DevOps

Week 14

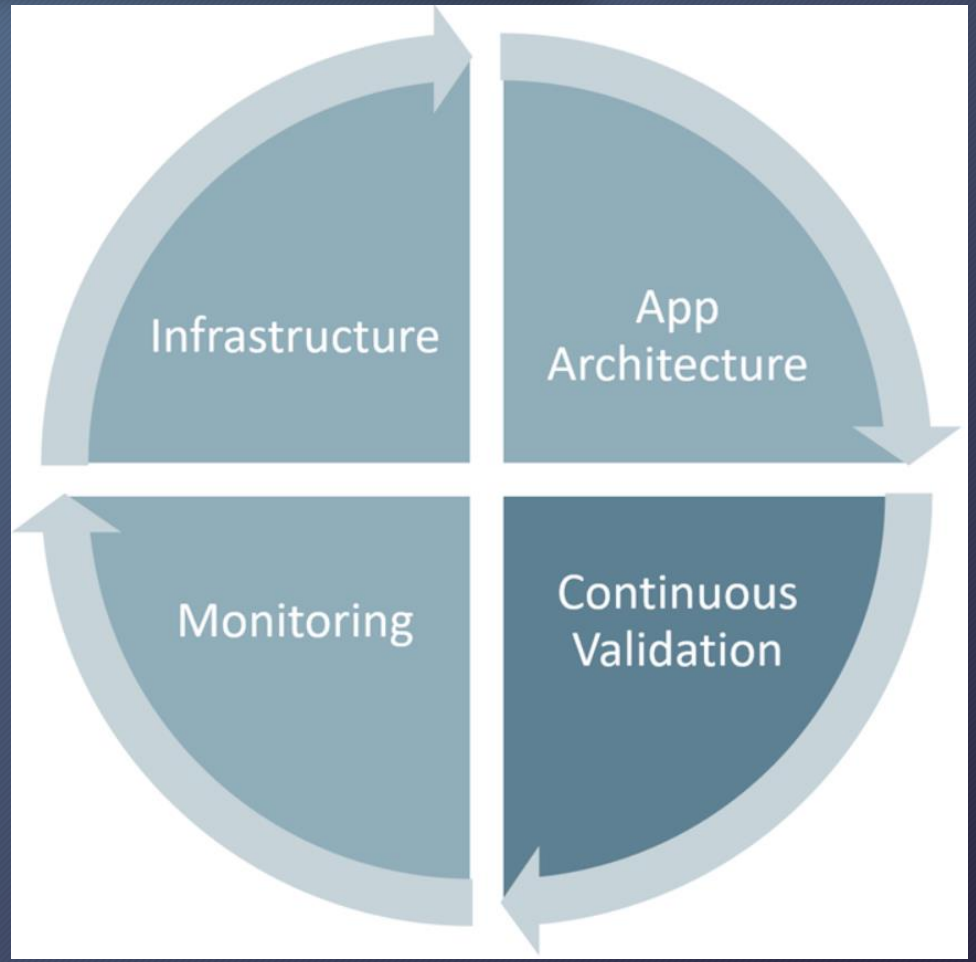
Murtaza Munawar Fazal

DevOps

- DevOps teams have access to unprecedented infrastructure and scale thanks to the cloud. They can be approached by some of the most nefarious actors on the internet, as they risk the security of their business with every application deployment.
- How do you ensure your applications are secure and stay secure with continuous integration and delivery?
- How can you find and fix security issues early in the process?
- It begins with practices commonly referred to as DevSecOps.

