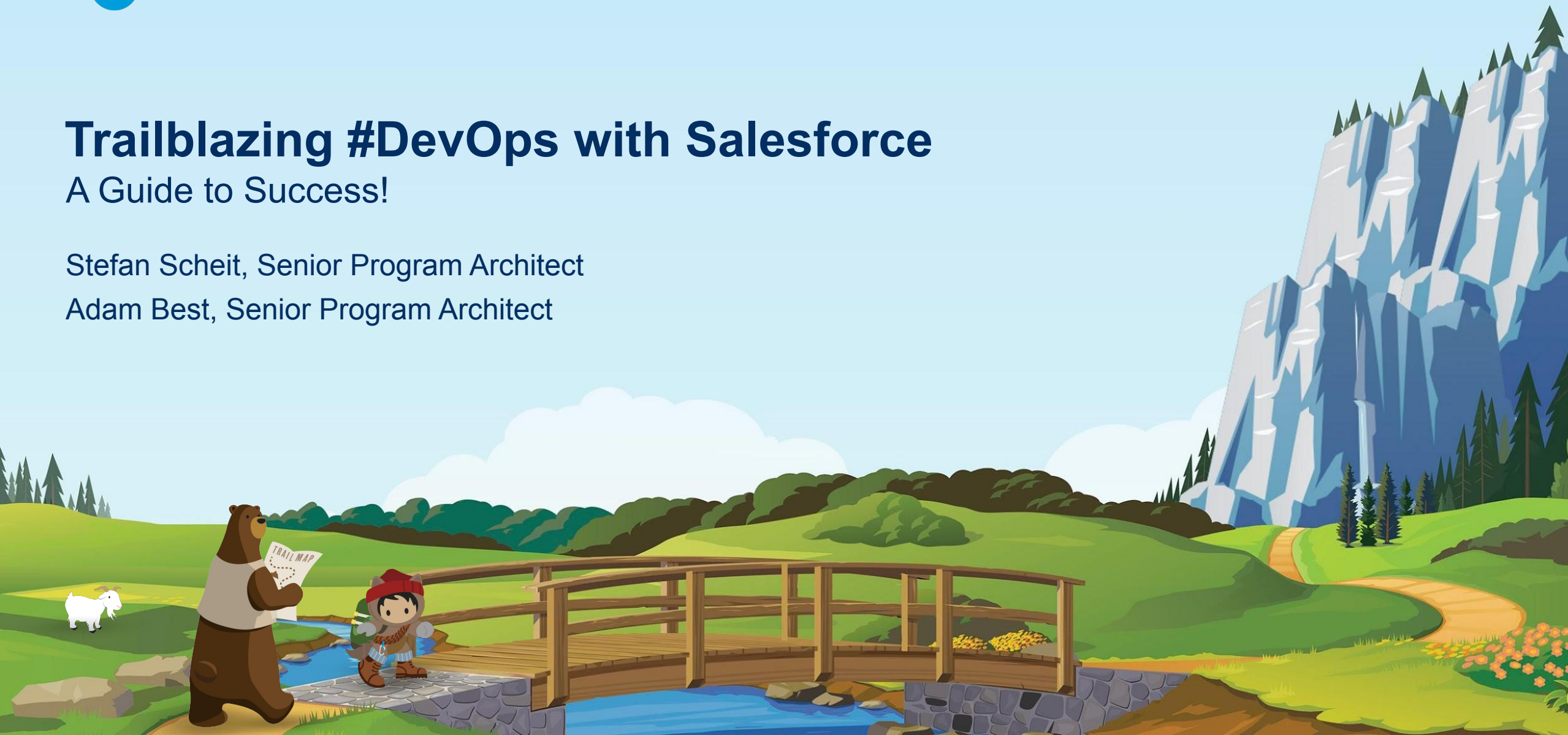
Trailblazing #DevOps with Salesforce

A Guide to Success!

Stefan Scheit, Senior Program Architect

Adam Best, Senior Program Architect



Forward-Looking Statement

Statement under the Private Securities Litigation Reform Act of 1995

This presentation may contain forward-looking statements that involve risks, uncertainties, and assumptions. If any such uncertainties materialize or if any of the assumptions proves incorrect, the results of salesforce.com, inc. could differ materially from the results expressed or implied by the forward-looking statements we make. All statements other than statements of historical fact could be deemed forward-looking, including any projections of product or service availability, subscriber growth, earnings, revenues, or other financial items and any statements regarding strategies or plans of management for future operations, statements of belief, any statements concerning new, planned, or upgraded services or technology developments and customer contracts or use of our services.

The risks and uncertainties referred to above include – but are not limited to – risks associated with developing and delivering new functionality for our service, new products and services, our new business model, our past operating losses, possible fluctuations in our operating results and rate of growth, interruptions or delays in our Web hosting, breach of our security measures, the outcome of any litigation, risks associated with completed and any possible mergers and acquisitions, the immature market in which we operate, our relatively limited operating history, our ability to expand, retain, and motivate our employees and manage our growth, new releases of our service and successful customer deployment, our limited history reselling non-salesforce.com products, and utilization and selling to larger enterprise customers. Further information on potential factors that could affect the financial results of salesforce.com, inc. is included in our annual report on Form 10-K for the most recent fiscal year and in our quarterly report on Form 10-Q for the most recent fiscal quarter. These documents and others containing important disclosures are available on the SEC Filings section of the Investor Information section of our Web site.

Any unreleased services or features referenced in this or other presentations, press releases or public statements are not currently available and may not be delivered on time or at all. Customers who purchase our services should make the purchase decisions based upon features that are currently available. Salesforce.com, inc. assumes no obligation and does not intend to update these forward-looking statements.


