

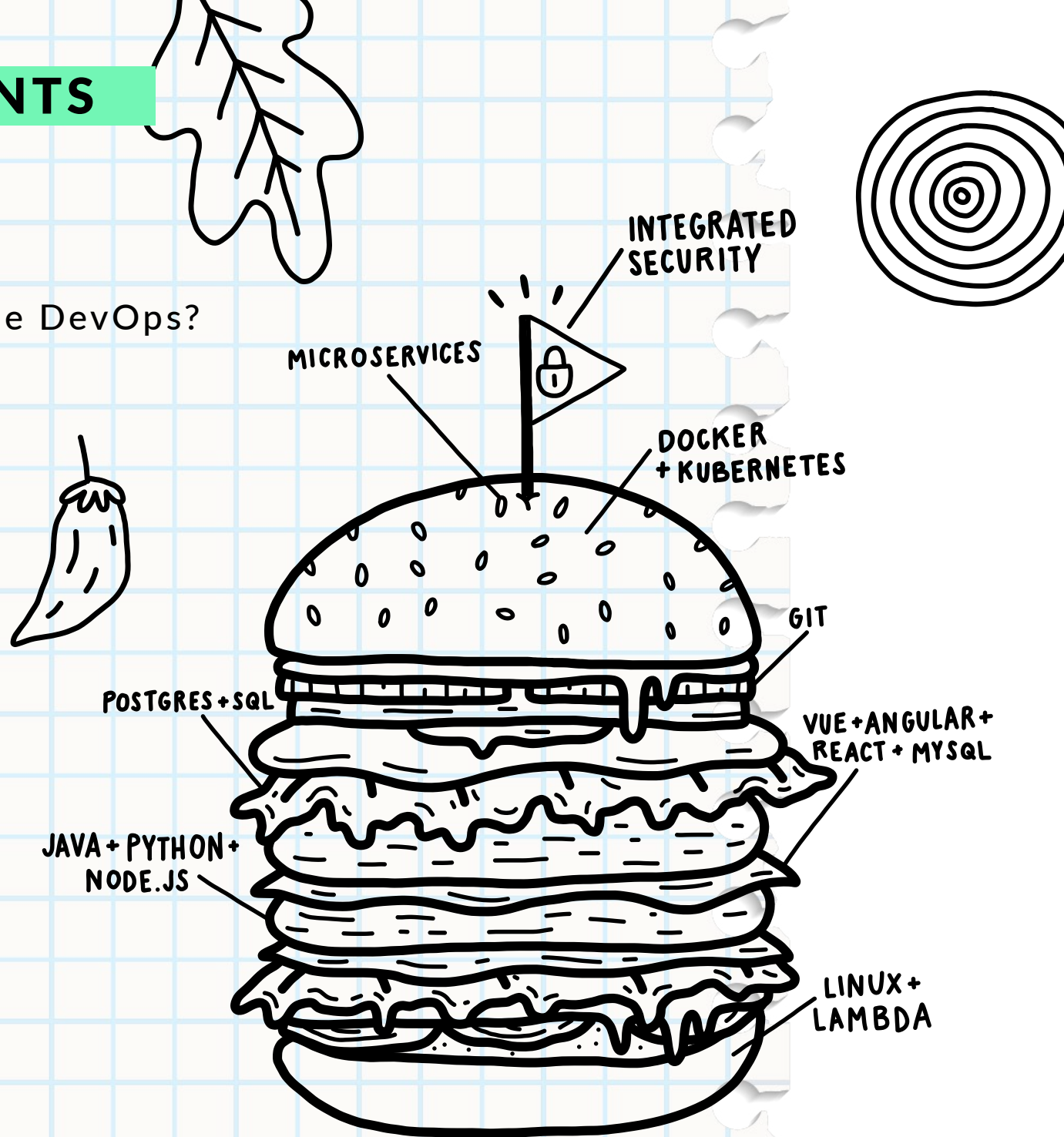


The DevOps Experience

How putting the right people in the right places can deliver better outcomes

TABLE OF CONTENTS

- 3 Intro
- 4 DevOps Hangry
- 5 Are you Ready for Agile DevOps?
- 6 The DevOps Crew
- 7 DevOps Workflow
 - 8 Plan
 - 9 Code
 - 11 Test
 - 12 Deploy
 - 13 Monitor
- 14 About the Author
- 15 MetroStar Systems
- 16 References



Intro

Imagine you're walking into a gourmet burger spot. You know, the one you *need to try* because everyone's talking about it. You place your order, take a seat in the ambient, open kitchen dining experience, and hope it lives up to the hype.

Your order is then passed to a crew of burger experts working together in a perfectly choreographed workflow to create a masterpiece that ensures customer satisfaction every time. This workflow is designed to ensure seamless coordination and collaboration between the customer, head and sous chefs, line cooks, servers, and maître d'. The restaurant industry has perfected these hand-offs over hundreds of years.

Unlike the restaurant industry, today's technology teams don't have the luxury of perfecting their delivery workflows over the course of a century. However, much like a gourmet burger spot, they can take cues from ensuring each piece of the assembly is perfectly orchestrated. This alignment is achieved by implementing DevOps: developers and operations working in constant coordination (along with security, UX, and QA) to ensure successful outcomes.

When it comes to DevOps (if we're being honest), it's all about having the right culture, people, and workflow. Let's dive in, we're already drooling to learn more.



DevOps Hangry

HANGRY

[Hang-gree] *adj.*

When your delivery teams can no longer work together because they are in a constant state of blame, frustration, and delay, impacting broader project and business goals. We'll call this the downward spiral.

