Introduction to DevOps

Sonika Rathi

Assistant Professor
BITS Pilani

Review CS#2

DevOps : Process Dimension

- DevOps Misconception

- DevOps Anti-Patterns

- Three Dimensions of DevOps
    - Process
    - People
    - Tools

- DevOps : Process Dimension
    - DevOps and Agile
    - Team Structure
    - Agile Methodology
        - TDD
        - BDD
        - FDD

Agenda

DevOps : People & Tools Dimension

- Transformation to Enterprise DevOps culture
- DevOps - People
- DevOps – Tools
- Cloud as a catalyst for DevOps
- Transition in IT
  - Building competencies, Full Stack Developers
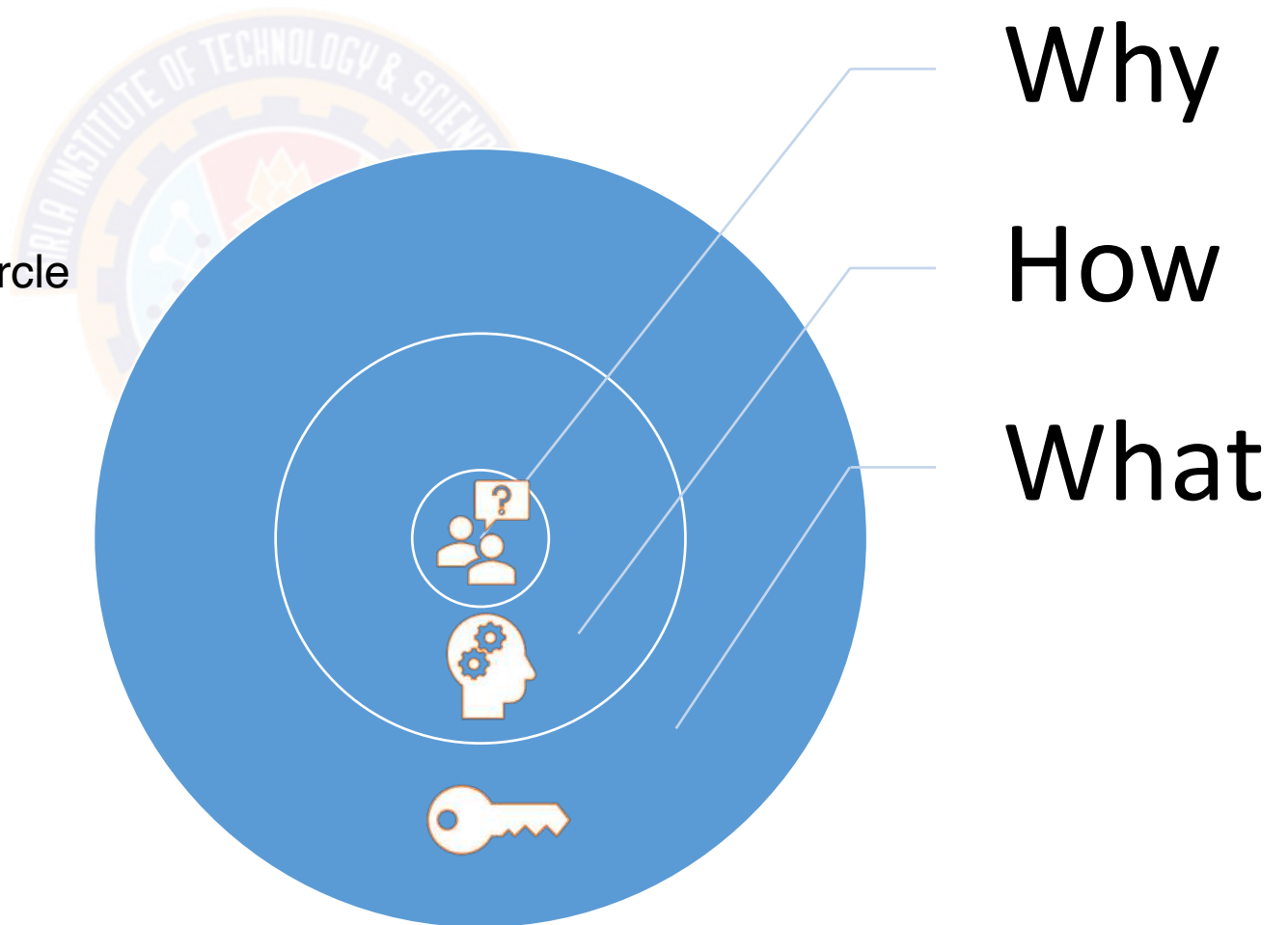 - Self-organized teams, Intrinsic Motivation
- Technologies in DevOps

Transformation to Enterprise DevOps culture

Anti-Patterns

- Constraints of Large Enterprise :: DevOps Transformation
  - Organization Structure
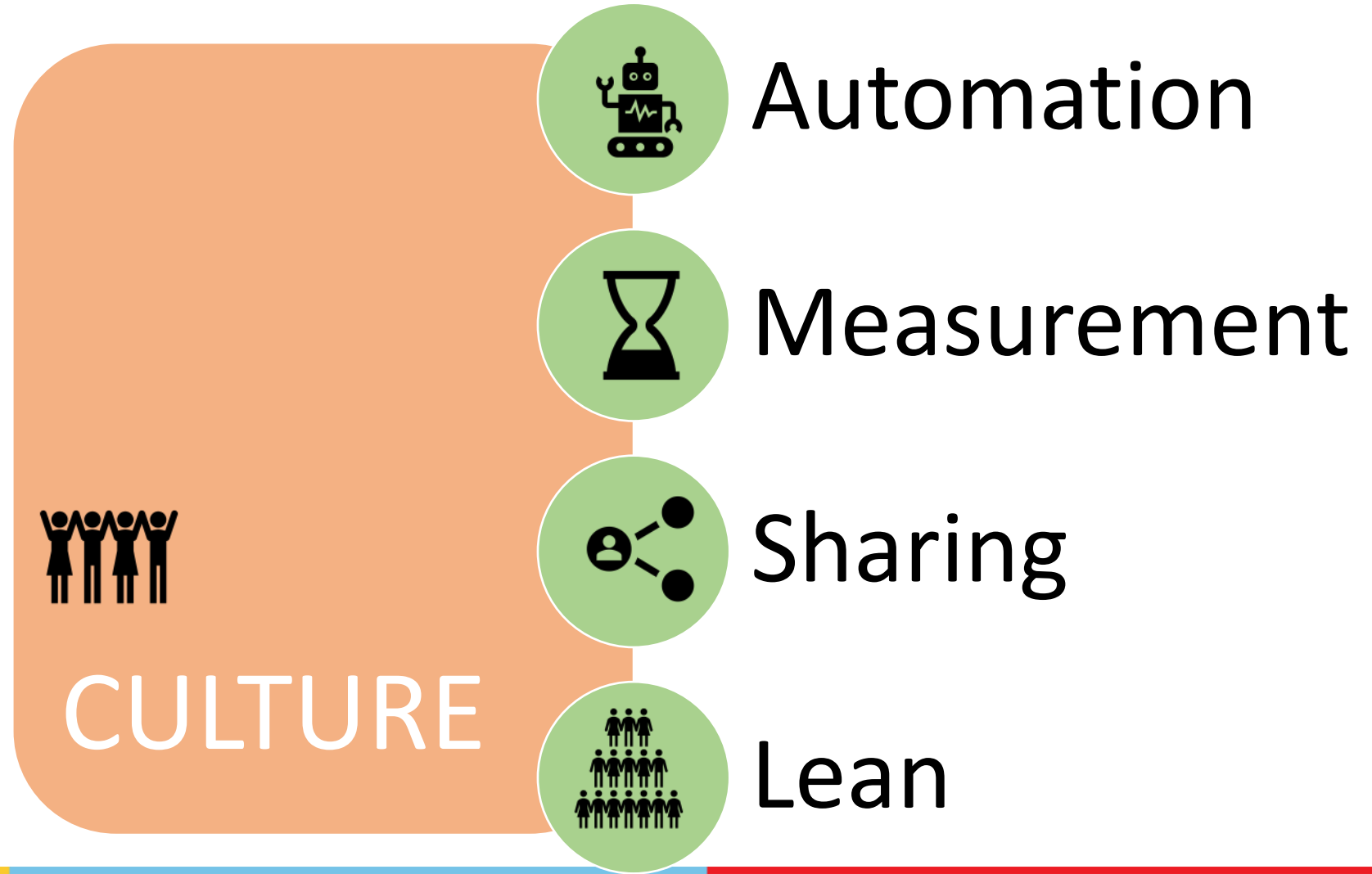- Legacy Technology Stack
- Organization Culture

