Liquibase

WHITE PAPER

Applying the CALMS Framework to Database DevOps

Applying the CALMS Framework to Database DevOps

Whether you've already applied the CALMS framework to your development process or you're just starting to investigate what DevOps principles can do for your team, this white paper will help you understand some key concepts that you can apply for a more functional full stack team that includes database deployments.

Many companies already apply agile and CALMS to their front-end development process. Advances in database development software now mean the same framework can be used to evaluate if your business is ready for database DevOps so that you can apply this approach to the full stack. That way, you can reap the full benefits of DevOps.

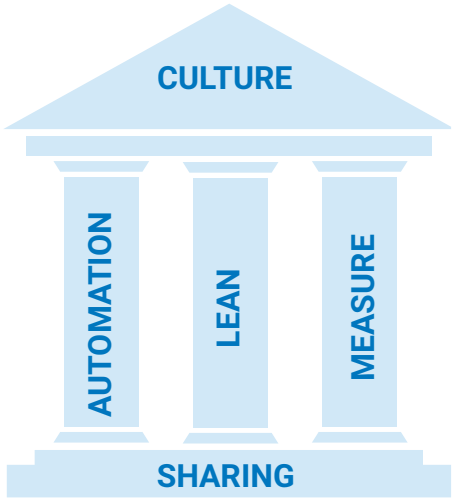
What is the CALMS framework?

For those of you that aren't familiar with CALMS, here's a quick primer.

CALMS is an acronym for: Culture, Automation, Lean, Measure, Sharing. In IT, you fight acronyms with acronyms. Prior to CALMS, ITSM (Information Technology Service Management) principles reigned supreme in the IT world. With ITSM, repeatability is a main component, but the structure and processes are not primary objectives.

In practice, CALMS is a framework for DevOps. Where ITSM is focused on documentation and change control, DevOps focuses on the people and culture aspects of software delivery.

CULTURE
AUTOMATION
LEAN
MEASURE
SHARING



Culture

With a lot of moving parts, something inevitably goes wrong when building and deploying software. As teams accelerate software delivery, it's important to plan for failure. The true measure of agility is how long it takes to resolve issues once they occur. To achieve agility, teams require a high degree of transparency and strong coordination, enabling them to quickly and effectively resolve and recover from issues.

In CALMS, culture is the key component. It's a nod to the fact that at the end of the day, the most important (and arguably the toughest) part to get right in DevOps is putting the culture in place to remove barriers and facilitate collaboration to hit team objectives. Culture takes time to cultivate by rewarding helpful behaviors. If you can get this part right, everything else tends to fall into place easily.

Culture for the Whole Stack

Most companies implement the culture change for DevOps within the development team first, which is great. The next step is to make sure the culture is in place throughout the entire process, including the data layer. Most of the time, DBAs are separated into their own silo. The database side may as well be in a completely different universe.

Since the Development team is running smoothly, when it comes time to incorporate Ops teams into the DevOps culture, groups such as DBAs and security are forced into the existing culture. This doesn't go well. Now, this doesn't mean you need to start from scratch to fit these teams into the model—just understand that everyone will need to make adjustments as other groups such as the database team are incorporated. This will take time.

The goal is to get the whole team—developers, security and DBAs—to share responsibility for smooth release cycles, including overall performance and where to build in security.

Automation

The whole reason most organization adopt DevOps is the promise of faster releases. In order to go as fast as possible, you need the help of your robot friends. Enter Continuous Integration practices and the Continuous Delivery pipeline (aka CI/CD).

