



Software Quality Group
of New England

Dedicated to Software Quality Professionals

www.sqgne.org

WELCOME TO SEASON 20!

SQGNE is made possible by the support of our sponsors:



Oracle and
Sun Integrated Systems



SOFTWARE QUALITY CONSULTING
consulting • training • auditing

Feb 2014

Logo design: [Sarah Cole Design](#)

Slide 1

WELCOME TO SEASON 20!

- ◆ All-volunteer **non-profit org.** with no membership dues!
- ◆ Supported entirely by our sponsors...
- ◆ Over **1,200** members on LinkedIn, **800** Constant Contact
- ◆ Monthly meetings - Sept to July on 2nd Wed of month

- ◆ **SQGNE Web site**   **www.sqgne.org**

Officers / Hosts / Mission

Current Officers:

- John Pustaver – Founder
- Steve Rakitin – President
- Stan Wrobel – Vice President
- Barbara Wioncek – Treasurer
- Dawn Wu – Clerk

At-large Directors:

- Robin Goldsmith
- Marge Shinkle
- Jim Turner

Our gracious Hosts:

- Paul Ratty
- Tom Arakel
- Marge Shinkle
- Jack Guilderson

Mission

- To promote use of engineering and management techniques that lead to delivery of high quality software
- To disseminate concepts and techniques related to software quality engineering and software engineering process
- To provide a forum for discussion of concepts and techniques related to software quality engineering and the software engineering process
- To provide networking opportunities for software quality professionals



WELCOME TO SEASON 20!

SQGNE 2013-14 Schedule

Speaker	Affiliation	Date	Topic
Brian LeSuer Nick Olivo	Zeenyx SmartBear	Sept 11	Test Tool Bake-off: Ascential Test from Zeenyx and Test Complete from SmartBear
Harish Narayan	Vistaprint	Oct 9	The Changing Role of a QA Engineer
Byron Mattingly		Nov 13	Testing Mobile Apps
Peter Varhol Gerie Owen	Telerik NSTAR	Dec 11	How'd I Miss That Bug?
Carlo Cadet	Perfecto Mobile	Jan 15	Testing Mobile Apps – Best Practices and Considerations
Matti Hjelm	SmartBear	Feb 12	(What's so Fun About) Testing APIs
Dave Grabel	Agile New England	Mar 12	How Agile elevates the role of the QA Engineer
David Marston	Pegasystems	Apr 9	Test Case Management Systems and Metadata
Robin Goldsmith	GoPro Management	May 14	Defining Quality in Positive Terms
Andrew Ambrose	Vistaprint	June 11	Building Quality Into Your Process Treasurer's Report and Annual Election of Officers
Everyone		July 9	Annual Hot Topics Night...



WELCOME TO SEASON 20!

Tonight's Topic

(What's so fun about) API Testing?

Matti Hjelm, SmartBear Software

Abstract:

Matti will talk about design, evaluation and testing of Web APIs. He will be covering why it's so important to test APIs, what you should test (and/or verify), and of course, how to test. There will be an overview, some theory, and practical walkthroughs

Bio:

Matti Hjelm is the Product owner of SoapUI at SmartBear Software. His 25 years of experience creating and testing software solutions combined with an interest for improving processes has led him from programming to evangelizing iterative software development in the Swedish Agile networks.

Matti is the founder of the Agile Sweden network and conferences.



Feb 2014

SMARTBEAR *WELCOME TO SEASON 20!*

Matti Hjelm – SmartBear Software

(WHAT'S SO FUN ABOUT) TESTING WEB APIS?

the goal of this presentation...



who are you?
who am I?
what is SoapUI?



Web APIs – what's the fuss?



Technically speaking...





POST http://127.0.0.1:8088/mockSampleServiceSoapBinding HTTP/1.1

Accept-Encoding: gzip,deflate

Content-Type: text/xml;charset=UTF-8

SOAPAction: <http://www.example.org/sample/login>

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sam="http://www.example.org/sample/">
```

```
  <soapenv:Header/>
```

```
  <soapenv:Body>
```

```
    <sam:login>
```

```
      <username>Login</username>
```

```
      <password>Login123</password>
```

```
    </sam:login>
```

```
  </soapenv:Body>
```

```
</soapenv:Envelope>
```



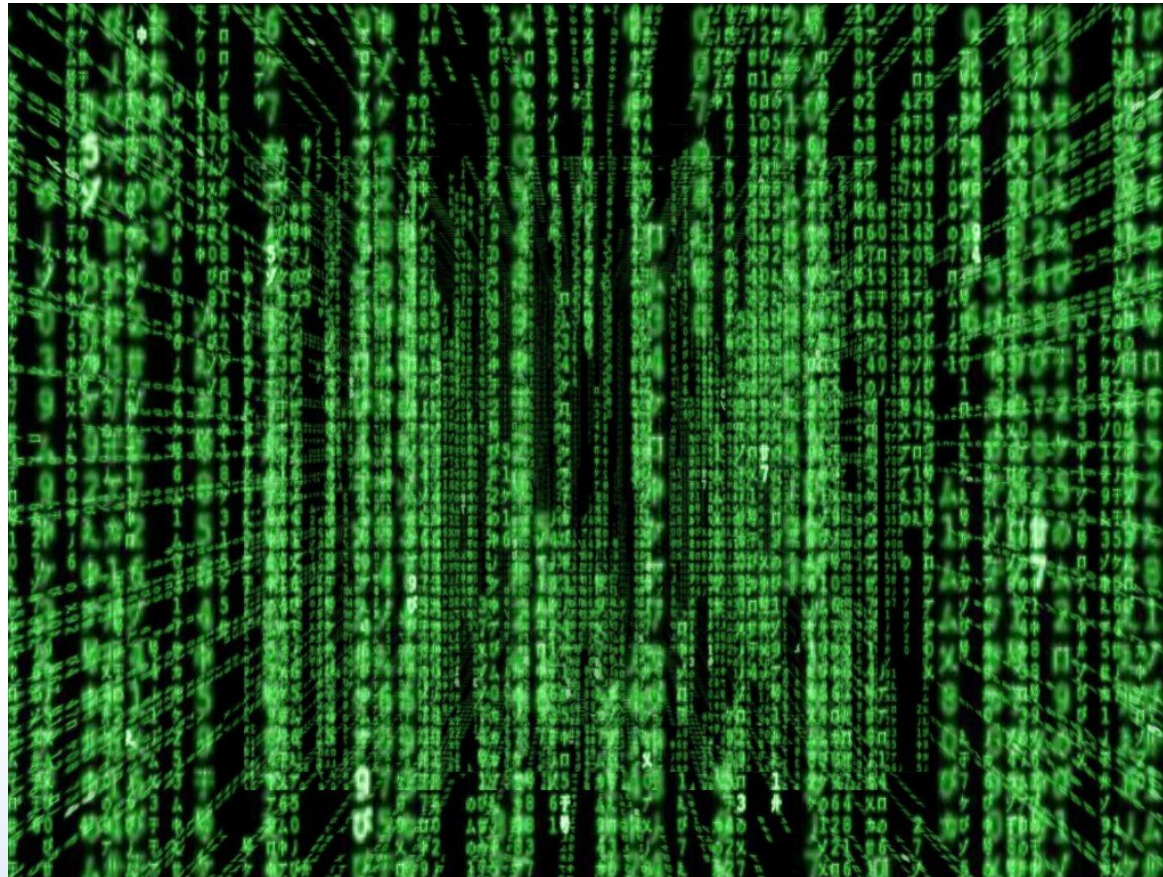
```
GET http://maps.googleapis.com/maps/api/geocode/xml?
address=Hornsbruksgatan%2028%2C%20Stockholm&sensor=false
HTTP/1.1
Accept-Encoding: gzip,deflate
Host: maps.googleapis.com
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

```
{
  "person":
  { "firstname": "matti",
    "lastname": "hjelm"
  }
}
```

SOAP /
XML

REST /
JSON

Async /
WebSockets



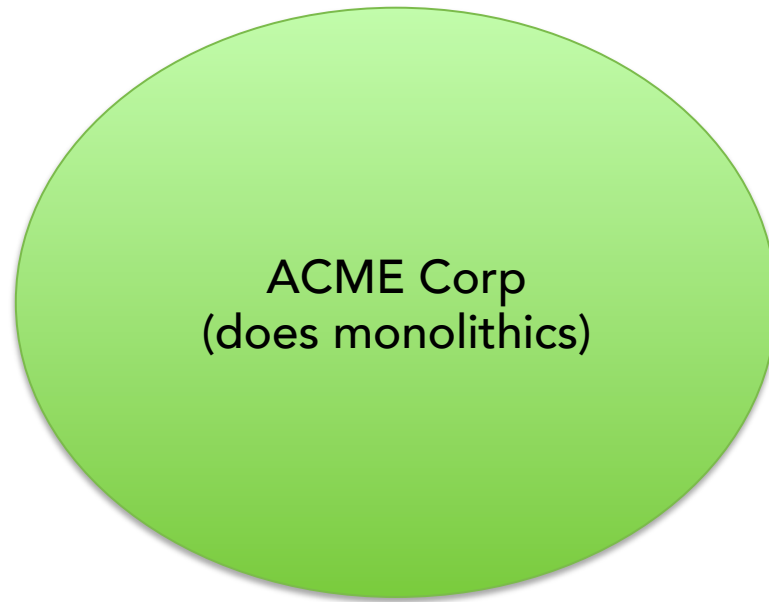
Technically speaking...



