|  |
| --- |
|  |
| TDD System Team Recommendations |
|  |

|  |
| --- |
|  |

**Contents**

[**SUMMARY** 2](#_Toc8044759)

[**SAFe Definition** 2](#_Toc8044760)

[**DevSecOps** 3](#_Toc8044761)

[**CI/CD** 3](#_Toc8044762)

[**BACKGROUND** 4](#_Toc8044763)

[**KEY FINDINGS** 5](#_Toc8044764)

[**RECOMMENDATIONS** 7](#_Toc8044765)

[**RESOURCES & FUTURE STATE** 10](#_Toc8044766)

# **SUMMARY**

This report documents the results of the [System Teams Roles and Responsibilities Matrix](https://collab.purplehub.fedex.com/Communities/DevSecOpsEnablement/Documents/System%20Teams%20Documentation/SystemTeamRolesAndResponsibilitiesMatrix.xlsx) survey. The Roles and Responsibilities were identified from SAFe (Scaled Agile Framework) best practices as well as from input from System Teams.

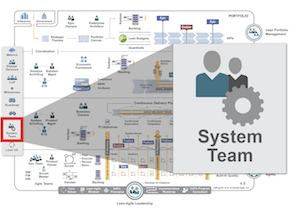
The intent of the System Team Matrix is to establish a baseline of the skills and tasks that System Teams are performing or should be capable of performing. Within this report, individual System Teams will receive guidance on potential opportunities to expand their services as well as on areas in which the System Team is already proficient.

## **SAFe Definition**

According to the SAFe [website](https://www.scaledagileframework.com/system-team/), a System Team is defined as:

“…a specialized [Agile Team](https://www.scaledagileframework.com/agile-teams/) that assists in building and supporting the Agile development environment, typically including development and maintenance of the toolchain that supports the [Continuous Delivery Pipeline.](https://www.scaledagileframework.com/continuous-delivery-pipeline/)  The System Team may also support the integration of assets from Agile teams, perform end-to-end [Solution](https://www.scaledagileframework.com/solution/) testing where necessary, and assists with deployment and [Release on Demand](https://www.scaledagileframework.com/release-on-demand/).

In SAFe, Agile teams are not stand-alone units. Instead, they are part of the [Agile Release Train (ART)](https://www.scaledagileframework.com/agile-release-train/), responsible collectively for delivering larger system and solution value. During the transition to Agile, additional infrastructure work is typically required to integrate solution assets more frequently. To accomplish this, one or more specialized System Teams are often formed. They help build the environments and assist with system and solution integration. They also help demo the solution as it evolves.”

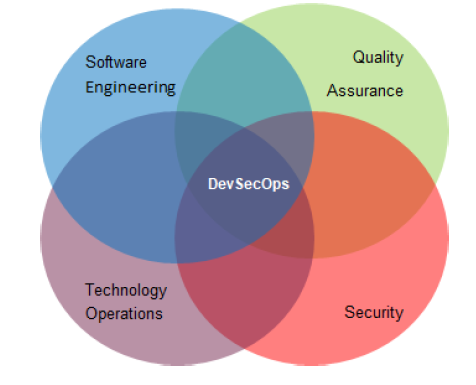
[](https://www.scaledagileframework.com/)

## **DevSecOps**

System Teams should promote the cultural foundations of [DevSecOps](https://collab.purplehub.fedex.com/Communities/DevOps%20Community%20of%20Practice/SitePages/teamhome.aspx). From the [RenewalTechnicalTenets](https://collab.purplehub.fedex.com/Communities/Renewal/Documents/Renewal%20Technical%20Tenets.pdf) documentation:

“DevSecOps (“Development”, “Security”, “Operations”) is a software development culture that emphasizes communication and collaboration between product owners, development, and information security and operations professionals. DevSecOps establishes a culture and environment where building, testing, scanning, and releasing software can occur rapidly, frequently, securely and reliably.

A DevSecOps culture seeks to automate the process of software integration, testing, deployment and infrastructure changes. DevSecOps practices also promote operational monitoring and feedback channels in order to proactively and continuously improve software quality.”



## **CI/CD**

Another [Renewal Technical Tenet](https://collab.purplehub.fedex.com/Communities/Renewal/Documents/Renewal%20Technical%20Tenets.pdf) which System Teams should advocate for is the use of CI/CD (Continuous Integration / Continuous Delivery) practices. From the [RenewalTechnicalTenets](https://collab.purplehub.fedex.com/Communities/Renewal/Documents/Renewal%20Technical%20Tenets.pdf) documentation:

“Continuous Integration (CI) is a DevSecOps and agile software development practice where developers regularly merge their code changes into a central repository, after which automated code scans, builds, and tests are run. Continuous Delivery (CD) is a software development and testing approach, inclusive of Continuous Integration, in which teams can deliver software in short cycles and ensure that it can be reliably released at any time.