The Culture Cure

The speed of IT demands a change in the way we deliver projects, with a focus on reducing the process from months and years to hours and days. The rapid adopters and early innovators of DevOps ([see the Phoenix Project](#)) realized that increased delivery speeds enabled broader enterprise transformation.

The Ways of DevOps

Knowing the culture drivers and corresponding gaps are critical insights into making the shift from siloed to unified teams. The marriage of each team's unique "flavor" streamlines workflows and ensures a prosperous technology value stream. Like all successful marriages, there must be understanding and commitment to the culture, security, and communication rhythms each team is expected to operate within, once unified under DevOps.

The Ways to an Award-Winning DevOps Culture



The First Way:

Flow – ensuring the whole is greater than the parts



The Second Way:

Feedback – enabling the reciprocal fast and constant feedback from Dev to Ops



The Third Way:

Continual Learning and Experimentation – creating a culture that is continually learning and growing

The foundational principles of Lean, Agile, and the "three ways" are prerequisites to overcoming DevOps hangry.

Are You Ready for DevOps?

DevOps: Well Done

Implementing DevOps goes beyond tools, technology, and delivery teams. To do DevOps well requires leadership buy-in, policies, metrics, and organizational alignment. You can assess your organization's DevOps readiness by answering the questions aligned to each way below.



Flow

- Is your work visible?
- Does your team use Kanban boards or sprint planning boards?
- Are you limiting Work in Progress (WIP)?
- Are you reducing your batch sizes?
- Are you eliminating hardship and waste in the value stream?



Feedback

- How quickly/frequently are you generating feedback?
- Are you seeing problems as they occur?
- Is your team swarming the problem to solve it?
- Is everyone in the value stream finding and fixing problems as part of their daily work?



Continual Learning & Experimentation

- Is your organization actively seeking and sharing information to better achieve your mission?
- Does your leadership put an emphasis on learning?
- Does your organization inject resilience patterns into daily work?

If most of your answers are no, it's time to build a DevOps culture roadmap...

If most of your answers are yes, your team is ready for DevOps...

The DevOps Crew

[Customer]



Product Owner

Provides requirements based on business drivers and validates that deliverables create value.

[Head Chef]



Solutions Architect

Coordinates and manages the customer's architectures and helps teams build automation and efficiencies into design.

[Sous Chef]



DevOps Engineer

Handles tasks across the development lifecycle including continuous integration, configuration management, and infrastructure.

[Server]



User Experience Designer

Participates at every stage of delivery to gain leadership buy-in and ensures the team stays focused on the value the project will deliver.

[Maître d']



Project Manager

Enables successful outcomes by determining key values the team will deliver based on the customer's unique needs and environment.

[Secret Saucier]



Cybersecurity Analyst

Ensures cybersecurity is integrated at each phase of development and uses real-time monitoring to identify and mitigate findings.

[Line Cook]



Software Developer

Codes business requirements using Test Driven Development (TDD) and assists with design and operational improvements.



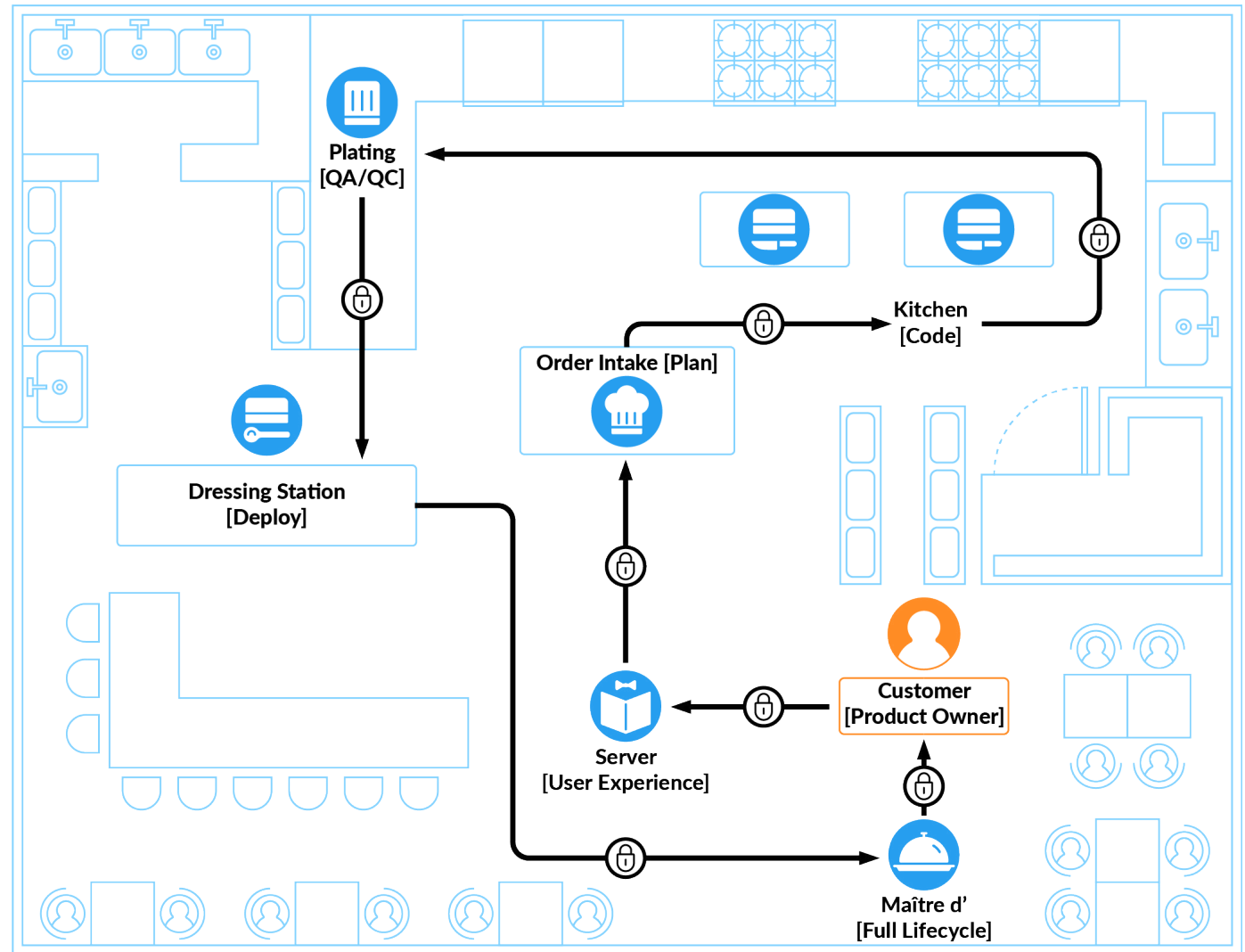
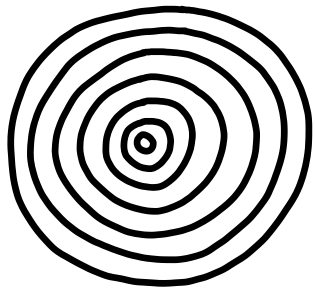
Transforming Individuals into a Crew

