Public

SAP's DevOps Journey: from building an app to building a cloud

Darren Hague, SAP
June 30, 2016

SAP

DEVOPS ENTERPRISE SUMMIT

About SAP

World leader in enterprise applications

- Founded in 1972
- Vision: **Help the world run better**
- Innovation focus: **Cloud & In-memory computing, IoT & digital business**

- Over 310,000 customers, 190 countries
- Over 78,000 employees in 130+ countries (~50% in technology-related roles)
- Over 110 million users of cloud software

SAP's customers produce 78% of the world's food & 82% of the world's medical devices

76% of the world's transaction revenue touches an SAP system

My background

2008 – 2014: Web app architect, Global IT

Team responsible for internal- & external-facing web apps, e.g. SAP Developer Network portal

2014 – now: Cloud platform architect, Products & Innovation

Team responsible for delivering Cloud & DevOps platform for SAP's internal & external cloud portfolio to run on

Continuous Delivery at SAP: From dinosaur to spaceship

Darren Hague / SAP Global IT
November 1st, 2013

SAP

FLOWCON
CONTINUOUS DESIGN DELIVERY DATA

The old days (up to 2010)

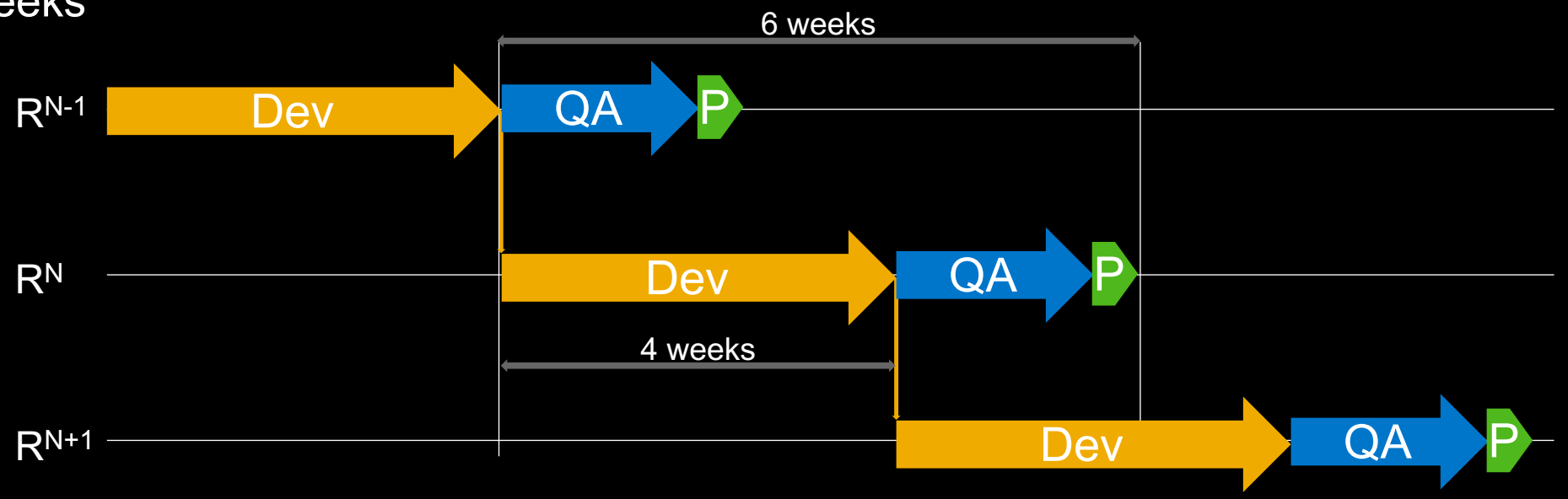
- **Good things**
  - Source code version control
  - Issue tracking
  - Build automation
  - Monthly releases

