
# Coca-Cola, SAP and Devops Go well together

Hemant Kochhar and Brian Toms, CONA Services

SID# ASUG 83506

# About the Speakers

**Hemant Kochhar**

- Director, CONA Services
- 19 Years in the Coca-Cola system, supporting Europe and North America
- 22 years SAP experience

**Brian Toms**

- Director, CONA Services
- 31 Years in the Coca-Cola system, domestic & international
- 20 years SAP experience

CONA SERVICES LLC
A COCA-COLA SYSTEM IT SERVICES COMPANY

ASUG

# Key Outcomes/Objectives

1. Understand the CONA Services Agile/DevOps Journey
2. Understand our Supporting Processes
3. Understand our Tools  & Technology

# Agenda

1. CONA Overview
2. Our Direction
3. Our Solution
4. Processes
5. Tools & Technology

# What Does CONA Stand For?

**Coke One North America**

- Standardized technology platform and collection of best business process practices
- Software solution built around SAP
- IT foundation that enabled the implementation of the 21st Century Beverage Partnership Model in North America
- Drives Direct Store Delivery (DSD) and Manufacturing process efficiencies for our bottlers

# CONA Provides Full Scope Solutions

## CUSTOMER

### Integrated Customer Care
- Central order capture
- Service & Issue management

### Integrated Account Management
- One view of the customer
- Integrated customer management
- Knowledge repository

### Sales Force Automation
- Order management
- In-store sales processes

### Pricing / Trade Promotion Mgmt
- Pricing hierarchy (Release 4)
- Promotion & Rebates Mgmt
- Revenue Growth Management
- On- and off-line pricing

### Customer Asset Management
- Equipment placement
- Service & Tracking

## OPERATIONS

### Integrated Operations Planning
- Demand, Operations and Inventory Planning
- Centralized Purchasing and Inventory Movement

### Manufacturing (Release 4)
- Production scheduling
- Production execution & reporting
- Line maintenance

### Warehouse & Transport Mgmt
- Inventory visibility
- Warehouse Productivity
- Spare Parts Inventory Mgmt
- Fleet Management

### Optimized Delivery
- Route & Vehicle space optimization
- Route settlement
- Invoicing

### Full Service Vending
- On-line / predictive ordering

## FINANCE & HR

### Optimize Working Capital
- Financial Accounting
- AR / Collections
- Dispute resolution
- Accounts Payable
- Credit Management
- Travel & Expense Management

### Single source of HR data
- Compensation & benefits
- Performance Management
- Talent Management
- Organization Management
- Learning

