

# 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 3

9

## Week 3 Agenda

- Recap of foundational concepts
- Crafting a DevOps strategy
- Assessing organizational readiness
- Exploring adoption frameworks
- Understanding architecture basics

© 2025 by Innovation in Software Corporation

10

Welcome to Week 3 of the “DevOps for Executive Leadership” bootcamp! We’re shifting from the introductory groundwork of Weeks 1 and 2 into the strategic core of DevOps—how to make it work for your organization’s goals. Today’s agenda is your roadmap: we’ll revisit core concepts, build a strategy that links DevOps to business outcomes, evaluate your readiness, explore frameworks to guide adoption, and examine the architectural foundations that keep it running smoothly. Think of this as your executive toolkit—practical, actionable insights with just enough zest to keep it engaging.

**Recap of foundational concepts:** We’ll refresh your memory on culture, tools, and metrics from Weeks 1 and 2—setting the stage without leaving anyone behind.

**Crafting a DevOps strategy:** This is about aligning DevOps with what drives your organization—revenue and efficiency, not just tech for tech’s sake.

**Assessing organizational readiness:** Before we accelerate, we’ll check if your processes, people, and systems are ready—or need a pit stop.

**Exploring adoption frameworks:** Frameworks like CALMS and DORA provide

structure, ensuring you're not lost in a sea of buzzwords.

**Understanding architecture basics:** We'll unpack the systems that power DevOps, giving you a clear view of the engine without needing a mechanic's license.

## **RECAP OF FOUNDATIONAL CONCEPTS**

*Week 2*

**Why DevOps Matters**

**The Three Ways**

**Essential Tools**

**Metrics Overview**

11

## Recap: Why DevOps Matters

A refresher on the core value of DevOps from Weeks 1-2.

- Unifies teams
- Accelerates delivery
- Reduces risks
- Enhances flexibility



12

Let's begin Week 3 with a quick look back at why DevOps isn't just another trend—it's a strategic asset. In Weeks 1 and 2, we established that DevOps breaks down barriers, speeds up delivery, lowers risks, and keeps your organization adaptable in a fast-paced world. This recap sets the foundation for today's focus on strategy and architecture, reinforcing why this matters to you as leaders—it's about driving value, not just keeping the IT lights on.

**Unifies teams:** DevOps brings development and operations together, fostering collaboration instead of the old “over-the-wall” chaos—think synchronized swimmers, not solo splashers.

**Accelerates delivery:** Features hit the market faster—more like express shipping than a slow boat—delivering business value without delay.

**Reduces risks:** Fewer outages and surprises mean reliable systems—no one wants a tech meltdown headline, right?

**Enhances flexibility:** Adapt to market shifts without breaking a sweat—agility that keeps you ahead, not playing catch-up.

## Recap: The Three Ways

Reviewing the guiding principles from Week 1.

- Flow streamlines work
- Feedback improves quality
- Learning drives progress
- Culture supports success



GENE KIM'S  
THREE WAYS OF DEVOPS

© 2025 by Innovation in Software Corporation

13

The Three Ways are the bedrock of DevOps, introduced in Week 1 as the principles that make it tick. They're about keeping work flowing, quality high, and improvement constant—all underpinned by a solid culture. This quick revisit ensures we're aligned as we move into strategy—it's the “why” that fuels the “how,” giving you a framework even busy executives can appreciate.

**Flow streamlines work:** Moves tasks from idea to production smoothly—like a well-oiled conveyor belt, not a traffic jam at rush hour.

**Feedback improves quality:** Tight loops catch issues early—more like a smoke detector than a post-fire report, saving headaches down the line.

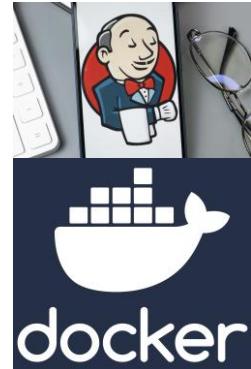
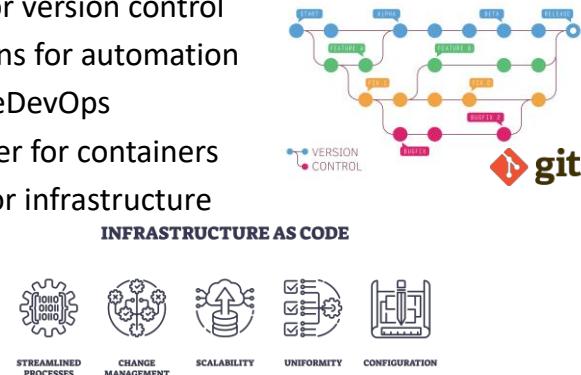
**Learning drives progress:** Every stumble's a lesson—think of it as turning a spilled latte into a new recipe, not a tantrum.

**Culture supports success:** Without team buy-in, these principles are just words—culture's the glue that keeps it all together.

## Recap: Essential Tools

Key tools introduced in Week 2 for DevOps success.

- Git for version control
- Jenkins for automation
- AzureDevOps
- Docker for containers
- IaC for infrastructure



© 2025 by Innovation in Software Corporation

14

Tools are the horsepower behind DevOps, and Week 2 handed us a shiny set to play with. This slide recaps Git, Jenkins, Docker, and Infrastructure as Code (IaC)—the must-haves that turn manual drudgery into automated elegance. They’re here to remind you of the toolkit we’ll build on today, proving that even executives can nod wisely when the IT team starts talking shop.

**Git for version control:** Tracks every code change like a meticulous record-keeper—no more “who broke this?” mysteries to solve.

**Jenkins for automation:** Your tireless build-and-deploy assistant—works around the clock so your team doesn’t have to pull all-nighters.