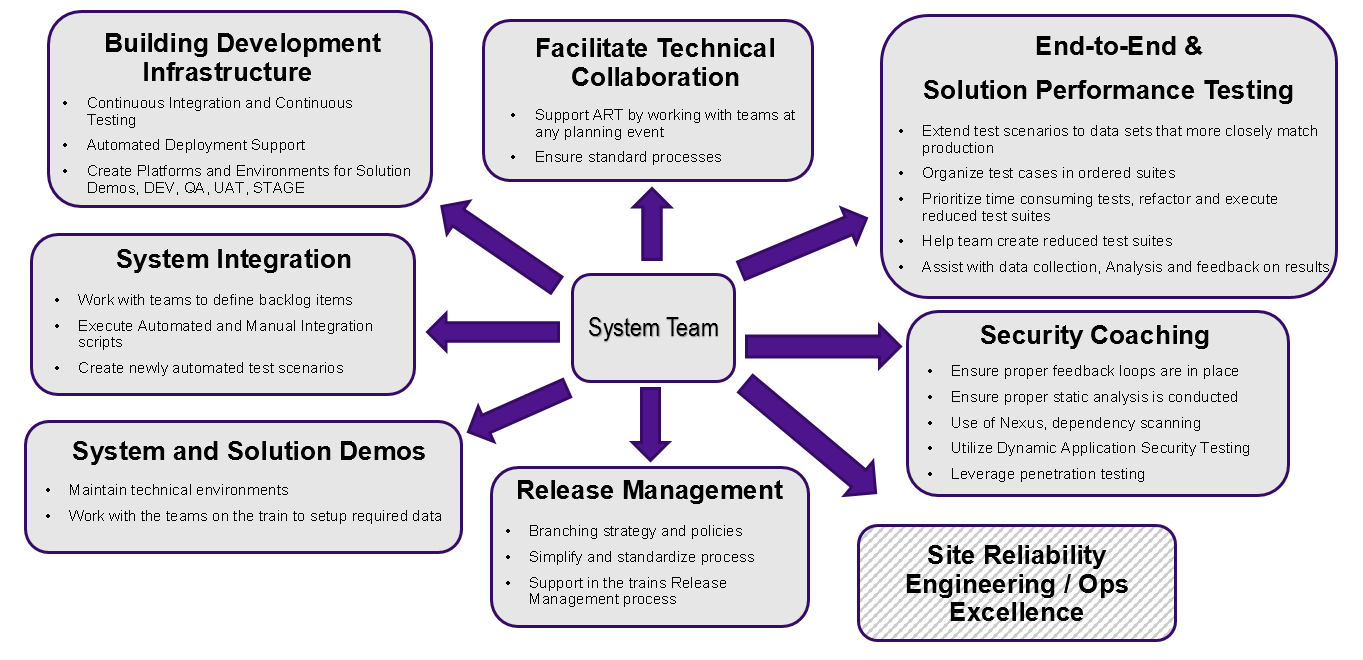
CI/CD increases the velocity of the development life cycle. It improves code quality and provides transparency. It integrates automated unit, integration, and system tests. It also accelerates business value delivery through reducing development time fragmentation.”

# **BACKGROUND**

Within the Matrix, each category contains multiple assessment points to help attain overall engagement in said category. Each System Team met with DevSecOps Enablement (DSOE) to determine their proficiency in each assessment item. The findings and recommendations resulting from the surveys are contained within this report (each System Team received their completed survey after the meeting was conducted).

The System Team Matrix is divided into seven major categories, each with a set of tasks that correspond to that subject. The categories are:

* System Integration
* Building Development Infrastructure
* Facilitate Technical Collaboration
* End-To-End and Solution Performance Testing
* System and Solution Demos
* Release Management
* Security Coaching



# **KEY FINDINGS**

The charts below detail the survey results broken down by each major Matrix category of the average FedEx Ground Services System Team versus the TDD System Team.

They also highlight your team’s relative strengths and weaknesses compared to peer groups.  Knowing this identifies areas of potential leadership as well as areas of opportunity.  Leveraging collaborative forums such as the System Team CoP can be useful to both teach and learn.  
  
Having awareness of your team’s capabilities in a given category can assist when considering future training opportunities as well as identifying skills to target when adding additional human capitol.

**Areas of high engagement**

* End-to-End and Solution Performance Testing is the category for which your team most significantly outpaces with your peers.
* System Integration has full engagement.

**Areas of opportunity**

* Release Management is the category which most significantly lags behind the average engagement scores.
* Security Coaching currently has no engagement.
* There are opportunities to increase engagement in the areas of Building Development Infrastructure, Facilitate Technical Collaboration, and System and Solution Demos.

