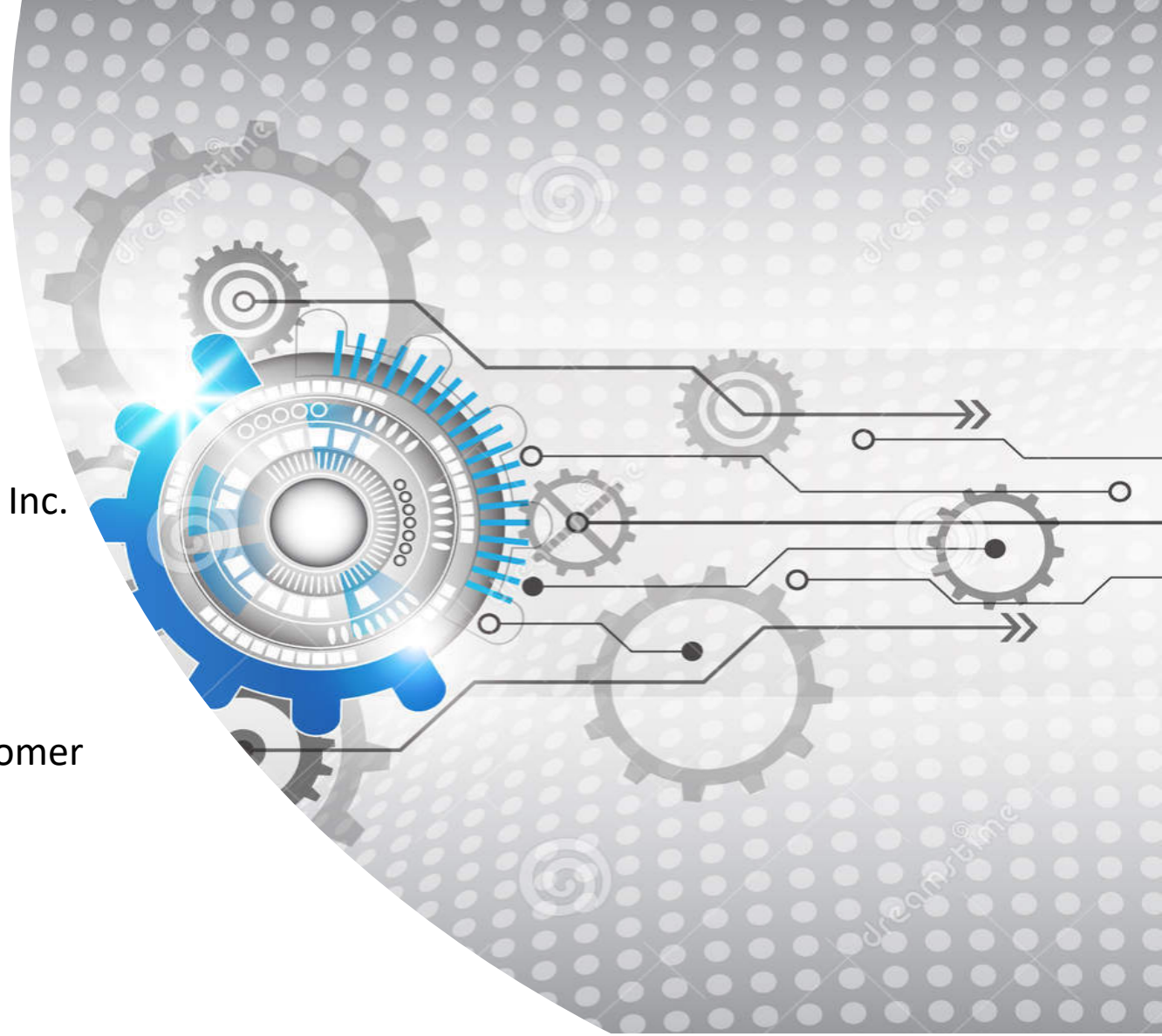
Welcome to DevOps Victoria

Continuous Integration Methods symposium

Topics Speakers



- Welcome & Introduction, Brent Reed – Founder, Tactec Inc.
- CI in Action, Karim Mustafa DevOps Engineer - BC Government (GDX)