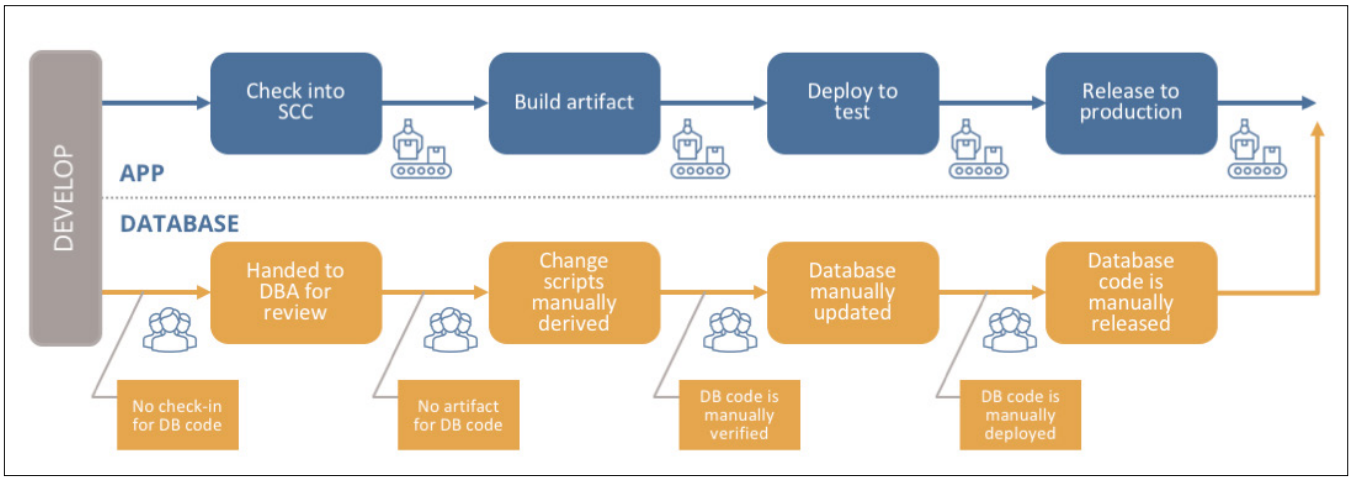
Small + Fast = Better

Some benefits of Continuous Delivery:

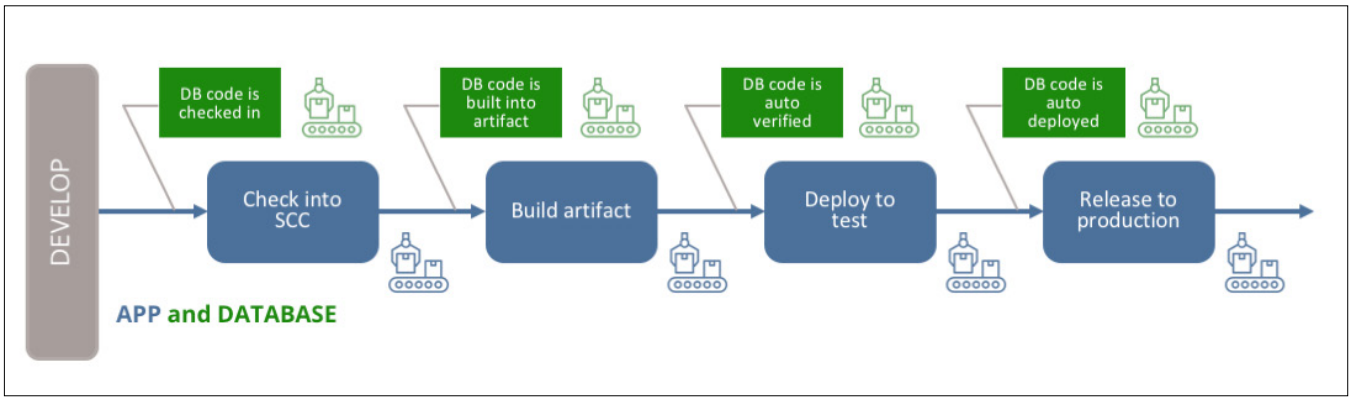
- Speed time-to-market
- Increases quality by eliminating manual errors
- Limits work-in-progress
- Decreases lead time for changes
- Improves mean time to recover
- Everything is traceable (hello, easier audits!)
- Increases release confidence

Automation for Database Changes

To make your Continuous Delivery pipeline work at its full capacity, you have to consider everything as code. This includes the data layer.



Continuous integration pipeline when you don't include the database.



Continuous integration pipeline with the database included.

When you include the database in continuous integration, the process is much simpler. Application code and database code follow the same release process, time market for new features is accelerated and database code quality is greatly improved.

Lean

You may have heard of Lean Manufacturing or Lean Startups. Lean principles originated with the Japanese manufacturing industry. The Lean in CALMS helps translate Lean to DevOps. At the most basic level, it's the systematic method for eliminating waste without sacrificing productivity.

In DevOps, you don't want to move big chunks of changes to production. DevOps helps create a constant flow of small, easy-to-control changes.

7 Principles of Lean Software Development

1. Eliminate waste
2. Amplify learning
3. Decide as late as possible
4. Deliver as fast as possible
5. Empower the team
6. Build integrity in
7. See the whole

7 Wastes of Lean Software

1. Partially done work
2. Extra features
3. Relearning
4. Handoffs
5. Delays
6. Task switching
7. Defects

Lean for the Database Workflow

See the whole! Database changes and application changes often go hand-in-hand; they are an essential part of the overall continuous delivery process. Since the whole system matters, it's crucial to consider applying Lean principles to the database team's workflow. Lean further emphasizes the importance of automating database change automation so that there are frequent incremental steps along the way to ensure your team delivers as fast as possible in small chunks.

Task switching

One of the Seven Wastes of Lean Software is task switching. One common place this occurs is with the relationship between the developer and the DBA.

Jen Roadrunner is writing code for Acme's new Wow feature. She just checked in her code and it requires a database change. Jen writes a script for the change.

Jen submits a ticket to her team's DBA, Jim Coyote, to make the database change.

< SWITCH > Jen moves on to something else.

