

# The Future of Testing: *How Service Virtualization Changes the Game in Testing Complex Applications*

*The new frontier for quality management and testing*



## Overview

*Today's complex applications provide a constant challenge for companies to balance the level of quality while ensuring rapid time to market. Development teams are delivering applications faster which is driving the need for change in the world of testing. This session will delve into these pressures and discuss the impact that service virtualization and continuous integration testing can have to improve quality, remove unnecessary costs, and reduce project risk.*

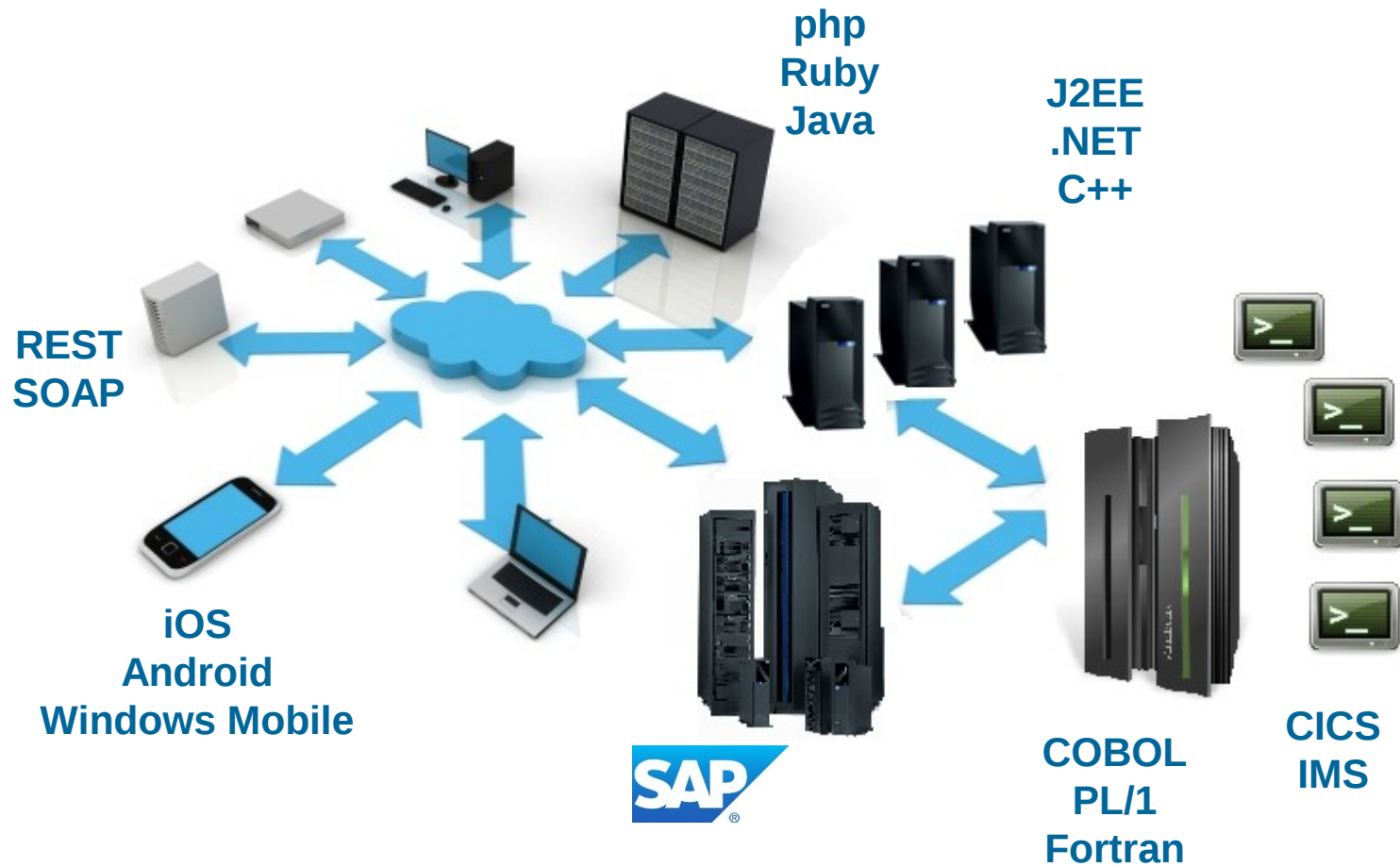
## Agenda

- **Looking back at the past** – *Those were the days!*
- Planning for flexibility
- Avoiding the big bang
- Wrapping up

# 40 years of technology...

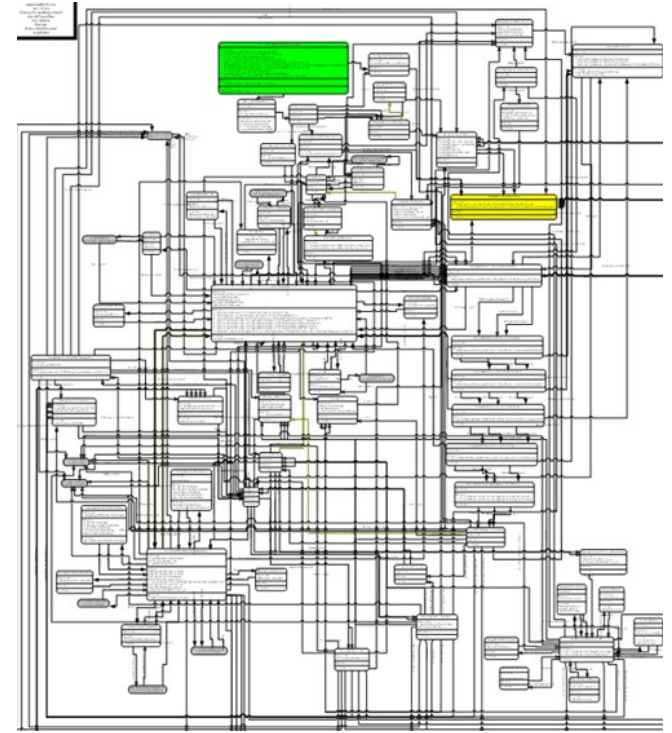


# Are delivering today's cutting edge IT solutions



## Development and Testing Complexity is a top customer concern ...

- Application complexity is exploding:
  - Mainframe (one interface) -> Client Server (a few interfaces) -> Web/SOA/Cloud apps (100's of interfaces)
- Development and Test teams are getting larger and more geographically disperse
- Costs are escalating and quality is suffering
- Waterfall models and serial projects are out, agile and continuous testing are in



***A new approach is required to keep up!***

# Development and test challenges bring opportunities across the software development life cycle

## Challenges

## Opportunities

### Quality

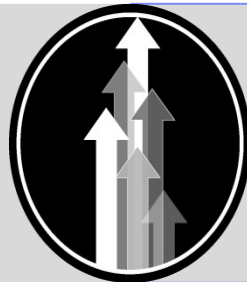
- Inadequate testing and test environments
- Defect escapes
- Test data consistency



- Drive testing earlier, discover defects early when they are cheaper to correct
- Automate testing that matters freeing up resources for other value add activities
- Improve resource governance and control

### Cost

- Escalating labor costs
- High cost of defect fixes
- Poor asset utilization



- Reduce infrastructure required to develop and test applications
- Increase resource flexibility and availability
- Improve asset utilization

### Time To Market

- Long cycle times
- Availability of resources



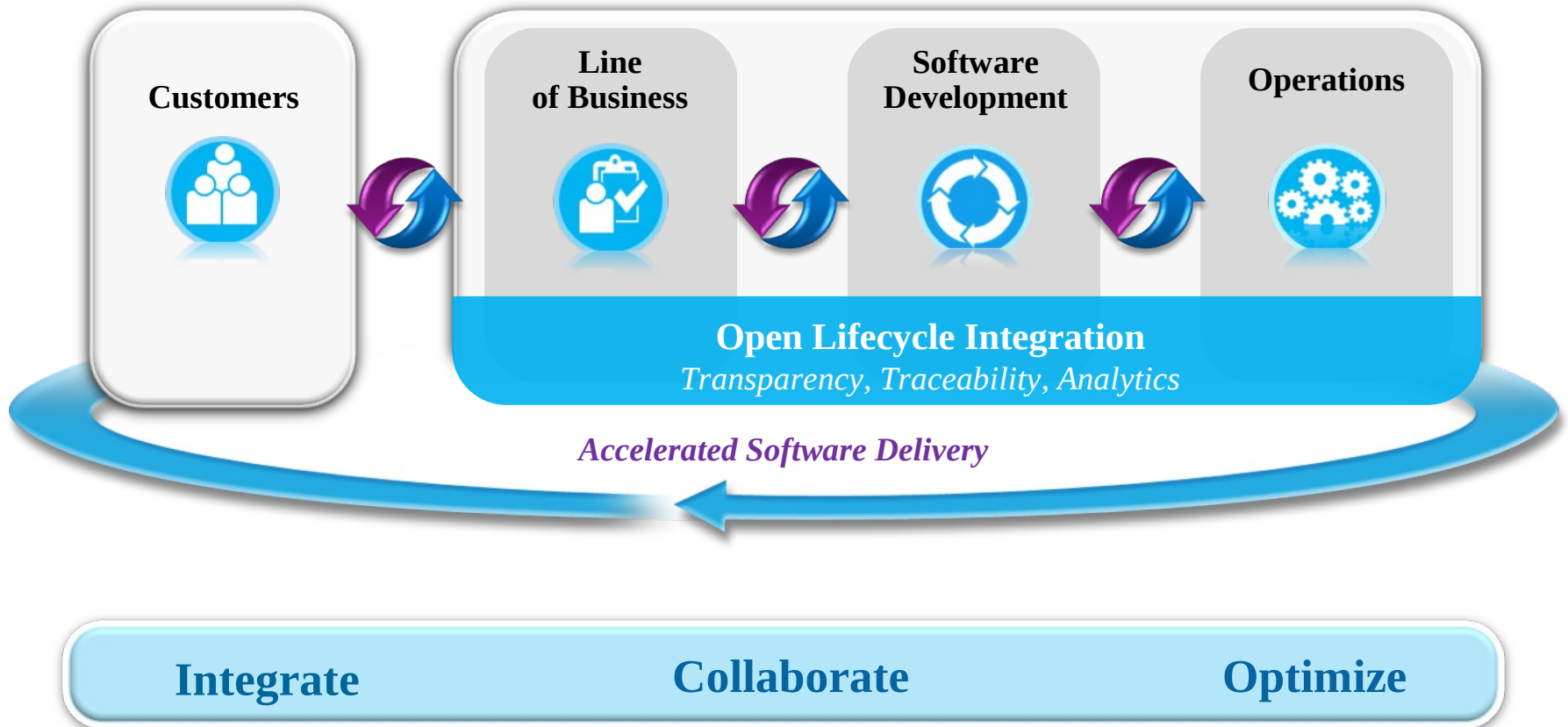
- Provide rapid delivery of services
- Allow testing when only portions of the final solution are available
- Increase levels of testing

