

- Learn about configuration management in an Agile and DevOps world
- Discover how new technology decreases the overhead of configuration management
- Learn how to use automated methods to describe software configuration
- Understand where to integrate these automated methods into the existing manual processes.
- Understand how to Integrate auditors into the process to ensure artifacts are acceptable for IV&V

The idea of configuration management is not new in the software industry, some organizations even have dedicated a role or position for it. However, in recent years technology has changed, making configuration management not just an organizational process, but a technological one. With the increase in the velocity in which software comes to market, this class explores the human processes that need to change along with the tools that are necessary to collect, represent, and make decisions on the large amount of information the software development process generates.

Who Should Attend

This course is appropriate for Configuration Managers, Project Managers, Developers, Product Owners, Agile Developers, Auditors and DevOps Engineers who have a high level knowledge of the Software Development Lifecycle (SDLC) and Continuous Integration/Continuous Development pipelines. Familiarity with the high level concept of infrastructure as code is also helpful.

Course Outline

History of Configurations Management

Saturn V
Reproducibility
Software complexity and Why it Needs Configuration Management

What is Software

Why the Developer IDE is important
Develop Locally, test locally
Unit tests and test coverage
What shifting left means to developers

Requirements and traceability in the age of Agile

SCM in General and Versioning
It Starts at Code Check in
GIT Flow

How code is Turned into Software

Compiled Languages
Interpreted Languages
Java

Testing

Weyuker's Axioms

Security

Constant Checking and Scanning

Packaging

Maven, Gradle, Ant/Ivy, and Binary Dependencies
Artifact Repositories
Release Packaging and Traceability
Different Package Types and What They Mean
NPM and Non-Binary Dependencies
Original Code Vs Third Party Code in the Age of Open Source

Where Software Lives

The Platform
The Stack
The Application

Persistent Data

Database Development
Database Versioning

CI/CD

The Role of Automation in DevOps
How to Keep Things Organized When you Move Fast

Mapping Test to Change
Integration Testing

Semantic Versioning

Changing the Tires While the Car is Moving

API Versioning

Blue/Green Deployments

Feature Flags

When am I done?

Conclusions

Auditing

Tracking all the Different Pieces

Keeping the Information useful