



Journey to APIs and Microservices: Best Practices

September 30, 2016

Deepak Nadig

Distinguished Architect, Intuit

@deepak_nadig





simplify the business of life



turbotax



quickbooks



proconnect



mint

Who we serve:

turbotax quickbooks proconnect mint

Consumers



Small Businesses

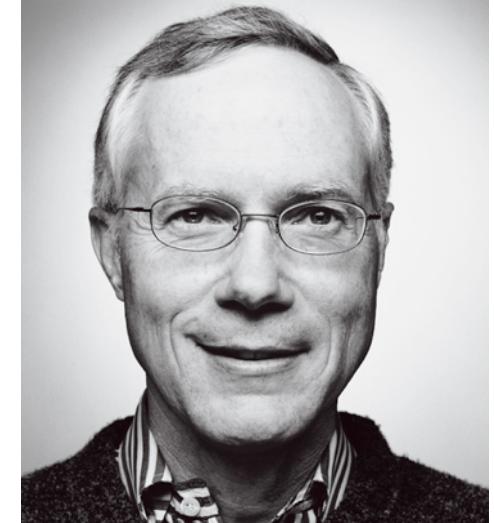


Accounting Professionals

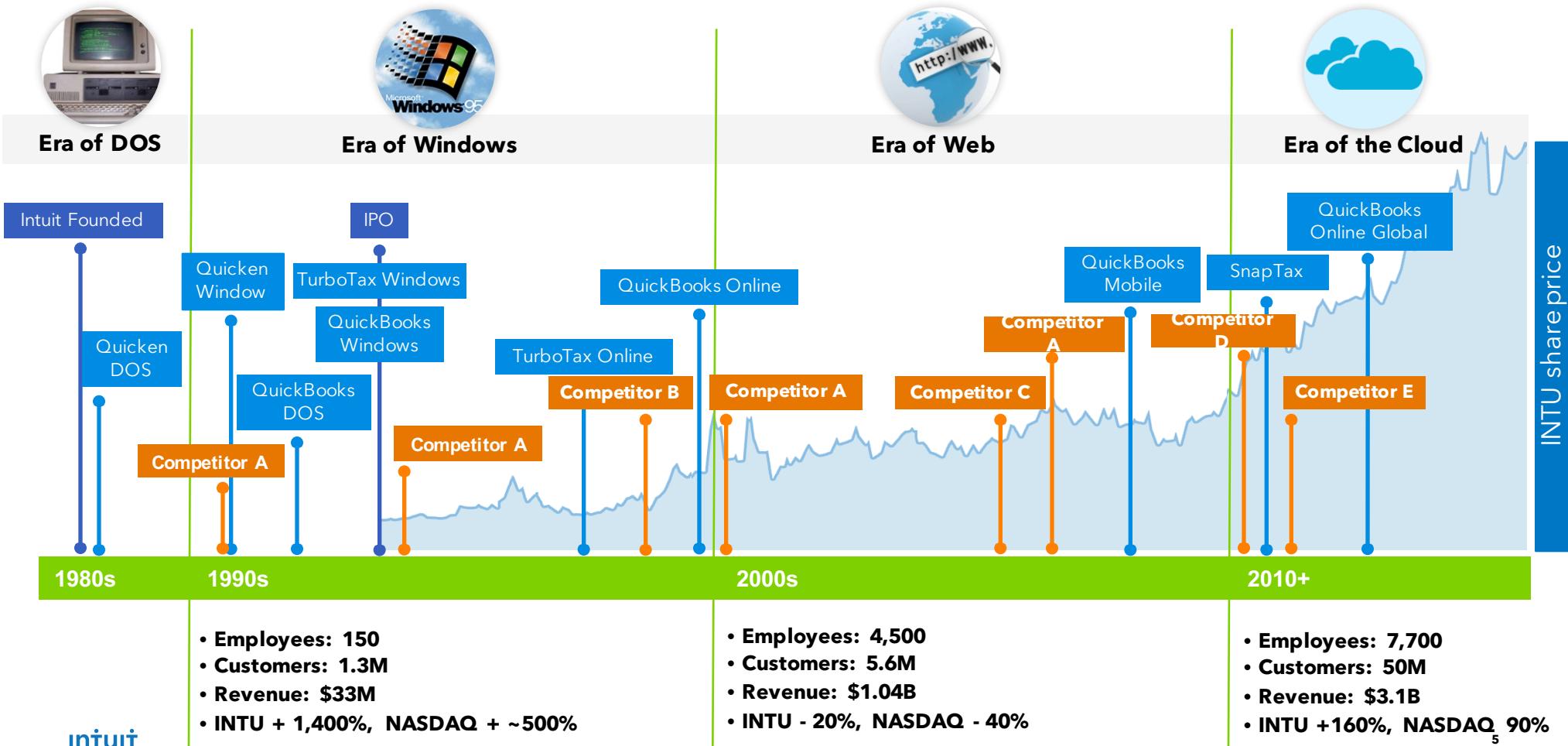


Where we started:

Intuit's journey began over 30 years ago when our founder Scott Cook sat at his kitchen table and watched his wife as she balanced their checkbook and thought **there must be a better way.**



Proven track record: capitalizing on change



Recognized as one of the world's leading companies:

MOST ADMIRE: SOFTWARE INDUSTRY

13 Years in a Row

2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016



MOST INNOVATIVE COMPANIES

intuit.