THANK YOU
RANGER
EXPLORER
ADVENTURER

Speaking from experience today…

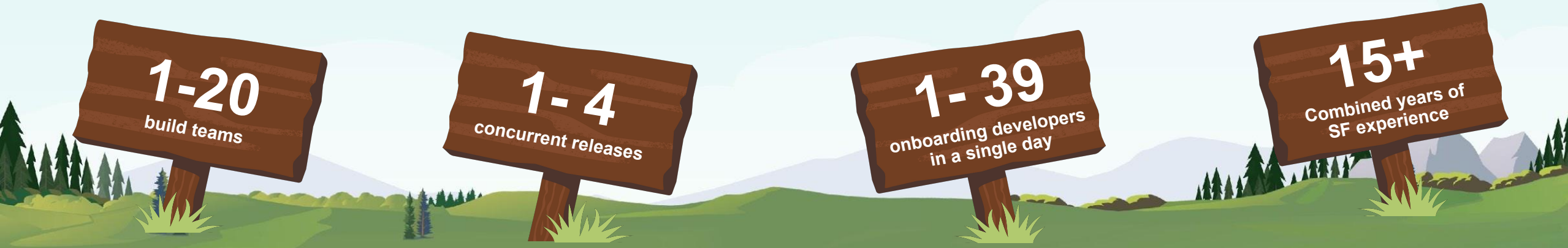
Advisory Services, salesforce success cloud

Stefan Scheit

Senior Program Architect

Adam Best

Senior Program Architect



First, some logistics

Questions, answers and staying in touch.

- **How do you ask a question?**

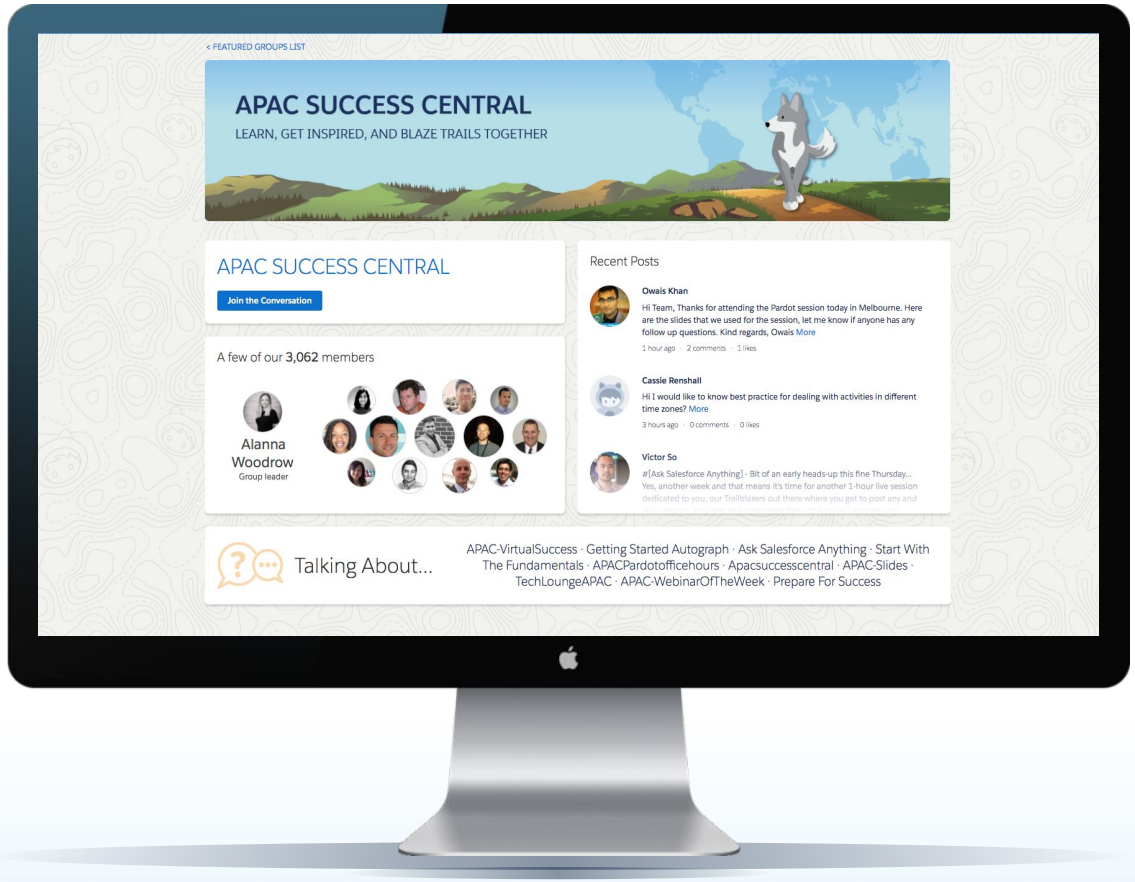
  Please ask in the APAC Success Central Community - Link is available in the chat window. We will answer these around half time and at the end

