Investing in Agile Development: Studying BDD, SAFe and DevSevOps

Saniora R. Duclervil, Dr. Jing-Chiou Liou
Computer Science, Kean University, duclersa@kean.edu

Objectives

To study current DevSecOps software development processes models and to develop evaluation criteria for performance comparison

Introduction

- Although the Waterfall method for software development is effective, the focus has been shifting into Agile development in recent years. However, Security is still a huge component. It is agreed upon that security needs to be embedded not matter what development process is being used.
- Now that software development is gearing towards Agile, I decided to analyze three models: BDD (Behavioral Driven Development), SAFe (Scaled Agile Framework), and DevSecOps

Methods and Materials

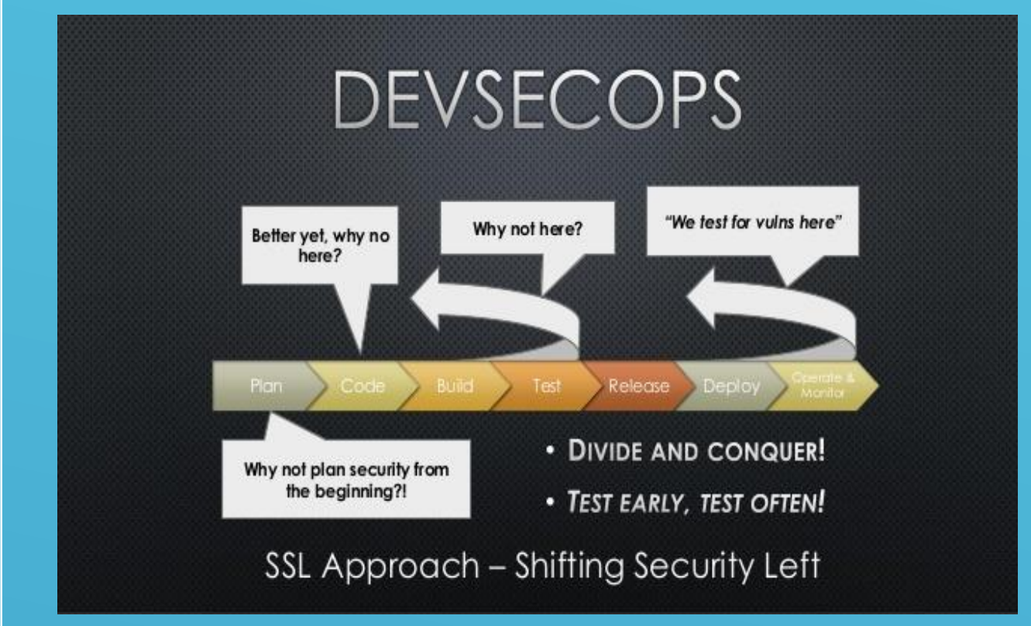
- Identified models
- Identified commonality and differences in each model
- Identified criteria
- Compared the models

