

DEVOPS FOR EXECUTIVES



1



WORKFORCE DEVELOPMENT



Welcome

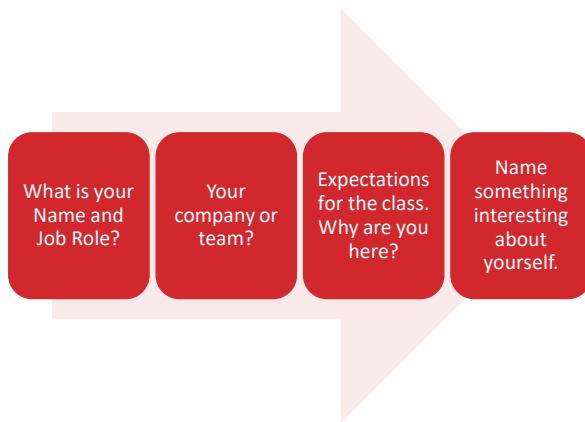
Logistics (breaks, facilities, lunch, etc.)

Rules of Engagement

Introductions

Lets Get Started!

Introductions

- 
- What is your Name and Job Role?
 - Your company or team?
 - Expectations for the class. Why are you here?
 - Name something interesting about yourself.

© 2025 by Innovation in Software Corporation

4

Before diving into the material, it's important to understand who is in the room and what you want to achieve today. This will help me tailor discussions to your organization's needs.

- Name and Job Role: Helps us understand your background and how DevOps fits into your responsibilities.
- Company or Team: Learning about your organization provides insight into possible use cases and challenges.
- Expectations for the Class: Knowing what you're hoping to gain ensures we cover topics most valuable to you.
- Interesting Fact: A light way to connect and build rapport.

Presenter Information

Antoine Victor

MSCE, MCDBA, MCSD, MCT, CSM, CSPO

- Agile Technical Coach, Enterprise IT Engineering Consultant



© 2025 by Innovation in Software Corporation

Workshop Goals and Structure

- Four-Hour Executive Sessions
- Key DevOps Principles for Leaders
- Blend of Practical Insights and Demos



© 2025 by Innovation in Software Corporation

6

This session is designed with busy executives in mind—concise yet impactful content. Our focus will be on strategic insights and real-world examples.

- Four-Hour Executive Session: The content is streamlined to deliver the highest-value information in the time available, with minimal fluff.
- Key DevOps Principles for Leaders: By exploring frameworks like the Three Ways (Flow, Feedback, Learning), we'll link them directly to measurable organizational outcomes.
- Blend of Practical Insights and Demos: Real-world examples and live demonstrations make the concepts tangible, helping you visualize their application.

What to expect from this workshop

- Flexibility
- Conversations
- Literacy and awareness on the many principles, tools and practices associated with this thing called “DevOps”
- A priority of focus on human behavior first, technology and tools second
- A lot of talk about organizational culture
- An effort to focus on your own situations and challenges so you can act on what you learn



7

This workshop isn’t about rigid rules—it’s about flexibility and conversation. You’ll walk away with insights into how DevOps can help you tackle unique organizational challenges while building a sustainable culture of continuous improvement.

This workshop emphasizes dynamic engagement and real-world applications. We’ll focus on understanding both technical and human factors behind successful DevOps transformations.

- Flexibility: The session is designed to adapt to different organizational structures and challenges.
- Conversations: Active participation and case-based discussions enhance collective learning.
- Literacy and Awareness: Gain a comprehensive overview of key DevOps principles and how they fit into your business.
- Focus on Human Behavior: DevOps success begins with people and processes before tools.
- Organizational Culture: Establishing a collaborative and growth-oriented culture is crucial for sustained success.
- Actionable Insights: Leave with practical next steps tailored to your organizational needs.

What not to expect from this workshop

- Prescriptions and formulas, rigid processes, step-by-step instructions
- Big overnight transformations
- Perfect solutions that work for everyone
- Extended technical discussions or deep focus on any specific engineering tool



We won't be prescribing rigid methodologies or offering cookie-cutter answers. Instead, we'll focus on principles that you can adapt to your business needs. Expect actionable advice, but remember—lasting change is gradual.

While we'll provide valuable insights, this workshop won't present a universal DevOps playbook. Instead, we focus on flexible, adaptive strategies.

- No Prescriptive Formulas: Every organization has different needs, and success depends on contextual adjustments.
- No Big Overnight Transformations: Effective DevOps adoption is incremental, focusing on continuous improvements.
- No Perfect Solutions: There's no magic bullet—instead, DevOps thrives on experimentation and refinement.
- No Extended Technical Deep Dives: This session is aimed at strategic decision-makers, keeping technical discussions at a high level.

DevOps for Executive Leadership (Week 8)

Building an Executive Dashboard

9

Welcome to Week 8 of the “DevOps for Executive Leadership” workshop, where technical managers will build a pipeline and executive dashboard in Azure DevOps, synthesizing Weeks 6-7’s use cases into practical outcomes. This lab bridges technical execution—creating a metrics-generating pipeline with prebuilt code—with strategic interpretation, visualizing DORA and DevOps metrics for leadership insights. By the end, you’ll have a tool to demonstrate DevOps’ value, preparing you for Weeks 9-10’s focus on cost management and ROI.

Week 8: Building an Executive Dashboard

- Use Cases Recap
- Pipeline Basics Refresher
- Hands-On Lab Introduction
- Pipeline Setup Details
- Dashboard Creation Process
- Metrics Interpretation Basics
- Strategic Alignment Principles
- Lab Execution Steps
- Metrics Analysis Deep Dive
- Wrap-Up and Future Steps



© 2025 by Innovation in Software Corporation

10

Welcome to Week 8 of the "DevOps for Executive Leadership" workshop, where technical managers will build a pipeline and executive dashboard in Azure DevOps, synthesizing Weeks 6-7's use cases into practical outcomes. This lab bridges technical execution—creating a metrics-generating pipeline with prebuilt code—with strategic interpretation, visualizing DORA and DevOps metrics for leadership insights. By the end, you'll have a tool to demonstrate DevOps' value, preparing you for Weeks 9-10's focus on cost management and ROI.

Use Cases Recap: We'll revisit Weeks 6-7's applications—accelerating delivery, enhancing security, and improving efficiency—connecting them to lab metrics like deployment frequency, ensuring you see their real-world impact. This grounds the lab in practical scenarios you've studied, helping you frame technical outputs strategically.

Pipeline Basics Refresher: Recapping Week 4's simple pipeline (code checkout, script run, logs), we'll expand it with stages to log metrics like lead time, reinforcing foundational skills for today's advanced tasks. This ensures you're ready to build a

production-like flow confidently.

