

Driving Change At Scale

April 27th, 2017

Anshu Verma and Soumendra Daas

Architects, Small Business Segment, Intuit

The Speakers



Soumendra Daas

**Chief Architect, Small Business
Segment**

Intuit India

@sdaas



Anshu Verma

**Chief Architect, Small Business
Segment**

Intuit India

@vermaanshu

Powering Prosperity Around the World



turbotax



quickbooks



proconnect



mint

1983

Founded

1993

IPO

8,000

Employees

21

Locations

\$5.2B
FY17

Revenue

46M

Customers



India's Best Companies
To Work For 2017

A STUDY BY

THE ECONOMIC TIMES

GREAT
PLACE
TO
WORK



Most Admired:
Software Industry

Ranked
#4

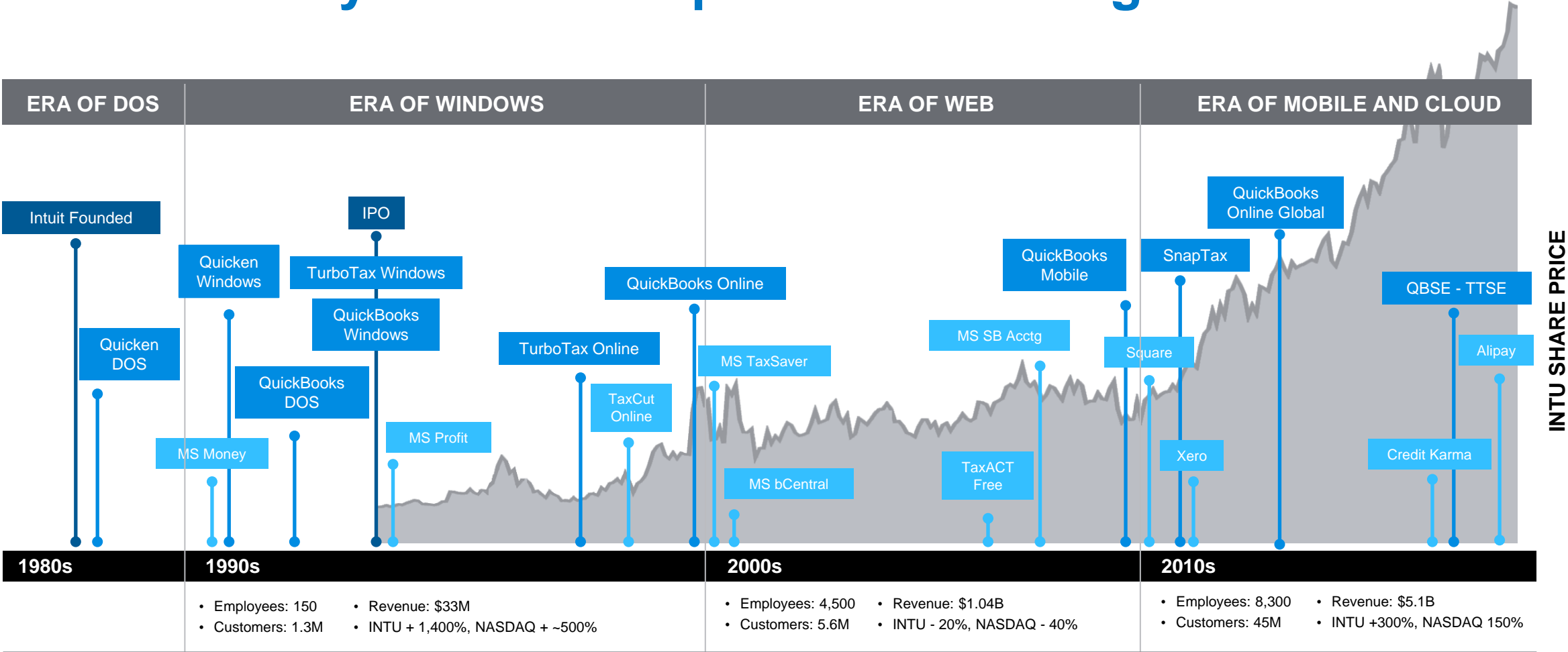
2004 - 2017

30+ Years of Growth Is Not So Easy

Only 8 software companies with
>\$4 billion in revenue are >30 years old



Intuit's history of self-disruption and re-imagination



CUSTOMER-OBSESSED • DESIGN-INSPIRED • TECHNOLOGY-POWERED

Transformation @ Scale

In the new world,

it is not the big fish
that eats the small
fish.