---------- 10 min break ----------

- DevOps Metamorphosis, Sunil Mavadia – Director Customer Success, XebiaLabs

TACTEC
Strategic Solutions

Upcoming Events & Thanks

Agile Open Canada, Banff – May 29, 30

DevOpdays Victoria, May 25

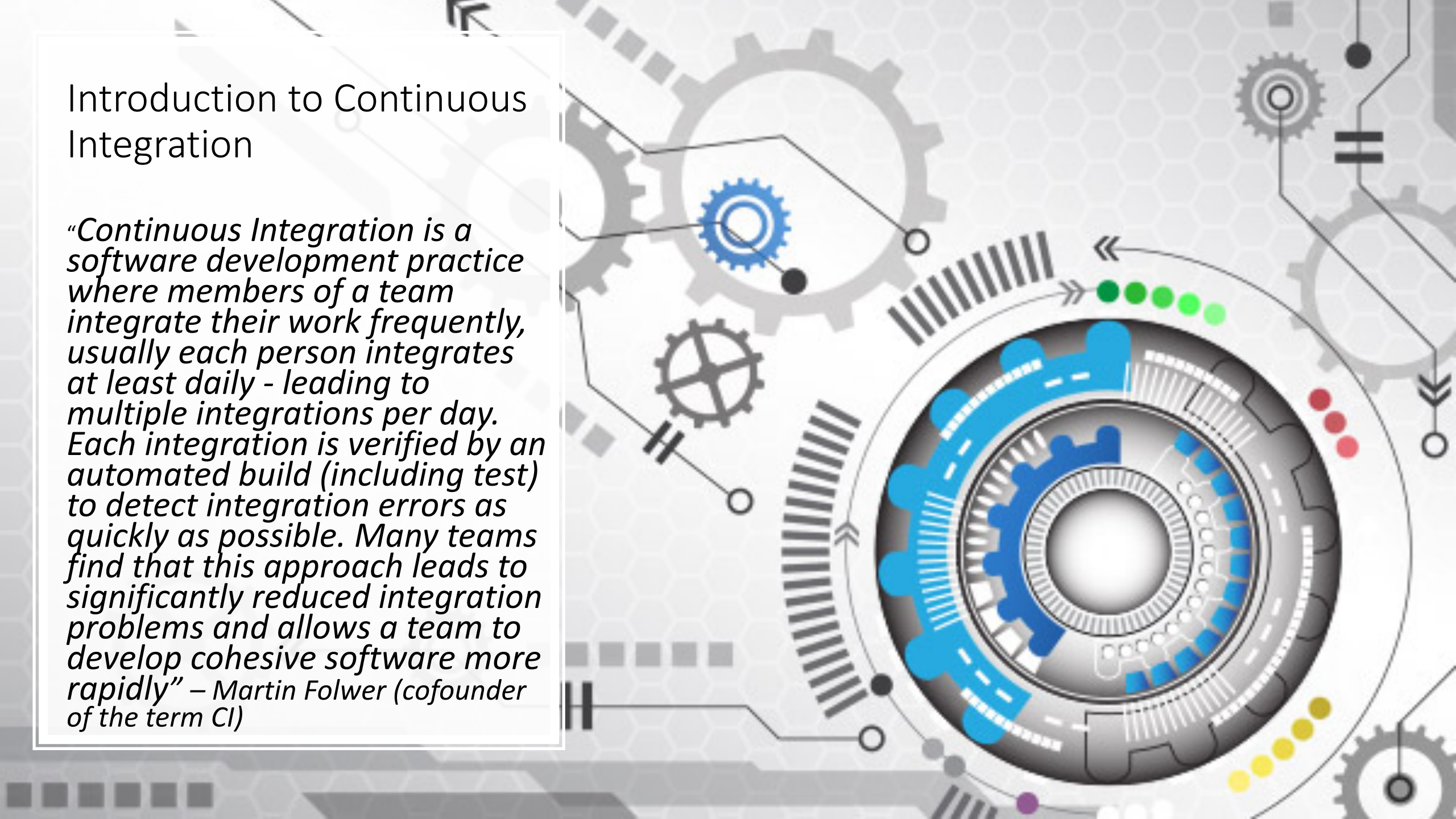
Tactec Inc. - Silver Sponsor

Facilities

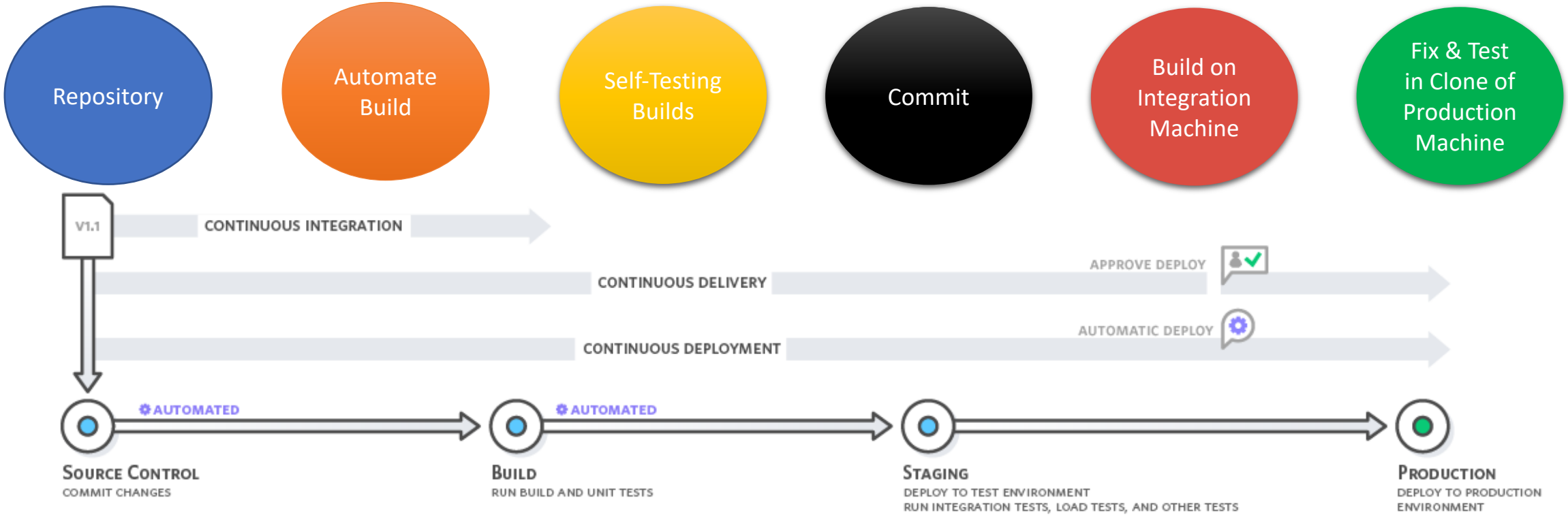
Sponsor & Host

Introduction to Continuous Integration

"Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly" – Martin Folwer (cofounder of the term CI)



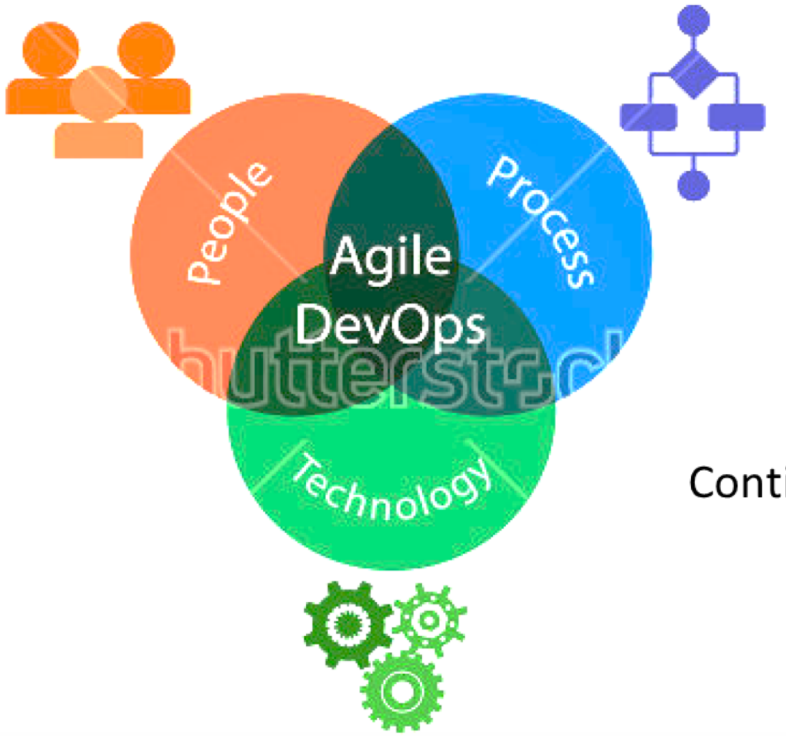
Continuous Integration - Overview



Communication is key – Let everyone see what's happening (notification, workflow, orchestration…)

TACTEC
Strategic Solutions

The big picture



Continuous Integration

Continuous Delivery

Continuous Deployment

Continuous Improvement

Presentation and Live Demo – 30 mins

Karim Mustafa – BC Gov - GCPE/GDX, DevOps Engineer



TACTEC
Strategic Solutions

DevOps Metamorphosis – 45 mins

- Sunil Mavadia – Director Customer Success, XebiaLabs



TACTEC
Strategic Solutions

Closing words – 5 mins

- More QA? – ask on our meetup discussion board ☺
- Panel Discussions?
- How about a DevOps hack-a-thon?
- Polls and suggestions coming on our meetup
- New venue for next meetup (June timeframe)
- Sponsors and volunteers?
- What more do you want?
- Follow us on Twitter @TactecSS, LinkedIn Tactec Strategic Solutions Inc.
- Blog coming May - on www.tactec.ca (links to eDevOps and materials)

TACTEC
Strategic Solutions

A Blueprint for a Successful DevOps Metamorphosis

Sunil Mavadia
Director of Customer Success

XebiaLabs

What is Driving the Interest in DevOps?