Each member of the DevOps crew holds a shared responsibility to deliver rapid, high quality builds. Successful crews operate on a set of shared characteristics: diversity, autonomy, colocation, transparency, peer recognition, and community.

DevOps Workflow

You've likely seen the DevOps process represented as an infinity symbol, which demonstrates continuous, iterative delivery.

To illustrate the more practical flow of the DevOps organization, we've aligned each DevOps phase to the assembly of our hamburger in our open kitchen restaurant.



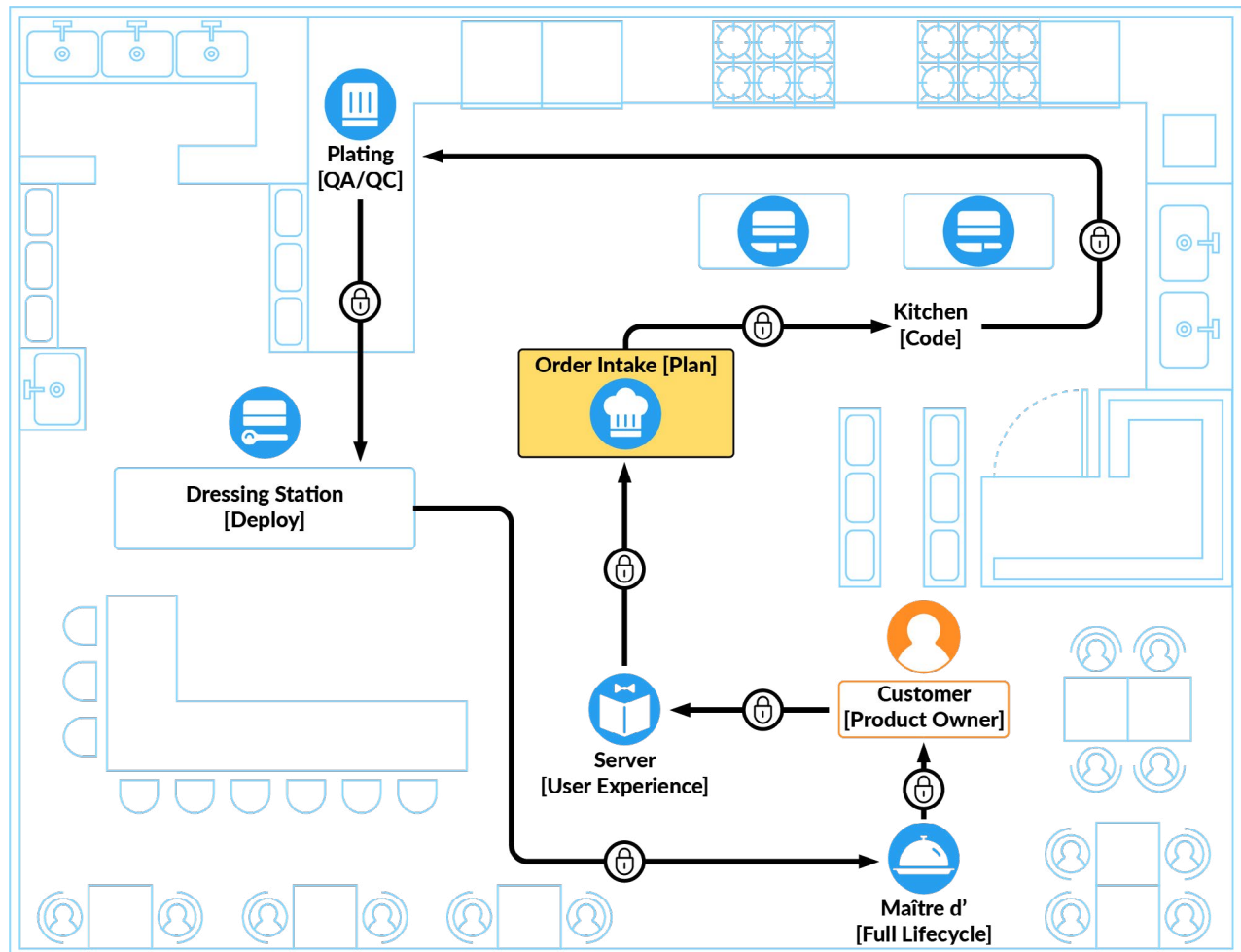
Plan

The planning phase of the DevOps lifecycle is about understanding the value the project will bring to the customer. This is accomplished by digging in to what the customer needs, and differentiating these from their wants.