- **Not so good things**
  - Months-long lead time for new hardware
  - Labour-intensive QA cycle
  - Code deployed manually to physical hardware during downtime
  - Development, Ops & Infrastructure in different business units
  - No version control for configuration data

2010 Development & Release Cycle (in theory…)

4-week cycle:

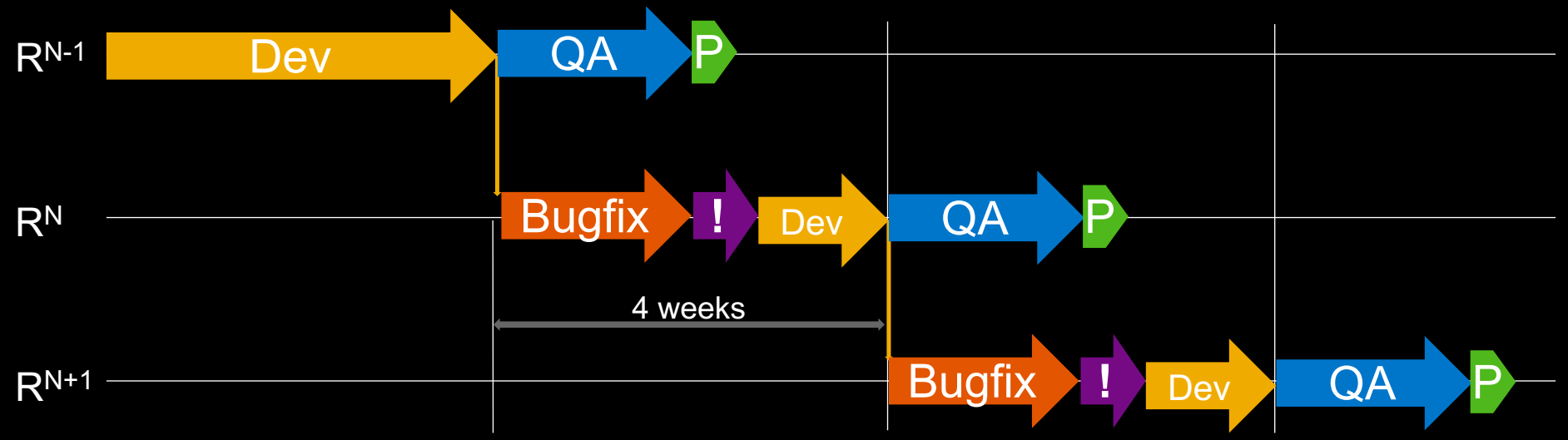
- 4 weeks of development per iteration
- 2 weeks of testing
- 1 weekend to deploy release to production
- 6 weeks from request to delivery
- Release every 4 weeks



2010 Development & Release Cycle (in practice)

4-week cycle:

- 4 weeks of development for the first iteration
- 2 weeks of testing & bugfixing
- 1 day to deploy release to production
- 2 days (and one night) debugging the production deployment
- **1.5 weeks of development per 4-week cycle**



Why DevOps?

2010 capacity in IT's web team

- 20000 PD of effort available (~100 staff)