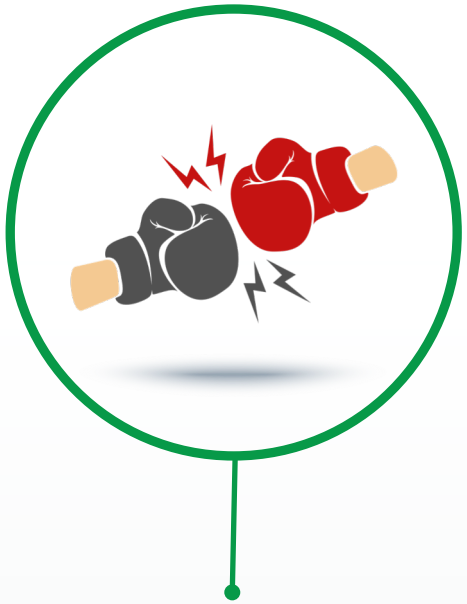
- Increased customer expectations, fuelled by mobile applications requiring fast changes to respond to short feedback cycles
- Increased Time-to-Market – streamlining the automated deployment approach
- Multi-channel apps: rise of the need to coordinate releases across multiple platforms, teams and technologies
- Increased competition and lowered barriers to entry
- Better to plan for greatly accelerated delivery cycles now than be rushed into it by competitive pressure later
- Central challenge: how to accelerate while maintaining or increasing quality?

Testing, testing, testing…which means deploying, deploying, deploying

Why Do We Need a Blueprint for DevOps?

More time testing, deploying and releasing than designing and building software

Production incidents resulting from human error and long manual releases

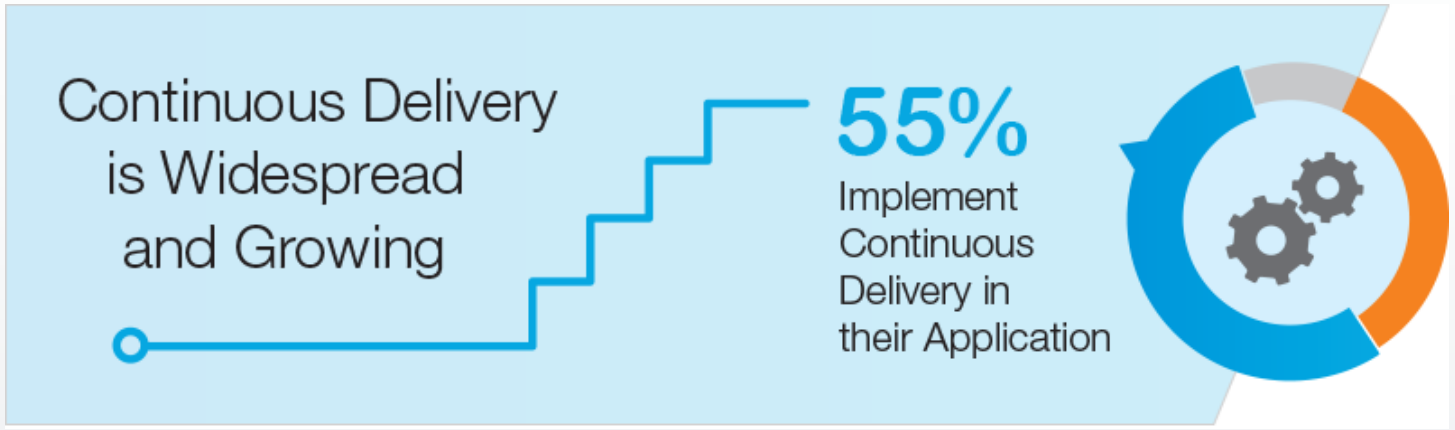


IT Development and Operations are rarely in alignment

The Big Picture: Deliver Faster!