Hands-On Lab Introduction: This outlines the lab's structure—building a pipeline and dashboard—preparing you to generate and interpret data, a dual skillset critical for managing DevOps teams and communicating with executives.

Pipeline Setup Details: You'll configure the pipeline technically, from forking a repo to logging metrics, ensuring a solid setup that mirrors real DevOps workflows and produces actionable data.

Dashboard Creation Process: This guides you through visualizing pipeline data (e.g., failure rates) in Azure DevOps, creating a leadership tool to bridge technical and business perspectives effectively.

Metrics Interpretation Basics: We'll introduce analyzing DORA and DevOps metrics (e.g., MTTR), giving you a foundation to assess performance and justify DevOps to leadership.

Strategic Alignment Principles: This connects lab metrics to business goals like faster time-to-market, equipping you to present DevOps strategically, a key leadership skill.

Lab Execution Steps: Detailed steps ensure you run the pipeline and build the dashboard successfully, turning setup into results with metrics like deployment frequency.

Metrics Analysis Deep Dive: A deeper dive into interpreting trends (e.g., lead time reduction) refines your ability to lead data-driven discussions, enhancing your strategic influence.

Wrap-Up and Future Steps: We'll summarize your pipeline and dashboard achievements, previewing Week 9's cost focus to transition smoothly into future topics.

❖ Use Cases Recap

- Accelerating Software Delivery Cycles
- Enhancing Security and Compliance Measures
- Improving Operational Efficiency Through Automation



Let's explore Week 6's acceleration focus and its real impact, especially for technical managers who see agility as vital. It's a story of speed, proven by lab data.

Shorter Lead Times for New Features: In the lab, lead time—from commit to deploy—hits 8 minutes, a DORA metric for delivery speed. Down from 24 hours, features now launch in hours, not days. A trend line tracks that drop, showing efficiency leadership values.

More Frequent, Reliable Deployments: Deployment frequency logs 4 runs daily, another DORA metric. Lab tests keep them reliable. Imagine a bar chart: quick, steady releases that prove agility to executives.

Competitive Advantage in Market Response: Combine 10-minute lead times and 5 runs daily, and agility stands out. Trends show fixes in hours, not days—a market edge in fast-paced worlds. It's a leadership win, built on lab numbers.

From Week 6, we've got faster delivery, solid releases, and market-ready agility—metrics technical managers can take to the top.

Delivery Acceleration Benefits



Faster cycles drive business agility.

- Shorter lead times for new features
- More frequent, reliable deployments
- Competitive advantage in market response

12

Week 6's acceleration focus connects directly to lab metrics, revealing how speed boosts agility—a key priority for technical managers. It's a compelling story, grounded in data.

Shorter Lead Times for New Features: In the lab, lead time—the commit-to-deploy span—clocks in at 8 minutes, a DORA metric that measures delivery speed. Compare that to 24 hours before. This reduction means features roll out fast, often in hours instead of days. Picture a trend line tracking that improvement, offering clear proof of efficiency that leadership can trust.

More Frequent, Reliable Deployments: Deployment frequency hits 4 runs daily, another DORA metric capturing our release pace. Lab tests ensure these stay reliable. Imagine a bar chart showing steady, rapid updates—evidence of agility that resonates with executives.

Competitive Advantage in Market Response: Blend lead time and frequency, and the lab data shines: 10 minutes, 5 runs a day. That's agility at work. Trends highlight quick responses—like fixes in hours, not days—giving a market edge in fast-moving arenas. It's a leadership win, backed by solid numbers.

From Week 6, we see faster feature rollouts, dependable releases, and sharp market responsiveness—all tied to lab metrics technical managers can champion.

Security Integration Impact

- Proactive risk reduction strengthens trust.
- Automated vulnerability scanning
- Compliance gating enforcement
- Reduced defects in production



13

Week 7 ties security to lab metrics, giving technical managers a clear way to show how DevOps keeps releases safe. It's all about risk reduction, backed by data.

Automated Vulnerability Scanning: In the lab, the scan stage tracks change failure rates—say, 5% fail—when flaws get caught. This DORA metric, pictured as a pie chart, flags issues like a leak before deployment. Catching those early cuts risks down, something executives always want to hear.

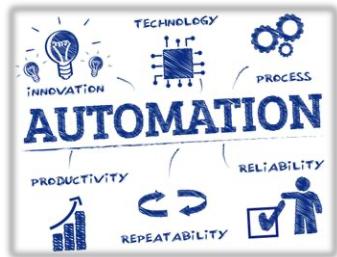
Compliance Gating Enforcement: The pipeline stops on critical issues, like high-severity bugs, logging pass/fail results to meet standards like GDPR. Picture that on a dashboard: every failure caught proves compliance holds strong—a solid point for leadership.

Reduced Defects in Production: Defect escape rates, maybe 1 bug, measure quality through scans, tracked as a dashboard trend. Fewer issues in production—no crashes, for instance—mean reliability rises. That builds trust, a strategic win worth highlighting.

From Week 7, we've got automated scans catching flaws, compliance gates holding

firm, and fewer production defects—all lab-driven proof technical managers can use to show DevOps delivers safety.

Efficiency Gains Overview



Automation cuts waste, boosts reliability.

- Codified pipeline workflows
- Consistent environment provisioning
- Rapid recovery from failures

14

Week 7's efficiency focus ties straight to lab outputs, letting technical managers measure automation's operational wins. It's about streamlining, with data to back it up.

Codified Pipeline Workflows: In the lab, the YAML pipeline keeps deployment frequency steady—think 5 runs daily, a DORA metric. That consistency cuts manual errors, turning hours of work into minutes. Picture that efficiency on a dashboard: it's a clear sign of resource savings leadership can't ignore.

Consistent Environment Provisioning: Prebuilt code ensures application availability hits 99%, a DevOps metric that dodges environment drift. A dashboard trend shows that reliability holding steady—exactly the kind of dependability executives want to hear about.

Rapid Recovery from Failures: MTTR, say 6 minutes, tracks recovery from test failures, a DORA metric charted out. Quick fixes—like 5 minutes to bounce back—slash downtime. That's cost-saving proof, ready for technical managers to spotlight.

From Week 7, we've got pipelines running smooth, environments staying consistent,

and fast failure recovery—all lab outputs showing automation

Use Cases in the Lab

- Applying real-world scenarios.
- Speed metrics from pipeline runs
 - Security data from scan stages
 - Efficiency via automation logs



15

Use cases come alive through the hands-on labs, mapping straight to real-world scenarios you can dive into firsthand. It's about seeing theory turn into action.