## REPORTING & BI

### Integrated Reporting
- CONA and Legacy data
- Self-Service Reporting
- Executive Dashboards

## Master Data Management

# CONA in Figures

## OPERATIONS to date

511 sites deployed

12 bottler partners

10,074 daily routes

## Focus areas

28,468 active users in ECC this month

1,933,502 ship-to customers

**1.2B** cases manufactured

**80,373** bottler employees

**43M** deliveries

**5.3M** equipment service orders

## Our Partners

ABARTA Coca-Cola Beverages

Coca-Cola Bottling Company United

Liberty Coca-Cola Beverages

Great Lakes Coca-Cola Bottling

Reyes Coca-Cola Bottling

Coca-Cola Beverages Florida

Coca-Cola Southwest Beverages

Swire Coca-Cola, USA

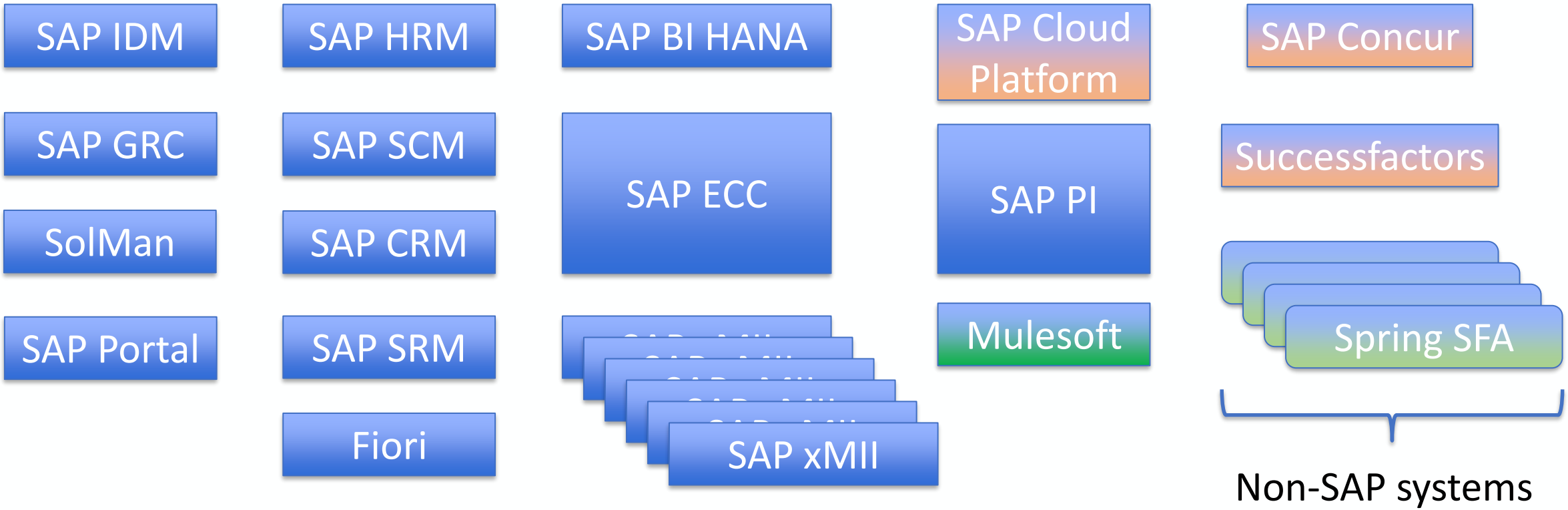
Coca-Cola Bottling Co. Consolidated

Coca-Cola Refreshments Canada

Coca-Cola Bottling Company of Northern New England

heartland Coca-Cola Bottling Company

# Our Technology landscape




Fun facts: ~ 1,400 servers; Largest system is > 24TB; Large scalable-on-demand MicroFocus SaaS performance testing environment – record 15,000+ concurrent users for 17+ hours

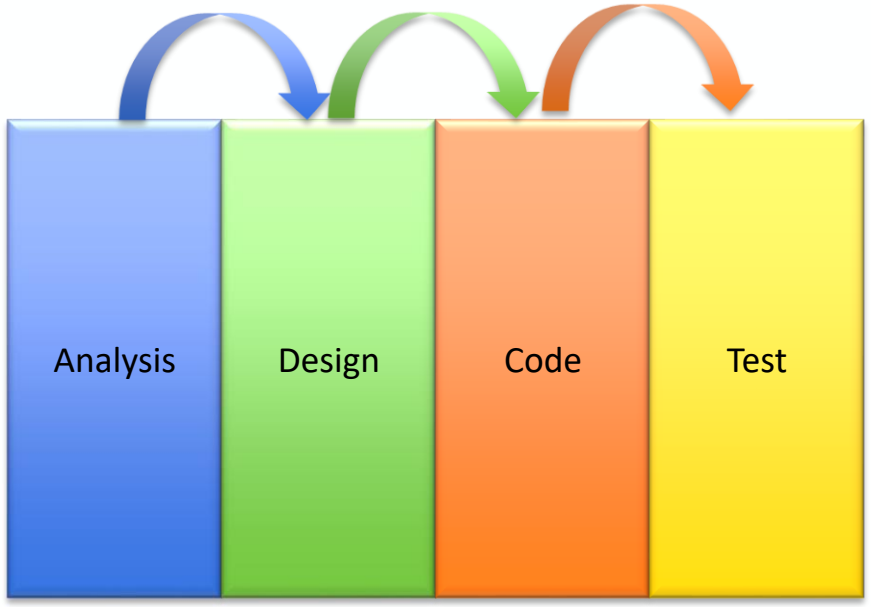
https://www.microfocus.com/media/success-story/cona_services_llc_ss.pdf

# Our Reality

- Historically focused on Deployments
- Migration to Operational State

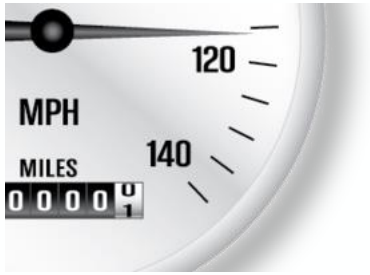
- Waterfall Methodology
  - Long lead times
  - Requirements unclear, frequently changing
  - Lots of re-work
  - High cost
  - Testing and documentation “squeezed”

- Major / Minor Releases
  - Monthly Release Exceptions
  - 2 Major Releases




Traditional Waterfall

# Our Direction



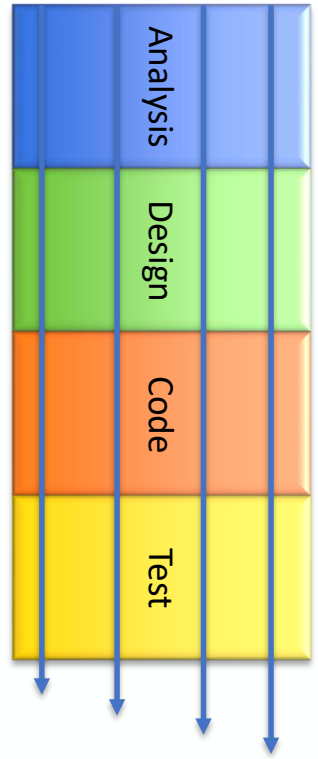


# Our Solution

**The Solution**

A shift from Waterfall Methodology to Agile/DevOps Methodology has the **potential** to deliver on all of these topics.  It's a Journey.