## Agenda

- Looking back at the past
- **Planning for flexibility** – *changing the Game!*
- Avoiding the big bang
- Wrapping up

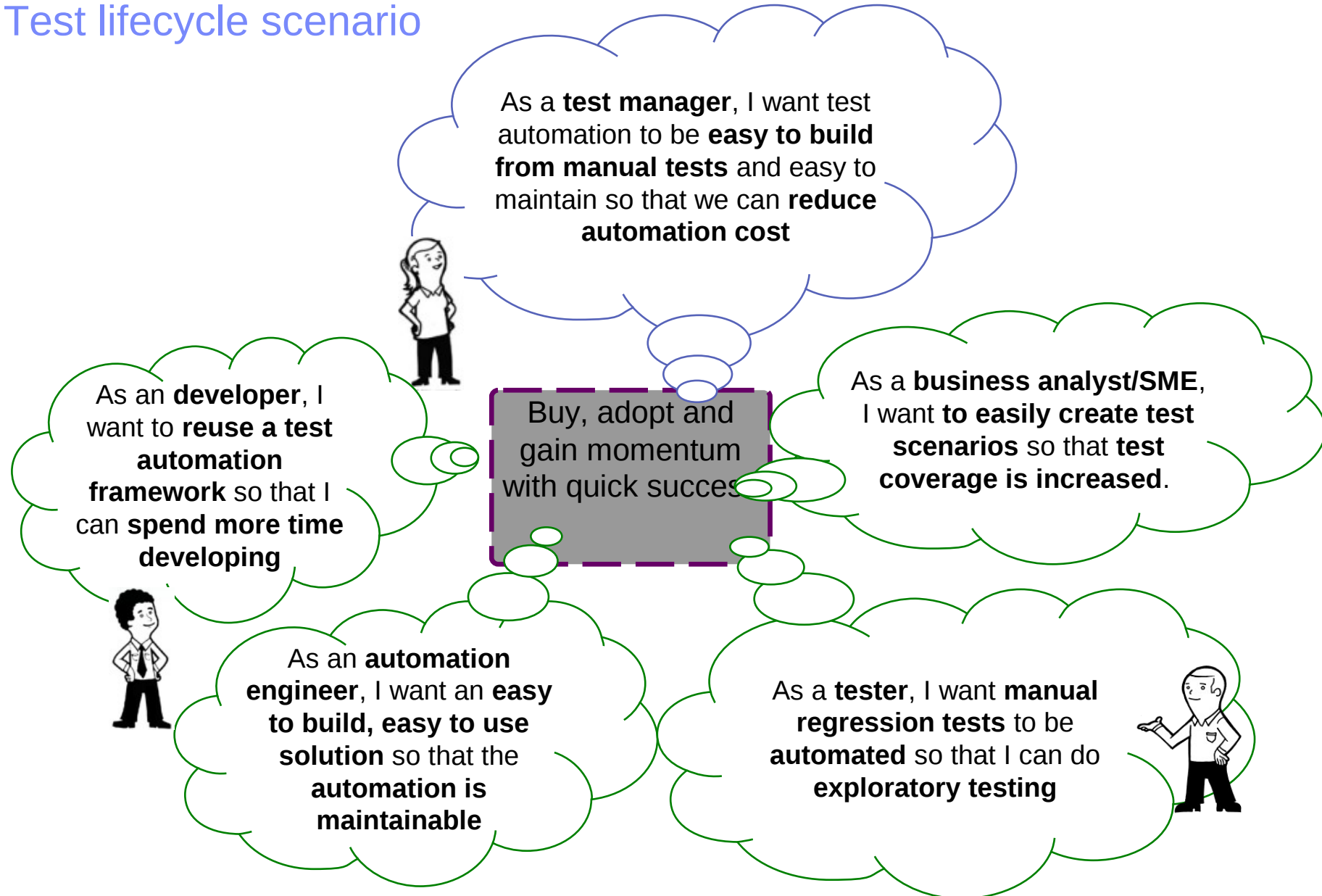


# Accelerating software delivery





## Test lifecycle scenario

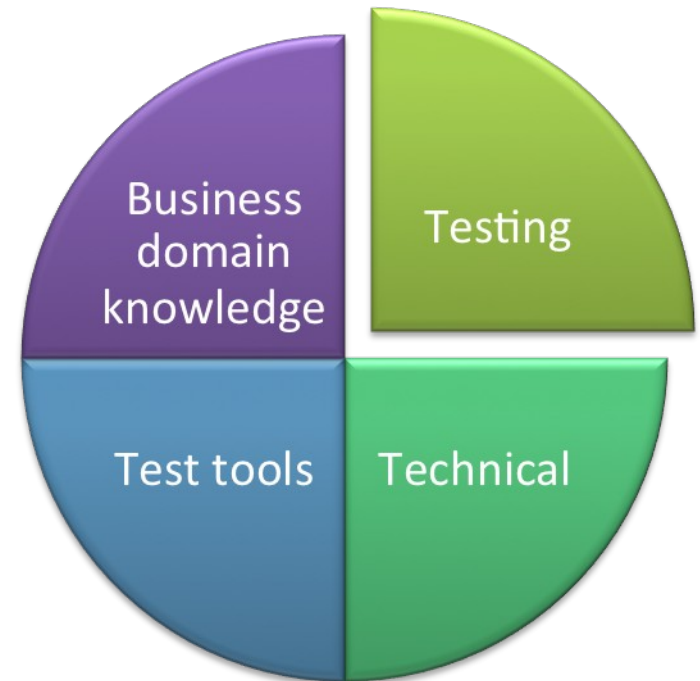


## The Godzilla Tester

Integration testers require:

- Technical knowledge
- Test tools knowledge
- Business domain knowledge
- Testing knowledge

Accelerate the learning process with a collaborative approach.

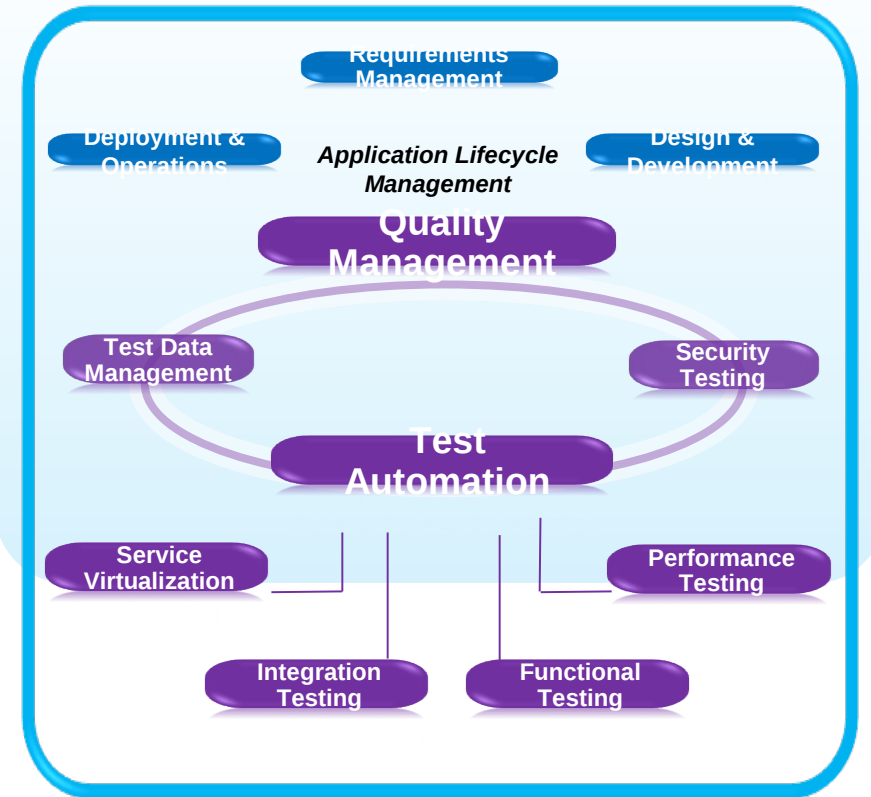


# The new frontier for quality management and testing

Combining IBM software solution capabilities to:

- collaborate in context with real-time test planning
- manage risk by testing smarter
- accelerate continuous delivery with automated testing that matters
- achieve predictability with an integration first approach