Start with the "Why" – The Golden Circle

Why

How

What

Transformation to Enterprise DevOps culture

DevOps Values

More than anything else, DevOps is a culture movement based on human and technical interaction to improve relationships and results

CULTURE

Automation

Measurement

Sharing

Lean

Transformation to Enterprise DevOps culture

## Initial Planning for Enterprise Readiness

- Participants should absolutely include all the towers that make up the solution delivery

- Design Thinking : It is a great method; it leverages the expertise of all stakeholders, enables them to come to a common understanding

- High Level Output

Participant & Inputs → Design Thinking → High Level Outputs

Transformation to Enterprise DevOps culture

Establish a DevOps Centre of Excellence

- At the right organizational level and enterprise authority
- Create a Face: It has to be led by an enterprise leader who has the support and buy-in from all the towers
- Active Participant from All Towers of Delivery & Participants must be chosen wisely
- These phases help concrete the vision and strategies of organization and help in setup the best practices to support cultural movement

Authority → Create a Face → Involve All
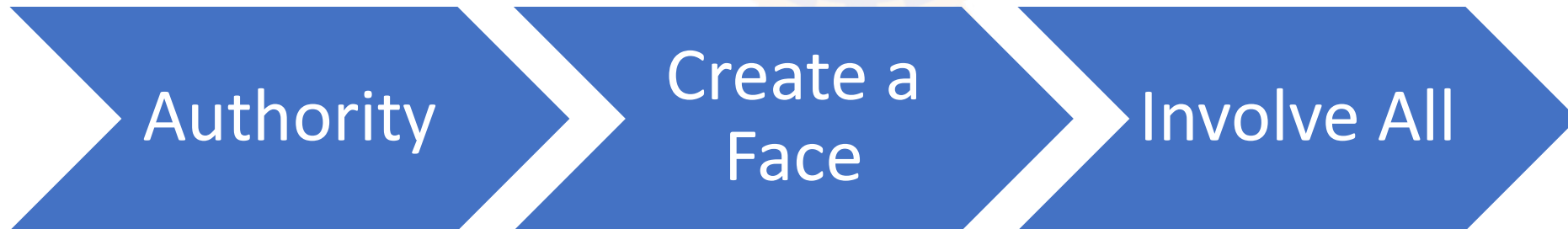
Transformation to Enterprise DevOps culture

Establish Program Governance

- Agile and DevOps, practitioners' roles and responsibilities will change
- Need awareness, enablement and empowerment to succeed
- KPIs must shift from individual metrics to holistic customer business outcomes

Create Communication Plan	Enablement Program	Establish Program KPI's

Transformation to Enterprise DevOps culture

Establish Project In-take Process

- DevOps SME's conduct in-take workshops for Scrum Teams
- Automation scripts, Infrastructure assets
- Test Automation, Branching and Merging & Lessons Learned
- Fit for purpose tool selection and Application Criticality

Reusable Assets → Best Practices → Right Sizing
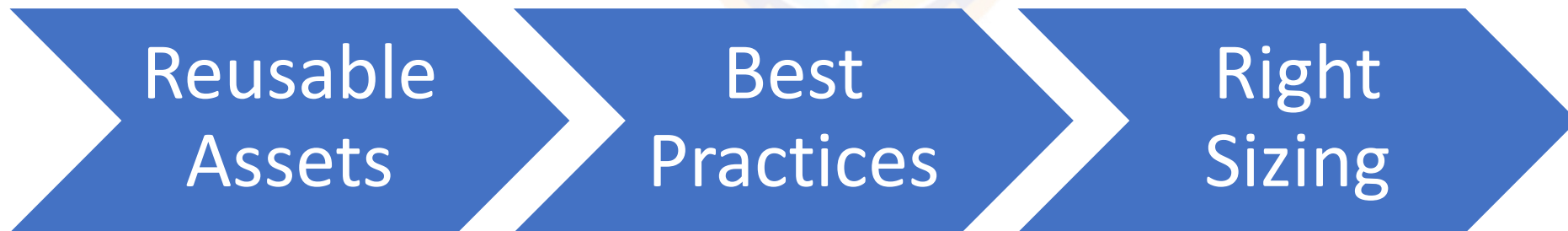
Transformation to Enterprise DevOps culture

Identify and Initiate Pilots

- Value stream-mapping exercise

- Level of detail necessary:

- To identify end to end as-is process, tooling, manual and automated processes

- And skills and people
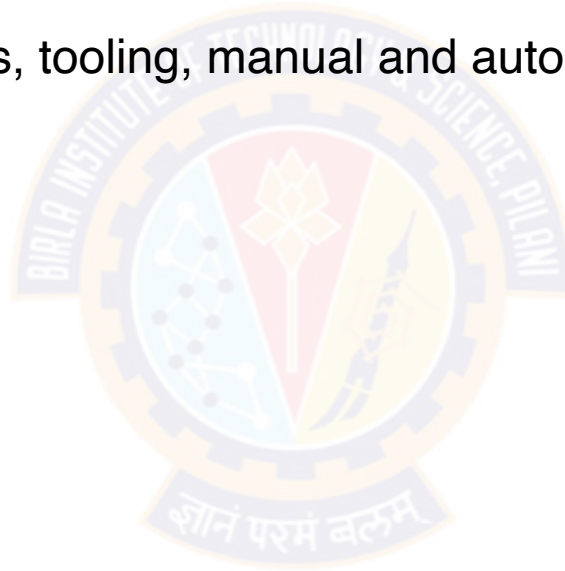
Transformation to Enterprise DevOps culture