The goal of this phase is to ensure the team delivers a delightful experience led by:

- **User Experience** team (represented by the server) gathers the requirements and vision (order) for the project
- **Project Manager** (represented by the Maître d') ensures all the roles are aligned and additional considerations around security and policy are incorporated
- **Project Manager** ensures the broader team understands the order and has the opportunity to provide feedback on timelines and milestones

The artifacts collected during the Plan phase are used to build a product roadmap, which guides and prioritizes future development. The roadmap is recorded and tracked using a ticket management system such as Jira or Azure DevOps. Tracking project progress, issues, and milestones is important for the process to flow to the next step.



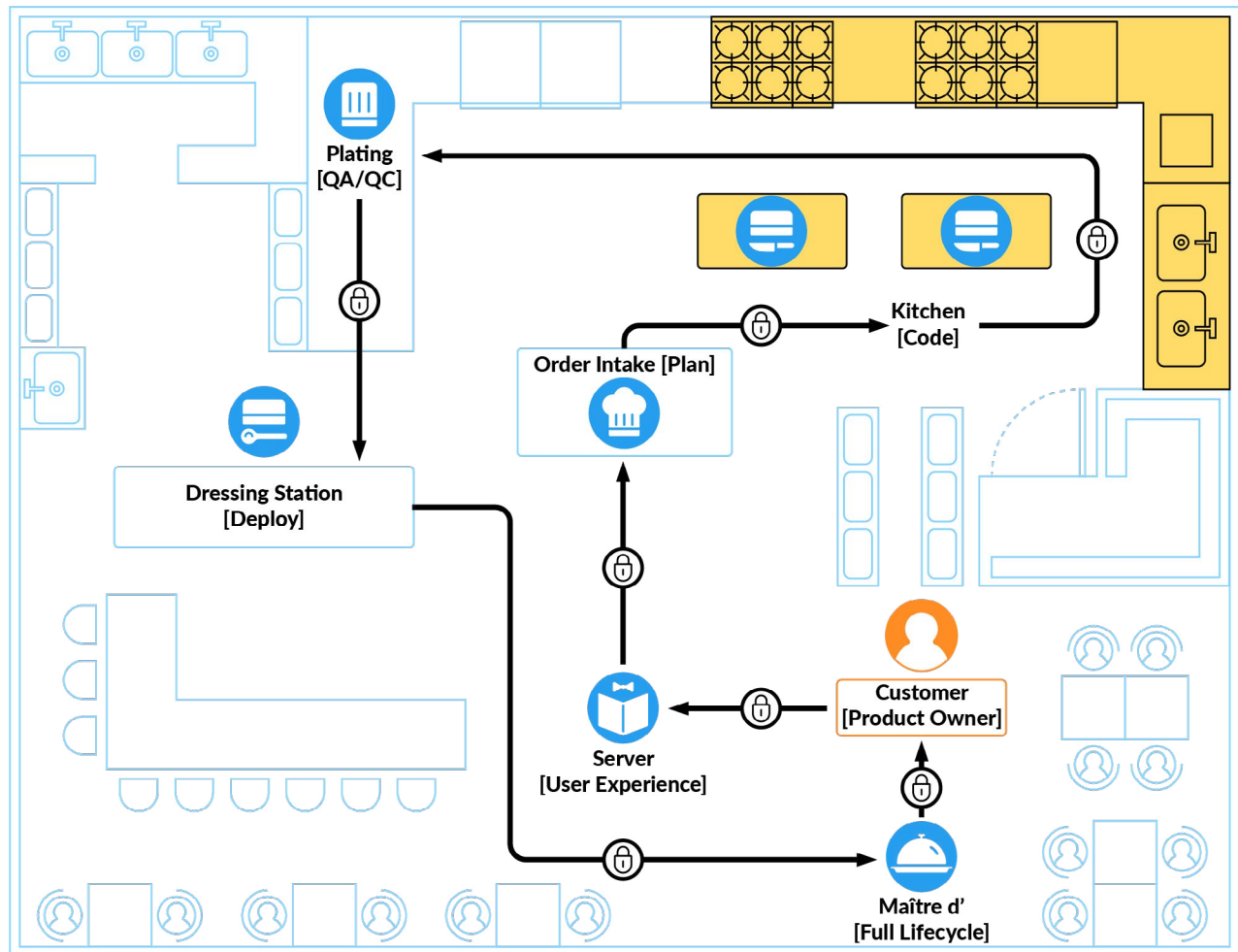
Key Activities

- Assess requirements
- Design mocks and wireframes to help scope requirements
- Create minimal viable product (MVP), with base features
- Align roles and responsibilities aligned to Agile methodology

Code

This phase is where the building happens. Our sous chef (DevOps Engineer) and line cooks (software developers) choose the right combination of ingredients: programming languages, source code management, automation tools, etc. that will make up our DevOps tech stack. From there, we move into the CI/CD development workflow that brings everything together:

- 1. Continuous Integration (CI)** - A software development practice commonly applied in the DevOps process flow. Developers regularly merge their code changes into a shared repository where those updates are automatically tested. This practice ensures the most up-to-date and validated code is always available for delivery.
- 2. Continuous Delivery (CD)** - is the next step from CI. Code changes are automatically built, tested, and packaged for release into production. The goal is to release end-user updates rapidly and without disruption. To do this, CD automates the release process (using the ready-to-deliver code that was automatically verified in CI) so new builds can be released at the click of a button.