## IBM Solution for Quality Management and Testing

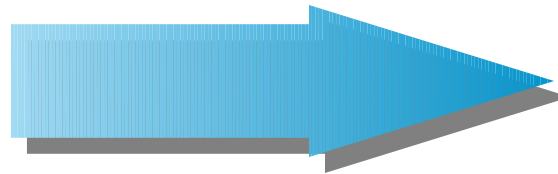


***Changing the game!***

# Shift from testing to quality management – from chaotic to optimized

## Testing

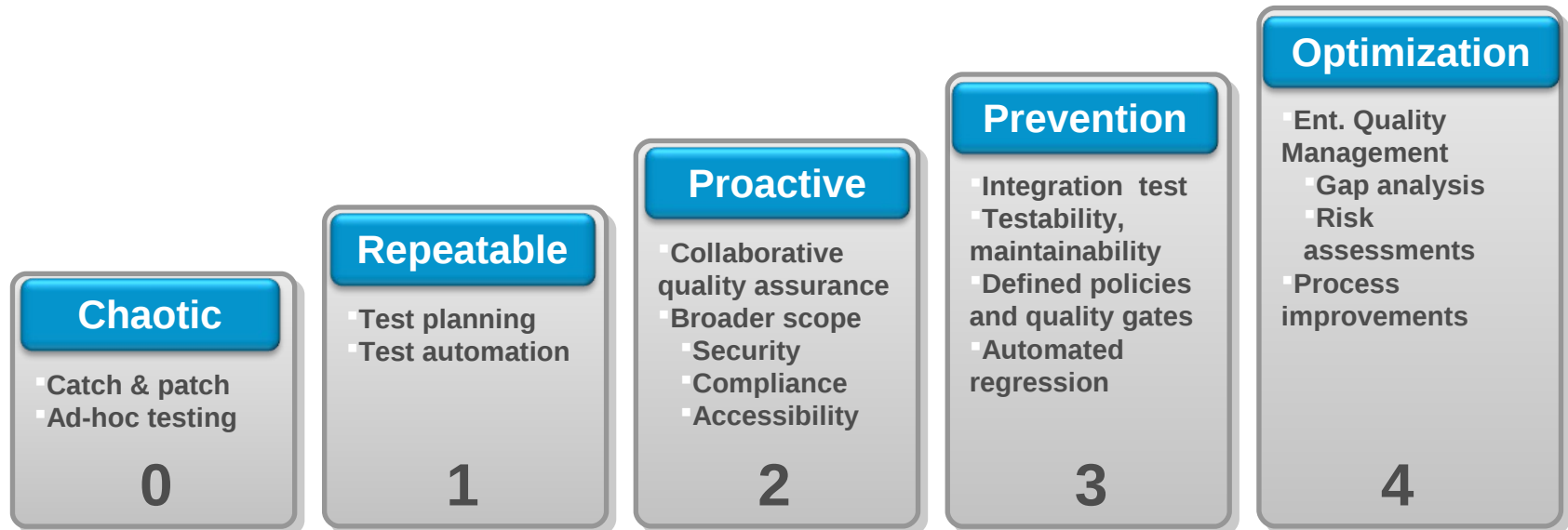
A technical investigation done to expose quality-related information about the product or service under test



*"I have hundreds of testers & lots of automation, but all I do is find more defects. I don't have a testing problem, I have a quality problem." – large global bank*

## Quality Management

Systematic monitoring and evaluation of the various aspects of a product or service, to maximize the probability that target quality standards are being attained



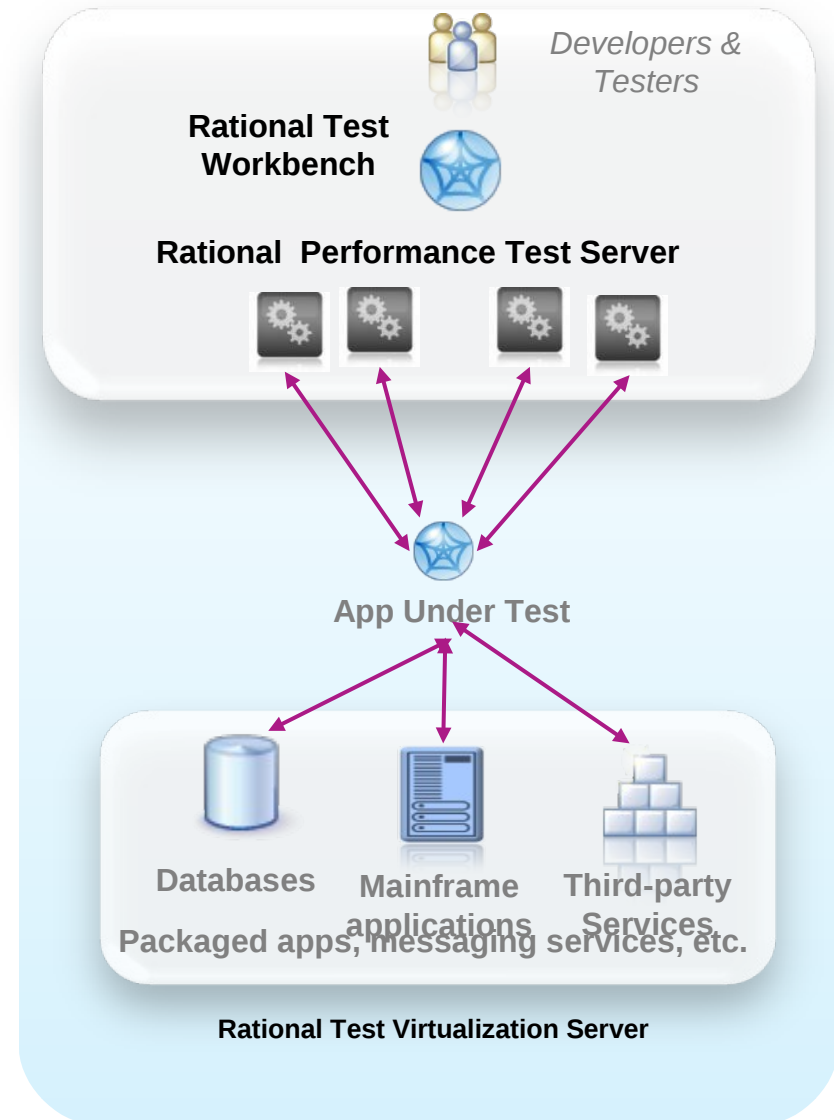
# IBM Rational Test Automation: *automated testing that matters*

- **Rational Test Workbench** is a desktop solution that enables testers/developers to:

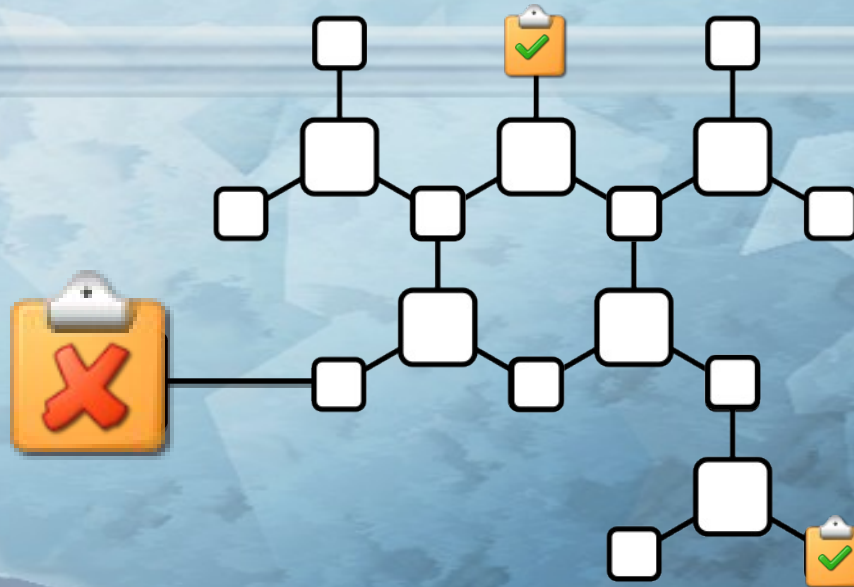
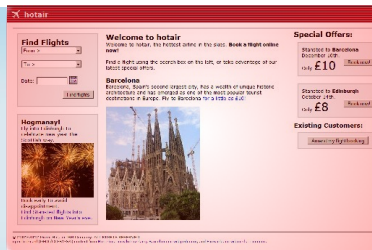
  - Capture and model virtual services
  - Test services and applications long before their user interfaces becomes available and do integration testing (SOA, BPM)
  
- **Rational Test Virtualization Server** is a server solution that:

  - Provides a central environment to virtualize heterogeneous hardware, software and services to provide 24x7 testing capabilities
  - Reduces infrastructure costs of traditional testing environments
  - Virtual Services can be built from the interface definition of the system for a wide variety of protocols, including HTTP, web services, SOA, JMS, TIBCO, IBM WebSphere MQ, Oracle, etc.
  
