# DevOps Automation

*Service Catalogue*

# CONTENTS

## AEM Background

AEM Corporation is a diversified services company that primarily supports federal agencies and Fortune 1000 clients. We employ leading experts in information technology; cybersecurity; data management and analysis; research, development, and evaluation; engineering; technical assistance; and operations management. Founded in 1986, we have leveraged these strengths to become one of America's fastest-growing companies. Learn more at aemcorp.com.

## DevOps Expertise

AEM works with complex applications on behalf of private-and public-sector clients, tailoring support to their needs. We supply measurable outcomes by offering substantial experience in software system development lifecycles, backed by leading qualifications in cloud, container, and CI/CD technologies.

AEM experts have deep expertise with the tools across the DevOps ecosystem that are essential to promoting a collaborative project environment. As such, we accelerate your initiatives by integrating DevOps tools and processes from the start, and we then sustain them by nurturing a continuous learning environment for your community of DevOps practitioners.
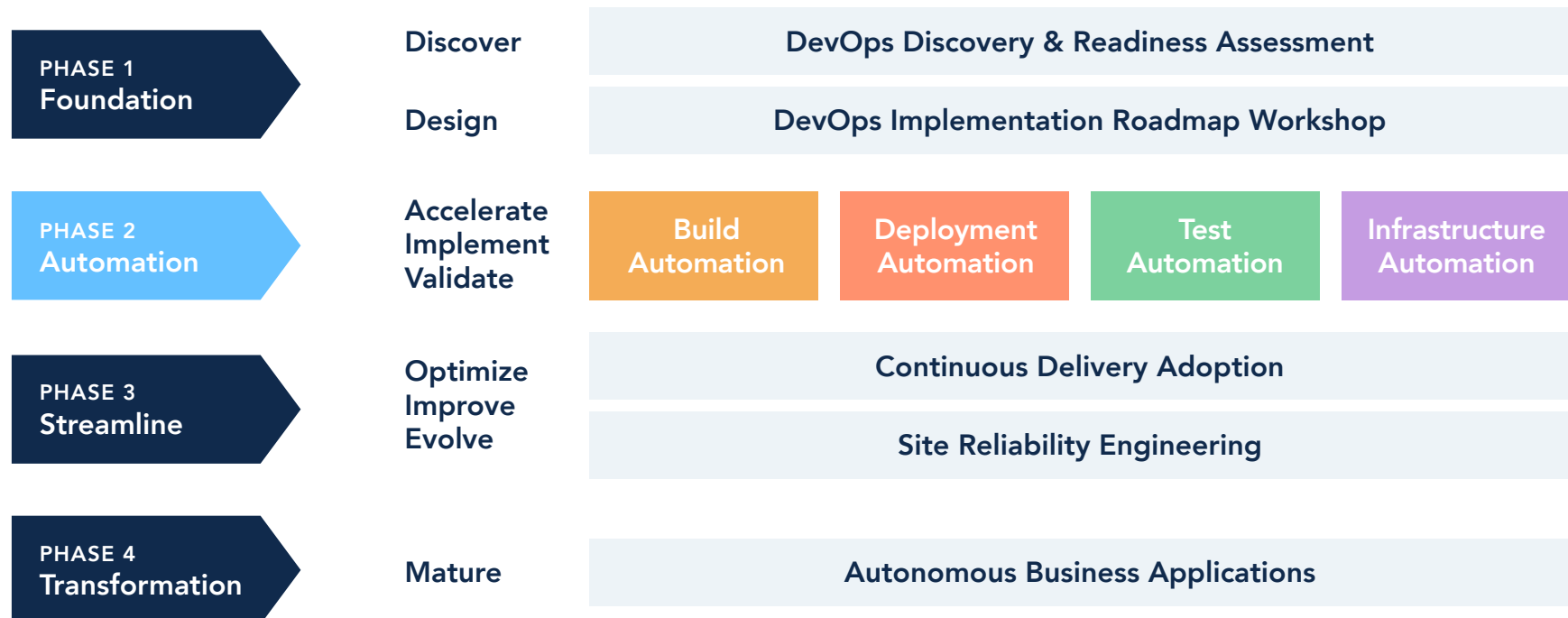
aem

# DevOps Automation

Just as the Industrial Revolution introduced the manufacturing industry to rolling assembly lines and automated processes, DevOps processes are ingrained with automation techniques that provide rapid feedback, repeatable processes, and consistent creation of business application systems. With the convergence of multiple systems and IT professionals all focusing on the delivery of these applications, these processes quickly become complex and can seem daunting to implement for the first time.

Our experts have grouped these processes into four areas of focus for implementing automation that can each help your organization gain efficiencies and improve your overall IT processes: Build, Deployment, Test, and Infrastructure. With each of these areas, AEM's experts will guide your organization in the development of optimized IT processes that adopt core enabling disciplines, monitor key performance indicators, and avoid potential red flag areas.