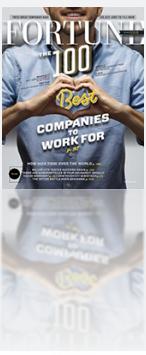


FORTUNE 100 BEST COMPANIES TO WORK FOR - 15 Years in a Row

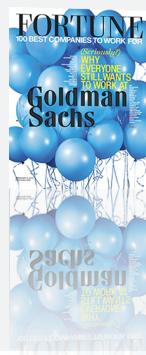
2016



2015



2014



2013



2012



2011



2010



2009



2008



2007



2006



2005



2004



2003



2002



intuit.

Intuit and QuickBooks by the numbers

1 Small Business Management. 7700 employees. \$4.7 Billion in revenue.

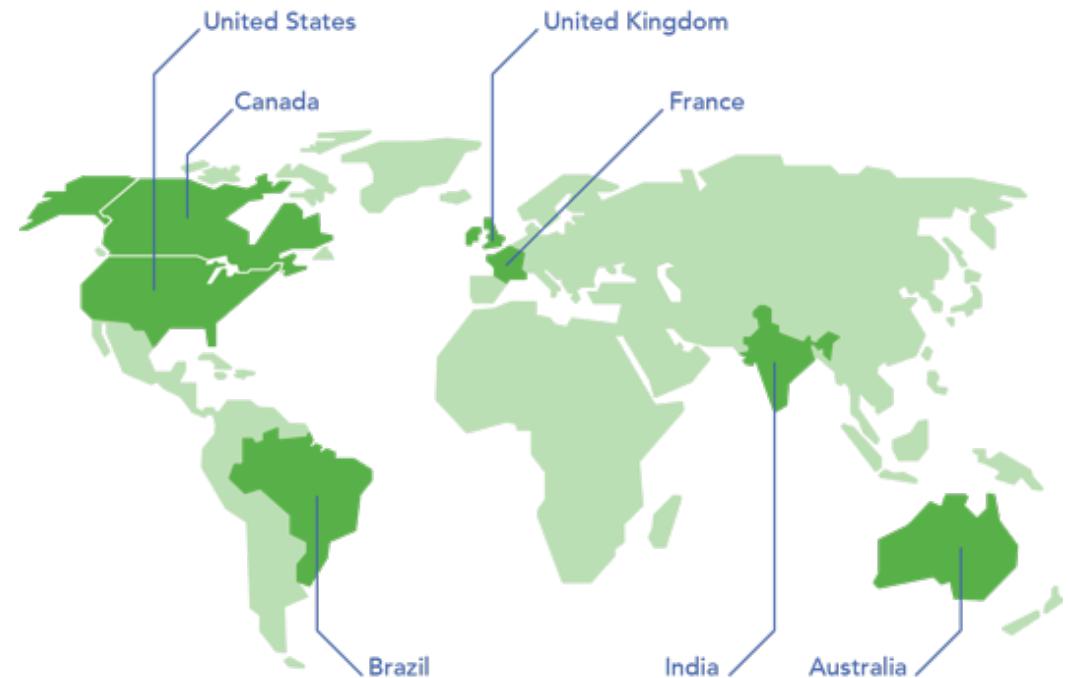
QuickBooks has more than 5 million customers. Online is 1.5 million (41% growth YoY).

26K developers. 2500 third-party apps. 15% of customers use at least one third-party app.

Yearly, 100 million invoices; \$1.5 trillion in commerce; payroll for more than 1 million employees (1/12 in US).

Small Business Owners spend 4 hours online every day.

QuickBooks is in 112 countries.