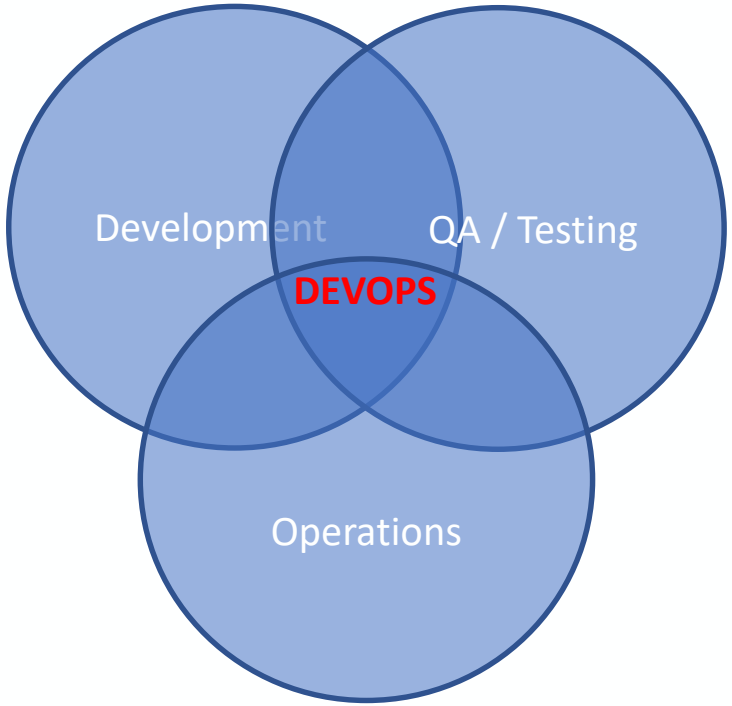
- Agile First
- Break up requests into small, manageable, independent pieces of work
- Document and confirm "what" is needed
- Deliver a Minimal Viable Product (MVP) as quickly as possible, if possible
- Test what needs to be tested
- Use automation where possible (Testing, Data, etc.)
- Release often
- Validate and fix issues immediately
- Simplify landscape
- Review and adjust frequently




Continuous Production Delivery

# What is DevOps?

- DevOps provides a set of practices and **cultural changes**—supported by complementary tools—that **automates** the software delivery pipeline, enabling organizations to win, serve, and retain consumers better and faster than ever before.
    - Amy DeMartine – Principal Analyst – Forrester*




