosts simultaneously receiving a deployment (3000)". Depending on how you run the numbers those 3000 hosts could all take nine hours to receive a single deployment and average out to 11.6 deploys / second / host. Unless their systems are *heavily* modularized, I have a bit of a hard time believing that something new at Amazon goes live every 11.6 seconds. Maybe I'm wrong, but I'd love to have a better grasp on the context involved here.'

**And it is not just
the big guys**

According to GitLab
DevSecOps Survey 2022

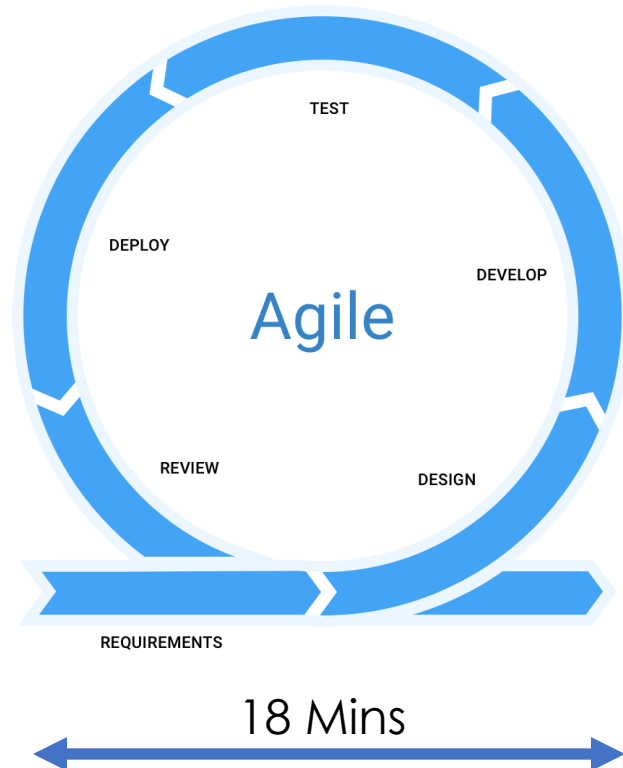


70%

of DevOps teams release code
continuously, once a day, or every
few days, up 11% from 2021.

This is so fast, how can you secure that?

Agile model



Why is so hard to continuously ship secure software

1. Lack of management support

OPINION

4 ways security has failed to become a boardroom issue

New research finds that despite being more engaged with cybersecurity, business executives and board members continue to view cybersecurity as a technology domain rather than a business concern.

2. Security and development are not on the same page

But how can they talk when they don't speak the same language



Development team speaks

Architecture

features

Releases

Code
Review

Feature
Branches

Sprint
Planning

Spring
retrospective

Security teams speak



3. Running Tools in CI <> Shipping Secure Code

Can DevSecOps be performed with one tool?

There are many tools that offer various types and combinations of services, but there is no single tool that can provide a DevSecOps process. Some vendors that offer [static application security testing](#) (SAST) tools are now adding [software composition analysis](#) (SCA) tools (and vice versa), but DevSecOps is more than just performing scans.

It's also important to note that no one tool fits in all environments, and often no one tool fits all companies. In addition to application testing tools, DevSecOps processes require reporting tools, defect tracking/management tools, environment building tools, and more. Also please note that security, build, and metric collection activities are not restricted to just the tools available in the market. Even scripts (Shell, PowerShell, Python, etc.) offer various capabilities.

So.....

How can we continuously ship
secure code

4 Main Ingredients of Shipping Secure Code

- Culture
- Process
- Tools
- KPIs

Culture....

The most important piece in the whole process

Security culture is the sum of the organization's ability to accomplish security related projects and goals

How to build a killer security culture

- 1- Security must switch to DevOps mindset
- 2- Implement security town halls
- 3- Develop security champions
- 4- Offer empathy
- 5- Lend a hand

Process....

The key is consistency

There are a ton of secure SDLC methodologies in the market

Don't follow any of them...

Why popular Secure SDLC processes don't work for most teams.

