

# Continuous Delivery in Application Economy

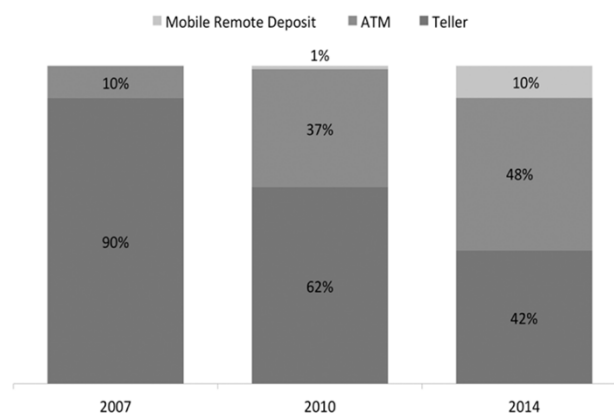
Karel Schmidtmayer

April 24<sup>th</sup> 2018

CA EXPO



JPMorgan Chase — Consumer Deposits By Channel



Source: JPMorgan Chase

BI INTELLIGENCE



Every Company is Becoming a Technology Company, Every Business a Digital Business

DIGITAL TRANSFORMATION

Re-thinking how the business operates to find new ways to create and deliver customer value



Eliminate constraints and critical dependencies



Rapidly deliver high-quality applications and services