Plan
Design
Build
Test
Release
Deploy
Operate
Monitor

*https://electric-cloud.com/blog/2016/06/wheres-heat-talking-devops-use-cases/

# How do we get to an ‘Agile’ way of working?

Problems

Small changes

Enhancements

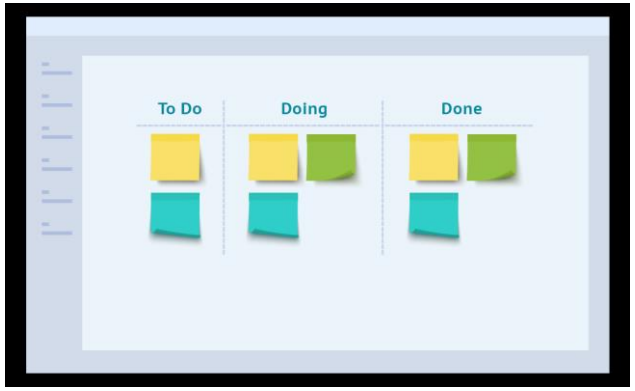
Projects

Single Product Backlog

Agile Scrum


Scrum execution

Kanban



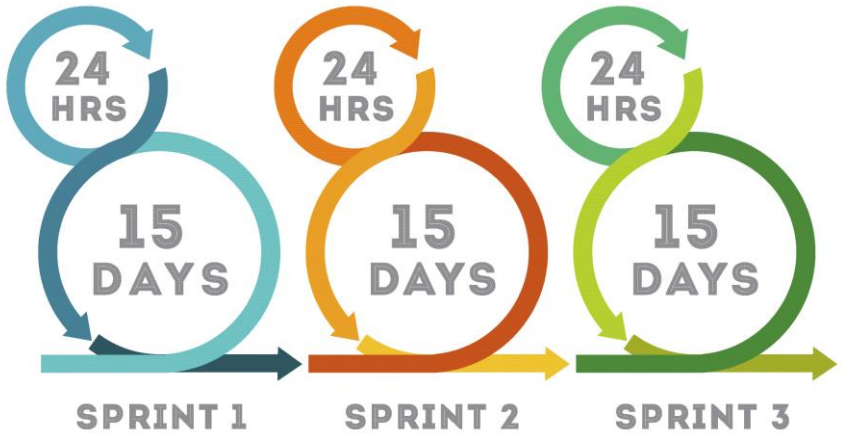

Kanban execution

Iterative refinement

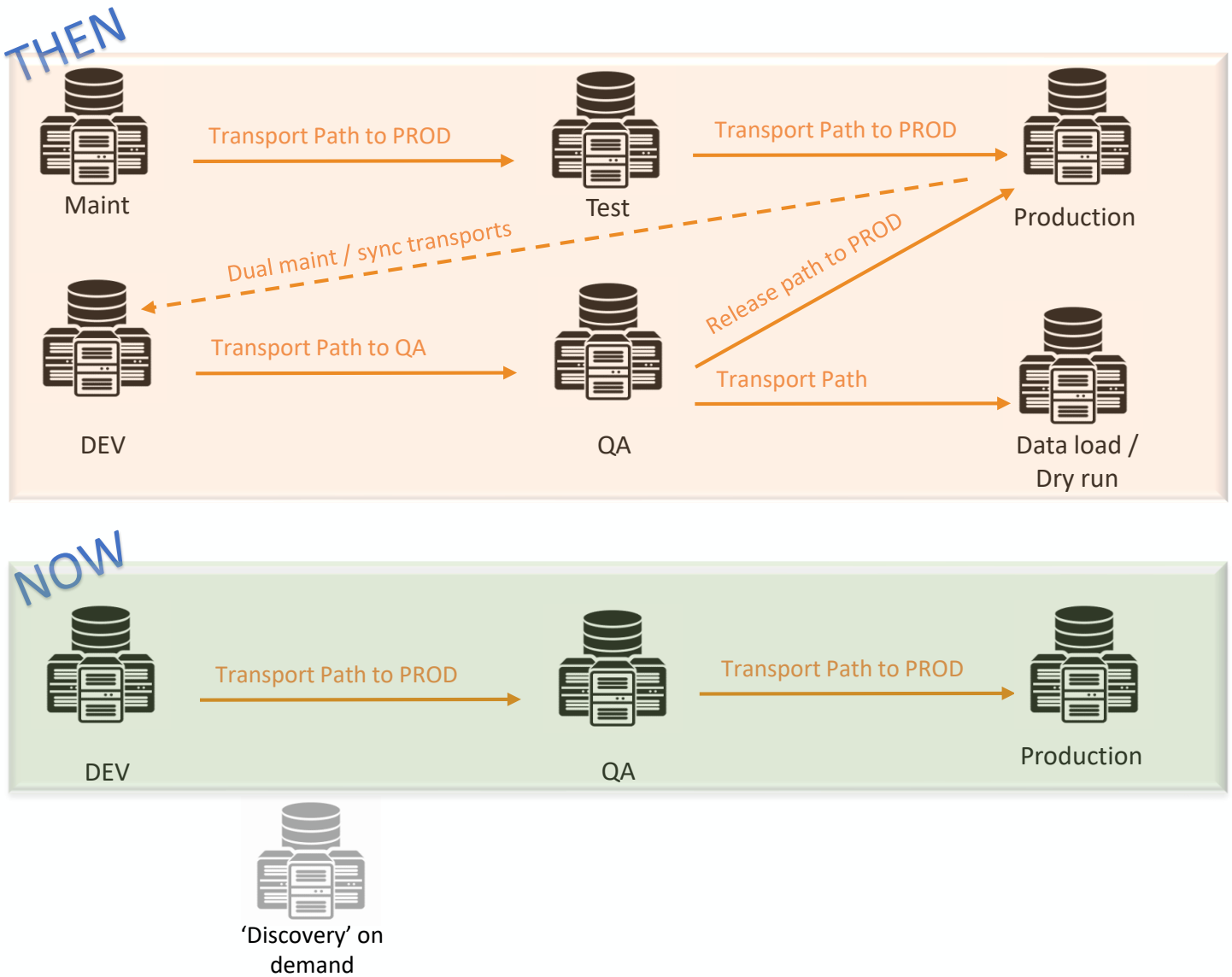
ASUG

# What have we done?

- Initially focused on 2 pilot functional areas with the largest immediate value
- Provided Agile/DevOps training for teams
  - Engaged 3rd party Agile coaches
  - Backlog development and grooming
  - Agile tool evaluations
  - Landscape simplification
  - Automated Testing
  - Automated data provisioning
  - Automated monitoring
  - Enable Now for Demos & Weekly Release Notes
- Single Product backlog per functional
  - Epics = Large Projects & Ongoing Initiatives
  - Features = Change Requests, Problem Tickets, Operational Changes
- Agile Scrum AND Kanban
  - Including all enabling functions (security, data, reporting, etc.)