Key Activities

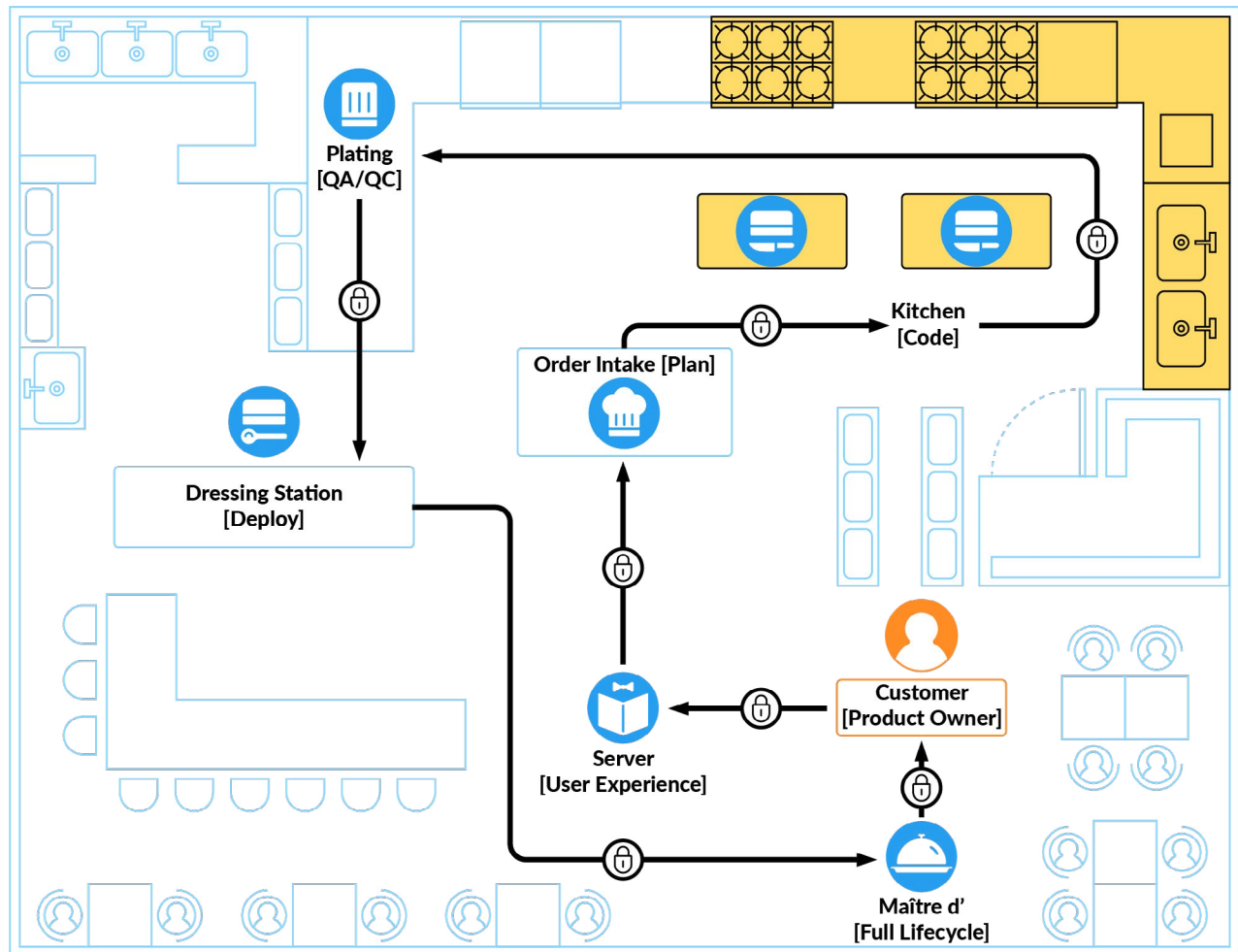
- Increase code integration rhythms to daily or hourly
- Leverage automation tools to check and test code quality
- Detect and fix bugs quickly
- Increase transparency between dev and ops teams

Code

- 3. Continuous Deployment** - As DevOps organizations mature, they adopt continuous deployment, which builds upon delivery. Continuous deployment is the fully automated version of CD with no human (i.e., manual) intervention necessary.

Automating manual processes, frees up the Ops teams (who often oversee the manual processes) and avoids slowing down delivery. Continuous deployment is a great goal for a DevOps team, but it is best applied to a stable and mature DevOps process, after all the kinks have been ironed out.

- 4. Continuous Monitoring** - Throughout the development pipeline, your team should have measures in place for continuous monitoring and reporting on the end products and systems. This process allows Ops to identify issues and notify developers in real-time. Continuous feedback ensures higher security, system availability, and a more agile response when issues do arise.