- **Rational Performance Test Server** enables Rational Test Workbench users to reuse test scripts to drive performance testing:

  - Can be used in combination with Virtual Services
  - Probe for identification of system bottlenecks

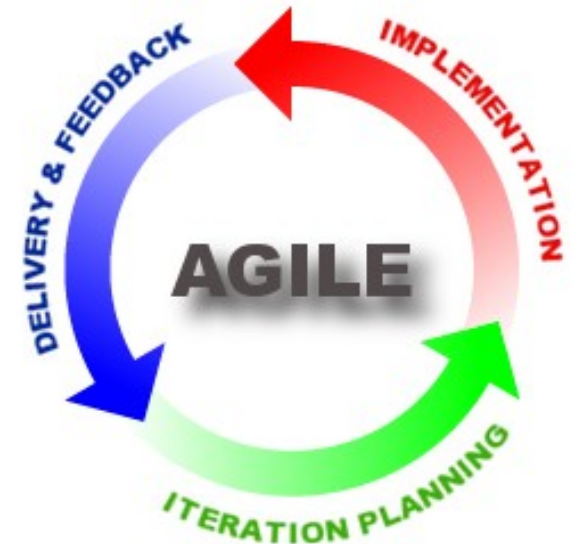


# A better approach = integration testing + service virtualization



## Planning for flexibility – Innovation over Remediation

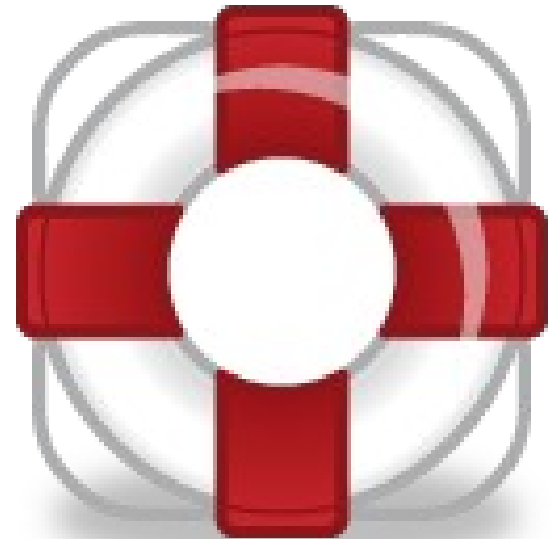
- **Collaborate in context** – *people and interactions over process and tools*
- **Embrace real time planning** – *change is inevitable*
- **Employ traceability across the lifecycle** – *measure impact of change and manage risk*
- **Share the responsibility of quality** – *everyone contributes (analysts, programmers, testers)*
- **Automate testing that matters** – *free up resources for value add activities*
- **Isolate defects at the source** – *faster defect resolutions = shorter, more efficient iterations*
- **Adopt continuous integration testing** - *Get to “Done Done” faster; test applications end to end earlier*
- **Eliminate Agile’s testing bottlenecks** – *Service Virtualization to the rescue*



**Faster time to market, increased quality, reduced costs**

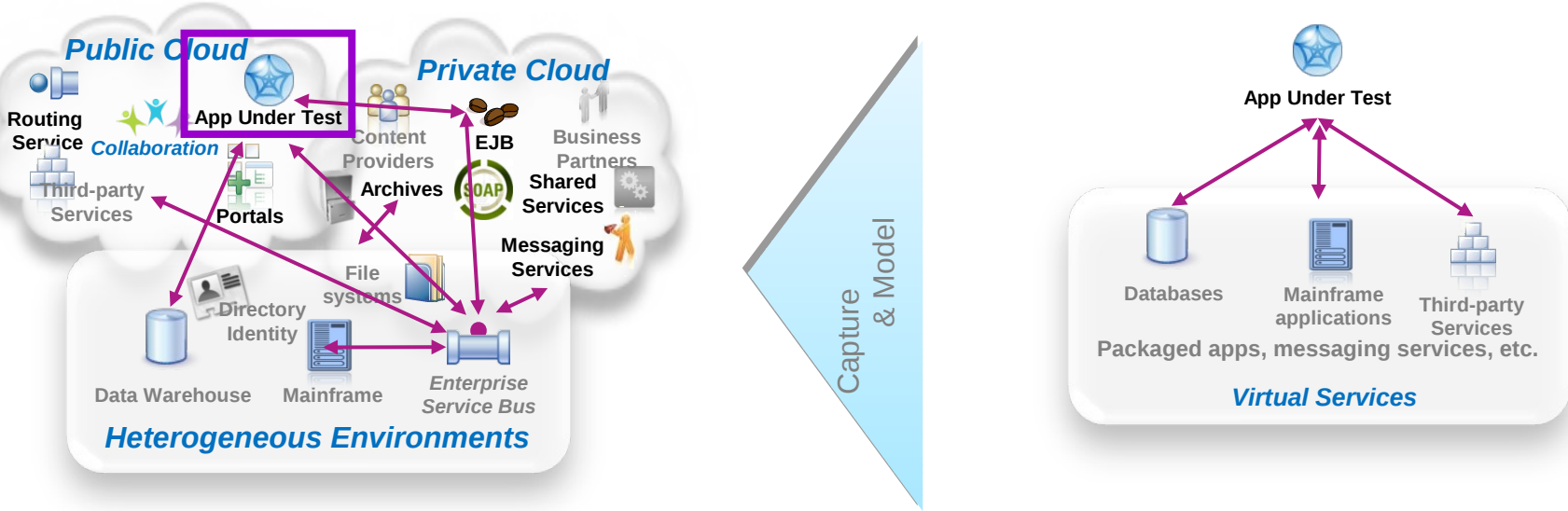
## Agenda

- Looking back at the past
- Planning for flexibility
- **Avoiding the big bang** – *Service Virtualization to the rescue*
- Wrapping up





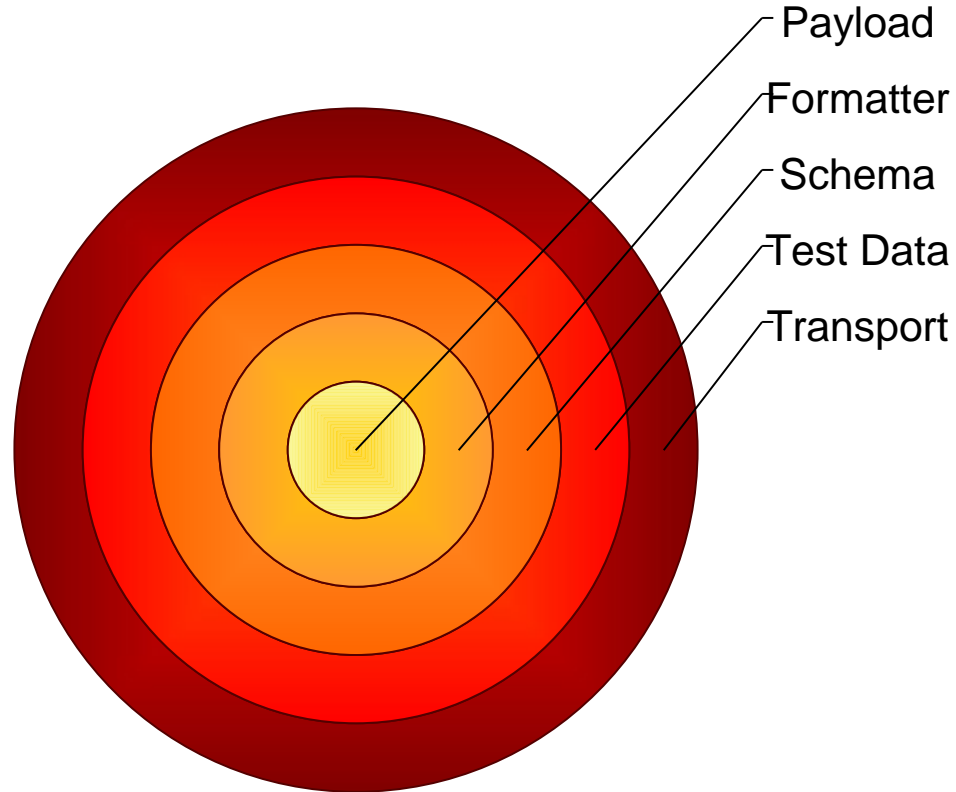
# What is Service Virtualization?



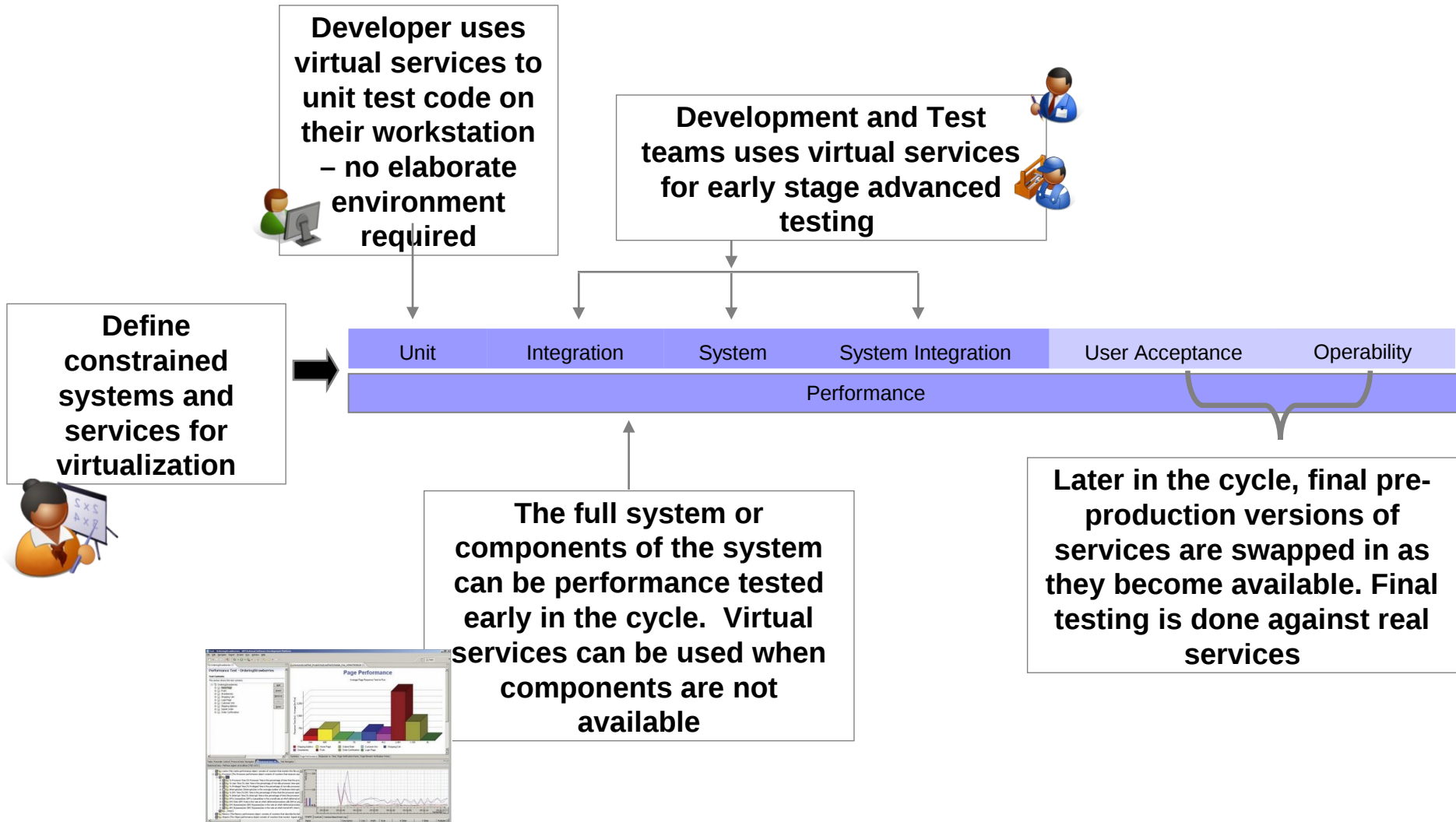
Service Virtualization enables you to create “*virtual services*”:

- *Virtual Services simulate the behavior of an entire application or system during testing*
- *Virtual Services can run on commodity hardware, private cloud, public cloud*
- *Each developer, tester can easily have their own test environment*
- *Developer and testers continue to use their testing tools (Manual, Web performance, UI test automation)*

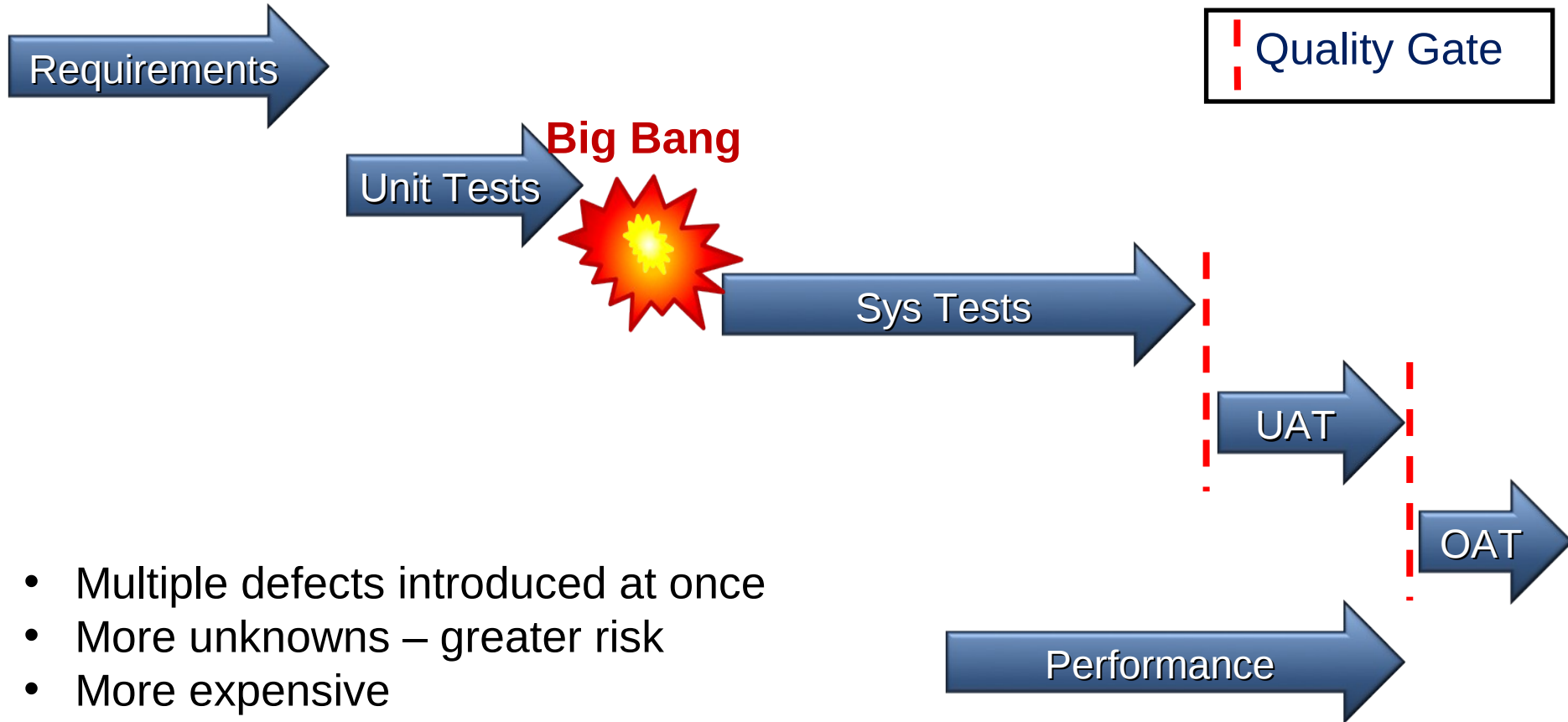
# What are we concerned with?