Exciting times:

- Adopting new technologies to compete with speed and agility
- New methods of providing IT infrastructure and runtime platforms: Cloud, Virtualization, Agile, Continuous Delivery, DevOps, etc.
- Microservices, Containerization and IoT are changing the way applications are developed

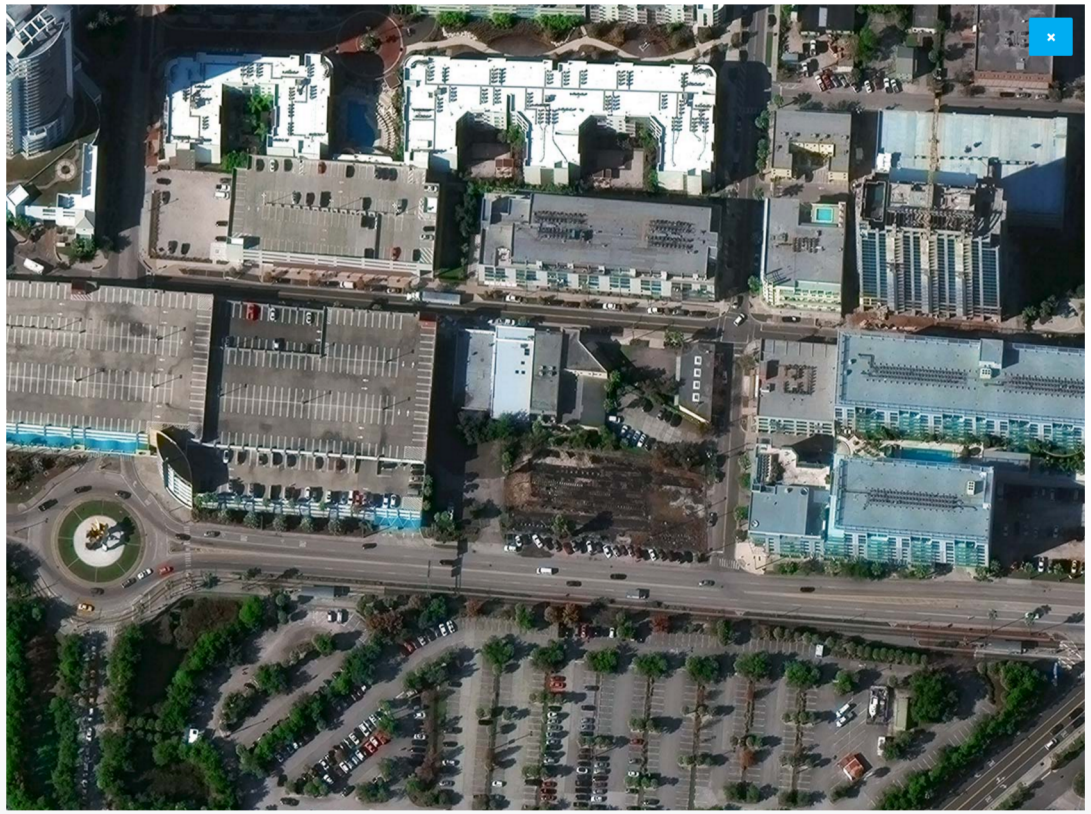


Asking the Right DevOps Questions…

You have a complex enterprise environment. How do you:

- **Leverage existing investment** in Cloud, Virtualization and other infra technologies?
- **Integrate existing applications and services** in your new patterns and processes?
- **Support a diverse set** of teams and applications at different stages of delivery maturity?
- **Identify the biggest bottlenecks** so that improvements and investments can be targeted to where they are needed most?
- **Measure changes** to track whether things are actually improving, and demonstrate value for the investment?

So How Did a Customer Do It?



- A new architecture/pipeline allowed customer to launch 2 satellites within a year.
- A record time considering the usual time frame was over 3 years to launch new satellites.