Create and deliver exceptional customer value

CONTINUOUS DELIVERY

## Chaotic and Complex Application Delivery

Developers Deadlocked

Constrained Environments

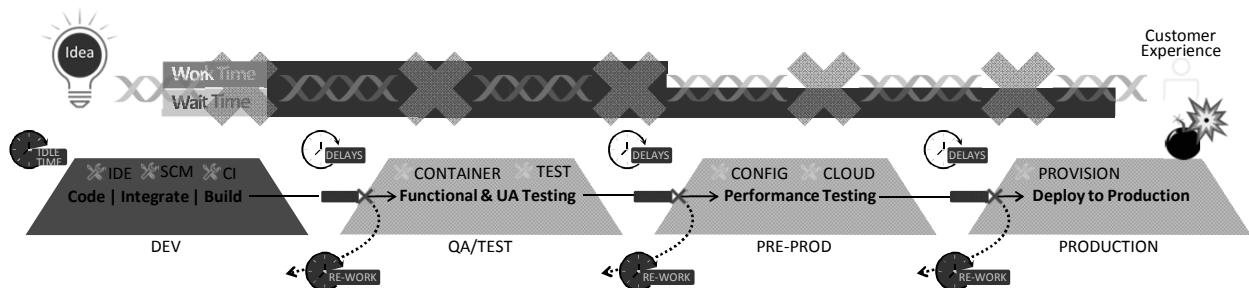
Manual Testing

Too little testing, too late in the cycle

Too many disparate tools

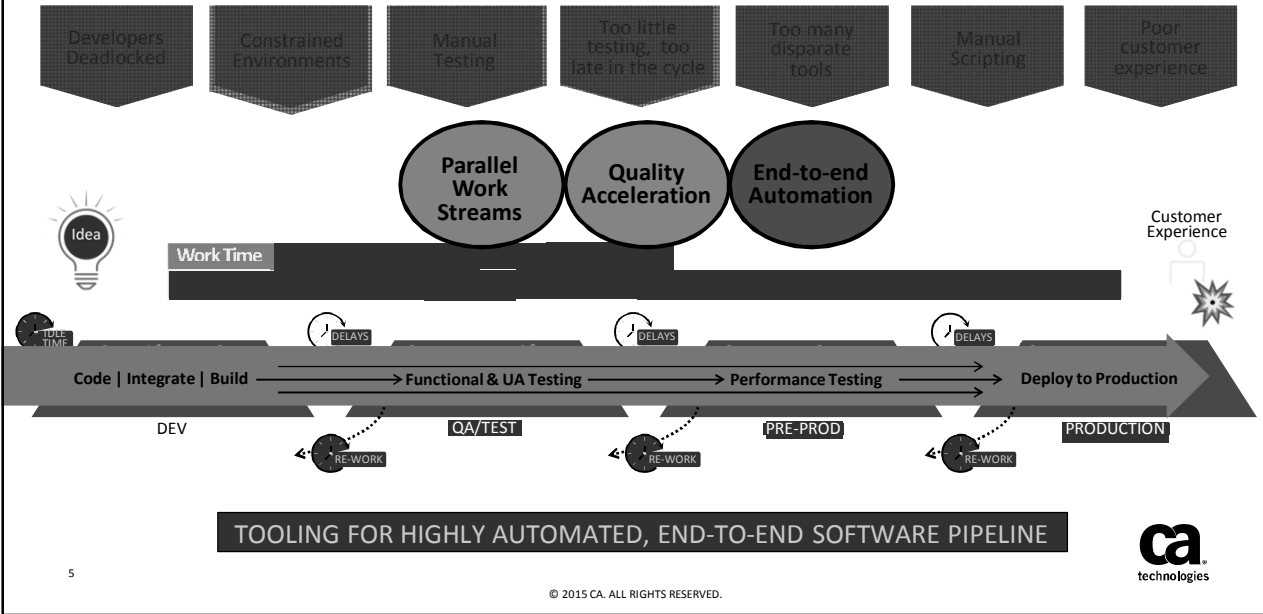
Manual Scripting

Poor customer experience



SLOW AND INEFFICIENT - EXCESSIVE IDLE TIME, DELAYS AND RE-WORK

# Continuous Delivery



5

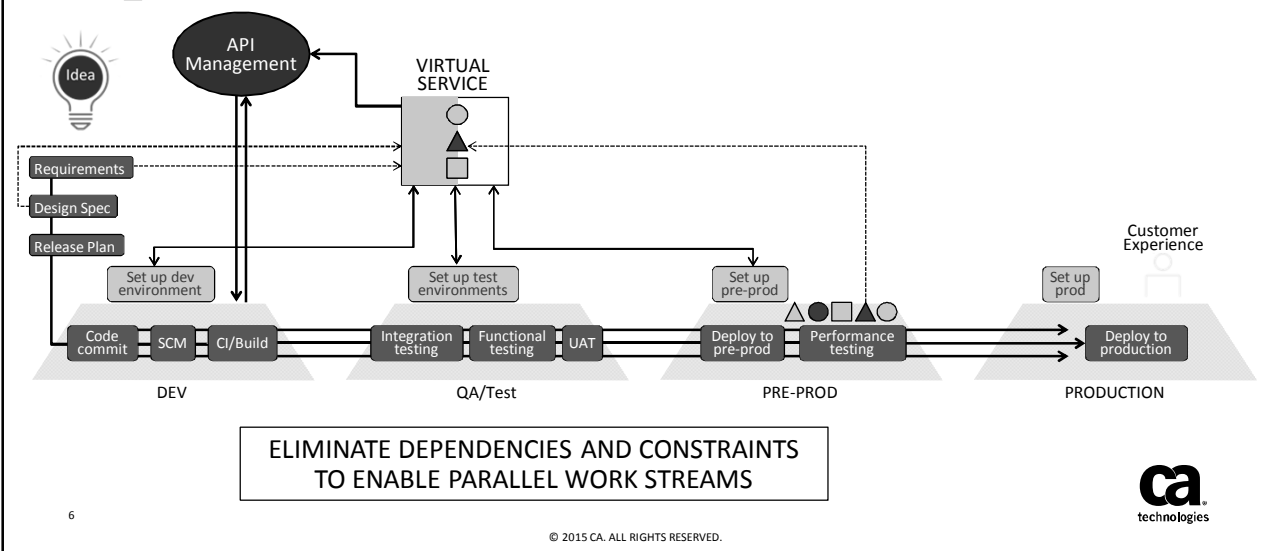
© 2015 CA. ALL RIGHTS RESERVED.