# Service Virtualization provides benefits across the software delivery life cycle

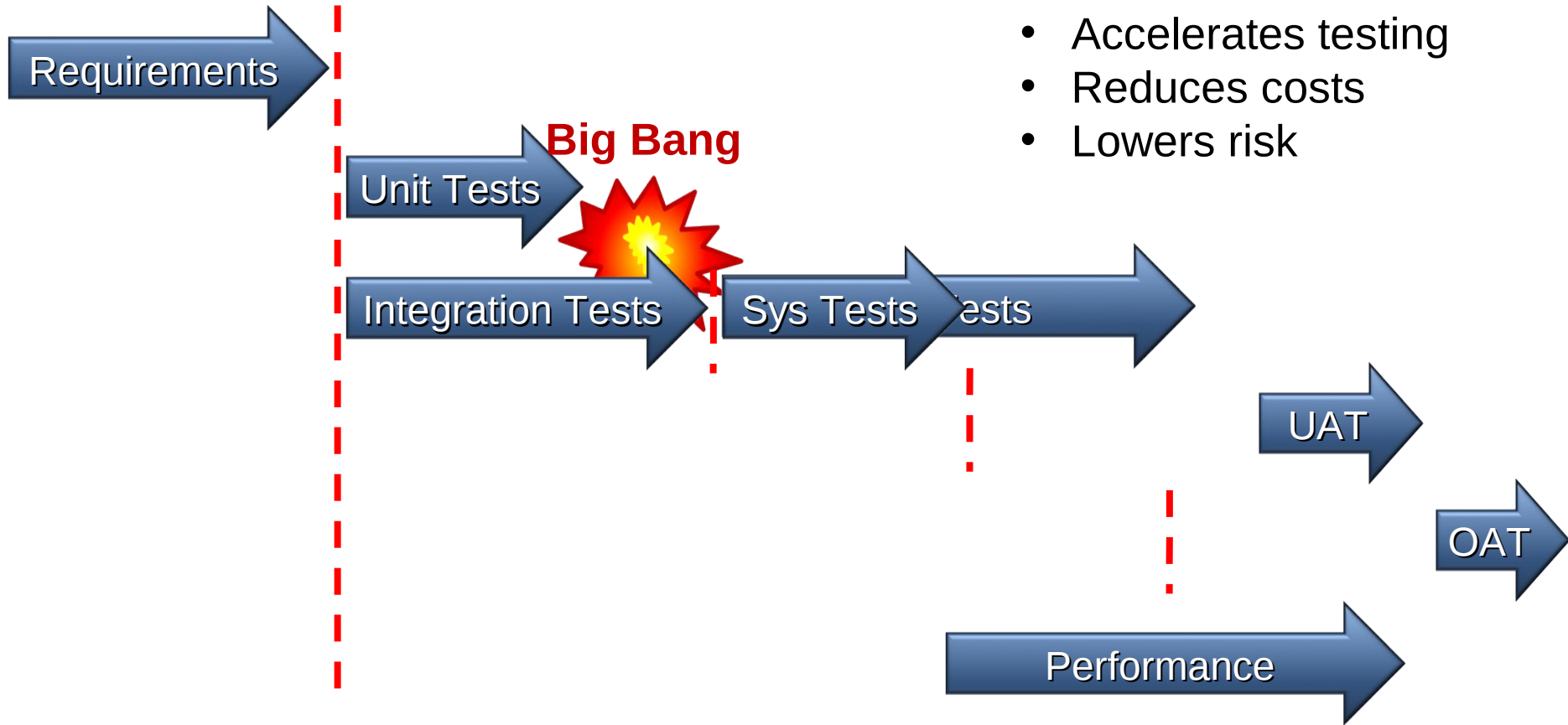


# The Old World

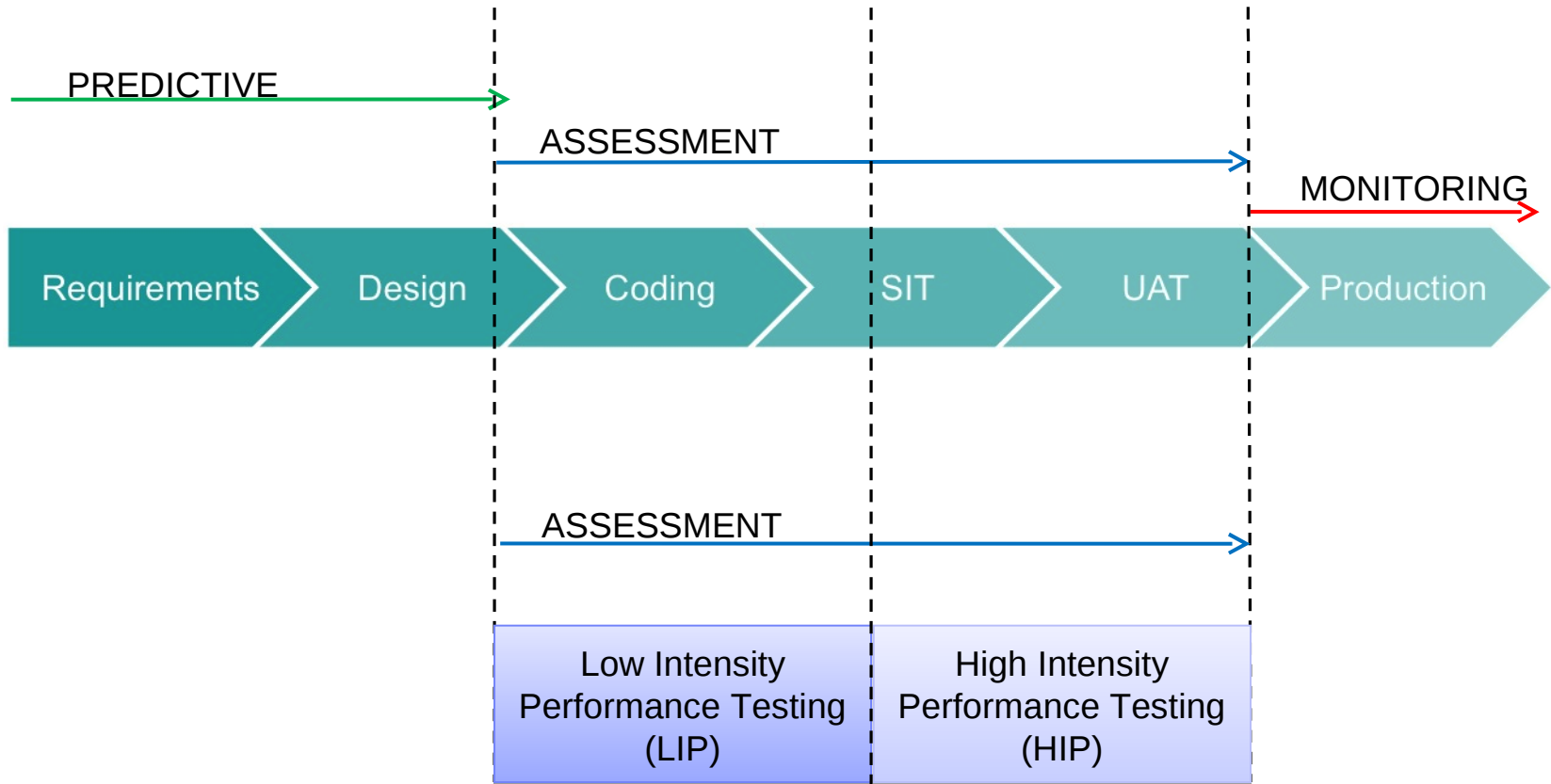


- Multiple defects introduced at once
- More unknowns – greater risk
- More expensive

# The New World

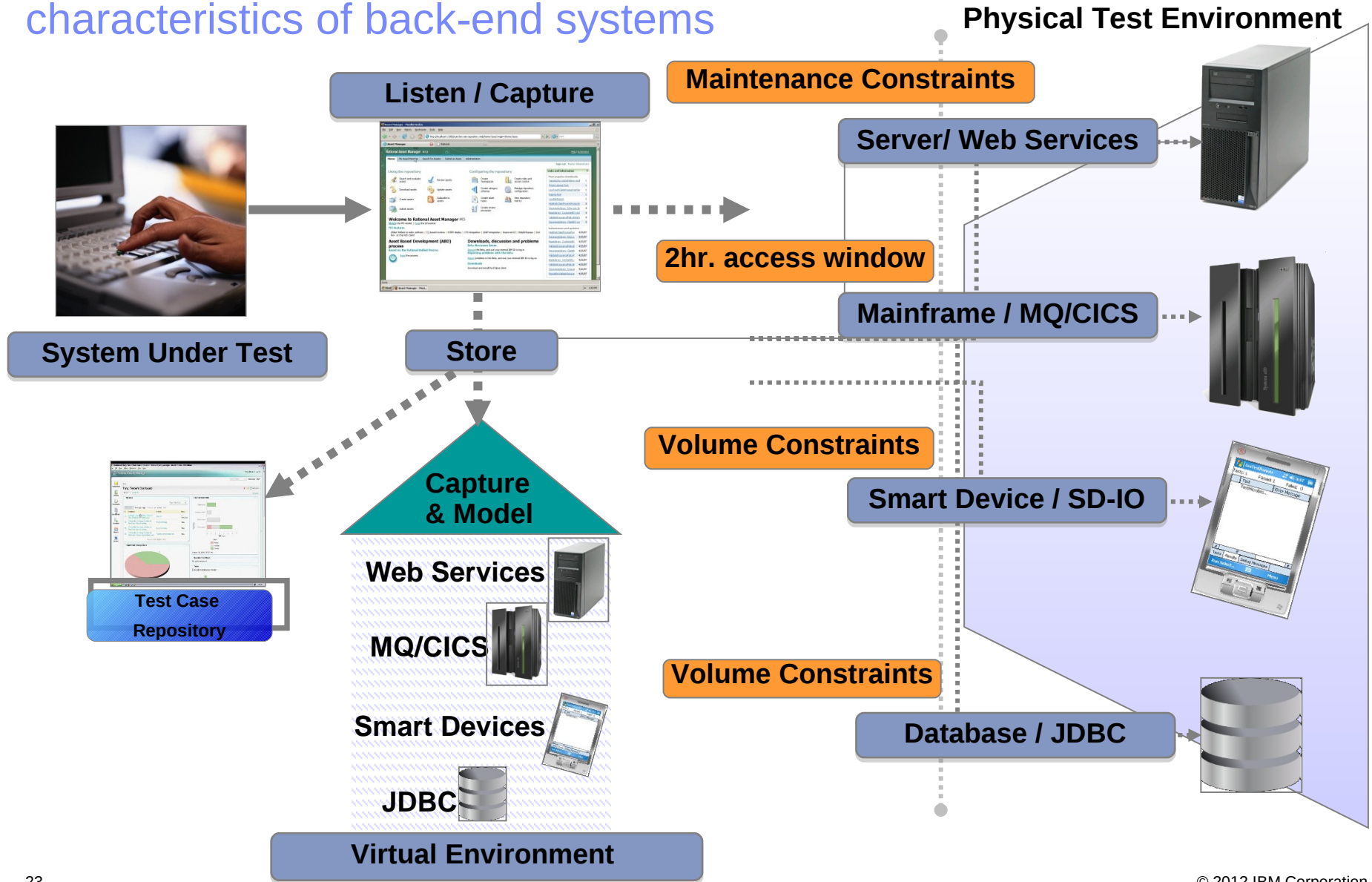


# Changing the Performance Test Model



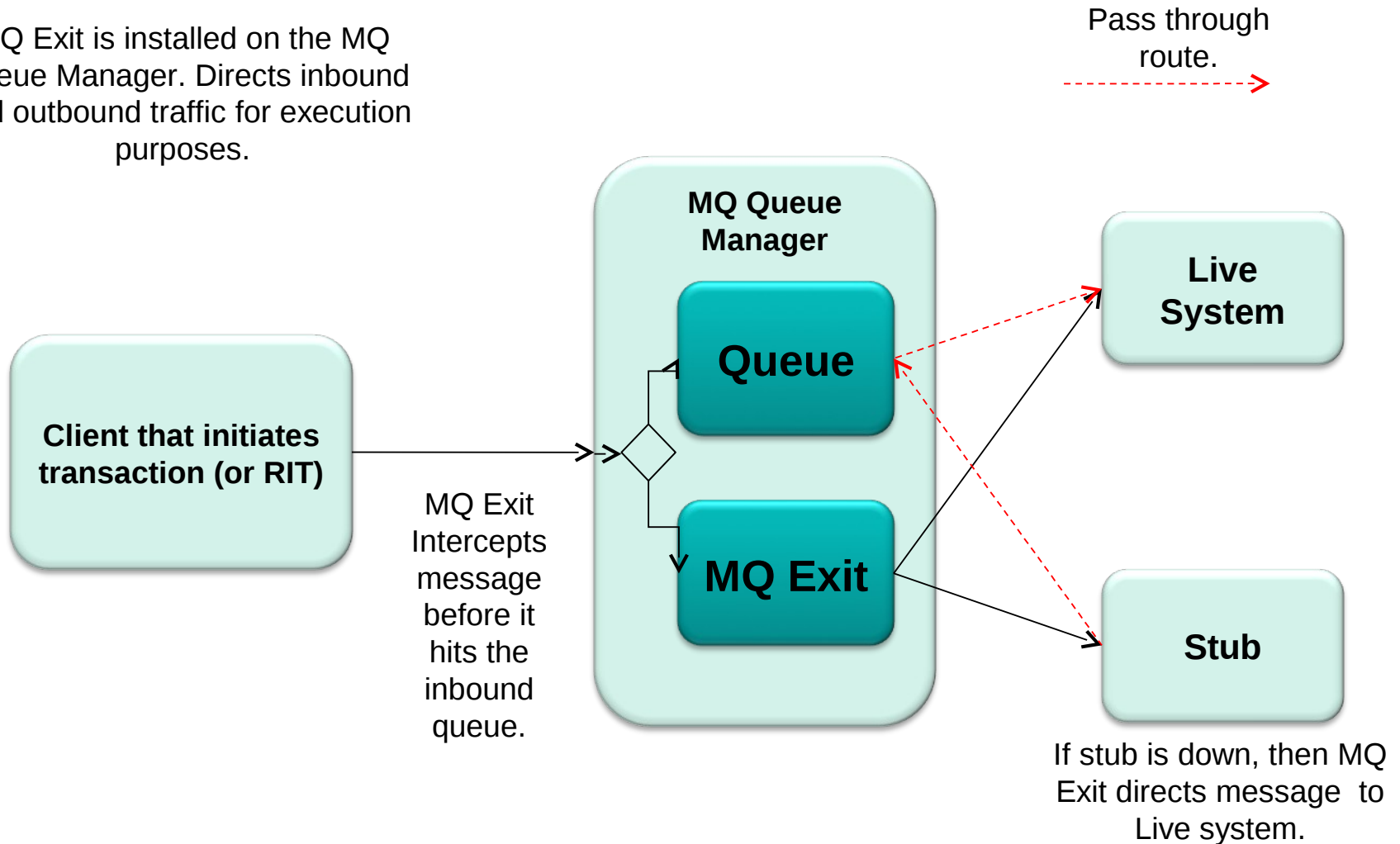
Incremental integration strategy defines HIP start point;  
focus of HIP is informed by LIP results