**Docker for containers:** Wraps apps in neat, portable packages—consistency that banishes the “works on my machine” excuse forever.

**IaC for infrastructure:** Manages servers as code—repeatable and reliable, like a recipe for IT success instead of a chaotic cook-off.

## Recap: Metrics Overview

Core metrics from Week 2 to gauge performance.

- Lead time efficiency
- Deployment frequency
- Failure rate trends
- Recovery speed

**FAILED**



**KPI**

© 2025 by Innovation in Software Corporation

15

Metrics are your DevOps compass—without them, you're navigating blind in a storm. In Week 2, we introduced lead time, deployment frequency, failure rate, and recovery speed as your key indicators, and they're back to anchor our strategic dive today. These aren't just numbers; they're your proof points for showing DevOps delivers, keeping it real for leaders who value results over rhetoric.

**Lead time efficiency:** Measures how fast code goes from commit to production—short and sharp means you're in the fast lane.

**Deployment frequency:** Tracks how often you ship—frequent deployments signal agility, not a once-in-a-blue-moon slog.

**Failure rate trends:** Monitors how often things break—low rates mean quality's on point, not a tech horror story.

**Recovery speed:** Gauges how quickly you're back up after a glitch—swift fixes keep the lights on without drama.

## GROUP EXERCISE



### Hit List group exercise.

- **Hit List Overview**
- **Hit List Steps 1-2**
- **Hit List Steps 3-4**

16

The Group Exercise section brings hands-on action with the Hit List—your chance to spot and tackle inefficiencies. We'll introduce the exercise, break it into steps to identify and prioritize long-running, manual, error-prone, frequent processes, and wrap it up with action plans. It's your executive workout to turn operational gripes into automation wins.

**Hit List Overview:** Sets up the goal—spotting tasks like manual server tweaks to zap.

**Hit List Steps 1-2:** Starts with brainstorming—list frequent, error-heavy chores.

**Hit List Steps 3-4:** Finishes with ranking and fixes—target the worst offenders.

## HIT LIST OVERVIEW

### Hit List group exercise

- Identify inefficiencies
- Form small groups
- List key issues
- Set automation goals



17

Time to roll up your sleeves—welcome to the Hit List exercise! In this hands-on activity you'll pinpoint inefficiencies ripe for automation, like long-running, manual, error-prone tasks that happen too often. It's your chance to play strategist, turning operational headaches into opportunities with a practical twist.

**Identify inefficiencies:** Hunt down the culprits—think manual server setups or daily report grinds that sap time and spirit.

**Form small groups:** Team up in squads—small teams spark sharper ideas, no lone wolves here.

**List key issues:** Jot down the pain—long-running config updates, error-prone data entry, frequent bottlenecks.

**Set automation goals:** Target the worst offenders—automate those repetitive, mistake-heavy chores first.

## HIT LIST STEPS 1-2



### Start building the Hit List.

- **Brainstorm pain points**
- **Identify frequent tasks**
- **Note error-prone areas**
- **Draft initial list**

18

Here's how to kick off the Hit List exercise—steps 1 and 2 get you rolling. This slide guides you through brainstorming pain points, spotting frequent tasks, noting error-prone areas, and drafting your list. It's your executive toolkit for turning frustration into a plan, focusing on real-world inefficiencies.

**Brainstorm pain points:** What's the worst? Long manual deployments or weekly report fixes—spill it all.

**Identify frequent tasks:** Daily config tweaks or constant server reboots—repetitive stuff that's crying for relief.

**Note error-prone areas:** Manual data entry with typos galore? Error-heavy processes need automation ASAP.

**Draft initial list:** Write it down—your hit list of culprits like slow patch rollouts starts taking shape here.

## HIT LIST STEPS 3-4

Complete and present the Hit List.

- Rank by impact
- Suggest automation fixes
- Prioritize top targets
- Share with the group



19

Finish strong with steps 3 and 4 of the Hit List exercise—this slide wraps it up with impact ranking, fix suggestions, prioritization, and sharing. It's your executive playbook for turning a list of gripes into a targeted automation plan, spotlighting processes like manual server provisioning or frequent error-prone updates.

**Rank by impact:** Sort the big pains—frequent manual builds or error-riddled reports hit hard, rank them high.

**Suggest automation fixes:** Propose solutions—scripted provisioning or automated testing to zap those woes.

**Prioritize top targets:** Pick the worst—like long-running config changes—automation's prime candidates.

**Share with the group:** Present your list—say, “manual patch delays kill us”—and steal good ideas too.

## ❖ Strategy

- Why It's Essential
- Business Alignment
- SMART Goals
- Roadmap Steps
- Revenue Growth Example
- Customer Experience Example
- Avoiding Common Pitfalls



© 2025 by Innovation in Software Corporation

20

The Strategy section is where we turn DevOps into a business powerhouse—it's your executive guide to making it work for your organization's goals. We'll cover why a strategy is critical, how it aligns with priorities, how to set clear objectives, and how to roll it out without tripping over your own feet. Plus, we'll throw in examples and pitfalls to keep it real and actionable for leaders like you.

**Why It's Essential:** Explains why skipping strategy is a recipe for chaos—links it to value, not just tech.

**Business Alignment:** Ties DevOps to what drives your organization—speed, customers, edge.

**SMART Goals:** Sets clear, measurable targets—no vague wishes, just solid wins.

**Roadmap Steps:** Lays out a phased plan—start small, scale smart, refine as you go.

**Revenue Growth Example:** Shows how DevOps boosts the bottom line—real dollars, real results.

**Customer Experience Example:** Highlights DevOps keeping customers happy—loyalty's the prize.

**Avoiding Common Pitfalls:** Flags traps to dodge—no tool binges or culture flops here.

## WHY IT'S ESSENTIAL

The critical role of a DevOps strategy.