## Parallel Work Streams

Accurately simulate constrained systems

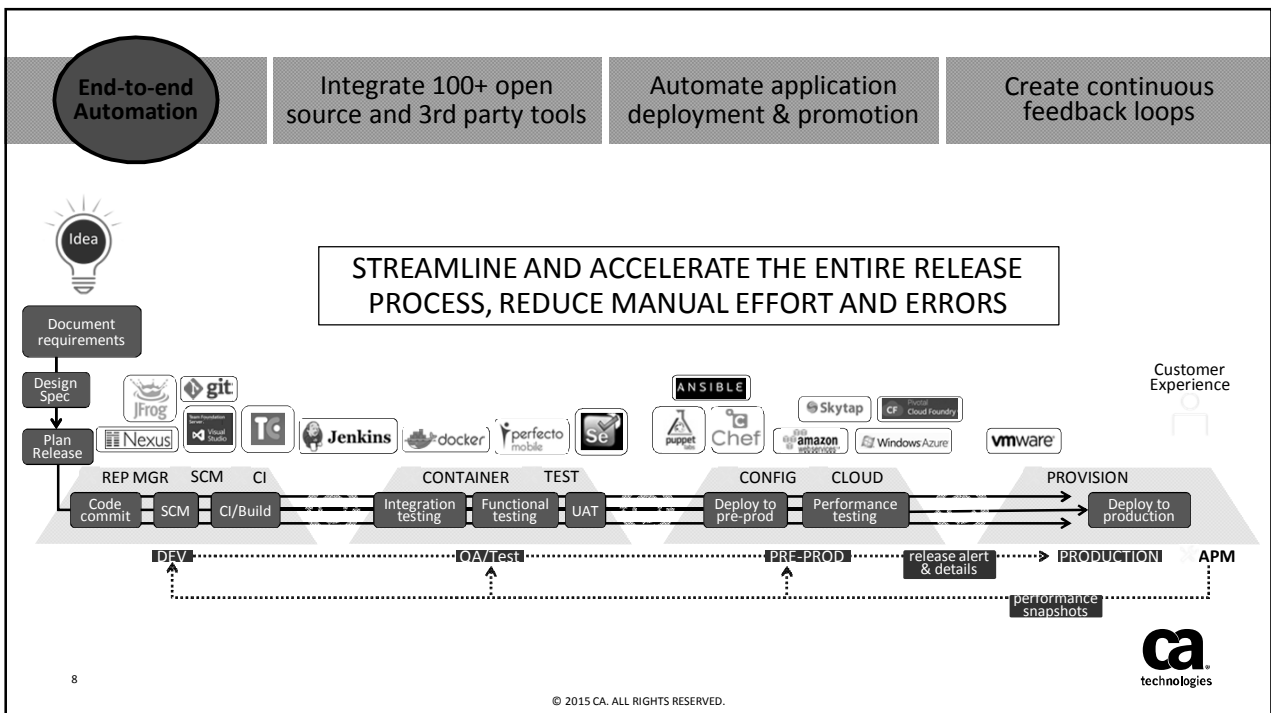
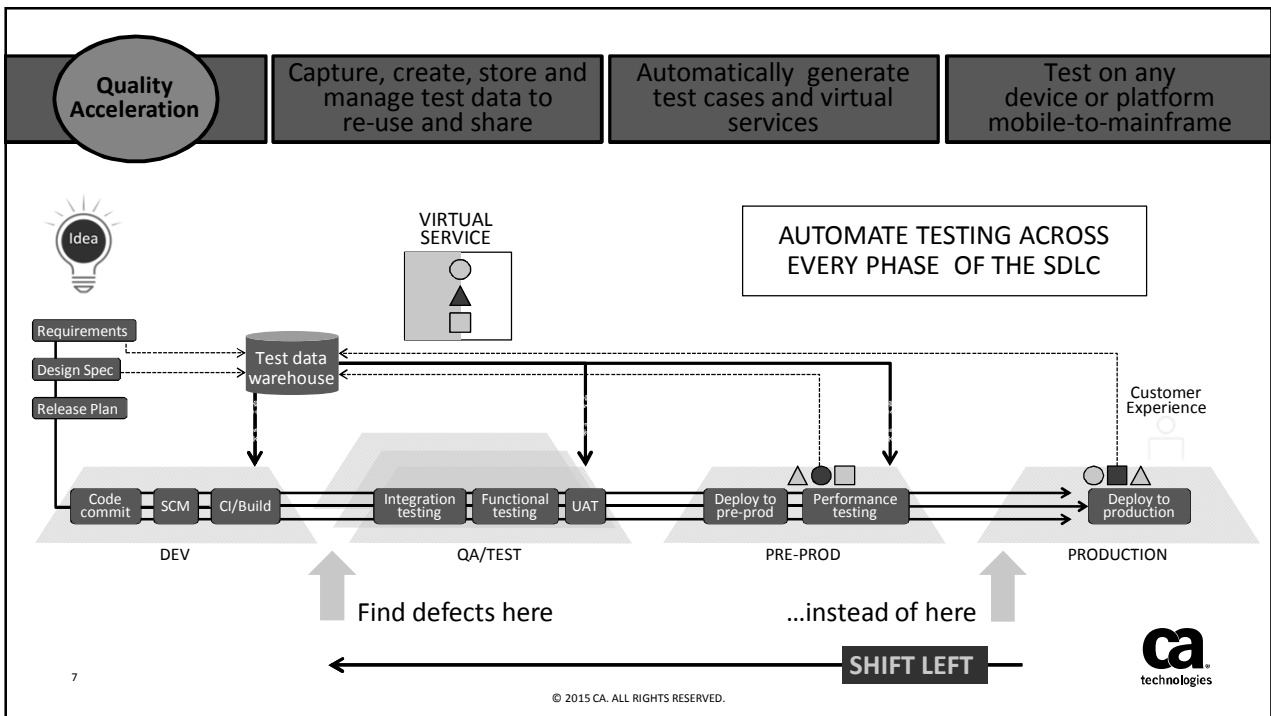
Quickly create, edit and deploy virtual services