## Our Pragmatic DevOps Services Framework

| | | |
|---|---|---|
| **PHASE 1**<br>**Foundation** | Discover | **DevOps Discovery & Readiness Assessment** |
| | Design | **DevOps Implementation Roadmap Workshop** |

| | | | | | |
|---|---|---|---|---|---|
| **PHASE 2**<br>**Automation** | Accelerate<br>Implement<br>Validate | **Build Automation** | **Deployment Automation** | **Test Automation** | **Infrastructure Automation** |

| | | |
|---|---|---|
| **PHASE 3**<br>**Streamline** | Optimize<br>Improve<br>Evolve | **Continuous Delivery Adoption** |
| | | **Site Reliability Engineering** |

| | | |
|---|---|---|
| **PHASE 4**<br>**Transformation** | Mature | **Autonomous Business Applications** |

# BUILD AUTOMATION

Build Automation is the process of scripting and automating the retrieval of software code from a repository, compiling it into a binary artifact, executing automated functional tests, and publishing it into a shared and centralized repository.
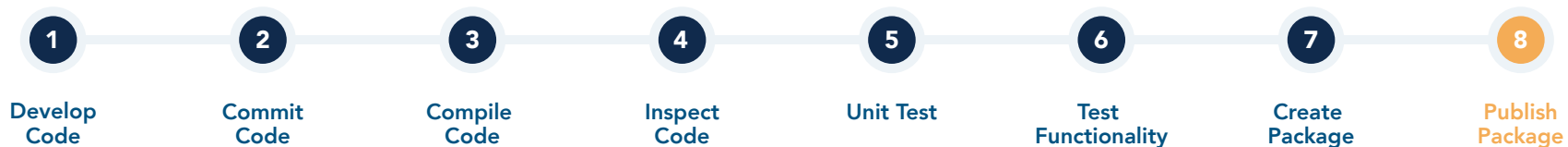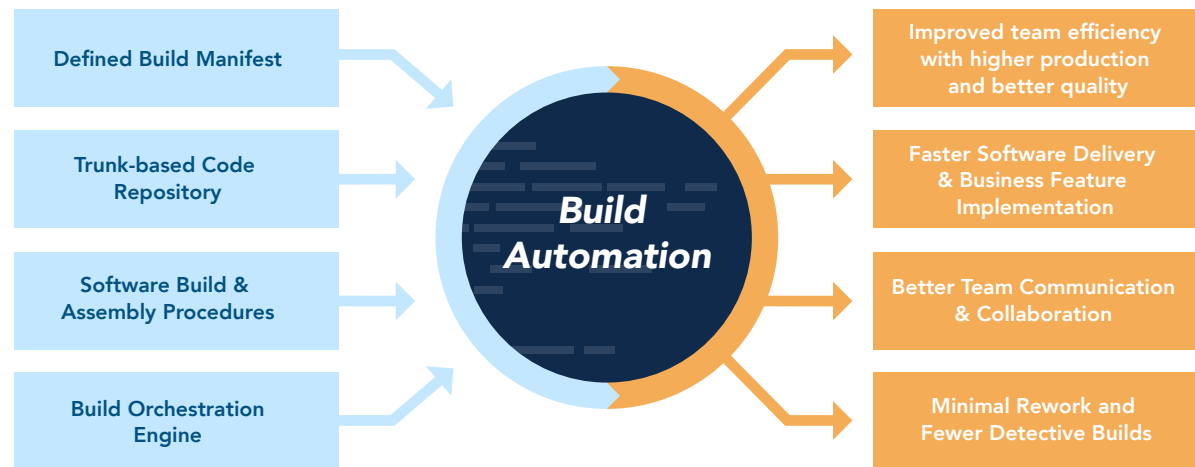
▸ Define and execute a consistent and repeatable process.

▸ Amplify feedback and improve team communication.

▸ Improve overall release deployment quality.

▸ Accelerate the implementation of desired business features/functionalities.

## Models

In this diagram, given the items on the left, implementing Build Automation will result in the items on the right. ▸

A typical automated build process is shown in the diagram below. This will vary slightly based upon team structure and organization focus. ▾

| Defined Build Manifest |
| Trunk-based Code Repository |
| Software Build & Assembly Procedures |
| Build Orchestration Engine |

**Build Automation**

| Improved team efficiency with higher production and better quality |
| Faster Software Delivery & Business Feature Implementation |
| Better Team Communication & Collaboration |
| Minimal Rework and Fewer Detective Builds |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| Develop Code | Commit Code | Compile Code | Inspect Code | Unit Test | Test Functionality | Create Package | Publish Package |

3

# Red Flag Areas

**Lack of Source Code Tool Expertise**
is a barrier to frequent code commits. This often leads to complicated integration issues and inhibits overall team visibility of the code base.

**Complicated Source Code Branching**
leads to duplicate changes and complex merges.

**Infrequent Check-ins of Code**
leads to complex change merges and delayed integrations.

**Broken Builds**
diverts focus of development teams away from new feature creation.

**Minimal Feedback**
negatively affects overall quality due to "lack of eyeballs."

**Excessive Notifications**
result in team members ignoring important alerts.

**Bloated Builds**
prevent rapid feedback and creates potential bottlenecks.

**Infrequent Builds**
delay feedback and identification of potential integration issues.

# Key Metrics

**Number of Features / User Stories per Build**
indicates the number of changes being implemented and maps to business value being created.

**Change Implementation Lead Time**
affects the number of releases per a given period and overall product roadmap planning.