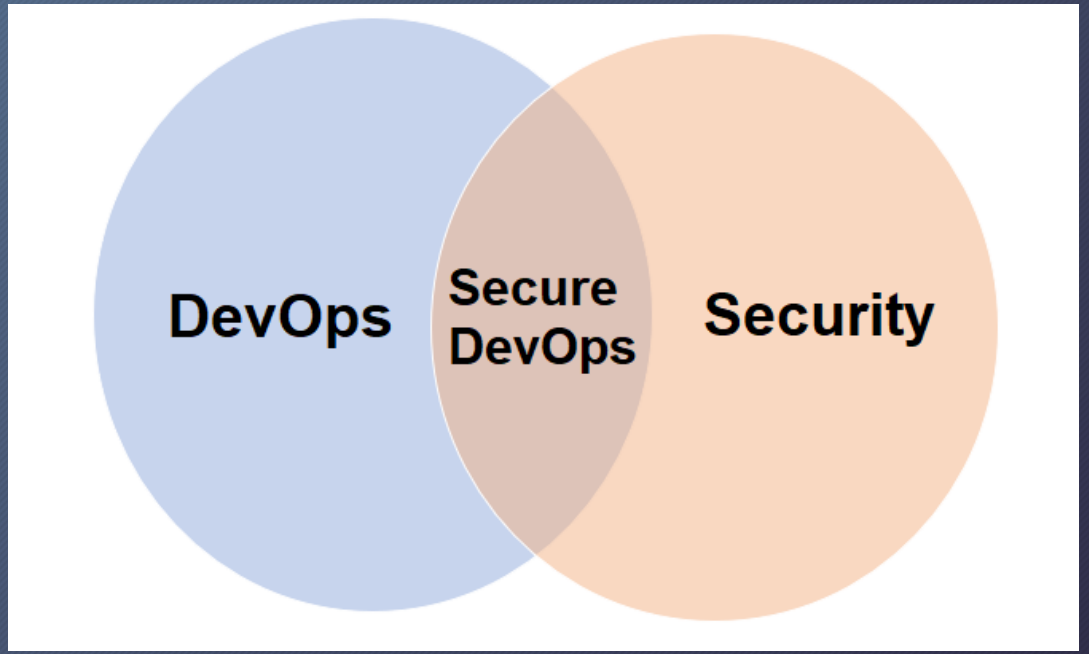
DevSecOps

- DevSecOps incorporates the security team and their capabilities into your DevOps practices making security the responsibility of everyone on the team.
- Security needs to shift from an afterthought to being evaluated at every process step.
- Securing applications is a continuous process encompassing secure infrastructure, designing architecture with layered security, continuous security validation, and monitoring attacks.



Secure DevOps (or DevSecOps)

- DevOps is about working faster.
- Security is about-emphasizing thoroughness.
- Security concerns are typically addressed at the end of the cycle. It can potentially create unplanned work right at the end of the pipeline.
- Secure DevOps integrates DevOps with security into a set of practices designed to meet the goals of both DevOps and safety effectively.
- A Secure DevOps pipeline allows development teams to work fast without breaking their project by introducing unwanted security vulnerabilities.



Secure DevOps (or DevSecOps)

- Secure DevOps addresses broader questions, such as:
  - Is my pipeline consuming third-party components, and are they secure?
  - Are there known vulnerabilities within any of the third-party software we use?
  - How quickly can I detect vulnerabilities (also called time to detect)?
  - How quickly can I remediate identified vulnerabilities?

- Security practices for detecting potential security anomalies must be as robust and fast as your DevOps pipeline's other parts. It also includes infrastructure automation and code development.

SQL injection

- SQL Injection is an attack that makes it possible to execute malicious SQL statements.
- Attackers can use SQL Injection vulnerabilities to bypass application security measures.
- They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database.
- They can also use SQL Injection to add, modify, and delete records in the database.
- Criminals may use it to gain unauthorized access to, delete, or alter your sensitive data:
  - customer information,
  - personal data,
  - trade secrets,
  - intellectual property, and more.