Virtualize APIs and manage access

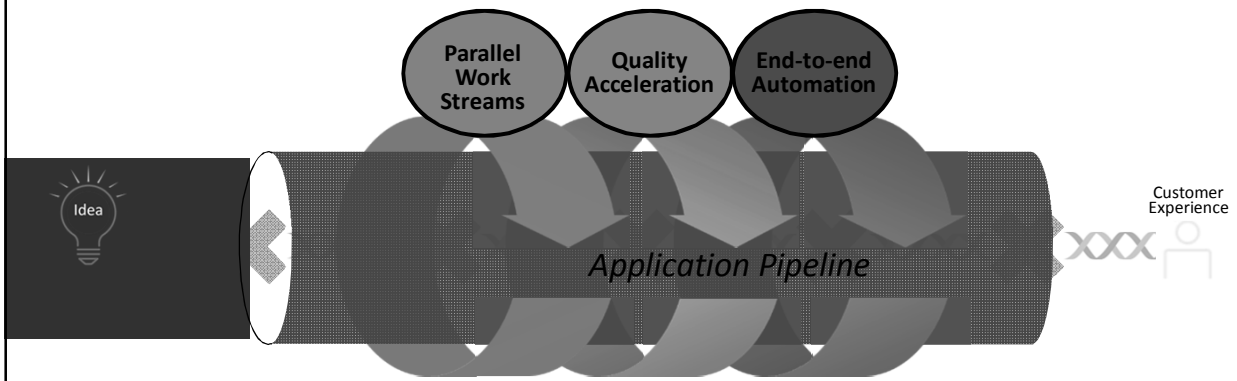


6

© 2015 CA. ALL RIGHTS RESERVED.



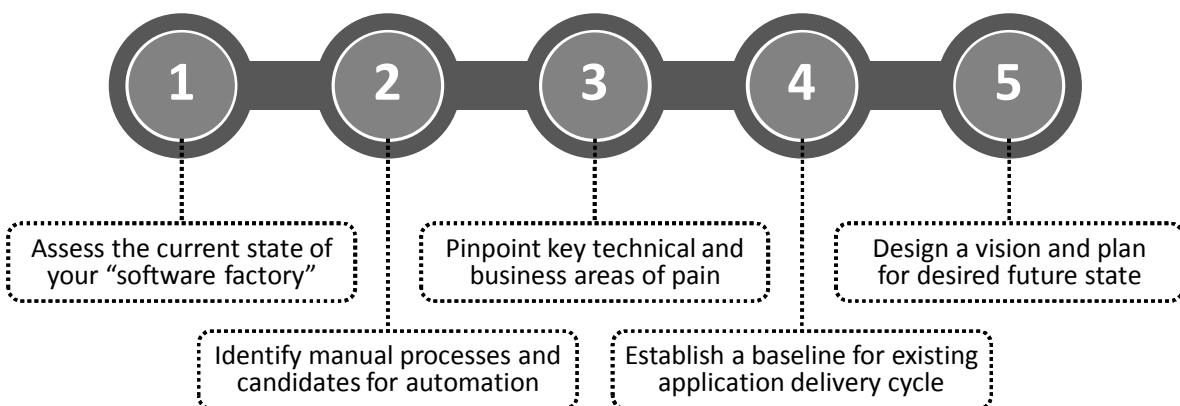
## Continuous Delivery REDEFINED



9

© 2015 CA. ALL RIGHTS RESERVED.