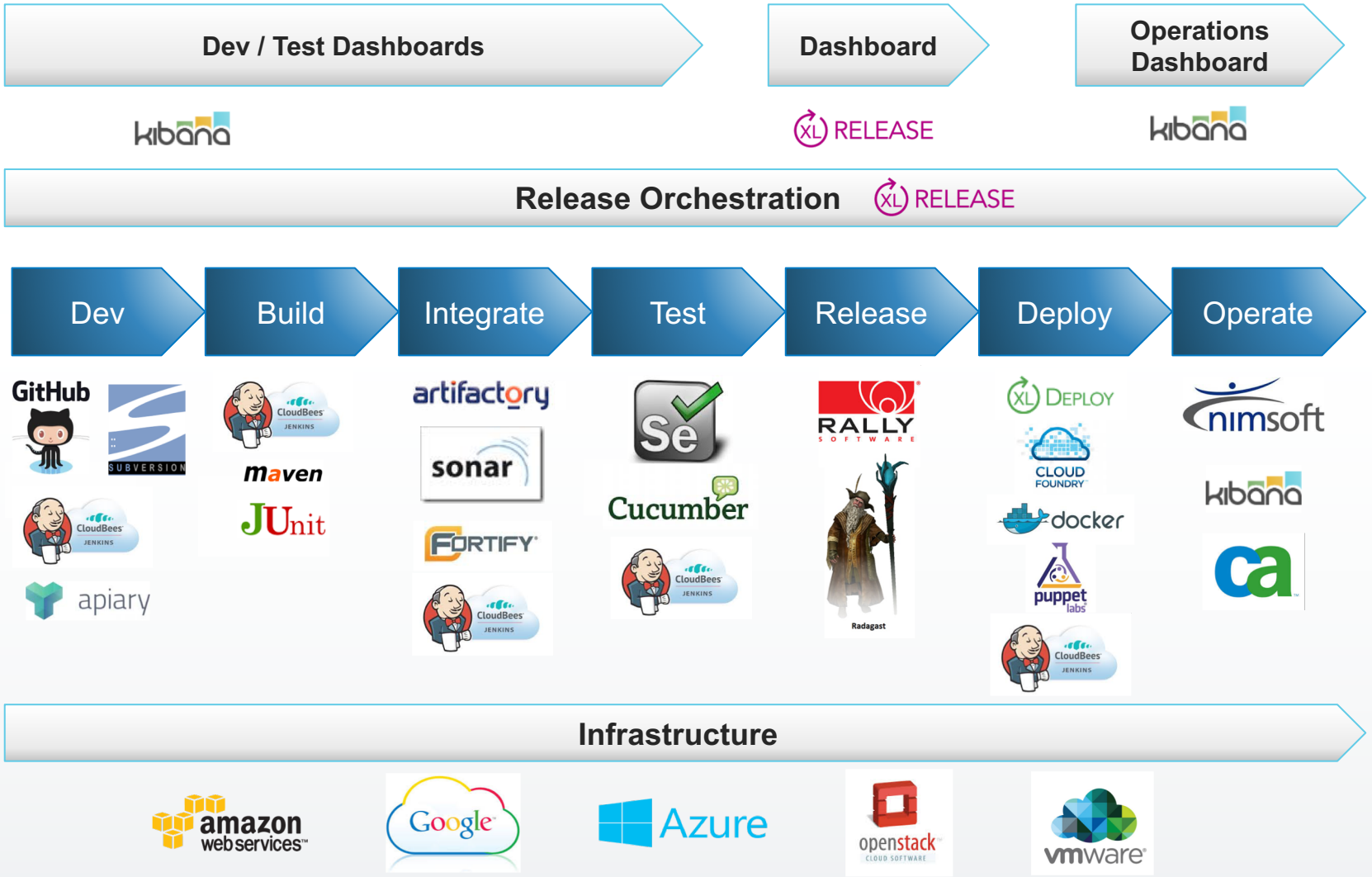
What Were the Challenges?

Challenges	Objectives
▪ Developers still spend a large degree of their time tooling up environments before generating code	→ Create instant development and deployment environments that are repeatable
▪ SDLC is defined, but application profiles and middle-ware are not standardized. Patterns exist, but have not yet been adopted	→ Generate application profiles or an “app-stack” to create repeatable patterns with a governed change and versioning process
▪ Developers don’t have access to production environments ▪ Disparate environments make it hard to do true testing early on ▪ Code promotion and integration is being done by a separate ops team using different tools, leaving knowledge gaps	→ Provide production like services (virtual networking, identity, and data-services) to be available throughout the entire development cycle
▪ Many mature processes and tools already exist; adoption of the new standards must be done in a manner complimentary (and not counter to) the current initiatives and way of working	→ Integrate into the current agile development toolsets already in use – Git, Github, Jenkins, etc., – and have one or two teams promote the benefits of the new standards

They Knew What They Had to Do..

- Provide a **more efficient code delivery framework** in support of a satellite launch in September 2016
  - Critical Development teams utilize the new code delivery framework by Q4 2015
  - Complete rollout of the new framework in 2016
- Leverage a **cloud agnostic service oriented architecture, including automation and orchestration** to deliver an agile platform for end users
- Achieve a higher level of agility and allow for rapid deployment of services to **enable Continuous Delivery**
- Leverage the existing work that has been done by customer

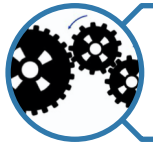
But How Do You Do It With So Many Tools?