**Average Build Time**
indicates the average time to perform a build.

**Frequency of Builds**
indicates the overall output and activity of the project.

**Percentage of Failed Builds**
impacts the overall team output due to rework.

# Key Roles

**Developer**
Responsible for taking business requirement and implementing in software code

**Build Engineer**
Responsible for defining the release pipelines and maintaining the build infrastructure

**Requirements Analyst**
Responsible for defining the user story to include the definition of acceptance criteria

# Service Engagements

AEM's assessment-derived solutions are tailored to your needs. We work with you to define the right offerings required to build a solid foundation based on DevOps principles. The time and scope will vary based on your needs, but representative offerings based on previous client engagements are shown below.

## Expert Consulting

### Mentoring and Implementation Consulting Services

Work collaboratively with AEM experts to mentor and provide expert consulting for automating your software build and packaging processes.

## Accelerators

### Continuous Integration Implementation Pilot

In this hands-on guided lab, participants will implement an automated software build process that includes static code analysis.

### Build Artifact Repository Implementation Pilot

Participants will expand on the automated build process to learn how to leverage central repositories for package and dependency management.

## Workshops

### Software Assembly Best Practices

Learn best practices for building and packaging software components. Topics include dependency management, build manifests, and source code management.

### Code Analysis Strategies & Tools

Learn about static code analysis techniques and common tools to use for improving software quality.

## Why AEM?

### Delivering business applications with confidence

Current work includes powering the collection of millions of data points from hundreds of stakeholders; enabling oversight for $1 billion+ in grants; and managing a global housing system that supports more than a million end users.

# 99%

success rate during production releases

Our DevOps approaches enable a 99% success rate during production release and 99%+ availability of all servers.

aem

# DEPLOYMENT AUTOMATION

Deployment Automation is the process of provisioning the application artifacts and configurations to the operating environments across the system development lifecycle. It entails a combination of application deployment automation, environment modeling, and workflow orchestration to achieve rapid delivery of application features in a reliable and orderly manner.
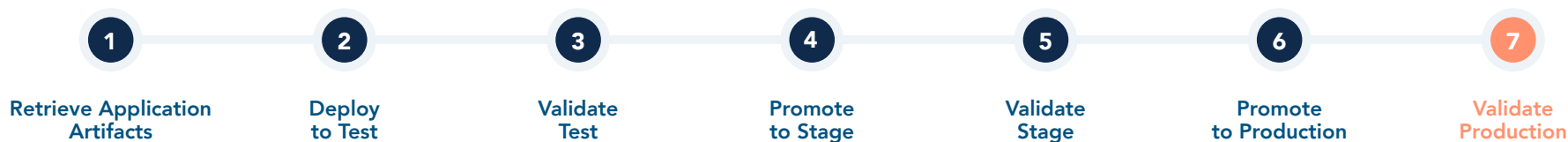
- ▶ Define and execute a consistent and repeatable process.
- ▶ Amplify feedback and improve team communication.
- ▶ Improve overall release deployment quality.
- ▶ Accelerate the delivery of application features to production.

## Models

In this diagram, given the items on the left, implementing Deployment Automation will result in the items on the right. ▶

A typical release process is shown in the diagram below. This will vary slightly based upon team structure and organization focus. ▼

| Inputs | | Outputs |
|---|---|---|
| Trunk-based Code Repository | | Reduced Employee Burnout |
| Version-controlled Application and System Configuration | **Deployment Automation** | Faster Software Delivery & Business Feature Implementation |
| Application Deployment Procedures | | Better Team Communication & Collaboration |
| Release Orchestration Engine | | Stable & Consistent Application Hosting Environments |

1. Retrieve Application Artifacts
2. Deploy to Test
3. Validate Test
4. Promote to Stage
5. Validate Stage
6. Promote to Production
7. Validate Production

aem
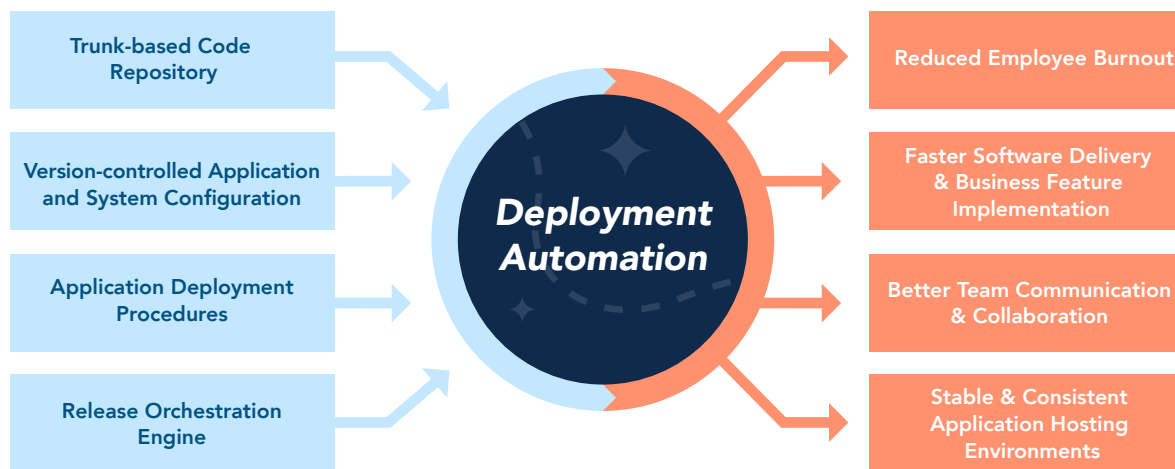
# Red Flag Areas

**Lack of Source Code Tool Expertise**
is a barrier to frequent code commits. This often leads to complicated integration issues and inhibits overall team visibility of the code base.