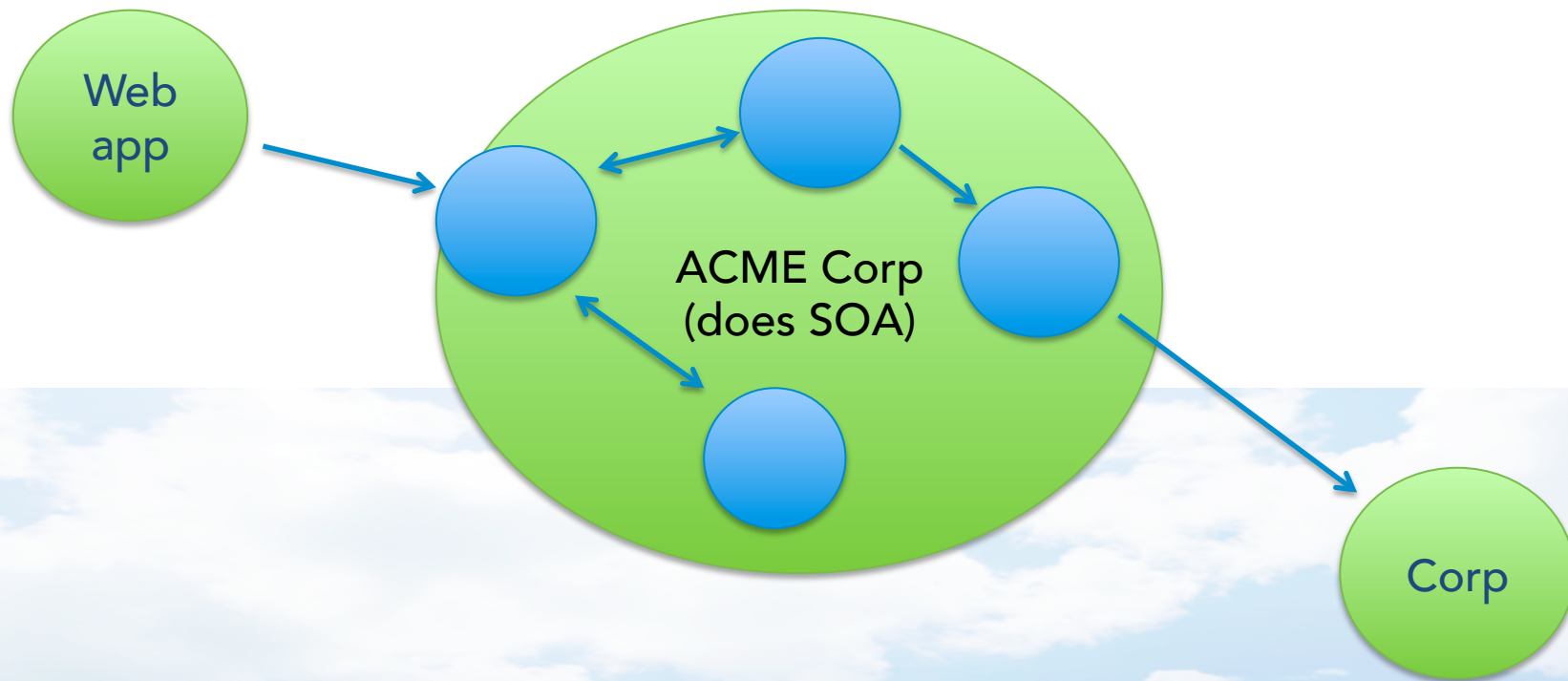
but that's missing the point...