- **Will this be recorded?**

  Yes! This will be shared on the community and post event email

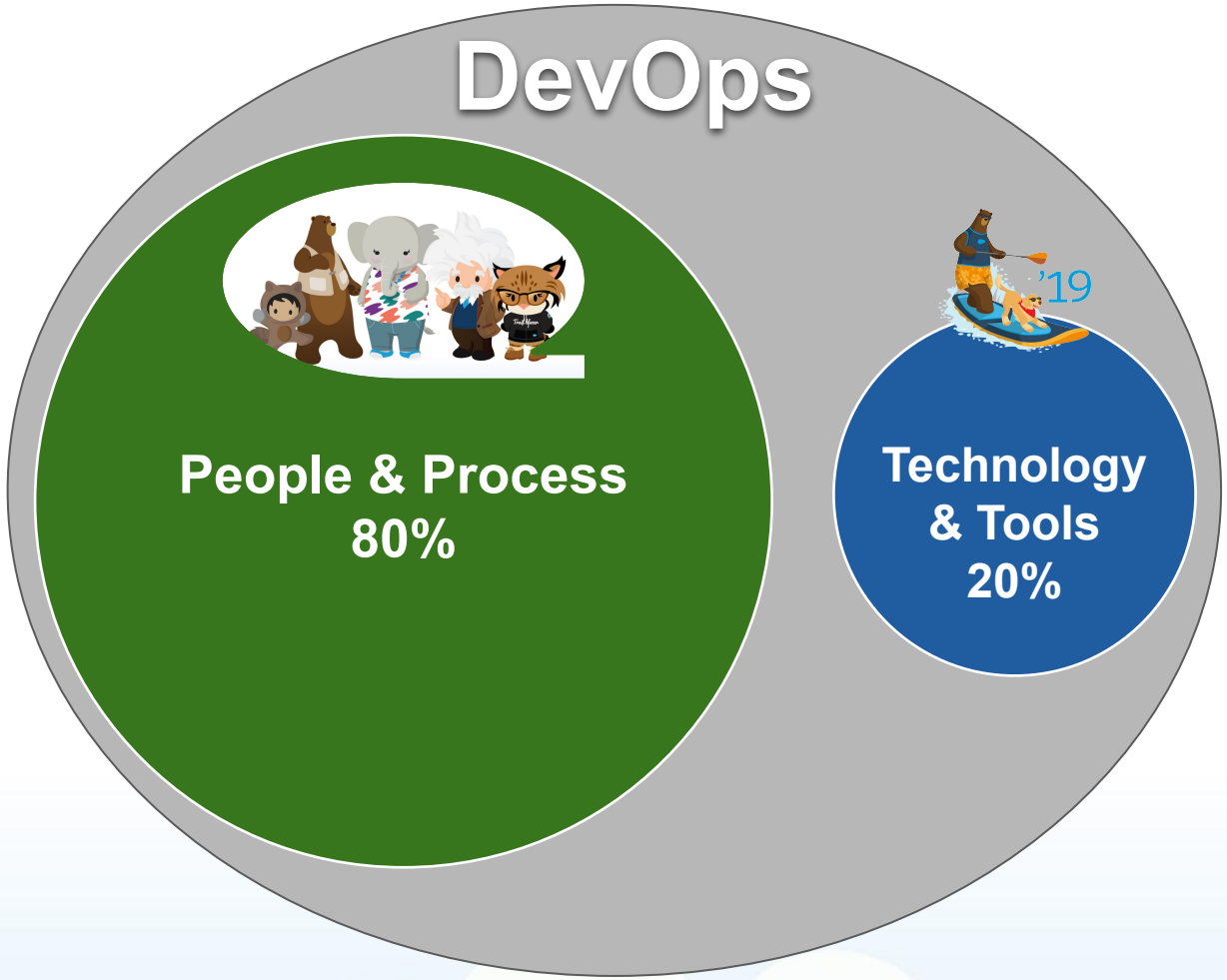
- **Where can you get the presentation?**

  This is posted in the APAC Success Central Community



DevOps, CICD & DX

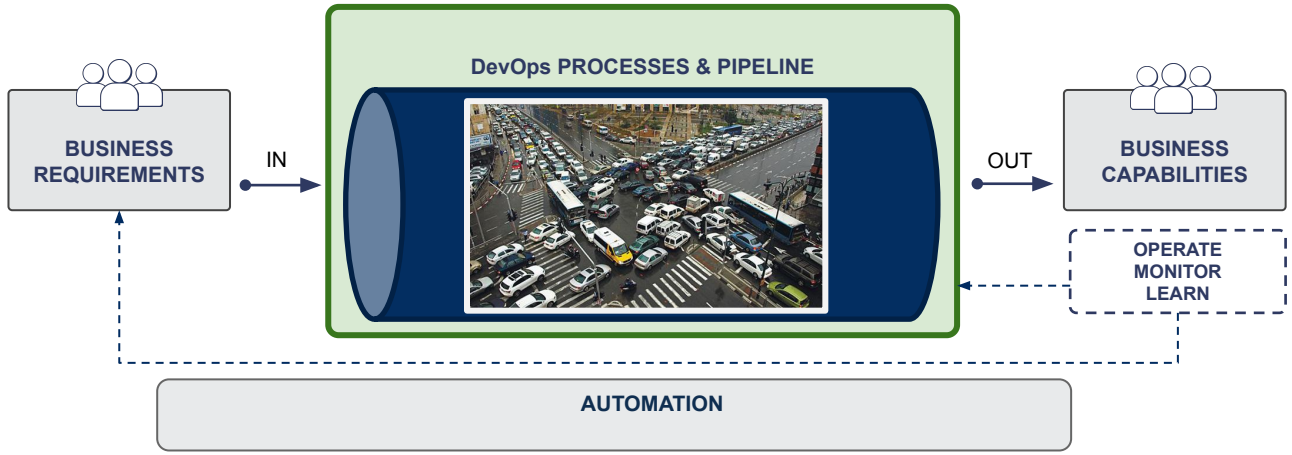
- **DevOps** (Development Operations) is an ***organisational capability*** - people, process, & technology all working together
- **CICD** (Continuous Integration Continuous Delivery) is a ***tooling technology*** that can be used as part of a well-functioning DevOps capability.
- **DX** is a specific set of **Salesforce *development & deployment technologies*** that can be used as part of a well-functioning DevOps capability.