It's the **fast fish** that
eats the **slow fish**.

THEN



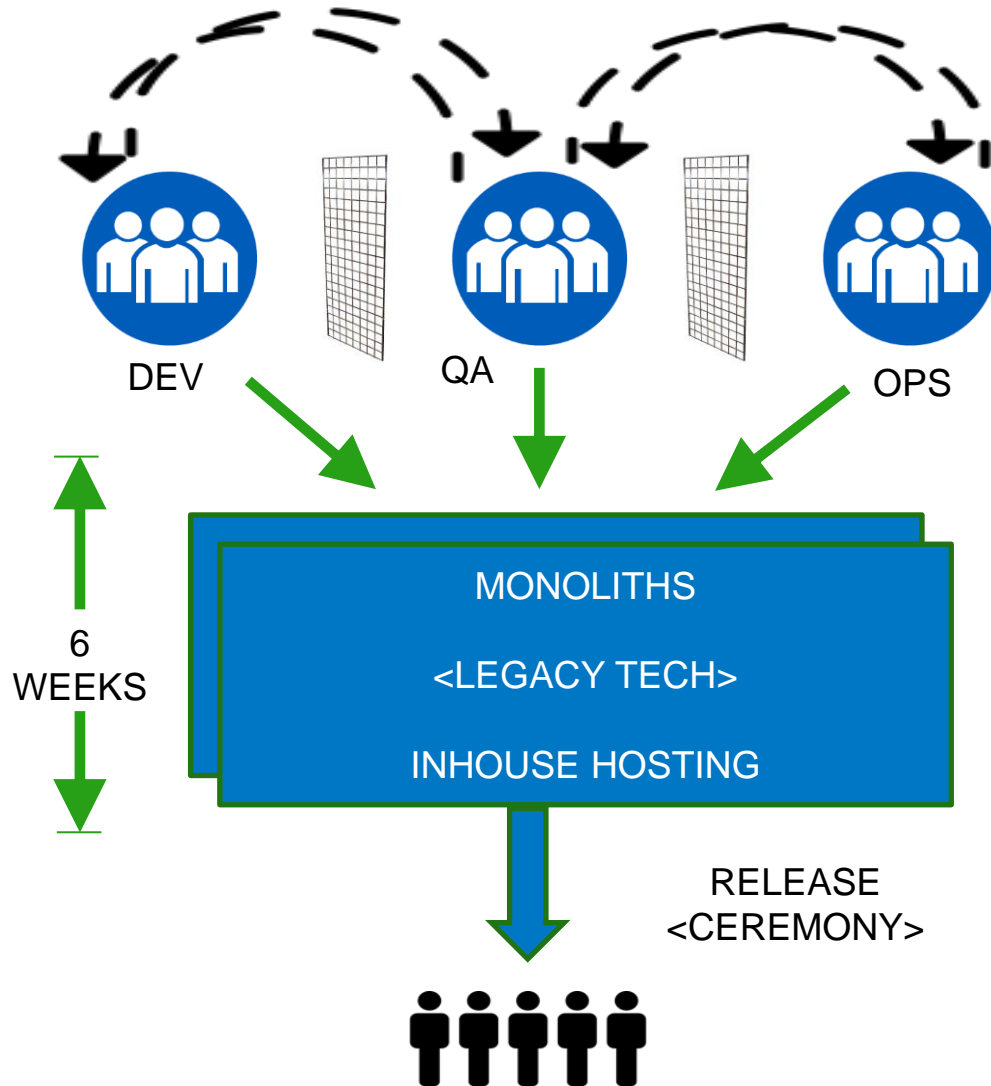
NOW