Key Activities

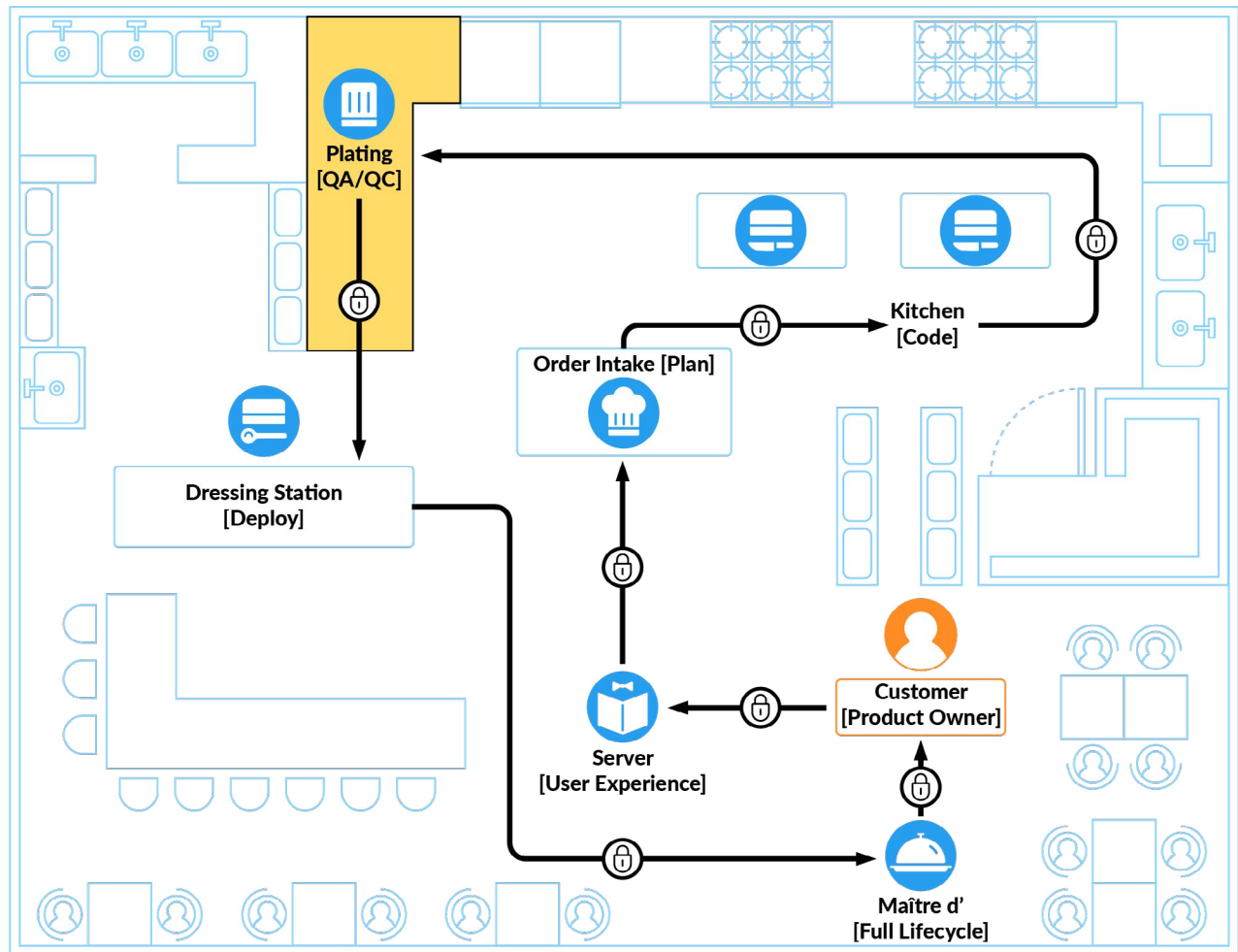
- Automate code releases into the repos
- Incorporate test, code release, and security automation at each stage
- Receive and incorporate user feedback continuously

Test

Testing is the continual process of validating what's being developed against the requirements. The plating team (QA/QC) is integrated at the planning phase to review documents, build test cases, review user stories, and assess requirements. The plating team partners heavily with ops and the Product Owner to build functional tests that support the projects needs.

Continuous testing removes the hand-off bottleneck between developers and testers common in the traditional software development lifecycle by cutting down the feedback cycle.

If a build is successful and validated by automated unit tests, it will automatically be deployed to a staging environment for a deeper review and analysis. This is truly a core piece of how a successful DevOps pipeline is created today.