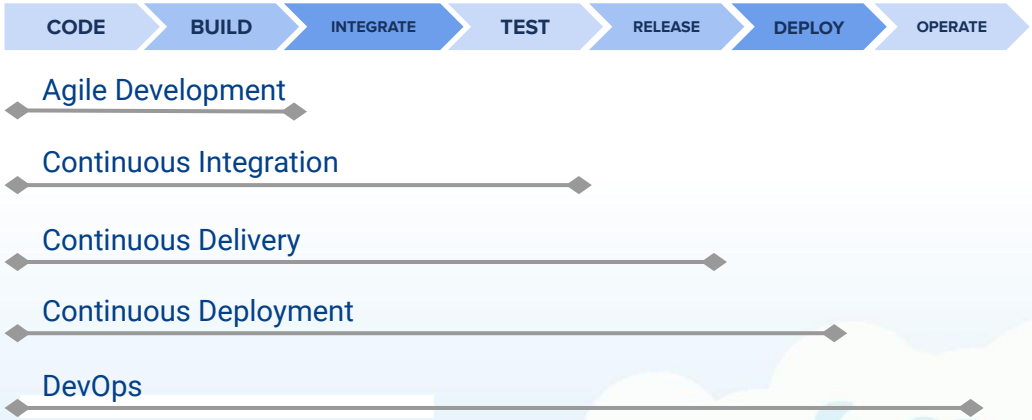
Increasing Software Delivery Performance with DevOps

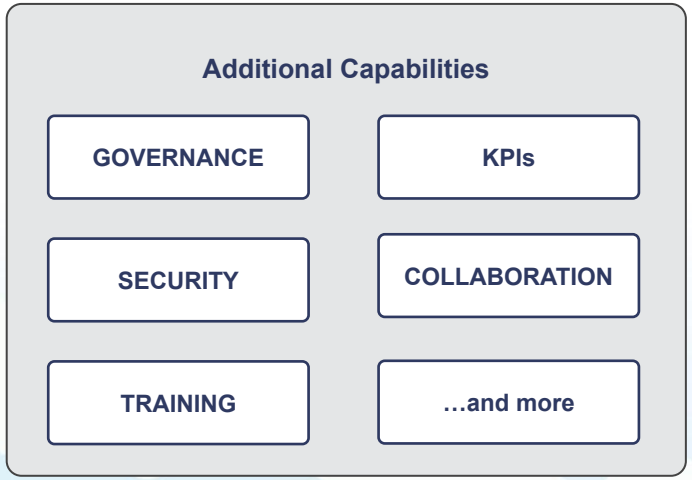
DevOps – A Visual Definition

What it does…



How it does it…





DevOps is a journey

Common customer journeys

Level 1
CRAWL

Establish DevOps Services and build **foundations** of **collaboration** to integrate and test code early, often & fast

- Establish **DevOps Services**
- **Single "source of truth"** for storing, versioning and tracking code
- **Integrate code early and often** (i.e. hourly, etc.)
- **(Automated) workflows to code reviews** to identify and fix issues early
- **Build test automation** in code

Level 2
WALK

Laser focus on **quality, test automation** and **automated test/deploy** - reduce and eliminate manual steps!

- **Automate build, test, deploy** (e.g. code merge triggers code integrate/tests, etc.)
- **Start "test driven development"** approach in small, select areas by writing tests before **code**
- **Start "selective tests"** approach in small, select areas
- **80%+ automated test coverage in code** with limited reliance on "human intervention" (e.g. functional, performance, security)

Level 3
RUN

Towards CD with **one-click automated releases** to customers

- **Implement "one-click" automated releases** to customers - 5 to 60 mins
- **Automatic provisioning** of **production-like** environments
- **Adoption of "test driven development"**
- **Adoption of "package driven development"**

DX

Level 4
FLY

Multiple releases in a day

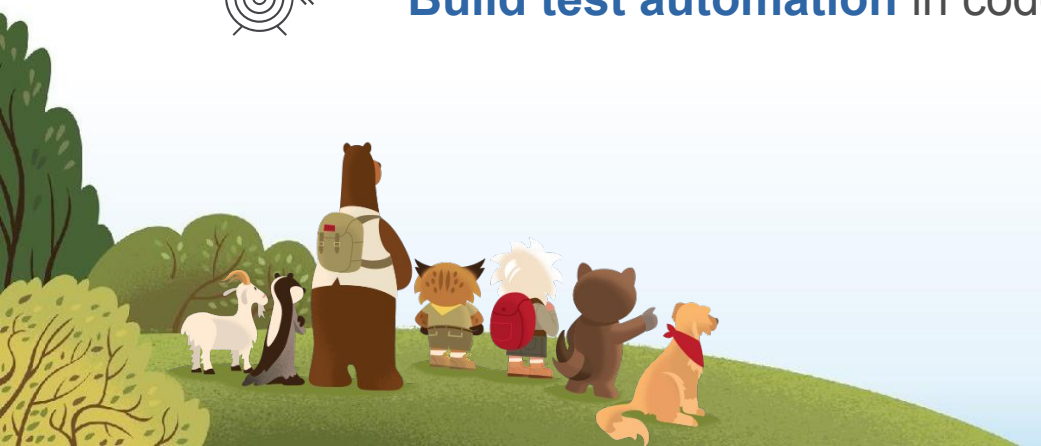
- **Multiple releases daily - in minutes.** with flexibility to target user segments (i.e. A/B testing)
- **Self-service access to pre-configured** environments

Level 1
CRAWL

Foundations of Collaboration

Establish DevOps Services and build **foundations** of **collaboration** to integrate and test code early, often & fast

- **Establish DevOps Services** and define ways of working
- **Single "source of truth"** for storing, versioning and tracking code
- **Integrate code early and often** (i.e. hourly, etc.)
- **(Automated) workflows to code reviews** to identify and fix issues early
- **Build test automation** in code