## Where do You Start?

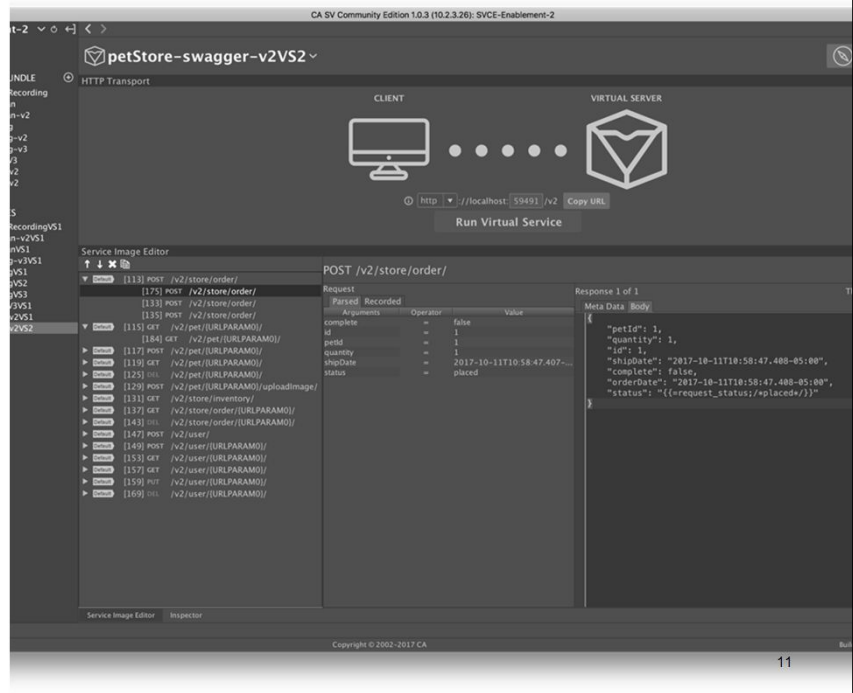


10

© 2015 CA. ALL RIGHTS RESERVED.

## For testers: SV Community Edition

<https://communities.ca.com/community/ca-devtest-community/pages/sv-community-edition>



**CodeSV**  
BY CA TECHNOLOGIES

Search the site

## For developers: CodeSV

<http://codesv.io>

Develop. Virtualize. Commit.

CodeSV is a **free, lightweight, Java library for developers** that provides a simple fluent API enabling you to easily define and setup virtual services directly in your **unit testing** code.

**Boost your productivity** by simulating services that your code depends on - especially when those services don't exist or are expensive to use.

Test **edge cases** and negative scenarios that are hard to reproduce with different services.

**CodeSV goes way beyond mocking** and empowers you to test the parts of your application that other mocking frameworks fail to reach.

Download

```
> @Rule
> public VirtualServerRule vs = new VirtualServerRule();
>
> @Test
> public void
testSimpleHttpGetWithResponseCodeAndStringBody() throws
IOException {
>     forGet(URL).doReturn(
>         aMessage(CUSTOM_STATUS_CODE)
>             .withStringBody(RESPONSE_BODY_GET)
>     );
>
>     HttpGet httpGet = new HttpGet(URL);
>     HttpClient httpClient =
HttpClientBuilder.create().build();
>     HttpResponse httpResponse = httpClient.execute(httpGet);
>
>     assertEquals(CUSTOM_STATUS_CODE,
httpResponse.getStatusLine().getStatusCode());
}
```

## Virtual Service in CodeSV – simple as that

```
public class ExampleTest {  
  
    @Rule  
    public VirtualServerRule vs = new VirtualServerRule();  
  
    @Test  
    public void testRequest() {  
        forGet(url: "http://www.ca.com").doReturn(  
            okMessage()  
                .withStringBody("Happy virtualizing!")  
        );  
    }  
}
```

13

Navštivte naší R&D zónu

Děkuji za pozornost

14

© 2015 CA. ALL RIGHTS RESERVED.