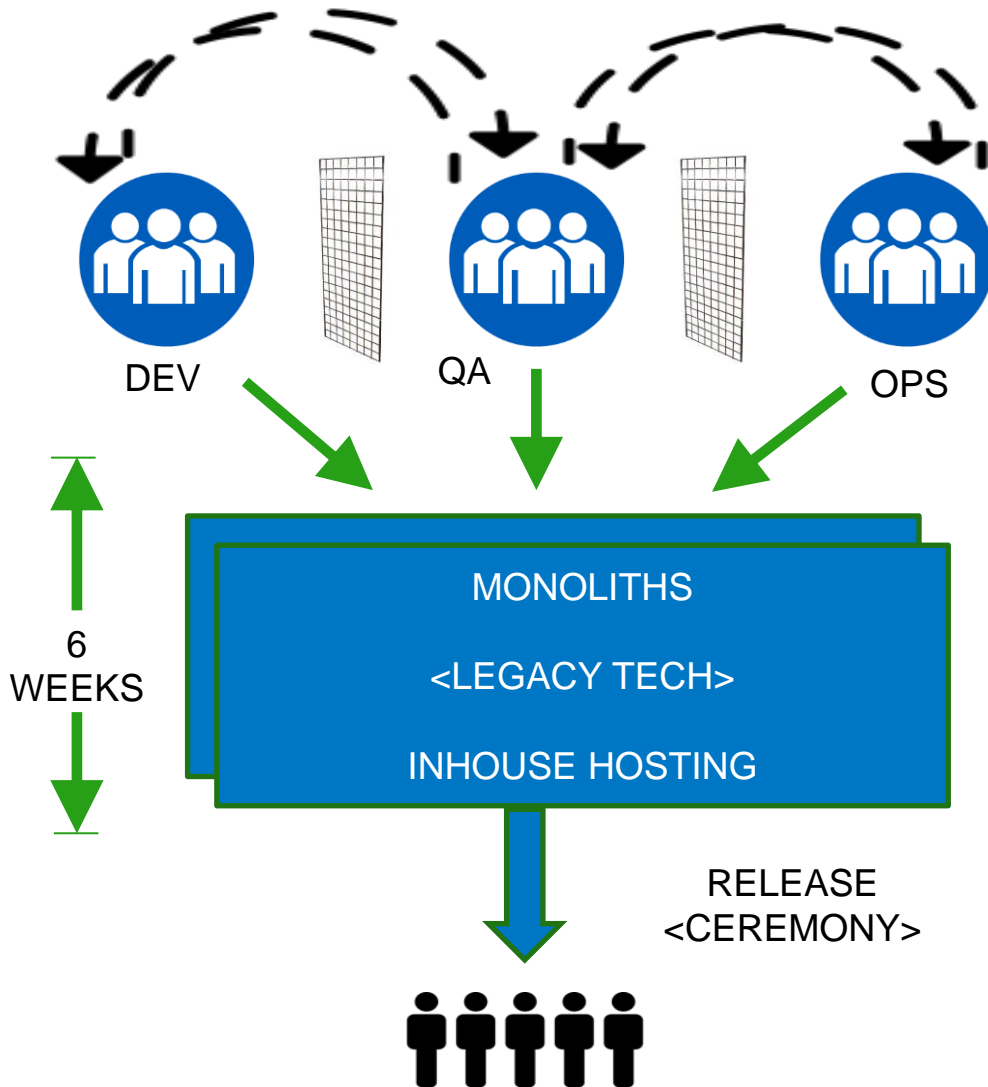
TORBEN RICK - WWW.TORBENRICK.EU



The need for change ...



The need for change ...