2010 demand for IT's web team

- Estimated 60000 PD of project effort

<u>No budget for another 200 staff</u>

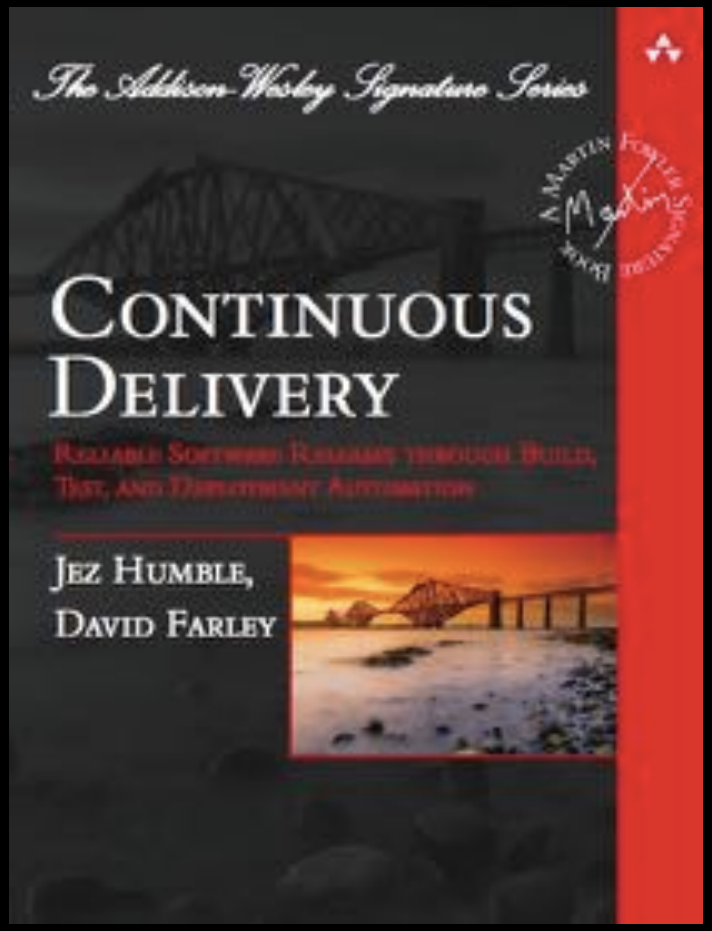
It's about the people: Vision & Cultural Change

Chief Architect (my boss)

- Promoted concepts of Continuous Delivery
  - Automate everything, especially testing & deployment
  - Version control everything
- Bonus-relevant objective: "Read the book"
- Start with one pilot project: SAP ID Service

Director of IT's Web unit (his boss)

- Provided trust
  - Agreed 10% of overall effort for a continuous delivery programme
- Provided cover
  - The 10% was "taxed" from individual project budgets
  - Not listed as an explicit line item



Pilot project: SAP ID Service

Unified SAP web experience

- Solved problem: one SAP, many sites & logins
- Single account for SAP web users
- Seamless sign-on to all SAP sites
- Social sign-on and integration with 3rd party apps

Scale & reliability (eventually, not part of MVP)

- Millions of users
- Needs to stay up, or nobody can log in to SAP sites
- SAP Cloud Identity product:
  - Needs to stay up, or customers can’t access their cloud software

It's about the people: SAP ID Service Project Team

Cross-functional team of 12

- Product Owner
- Scrum Master
- UI / UX designers
- Java developers & architects (**Dev**)
- Infrastructure engineers (**Ops**)
- QA specialists