Challenge for businesses today = Agility

The capability of **rapidly** and **cost efficiently** adapting to changes

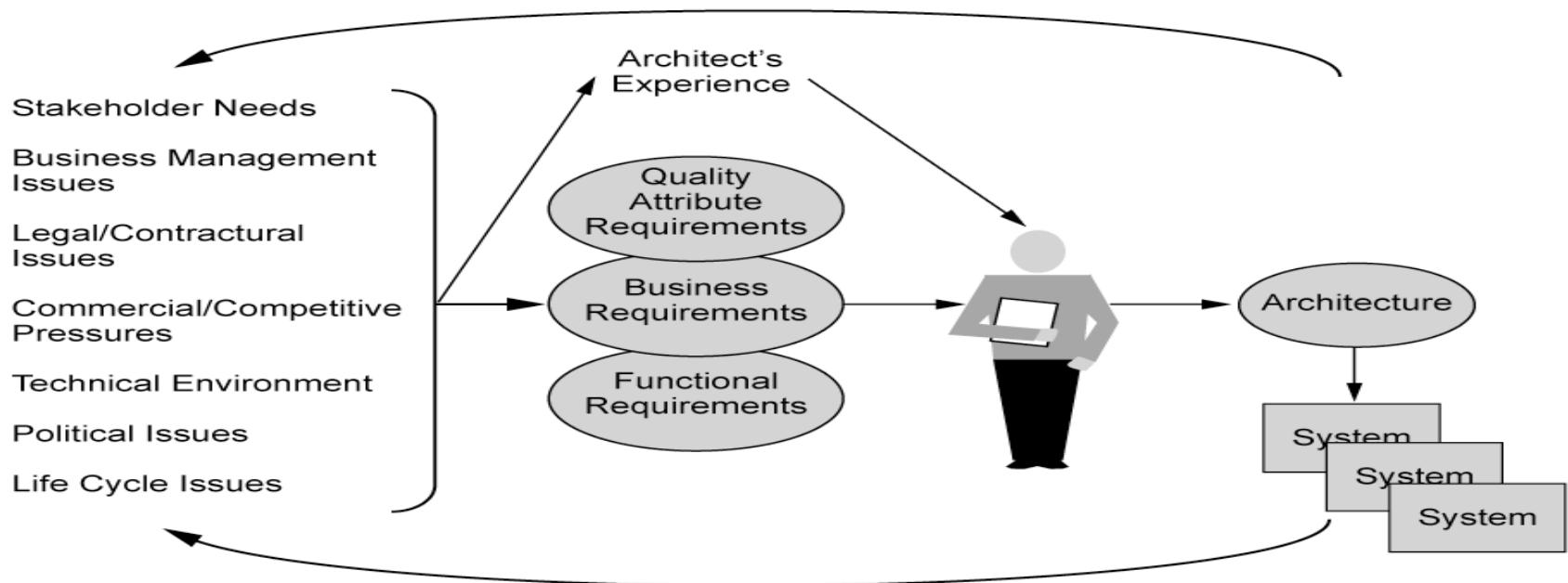
- Wikipedia

The ability of an organization to sense environment change and

to respond **efficiently** and **effectively** to it

- Gartner

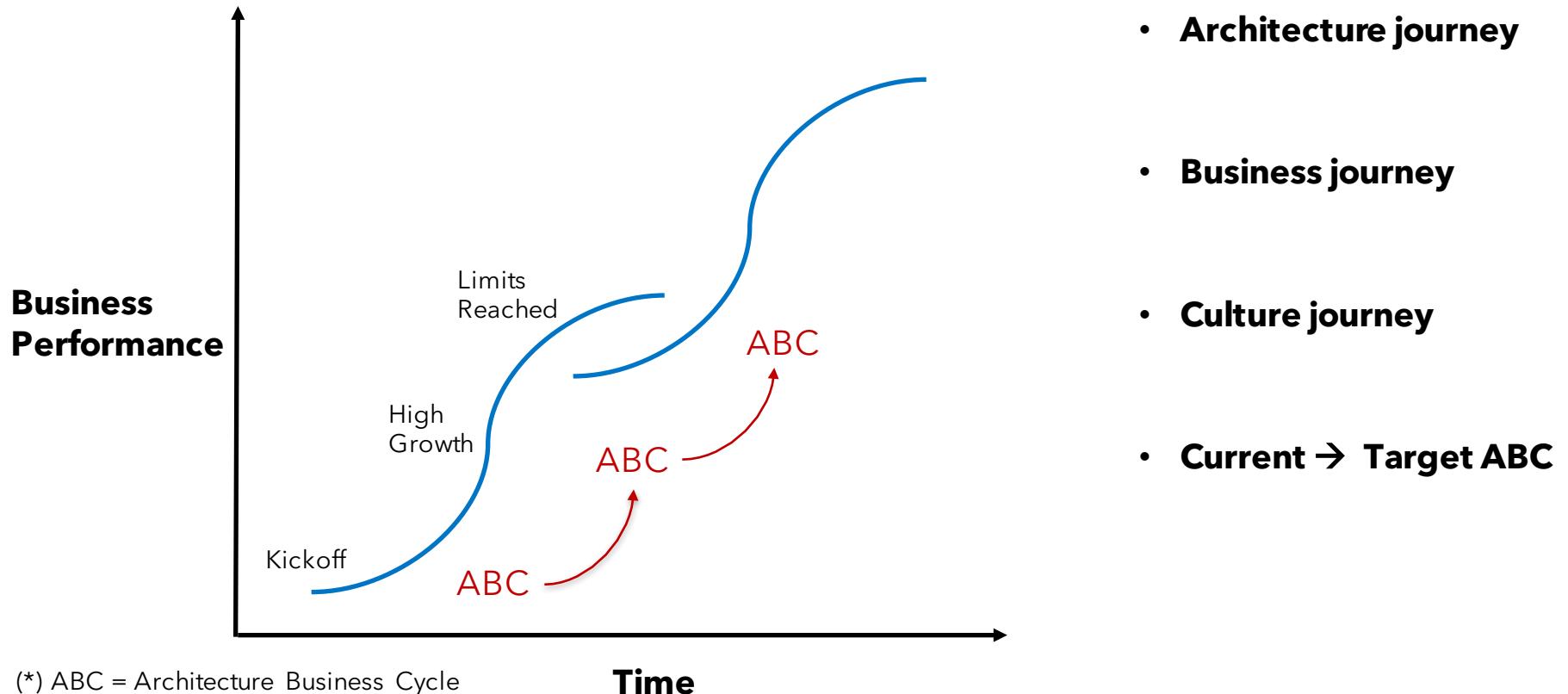
Architecture Business Cycle



Architecture Business Cycle has impact on Architecture, Business and Culture

Software Engineering Institute <http://sei.cmu.edu/>

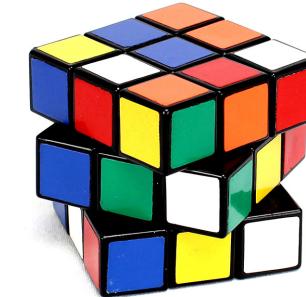
S-Curve



Flexibility and Extensibility

Flexibility

Ease with which a system or component can be **modified for use** in applications or environments **other than those for which it was specifically designed**