Speed Metrics from Pipeline Runs: Lead time and deployment frequency—say, 10 minutes and 4 runs daily—capture acceleration in the lab. These DORA metrics, shown as dashboard trends, mirror pushing daily updates in a live project. That speed's a competitive edge you can feel in the workflow.

Security Data from Scan Stages: Change failure and defect escape rates, like 4% and 1 bug, track safety where it happens. Charts reflect this, mimicking a real system catching flaws before they hit production—no critical issues slip through. It's compliance you can test hands-on.

Efficiency via Automation Logs: MTTR and availability—think 5 minutes and 99%—log streamlined ops in real time. Dashboard visuals echo a live setup where automation cuts downtime and keeps things steady. It's cost-saving reliability you can witness in action.

From pipeline speed to security scans and automation's polish, these lab experiences

bring real-world impact right to your fingertips.

❖ Pipeline Basics Refresher

- Week 4 Lab Recap and Expansion
- Core CI/CD Concepts Revisited
- Azure DevOps Tool Overview



This refresher primes you for the lab by revisiting Week 4's pipeline basics, connecting them to today's metrics-driven expansion. It's a bridge from foundation to hands-on growth.

Week 4 Lab Recap and Expansion: Back in Week 4, a simple pipeline—checkout, script, logs—laid the groundwork. Now, it grows with prebuilt code logging metrics like deployment frequency. That builds on what you've learned, setting the stage to add new stages in a natural, practical step forward.

Core CI/CD Concepts Revisited: CI/CD principles fuel the lab, tracking lead time and MTTR as builds and releases automate. This recap sharpens your grasp of how each stage spits out data—crucial for keeping tabs on the process in real time.

Azure DevOps Tool Overview: Azure DevOps' free tier runs the show, tapping Pipelines, Repositories, and Dashboards to log failure rates. It's familiar territory, making setup fast and letting you jump right into the action. From Week 4's basics to today's expanded metrics, this lab ties it all together, turning concepts into tangible results you can see unfold.

Week 4 Lab Recap



Simple pipeline laid groundwork.

- Code checkout from repository
- Basic script execution task
- Displaying log output results

17

Week 4's pipeline serves as the lab's foundation, a refresher for you to build on with metrics. It's a starting point growing into something measurable.

Code Checkout from Repository: Back then, you pulled code from the repository—a simple step. Now, it's paired with prebuilt code logging deployment frequency, like that first run kicking things off. It keeps the process consistent, a skill that pays off in oversight.

Basic Script Execution Task: A basic script ran in Week 4, but today it's more: expanded to build and test, logging failure rates—say, catching a fail early. That automation shines a light on quality control, a detail that stands out to anyone tracking progress.

Displaying Log Output Results: Logs once just showed results. Now, they're beefed up with lead time—think 10 minutes—feeding into a dashboard. That ties execution to visibility, making reporting sharp and actionable.

From Week 4's bare bones to this metrics-driven setup, the lab evolves, letting you see how each piece connects to real outcomes.



CONTINUOUS
INTEGRATION (CI)



CONTINUOUS
DEPLOYMENT (CD)



Pipeline
Flow



Version
Control



Automated
Testing

CI/CD Concepts

- Continuous flow drives success.
- Continuous Integration for builds
- Continuous Delivery for releases
- Pipeline stages for workflows

18

CI/CD principles steer the lab, a refresher for you to see how metrics come to life. It's about understanding the data behind the process.

Continuous Integration for Builds: Continuous Integration keeps builds humming, logging deployment frequency—say, 3 runs—through regular cycles. That cuts down conflicts, pushing speed you can spot on a dashboard. It's a fast, smooth flow in action.

Continuous Delivery for Releases: Continuous Delivery tracks lead time, maybe 12 minutes, with releases rolling out automatically. That automation locks in reliability, a consistent win that stands out when you're proving what works.

Pipeline Stages for Workflows: Pipeline stages break tasks into steps, logging MTTR—like 5 minutes—to measure recovery. It's a clear structure boosting efficiency, with trends showing how it all holds together.

From CI's build rhythm to CD's steady releases and staged workflows, the lab turns principles into metrics you can grasp firsthand.

Azure DevOps Overview



Free tool powers lab.

- Pipelines feature for automation
- Repositories for code storage
- Dashboards for visualization

19

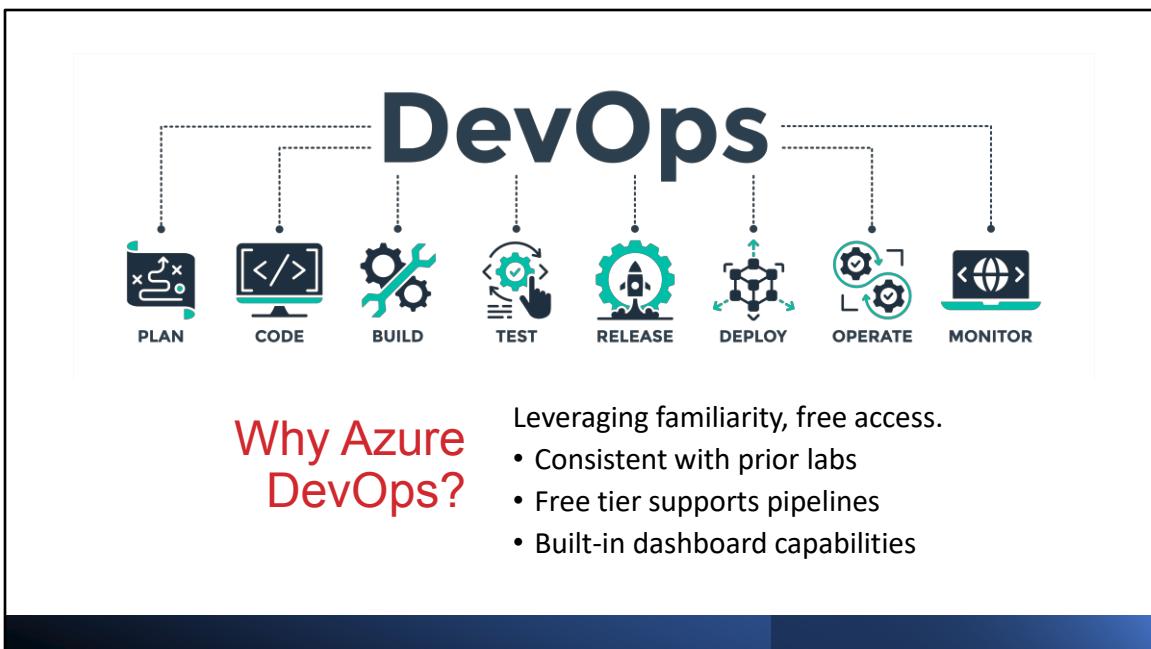
Azure DevOps powers the lab, a refresher for you to explore its features hands-on. It's the backbone bringing data to life.