Scale Out DevOps Program

- Onboard Parallel Release Trains

- Apply Intake Process

- Support, Monitor and Manage

DevOps - People

People

- DevOps is a cultural movement; it's all about people

- Building a DevOps culture, therefore, is at the core of DevOps adoption

An organization may adopt the most efficient processes or automated tools possible, but they're useless without the people who eventually must execute those processes and use those tools

- DevOps culture is characterized by a high degree of collaboration across roles, focus on business instead of departmental objectives, trust, and high value placed on learning through experimentation

- Building a culture isn't like adopting a process or a tool

- It requires social engineering of teams of people, each with unique predispositions, experiences, and biases

- This diversity can make culture-building challenging and difficult

DevOps – People

Managing People, Not Resources

"Managing teams would be easy if it wasn't for the people you have to deal with"

- Calling people as resources 🚫

- A resource is something you can manage without much tailoring

- With people, that is never true

- For an Example: We have probably all been planning for projects and creating a plan that includes a certain amount of "anonymous full-time equivalents (FTEs)." Then a bit later, we start putting names to it and realize that we need more or less effort based on the actual people we have available for the project

- One Java developer is just not the same as another; and honestly, we would never consider ourselves to be just a resource whose job someone else could do with the same efficiency and effectiveness

Vs

DevOps – People

Don't change Organization; Change your self

- You will not be able to get the organization to change

- But what you can do is change your part of the organization

- Manage your teams the way that you would like to be treated

- As you climb higher in the hierarchy, your area of influence will increase and, with that, a larger and larger part of the organization will work the way you would like it to

- Ex. "Jack Welch" he was the CEO of General Electric till 2001, his cabin was of all transparent glass; And this cabin did not have any door. So, it was an open-door cabin. And the label on the door was "PLEASE DISTRUB ME".
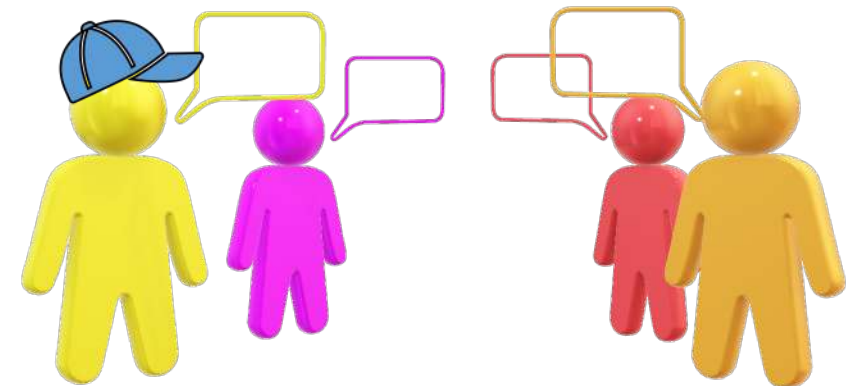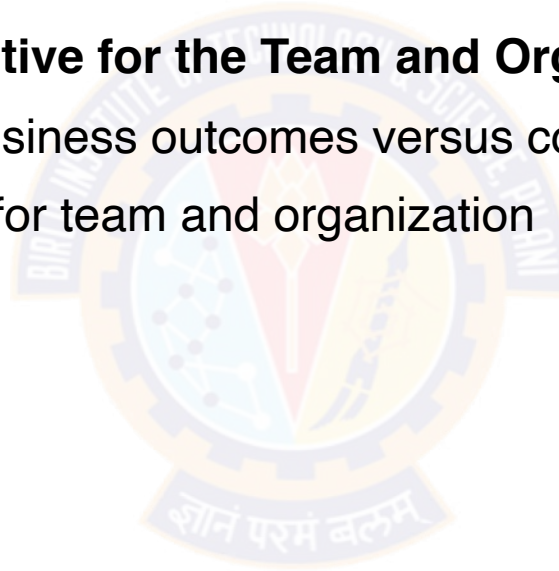
DevOps – People

Identifying Business Objective

- Getting everyone headed in the same direction
- And working toward the same goal

"Identify Common Business Objective for the Team and Organization"

- Incent the entire team based on business outcomes versus conflicting team incentives
- Easy to measure progress of goal for team and organization

DevOps isn't the goal. It helps you reach your goals

Effective Management of People

One-On-Ones

- Regular one-on-one meetings with the people who report directly to you

- Don't be Anonymous

- Having an open-door policy is NOT the same as setting up one-on-ones

- Let feel people are important to you

- You are making time for them

- Best Practices : have weekly or bi weekly 30 minutes one-on –one, learn more about person, let them raise first and provide your updates
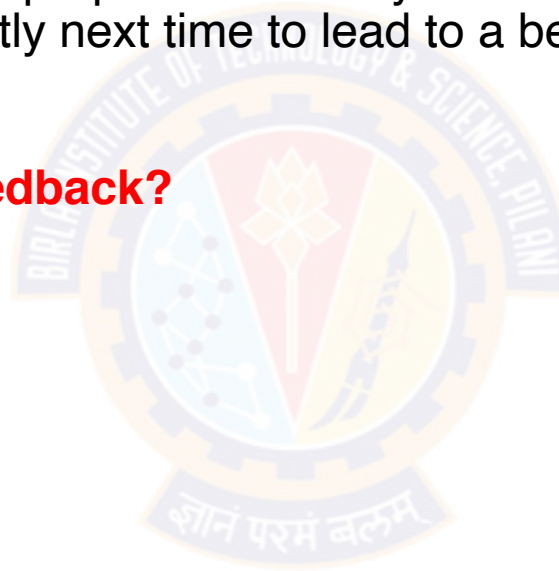
- Benefit to Manager?

- Benefit to Employee?
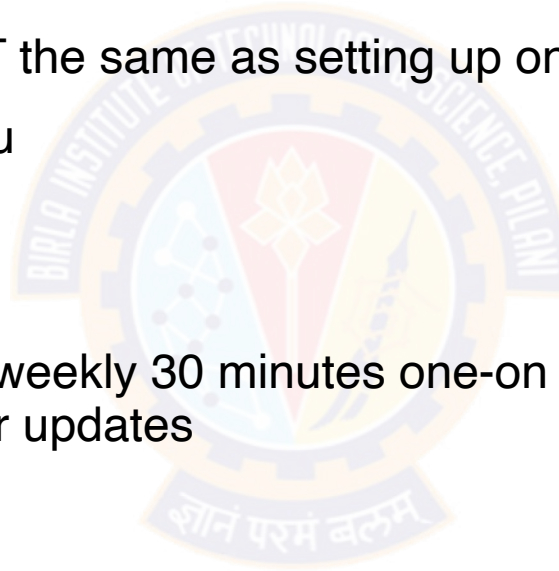
Effective Management of People

Feedback

- Everyone would like to get more feedback from his or her boss

- Dan Pink's mastery, autonomy, and purpose : "When you do x, y happens, which is not optimal. Can you find a way to do it differently next time to lead to a better outcome? "

What You hear in your Feedback?

- Focus on feedback?

Effective Management of People

Delegation

- Feel on delegation of JOB to others?

    Managerial Economics 101

Effective Management of People

Creating a Blameless Culture

- Who is being blamed?