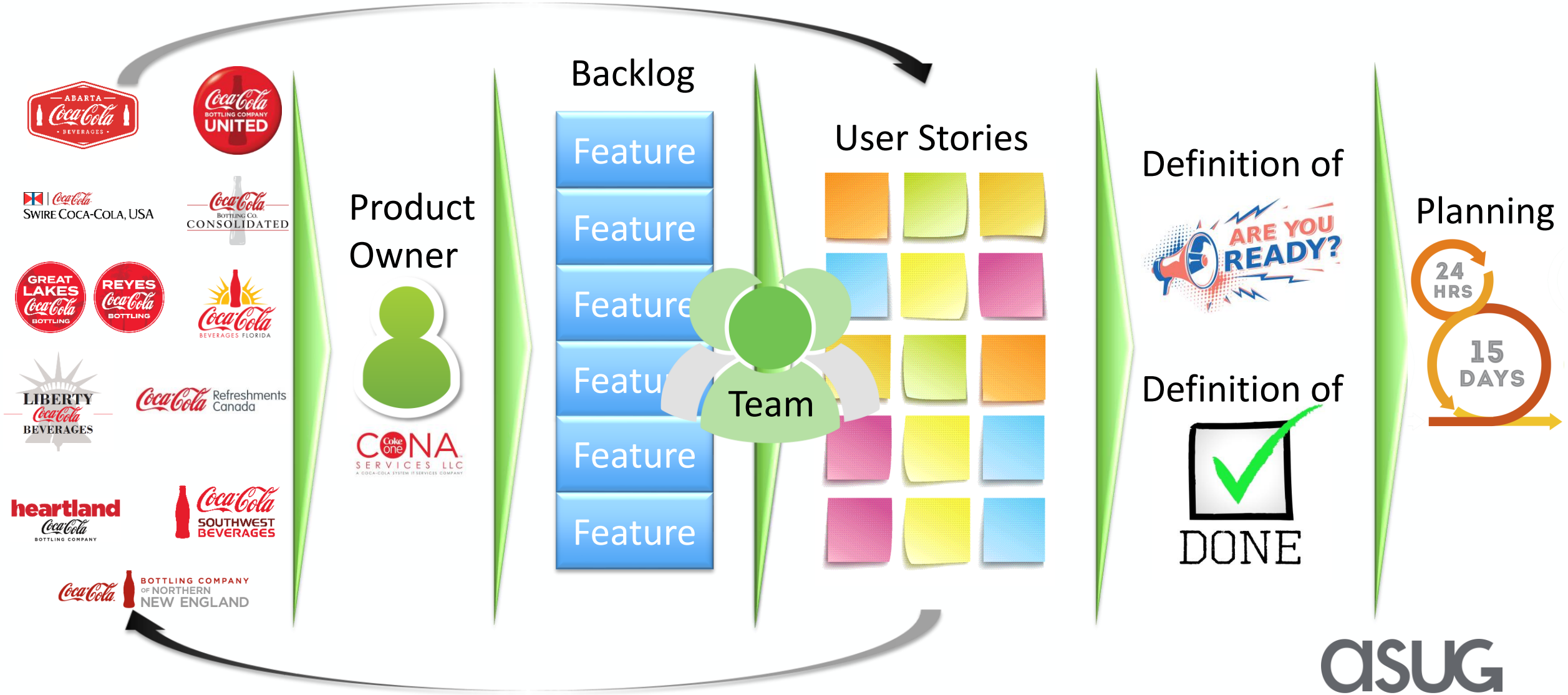


# Simplify SAP landscapes




- Project landscape + Maintenance landscape?
  - Parallel landscapes not needed **IF**
    - able to move changes through the environment rapidly
  - Use single 3 system landscape
    - DEV→QA→PROD

# Backlog Grooming & Planning



# DevOps – Shift Key Development efforts LEFT

## Development

- **Significantly** cheaper to address defects in development v/s QA or production
- **Developer Automation Tools**
  - SAP Code Inspector: Analyze Code Quality and Complexity
    - Standards, rule checks
    - Custom rules
  - Livecompare CodeWatch:
    - Code quality rules
    - Support pack checks
    - Upgrade checks
  - Livecompare Impact Analysis
    - Dependencies with other code
    - What is missing / out of sync
    - What will be impacted if code is changed

## Testing

- **Reduce risk - Focus** on testing everything that has changed
- **Tester Automation Tools**
  - Livecompare: Analyze changes made ( config or Code) and link to all test cases that need to be run
    - Limit to transactions / reports that are actually used !
  - Test Automation: Automate regression testing
    - Automate loading of test data sets
    - Automate running of test script sets
    - Alerts if any tests fail