Extensibility

Ease with which a system or component can be **modified to increase its functional capacity**



IEEE Standard Glossary of Software Engineering Technology

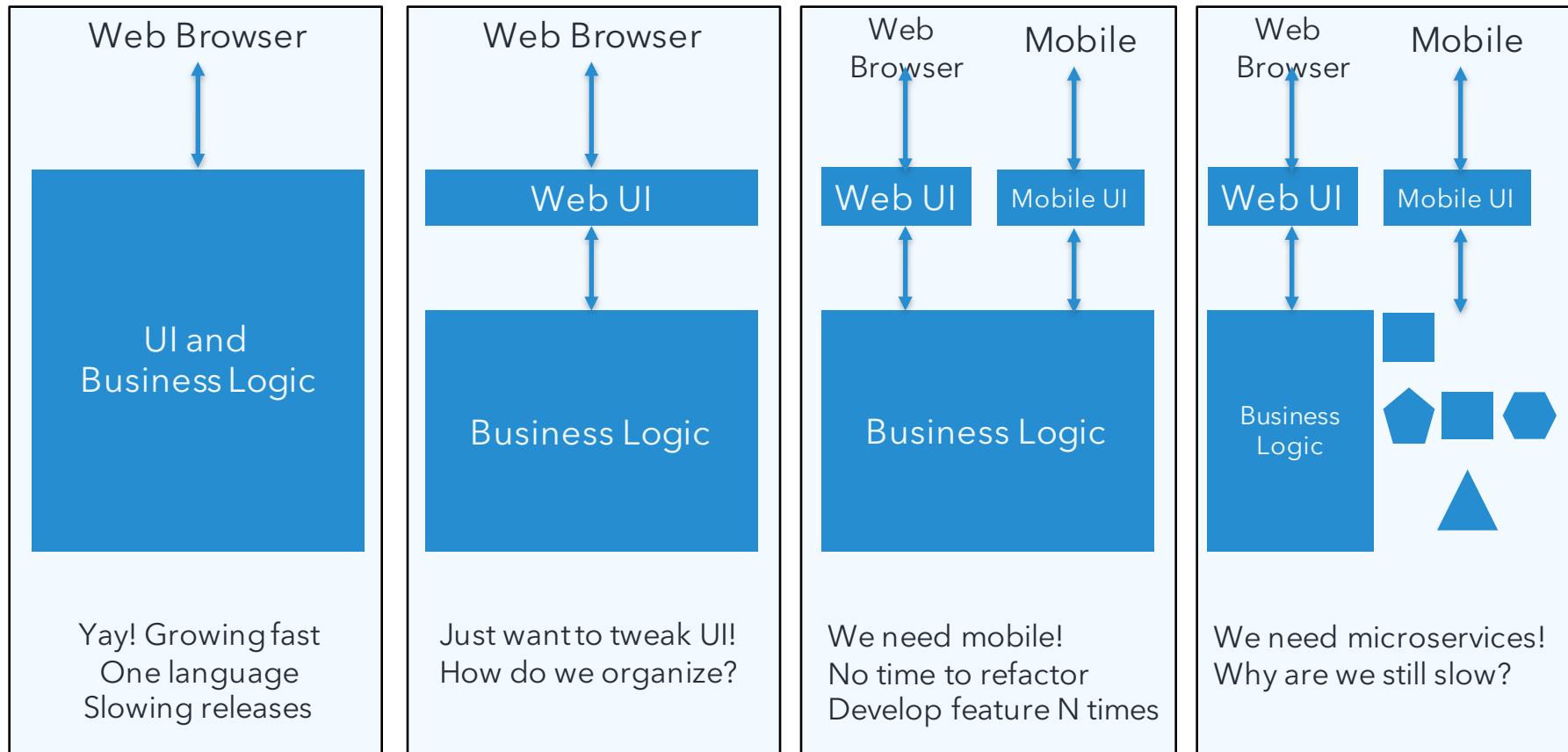
Why APIs and Microservices?