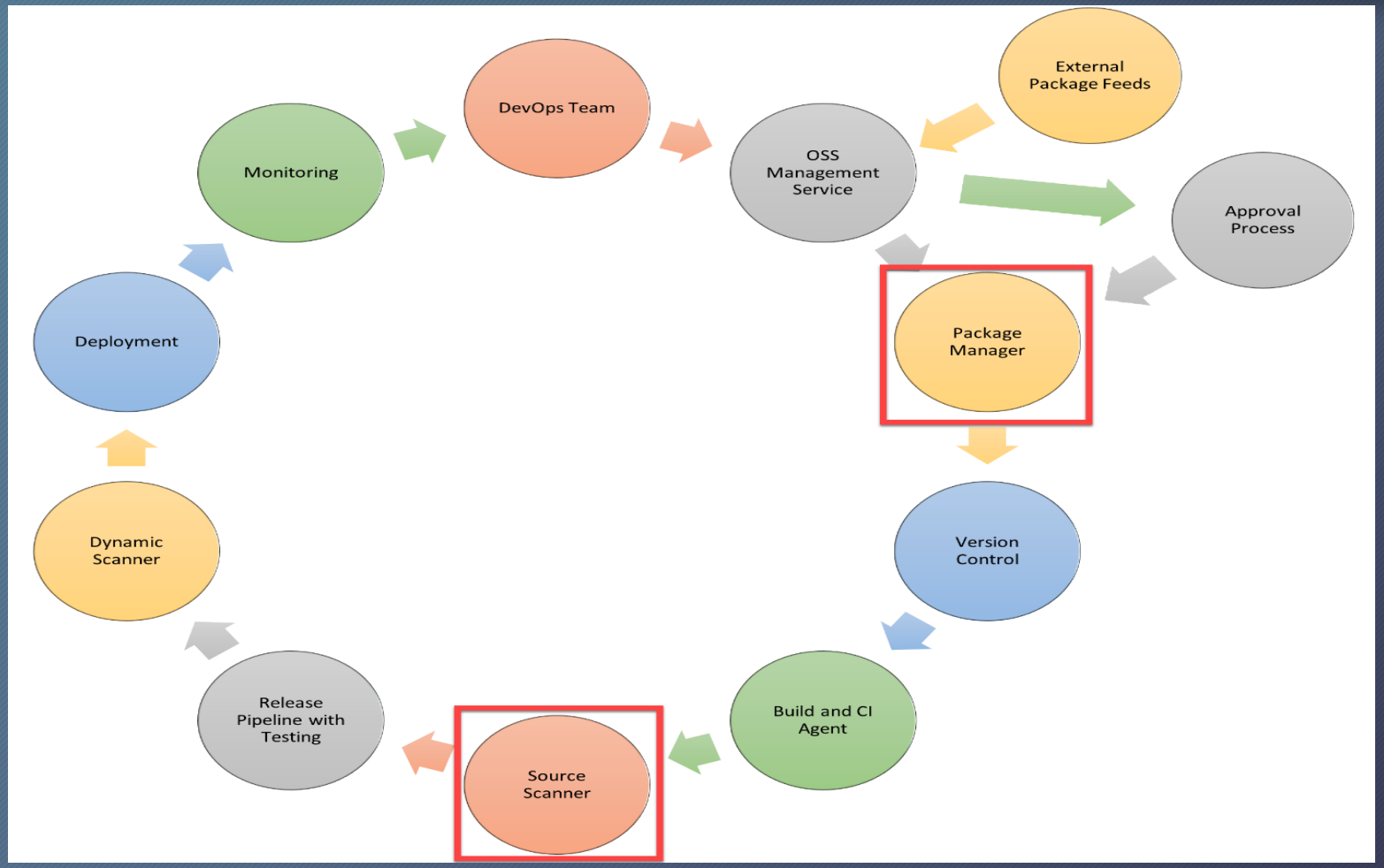
SQL injection

- SQL Injection attacks are among the oldest, most prevalent, and most dangerous web application vulnerabilities.
- The OWASP organization (Open Web Application Security Project) lists injections in their OWASP Top 10 2017 document as the number one threat to web application security.

Secure DevOps Pipeline

- The goal of a Secure DevOps Pipeline is to enable development teams to work fast without introducing unwanted vulnerabilities to their project.
- Two essential features of Secure DevOps Pipelines that aren't found in standard DevOps Pipelines are:
  - Package management and the approval process associated with it.
  - Source Scanner is also an extra step for scanning the source code.
    - This step allows for security scanning and checking for vulnerabilities that aren't present in the application code. The scanning occurs after the app is built before release and pre-release testing. Source scanning can identify security vulnerabilities earlier in the cycle.

Secure DevOps Pipeline



Continuous Security Validation

- With faster delivery and better productivity, open-source software (OSS) components are encouraged across many organizations.
- However, as the dependency on these third-party OSS components increases, the risk of security vulnerabilities or hidden license requirements also increases compliance issues.
- For a business, it's critical, as issues related to compliance, liabilities, and customer personal data can cause many privacy and security concerns.
- Identifying such issues early in the release cycle gives you an advanced warning and enough time to fix the problems. Notably, the cost of rectifying issues is lower the earlier the project discovers the problem.
- Many tools can scan for these vulnerabilities within the build and release pipelines.
- Once the merge is complete, the CI build should execute as part of the pull request (PR-CI) process.