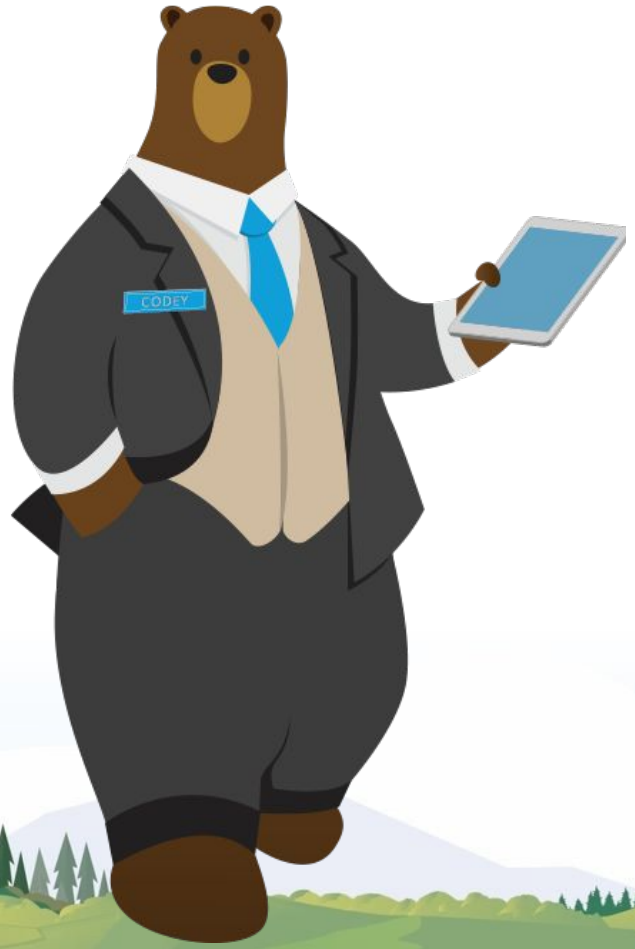
Foundations of Collaboration: DevOps Services

Establish DevOps Services and create some ways of working

- DevOps Services catalogue: listing of services and tasks you provide or perform
- Traceability by linking user stories to a change/version
- Project Documentation with a release runsheet
- Release Calendar with a standard release cadence
- Developer Guide with coding standards
- Environment Map with path to production design

Foundations of Collaboration: Release Runsheet

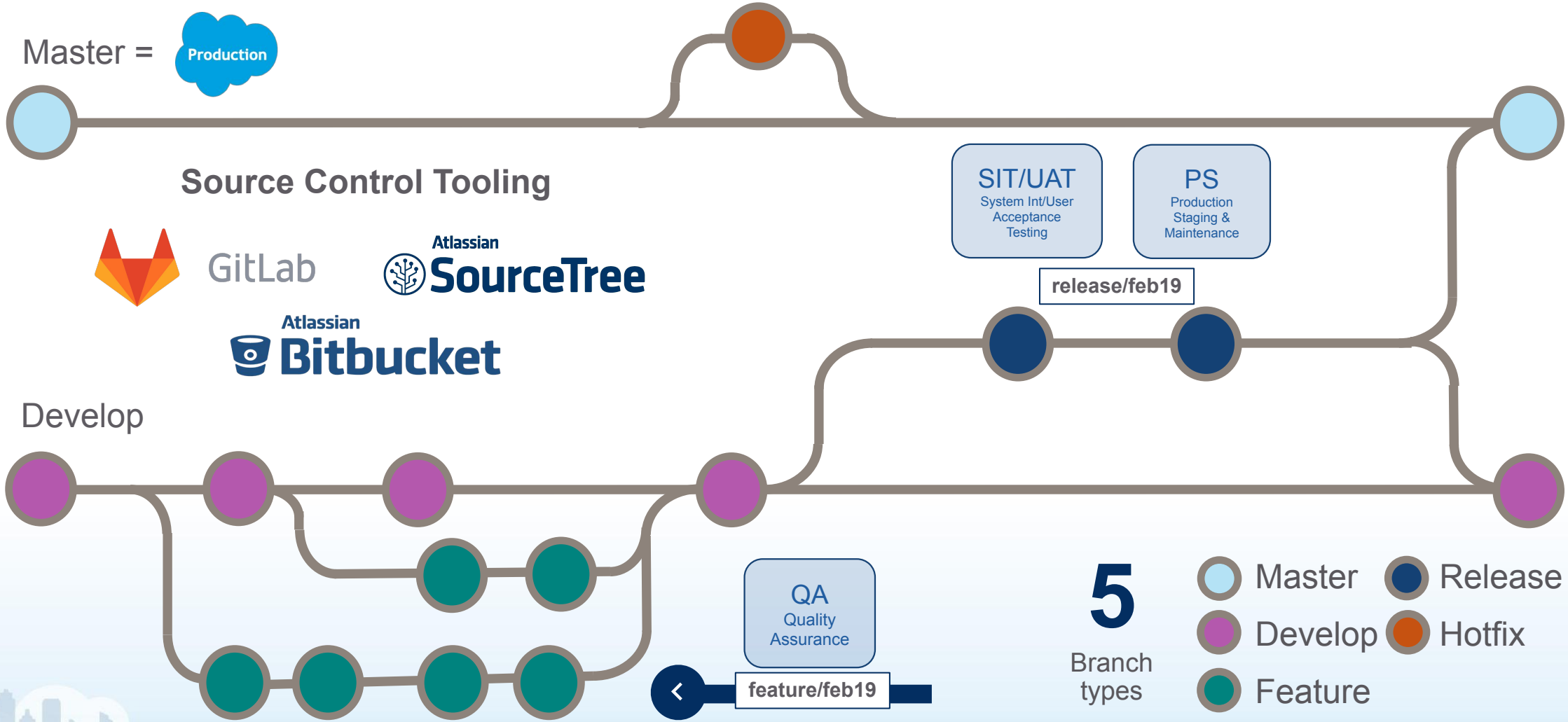
A step by step recipe to successfully deploy your project to another sandbox/production



- ➔ Project Name/s & Date/s
- ➔ Pre-deployment manual steps
- ➔ Deployment via change set or CICD
- ➔ Post deployment manual steps
- ➔ Smoke testing or Post Verification Testing

Foundations of Collaboration: Source Control

Single “source of truth” for storing, versioning and tracking code with Gitflow



Foundations of Collaboration: CICD

Integrate code early and often with Continuous Integration Continuous Delivery

What's right for me?