XebiaLabs

Create a Highly Performing DevOps environment

Optimize your DevOps tooling and processes

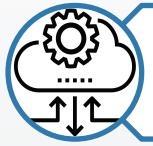


Strong source control

Zero-touch build and deploy

Test early and often

Automate everything in the pipeline, including testing



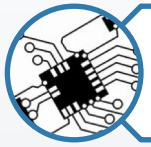
Elastic supply of cloud/infrastructure resources

Cohesive teams with shared objectives

Continuous Integration/ Continuous Delivery

Embrace failure, recover automatically and degrade gracefully

Experiment without regret

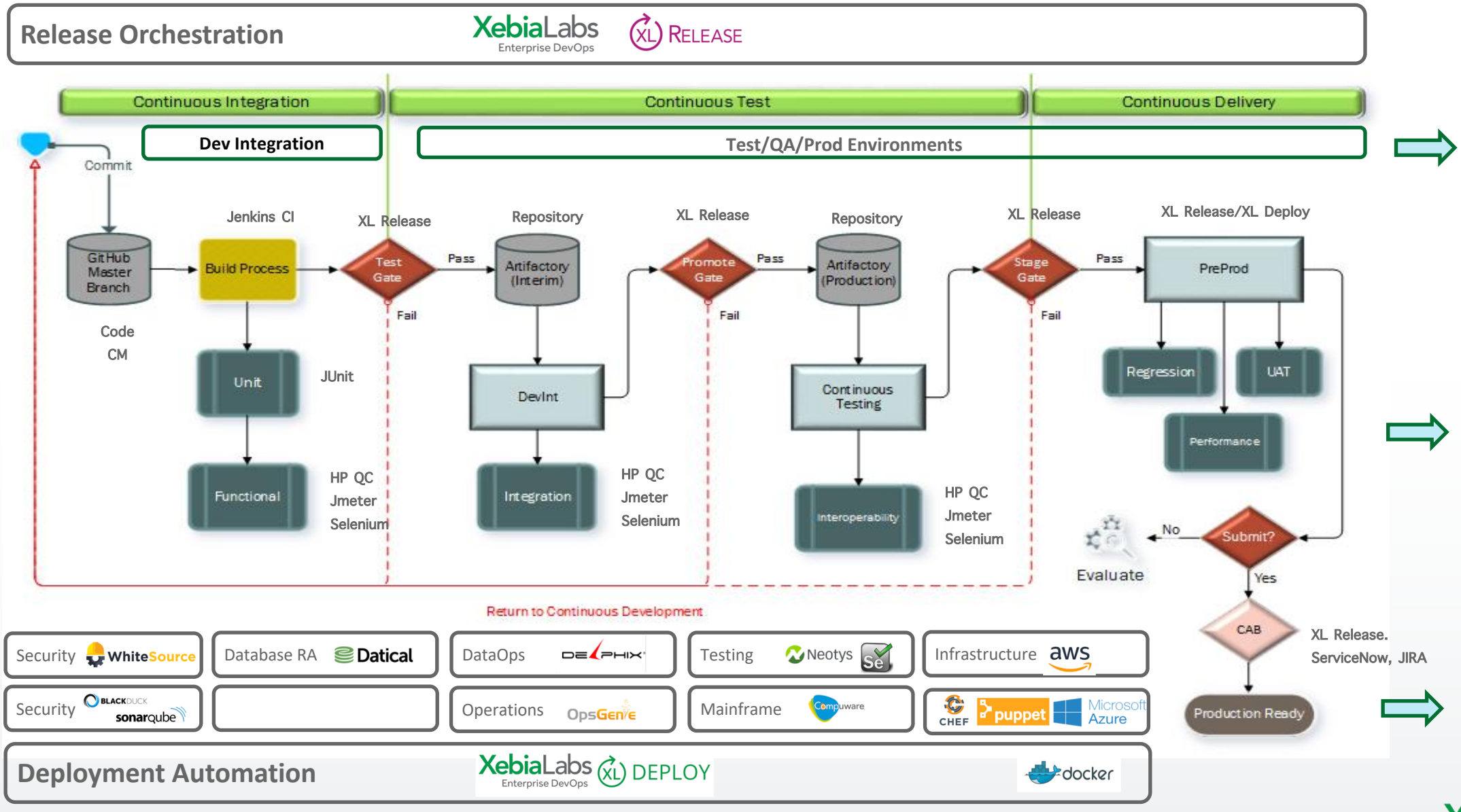


Fine-grained service architecture

Continuous Integration → Continuous Testing → Continuous Delivery

XebiaLabs

The BIG Picture



XebiaLabs

Thoughts Around Your DevOps Toolchain

- Use the DevOps toolchain as a **foundational starting point/reference of DevOps activity** to help make tool choice and usage decisions
- Use the DevOps toolchain design to **focus on ways to streamline activities** by combining, orchestrating or removing them — specific to your DevOps initiative goals (e.g., combine code, test and release, and remove tools that deliver a specific activity)
- Use the DevOps toolchain to **help identify tool integrations and handoffs**
- Understand the **skills and resources required** by understanding the activities and tools identified to deliver them