- You will have to cover it without delegating the blame to the person in your team who is responsible

- The team will appreciate this

- The more you share with the rest of the organization the positive impact a member of your team has made, the better you will look too

- These two practices empower the people on your team to do the best job they can for you

- Whenever you have failures or problems, your root-cause analysis should not focus on who did what but on how we need to change the system so that the next person is able to avoid making the mistake again

Effective Management of People

Measuring Your Organizational Culture

- There are a few different ways to measure culture

- The Westrum survey measures
  - On my team, information is actively sought
  - On my team, failures are learning opportunities, and messengers of them are not punished
  - On my team, responsibilities are shared
  - On my team, cross-functional collaboration is encouraged and rewarded
  - On my team, failure causes inquiry
  - On my team, new ideas are welcomed

- There are few more suggestions:
  - I would recommend the team to my friends as a good place to work
  - I have the tools and resources to do my role well
  - I rarely think about leaving this team or my company
  - My role makes good use of my skills and abilities

Remember that advice stated before: you can only control your part of the organization

DevOps – Tools

What is DevOps Tools:

- Tools can be used to improve and maintain various aspects of culture

- Software development tools help with the process of programming, documenting, testing, and fixing bugs in applications and services

- Not restricted to specific roles, these tools are important to anyone who works on software in some capacity
  - Local development environment,
  - Version control,
  - Artifact Management,
  - Automation and Monitoring
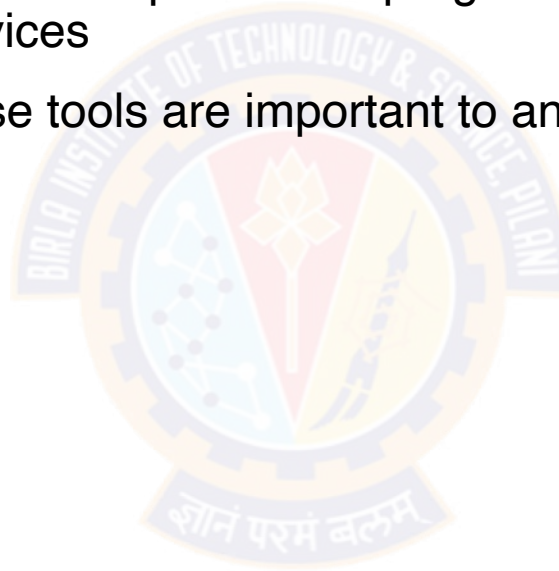
DevOps Tools

Local Development Environment

- A consistent local development environment is critical to quickly get employees started contributing to your product

- Don't Consider this as limitation:
    - This is not to say that individuals should be locked into a single standard editor with no flexibility or customization, but rather it means ensuring that they have the tools needed to get their jobs done effectively

- Minimal requirements may vary in your environment depending on individual preferences:
    - Multiple Displays
    - High-Resolution Displays
    - Specific Keyboards, Mice, and other Input Devices

DevOps Tools

Version Control

- Having the ability to commit, compare, merge, and restore past revisions of objects to the repository allows for richer cooperation and collaboration within and between teams

- Version control enables teams to deal with conflicts that result from having multiple people working on the same file or project at the same time, and provides a safe way to make changes and roll them back if necessary

- When choosing the appropriate version control system (VCS) for your environment, look for one that encourages the sort of collaboration in your organization that you want to see

  - Opening and forking repositories;

  - Contributing back to repositories;

  - Contributions to your own repositories;

  - Defining processes for contributing; and

  - Sharing commit rights

DevOps Tools

Artifact Management

- An artifact is the output of any step in the software development process

- When choosing between a simple repository and more complex feature-full repository, understand the cost of supporting additional services as well as inherent security concerns

- An artifact repository should be:
  - Secure;
  - Trusted;
  - Stable;
  - Accessible; and
  - Versioned

- You can store a versioned common library as an artifact separate from your software version control, allowing all teams to use the exact same shared library
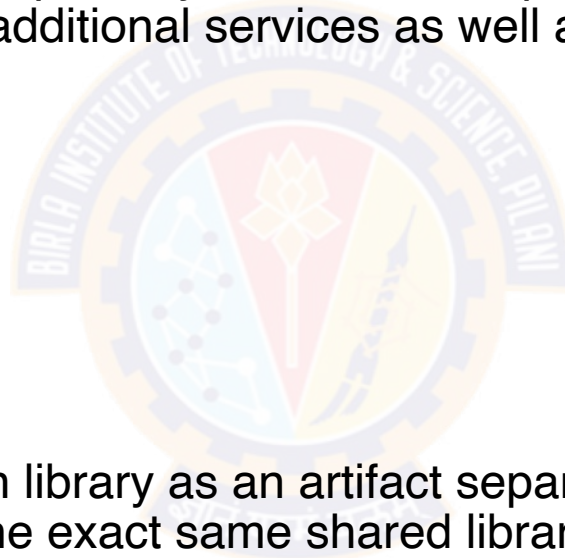
DevOps Tools

Other Automation Tools

- Automation tools reduce labor, energy, and/or materials used with a goal of improving quality, precision, and accuracy of outcomes

- Server Installation
  - Server installation is the automation of configuring and setting up individual servers

- Infrastructure Automation
  - Configuration Management
  - Capacity Management

- System Provisioning

- Test and Build Automation
  - On-demand automation
  - Scheduled automation
  - Triggered automation
  - Smoke testing
  - Regression testing
  - Usability testing

DevOps Tools

Monitoring

- Metrics
  - Metrics are the collection of qualitative and quantitative measurements
  - Generally they are compared against some benchmark or established standard, tracked for analytics, or tracked for historical purposes

- Logging
  - Logging is the generation, filtering, recording, and analysis of events that occur on a system due to operating system or software messages
  - When tracking down the source of a software issue, one of the first things that engineers often do is to check the logs for any relevant error messages
  - A single system can generate hundreds or even thousands of lines of logs a day

DevOps Tools

Monitoring Contd..

- Alerting
    - Monitoring and alerting are important not only from a performance perspective, but also from a preventative one, in that they help you find out about potential issues before they become actual issues for your customers
    - Impact
    - Urgency
    - Interested party
    - Resources
    - Cost

- Events
    - Event management is the element of monitoring that acts on existing knowledge around impacts to systems and services
    - This generally reflects the need for real-time information about the status of all of the different components of infrastructure
    - A system is configured to monitor a specific metric or log based on a defined event and to signal or alert if a threshold is crossed or an alert condition has been met
    - Many alerting and monitoring systems have built-in ways to automatically respond to a given event
    - Nagios, Zabbix, etc.
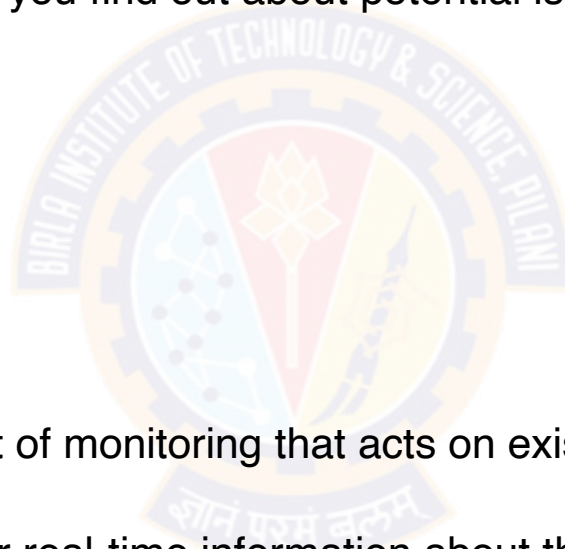
Evolution of the Ecosystem