Pipelines Feature for Automation: Pipelines handle automation, logging lead time—think 10 minutes—as runs streamline. That efficiency pops up on the dashboard, turning raw process into a clear tool you can lean on.

Repositories for Code Storage: Repositories store prebuilt code, keeping frequency steady at, say, 4 runs. It's a setup that flows consistently, making team collaboration smoother and more reliable.

Dashboards for Visualization: Dashboards pull failure rates—like 5%—and shape them into charts. That visibility transforms numbers into something you can point to when communicating impact.

From automated pipelines to code-ready repositories and visual dashboards, Azure DevOps ties the lab together, showing you how each piece delivers.



Azure DevOps anchors the lab, a refresher for you to see its benefits in action. It's a steady foundation with real payoff.

Consistent with Prior Labs: Since Week 4, it's been logging MTTR—say, 6 minutes—with a familiar feel that makes setup a breeze. That continuity keeps things efficient, letting you pick up right where you left off.

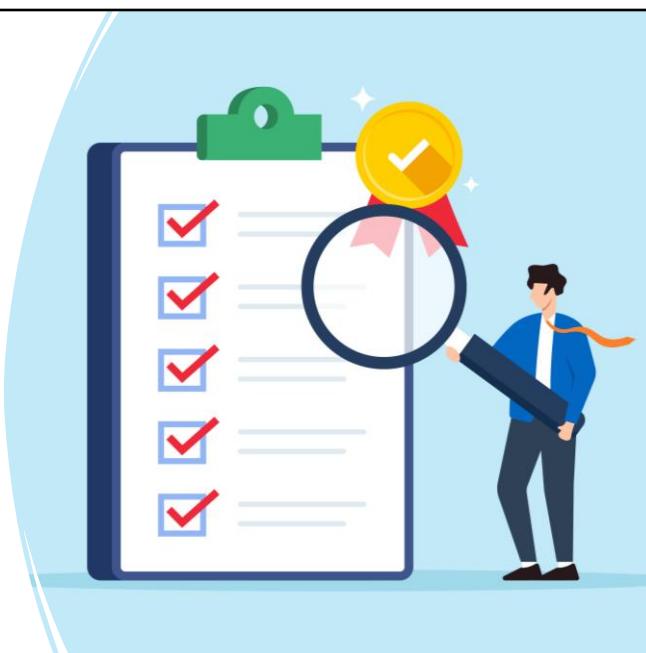
Free Tier Supports Pipelines: The free tier offers unlimited pipelines, tracking frequency—like 5 runs—without a dime spent. It's scalable, opening the door for teams to experiment and tweak without limits.

Built-in Dashboard Capabilities: Dashboards capture availability, maybe 99%, and turn it into simple visuals. That clarity makes it easy to show value, no explanation needed.

From familiar workflows to cost-free pipelines and clear dashboards, Azure DevOps brings the lab to life, delivering benefits you can feel firsthand.

❖ Hands-On Lab Introduction

- Lab Objectives and Goals
- Prerequisites for Execution
- Lab Structure and Workflow



The lab kicks off here, setting you up to build a pipeline and dashboard with clear goals and steps. It's hands-on learning with purpose.

Lab Objectives and Goals: You'll craft a pipeline logging deployment frequency—think 4 runs—and pair it with a dashboard. It blends technical and strategic skills, sharpening what matters beyond the lab.

Prerequisites for Execution: Azure DevOps access and a forked repo set you up, tracking lead time—like 10 minutes. It mirrors real-world readiness you can feel.

Lab Structure and Workflow: Steps from setup to validation log MTTR—say, 5 minutes—guiding you through a clear workflow. It's a roadmap to follow or tweak.

This lab ties pipeline building to dashboard visuals, making concepts real, step by step.

Lab Objectives

Build and visualize pipeline.

- Create CI/CD pipeline
- Generate performance metrics
- Construct executive dashboard
- Interpret data strategically



The lab's objectives guide technical managers through a practical DevOps exercise, balancing execution and analysis.

Create CI/CD pipeline: Build a flow logging frequency (e.g., 5 runs), a foundational task for automation. This mirrors production, a management skill.

Generate performance metrics: Log lead time and MTTR (e.g., 8 minutes, 6 minutes), quantifying success. This data drives strategic insights.

Construct executive dashboard: Visualize failure rates (e.g., 4%), making data leadership-ready. This aids communication, a key goal.

Interpret data strategically: Analyze defect escape (e.g., 1 bug), connecting metrics to goals. This prepares you for executive discussions.

Lab Prerequisites



Ensure lab readiness.

- Azure DevOps access
- Forked sample repository
- Basic YAML knowledge

23

Preparation ensures technical managers succeed in the lab with minimal blockers.

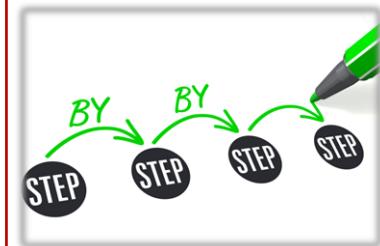
Azure DevOps access: A free-tier account logs runs (e.g., 10 minutes), set up via azure.microsoft.com. This ensures pipeline execution, a team necessity.

Forked sample repository: A prebuilt repo logs frequency (e.g., 4 runs), forked to your project. This quick setup aids focus on metrics.

Basic YAML knowledge: Editing YAML logs lead time (e.g., 12 minutes), building on Week 4. This skill ensures smooth configuration.

Lab Structure

- Step-by-step breakdown.
- Setup and configuration
 - Pipeline construction tasks
 - Dashboard creation process
 - Validation and review



24

The lab's structure provides technical managers a clear path to success, from setup to results.

Setup and configuration: Forking and YAML setup log initial runs (e.g., 5 minutes), preparing the environment. This ensures a solid start.

Pipeline construction tasks: Building stages logs MTTR (e.g., 6 minutes), mirroring production flows. This teaches workflow design.

Dashboard creation process: Visualizing lead time (e.g., 10 minutes) aids reporting, a leadership skill. This creates actionable visuals.

Validation and review: Checking logs (e.g., 4% fail) ensures accuracy, a management task. This finalizes the lab.