Key Activities

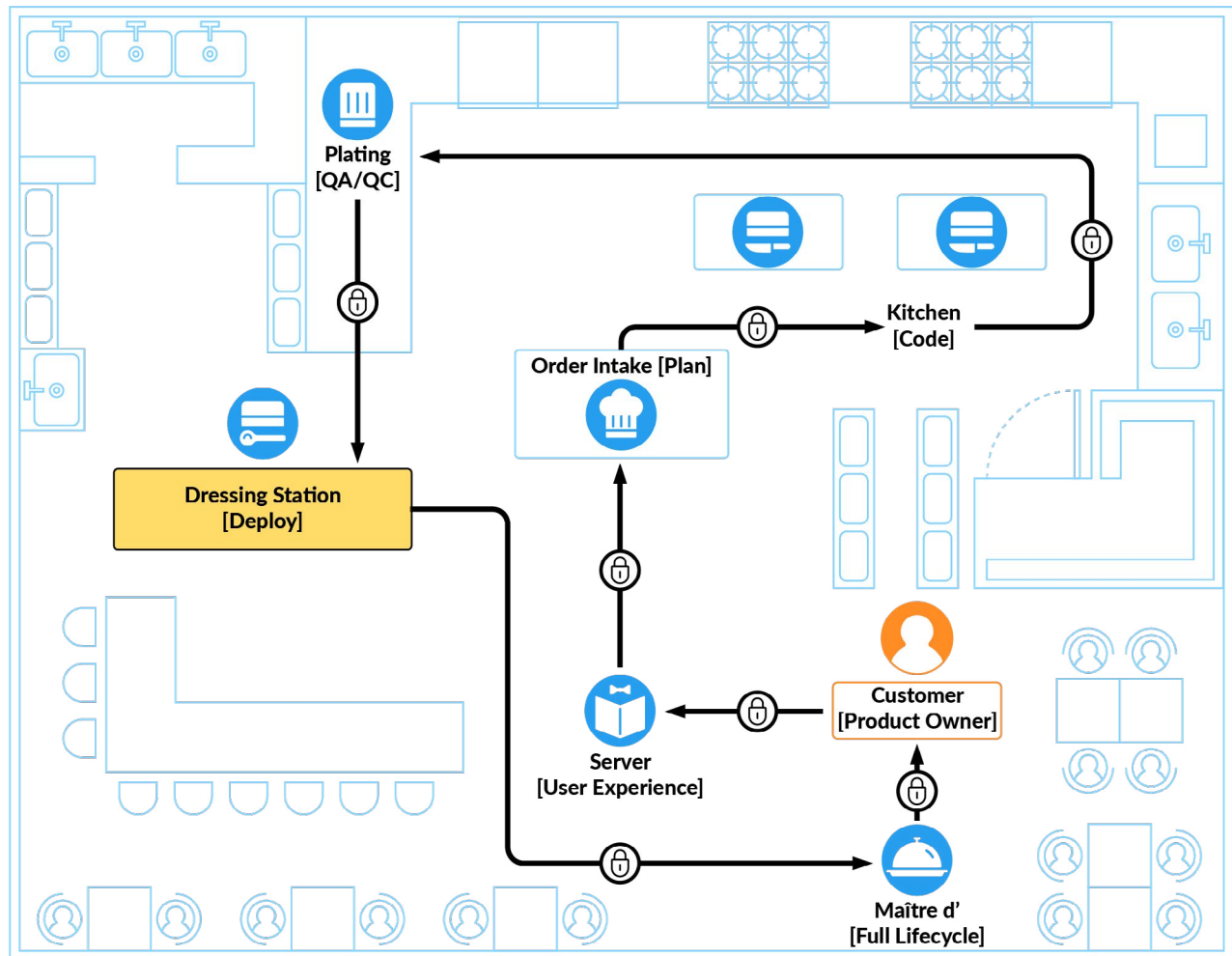
- Conduct Dev unit tests
- Release candidate branches
- Pull and build code for deployment automatically

Deploy

The deploy phase is where all of the collaboration and coordination of the previous phases comes together. All of the code that's been created, tested, automated, secured, and aligned to user requirements move from the test environment (back of the house) to the production environment (front of the house).

Great DevOps enables continuous deployment (releasing new code into production multiple times per day or week). Crews can fully automate the testing and delivery of your software in multiple stages all the way to production or set up semi-automated processes with approvals and on-demand deployments.

This phase of the process is less hectic since the same Infrastructure as Code that built the test environment was already successfully implemented. Fingers crossed things go off without a hitch.