< LOST TIME > Jim has backlog of tickets, so he doesn't see Jen's request for a week.

Jim finally opens the ticket from Jen and he sees that it's missing a key.

Jim replies back to Jen in the ticket.

< SWITCH > Jen opens the ticket and can't remember which piece of code this was for.

< LOST TIME > It takes Jen a couple of hours to find and correct the issue and send it back to Jim.

< SWITCH > Jim moved on to another ticket and doesn't get back to Jen's ticket for another day.

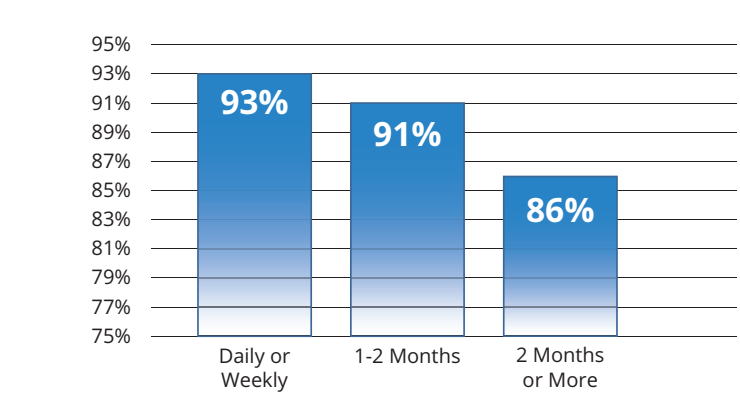
Jim sees that the key is updated and implements the change.

This is not efficient.

Database rework

In a recent survey, participants were asked how often they must re-work a database change before it's released to production. Database rework numbers are staggeringly high. For teams with daily or weekly release cycles, 93% say they have to rework database changes multiple times.

View the full **Dimensional Research survey report**.



Need to Rework a Database Multiple Times Before it's Released to Production

Measure

It's not possible to improve if you're not measuring. A successful DevOps implementation takes three different areas into account:

- Performance metrics
- Process metrics
- People metrics

The most successful DevOps organizations don't just track technical metrics, they also look at measurements of team health and performance.

Learn more

Read 9 Metrics DevOps Teams Should Be Tracking

Measuring the Data Layer with Database Deployment Information

To truly realize full stack DevOps success, most organizations need a clearer view into their database deployment process. How many database changes have been made? How many database changes are pending? How many build failures are happening because of database rules violations? Learn how Liquibase makes database deployment measurement possible.

Sharing

Sharing improves the communication flow and helps people work together more effectively. It's important to share ideas, experiences, and thoughts within the team and with the whole organization. Sharing key performance metrics with all stakeholders in your organization lets everyone monitor your DevOps efforts and prove success at every stage.

Sharing Between the Development and Operations

DevOps exists to bridge the gap between developers and operations teams. A big part of that process is to ensure that both sides are sharing responsibility for the overall quality of the application. That means no more finger pointing about database changes moving too slow or developers not following DBA rules. It's no longer us vs. them. In order to get to that happy place faster, it helps to have the right tools and processes in place to facilitate this relationship.

When you take advantage of including the database in your DevOps processes, barriers will come down and DBAs will see the value of working on complex changes with development teams early in the development process, instead of finding problems when the changes reach production.

Liquibase Makes DevOps Teams Successful

It's true that DevOps success relies on software tools and processes, but ultimately, it's about enabling people and shaping culture. Liquibase's ability to bring DevOps to the database empowers your DBAs and developers so your whole team can enjoy DevOps success.

Liquibase puts people and culture first

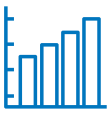
We partner with your team to ensure your success. We don't just give you a tool; we're with you every step of the way to integrate your database team and database changes into a smooth, repeatable application release process.

Liquibase provides database automation tools

When you include the database in continuous delivery, the process is much simpler. Liquibase customer benchmarks show that we make database releases 80% faster.

Liquibase supports Lean principles

Database changes are an essential part of the overall continuous delivery process. Liquibase helps you deliver faster by eliminating wasteful task switching and handoffs that cause delays.



Liquibase monitors and measures database deployment metrics

We provide a Deployment Management Console that automatically manages and monitors the status of every database deployment across the enterprise.

Liquibase helps developers and DBAs share

We help improve the developer-DBA relationship by helping these functions work closer together in the process so that they are sharing the responsibility for the application as a whole.

Ready to find out what Liquibase's database delivery solutions can do for your process?

Schedule a Demo >

About Liquibase

Liquibase is on a mission to transform the way businesses build software by radically improving and simplifying the application release process with automated database solutions. Technology executives need to speed up the application release process to deliver better customer experiences, faster. Liquibase enables enterprises to shorten the time it takes to bring apps to market while eliminating security vulnerabilities, costly database errors, data loss, and downtime. Learn more about Liquibase.

www.liquibase.com | info@liquibase.com