Business aspect

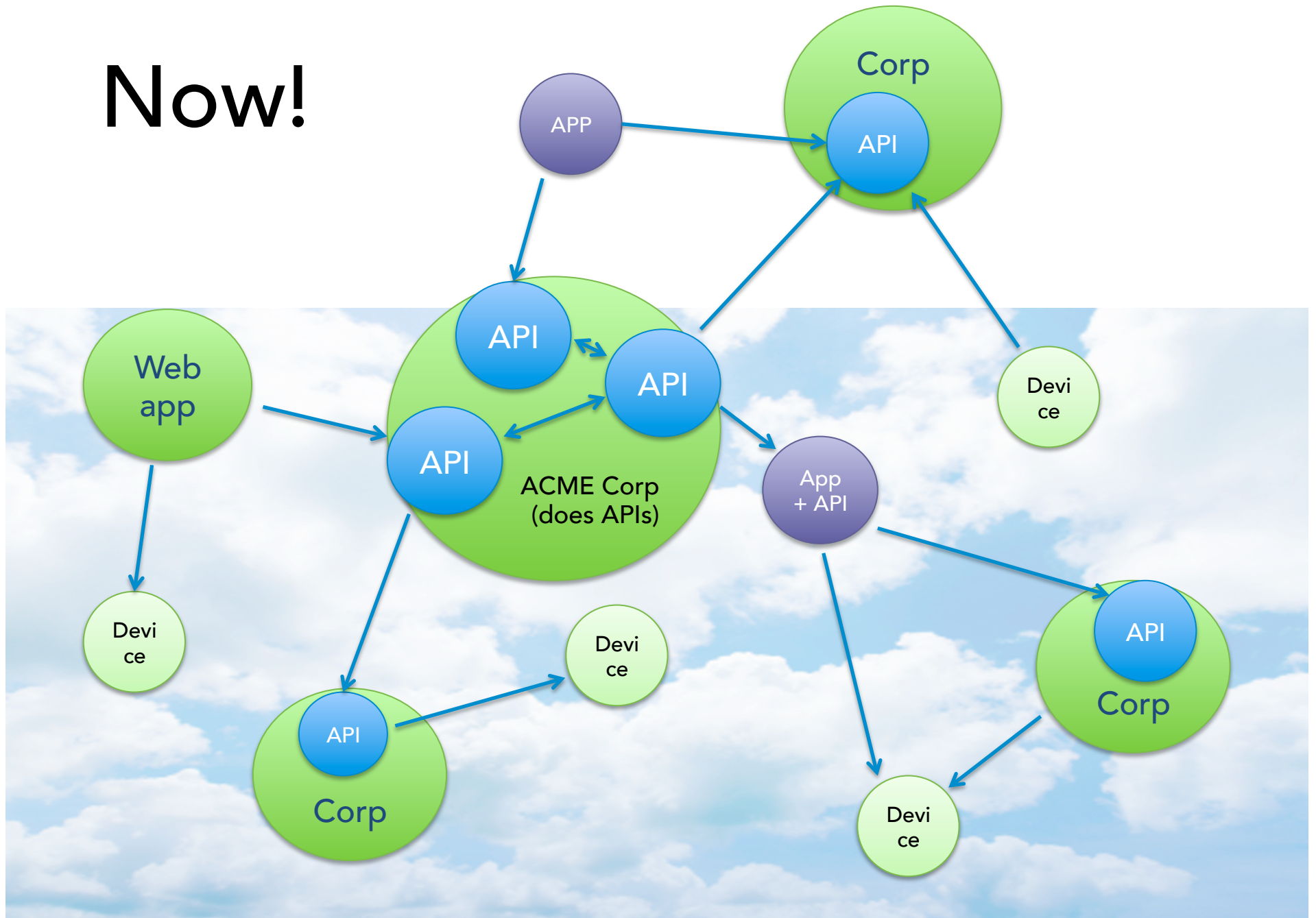
20 Years Ago



10 Years Ago



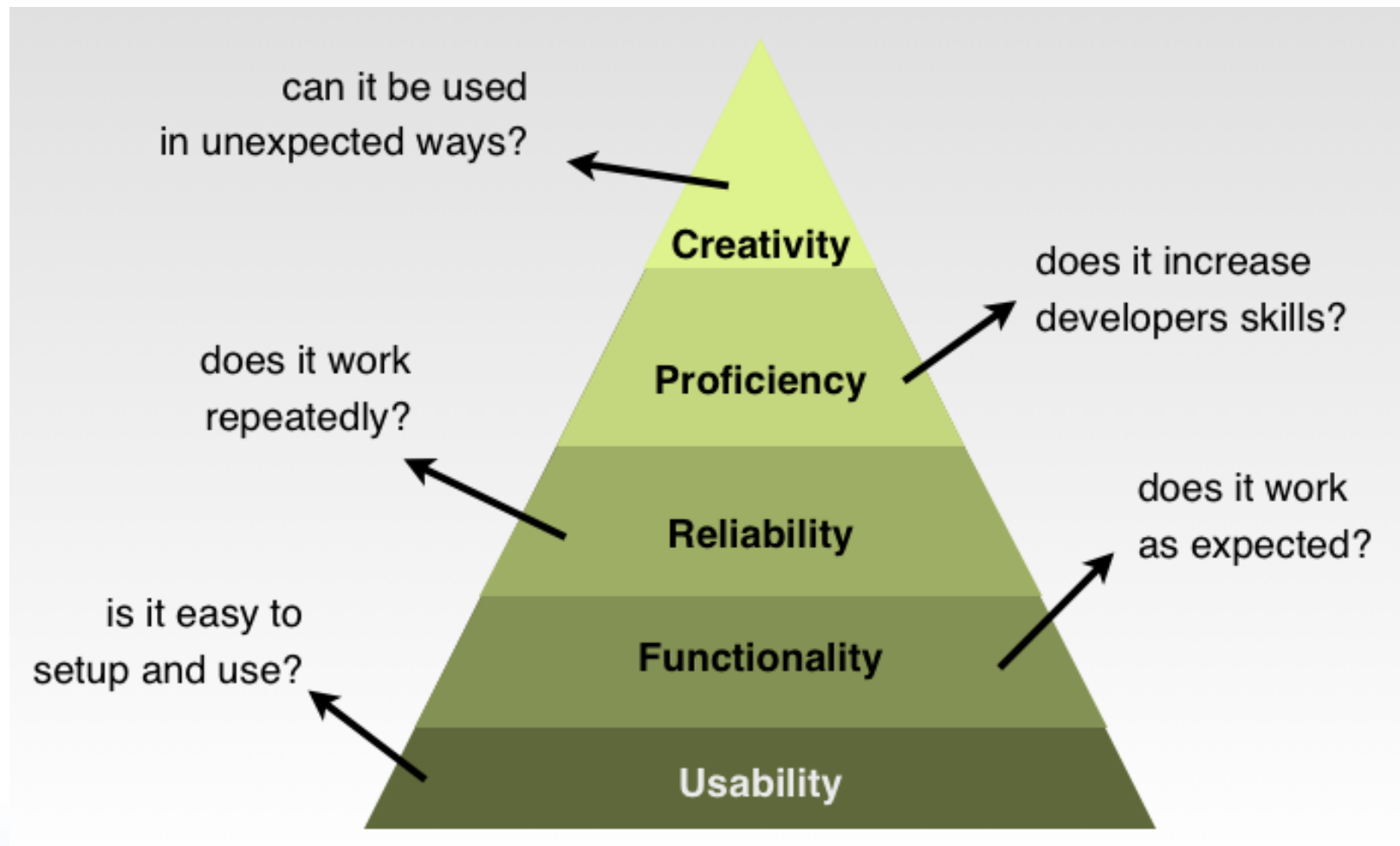
Now!



API Quality – Who needs it?



API Hierarchy of Needs





POST http://127.0.0.1:8088/mockSampleServiceSoapBinding HTTP/1.1

Accept-Encoding: gzip,deflate

Content-Type: text/xml;charset=UTF-8

SOAPAction: <http://www.example.org/sample/login>

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sam="http://www.example.org/sample/">
```

```
  <soapenv:Header/>
```

```
  <soapenv:Body>
```

```
    <sam:login>
```

```
      <username>Login</username>
```

```
      <password>Login123</password>
```

```
    </sam:login>
```

```
  </soapenv:Body>
```

```
</soapenv:Envelope>
```



API Usability is important!



User Experience

=

Developer Experience







Who is your target persona?



Use lifelike Personas



Our Tester Personas

<p>Calvin Brown QA Manager 45, US</p> <p>Calvin is a 45-year old QA manager living in California. He is a well known testing guru, often holding speeches at test conferences. At work he is a guarantee that the software delivered is of high quality.</p> <p>Characteristics</p> <ul style="list-style-type: none"> * Hard working * Active in the community * Intelligent * Experienced * Persistent <p>"Testing is NOT about finding bugs, it's about PREVENTING bugs"</p> <p>Goals</p> <ul style="list-style-type: none"> * Bug free software * Work efficiently * Advance testing to a higher level * Ensure software quality 	<p>Dinushka Charitha QA Tester 35, India</p> <p>Dinushka works as a QA tester in a team of 5 people. They are mainly responsible for maintaining part of a legacy system which is used by a government financial institute. She mostly runs regression tests. She also uses SoapUI and some of her test cases are automated using SoapUI. Usually, before releases she is stressed to meet deadlines. She has a big social life.</p> <p>Characteristics</p> <ul style="list-style-type: none"> * Structured * Loyal * Wants to get along * Likes to stay in her comfort zone <p>"A good day at work is when nothing breaks"</p> <p>Goals</p> <ul style="list-style-type: none"> * Do a good job * Frictionless environment * Wants SoapUI to bring structure to her tests 	<p>Joanna Wright Manual tester 30, US</p> <p>Joanna used to work in real-estate, now works as a tester in the Manual Testing team at a bank. Works according to strict processes. Cares about the tool because it helps her get her job done and saves her from manual testing procedures. Since not being very technical she finds it a bit hard communicating with the developers. Joanna loves her job and regrets not getting into IT earlier.</p> <p>Characteristics</p> <ul style="list-style-type: none"> * Likes feeling safe * Not very technical * Really enjoys testing * A smart person <p>"I love IT! Why didn't i change job earlier?"</p> <p>Goals</p> <ul style="list-style-type: none"> * Wants to finish on time * Would like to get rid of tedious work 
<p>Mark Bingham Developer doing tests 28, US</p> <p>Mark is passionate about learning, and wants to make sure that what he delivers is of high quality. He motivates co-workers to write more tests. Currently uses SoapUI to achieve his goals, but wouldn't mind switching if something else would work better. Extends the functionality of Pro by using scripting.</p> <p>Characteristics</p> <ul style="list-style-type: none"> * Pragmatic * Early adopter * Reads to learns new things * Likes flexibility * Enjoys communities <p>"I'm going to create the Next Big Thing"</p> <p>Goals</p> <ul style="list-style-type: none"> * To shine * Build the next-big-thing * High status in his workplace 	<p>Dmitry Levin Software Developer 29, Russia</p> <p>Dmitry is a developer doing a lot of full-time testing at a mid-sized IT consultant agency. Work is not everything and he has a rich spare time and prefers to spend it with his family or restoring his old car.</p> <p>Characteristics</p> <ul style="list-style-type: none"> * Takes pride in his work * Likes a stable work environment * Flexible attitude * Handful of certificates * Cares a lot about risks <p>"I can probably make a script for that"</p> <p>Goals</p> <ul style="list-style-type: none"> * Simplify testing * Work productively * Be able to perform low-level testing 	<p>Uwe Krull Developer 25, Germany</p> <p>Uwe is technically advanced. His focus can be perceived as having a one track mind. Looking for confirmation that his theory is right. Very goal oriented. Likes creating his own tools for work since it's more flexible. Has a collection of old gaming consoles in a closet. Uses SoapUI to verify his work.</p> <p>Characteristics</p> <ul style="list-style-type: none"> * One track or focused mind * Goal oriented * Technically advanced <p>"PRO features can easily be made in Groovy script"</p> <p>Goals</p> <ul style="list-style-type: none"> * Deliver the fastest possible system * Identify the tool that helps him with this 

Align with their API technology

SOAP / REST / etc...

XML / JSON / etc...

QoS / Security



Help them understand your API

The screenshot shows the Swagger API Explorer interface for 'my-awesome-api.com'. It displays a REST API for a resource named 'word'. The API is organized into sections: 'USER' and 'word'. The 'word' section lists various endpoints with their methods and descriptions:

- GET /word.json/{word}/entries: Return entries for a word
- GET /word.json/{word}/examples: Returns examples for a word
- POST /word.json/{word}/examples: Fetches examples for a word
- POST /word.json/{word}/wordForms: Adds a Relationship Map to a word
- GET /word.json/{word}/wordForms: Returns other forms of a word
- DELETE /word.json/{word}/wordForms: Deletes a relationship from a word
- GET /word.json/{word}: Given a word as a string, returns the WordObject that represents it

Below the endpoints is a 'Parameters' section with a table:

Parameter	Value	Description
word	<input type="text" value="(required)"/>	String value of WordObject to return
useCanonical	<input type="text"/>	If true will try to return the correct word root ('cats' -> 'cat'). If false returns exactly what was requested.
includeSuggestions	<input type="text"/>	Return suggestions (for correct spelling, case variants, etc.)
shouldCreate	<input type="text"/>	Create word if not existing

At the bottom of the 'word' section, there are more endpoints:

- GET /word.json/{word}/definitions: Return definitions for a word
- GET /word.json/{word}/stats: Returns word statistics
- GET /word.json/{word}/topExample: Returns a top example for a word
- GET /word.json/{word}/contextualLookup: Returns definitions for a word based on the sentence in which it is found
- POST /word.json/{word}/contextualLookup: Returns definitions for a word based on the sentence in which it is found
- GET /word.json/{word}/commentCount: Returns the number of comments on a word
- GET /word.json/{word}/related: Return related words (thesaurus data) for a word
- GET /word.json/{word}/listedIn: Returns WordLists containing a word
- GET /word.json/{word}/listedInCount: Returns a count of lists a word appears in

VS

The screenshot shows a Notepad window titled 'readme - Notepad' containing the README for 'Simple Twitter Connect'. The text is as follows:

```
==== Simple Twitter Connect ====
Contributors: Otto42
Donate link: https://www.paypal.com/cgi-bin/webscr?
cmd=_donations&business=otto%40otto&deconstruct%2ecom
Tags: twitter, connect, simple, otto, otto42, javascript
Requires at least: 3.0
Tested up to: 3.2
Stable tag: 0.15

==== Description ====

Simple Twitter Connect is a series of plugins that let you add any sort of Twitter
functionality you like to a WordPress blog. This lets you have an integrated site
without a lot of coding, and still letting you customize it exactly the way you'd like.

First, you activate and set up the base plugin, which makes your site have basic
Twitter functionality. Then, each of the add-on plugins will let you add small pieces
of specific Twitter-related functionality, one by one.

Bonus: Unlike other Twitter plugins for WordPress, this one helps you create your
own Twitter application and identity, so your tweets from here show up as being
from Your Blog, not from some plugin system. You'll never see "posted by Simple
Twitter Connect" in your tweet stream, you'll see "posted by Your Blog Name".
Great way to drive traffic back to your own site and to see your own Twitter
userbase.

Requires WordPress 3.0 and PHP 5.
```

A Google API

<https://developers.google.com/maps/documentation/geocoding/>

Introduction

Directions API

Distance Matrix API

Elevation API

Geocoding API

Time Zone API

Blog

Support

FAQ

Maps JavaScript API v3

Google Maps API for Business

Google Places API

Static Maps API

Street View Image API

Earth API

▶ Deprecated APIs

The Google Geocoding API

[What is Geocoding?](#)

[Before You Begin](#)

[Usage Limits](#)

[Geocoding Requests](#)

[Geocoding Responses](#)

[JSON Output Formats](#)

[XML Output Formats](#)

[Status Codes](#)

[Error Messages](#)

[Results](#)

[Address Component Types](#)

[Reverse Geocoding](#)

[Viewport Biasing](#)

[Region Biasing](#)

[Component Filtering](#)

This document discusses the Geocoding API v3. Note that the Geocoding API v2 has been turned down and is no longer available. You should [upgrade](#) to v3.

Looking to use this service in a JavaScript application? Check out the [Geocoder](#) class of the Google Maps API v3.

What is Geocoding?

Geocoding is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (-122.083739), which you can use to place markers or position the map.

Reverse geocoding is the process of converting geographic coordinates into a human-readable address.

Geocoding Requests

A Geocoding API request must be of the following form:

```
http://maps.googleapis.com/maps/api/geocode/output?parameters
```

where **output** may be either of the following values:

- **json** (recommended) indicates output in JavaScript Object Notation (JSON)
- **xml** indicates output as XML

To access the Geocoding API over HTTPS, use:

```
https://maps.googleapis.com/maps/api/geocode/output?parameters
```

HTTPS is recommended for applications that include sensitive user data, such as a user's location, in requests.

Provide API Metadata

Validation

Code Generation

Coverage

Understanding

Simulation

```
method: "GET",
summary: "Get user by user name",
notes: "",
type: "User",
nickname: "getUserByName",
- parameters: [
  - {
    name: "username",
    description: "The name that needs to be fetched. Use user1 for testing.",
    required: true,
    type: "string",
  }
],
- responseMessages: [
  - {
    code: 404,
  }
]
},
- {
  path: "/api/users/{username}/login",
  - operations: [
    - {
      method: "POST",
      summary: "Login user",
      notes: "",
      type: "string",
      nickname: "loginUser",
      - parameters: [
        - {
          name: "username",
          description: "The username for login",
          required: true,
          type: "string",
          paramType: "query",
        },
        - {
          name: "password",
          description: "The password for login in clear text",
          required: true,
          type: "string",
          paramType: "query"
        }
      ]
    }
  ]
}
```

wsdl, swagger,
wadl, raml, api
blueprint, hal, json
schema, apiary.io,
xml schema, ws-*,
apiary, api-docs,
iodocs, etc

Provide API Metadata

Swagger

pet : Operations about pets

Show/Hide | List Operations | Expand Operations

GET /pet/{petId} [Find pet](#)

Implementation Notes
Returns a pet based on ID

Response Class
Model | Model Schema

Pet {
id (integer): unique identifier for the pet,
category (Category, optional),
name (string),
photoUrls (array[string], optional),
tags (array[Tag], optional),
status (string, optional) = ['available' or 'pending' or 'sold']: pet status in the store
}

Category {
id (integer, optional),
name (string, optional)
}

Tag {
id (integer, optional),
name (string, optional)
}

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
petId	<input type="text" value="(required)"/>	ID of pet that needs to be fetched	path	integer


Error Status Codes

HTTP Status Code	Reason
400	Invalid ID supplied
404	Pet not found

[Try it out!](#)

Provide API Metadata

Validation can be
automated in your
test tool



Provide API Metadata
















Code generation
will be easier for
client developers,
and API Developers

Provide API Metadata

Coverage will be
correct

Provide API Metadata


Coverage will be correct

Element	Contract Coverage	
▼ Amazon	6% (0%)	 196/3299
▼ AWSECommerceServiceBinding	6% (0%)	 196/3299
▶ BrowseNodeLookup	0% (0%)	 0/163
▶ CartAdd	0% (0%)	 0/228
▶ CartClear	0% (0%)	 0/214
▶ CartCreate	0% (0%)	 0/230
▶ CartGet	0% (0%)	 0/214
▶ CartModify	0% (0%)	 0/224
▶ ItemLookup	23% (0%)	 155/666
▼ ItemSearch	6% (0%)	 41/706
▶ Request	9% (0%)	 6/70
▶ Response	6% (0%)	 35/636
▶ SimilarityLookup	0% (0%)	 0/654
▼ TestSuite 1	6% (0%)	 196/3299
▶ TestCase 1	6% (0%)	 196/3299

Provide API Metadata

Understanding –
more than the raw
interface

Describe flows,
data types, life
cycles



Provide API Metadata

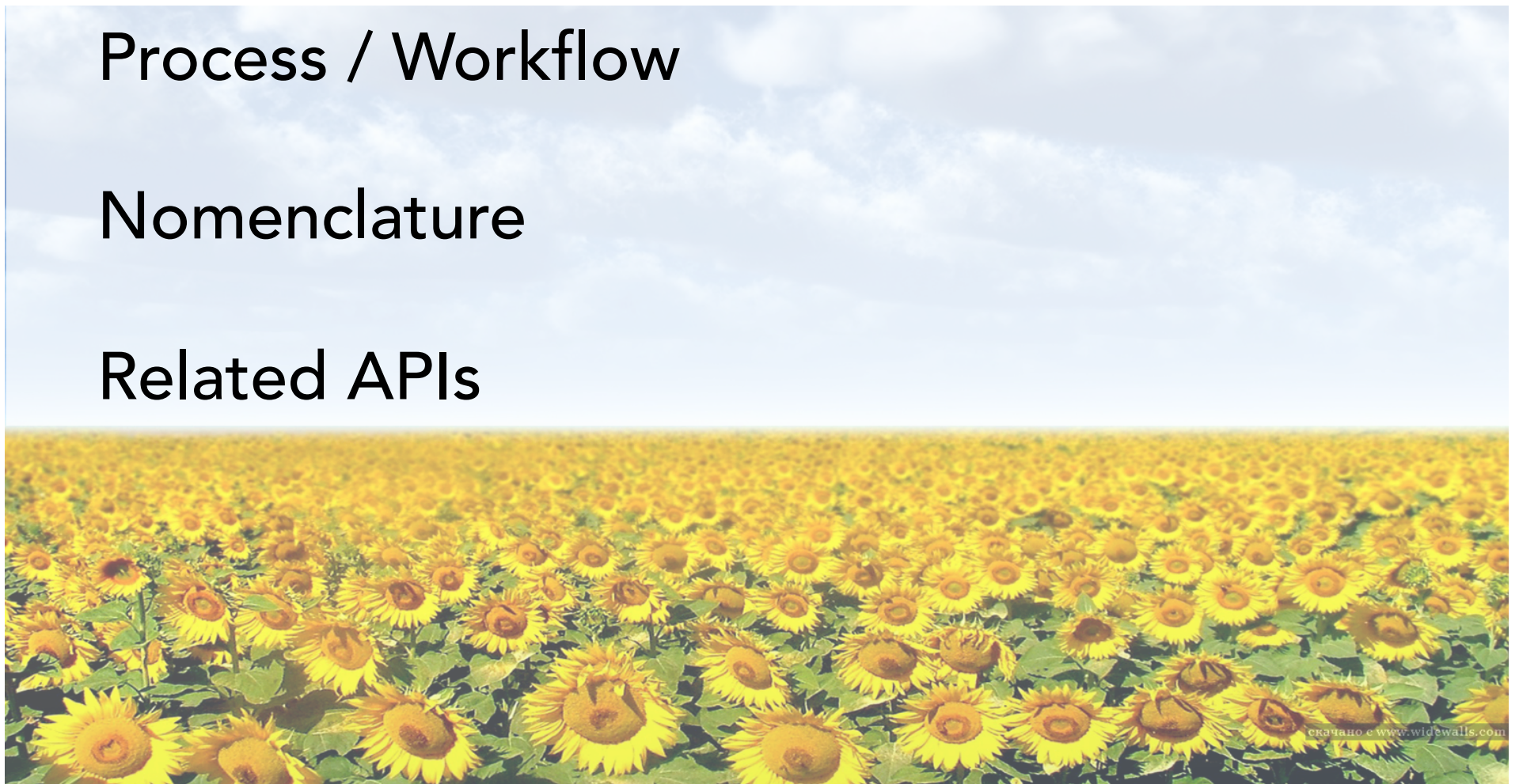
Simulation of your
API can be created
easier

Align your API with their domain

Process / Workflow

Nomenclature

Related APIs



A **3:30:3** Litmus test for APIs

3 Minutes to understand what an API does

30 seconds to sign up

3 minutes to the first request

(Ori Pekelman)



Summary Part I

- 1) APIs drive the new economy
- 2) Testers need to step in early to ensure the usability of the API, so it becomes part of the new economy
- 3) The usability of the API is the key to adoption of your api