Agility is enabled by Flexibility and Extensibility

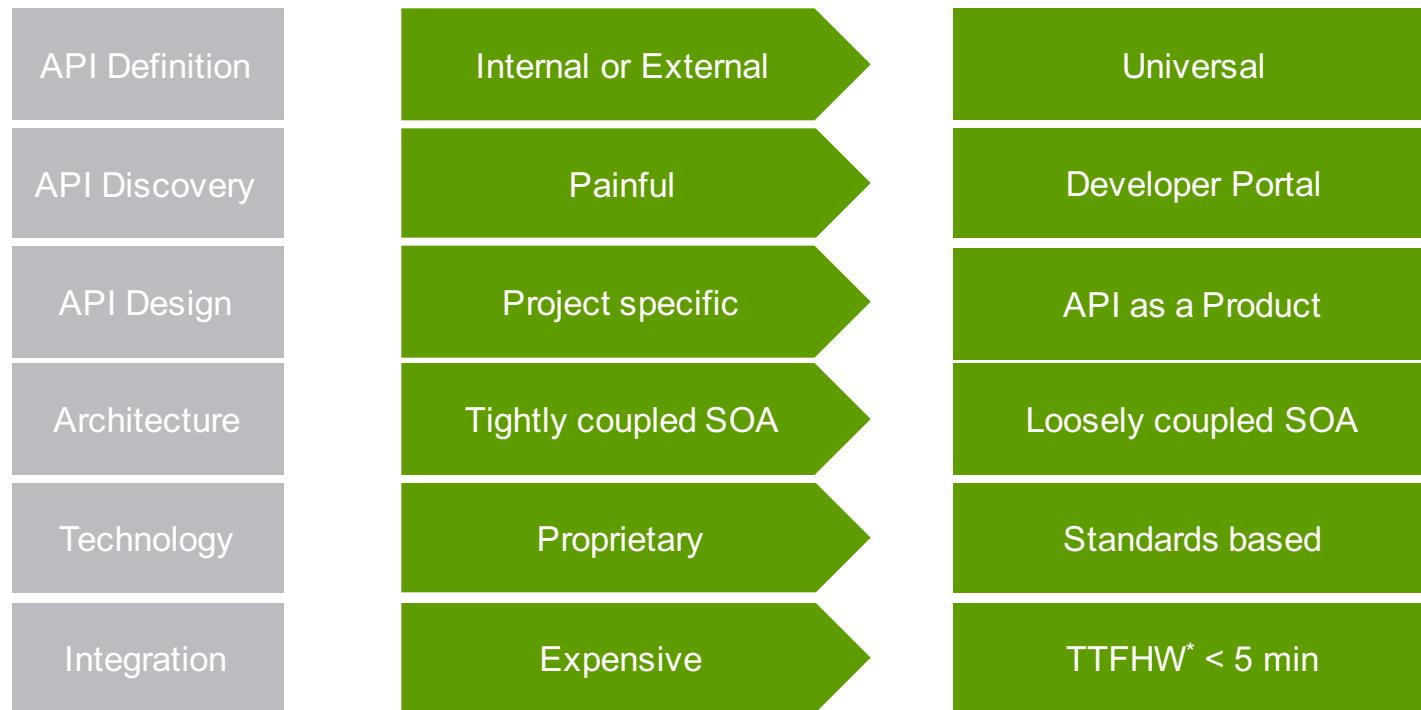
APIs enable flexibility

Microservices enable extensibility

Journey begins as one of the following ...



Target State



(*) TTFHW = Time To First Hello World

API Platform

Portfolio of APIs
aligned by business capabilities,
realized by isolated and encapsulated services,
that can be used by internal and external developers
to develop applications and integrations
quickly and cost effectively

API Platform Principles

API First

- Think API before experience
- API is a product

API Design

- Externalize-able
- Domain model & Integration scenarios

Developer Experience

- Easy to learn, integrate, diagnose
- Be the 'developer'

API Quality Attributes

- Response-time
- Availability

Service Architecture

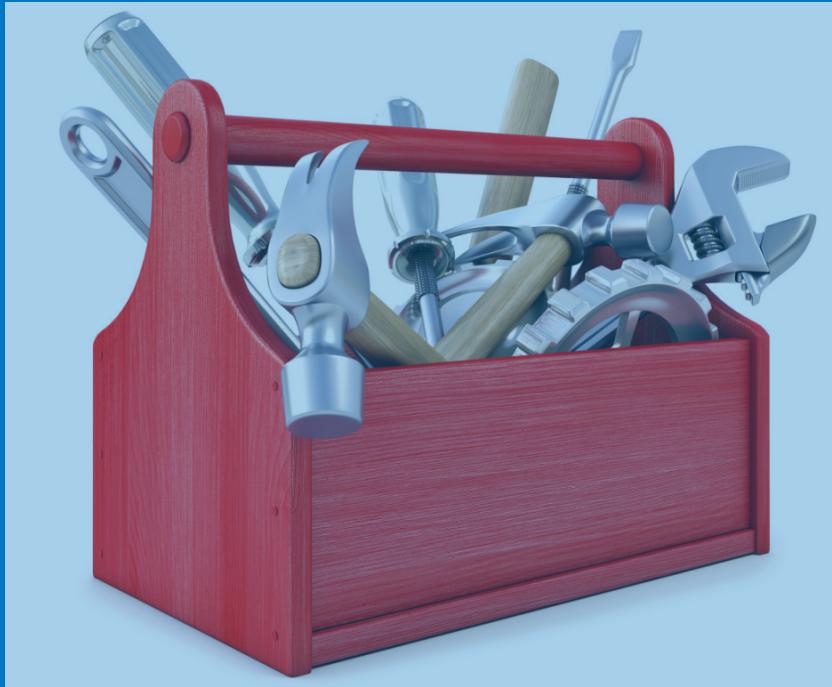
- Encapsulated, Isolated
- Craftsmanship

