KOENIG
step forward

Red Hat
Training Partner

TL500

Red Hat Training: DevOps Culture and Practice Enablement

Red Hat Training: DevOps Culture and Practice Enablement (TL500) is a five-day, immersive class offering students an opportunity to experience and implement cultural shifts that are utilized in many successful DevOps adoption journeys. Many agile training offerings focus on a particular framework, delivery mechanism, or technology. Instead, DevOps Culture & Practice combines the best tools from many leading frameworks to blend continuous discovery and continuous delivery with cultural and technical practices into a unique, highly-engaging experience simulating real-world scenarios and applications.

To achieve the learning objectives, participants should include multiple roles from an organization. Business product owners, architects, developers, and site reliability engineers will gain the experience of working outside of their traditional silos. The daily routine simulates a real-world delivery team, where cross-functional teams learn how collaboration breeds innovation. Armed with shared experiences and best practices, the team can apply what it has learned to help the organization's culture and mission succeed in the pursuit of new projects and improved processes.

This course includes a copy of the Red Hat Training: Open Practices for your DevOps Journey course book, which can be used as a resource as students take the learnings from this course and apply them to other real world scenarios.

What are the prerequisites?

- Knowledge of agile practices is helpful
- Experience using agile practices and methodologies such as scrum is beneficial

KOENIG
step forward

Red Hat
Training Partner

Outline for this course

What is DevOps?

Brainstorm and explore what principles, practices, and cultural elements make up a DevOps model for software design and development.

Collaborative practices to establish culture and shared understanding

Learn and experience practices that facilitate great conversation and alignment across stakeholder groups such as priority sliders, pair programming, mob programming, conducting retrospectives, visualizing work, assessing team sentiment, and performing agile estimation.

Understanding the Why and Who of software delivery

Use the impact mapping discovery practice to connect deliverables to measurable impact. Learn how to use human-centered design, design thinking, and Lean UX to develop empathy with users and stakeholders.

Domain-driven design and storytelling

Learn and practice the powerful Event Storming tool to visualize and map event-driven systems to produce emergent architectures for iterative and incremental delivery.

Prioritization and pivoting

Experience the collection of ideas, aligning them to target outcomes, and using economic prioritization practices and value slicing to build product backlogs that can deliver incremental value.

Agile practices

Cover agile delivery practices, including Kanban, Scrum, sprint planning, daily standup, showcase, retrospective, and backlog refinement.

Design of experiments

Set up, execute, and measure the results of experiments by utilizing platform's advanced deployment features, including A/B Testing, Blue/Green Deployments, Feature Toggles, Dark Launches, and Canary Deployments.

Value stream and process mapping

Delve into the practices of value stream mapping and metric-based process mapping to establish non-functional improvements that you can make to product delivery and execution of value streams.

Continuous integration, deployment, and delivery

Explore the foundational practices of continuous integration, continuous deployment, and continuous delivery.

Non-functional requirements

Learn how to elaborate non-functional areas that are unlikely to be captured by using practices primarily focused on the functional aspects of a solution.

Testing

Develop an understanding of test-driven development and business-driven development foundational practices, often called automated testing.

Everything as code and GitOps

Explore continuous integration/continuous delivery pipelines using Jenkins and Tekton and sing a GitOps approach to codify everything for repeatability. Experience how to extend pipelines to cover non-functional testing, monitoring, and observability.