Geographically distributed

- Germany
- Bulgaria
- UK
- Russia
- Israel

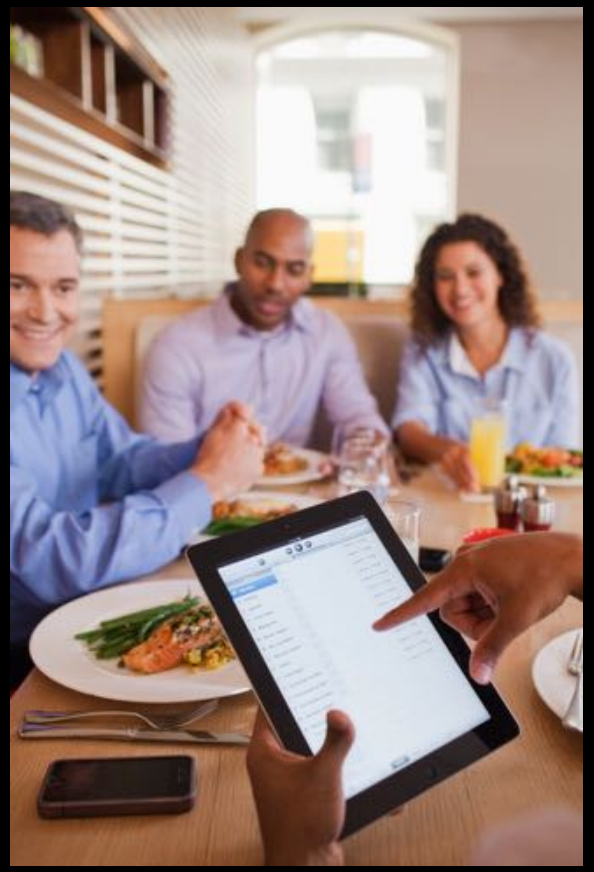
Job 1: Test coverage

Cucumber: Behavior-driven Testing

- Product owner works with team
- User stories from backlog are written in Gherkin:

```
@UserStory("SAPID-1522")
Scenario: Log on a user trying to access their profile

  Given I am a registered user

  When I try to access my SAP ID Service profile
  Then I should see the "SAP ID Service" login overlay

  When I login using my valid credentials
  Then I am logged in
  And the user profile page is displayed
```

- Written by QA & Product Owner pairing together
- Main purpose is to communicate system behaviour



Gherkin lines correspond to Java or Ruby methods

```
@UserStory("SAPID-1522")
Scenario: Log on a user trying to access their profile

  Given I am a registered user

  When I try to access my SAP ID Service profile
  Then I should see the "SAP ID Service" login overlay

  When I login using my valid credentials
  Then I am logged in
  And the user profile page is displayed
```

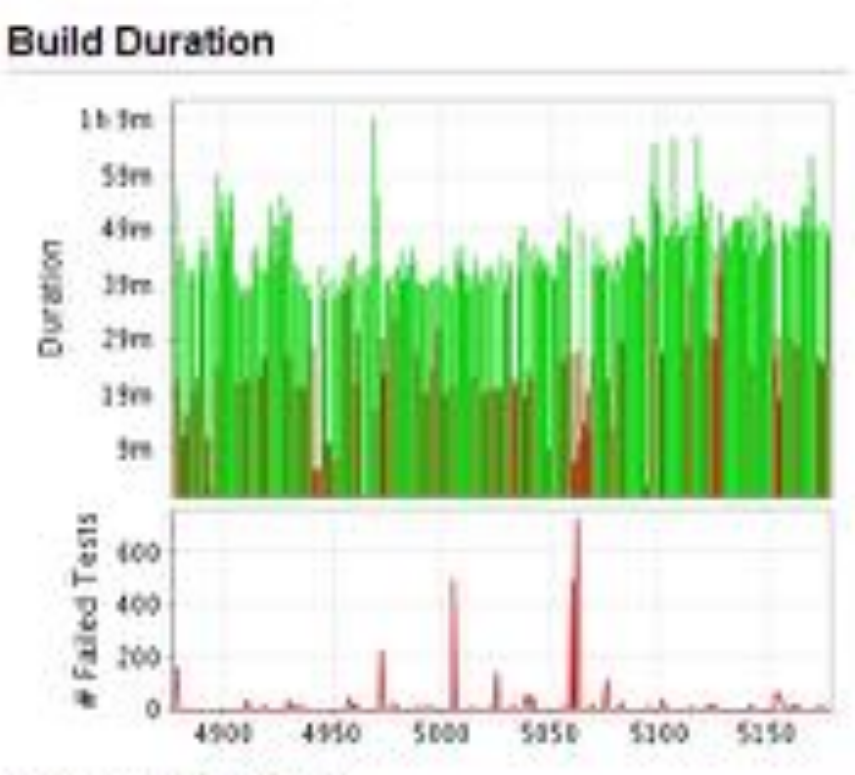
- QA & Developers work together to code the test logic

```
@When("^I login using my valid credentials$")
public void login_using_valid_credentials() {

    String loginName = getTestUserProfile().get(USER_PROFILE_ID);

    String password = getTestUserProfile().get(USER_PROFILE_PASSWORD);

    ((LoginPage) getWebPage()).login(loginName, password);
}
```

