Continuous Delivery with Secure Coding

by Anshika Verma

About Me

Pre-final year student at VIT Bhopal.
Intern at PingSafe AI.
Lucknow chapter lead at InfosecGirls.

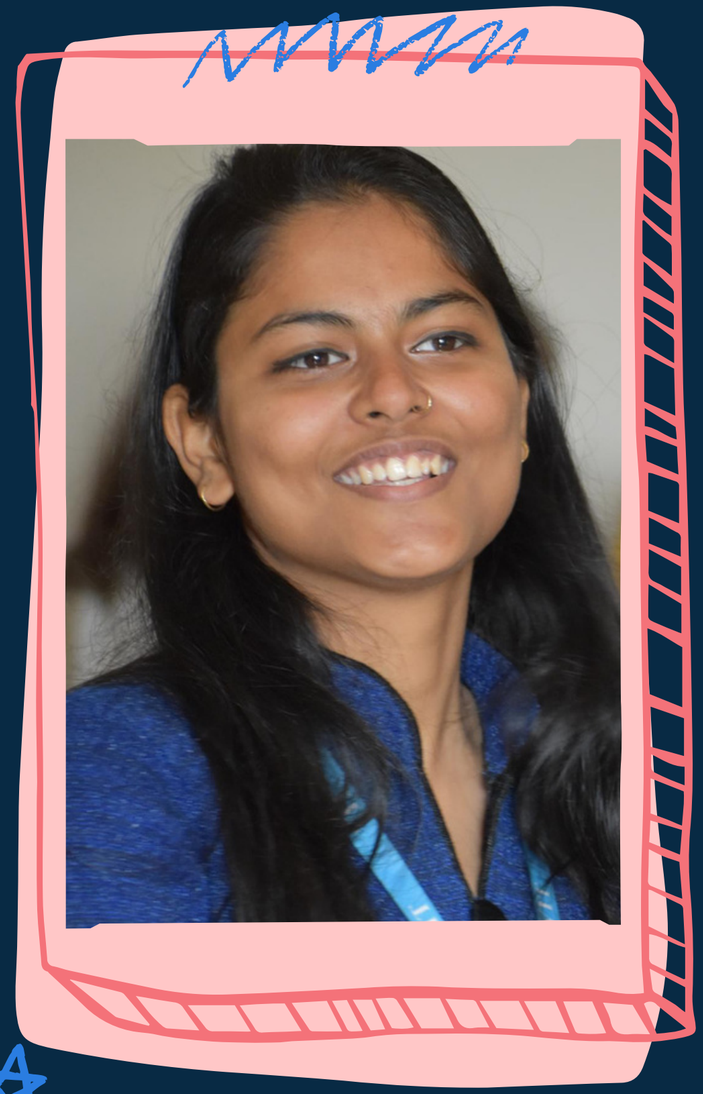


Table of Contents

Let's Talk History

Agile --> DevOps --> Continuous Delivery

2001 2007 2010

-2008

Workflow

CODE DEVELOPMENT

The developer(s) writes the code for modules.

Sent for Commits

BUILD

An automated state of compiling the different source codes.

TEST

Automated testing of integrated codes using tools.