- They are very prescriptive
- Mostly designed for large teams
- Very hard to implement within and agile environment

Here is an example – OWASP SAMM



Building a good secure SDLC

- 1- Consistency: just like any new good habbit, it has to be consistent
- 2- Diversity: pick activities that span different aspects of the SDLC
- 3- Coverage: maximize code coverage

Building a GREAT secure SDLC

- 1- Feature Flags
- 2- Ramp Ups
- 3- Build a Risk Acceptance Frameworks

GREAT secure SDLC...Risk Resolution Framework

1- Implement Service Level Agreements backed by business requirements.

a- Criticals: 5 business days

b- Highs: 30 days

c- Mediums: 90 days

d- Low: 180 day

GREAT secure SDLC...Risk Resolution Framework

2- Implement Risk Acceptance Policy

- a- Risk Elimination
- b- Risk Mitigation
- c- Risk Delegation
- d- Risk Acceptance

Tools....

If you implement the previous, tools would be the easy part...

Tools you can choose from

- 1- Threat modelling
- 2- Security requirements
- 3- SAST (Static Analysis Security Testing)
- 4- DAST (Dynamic Analysis Security Testing)
- 5- IAST (Interactive Analysis Security Testing)
- 6- SCA (Software Composition Analysis)
- 7- Penetration Testing

What to consider when choosing new tools

1- SDLC diversity: ensure that you targeting different aspects of the SLDC

2- Cloud vs on-prem: some tools work cloud only, some work on-prem only

3- Business requirements: You have to cover what's mandates by the business

KPIs...

The most underrated, underestimated and underdeveloped aspect

Why measuring is important...

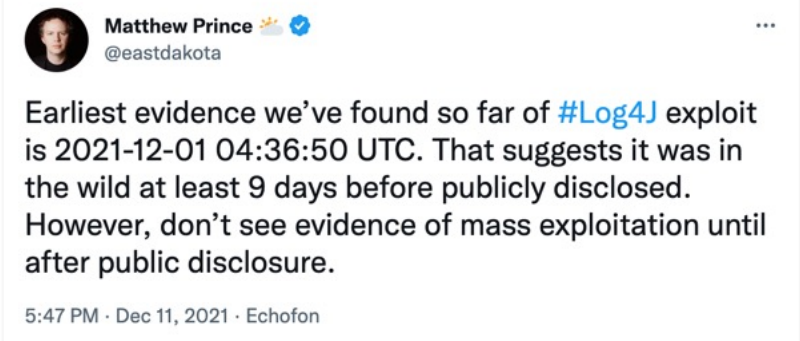
- 1- Easier to align the team
- 2- Easier to measure ROI
- 3- Easier to optimize (and expand) budgets
- 4- Easier to know where to focus

The most important KPI is

Time to remediation

Attackers need 8 days to weaponize a vulnerability on average and sometimes faster...

- 1- CVE-2022-1388 (a critical unauthenticated remote command execution vulnerability that was affecting F5 BIG-IP products): 8 hours
- 2- CVE-2021-44228 (Log4j vulnerability): 9 days before it was published



It takes, on average, the following to fix a vulnerability

97 Days

Other KPIs

- 1- # of vulnerabilities discovered by repo
- 2- # of vulnerabilities per source
- 3- # of vulnerabilities at each stage of the SDLC

A lightweight security process to kickstart things off

For teams building a new security culture

- 1- Establish alignment on security goals
- 2- Integrate DAST into the CI Pipeline
- 3- At least a yearly thorough pentest
- 4- Measure time to remediation

For teams with established good security culture

- 1- Establish risk acceptance criteria
- 2- Integrate SAST, SCA into the CI Pipeline
- 3- Penetration Testing as a Service
- 4- Optimize time to remediation

For teams who are pushing for the next level

- 1- Establish security requirements within user stories
- 2- Build feature flags
- 3- Optimize number of vulnerabilities by source

**Stop allowing security to be
thankless work**

SoftwareSecured

▮

THANK YOU 😊

sherif@softwaresecured.com

@skoussa