Continuous Security Validation

- These CI builds should run static code analysis tests to ensure that the code follows all rules for both maintenance and security.
- Several tools can be used for it:
  - SonarQube.
  - Visual Studio Code Analysis and the Roslyn Security Analyzers.
  - Checkmarx - A Static Application Security Testing (SAST) tool.
  - BinSkim - A binary static analysis tool that provides security and correctness results for Windows portable executables and many more.
- Also, to verify code quality with the CI build, two other tedious or ignored validations are scanning third-party packages for vulnerabilities and OSS license usage.

Using open-source software

- Packages contain components that are built from source code.
- Open-source code is publicly available for inspection, reuse, and contribution.
- Most commonly, open-source projects indicate how the sources can be used and distributed afterward.
- A license agreement comes with the source code and specifies what can and cannot be done.
- On average, the built software solution is around 80% based on existing components and maintained outside of the project.
- The components can be a commercial offering or free of charge.
- A considerable part of the publicly available and free components are community efforts to offer reusable components for everyone to use and build software. The persons creating and maintaining these components often also make the source code available.
- It's open-source code as opposed to closed source. A closed source means that the source code isn't available, even though components are available.

Corporate Concerns with Open-Source Software Components

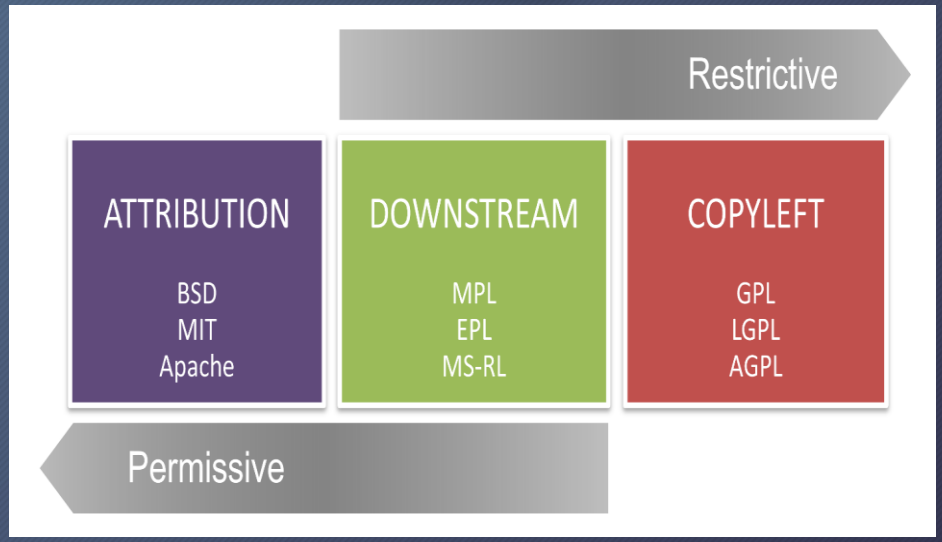
- The inclusion of software components that are not built by the companies themselves means no complete control over the sources.
- Being responsible for the source code used in components used within a company means that you must accept its risks.
- The concerns are that source code component:
  - Are of low quality. It would impact the maintainability, reliability, and performance of the overall solution.
  - Have no active maintenance. The code wouldn't evolve or be alterable without copying the source code, effectively forking away from the origin.
  - Contain malicious code. The entire system that includes and uses the code will be compromised. Potentially the whole company's IT and infrastructure is affected.
  - Have security vulnerabilities. The security of a software system is as good as its weakest part. Using source code with vulnerabilities makes the entire system susceptible to attack by hackers and misuse.
  - Have unfavorable licensing restrictions. The effect of a license can affect the entire solution that uses the open-source software.

Open-Source Licenses

- A license agreement accompanies open-source software and the related source code.
- The license describes how the source code and the components built from it can be used and how any software created with it should deal with it.
- According to the open-source definition of OpenSource.org, a license shouldn't:
  - Discriminate against persons or groups.
  - Discriminate against fields of endeavor.
  - Be specific to a product.
  - Restrict other software.
  - And more
- To cover the exact terms of a license, several types exist.
- Even though multiple contributors generally develop open-source software from the community, it doesn't guarantee that it's secure and without vulnerabilities.
- Multiple reviewers discover chances, but the discovery might not be immediate or before being consumed by others.
- Since the source code is open-source, people with malicious intent can also inspect the code for vulnerabilities and exploit it when possible.