```
public class TcpClientSample
{
    public static void Main()
    {
        byte[] data = new byte[1024]; string input, stringData;
        TcpClient server;
        try{
            server = new TcpClient(" . . . . ", port);
        }catch (SocketException){
            Console.WriteLine("Unable to connect to server");
            return;
        }
        NetworkStream ns = server.GetStream();
        int recv = ns.Read(data, 0, data.Length);
        stringData = Encoding.ASCII.GetString(data, 0, recv);
        Console.WriteLine(stringData);
        while (true) {
            input = Console.ReadLine();
            if (input == "exit") break;
            newchild.Properties["ou"].Add(
                "Auditing Department");
            newchild.CommitChanges();
            newchild.Close();
        }
    }
}
```

APIs need to work

Expected Results

```
"routes": [ {  
  "bounds": {  
    "northeast": {  
      "lat": 45.5450791,  
      "lng": -73.55  
    },  
    "southwest": {  
      "lat": 43.6532245,  
      "lng": -79.38  
    }  
  },  
  "copyrights": "Map data ©2013 Google",  
  "legs": [ {  
    "distance": {  
      "text": "627 km",  
      "value": 627411  
    },  
    "duration": {  
      "text": "1 day 8 hours",  
      "value": 116871  
    },  
    "end_address": "Montreal, QC, Canada",  
    "endLocation": {  
      "lat": 45.5085712,  
      "lng": -73.5537674  
    },  
  },  
],  
}
```

Expected Results

Values

```
  "routes": [ {
    "bounds": {
      "northeast": {
        "lat": 45.5450791,
        "lng": -73.55
      },
      "southwest": {
        "lat": 43.6532245,
        "lng": -79.38
      }
    },
    "copyrights": "Map data ©2013 Google",
    "legs": [ {
      "distance": {
        "text": "627 km",
        "value": 627411
      },
      "duration": {
        "text": "1 day 8 hours",
        "value": 116871
      },
      "end_address": "Montreal, QC, Canada",
      "endLocation": {
        "lat": 45.5085712,
        "lng": -73.5537674
      },
    },
  ],
}
```

Expected Results

Values

Datatypes

```
"routes": [ {  
  "bounds": {  
    "northeast": {  
      "lat": 45.5450791,  
      "lng": -73.55  
    },  
    "southwest": {  
      "lat": 43.6532245,  
      "lng": -79.38  
    }  
  },  
  "copyrights": "Map data ©2013 Google",  
  "legs": [ {  
    "distance": {  
      "text": "627 km",  
      "value": 627411  
    },  
    "duration": {  
      "text": "1 day 8 hours",  
      "value": 116871  
    },  
    "end_address": "Montreal, QC, Canada",  
    "endLocation": {  
      "lat": 45.5085712,  
      "lng": -73.5537674  
    }  
  },  
  ],  
  "start_address": "Montreal, QC, Canada",  
  "startLocation": {  
    "lat": 45.5085712,  
    "lng": -73.5537674  
  }  
},  
],  
"status": "OK",  
"warnings": []  
}
```

Expected Results

Values

Datatypes

Formatting

```
"routes": [ {  
  "bounds": {  
    "northeast": {  
      "lat": 45.5450791,  
      "lng": -73.55  
    },  
    "southwest": {  
      "lat": 43.6532245,  
      "lng": -79.38  
    }  
  },  
  "copyrights": "Map data ©2013 Google",  
  "legs": [ {  
    "distance": {  
      "text": "627 km",  
      "value": 627411  
    },  
    "duration": {  
      "text": "1 day 8 hours",  
      "value": 116871  
    }  
  },  
  "end_address": "Montreal, QC, Canada",  
  "endLocation": {  
    "lat": 45.5085712,  
    "lng": -73.5537674  
  },  
}
```

Expected Results

Values

Datatypes

Formatting

Consistency

```
"routes": [ {  
  "bounds": {  
    "northeast": {  
      "lat": 45.5450791,  
      "lng": -73.55  
    },  
    "southwest": {  
      "lat": 43.6532245,  
      "lng": -79.38  
    }  
  },  
  "copyrights": "Map data ©2013 Google",  
  "legs": [ {  
    "distance": {  
      "text": "627 km",  
      "value": 627411  
    },  
    "duration": {  
      "text": "1 day 8 hours",  
      "value": 116871  
    }  
  },  
  "end_address": "Montreal, QC, Canada",  
  "endLocation": {  
    "lat": 45.5085712,  
    "lng": -73.5537674  
  },  
}
```

Expected Results

Values

Datatypes

Formatting

Consistency

Errors

```
"routes": [ {  
  "bounds": {  
    "northeast": {  
      "lat": 45.5450791,  
      "lng": -73.55  
    },  
    "southwest": {  
      "lat": 43.6532245,  
      "lng": -79.38  
    }  
  },  
  "copyrights": "Map data ©2013 Google",  
  "legs": [ {  
    "distance": {  
      "text": "627 km",  
      "value": 627411  
    },  
    "duration": {  
      "text": "1 day 8 hours",  
      "value": 116871  
    }  
  },  
  "end_address": "Montreal, QC, Canada",  
  "endLocation": {  
    "lat": 45.5085712,  
    "lng": -73.5537674  
  },  
}
```

Bad Error Messages...

Change unexpectedly

Don't match their status code (HTTP)

Don't tell users what they did wrong

Don't tell users what they need to do right

Show clients stuff that could be misused