Elements in building an API Platform

- Know the end user
- Work back from the developer and integration
- Treat 1st party = 2nd party = 3rd party
- Make it easy to integrate
- Zero impact to integration as API evolves
- Loosely coupled services

Best Practices

- API as a product
- API design
- Experience vs Capability API
- API specification
- API documentation
- Service granularity
- Service development
- API testing
- API status & support
- Developer Portal



API as a Product

End users

Understand Segments
Their Experiences
And Expectations



Developers

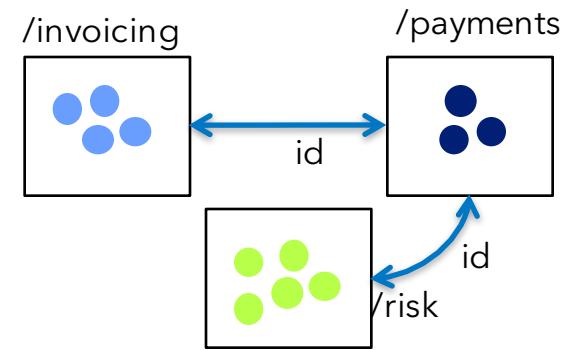
Know Tools and processes
Their Integration needs
Be Transparent on usage



API Design

API Portfolio

Define bounded contexts/capabilities
Organize cohesive concepts
Use orthogonality or distance to separate



API Design

Use problem space for domain model
Consider all interactions – Events, Batch
Plan to dog food API (and events)
Share designs (apistylebook.com)

Domain Model → API

- Entities, Attributes
- Verbs
- Relationships
- State machine

Experience vs Capability API

Capability API

What API teams deliver
Optimized for API provider
REST, Batch, ...

Experience API

Personalized for end user experience
Make it self-service for developer
Plan for multiple - GraphQL, Groovy



API Specification

Many choices

OAS*, JSON Schema

Capability, Experience API

Automatic Docs, SDK generation



Maintain as code

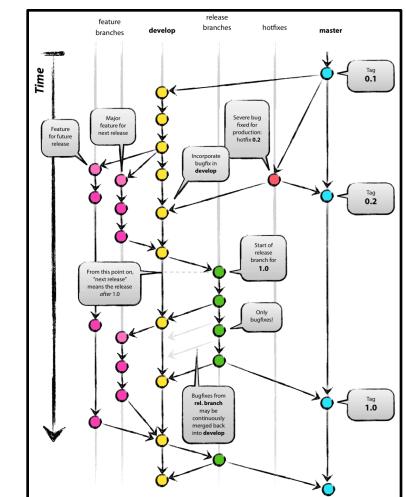
GitHub, GitFlow (agile versioning)

Pull Request (Governance)

Share it

(*) Open API Specification

A screenshot of the GitHub interface for the Elasticsearch repository. The main page shows the rest-api-spec branch. A vertical timeline on the left indicates commits over time. The commits include: "src/main/resources/rest-api-spec" (change health from string to enum), ".gitignore" (initial commit), "README.markdown" (remove reference to utils for generating REST docs), "build.gradle" (build: Add pom building and associated files to rest api spec jar), and "README.markdown". Below the commits, the text "Elasticsearch REST API JSON specification" is displayed, along with a note: "This repository contains a collection of JSON files which describe the Elasticsearch HTTP API."



API Documentation

By developers for developers

Developers contribute documentation

Technical writers are editors

Include in release criteria

Maintain as code

GitHub, GitFlow

Should be tested like code

Open it for contributions

Microsoft Azure

Documentation > Azure IoT Hub

Search for docs

Overview

Get started

Get started with IoT Hub

Connect your device

By Dominic Betts Updated: 09/12/2016 Contributors: +2 Edit on GitHub

Get started with Azure IoT Hub for .NET

Azure / azure-content

Code Pull requests Issues Gist

Branch: master azure-content / articles / iot-hub / iot-hub-csharp-csharp-getstarted.md

dominicbetts Freshness fe7717c 17 days ago

7 contributors

284 lines (189 sloc) 16 KB

Raw Blame History

Get started with Azure IoT Hub for .NET

Service Granularity

Why you should care?