# **RECOMMENDATIONS**

**Potential Methods to Improve *Building Development Infrastructure* Engagement:**

* Gain proficiency in creating Jenkins multibranch pipelines which span from code commit through production deployment, with quality gates included.
* Work with Scrum Teams to ensure [CI/CD minimums](https://collab.purplehub.fedex.com/Communities/DevSecOpsEnablement/Documents/CICD%20minimums.xlsx) are being met.
* Meet with Scrum Teams to understand all application system requirements.
  + Document application requirements checklist.
  + Submit and track requests to external groups to perform pertinent actions for overall setup of new environments.
  + Validate newly-provisioned environments to ensure satisfaction of checklist items.
  + Automate any aspect of this process when feasible.
* Maintain awareness of the [Infrastructure Life Cycle Management patched versions](https://collab.purplehub.fedex.com/Projects/WebLogic%20Server%20and%20Java%20Quarterly%20Updates/SitePages/teamhome.aspx) and validate that all application processes are compliant with the mandated versions. Perform bounces when necessary to instantiate the patches.
* Perform administrative tasks on all middleware environments.
* Gain and maintain familiarity with current monitoring and instrumentation standards. Communicate recommendations to Scrum Teams and assist with implementation of monitoring tools. Ensure that monitoring is sufficient and provides meaningful and accurate feedback.
* Promote the use of automation for any activity that will be performed more than once.
* Ensure that legacy environments and servers are decommissioned to promote Renewal and to avoid the accumulation of technical debt.

**Potential Methods to Improve *Facilitate Technical Collaboration* Engagement:**

* Work with Scrum Teams to ensure [CI/CD minimums](https://collab.purplehub.fedex.com/Communities/DevSecOpsEnablement/Documents/CICD%20minimums.xlsx) are being met.
* Deliver a consistent message to all Scrum Teams in order to promote standardized processes across the Release Train.

**Potential Methods to Improve *System and Solution Demos* Engagement:**

* Assess whether Solution and Security Demos are of value to the Train.
  + If so, assist Scrum Teams with Demo Environment setup and maintenance as well as Demo Data population.

**Potential Methods to Improve *Release Management* Engagement:**

* Support the Release Train’s Release Management lifecycle, from code management through Production installation.
  + Understand each Scrum Team’s code management strategy. When possible, advise the teams to follow a standardized process.
  + Coordinate the release schedule for all Scrum Teams.
  + Ensure all release artifacts are present and accurate in the Change Requests and follow the Change Requests’ progression to achieve compliance approvals.
  + Promote CI/CD principles so that applications are built agnostically and can be installed into any level either automatically or by the System Team.
* Compile pertinent data for each release that can be used to facilitate meaningful metrics. Examples include how often is a build put into Production, how many bugs get introduced into Production, MTTR (Mean Time To Recovery), length of time to go from backlog to MVP (Minimum Viable Product).
* Staff the System Team with a dedicated Release Management Specialist.

**Potential Methods to Improve *Security Coaching* Engagement:**

* Promote the utilization of security-based tools within Jenkins CI Pipelines, such as SonarQube for static code analysis / code coverage and NexusIQ for dependency scanning. Scan the code using Fortify to prevent security vulnerabilities.
* Ensure Scrum Teams are observing security best practices as dictated by the Enterprise.
* Staff the System Team with a dedicated Security Specialist.
* Work with Infosec to explore Penetration Testing possibilities (could be performed by Security Specialist if one exists).
  + Pluralsight Course: [**Penetration Testing: The Big Picture**](https://www.pluralsight.com/courses/penetration-testing-big-picture)
  + Utilize WebInspect for penetration testing.
    - Done by InfoSec for Externally Facing apps.
    - Currently exploring expanding the usage of this tool.

# **RESOURCES & FUTURE STATE**

As System Teams mature and their roles and responsibilities become standardized, the metrics will stress proficiency rather than engagement.

Below are some resources which can be utilized to assist System Teams in expanding on their areas of proficiency:

* [How do I CI Cookbook](https://apptx-cookbook.app.wtcbo1.paas.fedex.com/recipes/how-do-i-ci/)
* [Stack Overflow](https://stackoverflow.web.fedex.com/)
* Pluralsight Course [“DevOps: The Big Picture”](https://app.pluralsight.com/library/courses/devops-big-picture/table-of-contents)
* Pluralsight Course [“Implementing DevOps in the Real World”](https://app.pluralsight.com/library/courses/implementing-devops-real-world/table-of-contents)
* Pluralsight Course [“Continuous Integration and Continuous Delivery: The Big Picture”](https://app.pluralsight.com/library/courses/continuous-integration-delivery-big-picture/table-of-contents)
* Pluralsight Course: [“Penetration Testing: The Big Picture](https://www.pluralsight.com/courses/penetration-testing-big-picture)”