# Service Virtualization listens to and captures the behavior and characteristics of back-end systems



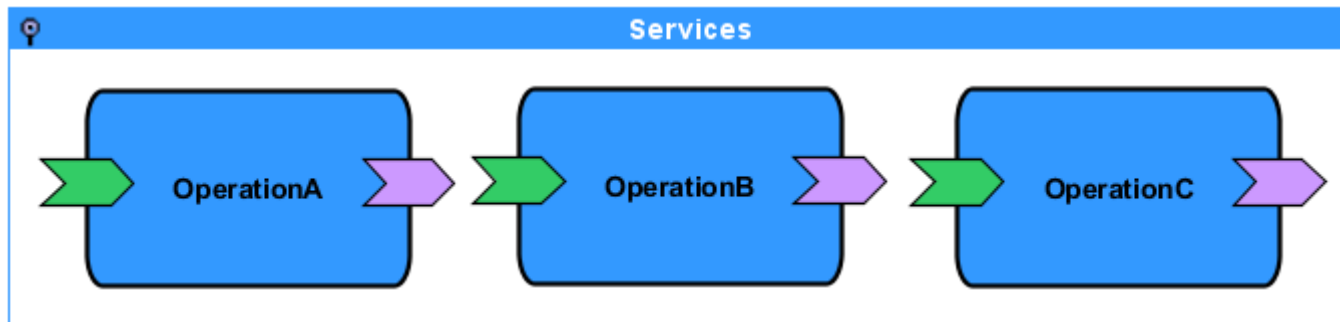
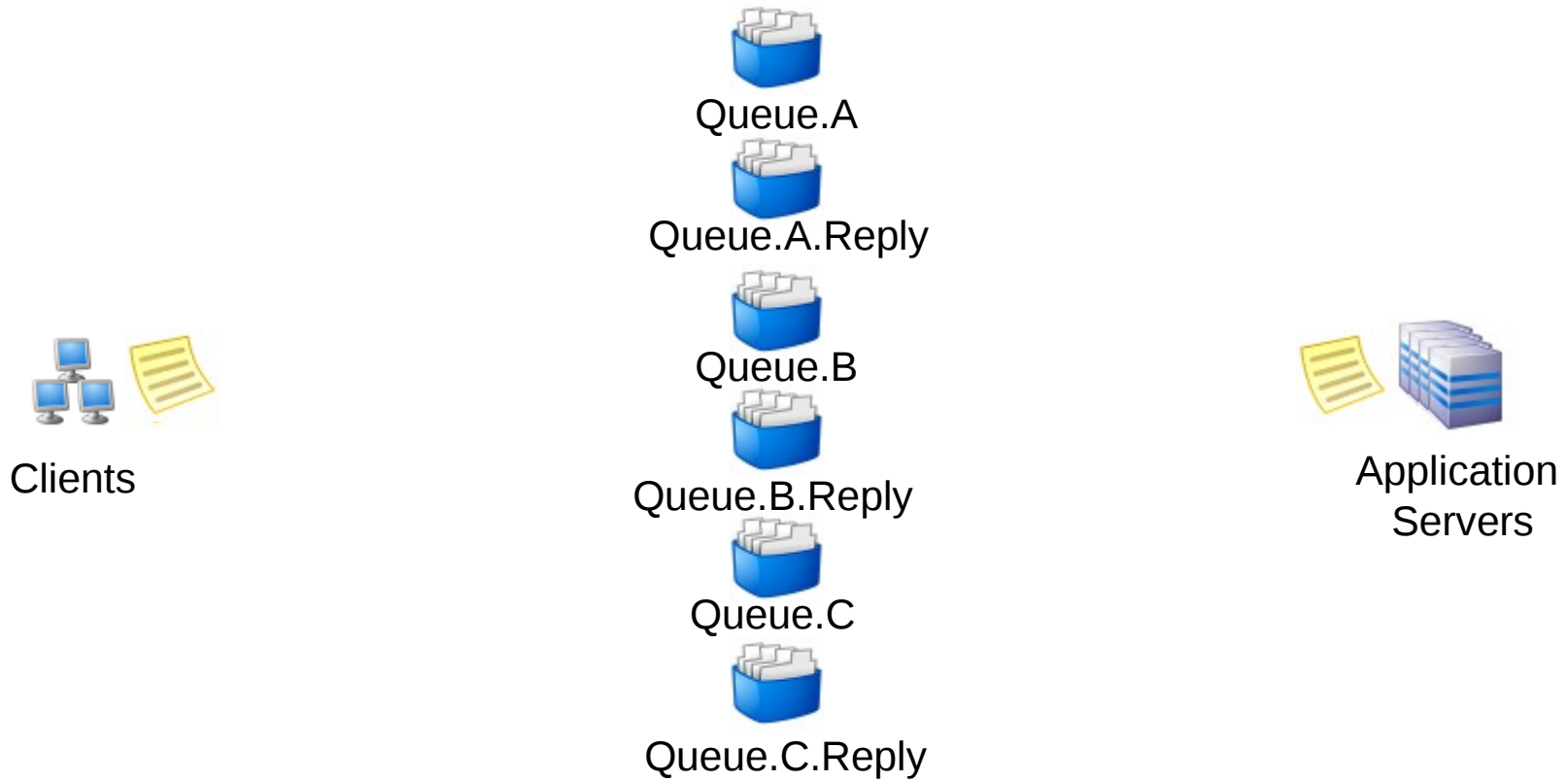
# How would this work?

MQ Exit is installed on the MQ Queue Manager. Directs inbound and outbound traffic for execution purposes.





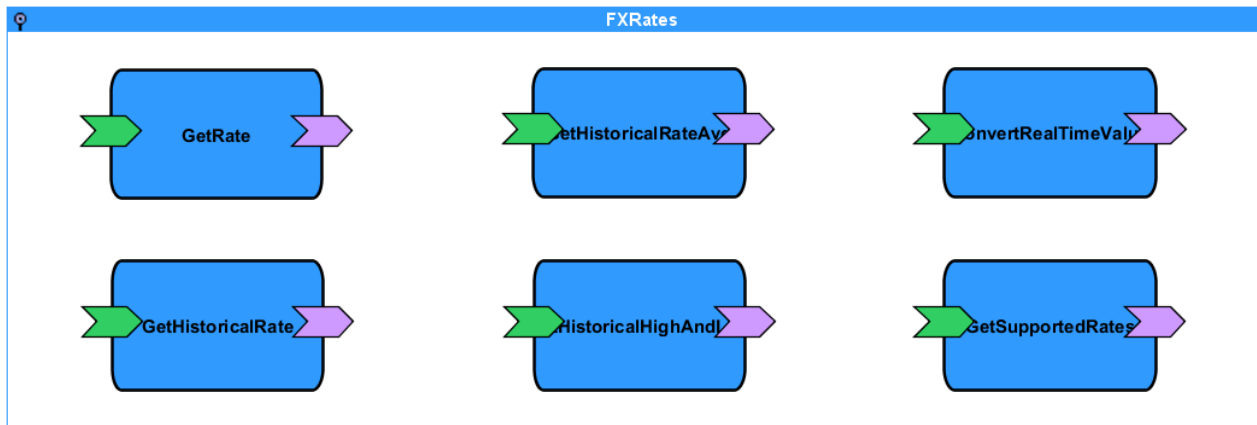
# Building a System Model from Recorded Events