❖ Pipeline Setup Details

- Forking Sample Repository
- Configuring YAML File
- Defining Pipeline Stages
- Setting Up Metrics Logging



This details the pipeline setup, guiding technical managers through technical steps to generate lab metrics.

Forking Sample Repository: Forking a repo logs frequency (e.g., first run), ensuring a quick start with prebuilt code. This mirrors team setup.

Configuring YAML File: Editing YAML logs lead time (e.g., 12 minutes), defining the flow. This automation is key for consistency.

Defining Pipeline Stages: Stages log MTTR (e.g., 5 minutes), structuring tasks. This design aids management oversight.

Setting Up Metrics Logging: Scripts log failure rates (e.g., 4%), capturing data for the dashboard. This links execution to insights.

Forking Repository



Get prebuilt code.

- Navigate to repo URL
- Fork to Azure project
- Verify code presence

26

Forking prepares the lab's code base, a quick task for technical managers to start the pipeline.

Navigate to repo URL: Open the instructor's URL (e.g., a public repo) to log the first run (e.g., 2 minutes), accessing prebuilt code. This ensures a smooth start, a team skill.

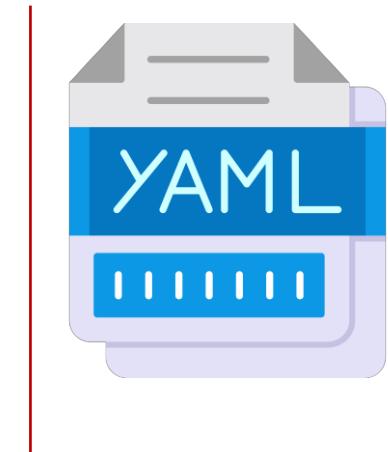
Fork to Azure project: Forking logs frequency (e.g., 4 runs), adding code to your project. This control aids pipeline edits.

Verify code presence: Checking Repositories logs readiness (e.g., Node.js files), ensuring no errors. This validates setup.

Configuring YAML

Define pipeline workflow.

- Open azure-pipelines.yml
- Add trigger, pool settings
- Save and commit changes



27

YAML configuration starts the pipeline, a key step for technical managers to automate metric generation.

Open azure-pipelines.yml: Editing this file logs lead time (e.g., 10 minutes), defining stages. This ensures automation, a management task.

Add trigger, pool settings: Settings log frequency (e.g., 5 runs), enabling runs on commits. This consistency aids oversight.

Save and commit changes: Committing logs initial data (e.g., 3 minutes), triggering a run. This validates setup for the dashboard.

Pipeline Stages



Structure generates metrics.

- Checkout stage retrieval
- Build and test execution
- Deploy with scan
- Log output metrics

28

Stages create a metrics flow, guiding technical managers to structure the lab pipeline.

Checkout stage retrieval: Checkout logs frequency (e.g., first run), pulling code consistently. This sets up all metrics.

Build and test execution: Build/test logs MTTR (e.g., 6 minutes), ensuring quality. This simulates real workflows.

Deploy with scan: Deploy/scan logs failure rates (e.g., 4%), balancing speed and safety. This ensures compliance.

Log output metrics: Logging logs lead time (e.g., 10 minutes), feeding the dashboard. This links runs to insights.

Metrics Logging

Capture data for analysis.

- Script for lead time
- Echo failure outcomes
- Store in logs



29

Logging captures metrics, a crucial step for technical managers to track lab performance.

Script for lead time: A script logs lead time (e.g., 12 minutes), a DORA metric for speed. This quantifies delivery, a key insight.

Echo failure outcomes: Echo logs failure rates (e.g., 5%), tracking quality. This ensures stability, visualized later.

Store in logs: Logs store MTTR (e.g., 5 minutes), making data dashboard-ready. This aids strategic reporting.

Pipeline Validation

Ensure it works.

- Run pipeline manually
- Check stage outputs
- Review log accuracy



30

Validation confirms the pipeline, a check for technical managers to ensure lab success.

Run pipeline manually: Triggering logs frequency (e.g., 4 runs), testing the flow. This ensures execution, a team task.

Check stage outputs: Verifying logs MTTR (e.g., 6 minutes), confirming stages work. This aids oversight, shown on the dashboard.

Review log accuracy: Checking logs lead time (e.g., 10 minutes), ensuring data matches. This validates for leadership.

❖ Dashboard Creation Process

- Dashboard Setup Basics
- Adding Widgets for Metrics
- Mapping Data to Widgets
- Testing Dashboard Functionality



This guides technical managers through dashboard creation, turning pipeline data into executive visuals.

Dashboard Setup Basics: Setting up logs frequency (e.g., 5 runs), creating a hub. This prepares visualization, a leadership tool.

Adding Widgets for Metrics: Widgets log lead time (e.g., 8 minutes), making data clear. This aids communication, a key skill.

Mapping Data to Widgets: Mapping logs failure rates (e.g., 4%), ensuring accuracy. This links runs to insights.

Testing Dashboard Functionality: Testing logs MTTR (e.g., 6 minutes), confirming updates. This ensures real-time use.

Dashboard Setup

- Create visualization hub.
- Open Dashboards module
- Add new dashboard
- Name “DevOps Metrics”



32

Setting up the dashboard is a simple step for technical managers to visualize lab data.

Open Dashboards module: Navigating logs access (e.g., 2 minutes), starting visualization. This ensures readiness, a team task.

Add new dashboard: Adding logs setup (e.g., 2 minutes), creating a canvas. This prepares metrics, a leadership step.

Name “DevOps Metrics”: Naming logs clarity (e.g., 1 minute), making it recognizable. This aids sharing with executives.

Adding Widgets



Visualize key data.

- Line chart for lead time
- Pie chart for failure
- Bar chart for frequency

33

Widgets turn data into visuals, a task for technical managers to prepare leadership insights.

Line chart for lead time: Charting logs lead time (e.g., 10 minutes), showing trends. This highlights speed, a key point.

Pie chart for failure: Pie logs failure rates (e.g., 5%), showing stability. This ensures quality, visualized clearly.

Bar chart for frequency: Bars log frequency (e.g., 4 runs), showing pace. This proves agility, a leadership win.

Mapping Data

- Link logs to visuals.
- Query log outputs
- Assign to widgets
- Customize display options



34

Mapping ensures dashboard accuracy, a technical step for managers to connect data to insights.

Query log outputs: Querying logs lead time (e.g., 12 minutes), extracting data. This ensures real metrics, a key task.

Assign to widgets: Assigning logs failure rates (e.g., 4%), linking visuals. This accuracy aids reporting.