Links to business goals

Prevents wasted effort

Sets clear priorities

Secures stakeholder support



21

A DevOps strategy isn't a nice-to-have—it's your ticket to avoiding a costly mess. This slide explains why it's essential: it ties DevOps to what drives your organization, keeps resources focused, clarifies next steps, and wins over the decision-makers. For leaders, it's the difference between a plan that delivers measurable value and a tech spree that ends in a budget review gone wrong.

**Links to business goals:** Ties DevOps to revenue and efficiency—not just a playground for IT enthusiasts, but a profit driver.

**Prevents wasted effort:** Keeps you from throwing time and money at distractions—strategy's your shield against chaos.

**Sets clear priorities:** Decides what's first—culture or tools?—so you're not guessing like it's a coin flip.

**Secures stakeholder support:** A sharp plan with clear benefits gets the nod from the top—no one backs a vague “trust me.”

## BUSINESS ALIGNMENT



Connecting DevOps to organizational priorities.

- Speeds market delivery
- Improves customer experience
- Drives cost efficiency
- Gains competitive edge

22

DevOps isn't just for the tech team—it's a strategic asset for your organization's top priorities. This slide highlights how it accelerates delivery, delights customers, trims costs, and keeps you ahead of the pack. It's your executive briefing on why DevOps deserves a seat at the table, delivering results that matter beyond the server room.

**Speeds market delivery:** Gets features out faster—think express lane launches, not a slow crawl to market.

**Improves customer experience:** Reliable, smooth systems keep users happy—nobody sticks around for a glitchy ride.

**Drives cost efficiency:** Automation slashes waste—more value without ballooning payroll, a CFO's dream.

**Gains competitive edge:** Agility lets you outpace rivals—they're still planning while you're live and winning.

## SMART Goals

Setting clear and actionable DevOps objectives.

- Specific targets
- Measurable outcomes
- Achievable plans
- Time-bound deadlines



© 2025 by Innovation in Software Corporation

23

Goals without structure are just daydreams—SMART goals give DevOps real bite. This slide introduces Specific, Measurable, Achievable, Relevant, Time-bound objectives to turn vague aspirations into concrete wins. It's how you shift from "let's improve" to "cut lead time by 50% by Q3" and actually make it happen, keeping your leadership focus sharp.

**Specific targets:** "Reduce downtime" becomes "cut outages to 1%"—precise, no fluff to muddy the waters.

**Measurable outcomes:** Numbers tell the tale—track lead time or failure rate to show real progress.

**Achievable plans:** Aim high but keep it realistic—no pie-in-the-sky flops that waste everyone's time.

**Time-bound deadlines:** "By June" beats "someday"—deadlines keep the momentum going strong.

## ROADMAP STEPS



**Building a phased DevOps implementation plan.**

**Launch a pilot project**

**Scale successful efforts**

**Gather ongoing feedback**

**Refine continuously**

24

A roadmap is your DevOps blueprint—start small, grow smart, and keep tweaking. This slide outlines a phased approach to avoid a reckless plunge that ends in chaos. It's about building steady momentum with practical steps, ensuring your organization's transformation is a marathon win, not a sprint to nowhere.

**Launch a pilot project:** Test one team or app—low risk, high learning, like dipping a toe before the cannonball.

**Scale successful efforts:** Roll out what works—think of it as franchising a hit concept across the board.

**Gather ongoing feedback:** Check what's clicking or clunking—adjustments beat stubborn marches every time.

**Refine continuously:** Keep polishing—DevOps isn't a “set it and forget it” gadget, it's a living process.

## REVENUE GROWTH EXAMPLE

**How DevOps enhances revenue opportunities.**

- **Rapid feature releases**
- **Reduced lead times**
- **Increased sales windows**
- **Retail case study**



25

Want proof DevOps boosts the bottom line? This slide shows how it drives revenue with faster releases and more sales opportunities, backed by a retail example. It's your executive snapshot of DevOps as a profit engine, not just a tech experiment, making it a no-brainer for leadership buy-in.

**Rapid feature releases:** New tools out fast—think holiday deals hitting before the rush, not after.

**Reduced lead times:** Code to market in a snap—speed turns into revenue quicker than you can blink.

**Increased sales windows:** Frequent updates mean more chances to sell—no waiting for the next big drop.

**Retail case study:** A retailer launches a checkout feature in days, cashing in on peak sales—DevOps pays dividends.

## CUSTOMER EXPERIENCE EXAMPLE



Improving satisfaction with DevOps practices.

- Frequent system updates
- Consistent uptime
- High performance levels
- Service industry case

26

DevOps keeps customers happy—and happy customers stick around. This slide highlights how updates, uptime, and performance enhance satisfaction, with a service industry example to seal the deal. It's your executive guide to using DevOps to boost loyalty without the tech-heavy fluff.

**Frequent system updates:** Fresh features keep things current—stale systems lose fans faster than you'd think.

**Consistent uptime:** Always-on service—no “sorry, we’re down” apologies to sour the mood.

**High performance levels:** Fast, smooth operation—users love it, and your reviews will too.

**Service industry case:** A service firm’s app stays flawless with constant tweaks—retention climbs steadily.

## Avoiding Common Pitfalls

Mistakes to steer clear of in DevOps adoption.

- Overemphasis on tools
- Neglecting culture
- Skipping readiness checks
- Ignoring performance metrics



© 2025 by Innovation in Software Corporation

27

DevOps has its share of traps—step wrong, and you’re in for a rough ride. This slide flags common pitfalls like tool obsession, culture neglect, blind leaps, and metric avoidance, offering a clear warning for leaders. It’s your executive checklist to keep your transformation on track and avoid a costly detour.