It is important to customize the toolchain activities specific to your DevOps project requirements

DevOps Toolchain Recommendations

Develop a toolchain strategy that establishes the tools' functional requirements to position existing tools and make decisions on new tools

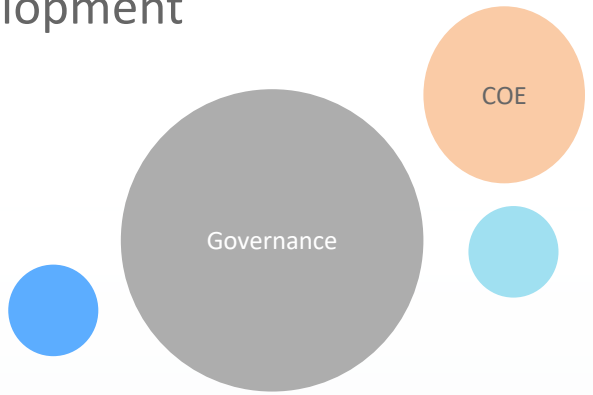
Ensure each DevOps team member understands the capabilities and role of each tool in the DevOps toolchain to avoid tool overlap and toolchain functionality gaps

Other Recommendations…

- **Set up Bodies of Governance or Centers of Excellence**

  Governance bodies must formalize, and periodically audit, the guidelines—or working agreements—currently in place for controlling which stories/work items various development teams work on

  Example areas of concentration:
  - Build
  - Test
  - Release Management
  - Security



- **Establish minimum set of criteria** governing the committal of code to the master branch of the source code repository. This provides:
  - Higher level of confidence in its quality
  - Much higher level of traceability
  - Much tighter security control – since security is involved from the beginning

Managing the Metamorphosis

- Manage and govern your DevOps adoption – across all silos
- Design approach for the support of your DevOps tools and their implementation
  - Continuous Integration
  - Continuous Testing
  - Continuous Delivery
- Establish DevOps tooling across all platforms – cloud and internal
- Implement your DevOps practices through Organizational Change Management
- Use the “As a Service” model to provide your services across all platforms

Areas of Concentration

Continuous Integration

- Create build governance specs for your target technologies, e.g., Java, Ruby, Python, .NET
- Establish ground rules for CI (Master branch check-in initiates a release so you better have tested your code)

Continuous Testing

- Test automatically at every stage without exception. Automate every aspect of testing possible

Continuous Delivery

- Think beyond code – treat all aspects of your product as configuration items, including Infrastructure. (IaC – Infrastructure as Code)
- Automate, Automate, Automate
- Make everything fully visible - full transparency
- Track every change
- Standardize by putting everything in one place

Continuous Improvement

Improve Traceability

Automate, Automate, Automate

Improve Auditability

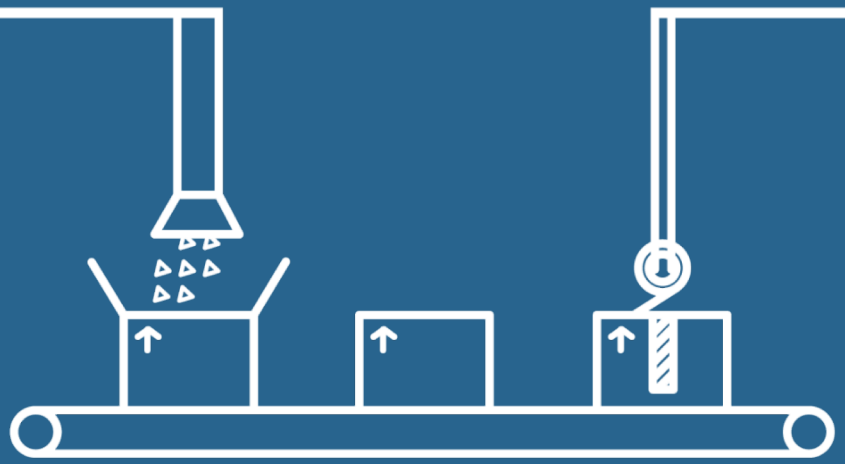
Improve Visibility

Improve Security

XebiaLabs

One Last Thing…

Self-Service Automation



Thank You!

XebiaLabs