Types of licenses

- **Attribution**
  - There are the "attribution" licenses. They're permissive and allow practically every type of use by the software that consumes it.
    - An example is building commercially available software, including the components or source code under this license.
  - The only restriction is that the original attribution to the authors remains included in the source code or as part of the downstream use of the new software.



- Copyleft
  - The "copyleft" licenses are considered viral, as the use of the source code and its components, and distribution of the complete software, implies that all source code using it should follow the same license form.
  - The viral nature is that the use of the software covered under this license type forces you to forward the same license for all work with or on the original software.
- Downstream
  - The "downstream" or "weak copyleft" licenses requires that it must do so under the same license terms when the covered code is distributed.
  - Unlike the copyleft licenses, it doesn't extend to improvements to additions to the covered code.

Integrate software composition analysis checks into pipelines

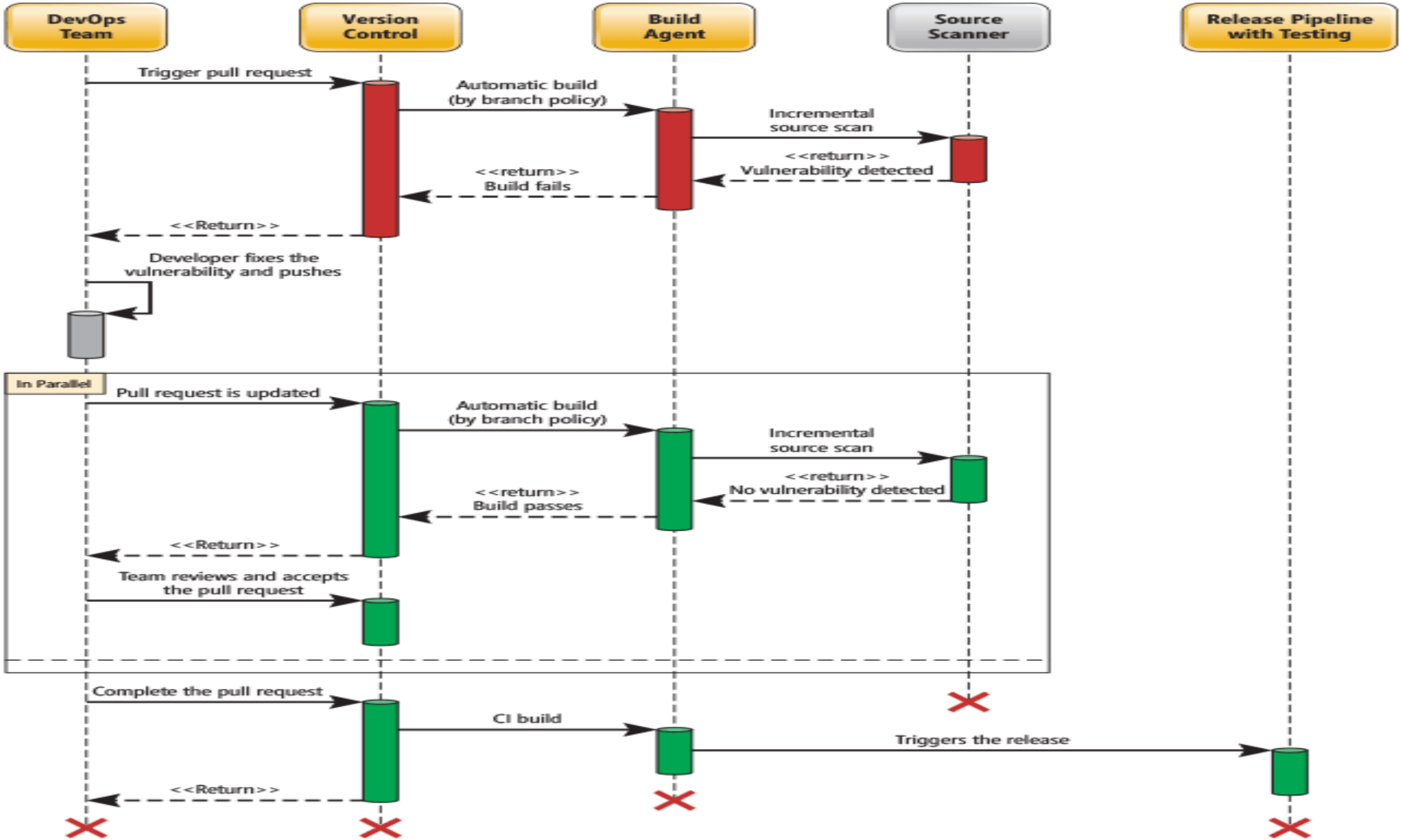
- Previously Security scanning used to be performed once per release by a dedicated security team whose members had little involvement with other groups.
- This practice creates a dangerous pattern in which security specialists find large batches of issues at the exact time when developers are under the most pressure to release a software product.
- The pressure often results in software deployment with security vulnerabilities that need to be addressed after a product has been released, integrating scanning into a team's workflow at multiple points along the development path.
- Secure DevOps can help make all quality-assurance activities, including security, continuous and automated.