**Overemphasis on tools:** Chasing every shiny gadget wastes cash—strategy trumps a tool spree every time.

**Neglecting culture:** Tech without teamwork is a car without a driver—heeded for a crash, not a win.

**Skipping readiness checks:** Jumping in blind risks a flop—assess first or pay later.

**Ignoring performance metrics:** No data means no proof—you’ll be guessing instead of leading.

## ❖ Organizational Readiness

- Purpose of Assessment
- Evaluating Processes
- Cultural Assessment
- Technical Maturity
- Identifying Pain Points
- Assessment Tools



© 2025 by Innovation in Software Corporation

28

The Readiness section is your pre-game check-up—ensuring your organization's ready to hit the DevOps field without tripping. We'll cover why assessment matters, how to evaluate processes, culture, and tech, and how to spot pain points like manual, error-prone tasks that need fixing. It's your executive diagnostic to start smart and avoid a false start.

**Purpose of Assessment:** Why bother? It's your gap-finder and trust-builder.

**Evaluating Processes:** Maps workflows—where's the slow or shaky stuff?

**Cultural Assessment:** Checks team vibe—collab or chaos?

**Technical Maturity:** Assesses tools and skills—are you modern or medieval?

**Identifying Pain Points:** Spots the thorns—long-running manual messes to zap.

**Assessment Tools:** Frameworks and surveys—data over guesswork.

## Purpose of Assessment

Why readiness evaluation is a critical first step.

- Identifies weaknesses
- Shapes strategic focus
- Conserves resources
- Builds stakeholder trust



© 2025 by Innovation in Software Corporation

29

Before you unleash DevOps, assessing readiness is your safety net—it's like checking the weather before a big trip. This slide explains why it's essential: spotting weaknesses, guiding strategy, saving effort, and earning trust from the top. It's your leadership tool to ensure you're starting strong, not stumbling out of the gate.

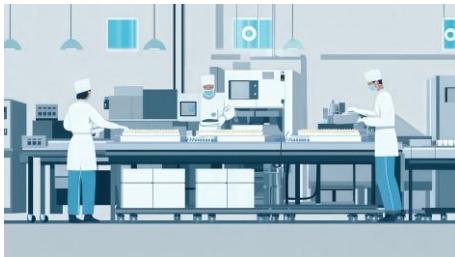
**Identifies weaknesses:** Pinpoints gaps—processes, tech, wherever it's shaky—so you're not caught off guard.

**Shapes strategic focus:** Weak spots show where to aim—fix what's broken, not what's already humming.

**Conserves resources:** Targets effort where it counts—no throwing time and money at shadows.

**Builds stakeholder trust:** Solid data wins over doubters—executives respect a plan with evidence.

## Evaluating Processes



Analyzing current operational workflows.

- Map process flows
- Measure release cycles
- Identify bottlenecks
- Assess automation use

30

Let's put your processes under the microscope—this is where we spot the inefficiencies begging for a fix. This slide guides you through mapping workflows, timing releases, finding clogs, and checking automation levels. It's your diagnostic toolkit to see if your operations are ready to roll or need a serious overhaul first.

**Map process flows:** Chart the journey from idea to live—where's it getting messy or bogged down?

**Measure release cycles:** Time how long releases take—slow as a turtle means trouble's brewing.

**Identify bottlenecks:** Pinpoint delays—approvals, testing?—name the culprits holding you back.

**Assess automation use:** Manual tasks dragging on? Automation's your friend—where's it missing?

## Cultural Assessment

Evaluating team mindset and preparedness.

- Collaboration levels
- Response to failure
- Change adaptability
- Tools for cultural insight



© 2025 by Innovation in Software Corporation

31

Culture's the heartbeat of DevOps—without it, you're just spinning wheels. This slide digs into assessing your team's readiness: how well they collaborate, handle setbacks, adapt to change, and how to measure it all. It's your executive lens to ensure your people are primed for DevOps, not poised to trip it up.

**Collaboration levels:** Are teams working together or playing tug-of-war? Unity's the DevOps gold standard.

**Response to failure:** Do they learn from flops or point fingers? Growth's the goal, not grumbling.

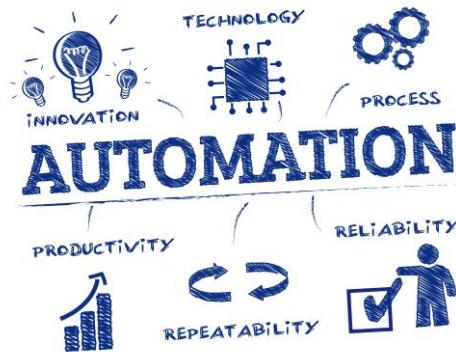
**Change adaptability:** "We've always done it this way" is a red flag—flexibility fuels progress.

**Tools for cultural insight:** Surveys or frameworks like CALMS—data beats guesswork every time.

## Technical Maturity

Assessing current technical capabilities.

- Tool adoption status
- Automation coverage
- Skill proficiency levels
- System integration



© 2025 by Innovation in Software Corporation

32

Your technical setup is the backbone of DevOps—weak spots here can derail everything. This slide evaluates your tools, automation, skills, and integration to see if you're a contender or still warming up. It's your leadership checkpoint to ensure your tech's ready to support a DevOps transformation without buckling.

**Tool adoption status:** Are modern tools like CI/CD in play, or are you stuck with relics? Progress depends on it.

**Automation coverage:** How much is automated versus manual? More automation equals less grunt work.

**Skill proficiency levels:** Is your team skilled or scrambling? Proficiency drives execution.

**System integration:** Do systems communicate smoothly or clash? Integration's the glue for efficiency.

## Identifying Pain Points

Spotting inefficiencies in current operations.