- ➔ Technology already in use by organisation
- ➔ Cyber compliance
- ➔ Skills in team
- ➔ Budget

AppExchange (Click) solutions

AutoRABIT Salesforce Release Automation

OR

Gearset

OR

copado

- Deploy data, users, profiles, permission sets, custom settings values, translations. flows
- ALM integration
- Version Control and Branch Management
- Compare and Deploy Org Metadata Differences
- Automated Metadata backups
- Automated Apex testing
- Webhook API and CLI
- Static Code Analysis
- Invoke Automated Testing

Custom (Code) solutions

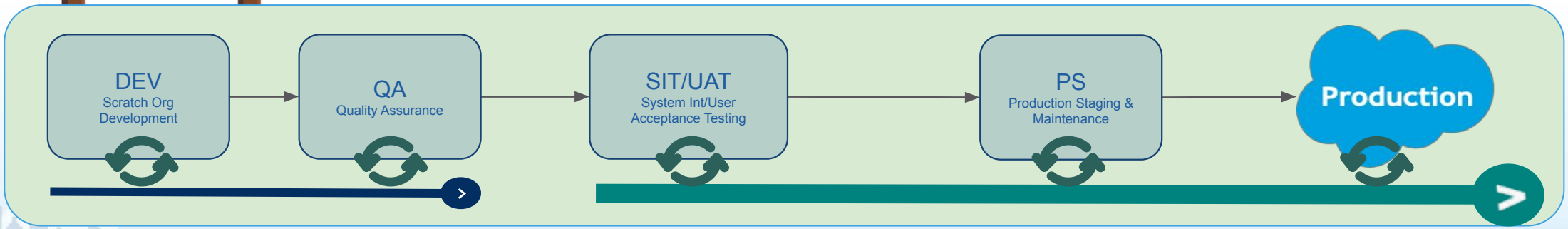
amazon web services EC2 | Microsoft Azure

+

Jenkins | Bamboo | Buildkite

+

{/code.scan} Code Analysis for Salesforce | CHECKMARX | PMD



Q&A

Stefan Scheit

Senior Program Architect

Adam Best

Senior Program Architect

Please ask your questions in the *APAC Success Central* Trailblazer Community

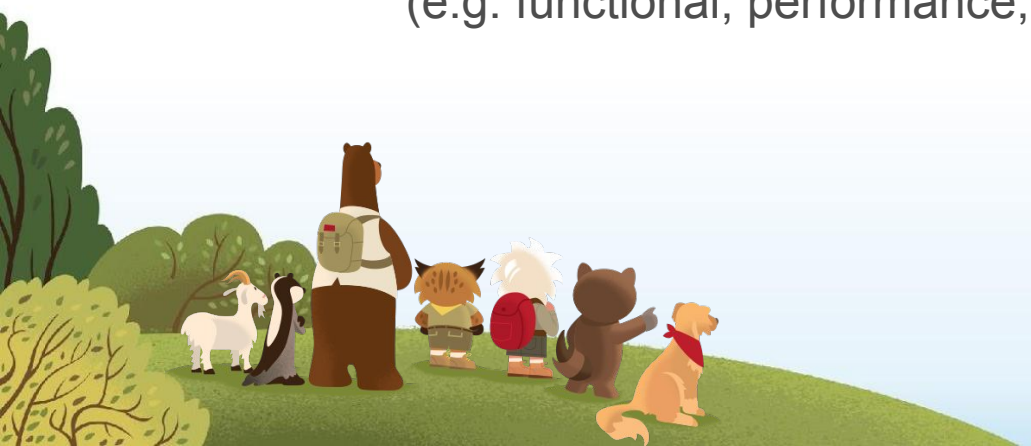
Level 2
WALK

Release Optimisation

Laser focus on **quality, test automation** and **automated test/deploy** - reduce and eliminate manual steps!

- **Automate build, test, deploy** (e.g. code merge triggers build/validation and tests etc.)
- **Start “test driven development”** approach in small, select areas by writing tests before **code**
- **Start “selective tests” deployment** approach for frequently changing components
- **80%+ automated test coverage in code** with limited reliance on “human intervention”

  (e.g. functional, performance, security)



Release Optimisation: Reduce Manual Steps

Automate build, test, deploy (e.g. code merge triggers code integrate/tests, etc.)

QA Gate

Package deployments, Apex unit tests, Automated Testing

SIT Gate

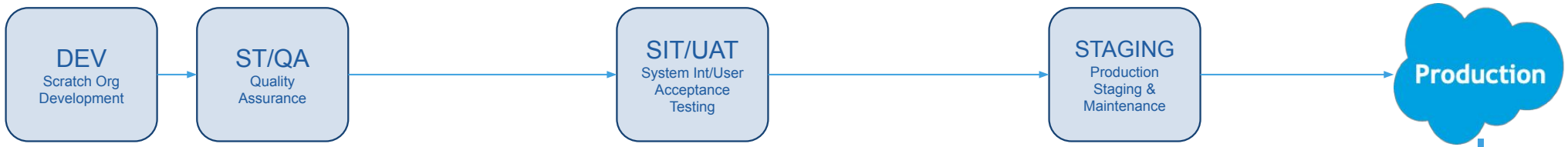
Package deployments, Apex unit tests, Automated Testing

STG Gate

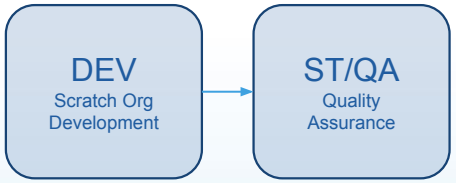
Package deployments, Apex unit tests, Automated Testing

PROD

Package deployments, Apex unit tests, Sanity Testing



Build Team 1



Build Team 2

Sandbox Refresh

Lock sandbox talking to production, data loads etc.

Prod Sync

Backup changes from production, massage metadata & commit to version control repository, create pull requests to all build team feature branches

Release Optimisation: Automated Testing

80%+ automated test coverage in code (e.g. functional, performance, security)

A conversation at the feature conception

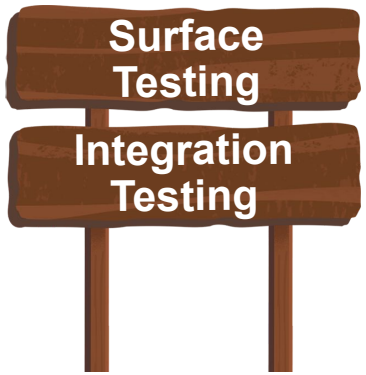
Automated testing should be baked into every project and created either in the UAT cycle of the feature creation or at the very latest in the warranty period of that feature.