- This gives us an **executable specification**

Cycle time is critical

- **Minimise the time from commit to green build**
- **Continuously monitor & improve build performance**
  - < 15 minutes for developer build and test on own machine
  - < 40 minutes for full build, integration test & deploy to QA
- **Parallelisation is key, especially for tests**
  - We have >1000 scenarios and >10000 steps
  - Aim to keep each suite of tests to < 3 minutes
  - If a suite exceeds this, split it into 2 suites
  - Add test machines as number of suites grows
  - The Cloud is your friend: elastically scale the QA landscape



It's about the people

- **Challenge**:
  Getting developers to write lots of automated tests with no previous culture of test-driven development
- Focus on the **Happy Path** – test cases that demonstrate how the system should work
- Developers also write some low-level tests for corner cases etc.
- QA folks still do manual exploratory testing
  - Found a bug? Write a test to reproduce it
- Small amounts of ~~shouting~~ encouragement may be involved

  All tests pass = **confidence to release**

Job 2: Automate deployment

Using Chef for Deployment Automation

Code deployed & configured automatically

- No manual scripts

Recipes & config data stored in version control

- Can review before deployment

Idempotence of recipes = consistency

- Run chef-client again: get the same result

Blue/Green deployment

- Deploy latest release to alternate pool of servers

Testing: Confidence to release a version

Chef: Confidence to deploy it automatically

It’s about the people

DevOps skills needed

- All team members need to learn Chef & Ruby
- Ops & Developers pair up to write Chef recipes
  - “install DB”, “install appserver”, “deploy latest release”, etc.
- Learn how to store config values in a source-code repository
  - Data bags for Dev, Test, Production, …
- Put in the effort to migrate all those manual scripts and Excel sheets
- Learn how to unit test recipes and cookbooks

Job 3: Keep evolving

DevOps 2010: Cocktail – Project Scale

To get Identity Service app running we had to manually:

- Create virtual machines
- Register each VM with Chef server
- Execute chef-client
- Validate the installation
- Test behaviour with Cucumber

"Cocktail" was developed to automate all these actions

- With a pool of VMs, create a complex landscape with a few commands
- Deployment time reduced from hours to minutes
- **Monthly deployment**, much closer to "4 weeks development" vision

It's about the people: Culture of Continuous Improvement

- **Proper Agile/Scrum coaching was really important**
  - Fundamentally different to learning by reading articles & blogs
  - Learning by doing, playing, etc. is much more effective
  - Coach encouraged accountability, responsibility, experimentation
  - Team make-up
    - Differing personalities and working styles
    - May need to shuffle a few people to get the best out of everyone
- **Team is always working to improve itself**
  - Evaluate new tools & techniques
  - Encourage "spikes" to see if things work
- **Retrospective at the end of each sprint**
  - Several improvement suggestions each time around – pick top 3
  - Focus on team behaviours, not product scope

DevOps 2011: Barkeeper – Department scale

- Allocate VMs via Cloud API
- Manage Chef servers
- Project self-service
- Web UI and REST API
- Everything under version control
- Install VMWare on bare metal, everything else is automated

- <u>Scheduled deployment every 2 weeks, but 2-3 times a week not uncommon</u>

It's about the people: Spread the Word

- Spread DevOps principles throughout several IT teams
- Barkeeper used for a dozen or so web apps in SAP IT
- Agile/Scrum coaching delivered throughout IT
- Ops Engineers embedded in teams became the norm
- Encourage shared ownership within teams
- Encourage teams to share what they learn

Pivot: Create a DevOps platform for the company. Using DevOps.