- Manual repetitive tasks
- Frequent system outages
- Slow delivery cycles
- High error rates



4 Types of Customer Pain Points



© 2025 by Innovation in Software Corporation

33

Pain points are the thorns in your operational side—left unchecked, they'll sting. This slide helps you identify manual tasks, outages, slow cycles, and errors that need attention. It's your executive radar for pinpointing what's holding back progress, setting the stage for smarter fixes.

**Manual repetitive tasks:** Hand-done chores eating time? Automation's the cure—just say goodbye to drudgery.

**Frequent system outages:** Too many crashes? Reliability's crying for help, and customers notice.

**Slow delivery cycles:** Weeks to ship? That's a snail's pace in a world that demands speed.

**High error rates:** Errors piling up? Quality's slipping—time to tighten the screws.

## Assessment Tools

Frameworks and tools to measure readiness.

- CALMS framework
- DORA metrics analysis
- Feedback surveys
- Process audits



© 2025 by Innovation in Software Corporation

34

No more relying on gut feelings—assessment tools bring science to readiness. This slide covers CALMS, DORA, surveys, and audits to give you a clear picture of where you stand. It's your executive toolkit for turning vague impressions into actionable data, ensuring your DevOps journey starts on solid ground.

**CALMS framework:** Culture, automation, lean—your comprehensive checklist for readiness across the board.

**DORA metrics analysis:** Lead time, MTTR—your performance stats, measured and meaningful.

**Feedback surveys:** Ask your teams—frontline insights beat top-down assumptions every time.

**Process audits:** Dive into workflows—uncover the hidden snags with a systematic sweep.

## ❖ FRAMEWORKS



- **Purpose and Value**
- **CALMS Overview**
- **CALMS Sharing**
- **DORA Metrics Intro**
- **DORA Benchmarking**

35

The Frameworks section is your DevOps navigation system—guiding you through the adoption maze with proven tools. We'll cover why frameworks matter, dive into CALMS and its Sharing pillar, and explore DORA metrics for benchmarking. It's your executive playbook to keep efforts structured and measurable, not a wild goose chase.

**Purpose and Value:** Why frameworks? They're your structure and focus lifeline.

**CALMS Overview:** Introduces the five pillars—your holistic guide.

**CALMS Sharing:** Dives into Sharing—keeping teams sharp and synced.

**DORA Metrics Intro:** Unpacks DORA's four metrics—your performance pulse.

**DORA Benchmarking:** Uses DORA to measure up—goals and trends in view.

Image Source: <https://www.xalt.de/en/start/calms-framework/>

Image Source: <https://www.sourcedgroup.com/blog/dora-metrics-measuring-what-really-matters-about-your-software-delivery/>

## Purpose and Value

Why adoption frameworks are essential for success.

- Provide structured guidance
- Prioritize key focus areas
- Enable progress tracking
- Foster team alignment



© 2025 by Innovation in Software Corporation

36

Frameworks are your DevOps lifeline—without them, you’re guessing in a fog. This slide explains how they offer structure, focus, tracking, and alignment, keeping your efforts on track instead of veering into chaos. They’re the guardrails that turn ambition into achievement with less drama.

**Provide structured guidance:** A clear path beats wandering—frameworks map it out, no detours needed.

**Prioritize key focus areas:** Culture or tech first? They decide, so you’re not spinning wheels.

**Enable progress tracking:** Are you improving? Metrics give you the proof, not just hope.

**Foster team alignment:** Everyone’s rowing together—no rogue paddlers sinking the boat.

## CALMS Overview

Introduction to the CALMS (Culture, Automation, Lean, Measurement) model components.

- Culture emphasis
- Automation priority
- Lean efficiency focus
- Measurement standards



© 2025 by Innovation in Software Corporation

37

CALMS is your DevOps compass—five pillars to steer by. This slide introduces Culture, Automation, Lean, and Measurement (with Sharing up next), giving you a framework to assess and advance your efforts. It's your executive shorthand for mastering DevOps without drowning in complexity.

**Culture emphasis:** People come first—trust and teamwork set the tone for everything else.

**Automation priority:** Machines take the grunt work—efficiency's the name of the game here.

**Lean efficiency focus:** Trim the fat—sleek operations outpace bloated ones every time.

**Measurement standards:** Data drives decisions—without it, you're just guessing in the wind.

## CALMS Sharing

The Sharing pillar of the CALMS framework.

- Knowledge dissemination
- Lessons learned exchange
- Cross-team alignment
- Collaboration tools



© 2025 by Innovation in Software Corporation

38

Sharing's the secret sauce in CALMS—keeping your team sharp and connected. This slide explores spreading knowledge, exchanging lessons, aligning efforts, and leveraging tools to make it happen. It's your executive guide to ensuring no one's hoarding wisdom when collaboration's the key to success.

**Knowledge dissemination:** Info flows freely—documents and chats beat silos hands down.

**Lessons learned exchange:** Every stumble's a lesson—share the gold, not the grumbles.

**Cross-team alignment:** One vision unites everyone—no splintered efforts derailing the plan.

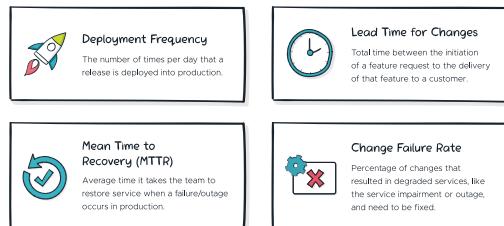
**Collaboration tools:** Platforms like wikis or chat apps—your teamwork turbochargers.

## DORA Metrics Intro

Introduction to the DORA (DevOps Research and Assessment) metrics framework.