USER ACCEPTANCE

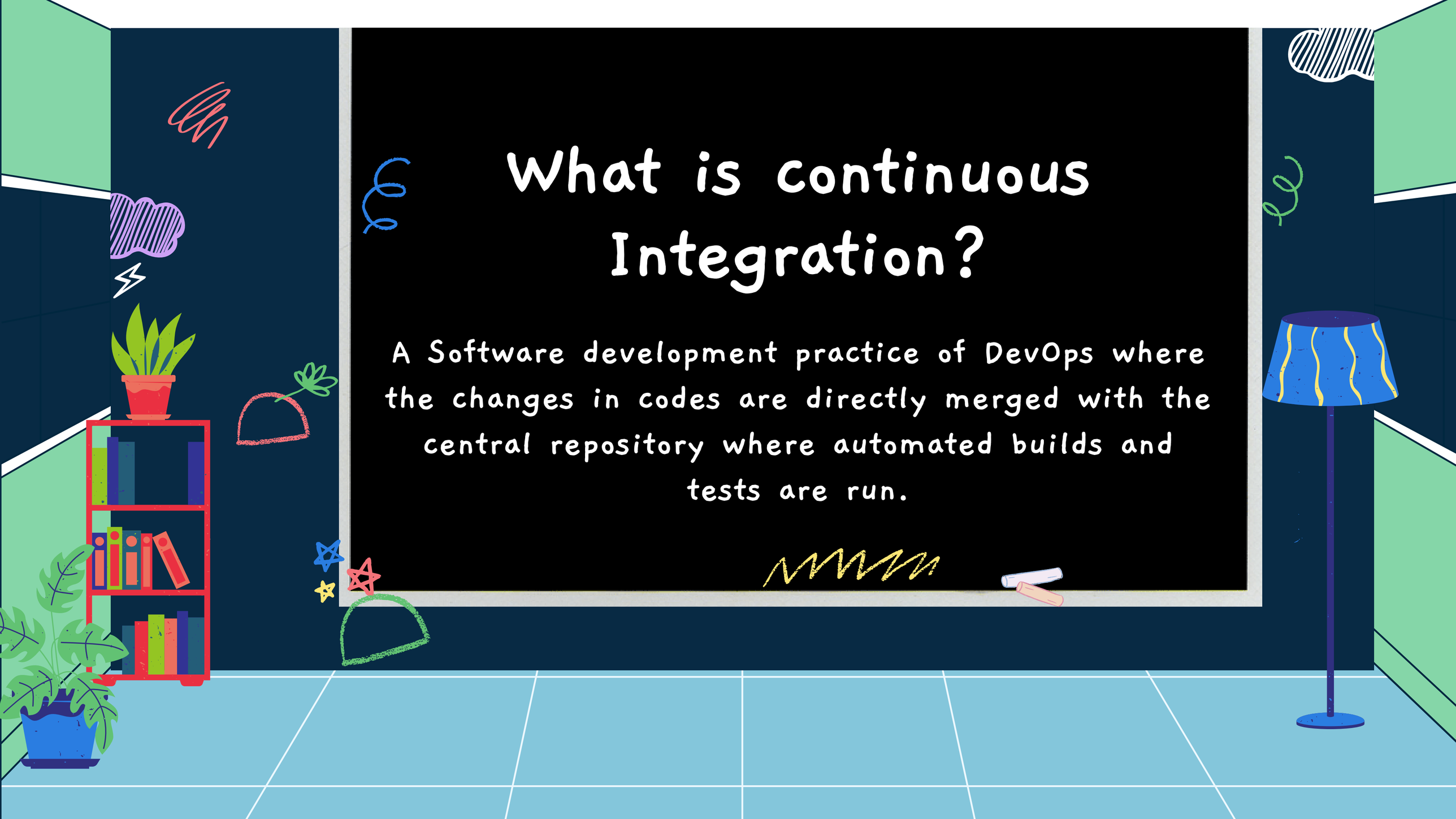
more comprehensive, automated acceptance tests