About BDD, SAFe, and DevSecOps

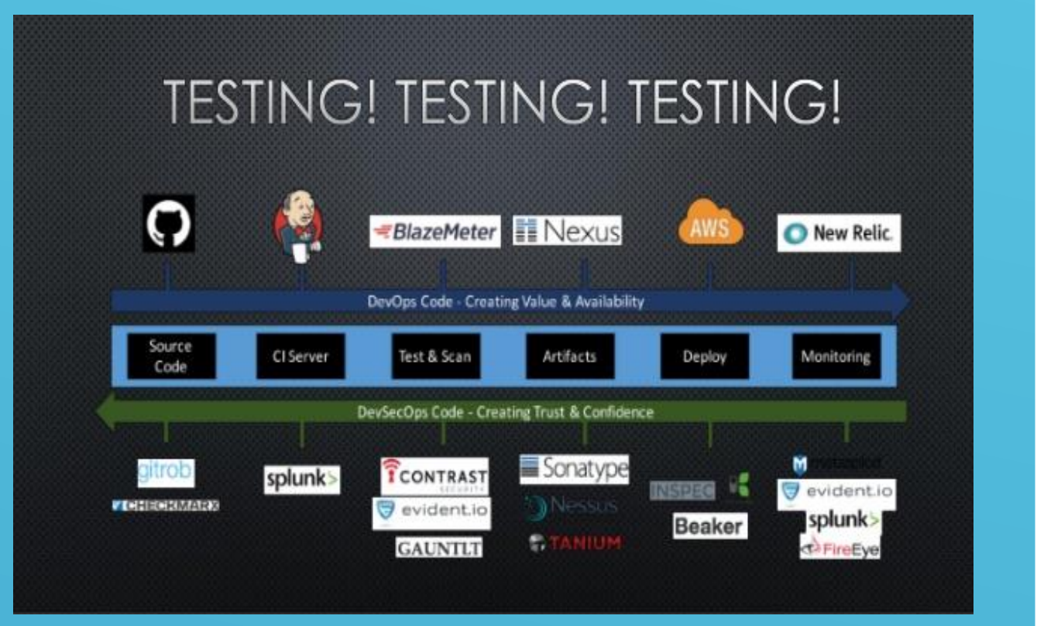
- BDD: BDD is an agile software development process designed to focus on the behavior of the system rather then its current state. BDD focuses on the behavior of the software by suing scenarios and features that users come up with. one of the characteristics is ubiquitous language.
- SAFe is software development process that incorporates Lean-Agile Development, principles and practices that are used to develop software quickly, with high quality and on time.
- DevSevOps: is development, security and operation. Originally it was DevOps which is used in SAFe and other development processes. DevOps uses the continues delivery pipeline where there is continuous development, operation, integration and delivery. People are now focusing on incorporating security into DevOps which brings us into DevSecOps which not only incorporates continuous operation and delivery, but continuous security.
- After our initial study, we decided to focus on DevSevOps because it focuses more on incorporating security into the development process.

DevSecOps

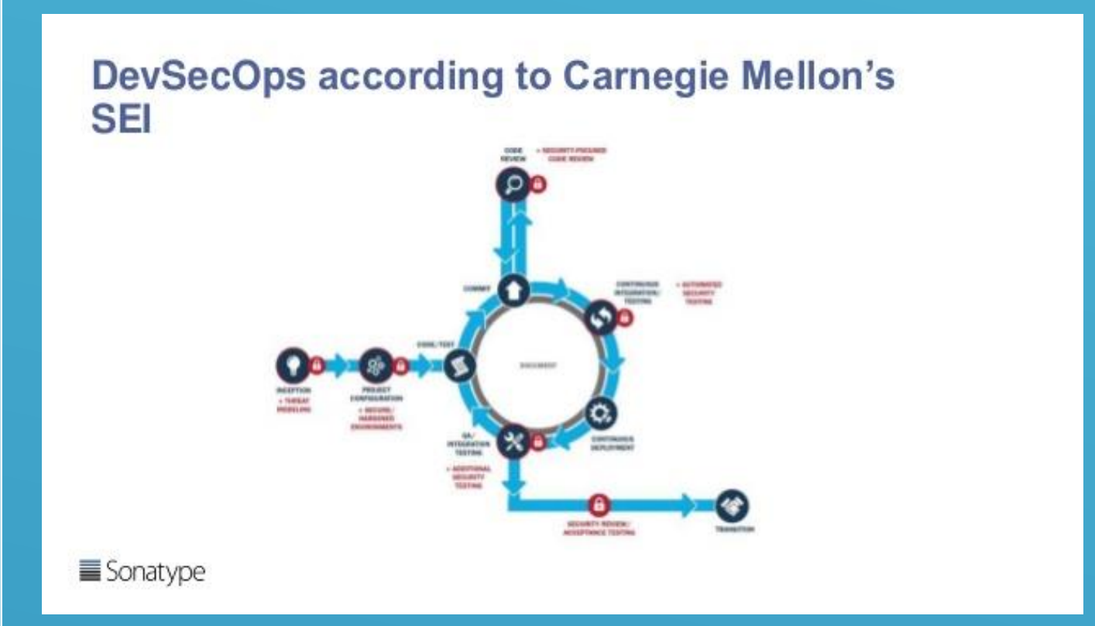
- As mentioned before, DevSecOps incorporates continuous security. Here are the different models that many people have incorporated:

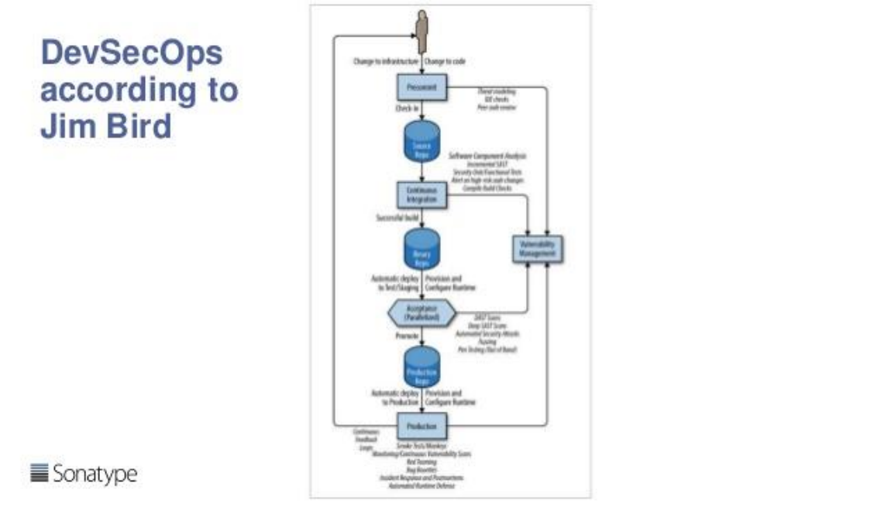


Shift Security Left model



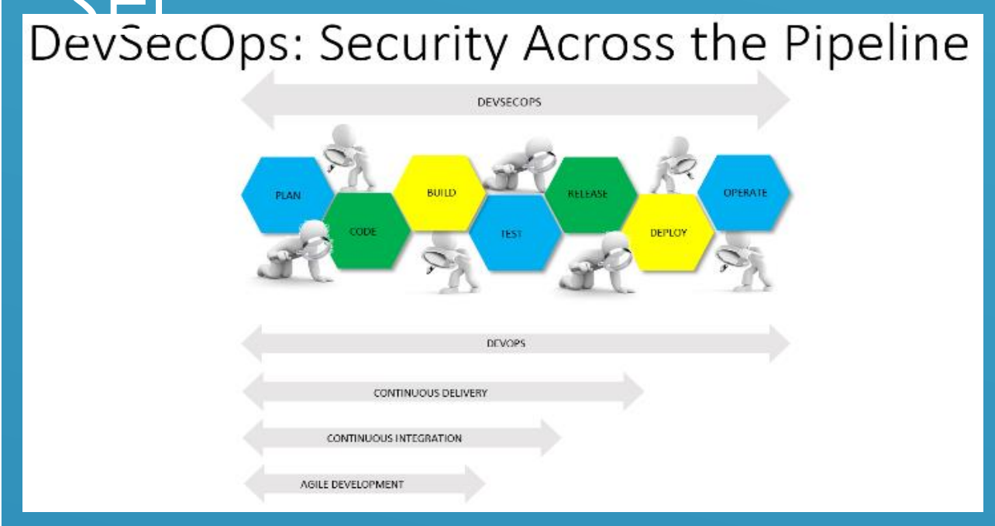
Mango Rodrigues DevSecOps pipeline model