```
No access to table Sales.
```

APIs are global from day 1

Dates & Timezones

Regional Formatting

Localized Messages / Errors



ConsistencyConsistencyConsistencyCon
sistencyConsistencyConsistencyConsist
encyConsistencyConsistencyConsistenc

Formatting

Nomenclature

Metadata Compliance



API Functional Test Automation

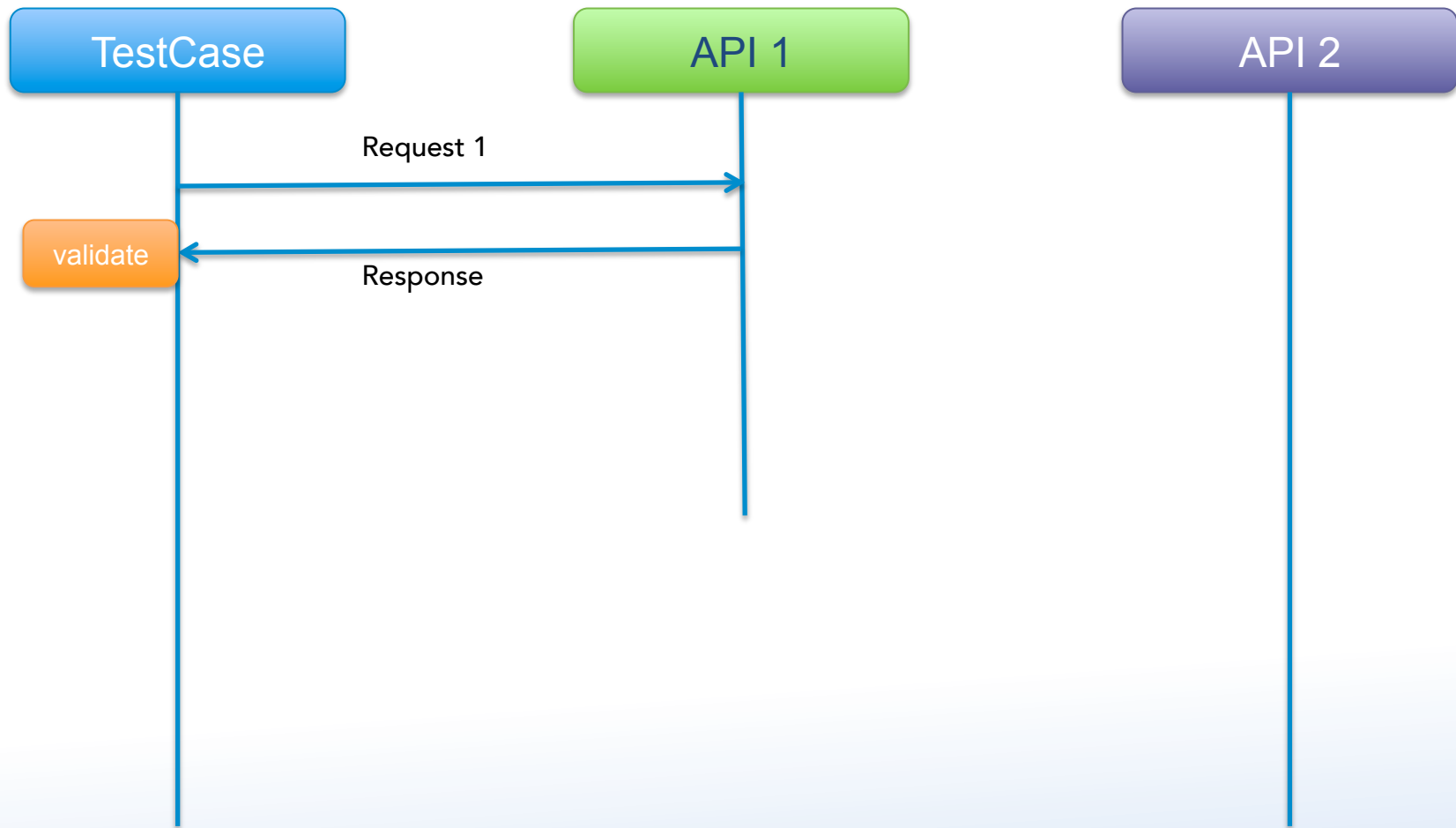
TestCase

API 1

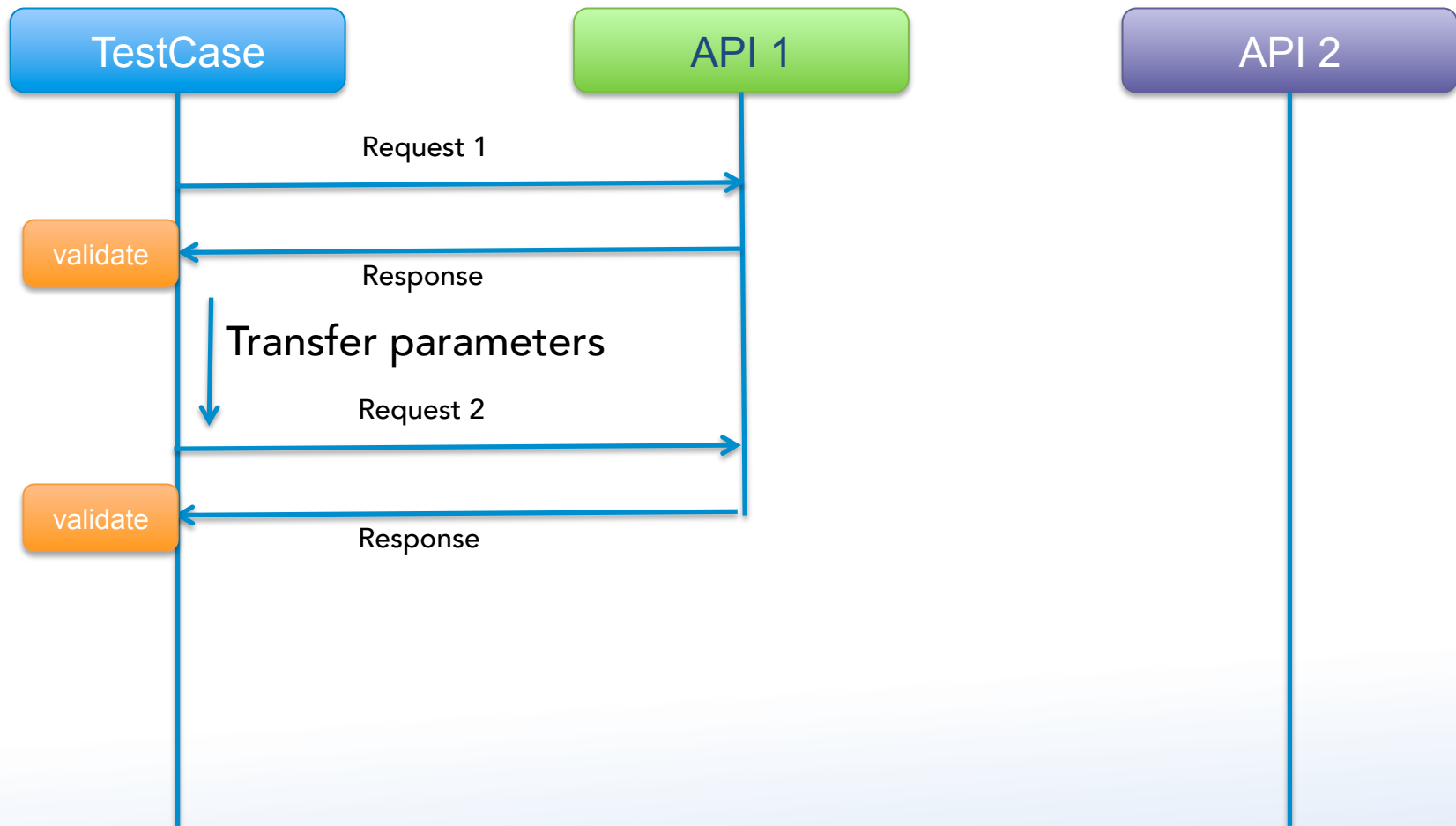
API 2



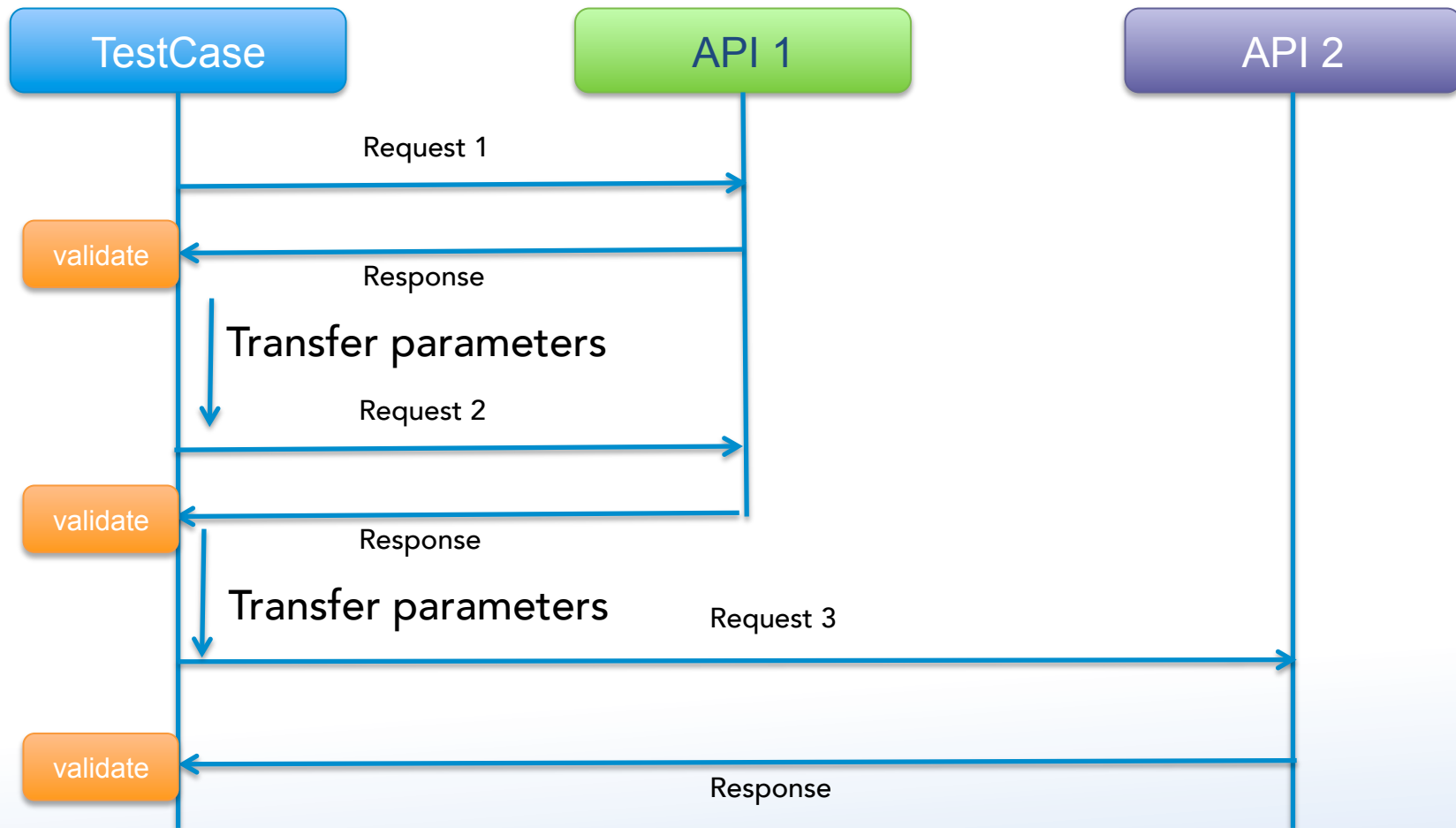
API Functional Test Automation



API Functional Test Automation



API Functional Test Automation



Summary Part II

Functional testing

Automated, scenario based testing



Learn more on Functional testing at <http://www.soapui.org/Dojo/overview.html>



Welcome to the Dojo ▾



Overview

- World Of API Testing ▶
- Best Practices ▶
- Testing Katas ▶
- Resources ▶

What is the API Testing Dojo?

We know that testing software is very important. As our lives become more intertwined with software enabled devices, the more important testing becomes. Businesses understand the merit in testing to increase overall software quality, decrease the amount of bugs, save time and money, and ultimately to improve brand reputation. Yet, although software testing gains prominence, API testing remains an obscure, undervalued, and often misunderstood practice.