**Embedded System and/or Application Configurations**
often require complicated code changes to account for specific environment configurations and may introduce complex code merges for different target environments.

**Infrequent Check-ins of Configurations**
lead to misconfigured environments and tribal knowledge of environment requirements.

**Inconsistent Environments**
increase deployment tasks and introduce instability into environments due to variations in topology and configurations.

**Excessive Notifications**
result in team members ignoring important alerts.

**Long-running Deployments**
prevent rapid feedback and create potential bottlenecks.

**Infrequent Deployments**
lead to the outdated system provisioning procedures and environment definitions, which ultimately impact the overall system stability.

# Key Metrics

**Time to Fulfill Environment Provisioning Request**
redirects resources from application and business feature development and impacts time required for each release.

**Average Deployment Time**
impacts the available time for deployments.

**Number of Features/ User Stories per Build**
indicates the number of changes being implemented and maps to business value being created.

**Percentage of Failed Deployments**
impacts the overall team output due to rework.

**Frequency of Deployments**
indicates the overall output and activity of the project.

# Key Roles

**System Administrator**
Responsible for preparing and modeling the target environments and deploying the application into each of them

**Build Engineer**
Responsible for defining the release requirements and supporting the application deployment
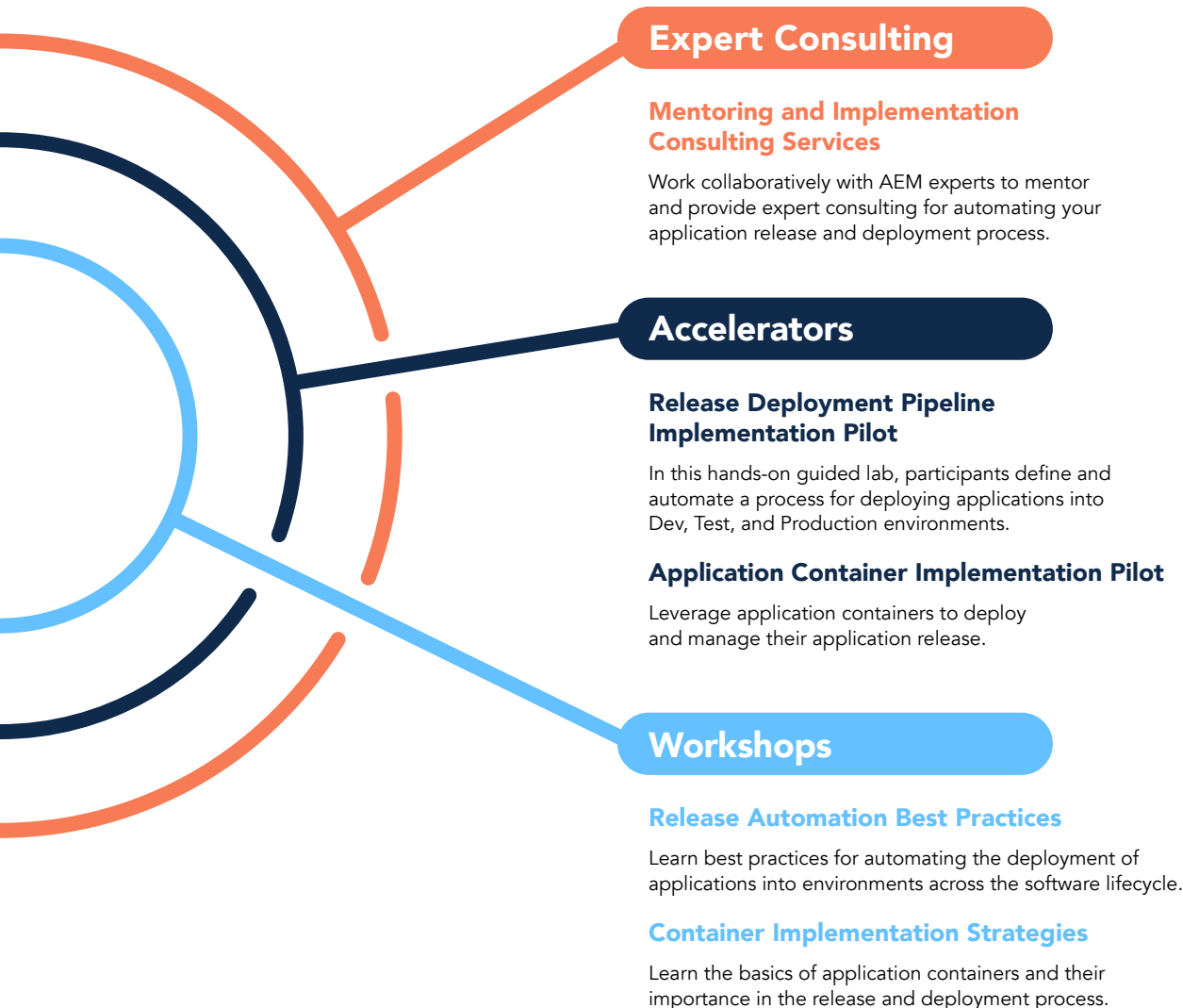
**Release Manager**
Responsible for defining and communicating the application release plans and reporting on the progress of the deployments

# Service Engagements

AEM's assessment-derived solutions are tailored to your needs. We work with you to define the right offerings required to build a solid foundation based on DevOps principles. The time and scope will vary based on your needs, but representative offerings based on previous client engagements are shown below.