Extensibility

Granularity Principles

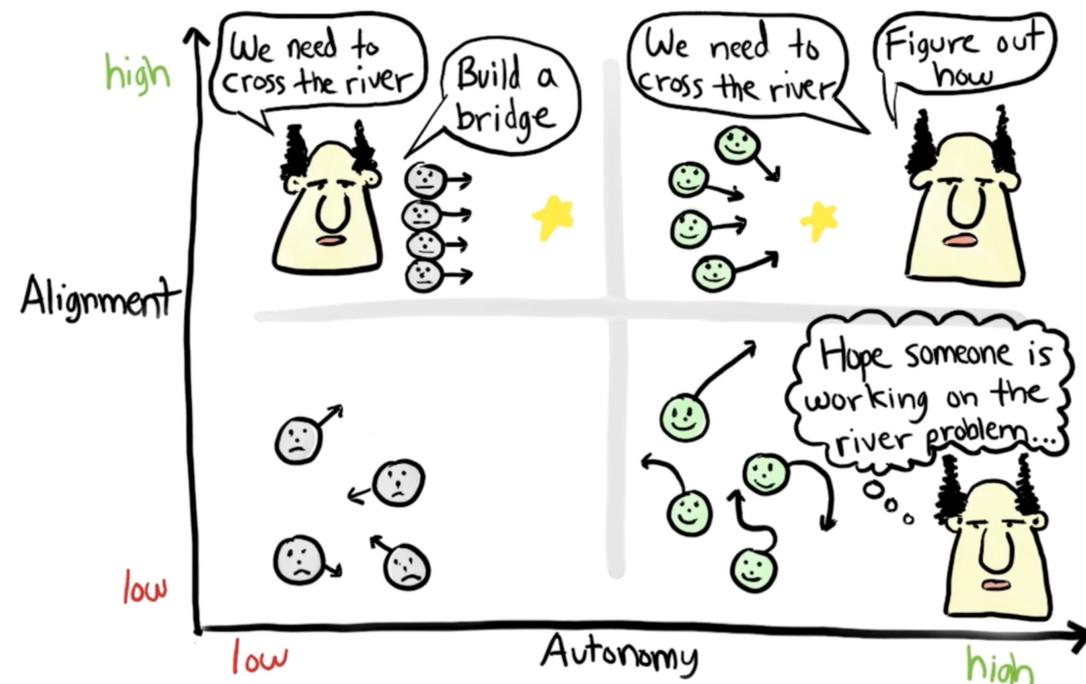
Team boundary

Cohesion

Security

Rate of change

Scalability



Credit: Spotify Engineering Culture

Service Development

Getting to microservices

Code complexity vs. change

Pick the first, complete, repeat

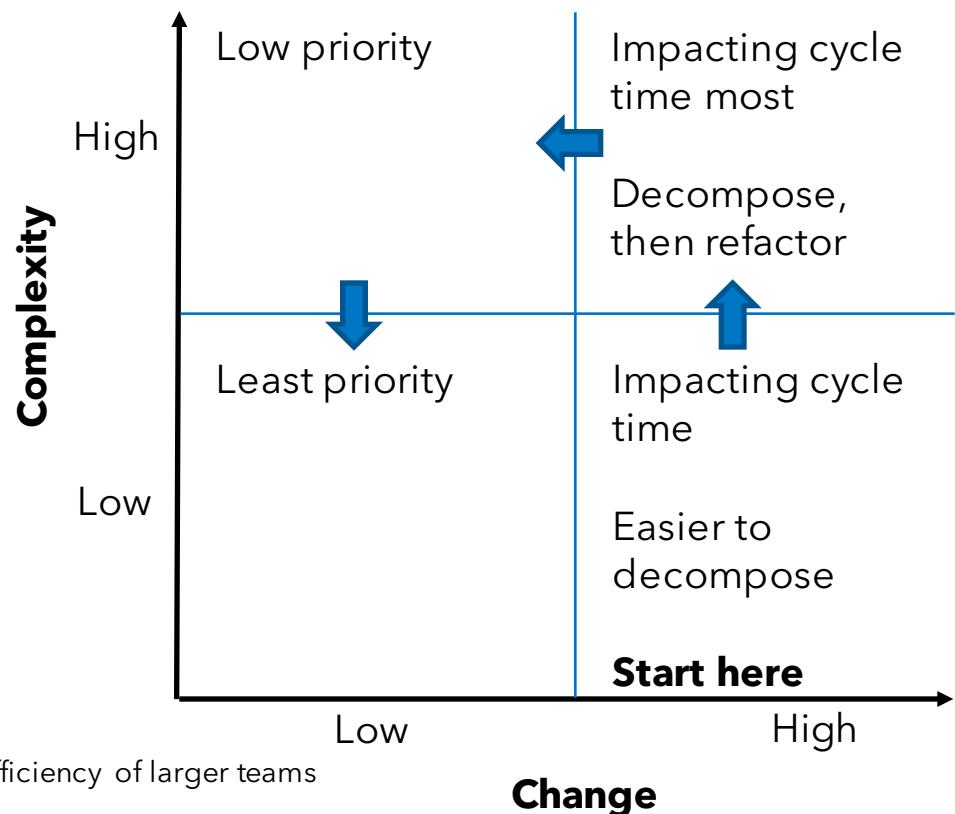
Team autonomy, diagnosing issues

New microservice

Plan for multiple languages

Plan for CI

Team (4) > Team (2) (45% more time)*



(*) The team scaling fallacy: Underestimating the declining efficiency of larger teams

API Testing

API Testing

Test independently

API specification can help bootstrap
Complete scenarios (e.g. Webhooks)



Sandbox

Sandbox = Production

For internal and external developers

Support automated testing



API Status & Support

API Status

Developers need to know
Health dashboard
Report status promptly on issues

Diagnosing issues

Treat developers alike
Provide API logs with messages
Be proactive using API analytics

The image shows two screenshots of the Intuit developer support interface. The left screenshot is a 'Status' page listing various API services with their current status: QuickBooks Apps.com (Operational), Intuit Developer Portal (Operational), QuickBooks Online API (Operational), QuickBooks Payments API (Operational), QuickBooks Sandbox (Operational), Intuit AppConnect (Operational), Financial Data (CAD) API (Operational), Financial Data (CAD) FI Health (Operational), and Webhooks (Operational). The right screenshot is the 'Intuit Developer Forum' page, which includes sections for asking questions, viewing platform status (with a clock icon and a link to the Platform Status Dashboard), and submitting support tickets (with a question mark icon and links to Submit Ticket and View Existing Tickets).