The API Testing Dojo is a space to learn and hone your skills in API testing. It's meant for developers of APIs, professional testers, and anyone else who might need to determine the efficacy of an Application Programming Interface or web service. It's organized into three main sections:

藩

World of API Testing

This section is an introduction to the domain of

練習

Best Practices

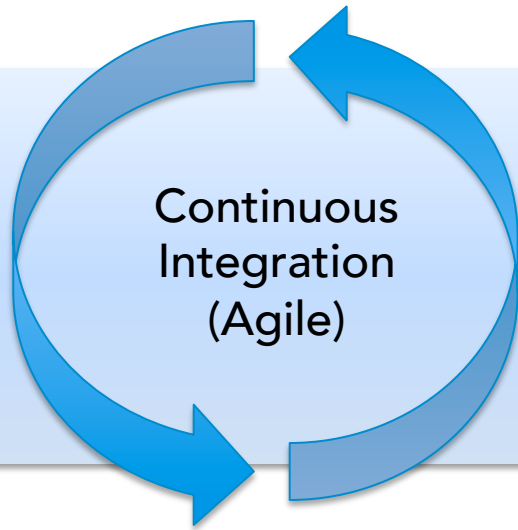
The Best Practices area is dedicated to describing API

型

Testing Katas

The Testing Katas lets you assess your API testing

Pre vs Post-Deployment Quality



Automated
Test Execution

Pre vs Post-Deployment Quality

months/weeks/
days/hours

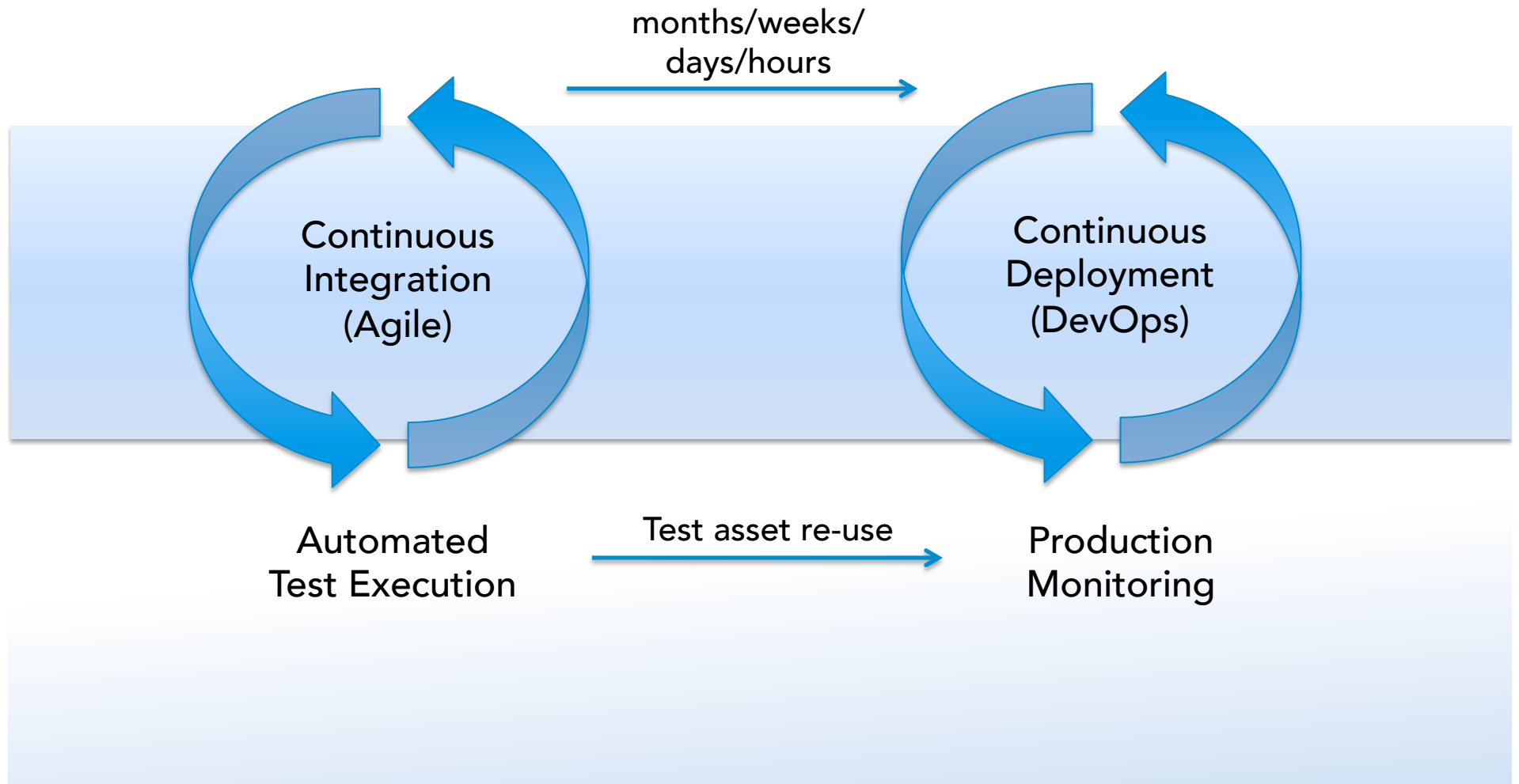


Continuous
Integration
(Agile)

Automated
Test Execution

Continuous
Deployment
(DevOps)

Pre vs Post-Deployment Quality



A collection of 3D rendered human figures in a light blue color, scattered across a light blue background. Each figure is carrying a briefcase, suggesting a business or professional context. The figures are positioned at various angles and distances, creating a sense of a busy, diverse group of people.

Customers rely on your API

Your API is key to *their* success

Is your API available?

Is your API consistent?

Is your API transparent?