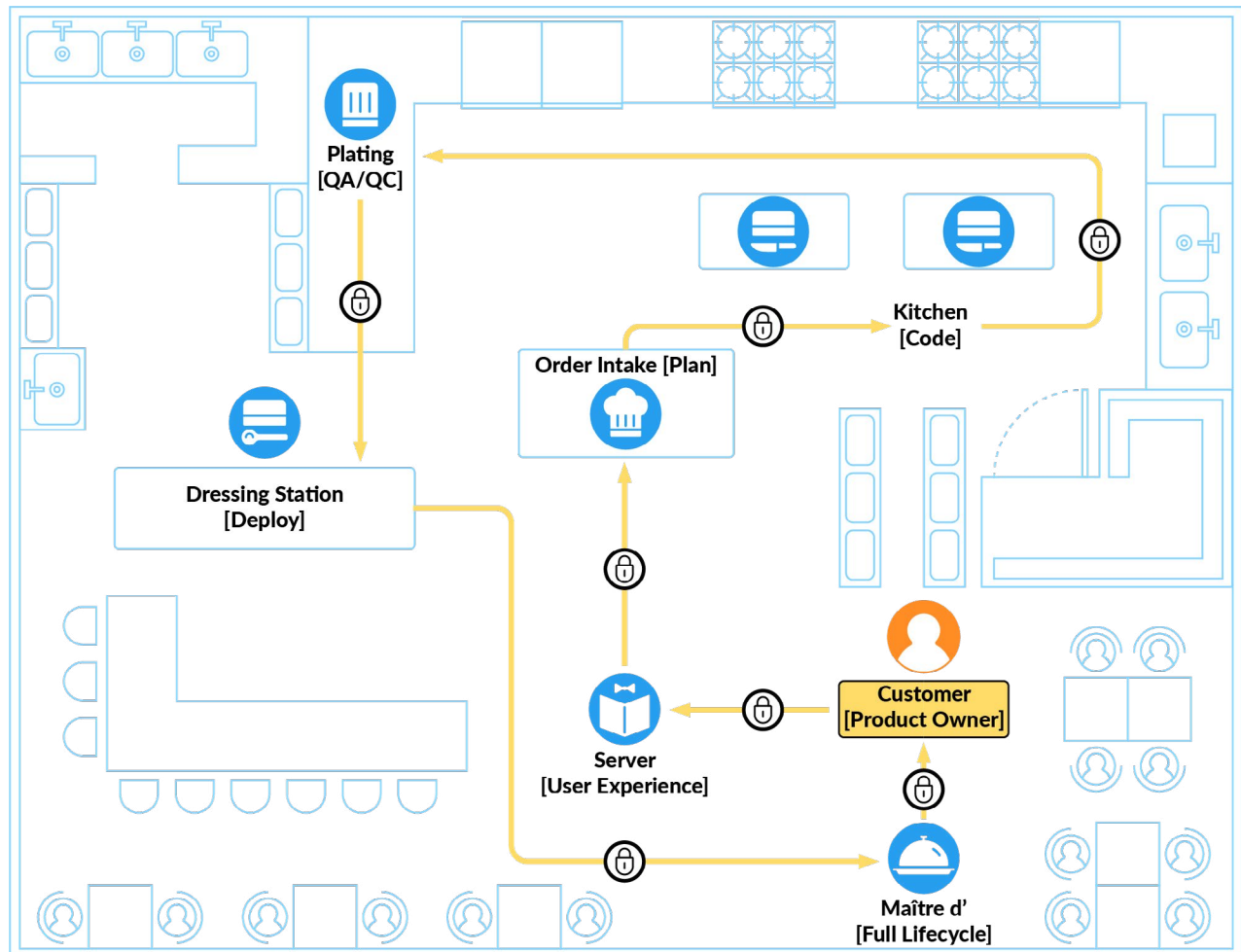
Key Activities

- Accelerate feedback loops using more frequent releases
- Test and verify new features early
- Reroute changes if problems or errors arise in production

Monitor

To properly execute the monitor phase of the process, the team needs the ability to watch over different application and infrastructure components that require compute, storage and networking resources at scale.

It's common for developers to carry "paggers" to continuously improve the operations over time by detecting downtime and correcting application of infrastructure performance or outages.



Key Activities

- Automatic detection of failed processes and resources
- Visualize throughput of your systems
- Keep your environments stable and efficiently manage resources
- Triage alerts directly to the right people to maximize uptime

About the Author



Vy Truong
Chief Innovation Officer, Client Solutions Group
MetroStar Systems



Vy joined the company in 2004. His career with MetroStar began at the Department of State (DoS) where he quickly translated his technical delivery expertise into organic growth strategies transforming DoS into one of MetroStar's largest customers.

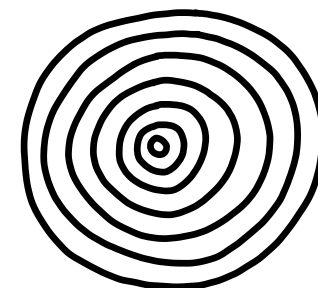
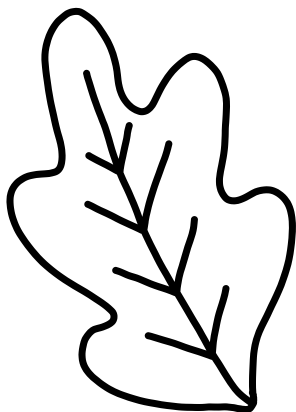
As former head of Civilian Operations, he managed a \$250M portfolio with 150+ resources across 40 contracts and task orders leading the integration of technology teams and customer delivery.

Today, he oversees the Client Solutions Group (CSG) a dynamic group of technologists developing cutting-edge solutions in Cybersecurity, Digital, and Enterprise IT. The CSG also leads the execution of DevOps roadmap development and integration for public-sector customers.

About the Client Solutions Group

Client Solutions Group is a group of technologists hyper-focused on R&D that pushes the boundaries and executes complex solutions using cutting-edge tools, technologies, and approaches. We align innovation to corporate strategy by amplifying thought leadership, supporting customer delivery, building technology partnerships, and mentoring future technologists.

To book a demo or whiteboarding session with our Client Solutions group, email us at: info@metrostarsystems.com



MetroStar Systems



We are builders—technologists, artists, and everything in between. We know the best teams are ones where different backgrounds come together to power change. Together, we build technology solutions that dramatically transform the way government and citizens interact in the digital age. We are not like other government contractors, by design. We know there are always better, faster and more cost-conscious ways to achieve transformation. We use this curiosity to drive continuous improvements at the individual, team, corporate, and customer levels. We love technology. But we love people more. Our commitment to our customers, and to each other, means we always put people first. We don't skimp on the important stuff: a culture where all people feel included and know they belong to MetroStar's award-winning community and culture. This commitment has allowed us to build trusted partnerships across dozens of federal and DoD customers, many of which have chosen us as their partner of choice for mission-critical projects for over a decade.

To learn more about MetroStar, please visit metrostarsystems.com

References

“DevOps Checklist - Azure Design Review Framework.” DevOps checklist - Azure Design Review Framework | Microsoft Docs. Accessed January 6, 2020. <https://docs.microsoft.com/en-us/azure/architecture/checklist/dev-ops>.

Kim, Gene, Kevin Behr, and George Spafford. The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win. Portland, OR: IT Revolution, 2018.

Kim, Gene, Patrick Debois, John Willis, Jez Humble, and John Allspaw. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. Portland, OR: IT Revolution Press, LLC, 2017.