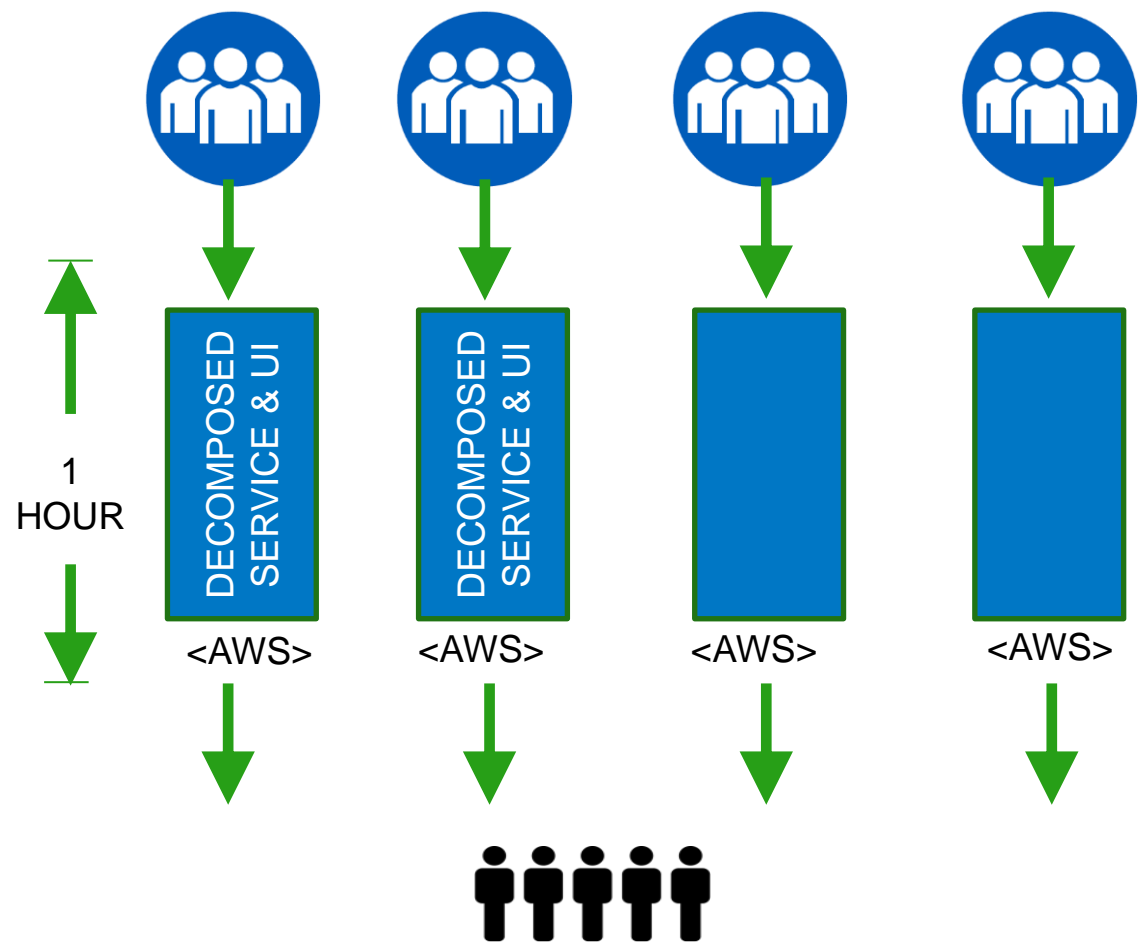


Slow to innovate

- **Monoliths: Long time to deliver benefits and features to customer**
- **Small change -> Regress everything**
- **Large "blast radius"**
- **Org moving in lockstep: Poor developer productivity**