VALIDATION

decision of whether or not the product will be released

RELEASE MANAGEMENT

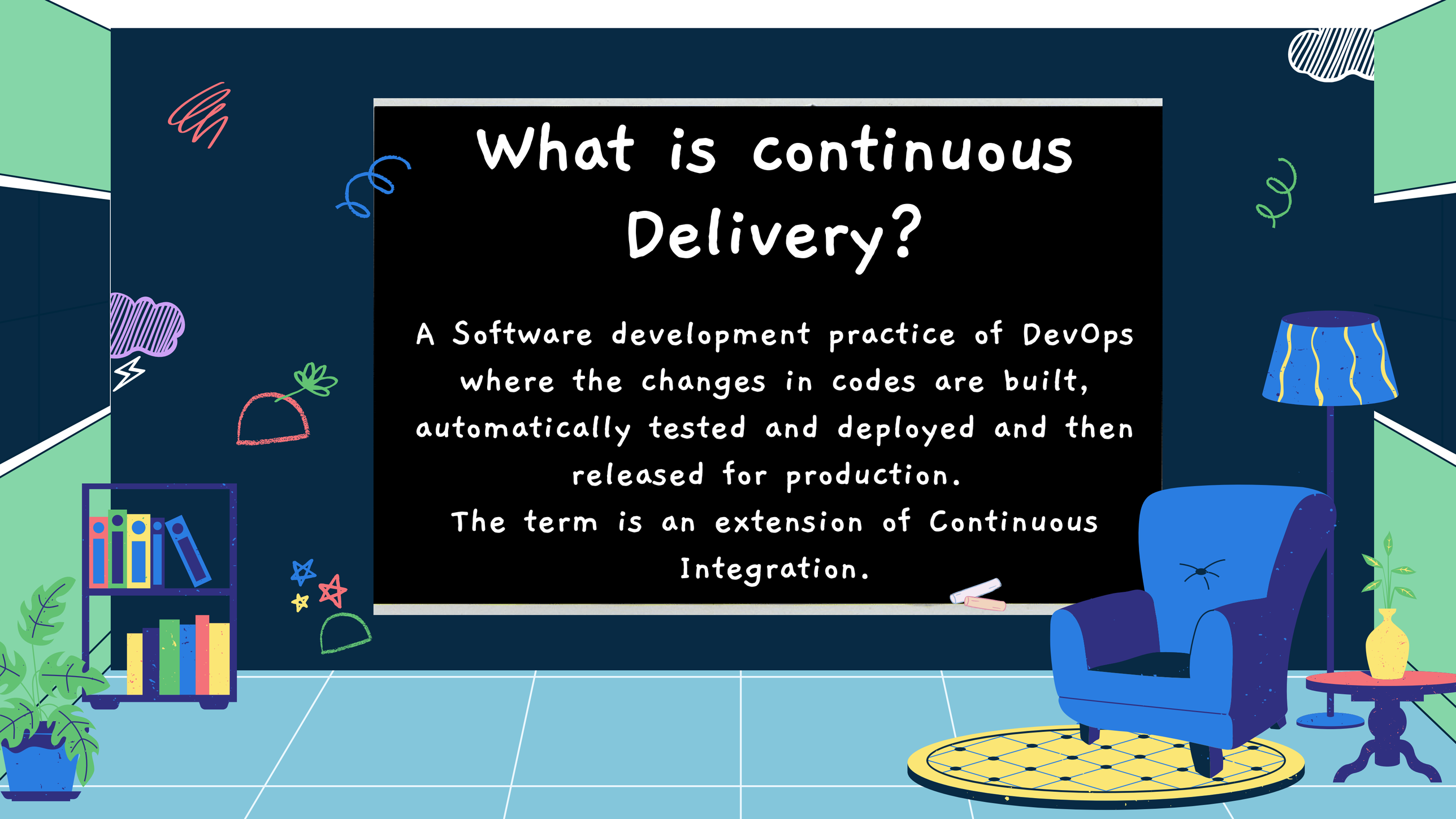
heading towards the operational team.


What is continuous Integration?
A Software development practice of DevOps where the changes in codes are directly merged with the central repository where automated builds and tests are run.

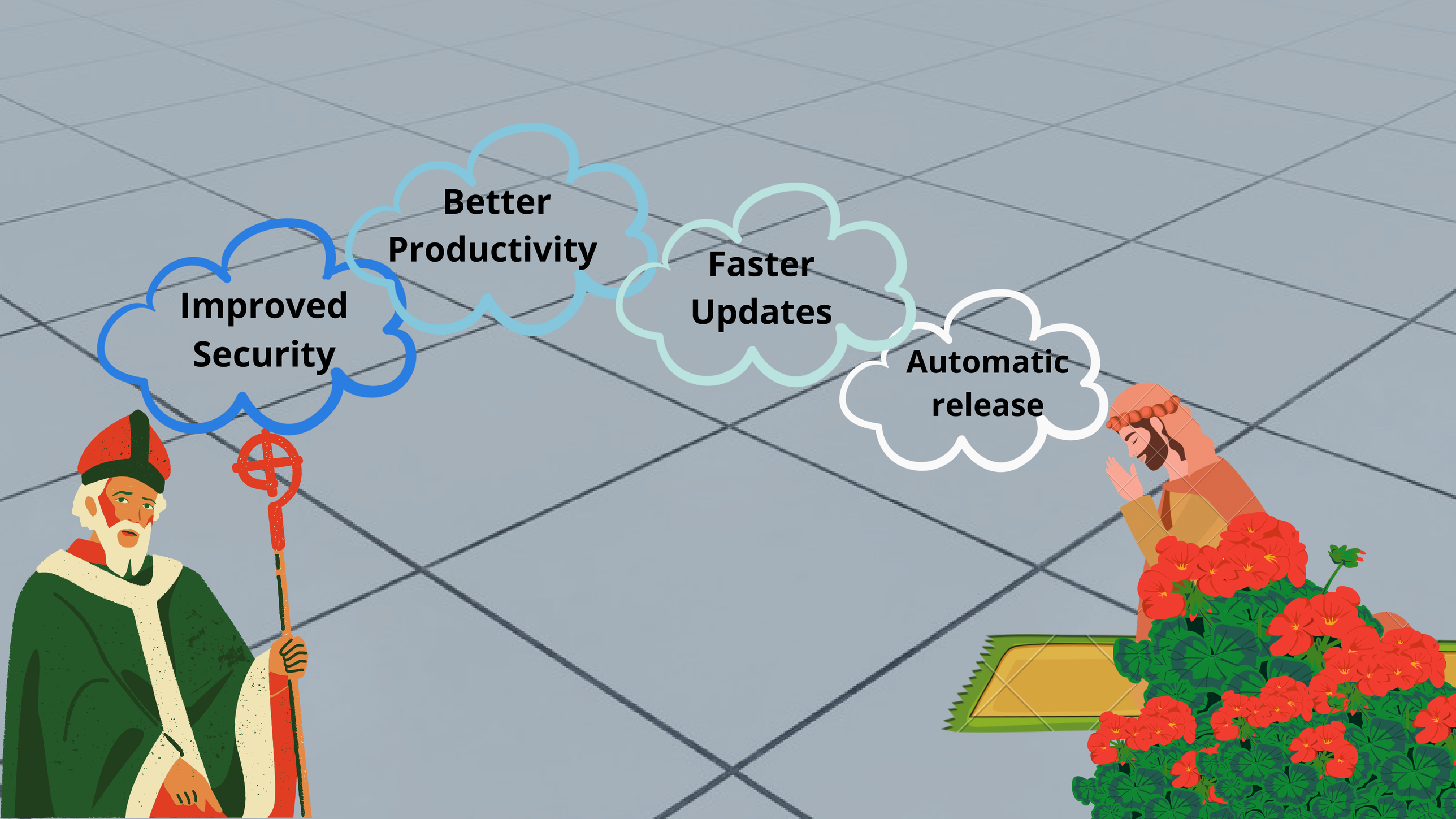
What is continuous Delivery?