## Operations

- **Limit risk and outages in production**
- **Operations (Release) Automation Tools**
  - Rev-Trac:
    - Transport dependencies and management
    - Overtake & Overwrite protection
  - Rev-Trac: Sensitive objects checks ( DB, Indexes, etc )
- **Automated Archiving**
  - PBS & TJC ( Serrala ) toolset
  - Over 60TB Archived over the last couple of years
- **Cloud platform automation**
  - Monitor and de-provision as needed

# New ways of working

- **Shift Left**
  - Product Owner and bottlers approve functionality during Sprint Reviews
  - Testing & Reviews in **Dev** → Need good test data
  - Training created during Sprints
- **Visibility**
  - Teams own making issues & success visible
  - Blockers made visible each day
- **Change ways of working**
  - Reduce need for formal documentation up front
  - The team decides how much can be accomplished at the beginning of a sprint; track metrics and determine velocity
  - Scope Change is measured by user story and Definition of Done; not by specs doc
  - More flexibility to deliver a better product

# Metrics & KPIs – measure success

| Measure | Items |
|---|---|
| Backlog trends | Service Catalog Requests, Problems, CRs, Incidents, Defects, etc. |
| Speed to deliver to UAT<br>( Velocity ) | Clearer requirements (Definition of Done)<br>Increase Automated Testing (Dev & QA) |
| Speed to deploy | Approval to UAT; UAT to Production<br>Approval to Production; Daily Production Moves |
| Quality | Duration of UAT; Number of defects in UAT; Number of Incidents in Production |
| Cost | Overall Cost from Approval to Deploy |

# DevOps - Release Process

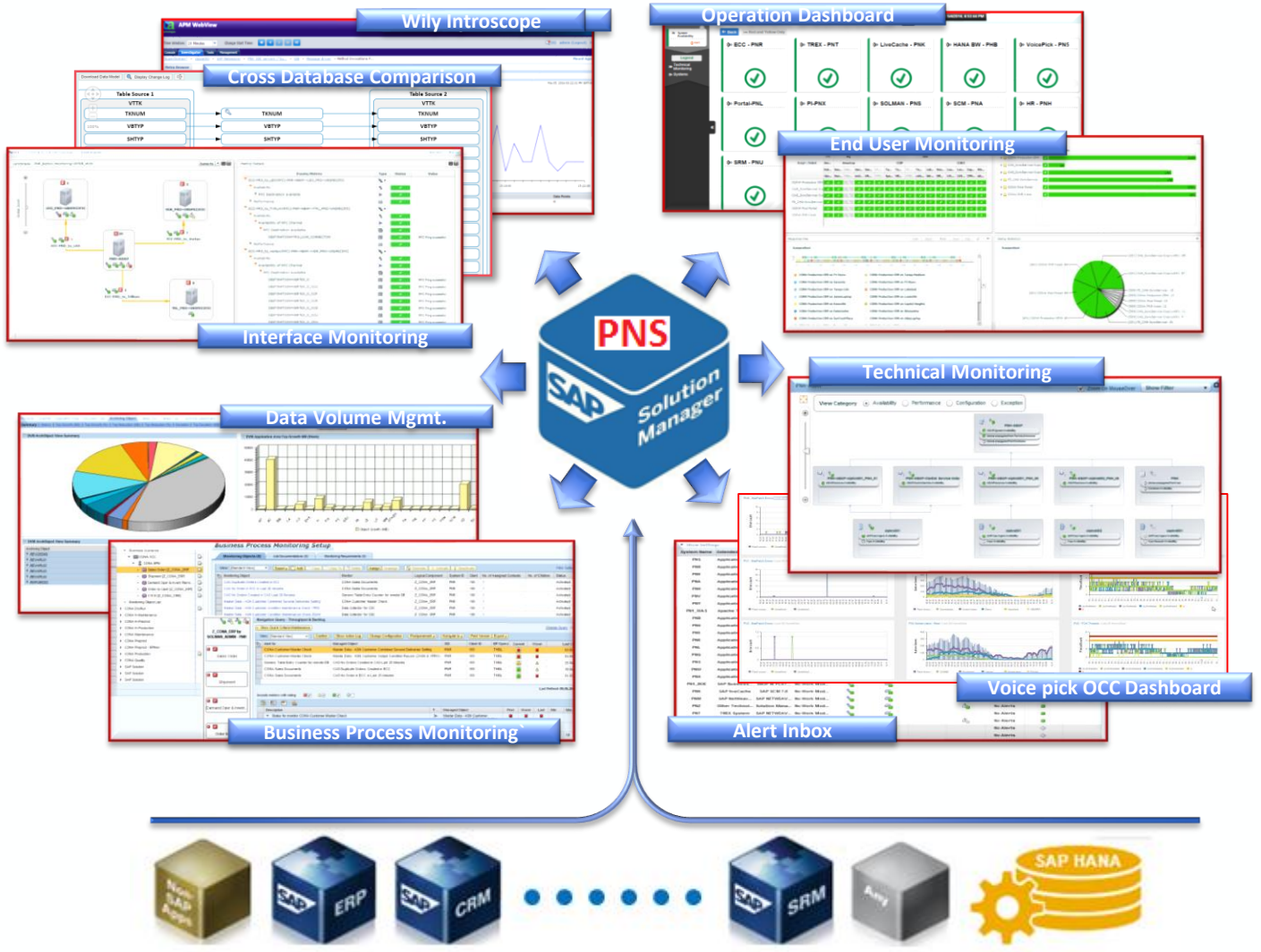
## Release code and config to production when ready