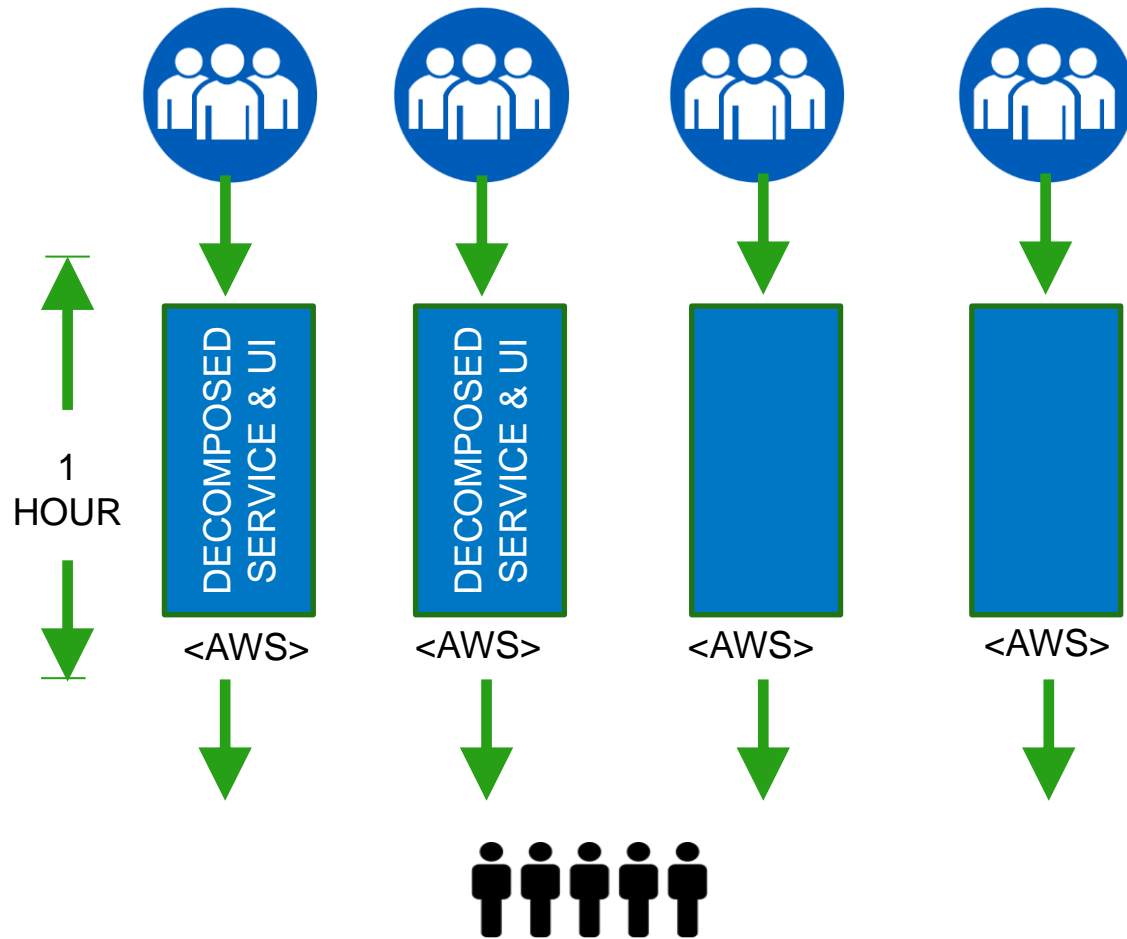
Where we wanted to be ...

SMALL AGILE SCRUM TEAMS



Where we wanted to be ...

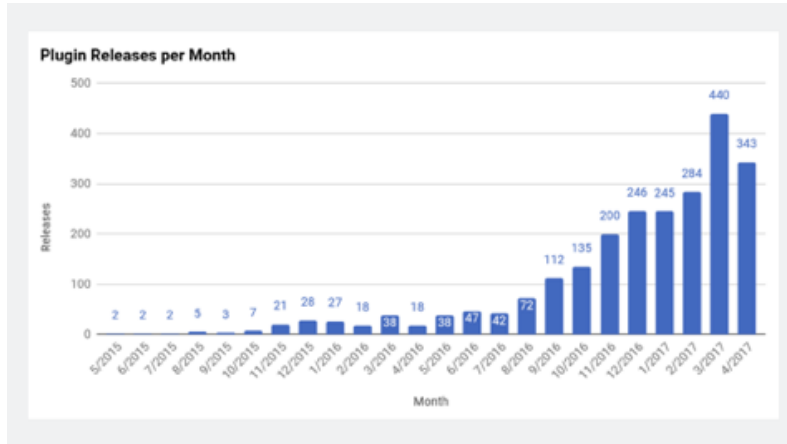
SMALL AGILE SCRUM TEAMS



Fast Pace of Innovation

- **Decomposed Services & UI in cloud – release in <1 hour**
- **Continuous Automation & Delivery**
- **Small parts fail; Fast recovery**
- **Autonomous team; Productive Modern tech**

Benefits ...