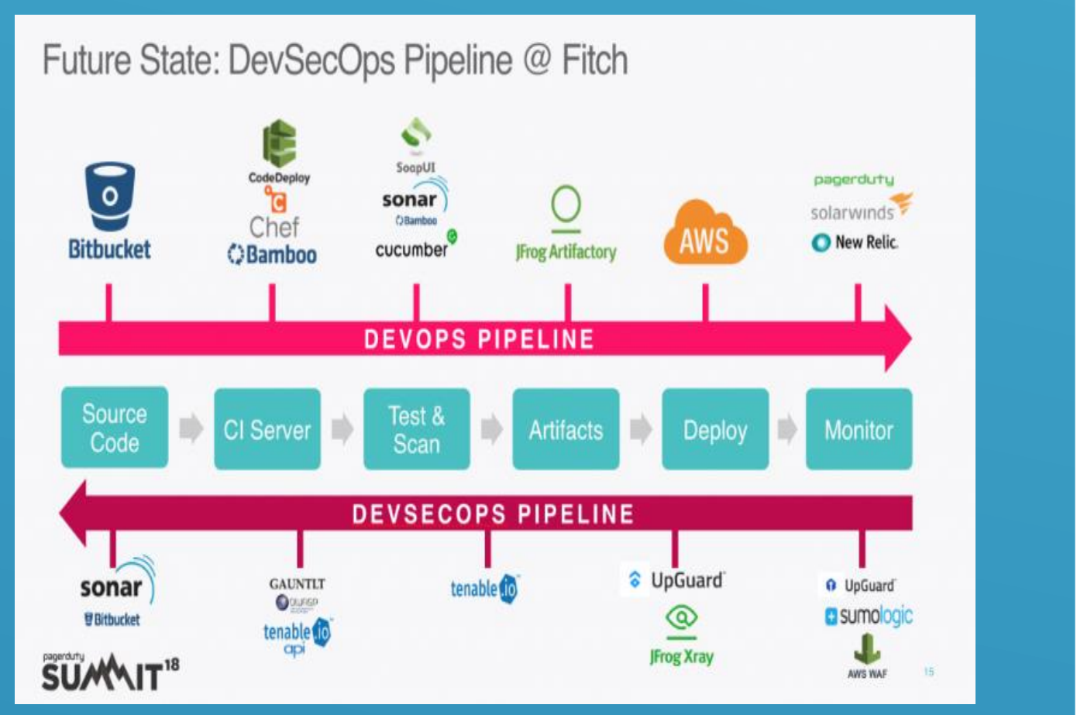


DevSecOps according to Carnegie Mellon's SEI

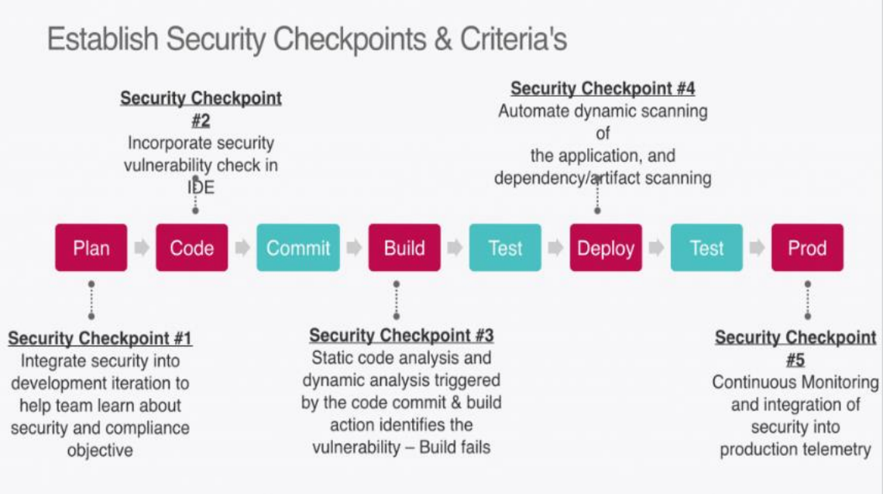
DevSecOps According to Jim Bird



DevSecOps by Tom Porter



Fitch DevSecOps pipeline model



Establish Security Checkpoints & Criteria's

Security Checkpoint #1: Integrate security into development iteration to help team learn about security and compliance objective