## Expert Consulting

### Mentoring and Implementation Consulting Services

Work collaboratively with AEM experts to mentor and provide expert consulting for automating your application release and deployment process.

## Accelerators

### Release Deployment Pipeline Implementation Pilot

In this hands-on guided lab, participants define and automate a process for deploying applications into Dev, Test, and Production environments.

### Application Container Implementation Pilot

Leverage application containers to deploy and manage their application release.

## Workshops

### Release Automation Best Practices

Learn best practices for automating the deployment of applications into environments across the software lifecycle.

### Container Implementation Strategies

Learn the basics of application containers and their importance in the release and deployment process.

## Why AEM?

## Delivering faster...

By implementing application build & release pipelines into the Department of Education's non-production environments, AEM streamlined the process for 11 applications and reduced required time by one man-week per month.

## With less...

After AEM helped a major publishing company implement a complete release management pipeline, they were able to manage the release of over 100 microservices applications with a release management team of only three people.

**aem**

# TEST AUTOMATION

Test Automation is a practice where application tests are run automatically and continuously throughout the development process. Test-driven development and the use of unit tests are used to create and maintain acceptance tests that are reproducible and executed with each build.
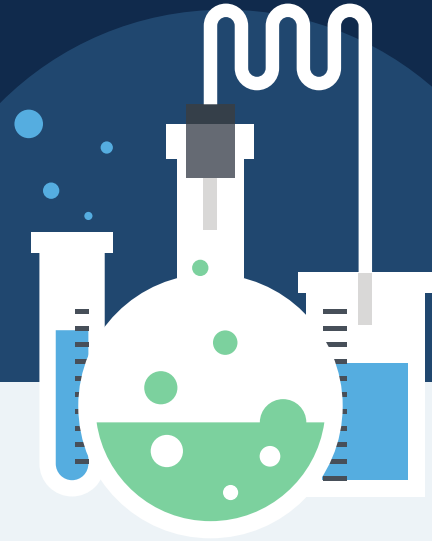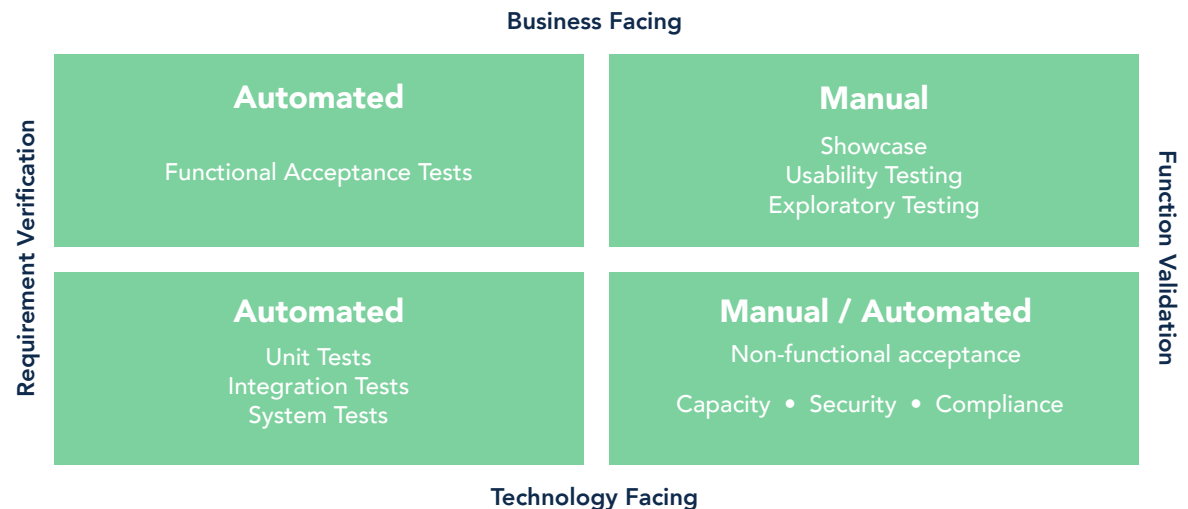
▸ Define and execute a consistent and repeatable process.

▸ Amplify feedback and improve team communication.

▸ Improve overall release deployment quality.

▸ Accelerate the delivery of application features to production.

## Models

Applications require several techniques and levels of testing to meet quality and user expectations. A typical project will implement a strategy similar to that of the Agile Testing Quadrants model shown here. ▸

The stages of Test Automation are shown in the diagram below. These will vary slightly based upon team structure and organization focus. ▾

**Business Facing**

| | |
|---|---|
| **Automated**<br><br>Functional Acceptance Tests | **Manual**<br><br>Showcase<br>Usability Testing<br>Exploratory Testing |
| **Automated**<br><br>Unit Tests<br>Integration Tests<br>System Tests | **Manual / Automated**<br><br>Non-functional acceptance<br><br>Capacity • Security • Compliance |

*Requirement Verification* (left axis)     *Function Validation* (right axis)

**Technology Facing**

**1** Commit Stage
• *Unit Test*
• *Code Analysis*

**1** Retrieve Application Artifacts

**2** Automated Acceptance Testing
• *Capacity*
• *External Dependencies*

**3** Automated Performance Testing

**8** Release to Production

aem

# Red Flag Areas

**Lack of Code Coverage Statistics**
indicates limited testing being done during development.

**Lack of Quality Test Data**
indicates poor test data management, which will produce unexpected application results in production.