What should you test?

- ➔ Top 5-10 Critical Paths
- ➔ For each user group

5 + 5 + 5 =

15 Test Cases



Popular Tooling

solutions

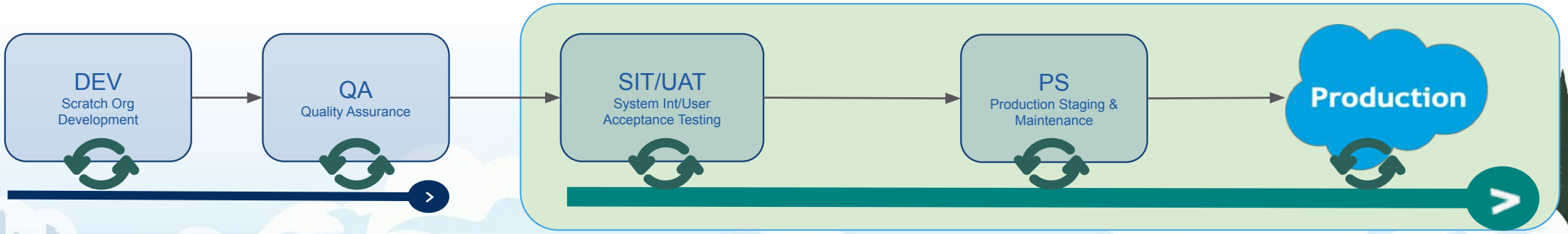
Se + Robot Framework

TRICENTIS Tosca — Service Virtualization

What's right for me?

considerations

- ➔ Technology already in use
- ➔ Cyber compliance
- ➔ Compatibility with CICD Tool
- ➔ Skills in team
- ➔ Budget



Release Optimisation: Selective Tests

Start “selective tests” deployment approach in small, select areas

4 hours

Run All/Local Tests
(per 1,000 unit tests)

- 1 line of code change
- Requires 75%+ Org wide code coverage
- Allows for developers to deploy an Apex class with 0% code coverage

5 mins

Run Selective Tests

- 1 line of code change
- Requires 75%+ code coverage per Apex class
- Does not allow developers to deploy an Apex class with less than 75% code coverage

Level 3
RUN

Release Acceleration

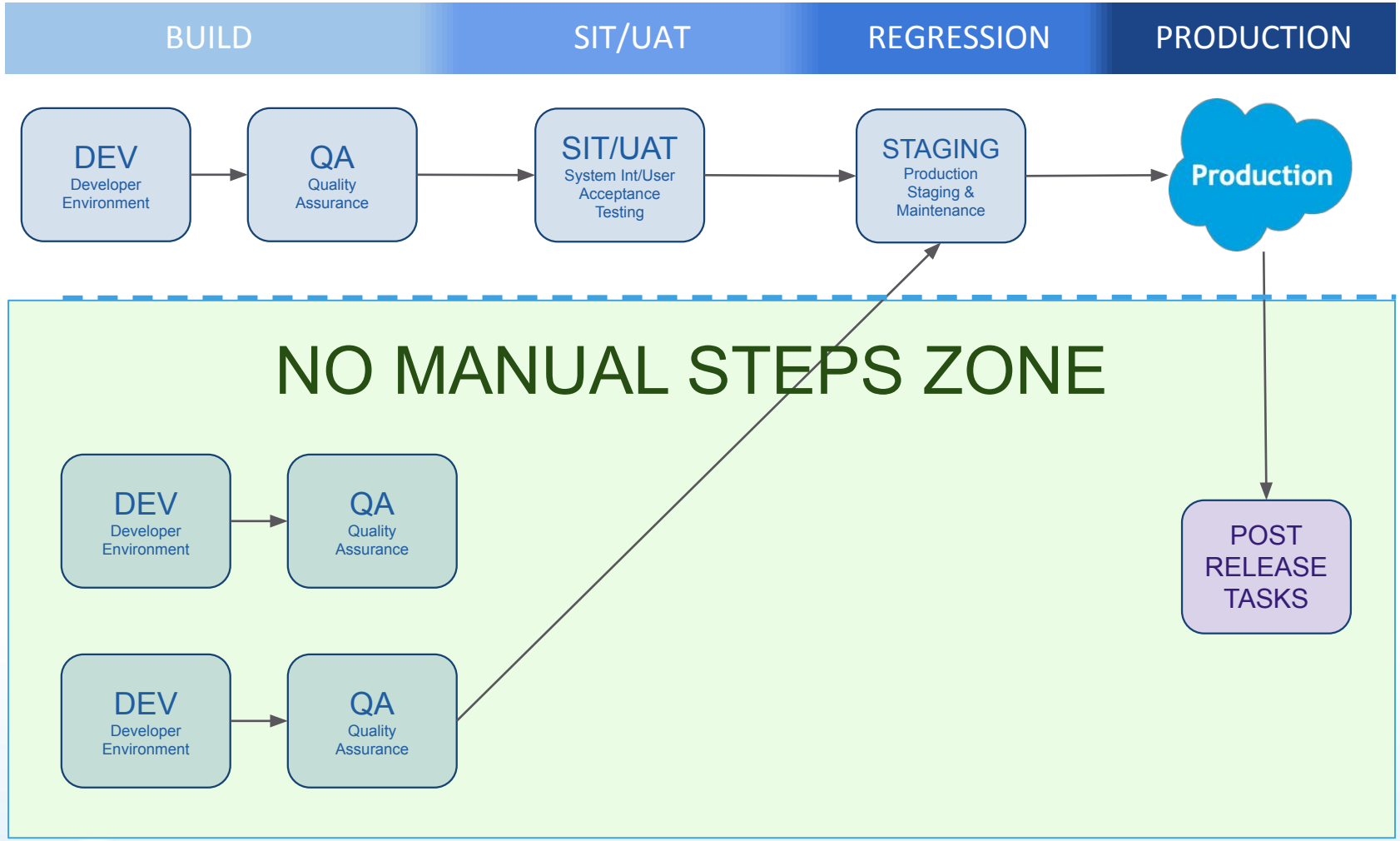
Start to achieve CD w. **one-click automated releases** to customers