- at least once a day
- typically several times a day – based on the right time for each business process
- Brief review every morning

## Small numbers of changes to production at one time

- Identify issues and root cause immediately
- Have the right folks ready to jump in  if there is a problem
- ALWAYS validate with an end user right after the change is in production

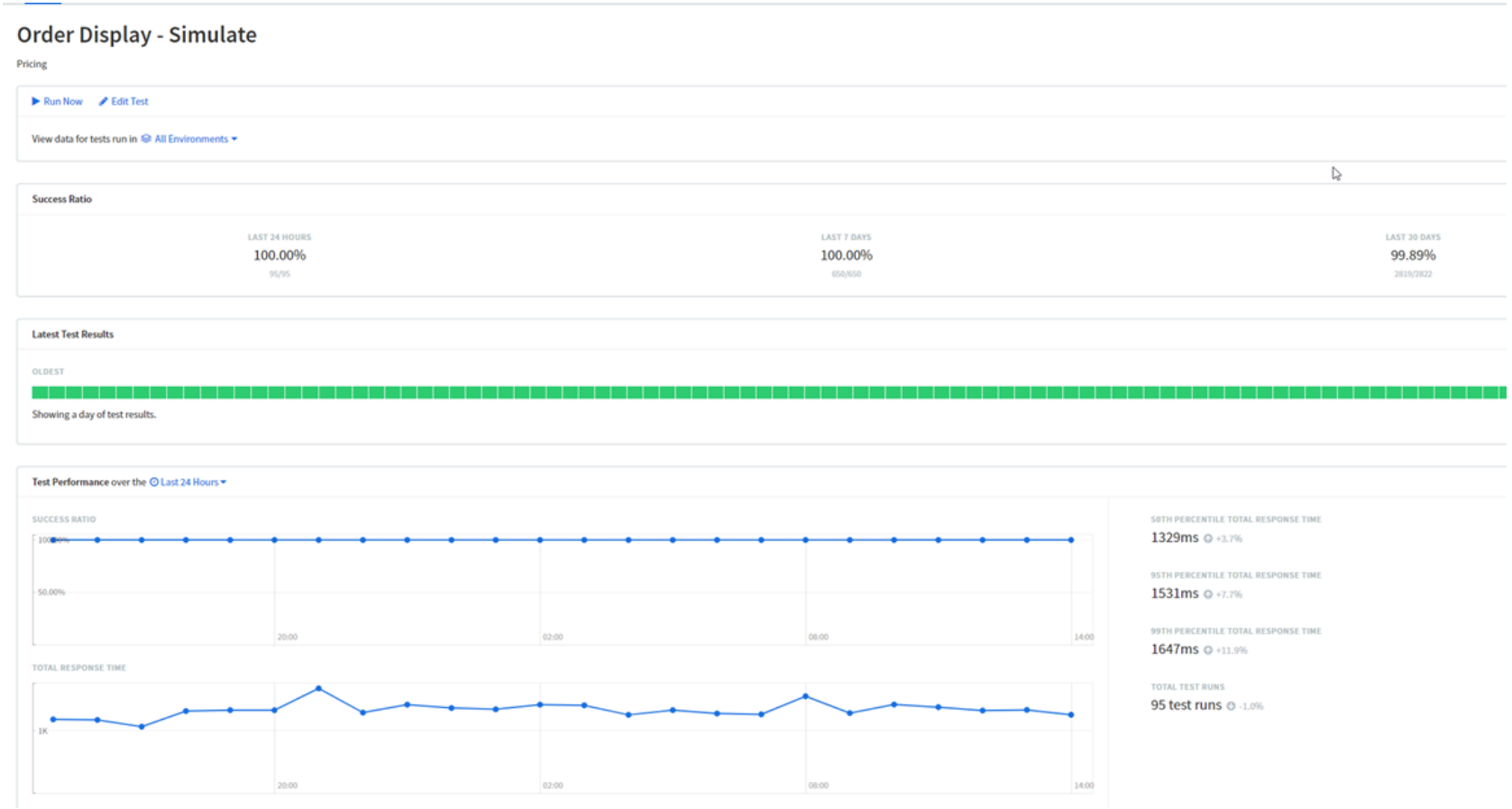
# DevOps – Monitoring and Automation




- Alerts
- Automated, rules based actions
  - Thresholds
  - Incident creation

- Response times
- Dumps
- Queues
- DB performance
- Transaction performance
- User experience in different locations (UxMon)