**Long-running Test Suite**
conflicts with the need for fast builds and often results in tests being skipped during builds.

**Excessive Notifications**
result in team members ignoring important alerts, particularly for test results that produce false positives for defects.

# Key Metrics

**Test Code Coverage**

identifies the percentage of code in which functionality has been verified.

**Defects Reported Post Release**

indicates that requirements may not have a common understanding and/or automated testing is incomplete.

**Average Test Execution Suite Time**

impacts the available time for builds.

# Key Roles

**System Administrator**
Responsible for preparing and modeling the test environments and deploying the application into each of them

**Test Engineer**
Responsible for defining the release pipelines and maintaining the build infrastructure

**Developer**
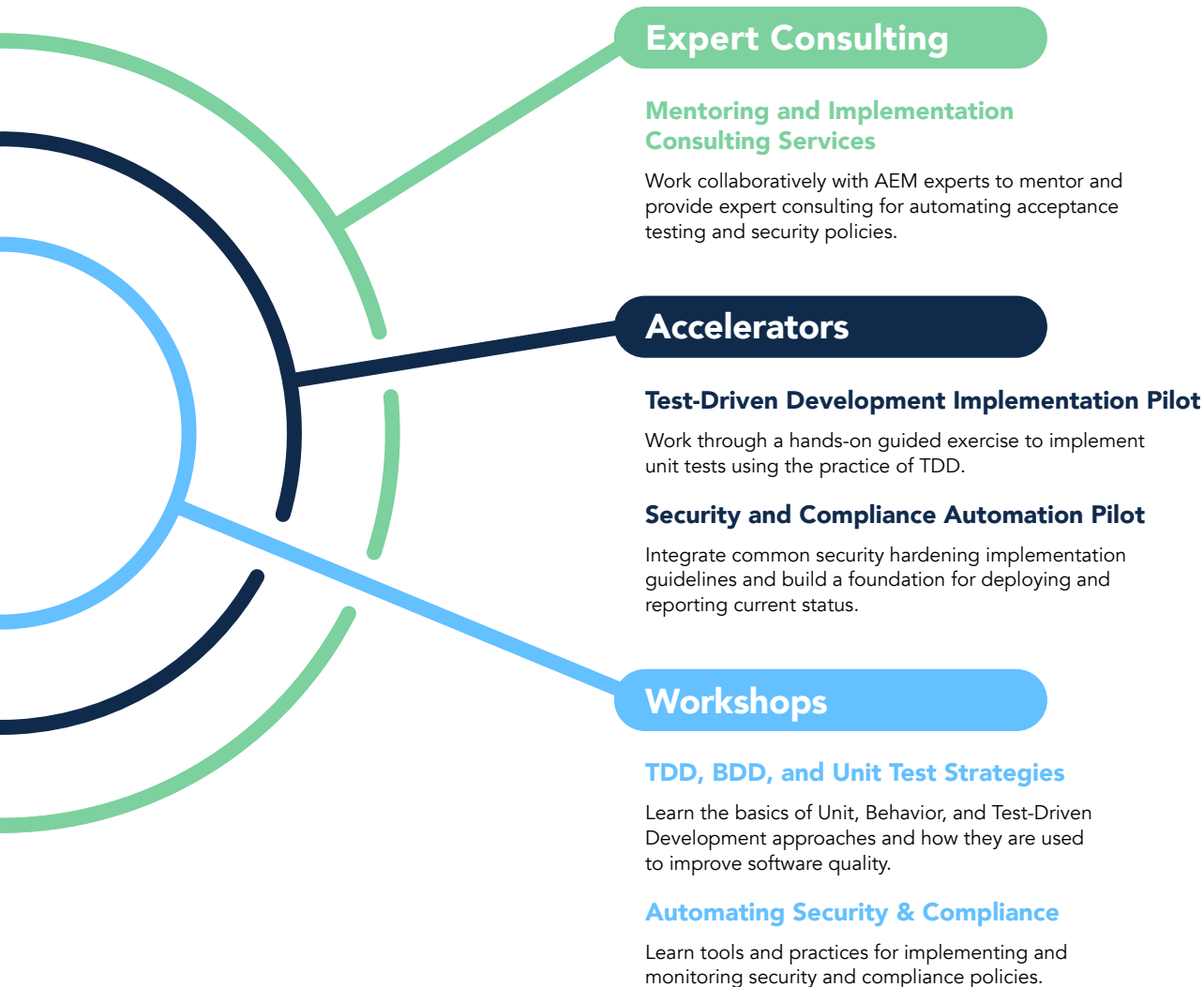Responsible for the application development and defect resolution

**Release Manager**
Responsible for defining and communicating the application release plans and reporting on the progress of the deployments

# Service Engagements

AEM's assessment-derived solutions are tailored to your needs. We work with you to define the right offerings required to build a solid foundation based on DevOps principles. The time and scope will vary based on your needs, but representative offerings based on previous client engagements are shown below.

## Expert Consulting

### Mentoring and Implementation Consulting Services

Work collaboratively with AEM experts to mentor and provide expert consulting for automating acceptance testing and security policies.