- **Implement “one-click” automated releases** to customers
- **Automatic provisioning** of **production-like** environments
- **Adoption of “test driven development”**
- **Adoption of “package driven development”**

DX

One Click Deployments

Implement “one-click” automated releases to customers



QUALITY CHECKS

- ★ Package Compile
- ★ Static Code Analysis
- ★ Apex Unit Test Passing
- ★ Selective Test Apex Unit Testing Coverage 75%+
- ★ Auto Regression Testing Pass

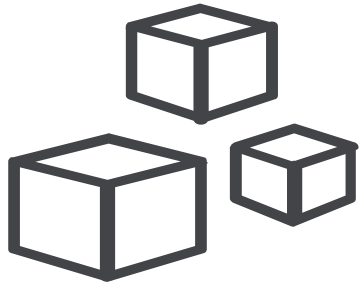
Source: placeholder

Adoption of “package driven development”

Existing org enablement

Moving from a single repository source control delivery to a DX package driven development will require what’s essentially an ***org migration*** of your components into a vanilla version of Salesforce called a ***scratch org***.

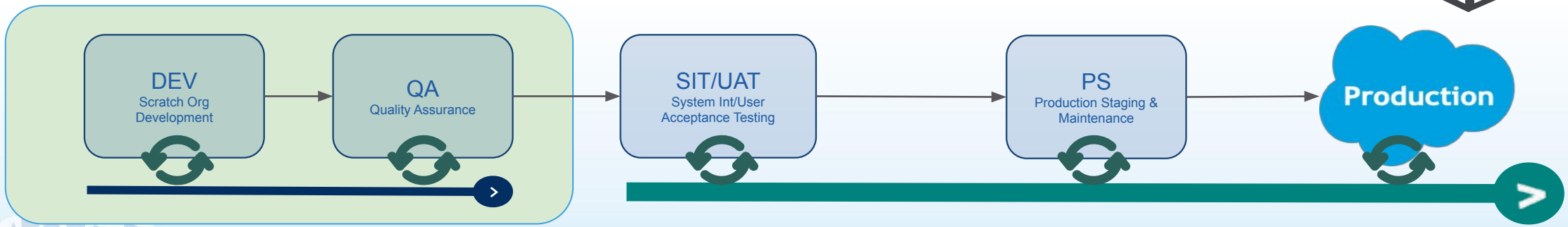
Splitting this up into meaningful packages, achieving a successful compile and testing this is working as designed **will take time**.



Splitting the existing org components up into meaningful packages is ***recommended***

OR

Put all existing org components in 1 package



State of the Art

Tools for Salesforce

PLAN | DESIGN | BUILD | DEPLOY | TEST | RELEASE | OPERATE

BUILD: Development | Source Control | Code Review | Code Quality | Continuous Integration

PLAN

JIRA

With **Jira**

- ★ Epics
- ★ Stories
- ★ Tasks
- ★ Bugs

DESIGN

Confluence

With **Quip** or **Confluence**

- ★ Documentation
- ★ Release Runsheet
- ★ Knowledge Sharing

BUILD: Development

DX

salesforce

Code written and committed locally in **Visual Studio Code** or similar IDE

Developer builds features, runs unit tests using **Salesforce DX Scratch Org** or **Sandbox**

BUILD: Source Control

GitHub

Bitbucket

Source Code Management using **BitBucket Server**

Version Control PC / MAC Software using **SourceTree**

BUILD: Code Review

GitHub

Bitbucket

Peer review by developers using **BitBucket Server**

BUILD: Code Quality

{/code.scan} Code Analysis for Salesforce

CHECKMARX

PMD DON'T SHOOT THE MESSENGER

PMD, Checkmarx or **CodeScan** is used for static code analyse and report code quality

BUILD: Continuous Integration

Bitbucket PIPELINES

Jenkins

Integration of feature & release branches and triggering of deployments using **Jenkins** or **Bitbucket Pipelines**

DEPLOY

Bitbucket PIPELINES

Jenkins

Deployments to all Salesforce Environments using **Jenkins** or **Bitbucket Pipelines**

TEST

odaseva

Load and mask production data to Partial or DevPro Sandboxes using **Odaseva**

Se

Katalon

Regression testing to validate deployment using **Selenium / katalon**

RELEASE

Bitbucket PIPELINES

Jenkins

Release to Production Salesforce Environment using **Jenkins**

OPERATE

odaseva

Backup and restore data using **Odaseva**

Jenkins

Operational Activities (provisioning sandboxes etc) using **Jenkins**

splunk>

Monitoring security events, anomalies using **Splunk**

Key Takeaways

What we believe is the secret to become sustainably successful

- People & collaboration are essential:
  Release management & governance, planning, and prioritisation

- Truly integrate your (agile) ways of planning and working with how you release

- Treat your CICD pipeline like a product:
  Your customers are often some of the brightest minds

Q&A

Stefan Scheit

Senior Program Architect

Adam Best

Senior Program Architect

Where Can You Go Next To Learn More

Trailhead, Chatter, Contact

Visit the following Trailheads to learn more about DX

- Application Lifecycle Management
- App Development with Salesforce DX
- Continuous Integration Using Salesforce DX
- Salesforce DX Development Model

Check out the following resources

- Salesforce Architects - Transforming Business Through Strategic and Technical Guidance Ebook
- Creating Agile and Innovative Business through an Optimised Security Governance Strategy - Blog

Contact your Salesforce Success Manager or Account Executive (AE)

salesforce success cloud


thank you