Security Checkpoint #2: Incorporate security vulnerability check in IDE

Security Checkpoint #3: Static code analysis and dynamic analysis triggered by the code commit & build action identifies the vulnerability – Build fails

Security Checkpoint #4: Automate dynamic scanning of the application, and dependency/artifact scanning

Security Checkpoint #5: Continuous Monitoring and integration of security into production telemetry

Incorporating Security checkpoints in Fitch's DevSecOps pipeline

References

- Logan, Magno. "DevSecOps - Integrating Security in the Development Process (with Mem..." LinkedIn SlideShare, 12 Nov. 2017, www.slideshare.net/magnologan/devsecops-integrating-security-in-the-development-process-with-memes-magno-logan.
- Sonatype. "DevSecOps Reference Architectures 2018." LinkedIn SlideShare, 1 Feb. 2018, www.slideshare.net/SonatypeCorp/devsecops-reference-architectures-2018.
- Bird, Jim. Delivering Secure Software through Continuous Delivery. 24 June 2016, iotiran.com/media/k2/attachments/devopssec.pdf.
- Porter, Tom. "DevSecOps - A New Chance for Security - DZone DevOps." Dzone.com, 5 June 2018, dzone.com/articles/shifting-left-devsecops.
- TC Currie. "100-Year-Old Fitch Ratings Upgrades to DevSecOps." The New Stack, 23 Oct. 2018, thenewstack.io/100-year-old-fitch-ratings-upgrades-to-devsecops/.

Results

In the preliminary stage, we will mostly be evaluating the models based on their security implementations and tools used.

Models	Criterion 1: Type of security testing incorporated or tools	Criterion 2: Each of the phases incorporates security	Criterion 3: Uses different tools to code, build, test and release software	Criterion 4: Incorporates at least or like the coding, building, testing, release, and operation phase	Criterion 5 Is there a preliminary phase in the process model
Mango Rodrigues's DevSecOps pipeline	Use of code review, you of security testing software throughout all phases. CI server and continues security. Security plugins available	yes	yes	Yes	no
DevSecOps according to Carnegie Mellon's SEI	Threat modeling, code review, continuous integration testing, Quality Assurance, security review using acceptance testing	yes	yes	yes	yes
DevSecOps According to Jim Bird	Threat modeling, static analysis, code reviews, incremental static analysis, security automated configuration management, automated testing using security tools, pen testing	yes	No	yes	yes
DevSecOps by Tom Porter	Security analysis, test plan, Git and IDE controls, linters and unit tests, pen testing, vulnerability scanning, SAST, DAST, integration tests	yes	yes	yes	yes
Fitch's Security checkpoint & criteria model and DevSecOps pipeline	Includes security checkpoint in each phase. DAST, SAST, vulnerability testing	yes	yes	yes	no

The table above shows the comparisons of the models. Not all the information are shown because there was enough space.

, we found three most effective models that incorporates security and are the top 3, best fit models for DevSecOps

- Carnegie Mellon's SEI DevSecOps model
- Jim Bird's DevSecOps
- Fitch's DevSecOps pipeline and security checkpoints

- Carnegie Mellon's model would come in third. It incorporates many security components but not as much as Jim Bird's model.
- According to the security components, Jim Bird's model would come in second.
- In first pace we have fitch's model with security checkpoints. Not only does this model incorporate security components, security and other tools are suggested for use. Security and many other tasks are automated in Fitch's model so there less room for human error.