- Lead time efficiency
- Deployment frequency
- Mean time to recovery
- Change failure rate

Key DORA Metrics



© 2025 by Innovation in Software Corporation

39

DORA metrics are your DevOps scorecard—how the pros measure up. This slide introduces lead time, deployment frequency, MTTR, and failure rate, giving you a clear lens to track performance. It's your executive tool to see if you're hitting the mark or still warming up, without needing a tech decoder ring.

**Lead time efficiency:** Code to live—fast means you're in the race, not on the sidelines.

**Deployment frequency:** Ship often—frequent flyers signal agility, not inertia.

**Mean time to recovery:** Bounce back quick—speedy fixes keep the show running.

**Change failure rate:** Keep flops low—quality's your badge of honor.

## DORA Benchmarking

Using DORA to benchmark performance

- Compare to industry peers
- Set improvement targets
- Monitor trends over time
- Focus strategic efforts



© 2025 by Innovation in Software Corporation

40

DORA isn't just numbers—it's your competitive edge wrapped in data. This slide shows how to use DORA metrics to stack up against peers, set goals, track progress, and sharpen your strategy. It's your executive playbook for turning metrics into actionable wins, not just dashboard decorations.

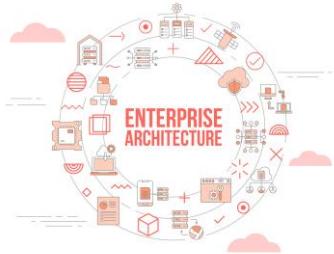
**Compare to industry peers:** Are you elite or trailing? DORA shows where you stand in the pack.

**Set improvement targets:** Monthly to daily deploys? Goals give you something to chase.

**Monitor trends over time:** Progress isn't a hunch—watch the numbers tell the story.

**Focus strategic efforts:** Metrics point the way—tackle what matters most, not just what's loudest.

## ❖ ARCHITECTURE



- **What It Means**
- **Core Elements**
- **Pipeline Basics**
- **Integration Value**

41

The Architecture section is your peek into the DevOps engine room—how systems support your goals. We'll define what it means, cover its core pieces, explore the delivery pipeline, and explain why integration ties it all together. It's your executive overview to grasp the backbone without needing a tech degree.

**What It Means:** Defines architecture—your system's role in DevOps success.

**Core Elements:** Unpacks the key pieces—microservices, pipelines, and more.

**Pipeline Basics:** Explains the delivery pipeline—your code-to-live highway.

**Integration Value:** Highlights why integration's the glue—sync or sink.

## What It Means

Defining the role of DevOps architecture.

- Supports DevOps goals
- Enables rapid delivery
- Ensures scalability
- Maintains resilience

99.999%



© 2025 by Innovation in Software Corporation

42

Architecture's the backbone of DevOps—get it right, and your goals stay upright. This slide defines it as the system that supports rapid delivery, scales with demand, and stays resilient under pressure. It's your executive overview of the engine room, keeping it clear and relevant without needing a tech manual.

**Supports DevOps goals:** Built to deliver—not a vanity project, but a purpose-driven design.

**Enables rapid delivery:** Fast deploys—architecture keeps the pace without stumbling.

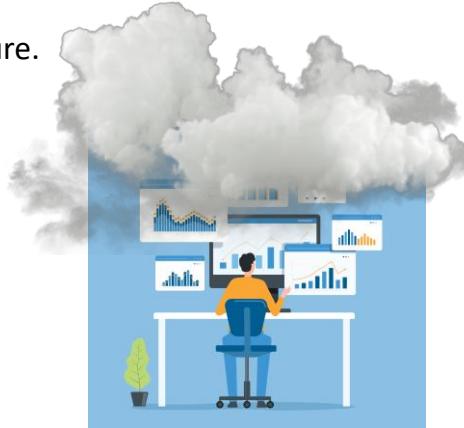
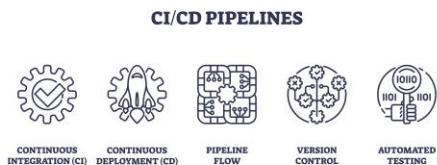
**Ensures scalability:** Grows with demand—no choking when the heat's on.

**Maintains resilience:** Stays tough—built to bend, not break, under stress.

## Core Elements

Key components of DevOps architecture.

- Microservices design
- CI/CD pipelines
- Cloud infrastructure
- Monitoring systems



© 2025 by Innovation in Software Corporation

43

DevOps architecture has pieces that fit like a well-crafted puzzle. This slide covers microservices, CI/CD pipelines, cloud infrastructure, and monitoring—the essentials that make it tick. It's your executive snapshot of what keeps the system humming, stripped of unnecessary complexity.

**Microservices design:** Small, agile apps—not clunky monoliths that drag you down.

**CI/CD pipelines:** Automated flow from code to live—smooth as a freshly paved road.

**Cloud infrastructure:** Flexible and scalable—cloud's your growth partner.

**Monitoring systems:** Keeps watch—catches issues before they turn into headlines.

## Pipeline Basics

The role of the delivery pipeline in DevOps.

- Automates builds
- Runs automated tests
- Deploys seamlessly
- Provides feedback loops



© 2025 by Innovation in Software Corporation

44

The delivery pipeline's your DevOps conveyor belt—code to production without a hitch. This slide breaks down how it automates builds, tests, deploys, and loops feedback to keep things moving. It's your executive guide to the heartbeat of rapid, reliable delivery.

**Automates builds:** Code compiles itself—no hand-holding required, just efficiency.

**Runs automated tests:** Bugs get caught early—your safety net's always up.

**Deploys seamlessly:** Goes live without drama—smooth transitions win the day.

**Provides feedback loops:** Quick insights—learn fast and keep improving.

## Integration Value

Why integration is critical in architecture.

