TL500
Red Hat Training: DevOps Culture and Practice Enablement

What are the prerequisites?

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

Technology requirements

- This class will require internet access to certain sites (AWS, youtube, google drive, online chat tools)
- This course requires you to "bring your own developer workstation" (BYDW). so you will be expected to bring your own device
- Chrome browser suggested

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.