DevOps 2013: Monsoon – Company scale

The product

- Custom developed private cloud & deployment automation platform
  - IaaS layer roughly equivalent to OpenStack (Nova, Cinder, Keystone, Murano, Designate)
  - Automation framework using Chef & MCollective

The platform

- Ruby on Rails components running as microservices
  - VCR to help with microservices testing
- Git repos for Chef cookbooks and configuration databags
- VMWare vSphere : Hypervisor & Block Storage
- F5 BigIP Loadbalancers (providing self service ELB scenarios)

The process

- 2 Week development cycle, continuous integration and testing
- **Daily automated deployment**

Result: Cloud & DevOps culture throughout SAP

As of late 2015:

- **100s of internal apps and external cloud services running on the Monsoon platform:**
  SAP Anywhere, SAP Business One, Multiposting,
  SuccessFactors, Afaria MobileSecure, SeeWhy,
  Ariba TradeWorld, Hybris, FieldGlass,
  SAP Cloud for Customer, SAP Cloud Identity, …
- **1000s of developers using the Monsoon platform**
- **10000s of VMs**
- **10000s of storage volumes**
- **Deployed in 6 regions & 12 availability zones**

It's about the people: Monsoon team

The core team

- Ruby development & web design: 15-20 FTE
- Infrastructure Architecture: 2-3 FTE
- QA: 0.5 FTE

Working mode

- ChatOps via internal IRC server
- UK & Germany, mix of home & office working
- Pairing & team meetings conducted largely online
- Every 3-6 months get together & break bread

- First level support in overseas teams
  - Knowledge transfer from core team
  - Overnight cover for Europe
- Second level support in development team
  - You build it, you run it: MOOPS (Monsoon Ops)
  - One team member always on "pager duty"
  - Weekly rotation

Space age

To Boldly Go…

Pets and Cattle

- Pets are given names like pussinboots.cern.ch
- They are unique, lovingly hand raised and cared for
- When they get ill, you nurse them back to health

- Cattle are given numbers like vm0042.cern.ch
- They are almost identical to other cattle
- When they get ill, you get another one

via Gavin McCance (CERN), who borrowed it from @randybias at Cloudscaling

Future application architectures should use Cattle <u>but</u> Pets with strong configuration management are viable and still needed

Kubernetes: Production-Grade Container Orchestration

Originally from Google, now open source

From http://kubernetes.io/:

Automated container deployment, scaling, and management

- Automatic bin-packing
- Horizontal scaling
- Automated rollouts and rollbacks
- Storage orchestration
- Self-healing
- Service discovery and load balancing
- Secret and configuration management
- Batch execution

In other words: **the ability to treat containers as cattle, not pets**

DevOps 2016 – Cloud Scale

The platform

- Kubernetes (with Docker) running on baremetal nodes
- Openstack in containers (based on Kolla)
- Networking hardware (via OpenStack Neutron drivers)
- Loadbalancing (via Neutron LBaaS)
- Both VMWare & KVM Hypervisors
- Storage hardware (via OpenStack Cinder & Manila)
- New automation agent (Arc)
- New dashboard for OpenStack (Elektra)

DevOps 2016 – Cloud Scale

The product

- OpenStack running in containers on Kubernetes across 13 Regions & 18+ datacenters
- Bare metal to scaled OpenStack cluster in < 60 minutes
- Platform for SAP cloud offerings and internal innovation
- Offering Bare metal, VMs and Container resources
- Monitoring, Logging, Alerting & Billing services
- Automation agent (Arc) being open-sourced

The process

- 4 week sprint cycle, continuous integration and testing
- <10% new development, >90% open source enhancements
- Open source enhancements contributed back to community (OpenStack, Kubernetes, Docker, Grafana, …)
- <u>Continuous automated deployment of infrastructure & code</u>

Another book to read - but beware “web scale envy”

“Google’s SREs have done our industry an enormous service by writing up the principles and patterns – architectural and cultural – that enable their teams to combine continuous delivery with world-class reliability at ludicrous scale. You owe it to yourself and your organization to read this book and try out these ideas for yourself”

- Jez Humble, author of “Continuous Delivery”

“I heavily suggest to buy and read this book !!!”