Customize display options: Customizing logs clarity (e.g., colors), enhancing impact. This makes data executive-ready.

Testing Dashboard



Confirm real-time function.

- Trigger new run
- Watch widget refresh
- Validate data accuracy

35

Testing ensures the dashboard works, a validation for technical managers to guarantee usability.

Trigger new run: Running logs frequency (e.g., 5 runs), testing updates. This ensures fresh data, a team check.

Watch widget refresh: Refresh logs lead time (e.g., 10 minutes), confirming live use. This aids visibility, a key feature.

Validate data accuracy: Validating logs MTTR (e.g., 6 minutes), ensuring trust. This proves reliability to leadership.

Dashboard Best Practices

Optimize for executives.

- Keep visuals simple
- Highlight critical metrics
- Ensure regular updates



36

Best practices enhance dashboard use, a guide for technical managers to present data effectively.

Keep visuals simple: Simple logs lead time (e.g., 3 charts), aiding clarity. This ensures quick insights, a leadership need.

Highlight critical metrics: Highlighting logs failure rates (e.g., bold), focusing attention. This prioritizes key data.

Ensure regular updates: Updates log frequency (e.g., daily), keeping it current. This proves ongoing value.

❖ Metrics Interpretation Basics

- Understanding DORA Metrics
- Exploring Six DevOps Metrics
- Strategic Insights Overview



This introduces metric interpretation, a foundation for technical managers to assess lab data strategically.

Understanding DORA Metrics: DORA metrics log frequency and MTTR (e.g., 4 runs, 5 minutes), benchmarking maturity. This shows performance, a leadership baseline.

Exploring Six DevOps Metrics: Six metrics log availability (e.g., 99%), expanding the view. This holistic data aids oversight.

Strategic Insights Overview: Insights log lead time (e.g., 10 minutes), connecting to goals. This prepares strategic communication.

DORA Metrics



Measure pipeline performance.

- Deployment frequency pace
- Lead time speed
- MTTR recovery
- Change failure stability

38

DORA metrics benchmark lab performance, a focus for technical managers to interpret data.

Deployment frequency pace: Frequency logs runs (e.g., 5 daily), a DORA metric for agility. This pace proves speed, shown as bars.

Lead time speed: Lead time logs time (e.g., 12 minutes), a DORA metric for delivery. This speed aids competitiveness, a trend.

MTTR recovery: MTTR logs recovery (e.g., 6 minutes), a DORA metric for resilience. This ensures uptime, a charted win.

Change failure stability: Failure logs rates (e.g., 4%), a DORA metric for quality. This balances safety, a pie chart.

Six DevOps Metrics

Holistic success indicators.

- Deployment frequency
- Lead time changes
- Mean time recovery
- Change failure rate
- Application availability
- Defect escape rate



39

Six metrics expand DORA, a comprehensive view for technical managers to assess lab success.

Deployment frequency: Logs runs (e.g., 4 daily), tracking pace. This shows output, a bar chart.

Lead time changes: Logs time (e.g., 10 minutes), measuring speed. This proves efficiency, a trend.

Mean time recovery: Logs recovery (e.g., 5 minutes), ensuring uptime. This aids cost savings.

Change failure rate: Logs fails (e.g., 4%), ensuring quality. This balances speed, a pie.

Application availability: Logs uptime (e.g., 99%), showing reliability. This builds trust.

Defect escape rate: Logs bugs (e.g., 1), reducing incidents. This proves quality.

Strategic Insights



Translate metrics to value.

- Speed and stability
- Risk reduction proactive
- Efficiency and costs

40

Insights connect metrics to value, a skill for technical managers to present strategically.

Speed and stability: Frequency and failure (e.g., 5 runs, 4%) log balance, proving agility. This shows sustainable delivery.

Risk reduction proactive: MTTR and defects (e.g., 6 minutes, 1 bug) log safety, ensuring trust. This reduces incidents.

Efficiency and costs: Availability (e.g., 99%) logs savings, cutting waste. This justifies investments.

❖ Lab Execution Steps

- Running the Pipeline
- Building the Dashboard
- Validating the Results



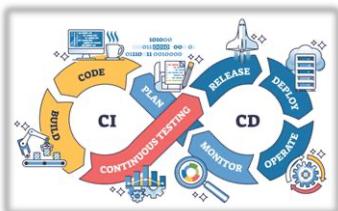
This provides execution steps, ensuring technical managers complete the lab with metrics.

Running the Pipeline: Triggering logs frequency (e.g., 4 runs), executing the flow. This generates data, a team task.

Building the Dashboard: Widgets log lead time (e.g., 10 minutes), visualizing insights. This aids reporting, a leadership step.

Validating the Results: Checking logs MTTR (e.g., 6 minutes), ensuring accuracy. This confirms success for executives.

Running Pipeline



Execute CI/CD flow.

- Trigger manually
- Monitor stage completion
- Check initial logs

42

Running the pipeline starts the lab, a step for technical managers to generate data.

Trigger manually: Triggering logs frequency (e.g., 5 runs), testing execution. This ensures flow, a team check.

Monitor stage completion: Watching logs MTTR (e.g., 6 minutes), verifying stages. This aids oversight, a key task.

Check initial logs: Reviewing logs lead time (e.g., 10 minutes), starting insights. This links runs to the dashboard.

Building Dashboard

Visualize pipeline data.

- Add metric widgets
- Map logs to charts
- Customize for clarity



43

Building the dashboard visualizes data, a task for technical managers to prepare insights.

Add metric widgets: Widgets log lead time (e.g., 12 minutes), showing trends. This aids clarity, a leadership tool.

Map logs to charts: Mapping logs failure (e.g., 4%), ensuring accuracy. This links data to visuals.

Customize for clarity: Customizing logs MTTR (e.g., 5 minutes), enhancing impact. This makes it executive-ready.

Validating Results



Ensure accuracy, function.

- Run pipeline again
- Check refresh
- Confirm alignment

44

Validating ensures lab success, a check for technical managers to confirm reliability.

Run pipeline again: Running logs frequency (e.g., 4 runs), testing updates. This ensures fresh data, a team step.

Check refresh: Refresh logs lead time (e.g., 10 minutes), confirming live use. This aids visibility, a key feature.

Confirm alignment: Matching logs MTTR (e.g., 6 minutes), ensuring trust. This proves accuracy to leadership.

❖ Metrics Analysis Deep Dive

- Analyzing Speed Metrics
- Evaluating Stability Metrics
- Assessing Efficiency Metrics



This deepens metric analysis, refining technical managers' skills to interpret lab data strategically.