Developer Portal

Complete developer experience

For 1st and 3rd party developers

Time To First Hello World < 5 min

The screenshot shows the Intuit Services Portal homepage with a navigation bar at the top. Below the navigation, a breadcrumb trail shows 'Home > Company > Swagger'. The main content area features a 'Company' card with a 'Published by SBG | Capability: Small Business' badge. A 'TRY IT' button is highlighted. To the right, a 'Swagger Playground' section displays two API endpoints for 'AccountResource':

- GET /accounts[id]/bank-accounts**: Read bank account information of a payment account.
- GET /accounts**: Reads all the realms from an associated given auth and returns the payment account information, if available for each realm-id, refer to More Info.

The screenshot shows the Intuit Developer portal for QuickBooks Online. The top navigation bar includes links for 'APIs', 'Docs & Tools', 'Community', 'Help', 'Sign In', and a search bar. The main content area has tabs for 'ESSENTIALS', 'DEV GUIDES', 'REFERENCES', and 'TOOLS', with 'REFERENCES' currently selected. On the left, a sidebar shows 'Payments API Reference' and a tree view of resources under 'Charges', including 'The charge object', 'Create a charge', 'Retrieve a charge', 'Capture charge funds', 'Retrieve a refund by id', and 'Refund a charge'. The main panel displays the 'Charges' section with a brief description: 'Send and receive payments using credit cards.' Below this is a detailed description of 'The charge object' with fields for 'id' and 'created'.

Culture



What is culture?

A way of thinking, behaving, or working
that exists in a place or organization

- Merriam-Webster

Behavior of humans within an organization and
the meaning that people attach to those behaviors.

- Wikipedia

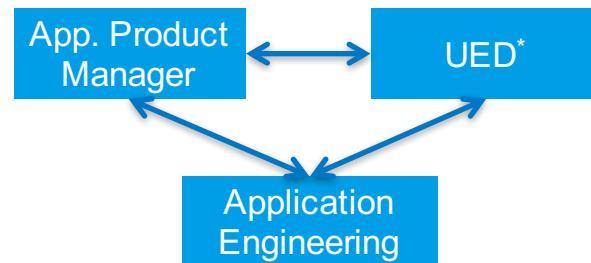
API Engineering Processes

Application development

Web page design Process

Site design/Portfolio Management

Engineers work with page designs

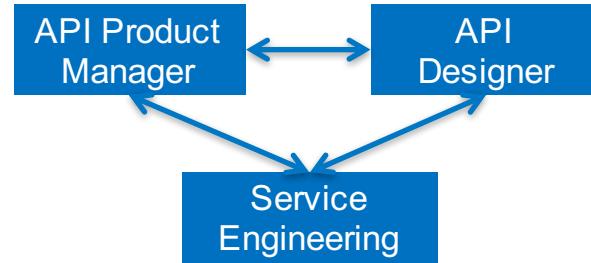


API development

API Design Process

API Portfolio Management

Engineers work with API designs



(*) UED = User Experience Design

Organization Evolution

Be aware of Conway's law

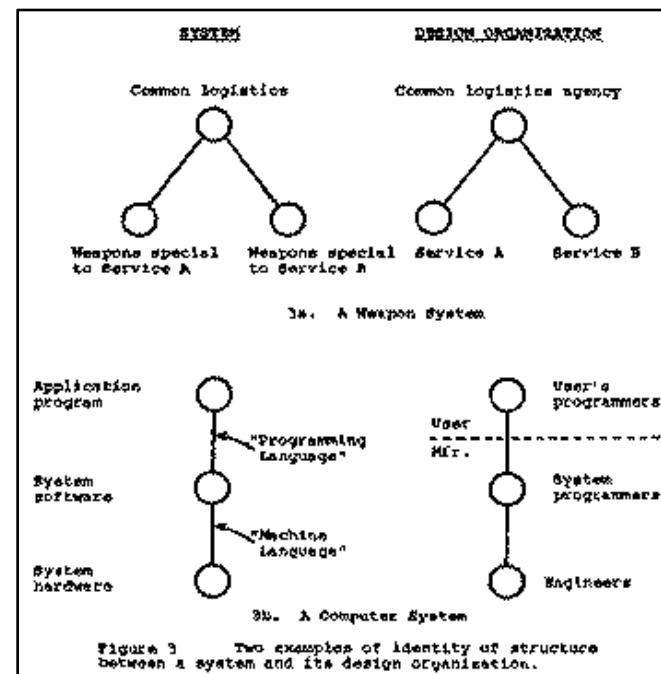
Namespace != Organization

Reuse = Coupling

Agreements based on standards

Teams evolve at different rates

Use maturity model to measure



How do committees invent?

Hackathons

Build something in 24 hours

Internal hackathons

Usability testing for APIs

Pride of ownership for API teams

Developer advocacy



Summary

- Agility is a business imperative
- Agility comes from flexible and extensible architectures
- APIs enable flexibility
- Microservices enable extensibility
- Architecture and culture evolve together

Thank You