- Synchronizes tools
- Ensures data flow
- Unites team efforts
- Reduces error rates



© 2025 by Innovation in Software Corporation

45

Integration's the glue that holds DevOps architecture together—without it, you've got a pile of parts, not a system. This slide explains how it syncs tools, flows data, unites teams, and cuts errors. It's your executive insight into why everything needs to play nice together, not a collection of solo acts.

**Synchronizes tools:** Keeps your tech stack talking—no tool tantrums allowed.

**Ensures data flow:** Info moves freely—bottlenecks are the enemy here.

**Unites team efforts:** Dev and ops in sync—harmony beats discord every time.

**Reduces error rates:** Tight connections mean fewer slip-ups—precision matters.

## ❖ Architectural Principles



- Scalability Design
- Reliability Practices
- Resilience Techniques

46

The Principles section is your guide to building DevOps systems that last—scalability, reliability, and resilience. We'll cover designing for growth, ensuring consistent performance, and crafting systems that bounce back fast. It's your executive primer on the rules that keep your architecture from crumbling under pressure.

**Scalability Design:** How to grow without breaking—your growth playbook.

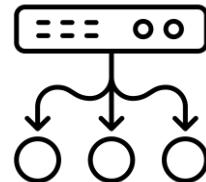
**Reliability Practices:** Keeps systems steady—no outages to haunt you.

**Resilience Techniques:** Builds toughness—recovers quick, stays strong.

## Scalability Design

Designing systems to handle growth effectively.

- Horizontal scaling
- Load balancing
- Elastic resource use
- Cloud utilization



© 2025 by Innovation in Software Corporation

47

Scalability's your strategy for growth—systems that stretch without snapping. This slide covers designing with horizontal scaling, load balancing, elasticity, and cloud power to handle demand spikes. It's your executive primer on keeping pace with success, not buckling under it.

**Horizontal scaling:** Adds servers—not bigger ones—growth that's smart and manageable.

**Load balancing:** Spreads work evenly—no server gets overwhelmed, no crashes.

**Elastic resource use:** Flexes up or down—resources match need, not waste.

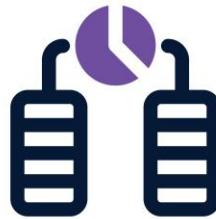
**Cloud utilization:** Taps cloud flexibility—scales effortlessly, no sweat required.

## Reliability Practices

Ensuring consistent system performance:

- Redundant systems
- Failover mechanisms
- Error management
- Uptime targets

99.999%



© 2025 by Innovation in Software Corporation

48

Reliability's your uptime promise—systems that don't flake out when it counts. This slide dives into redundancy, failover, error handling, and uptime goals to keep your operations rock-solid. It's your executive assurance that DevOps delivers stability, not sleepless nights.

**Redundant systems:** Backups in place—no single failure takes you down.

**Failover mechanisms:** Switches to spares fast—keeps the lights on without a flicker.

**Error management:** Handles flops smoothly—no chaos when things go awry.

**Uptime targets:** Aiming for 99.99%—reliability's your badge of trust.

## Resilience Techniques

Building systems that recover quickly.

- Failure planning
- Self-healing design
- Stress testing
- Rapid recovery focus



© 2025 by Innovation in Software Corporation

49

Resilience means your systems can take a hit and keep ticking—toughness baked in. This slide explores planning for failures, self-healing, stress testing, and quick recovery to ensure your operations stay robust. It's your executive guide to bouncing back without breaking a sweat.

**Failure planning:** Preps for the inevitable—hope's nice, but plans are better.

**Self-healing design:** Systems fix themselves—less babysitting, more brilliance.

**Stress testing:** Pushes limits—finds cracks before they split wide open.

**Rapid recovery focus:** Back up fast—downtime's the enemy, speed's the ally.

## Demo

- Azure DevOps Pipeline Metrics Intro
- Azure DevOps Pipeline Metrics Features
- Azure DevOps Pipeline Metrics Insights



© 2025 by Innovation in Software Corporation

50

The Demo section offers a live look at the Azure DevOps Pipeline Metrics Dashboard—your window into pipeline performance and health. We'll introduce the purpose, showcase key features, and highlight insights from metrics like build success and deployment frequency. It's your executive view of how Azure DevOps turns data into decisions, keeping it practical and relevant.

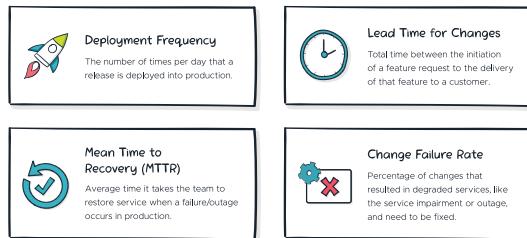
- Azure DevOps Pipeline Metrics Intro\*\*: Sets up the demo—why pipeline metrics matter for strategy.
- Azure DevOps Pipeline Metrics Features\*\*: Shows the dashboard—key metrics in action.
- Azure DevOps Pipeline Metrics Insights\*\*: Ties trends to outcomes—speed, quality, and value.

# Azure DevOps Pipeline Metrics Intro

Introduction to the Azure DevOps Pipeline Metrics demo.

- Purpose of pipeline metrics
- Key metrics overview
- Dashboard setup preview
- Strategic insights goal

Key DORA Metrics



© 2025 by Innovation in Software Corporation

51

Welcome to the Azure DevOps Pipeline Metrics demo—a live peek at how pipelines drive DevOps success! This slide sets the stage, explaining why pipeline metrics matter, what we'll track, how the dashboard works, and the insights you'll gain. It's your executive intro to seeing operational health in action, tailored for strategic impact.