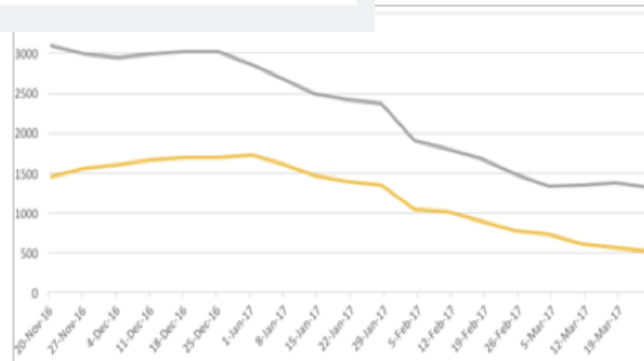


400x UI releases

+71 increase in developer NPS

80% reduction in defects

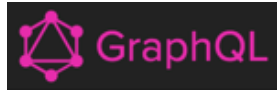
50% improvement in reliability



>50% online subs growth
+11 point increase in NPS.

Moving faster; Growing faster; Reduced Impact to customers; Better Product/Platform

Components of transformation



1. **Microservices**
2. **Modern Open Source Tech**
3. **Cloud Native**
4. **Mind-set and Engineering culture**
5. **Simplify**

Unpacking The Journey ...

Our Decision-Making Framework For Driving Change

WHY?

Why should we do this ? What are the consequences of NOT doing this.

WHAT?

Have a clear vision of what success looks like, including metrics.

HOW?

Establish clear principles for driving and executing this change.

WHO?

Business leaders, product managers, engineering – dev, quality, operations.

Why Move To The Cloud ?

Employee Benefit

Increased velocity, rapid experimentation and innovation, and faster delivery of features

Customer Benefit

Improved latency, resiliency, and availability

Shareholder Benefit

Reduction in capital expenses around hardware

Have clarity on *why* the change is needed ...

What Defines Success (Exemplars)

Customer Metrics

Increase Service Availability from 99.9% to 99.99%

TP95 variations for top N workflows < 3% across the globe

Reduce Unplanned Disaster Recovery from 20mins to < 5 minutes

Business Metrics

30% reduction in operational cost for elastic workloads

Hardware refresh in < 30 mins instead of 2 months.

Developer / Operations Metrics

Environment (including compute, n/w, and storage) provisioning in < 3 min

Define clearly what success looks like ...

Key Principles for ...

- **Readiness Assessment**
- **Architecting for The Cloud**
- **Managing the Product Lifecycle in the Cloud**

Establish Principles for Target State and Migration Process

Assessment Checklist : What does Current State look like ?

Assess Team Skills

- Cloud Architecture, Cloud Dev, DevOps, SysOps

Assess Current Architecture and Technology

- Hardware and OS specificity
- Software stack, including 3rd party libraries, databases, and message buses
- Data sources, including consistency, replication, retention
- Monitoring solutions

Baseline Existing System

- Performance and Availability : RTO, RPO, latencies, etc.
- SDLC metrics – time to develop, test, build, and deploy

Determine The Current State – Technology and Organization

Principles : Architecting for the cloud (Exemplars)

Exploit Provider Capabilities

Leverage IaaS and PaaS to accelerate automation, agility, and innovation

Scale Horizontally

Add new instances of components. Shard data across databases. Avoid monoliths

Keep the Compute Tier Stateless

Don't store state/data on the compute instances. Assume that they can "disappear" at any time

Don't Trust the Execution Environment

Assume that the compute, network, and storage are inherently unreliable and insecure.

CHALLENGES, CHALLENGES, CHALLENGES ...

Architecture Snowflake

Lack of standardized architectural patterns leads to “one-off” designs

Security

Security of data in transit and at rest, governance of management credentials etc.

Tooling and Automation

Can't achieve velocity in the cloud without the right tooling/automation in place.

Governance

Policy as code and mechanisms for enforcement

Establish clear strategies for Engineering Challenges

Build the Organizational Capabilities

Security Mindset

From: Perimeter defense driven by a central security team

To: Security by design, layered defenses, automated detection and remediation

Relentless Automation

From: Inconsistent and manual methods for build, testing, and deployment

To: Fully automated everything

Skills and Competencies

From: Siloed Dev, QE, Ops, and Security Team

To: Dev sys Ops

Solve the Most Difficult Problems First



The Cloud Enablement Lifecycle ...