Pull request code scan analysis integration

- DevOps teams can submit proposed changes to an application's (main) codebase using pull requests (PRs).
- Azure Marketplace extensions that help integrate scans during PRs include:
  - Mend. Helps validate dependencies with its binary fingerprinting.
  - Checkmarx. Provides an incremental scan of changes.
  - Veracode. Implements the concept of a developer sandbox.
  - Black Duck by Synopsis. An auditing tool for open-source code to help identify, fix, and manage compliance.
- These extensions allow developers to experiment with changes before submitting them as a PR.

Interpret alerts from scanner tools

- To correctly interpret the results of scanning tools, you need to be aware of some aspects:
  - False positives:
    - The tooling is an automated way to scan and might be misinterpreting specific vulnerabilities. On the other hand, scan results aren't guaranteed to be 100% accurate.
  - Security bug bar:
    - Most likely, many security vulnerabilities will be detected—some of these false positives, but still many findings. More findings can often be handled or mitigated, given a certain amount of time and money. The bug bar makes sure that it's clear what must be taken care of and what might be done if time and resources are left.
- By setting a security bug bar in the Definition of Done and specifying the allowed license ratings, one can use the reports from the scans to find the work for the development team.


DevOps Team
Version Control
Build Agent
Source Scanner
Release Pipeline with Testing
Trigger pull request
Automatic build (by branch policy)
Incremental source scan
<<return>> Vulnerability detected
<<return>> Build fails
<<Return>>
Developer fixes the vulnerability and pushes
In Parallel
Pull request is updated
Automatic build (by branch policy)
Incremental source scan
<<return>> No vulnerability detected
<<return>> Build passes
<<Return>>
Team reviews and accepts the pull request
Complete the pull request
CI build
Triggers the release
<<Return>>

SonarCloud (Static Analyzer)

- Technical debt saps productivity by making code hard to understand, easy to break, and difficult to validate, creating unplanned work and ultimately blocking progress.
- Technical debt is inevitable! It starts small and grows over time through rushed changes, lack of context, and discipline.
- The hardest part of fixing technical debt is knowing where to start.
- SonarQube is an open-source platform that is the de facto solution for understanding and managing technical debt.
- Originally famous in the Java community, SonarQube now supports over 20 programming languages.
- The joint investments made by Microsoft and SonarSource make SonarQube easier to integrate with Pipelines and better at analyzing .NET-based applications.

OWASP Secure Coding Practices (Dynamic Analyzer)

- The Open Web Application Security Project (OWASP) is a global charitable organization focused on improving software security.
- OWASP's stated mission is to make software security visible so that individuals and organizations can make informed decisions.
- They offer impartial and practical advice.
- OWASP regularly publishes a set of Secure Coding Practices. Their guidelines currently cover advice in the following areas:
  - Input Validation
  - Output Encoding
  - Authentication and Password Management
  - Session Management
  - Access Control
  - Cryptographic Practices
  - Error Handling and Logging
  - Data Protection
  - Communication Security
  - System Configuration
  - Database Security
  - File Management
  - Memory Management
  - General Coding Practices

OWASP ZAP penetration test

- ZAP is a free penetration testing tool for beginners to professionals.
- ZAP includes an API and a weekly docker container image to integrate into your deployment process.
- The baseline scan is designed to identify vulnerabilities within a couple of minutes, making it a good option for the application CI/CD pipeline.
- The Nightly OWASP ZAP can spider the website and run the full-Active Scan to evaluate the most combinations of possible vulnerabilities.
- OWASP ZAP can be installed on any machine in your network.

OWASP ZAP penetration test