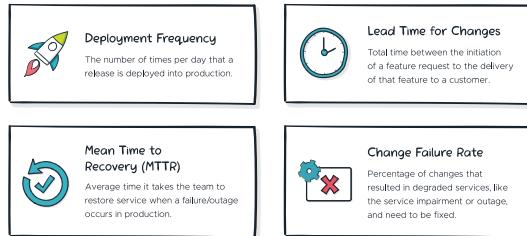
- Purpose of pipeline metrics\*\*: Measures pipeline efficiency and reliability—your DevOps pulse beyond buzzwords.
- Key metrics overview\*\*: Build success, deployment pace—your vital signs for performance.
- Dashboard setup preview\*\*: Built in Azure DevOps—simple widgets, no tech overload here.
- Strategic insights goal\*\*: Spot trends—faster delivery, fewer flops, all tied to business wins.

## Azure DevOps Pipeline Metrics Features

Key features showcased in the demo.

- Build success rate trends
- Deployment frequency graph
- Pipeline duration view
- Failed deployment alerts

Key DORA Metrics



© 2025 by Innovation in Software Corporation

52

The DORA Dashboard demo continues—this slide highlights what you'll see in action. Lead time, deployment frequency, MTTR, and failure rate trends come to life, giving you a clear picture of your DevOps health. It's your executive snapshot of performance, minus the tech overload.

**Lead time trends:** Watch speed improve—days to minutes, tracked live on the screen.

**Deployment frequency view:** See the pace—frequent or sluggish, it's all there.

**MTTR analysis:** Recovery speed shines—fast fixes mean less downtime pain.

**Failure rate insights:** Flops drop—quality trends you can brag about.



# Quiz

Let's review what we discussed today

53

The Quiz section wraps up with five review questions to test your Week 3 takeaways. We'll cover strategy benefits, CALMS pillars, DORA metrics, scalability features, and the Hit List exercise—each with a question and answer slide. It's your executive chance to flex your DevOps smarts with a quick, no-pressure check-in.

**Question 1:** Tests strategy's value—why it's a must-have.

**Question 1 Answer:** Confirms the answer—aligns with business goals.

**Question 2:** Checks CALMS—Lean's role in the mix.

**Question 2 Answer:** Explains efficiency emphasis—Lean's core.

**Question 3:** Digs into DORA—recovery speed metric.

**Question 3 Answer:** Highlights MTTR—your bounce-back stat.

**Question 4:** Scales up—key scalability feature.

**Question 4 Answer:** Locks in horizontal scaling—growth smart.

**Question 5:** Hits the Hit List—its purpose in action.

**Question 5 Answer:** Confirms inefficiencies—automation targets.

## Question 1

What's a key benefit of a DevOps strategy?

- A) Increases manual effort
- B) Aligns with business goals
- C) Slows delivery cycles
- D) Reduces team collaboration

© 2025 by Innovation in Software Corporation

54

Let's kick off the review with Question 1—a quick test on why a DevOps strategy matters, tied to Slide 7 (Why It's Essential). This multiple-choice question checks if you've grasped strategy's core value beyond tech tweaks. It's a straightforward way to spark discussion and lock in a key takeaway.

**What's a key benefit of a DevOps strategy?**: From Slide 7—pick the standout reason it's worth your time.

- A) Increases manual effort**
- B) Aligns with business goals**
- C) Slows delivery cycles**
- D) Reduces team collaboration**

## Question 2

What does ‘Lean’ mean in CALMS?

- A) Speed focus
- B) Cost reduction only
- C) Efficiency emphasis
- D) Tool selection

© 2025 by Innovation in Software Corporation

56

Question 2 tests your understanding of the CALMS framework from Slide 21 (CALMS Overview)—what’s ‘Lean’ all about? This multiple-choice question checks if you’ve caught one of the model’s core pillars, keeping it simple and tied to the content.

**What does ‘Lean’ mean in CALMS?:** From Slide 21—pick the right definition for this CALMS pillar.

- A) Speed focus**
- B) Cost reduction only**
- C) Efficiency emphasis**
- D) Tool selection**

## Question 3

Which DORA metric tracks recovery speed?

- A) Lead time
- B) Deployment frequency
- C) Mean time to recovery
- D) Change failure rate

© 2025 by Innovation in Software Corporation

58

Question 3 dives into DORA metrics from Slide 23 (DORA Metrics Intro)—which one's about bouncing back fast? This multiple-choice question tests your recall of a key DORA metric, keeping it straightforward and linked to the content.

**Which DORA metric tracks recovery speed?**: From Slide 23—pick the one that measures bounce-back time.

**A) Lead time**

**B) Deployment frequency**

**C) Mean time to recovery**

**D) Change failure rate**

## Question 4

What's a key scalability feature?

- A) Horizontal scaling
- B) Manual operations
- C) Fixed resources
- D) Single server reliance

© 2025 by Innovation in Software Corporation

60

Question 4 tests your grasp of scalability from Slide 29 (Scalability Design)—what helps systems grow? This multiple-choice question checks if you've caught a core scalability concept, keeping it tied to the content.

**What's a key scalability feature?:** From Slide 29—pick what supports system growth.

- A) Horizontal scaling**
- B) Manual operations**
- C) Fixed resources**
- D) Single server reliance**

## Question 5

What's the Hit List exercise for?

- A) Tool inventory
- B) Spotting inefficiencies
- C) Budget planning
- D) Team assignments

© 2025 by Innovation in Software Corporation

62

Question 5 wraps up with the Hit List exercise from Slide 36 (Hit List Overview)—what's its goal? This multiple-choice question tests if you've nailed the purpose of spotting automation targets like long-running, manual processes.

**What's the Hit List exercise for?**: From Slide 36—pick the core goal of this activity.

- A) Tool inventory**
- B) Spotting inefficiencies**
- C) Budget planning**
- D) Team assignments**