Assess current state, plan, and deploy correct architectures that deliver the right customer experience

Capabilities to monitor, control, manage their environments. Includes CI/CD, security and and cost

Identify opportunities to optimize on usage and cost

Identify principles, architecture & code patterns, tools and services ...

Cloud Enablement : Principles for the MOVE Phase

Complexity Kills – Keep it Simple

Lack of experience and early optimization leads to over-engineering and complexity. Leverage simple architecture and best-practices from those who have gone before you. Avoid early optimization.

”Lift and Shift” vs. “Refactor in the Data Center”

Create a MVC (minimum viable cloud) application. Refactor in the cloud based on real data and learnings.

Security by Design

Prepare for the worst. How to respond to a breach or account-level compromise. Fully automated security controls for infrastructure, service, management that are always on.

Cloud Enablement : Principles for the RUN Phase

Automate Everything

Everything is programmable and declarative

Relentless automation – config, build, test, deployment, compliance, and security.

Maintain Visibility

Standardized monitoring, alerting, and automated remediation

High-Resolution, Full Stack Monitoring correlating biz SLA with app performance and availability

Defend the Management Plane

Federated access by workforce across the organization to AWS using short-term credentials that expire automatically.

Cloud Enablement : Principles for the OPTIMIZE Phase

Avoid Early Optimization for cost and/or efficiency

Use real-usage information to determine the right compute size and mix

For example:

- EC2 Reserved Instances can be 66% cheaper than On-Demand Instances
- EC2 Spot Instances can be 80% cheaper than On-Demand Instance

Favor auto-scaling to add/remove compute capacity instead of over-provisioning

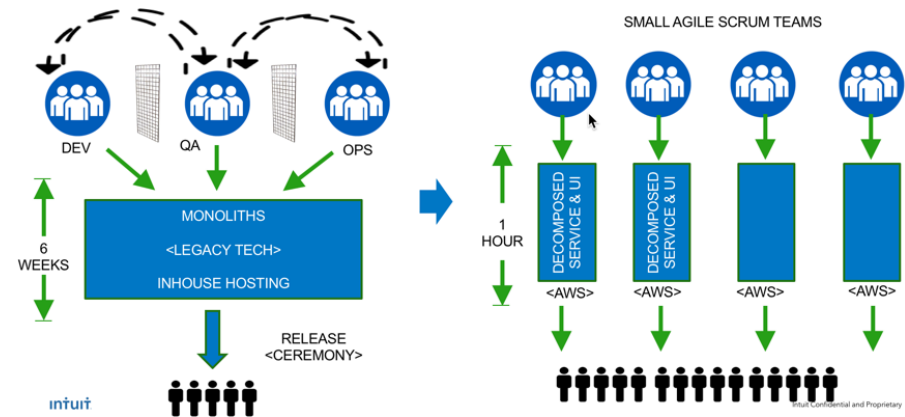
Deploy automated scans to detect unused/underused resources

Summarizing ...

Key Takeaways ...

- **Establish the need for change and success metrics**
- **Establish Target State and principles for moving towards target state**
- **Move towards target state in bite size chunks, while continuously delivering business value**
- **Establish patterns and processes that work across the organization**
- **Solve the most difficult problems first**

Transformation



Slow to innovate

- **Monoliths:** Long time to deliver benefits and features to customer
- **Small change; Regress everything**
- **Large "blast radius"**
- **Org moving in lockstep: Poor developer productivity**

Fast Pace of Innovation

- **Decomposed Services & UI in cloud – release in <1 hour**
- **Continuous Automation & Delivery**
- **Small parts fail; Fast recovery**
- **Autonomous team; Productive Modern tech**

In the new world,

it is not the big fish
that eats the small
fish.

It's the **fast fish** that
eats the **slow fish**.

THEN



NOW



TORBEN RICK - WWW.TORBENRICK.EU



Q&A

Your opportunity to ask
and learn

GREAT INDIAN **DEVELOPER**TM **SUMMIT** 2019

Conference : April 23-26, Bangalore



Register early and get the best discounts!



www.developersummit.com



[@greatindiandev](https://twitter.com/greatindiandev)



bit.ly/gidslinkedin



facebook.com/gids19



bit.ly/saltmarchyoutube



flickr.com/photos/saltmarch/