# DevOps example: Automation of API monitoring




- Tools (Runscope) that call APIs in **production** from distributed locations in North America
  - Track response times
  - If thresholds exceeded, trigger alerts
  - If errors reported, trigger alerts
- Know about issues before users experience them
  - Take action

# Seven Key takeaways

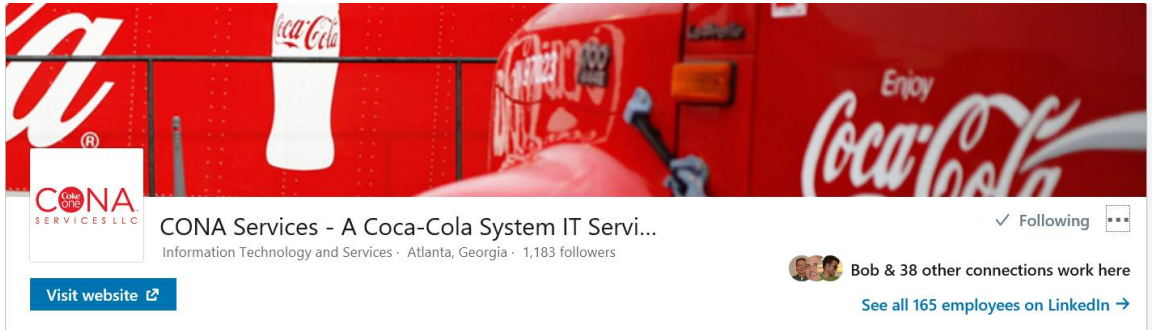
- **Change Management** – it is a cultural shift…and everyone must be onboard
- **Backlog** – its all about a single backlog of items to be delivered
- **Shift Left, Definition Of Ready (DOR), Definition Of Done (DOD)**
- **Automation** – automate everything you can, especially the testing !!
- **Monitoring** – Use automation to alert and react
- **Measuring** – Use KPIs that give visibility to issues and successes
- **Retrospective** – look back, review and refine - constantly

# Glossary

- **Agile** – A set of development methodologies based on iterative development, where solutions evolve through collaboration between self-organizing cross-functional teams.
- **Backlog** – A document tracking all requirements the scrum team is aware of, ranked by priority driven by the CONA and the bottlers.
- **Definition of Done** – A set of rules agreed upon by bottlers and CONA that determines when a User Story has been fully developed with all expected functionality.
- **Definition of Ready** – A set of rules agreed upon by bottlers and CONA that determines when a User Story is ready to begin work as part of a Sprint.
- **DevOps** – The concept of using automation to make both Development and Operations teams more efficient. **Not** a cure-all for poor processes or communication, this is why Agile is a precursor to effective DevOps– if treated as such it will simply deploy bugs quicker.
- **Feature** – A grouping of User Stories that make up a larger, unified piece of CONA functionality (Change Requests, Problem Tickets, etc.).
- **Grooming** – One of the Scrum ceremonies that is oriented around adding appropriate detail and re-prioritizing the various User Stories that exist on the Backlog.
- **KanBan** – One flavor of Agile, based on the movement of requirements through a column-based board that tracks the full lifecycle of their development. These items generally do not follow the full Scrum process– because they represent common/repeatable tasks that do not require further breakdown into smaller tasks (i.e., Service Catalog Changes – adding a new printer).
- **Scrum** – A second flavor of Agile best tailored to more complex development where solutioning is required. Scrum is based on the concept of using Sprints to address User Stories via timeboxed ceremonies.
- **Sprint** – A timebox (generally 1-4 weeks) based on a Sprint Plan that establishes a Scrum team's workload. Work is delivered throughout the Sprint and does not wait until the end of the Sprint to deliver.
- **User Story** – A specific item of work under Scrum that aims to define the smallest possible increment of realizable business value that the Scrum team can deliver independent of other requirements. The Sprint Plan is made of up a collection of User Stories.
- **Velocity** – The overall per-Sprint work capacity of a Scrum team (or an individual Scrum team member) based on estimates of all assigned User Stories

# Follow Us

LinkedIn




www.conaservices.com


ASUG

# Take the Session Survey.

We want to hear from you! Be sure to complete the session evaluation on the SAPPHIRE NOW and ASUG Annual Conference mobile app.


ASUG

# Presentation Materials

Access the slides from 2019 ASUG Annual Conference here:

http://info.asug.com/2019-ac-slides

# Q&A

For questions after this session, contact us at hkochhar@conaservices.com and btoms@conaservices.com.

# Let’s Be Social.

Stay connected. Share your SAP experiences anytime, anywhere.

Join the ASUG conversation on social media: **@ASUG365 #ASUG**


ASUG