Analyzing Speed Metrics: Frequency and lead time (e.g., 5 runs, 10 minutes) log agility, a DORA focus. This proves competitiveness, a trend.

Evaluating Stability Metrics: Failure and defects (e.g., 4%, 1 bug) log quality, ensuring safety. This builds trust, a chart.

Assessing Efficiency Metrics: MTTR and availability (e.g., 6 minutes, 99%) log savings, a cost win. This justifies investments.

Speed Metrics Analysis



Focus on pace.

- Lead time trends
- Frequency increases
- Faster delivery

46

Speed analysis quantifies agility, a deep dive for technical managers to show delivery improvements.

Lead time trends: Trends log lead time (e.g., 8 minutes), a DORA metric for speed. This proves efficiency, a leadership point.

Frequency increases: Frequency logs runs (e.g., 5 daily), showing pace. This highlights agility, a competitive edge.

Faster delivery: Combined logs delivery (e.g., 10 minutes, 4 runs), proving responsiveness. This wins markets, a strategic win.

Stability Metrics Analysis

Ensure reliable releases.

- Lower failure rates
- Reduced defects
- Consistent quality



47

Stability ensures quality, a deep dive for technical managers to prove safe releases.

Lower failure rates: Rates log fails (e.g., 4%), a DORA metric for stability. This reduces chaos, a trust point.

Reduced defects: Defects log bugs (e.g., 1), ensuring quality. This saves rework, a leadership win.

Consistent quality: Combined logs reliability (e.g., 3%, 1 bug), maintaining trust. This proves consistency, a key goal.

Efficiency Metrics Analysis



Optimize performance.

- High availability
- Quick MTTR
- Cost-effective workflows

48

Efficiency saves resources, a deep dive for technical managers to justify DevOps financially.

High availability: Availability logs uptime (e.g., 99%), showing reliability. This ensures trust, a customer win.

Quick MTTR: MTTR logs recovery (e.g., 5 minutes), cutting costs. This proves resilience, a financial point.

Cost-effective workflows: Combined logs savings (e.g., 99%, 6 minutes), reducing waste. This justifies investments.

❖ Strategic Alignment Principles

- Mapping Business Goals
- Executive Communication
- Planning Next Steps



This aligns metrics with strategy, a skill for technical managers to influence leadership.

Mapping Business Goals: Goals log lead time (e.g., 10 minutes), linking to priorities. This proves value, a strategic step.

Executive Communication: Communication logs failure (e.g., 4%), aiding clarity. This ensures impact, a leadership skill.

Planning Next Steps: Steps log MTTR (e.g., 6 minutes), guiding growth. This prepares future success.



Mapping shows value, a strategic task for technical managers to justify DevOps.

Time-to-market speed: Speed logs lead time (e.g., 8 minutes), a competitive edge. This wins markets, a leadership win.

Reduced incidents: Incidents log failure (e.g., 4%), ensuring trust. This reduces risks, a key point.

Cost optimization: Optimization logs availability (e.g., 99%), saving costs. This proves financial value.

Executive Communication



Present with impact.

- Clear visuals
- Highlight trends
- Tie to KPIs

51

Communication wins support, a skill for technical managers to present data.

Clear visuals: Visuals log lead time (e.g., 10 minutes), aiding clarity. This ensures understanding, a leadership need.

Highlight trends: Trends log failure (e.g., 4%), showing progress. This proves value, a key point.

Tie to KPIs: KPIs log availability (e.g., 99%), aligning goals. This connects to strategy, a win.

Planning Next Steps

Evolve DevOps.

- Refine pipeline
- Expand metrics
- Prepare trends



52

Planning ensures growth, a step for technical managers to lead DevOps forward.

Refine pipeline: Refining logs frequency (e.g., 5 runs), enhancing flows. This improves efficiency, a team goal.

Expand metrics: Expanding logs MTTR (e.g., 6 minutes), deepening insights. This aids oversight, a key task.

Prepare trends: Trends log lead time (e.g., 10 minutes), guiding future. This prepares innovation, a win.

❖ Wrap-Up and Future Steps

- Lab Recap Achievements
- Key Takeaways Week 8
- Preview Week 9 Content



This wraps up Week 8, summarizing achievements and previewing next steps for technical managers.

Lab Recap Achievements: Recap logs pipeline (e.g., 4 runs), celebrating success. This reinforces skills, a leadership point.

Key Takeaways Week 8: Takeaways log lead time (e.g., 10 minutes), highlighting value. This aids communication, a key skill.

Preview Week 9 Content: Preview logs cost (e.g., savings), transitioning forward. This prepares ROI focus.

Lab Recap

Celebrate your work.

- Built metrics pipeline
- Created dashboard
- Linked to strategy



© 2025 by Innovation in Software Corporation

54

This recaps lab success, a celebration for technical managers of your achievements.

Built metrics pipeline: Pipeline logs frequency (e.g., 5 runs), a technical win. This proves automation, a team skill.

Created dashboard: Dashboard logs lead time (e.g., 8 minutes), a visual tool. This aids reporting, a leadership step.

Linked to strategy: Linking logs MTTR (e.g., 6 minutes), proving value. This connects to goals, a strategic win.

Key Takeaways



Lessons for leadership.

- Metrics drive decisions
- Dashboards bridge gaps
- Automation enhances outcomes

55

Takeaways highlight Week 8's value, lessons for technical managers to lead with.

Metrics drive decisions: Metrics log failure (e.g., 4%), guiding choices. This justifies DevOps, a leadership tool.

Dashboards bridge gaps: Dashboards log availability (e.g., 99%), connecting tech. This aids communication, a key skill.

Automation enhances outcomes: Automation logs MTTR (e.g., 5 minutes), improving results. This proves efficiency, a win.

Week 9 Preview

- Cost and ROI ahead.
- DevOps economics
 - Cost-saving tools
 - ROI calculation lab



Sneak Peek

56

This previews Week 9, transitioning technical managers to cost-focused topics.

DevOps economics: Economics log savings (e.g., uptime), exploring costs. This sets up ROI, a leadership focus.

Cost-saving tools: Tools log MTTR (e.g., 6 minutes), optimizing spend. This proves value, a financial win.

ROI calculation lab: Lab logs lead time (e.g., 10 minutes), calculating returns. This justifies investments, a key step.

❖ Review Questions

5 Questions

Question 1

What tracks release pace?