## Accelerators

### Test-Driven Development Implementation Pilot

Work through a hands-on guided exercise to implement unit tests using the practice of TDD.

### Security and Compliance Automation Pilot

Integrate common security hardening implementation guidelines and build a foundation for deploying and reporting current status.

## Workshops

### TDD, BDD, and Unit Test Strategies

Learn the basics of Unit, Behavior, and Test-Driven Development approaches and how they are used to improve software quality.

### Automating Security & Compliance

Learn tools and practices for implementing and monitoring security and compliance policies.

## Why AEM?

### Building quality in...

For the Department of Education's GENERATE project, AEM executes over 1,500 unit and integration tests with each TeamCity build to enhance the quality of each release.

### To provide consistent functionality...

AEM worked with ITHAKA to build a Selenium-based test suite for their JSTOR application. Using this automated test suite, they were able to validate and maintain functionality across multiple browser platforms with each application release.
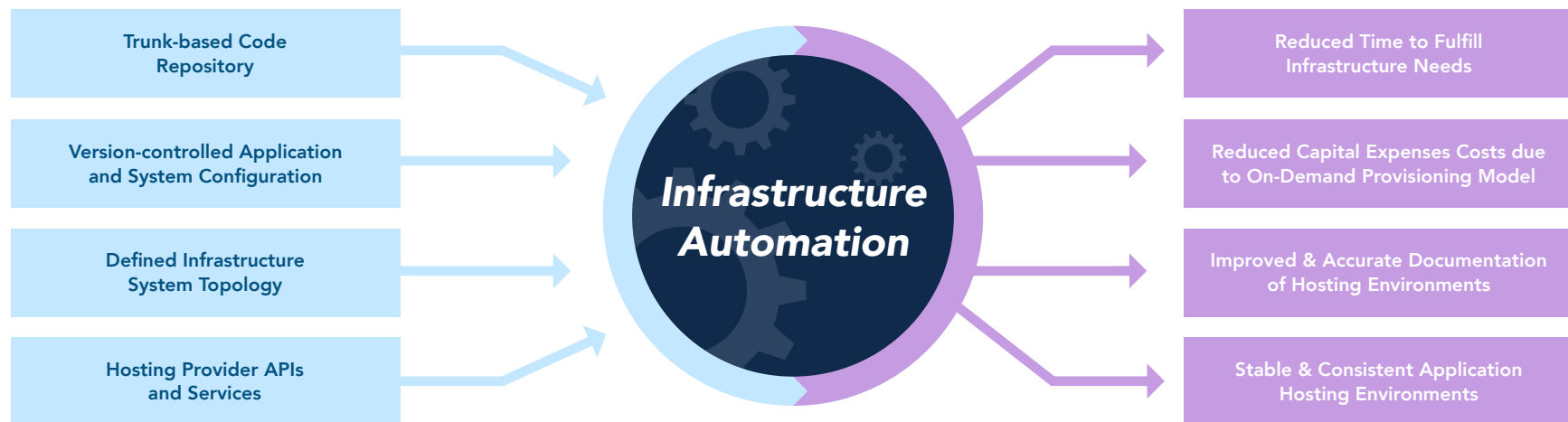
# INFRASTRUCTURE AUTOMATION

Infrastructure Automation is the process of creating and tearing down server and application infrastructure to support the development, testing, and production environments.

▶ Define and execute a consistent and repeatable process.

▶ Amplify feedback and improve team communication.

▶ Improve overall release deployment quality.

▶ Accelerate the delivery of application features to production.

## Models

In the diagram below, given the items on the left, implementing Infrastructure Automation will result in the items on the right. ▼

| | | |
|---|---|---|
| Trunk-based Code Repository | | Reduced Time to Fulfill Infrastructure Needs |
| Version-controlled Application and System Configuration | **Infrastructure Automation** | Reduced Capital Expenses Costs due to On-Demand Provisioning Model |
| Defined Infrastructure System Topology | | Improved & Accurate Documentation of Hosting Environments |
| Hosting Provider APIs and Services | | Stable & Consistent Application Hosting Environments |

aem

# Red Flag Areas

**Embedded System and/or Application Configurations**
lead to complicated code changes to handle environment changes and target releases.

**Infrequent Check-ins of Configurations**
lead to misconfigured environments and tribal knowledge of environment requirements.

**Inconsistent Environments**
increase deployment tasks and introduce instability into environments due to variations in topology and configurations.

**Excessive Notifications**
result in team members ignoring important alerts.

**Long Provisioning Request Fulfillments**
prevent rapid changeover and on-demand creation and scaling of supporting infrastructure elements.

**Infrequent Deployments**
system provisioning runbooks could atrophy over time and without regular exercising of procedures, environment definitions, which could become outdated.

# Key Metrics

**Time to Fulfill Environment Provisioning Request**

redirects resources from application and business feature development and impacts time required for each release.

**Frequency of Deployments**

indicates the overall output and activity of the project.

**Average Deployment Time**

impacts the available time for deployments.

**Percentage of Available Capacity**

indicates whether an environment is sized correctly and identifies potential stability issues that will occur when available resources are exhausted.