Is your API Secure?

Do you handle identity?

Do you maintain integrity?

Do you assess vulnerabilities?

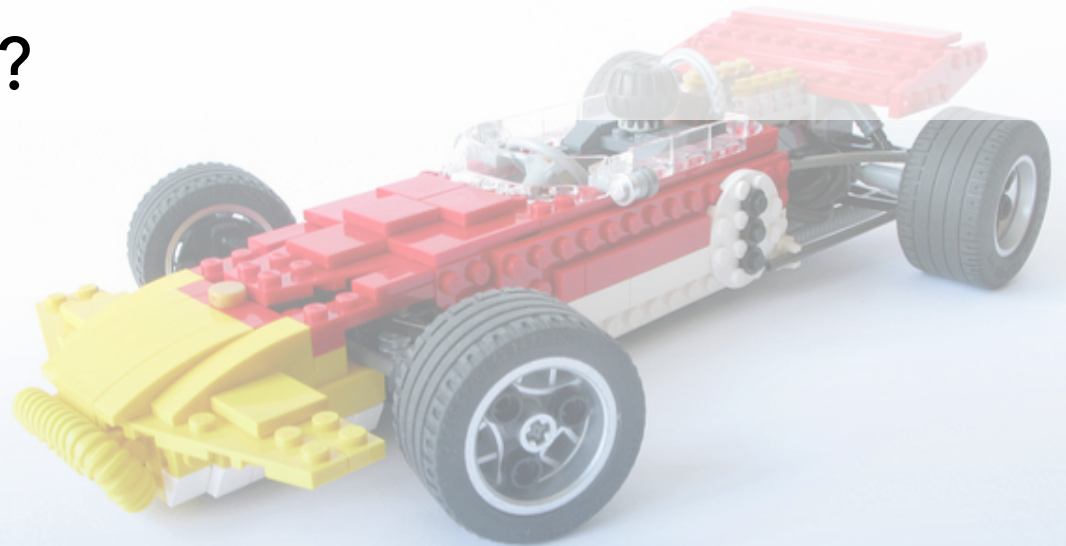


API Performance

Does the API perform consistently?

Does the API recover?

Does the API scale?



Core elements of API Quality

Usability



Functionality

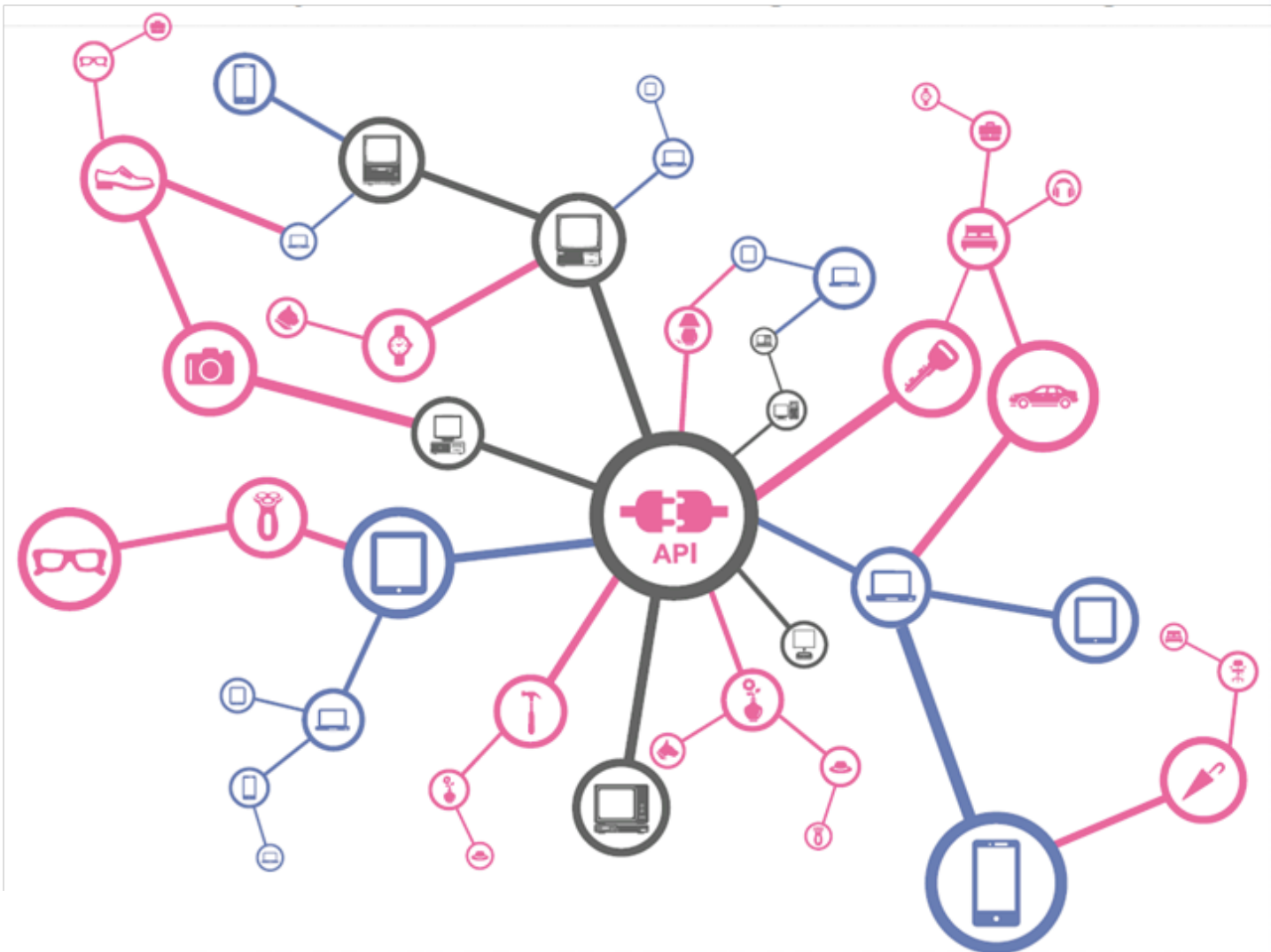
```
public class TcpClientSample
{
    public static void Main()
    {
        byte[] data = new byte[1024]; string input, stringData;
        TcpClient server;
        try{
            server = new TcpClient(" . . . . ", port);
        }catch (SocketException){
            Console.WriteLine("Unable to connect to server");
            return;
        }
        NetworkStream ns = server.GetStream();
        int recv = ns.Read(data, 0, data.Length);
        stringData = Encoding.ASCII.GetString(data, 0, recv);
        Console.WriteLine(stringData);
        while(true){
            input = Console.ReadLine();
            if (input == "exit") break;
            newchild.Properties["ou"].Add("Auditing Department");
            if (input == "Auditing Department")
                newchild.CommitChanges();
            newchild.Close();
        }
    }
}
```

Reliability



so – let's get back to that goal...







Questions

Matti and Mike