- A. Lead time
- B. Deployment frequency
- C. MTTR
- D. Defect escape rate



© 2025 by Innovation in Software Corporation

58

This question tests your understanding of DORA metrics, specifically which one measures how often releases occur, a key focus from the lab's pipeline metrics. It's drawn from Slide 12 (Delivery Acceleration Benefits), where deployment frequency is highlighted as a metric for release pace. In the lab, you logged this metric (e.g., 5 runs daily) to show how frequently the pipeline deploys, reflecting agility. Understanding this helps technical managers quantify delivery speed for executives, linking technical output to business value like market responsiveness. The possible answers include other DORA and DevOps metrics to challenge your recall and differentiation skills.

Question 1

What tracks release pace?

- A. Lead time
- B. Deployment frequency**
- C. MTTR
- D. Defect escape rate



© 2025 by Innovation in Software Corporation

59

B. Deployment frequency is correct because it logs how often releases occur (e.g., 4 runs daily in the lab), a DORA metric tracking pace, visualized as a bar chart on the dashboard to show agility. **A. Lead time** is incorrect—it measures the time from commit to deploy (e.g., 10 minutes), not frequency, focusing on speed per release rather than how often releases happen. **C. MTTR** is incorrect—it tracks recovery time from failures (e.g., 6 minutes), a resilience metric unrelated to release pace. **D. Defect escape rate** is incorrect—it logs bugs missed pre-deploy (e.g., 1 bug), a quality metric, not a frequency measure.

Question 2

Which logs failure rates? (choose 2)

- A. Checkout
- B. Build and test
- C. Deploy with scan
- D. Dashboard setup



© 2025 by Innovation in Software Corporation

60

This question assesses your knowledge of pipeline stages from Slide 20 (Pipeline Stages), where failure rates are generated, requiring you to identify two stages from the lab. In the lab, you configured stages to log metrics like change failure rate (e.g., 4%), a DORA metric tied to stability. “Choose 2” tests your ability to pinpoint multiple sources of failure data, reflecting real-world pipeline complexity. This is crucial for technical managers to understand where quality issues arise, aiding oversight and reporting to leadership on stability.

Question 2

Which logs failure rates? (choose 2)

- A. Checkout
- B. Build and test**
- C. Deploy with scan**
- D. Dashboard setup



© 2025 by Innovation in Software Corporation

61

B. Build and test and **C. Deploy with scan** are correct. **Build and test** logs failure rates (e.g., a test fail at 5%) by running checks like _____, identifying code issues early, a key stability metric. **Deploy with scan** logs change failure rates (e.g., 4%) via a mock scan, catching deployment flaws, ensuring compliance and quality. **A. Checkout** is incorrect—it only pulls code (e.g., first run logged), not testing or failing, so no failure data is generated. **D. Dashboard setup** is incorrect—it visualizes data (e.g., charts), not logging failures itself, only displaying pipeline results.

Question 3

First dashboard step?

- A. Add widgets
- B. Create dashboard
- C. Map data
- D. Test refresh



© 2025 by Innovation in Software Corporation

62

This question evaluates your recall of the dashboard creation process from Slide 25 (Dashboard Setup), testing the initial step in visualizing lab metrics. In the lab, you started with a blank dashboard before adding widgets or mapping data, a foundational action logged as a setup task (e.g., 2 minutes). This sequence is critical for technical managers to understand, ensuring a logical workflow when building tools for executive insights. The options reflect later steps to differentiate the correct starting point.

Question 3

First dashboard step?

- A. Add widgets
- B. Create dashboard**
- C. Map data
- D. Test refresh



© 2025 by Innovation in Software Corporation

63

B. Create dashboard is correct—it's the first step (e.g., 2-minute setup in the lab), logged as opening Dashboards and clicking “New Dashboard” to start visualization, essential before other actions. **A. Add widgets** is incorrect—it follows creation (e.g., adding lead time chart), requiring a dashboard first, not the initial step. **C. Map data** is incorrect—it comes after widgets (e.g., linking logs), dependent on an existing dashboard, not the start. **D. Test refresh** is incorrect—it's last (e.g., checking live updates), validating the dashboard, not initiating it.

Question 4

Metric for recovery speed?

- A. Deployment frequency
- B. Lead time
- C. MTTR
- D. Change failure rate



© 2025 by Innovation in Software Corporation

64

This question probes your understanding of DORA metrics from Slide 30 (DORA Metrics), focusing on which measures recovery speed after failures. In the lab, you logged MTTR (e.g., 6 minutes) to track downtime recovery, a key resilience metric visualized on the dashboard. This knowledge helps technical managers assess operational health, a critical aspect for reporting reliability to leadership. The options include other DORA metrics to test your ability to distinguish recovery-specific measures.

Question 4

Metric for recovery speed?

- A. Deployment frequency
- B. Lead time
- C. **MTTR**
- D. Change failure rate



© 2025 by Innovation in Software Corporation

65

C. MTTR is correct—it logs recovery time (e.g., 6 minutes in the lab), a DORA metric measuring speed to fix failures, shown as a chart for resilience. **A. Deployment frequency** is incorrect—it tracks release pace (e.g., 5 runs daily), not recovery, focusing on frequency over time. **B. Lead time** is incorrect—it measures commit-to-deploy speed (e.g., 10 minutes), not failure recovery, a different speed aspect. **D. Change failure rate** is incorrect—it logs failed deploy percentage (e.g., 4%), a stability metric, not recovery speed.

Question 5

- Goal tied to stability?
- A. Time-to-market
 - B. Reduced incidents
 - C. Cost optimization
 - D. More features



© 2025 by Innovation in Software Corporation

66

This question tests your ability to align metrics with business goals from Slide 35 (Mapping Goals), focusing on stability's strategic outcome. In the lab, stability metrics like change failure rate (e.g., 4%) were logged and visualized, directly relating to incident reduction. This understanding is vital for technical managers to connect technical performance to leadership priorities, ensuring DevOps supports organizational objectives. The options reflect other goals to challenge your mapping skills.

Question 5

Goal tied to stability?

- A. Time-to-market
- B. Reduced incidents**
- C. Cost optimization
- D. More features



© 2025 by Innovation in Software Corporation

67

B. Reduced incidents is correct—it aligns with stability metrics like change failure rate (e.g., 4% in the lab), logged to minimize disruptions, a key goal shown as pie charts for trust. **A. Time-to-market** is incorrect—it ties to speed (e.g., lead time, 10 minutes), not stability, focusing on delivery pace over safety. **C. Cost optimization** is incorrect—it relates to efficiency (e.g., MTTR, 99% uptime), not directly stability, though indirectly linked. **D. More features** is incorrect—it's a development goal, not tied to stability metrics like failure rates.