# Key Roles

**System Administrator**

Responsible for preparing and modeling the target environments and deploying the application into each of them

**Build Engineer**

Responsible for defining the release requirements and supporting the application deployment
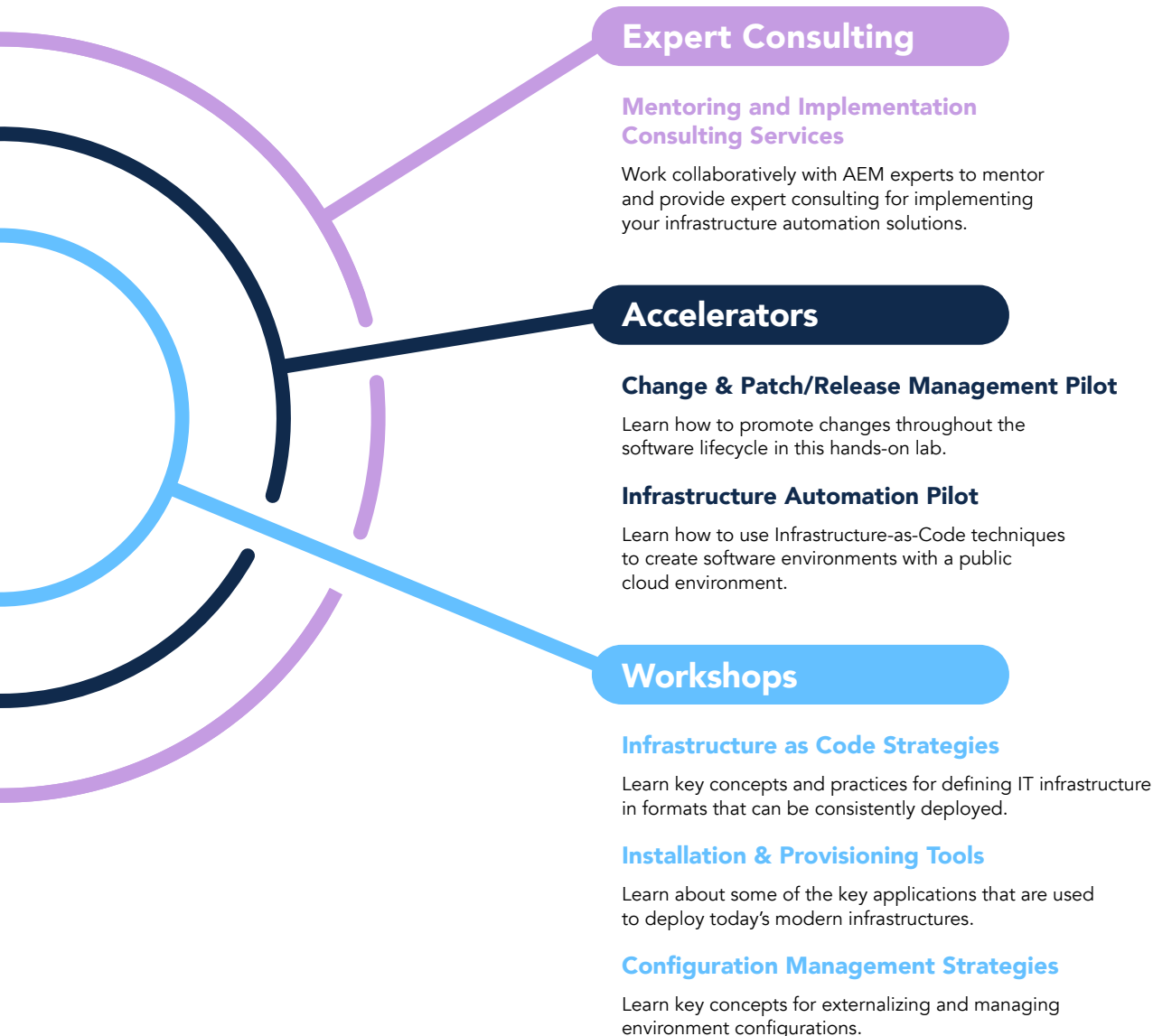
**Release Manager**

Responsible for defining and communicating the application release plans and reporting on the progress of the deployments

13

# Service Engagements

AEM's assessment-derived solutions are tailored to your needs. We work with you to define the right offerings required to build a solid foundation based on DevOps principles. The time and scope will vary based on your needs, but representative offerings based on previous client engagements are shown below.

## Expert Consulting

### Mentoring and Implementation Consulting Services

Work collaboratively with AEM experts to mentor and provide expert consulting for implementing your infrastructure automation solutions.

## Accelerators

### Change & Patch/Release Management Pilot

Learn how to promote changes throughout the software lifecycle in this hands-on lab.

### Infrastructure Automation Pilot

Learn how to use Infrastructure-as-Code techniques to create software environments with a public cloud environment.

## Workshops

### Infrastructure as Code Strategies

Learn key concepts and practices for defining IT infrastructure in formats that can be consistently deployed.

### Installation & Provisioning Tools

Learn about some of the key applications that are used to deploy today's modern infrastructures.

### Configuration Management Strategies

Learn key concepts for externalizing and managing environment configurations.

# Why AEM?

## Fortune 500 Financial Services Company

### Challenge

Modernize an underfunded IT infrastructure to improve business agility while adhering to industry regulations.

### Solution

Automate Day 1 and Day 2 operations to accelerate the ability to respond to business needs.

### Results

▶ 98% reduction in time to fulfill IT provisioning requests

▶ Created a single platform to provide Infrastructure, Platform, and Database-as-a-Service capabilities