DevOps Ecosystem

- There is a trend to simplify and remove the repetitive tasks that can be subject to human error

- As automation is added to the different parts of the environment, new patterns are discovered

- Continuous delivery and continuous deployment have freed humans up to focus on what matters

- Automated shortened feedback cycles through build automation, tests automation give us additional confidence and insight into our systems

- The ecosystem will continue to evolve as application development adopts increased operability

- These trends bring into focus the tools that stress the "we" over "me," building understanding across teams and encouraging time spent on valuable outcomes

DevOps Tools Ecosystem
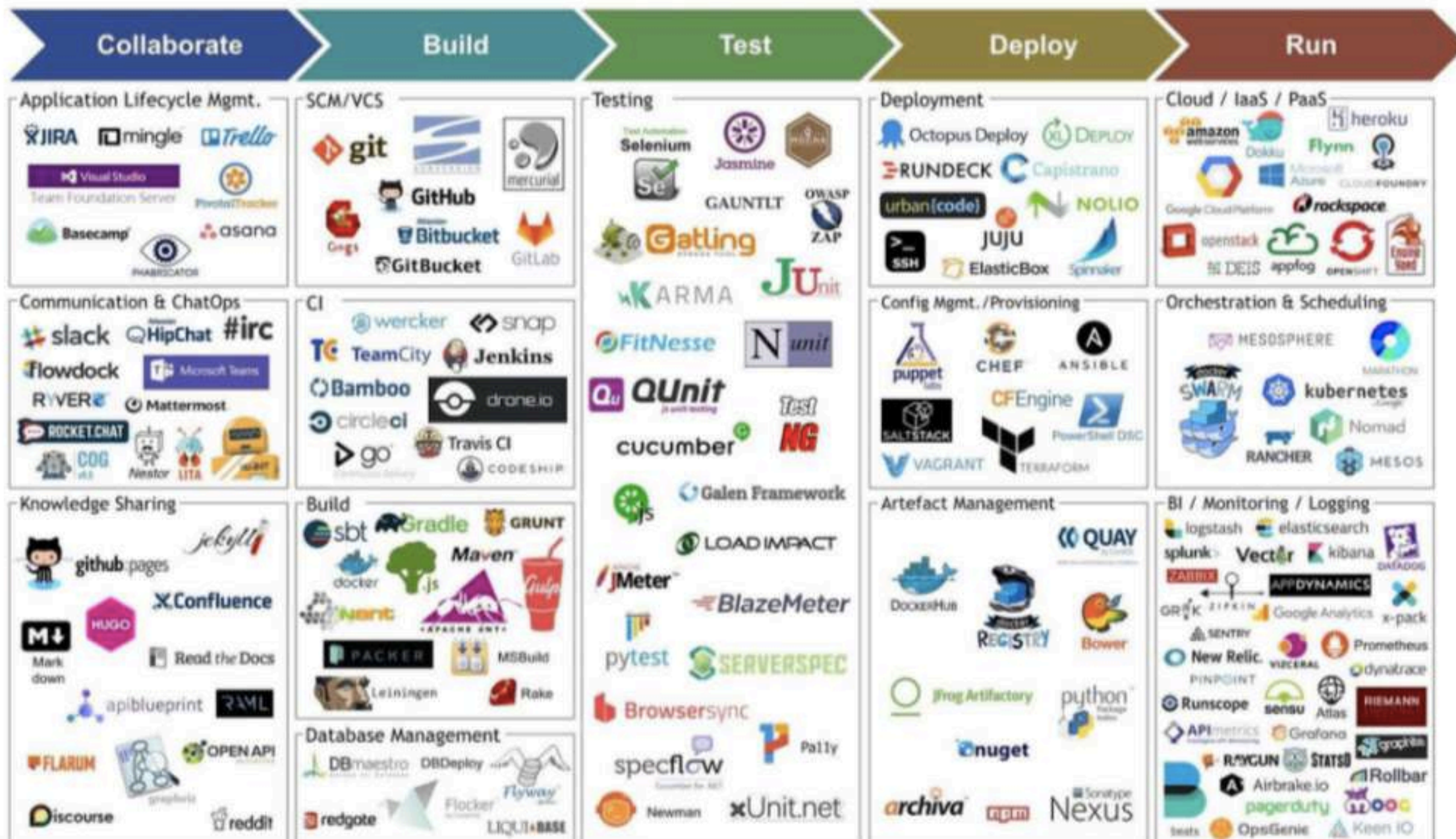
DevOps Tools

Cloud as a Catalyst for DevOps

- Characterization of the cloud by National Institute of Standards and Technology (NIST)
    - On-demand self-service
    - Broad network access
    - Resource pooling
    - Rapid elasticity
        - It is the Capabilities can be elastically provisioned and released

- Measured service

- Software as a Service (SaaS)

- Platform as a Service (PaaS)

- Infrastructure as a Service (IaaS)

Cloud as a Catalyst for DevOps

Software as a Service (SaaS)

- In this the consumer is provided the capability to use the provider's applications running on a cloud infrastructure

- The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based e-mail) or an application interface

- The consumer does not manage or control the underlying cloud infrastructure including networks, servers, operating systems, storage.

- For an example, you can relate google apps, Cisco WebEx, as a Service, Office 365 service, where Provider deals with the licensing of software's

Cloud as a Catalyst for DevOps

Platform as a Service (PaaS)

- The consumer is provided the capability to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider

- The consumer does not manage or control the underlying cloud infrastructure including networks, servers, operating systems, or storage, but consumer has control over the deployed applications and possibly configuration settings for the application-hosting environment

- For an Example: .NET Development platform is considered as a platform
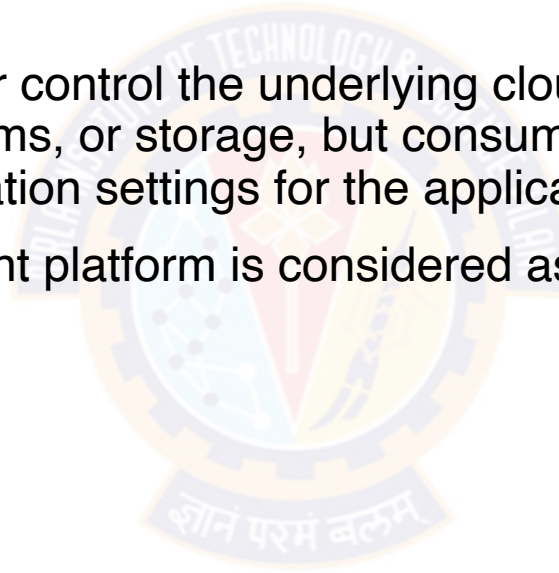
Cloud as a Catalyst for DevOps

Infrastructure as a Service (IaaS)

- The consumer is provided the capability to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications

- The consumer does not manage or control the underlying cloud infrastructure but consumer has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls). For this you can consider any Server Provisioning is IaaS,
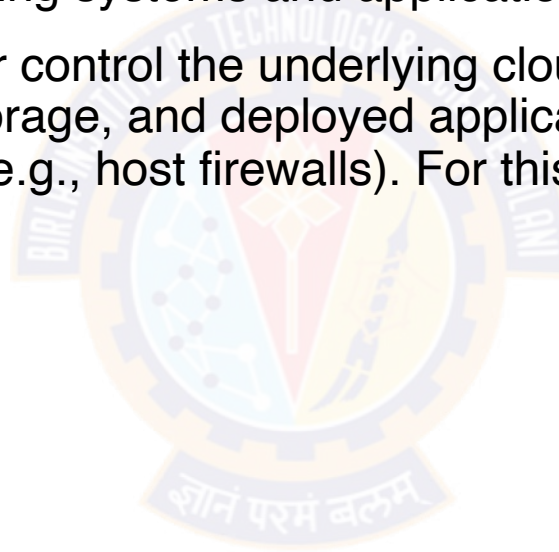
Cloud as a Catalyst for DevOps

Three of the unique aspects of the cloud that impact DevOps

- Three of the unique aspects of the cloud that impact DevOps:
- The ability to create and switch environments simply
  - Simply create and migrate environments is—as is the ease of cloning new instances
- The ability to create VMs easily
  - Administering the running VMs are important to find out for which VM we are paying but not using it
  - Tool such as, Janitor Monkey to scan an account
- The management of databases
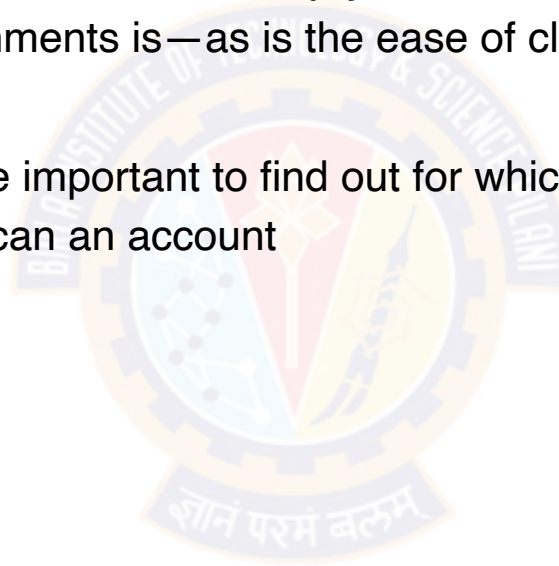
Transition in IT:

State 1 Traditional Agile teams
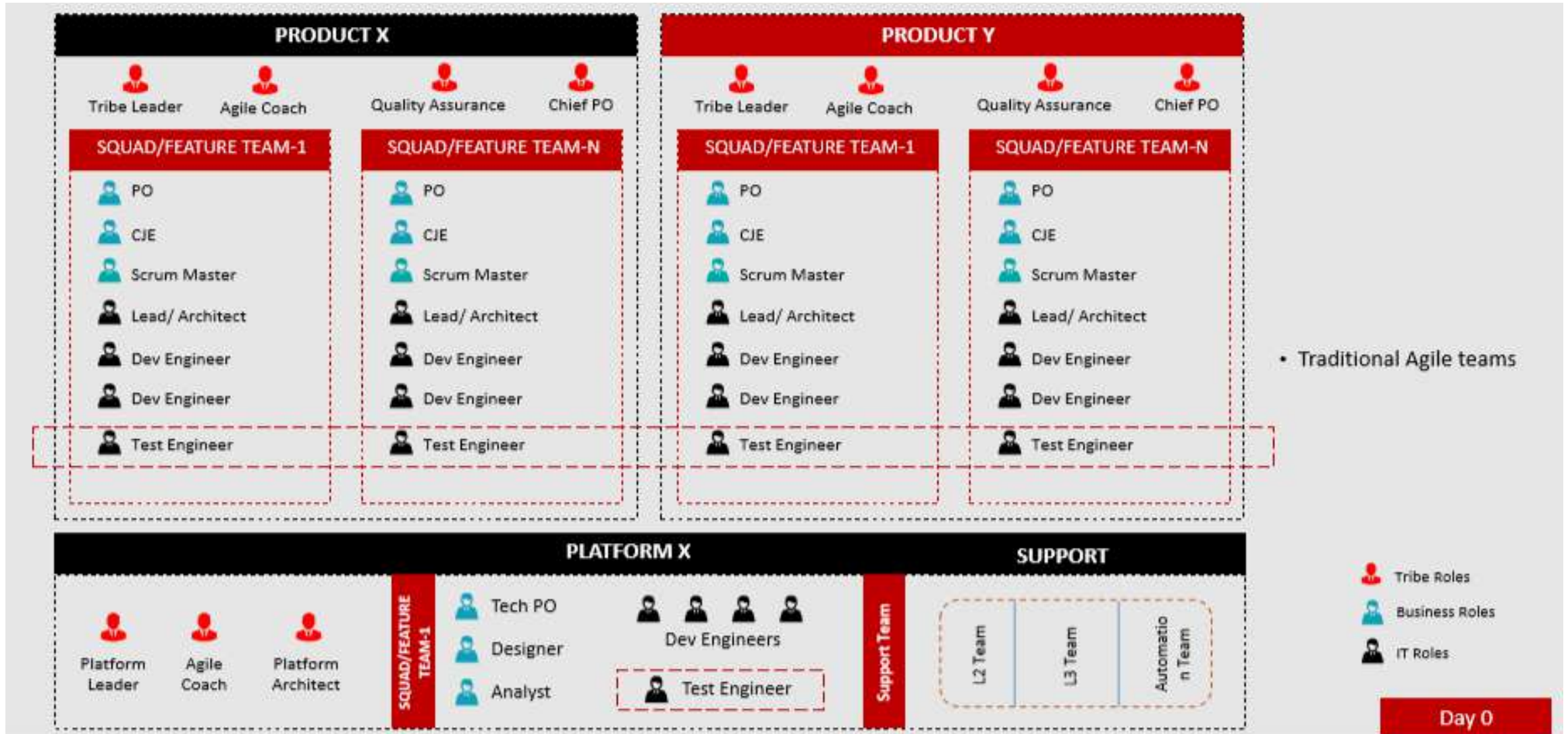
Transition in IT:

State 2 Embedding full-stack engineers and driving Test automation
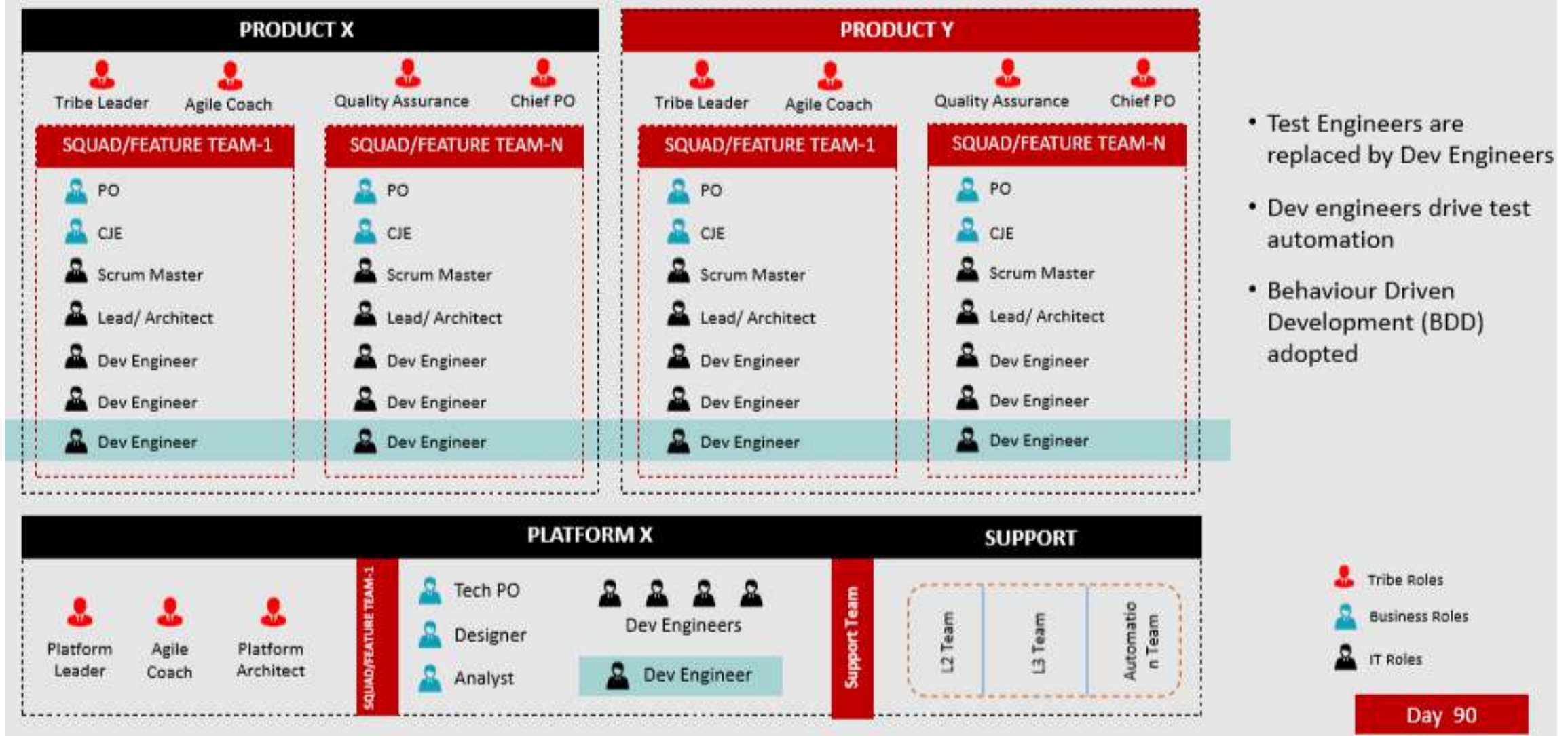
Transition in IT:

State 3 Introduce Chapter Leads
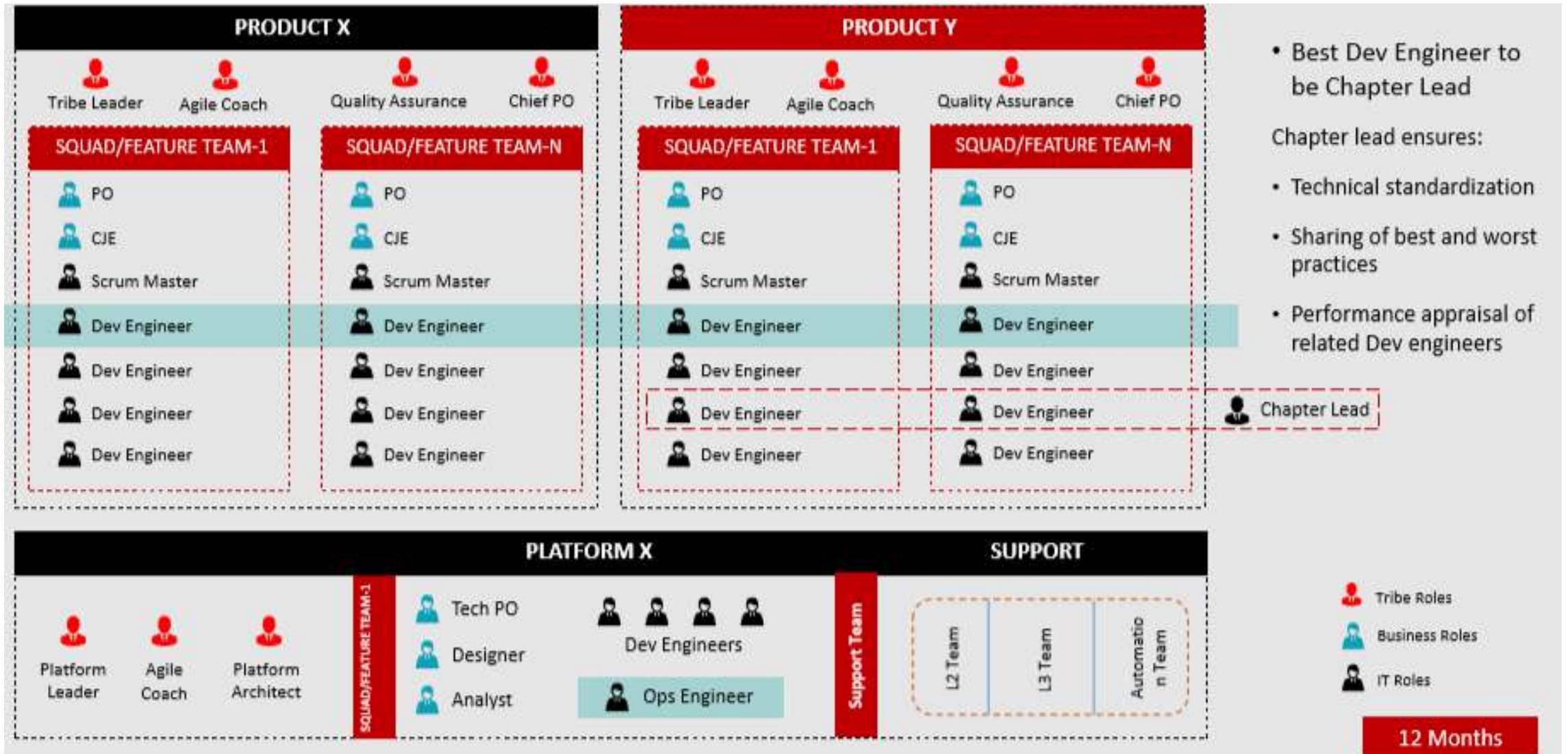
Transition in IT:

State 4 Zero scrum master
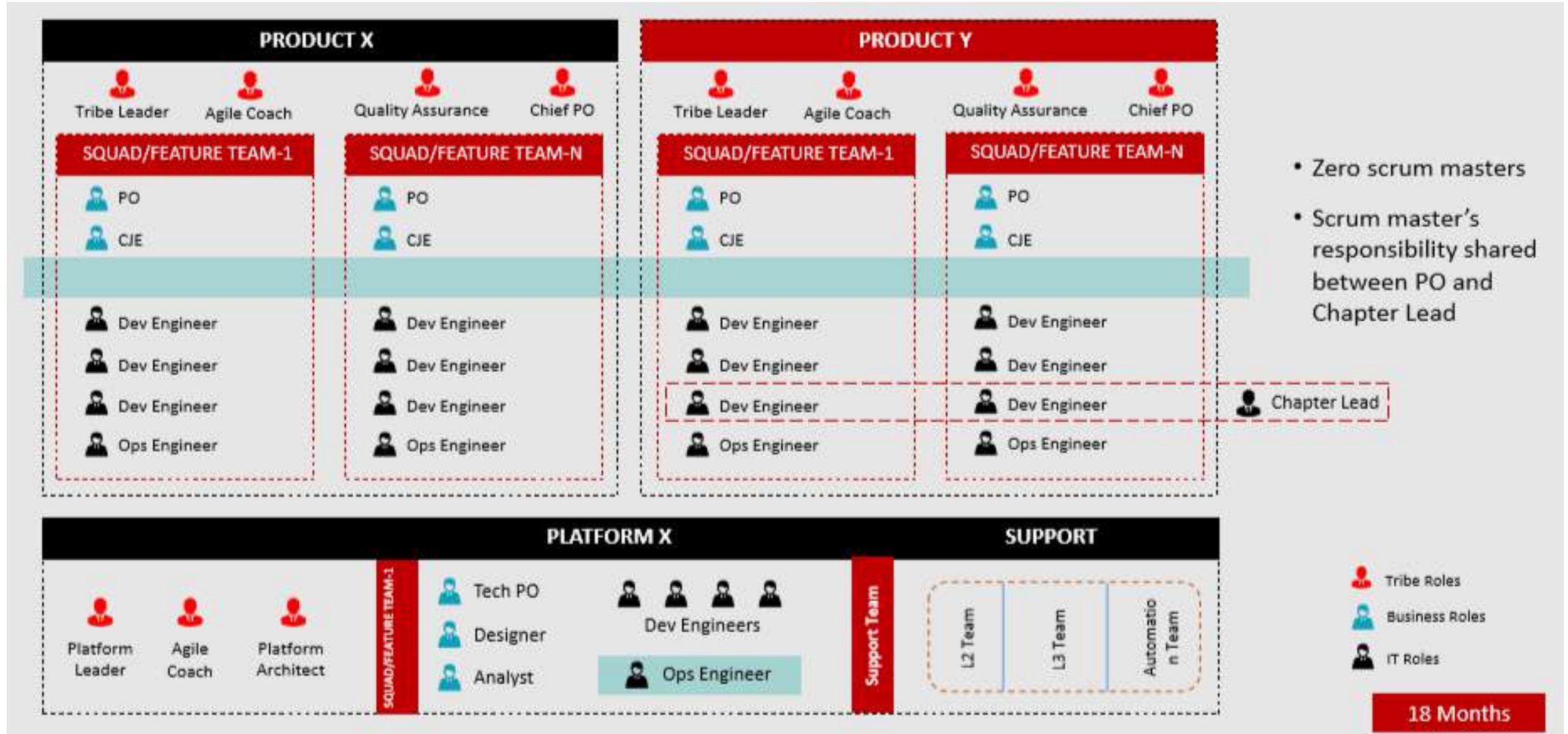
Transition in IT:

ITIL to Google SRE based support
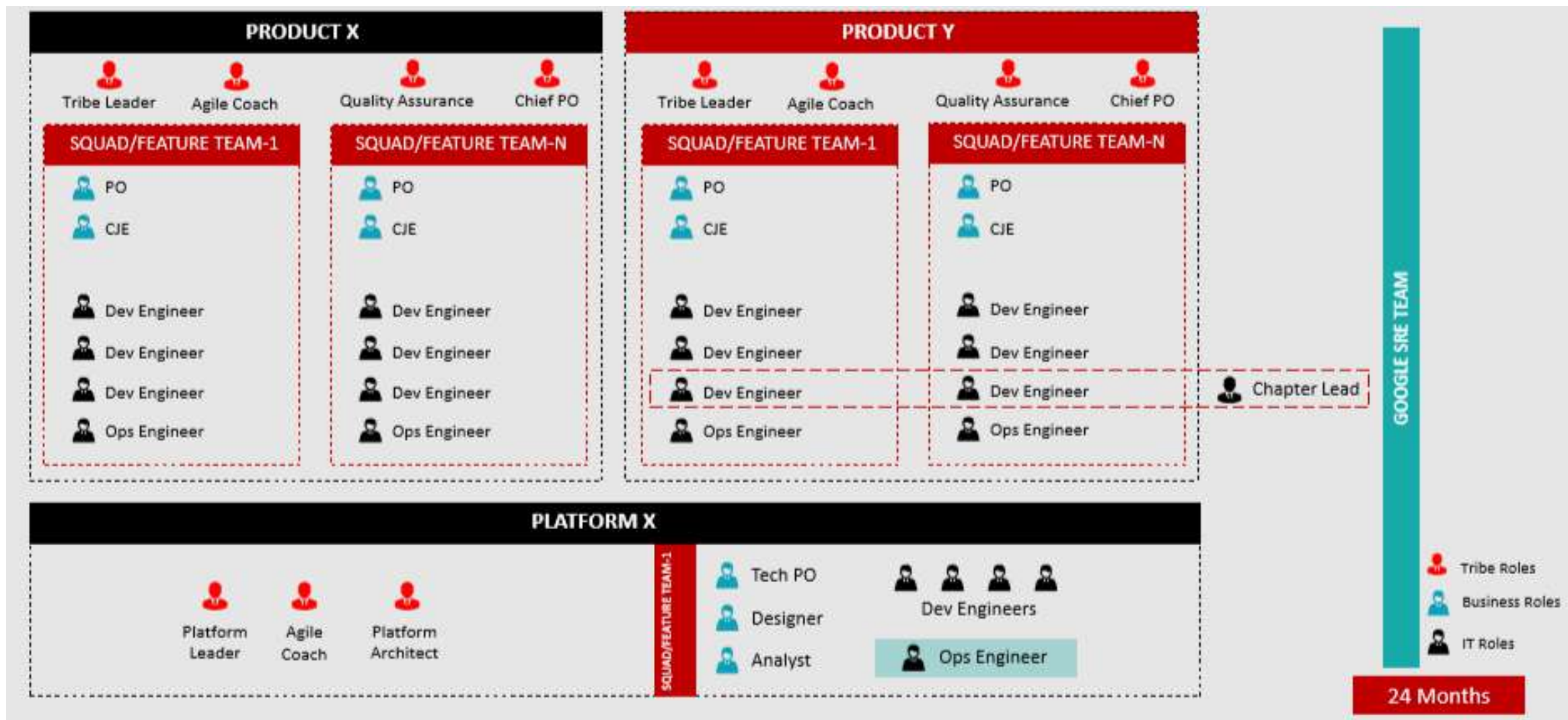
PERIODIC TABLE OF DEVOPS TOOLS (V2) (V1)

Q&A

Thank You!