A Software development practice of DevOps where the changes in codes are built, automatically tested and deployed and then released for production.

The term is an extension of Continuous Integration.



Hmm..
But, Why Continuous Delivery ?


Improved Security
Better Productivity
Faster Updates
Automatic release

What is Secure Coding?

Coding enabled with best security practices and compliances to safeguard the organization from existing and upcoming cyber threats.

Why secure coding?

More than 50 coding updates a day on an average in any organization.
With Secure Coding, mostly vulnerabilities are removed in the initial stage on which many exploits rely on.

OWASP Secure Coding Practices

Input Validation
Output Encoding
Authentication & Password Management
Session Management
Access Control
Cryptographic Practices
Error Handling & Logging
Data Protection
Communication Security
System Configuration
Database Security
File Management
Memory Management

PAIR PROGRAMING

Practice of developing codes in a pair where one drives the keyboard and other works on the upcoming parts.

PEER REVIEWING

Reviewing of code amongst the peers

Static Analysis Software Testing

BUILT-IN CHECKER FEATURE

Connecting plugins with the IDE to find common coding errors.

FIND SUBTLE MISTAKES

Which can be missed by other methods and developers.

CATCH ANTI-PATTERNS

wire the incremental static tool and scan the changed part of code.

Some Open Source Tools

VisualCodeGrepper

- Using this tool you can analyze most of the modern as well as the old popular programming language like C, C++, Java, PHP, COBOL, etc.
- Provides a nice pie chart for the entire codebase which shows relative proportions of code, whitespace, comments, and bad code.
- Performs many complex checks and allows you to add any bad functions that you want to search for with a config file for each language.
- You can run several scan operations depending upon the type and complexity of your project. Among the possible operations, it helps you to trigger a full scan process for code and during this process, a new window brought up instantly with chard displaying each component for better analysis.

RISP

- Language-specific static code analysis tool for PHP, Java, and Node.Js. It automatically detects the security vulnerabilities in PHP and Java applications and is an ideal choice for application development
- Supports all major PHP and Java frameworks and can be deployed as a self-hosted software or used as a cloud service. with SDLC integration and relevant industry standards.
- Tracks your application progresses throughout the development lifecycle and finds the risks and vulnerabilities in your code instantly so that you can fix the issues as soon as possible

Brakeman

- Free and open-source code vulnerability scanner and specially designed for the Ruby on Rails applications.
- Static code analyzer that scans the Rails application code to find security issues at any stage during development.
- Requires no prior setups or configuration once it is installed.
- Provides Flexible Testing, each check performed is independent, so testing can be flexible with Barkman,
- It is much faster than “black box” website scanners and even the large applications can be scanned within a few minutes

Bandit

- Free SAST tool especially designed to find common security issues in Python code.
- Processes each file with appropriate plugins and generates a detailed report of possible security bugs in the python code.
- Command-line interface to scan your python code.
- Allows specifying the path of a baseline report for ignoring known vulnerabilities that you believe are non-issues.
- Allows users to write and register extensions for checks and formatters.

Tell me and I forget, teach me and I may remember, involve me and I learn.

- Benjamin Franklin

Thank you for your time!!