- My boss

It’s about the people

The teams

- OpenStack core DevOps: 10-15 FTE
- Monitoring, Alerting & Billing: 5-10 FTE
- User-facing & automation tools: 5-10 FTE
- QA 0.5 FTE
- First level support in overseas teams
- Second level support in development teams

Managing Complexity

- 2x MOOPS, plus INFRAOPS & AUTOPS
- Cognitive overload from new technologies
  - Split into 3 sub-teams
  - Pairing within teams
  - Sharing sessions across teams

Learning from Open Source behaviours

- Use GitHub pull requests for sharing & review
- Just enough documentation, stored in GitHub repo
- Use public GitHub where possible (github.com/sapcc)
- On-premise GitHub Enterprise for private repos (e.g. config)
- Engage with & contribute to open source projects (pros & cons)

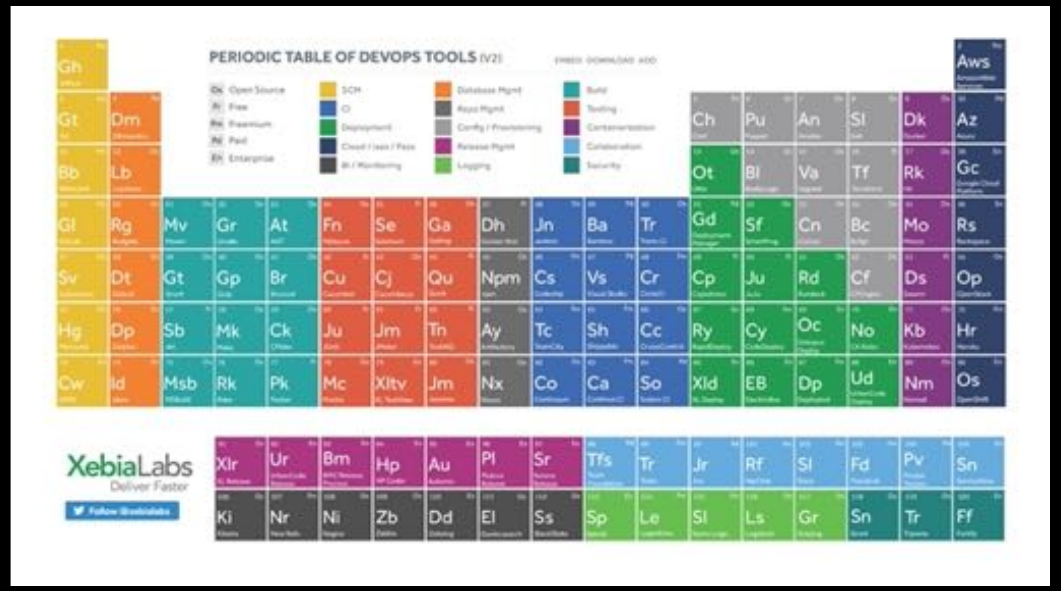
The journey continues

Then & now:

- 2010 – Pilot project, 10-15 people involved
- 2016 – Global Cloud & DevOps platform, 20-30 people involved

Current challenges:

- Dealing with technology explosion
  - See Periodic table of DevOps tools
- Dealing with knowledge explosion
  - Kubernetes, OpenStack, related tools, concepts & technologies
  - Need to spend time adding features & value in own area
  - Also need to know everyone else's stuff for Ops duty shifts
- Dealing with Open Source communities
  - Not always easy to get changes merged upstream
  - Stakeholders outside of the business may have different aims & priorities
- Getting corporate HTTP proxy to work with open-source tools



Thank you

(P.S. It's about the people)

Contact information:

Darren Hague
Cloud Infrastructure Architect
SAP UK Ltd

d.hague@sap.com

Twitter: @dhague