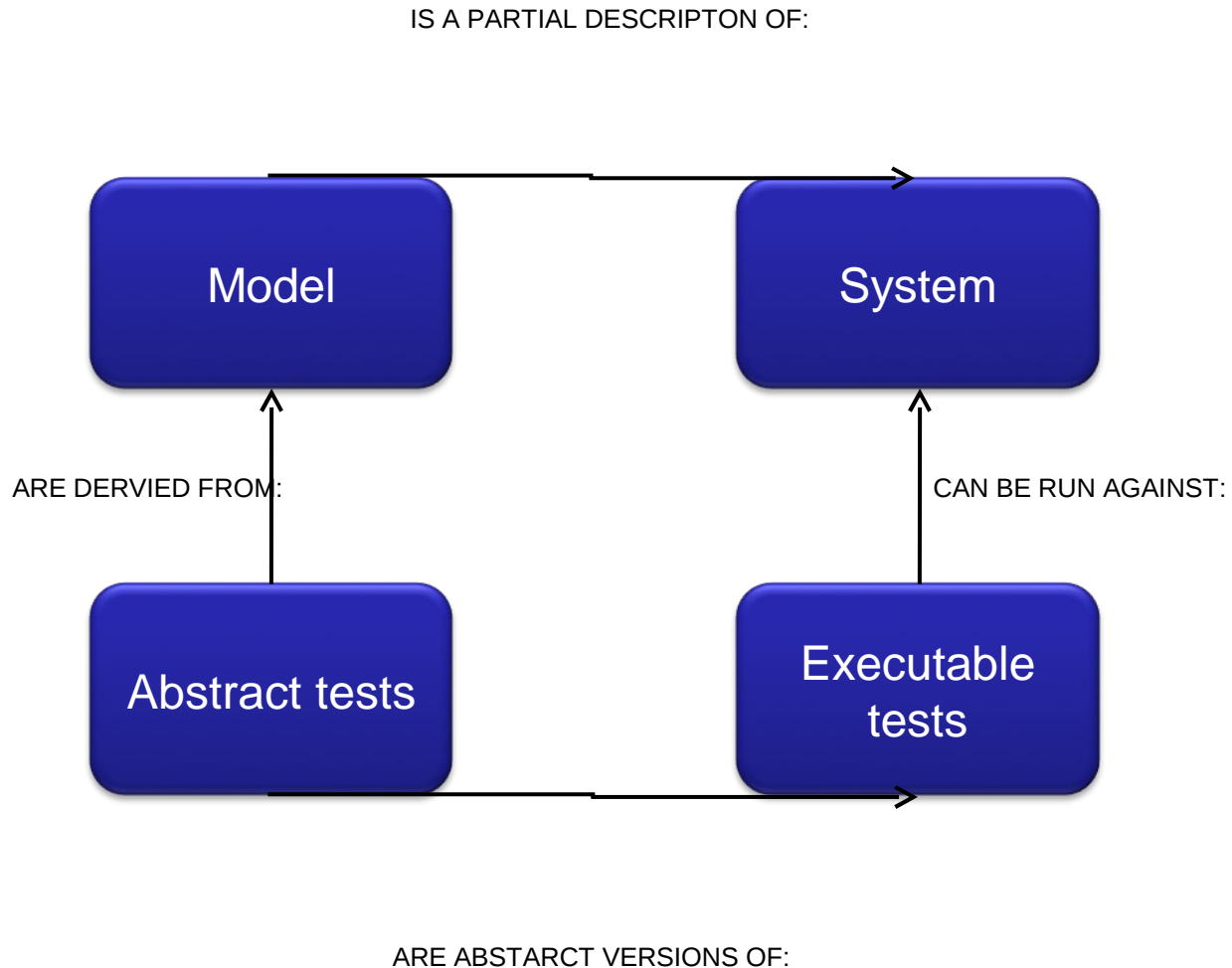
# Synchronization

Synchronization provides an interface for importing project assets, and mirroring updates to those assets.

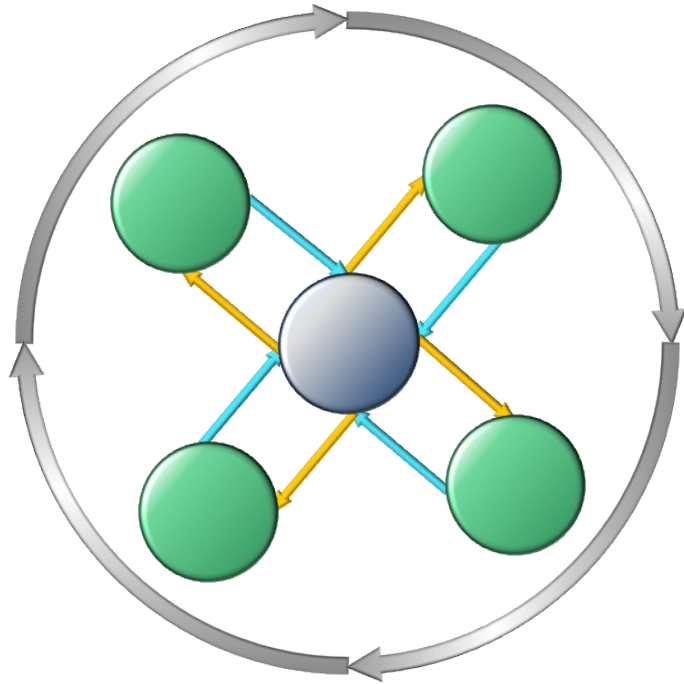
```
>>> <portType name="FXRates">  
>>>   <operation name="GetRate"> ... </operation>  
>>>   <operation name="GetHistoricalRate"> ... </operation>  
>>>   <operation name="GetHistoricalRateAvg"> ... </operation>  
>>>   <operation name="GetHistoricalHighAndLow"> ... </operation>  
>>>   <operation name="ConvertRealTimeValue"> ... </operation>  
>>>   <operation name="GetSupportedRates"> ... </operation>  
>>> </portType>
```



# Synchronization and Model Based Testing



# Incremental Testing

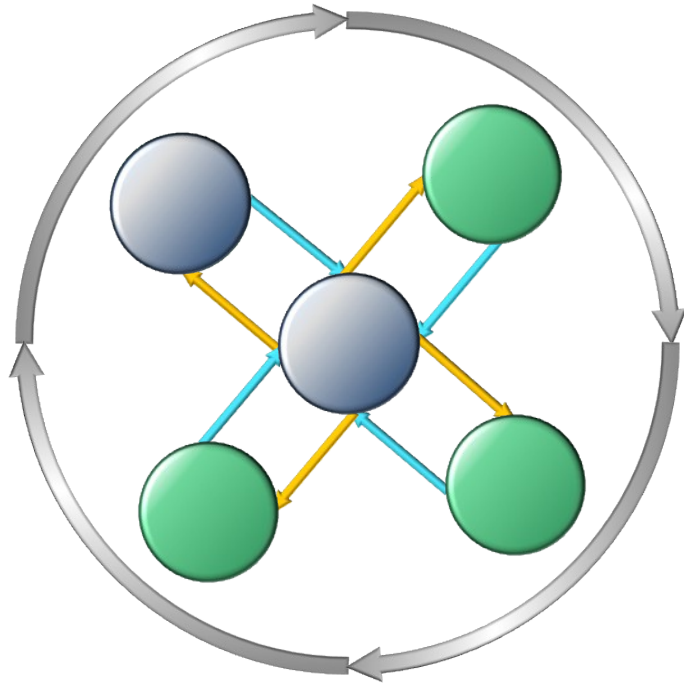


Initial Stage:

A single component can be tested in the context of an end to end environment, supplied by a set of virtualized components.

- Actual Component
- Virtualized Component

## Incremental Testing

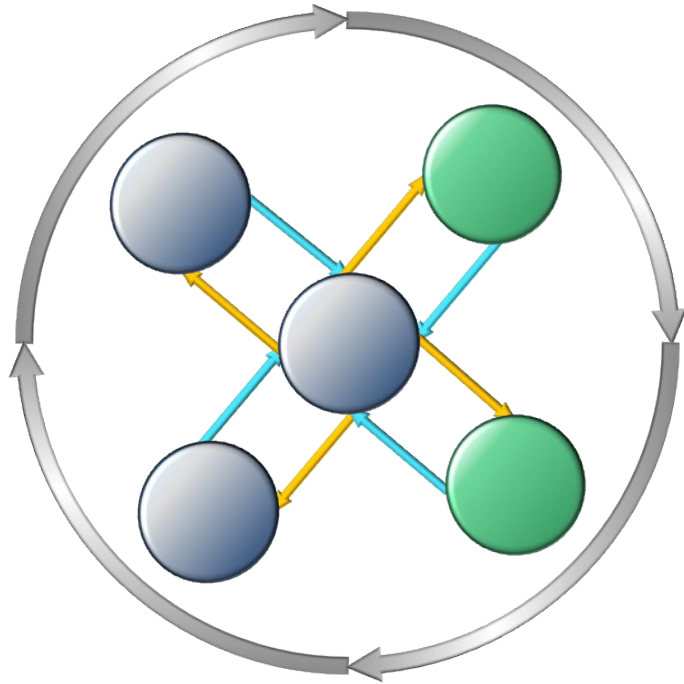


- Actual Component
- Virtualized Component

As components are built:

The same end to end tests can be run, replacing virtualized components with actual components.

## Incremental Testing

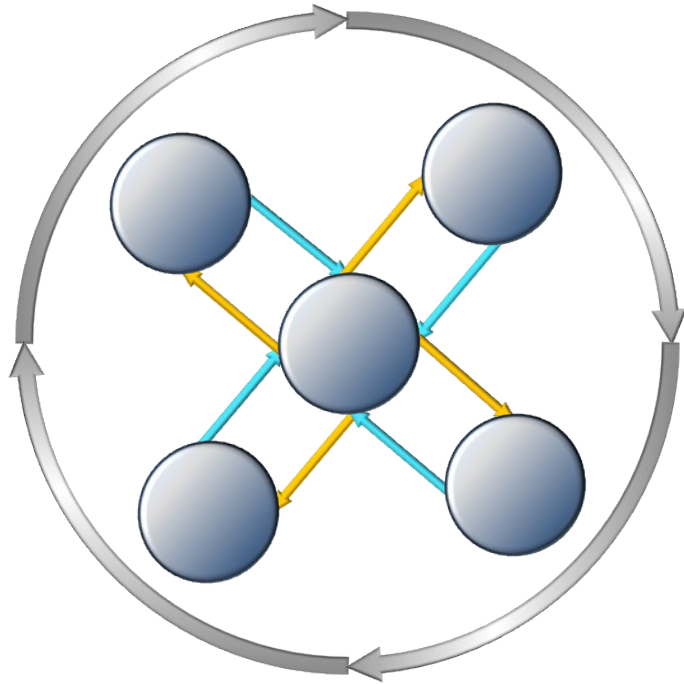


● Actual Component  
● Virtualized Component

As components are built:

This enables us to test downstream dependencies as they are built.

## Incremental Testing



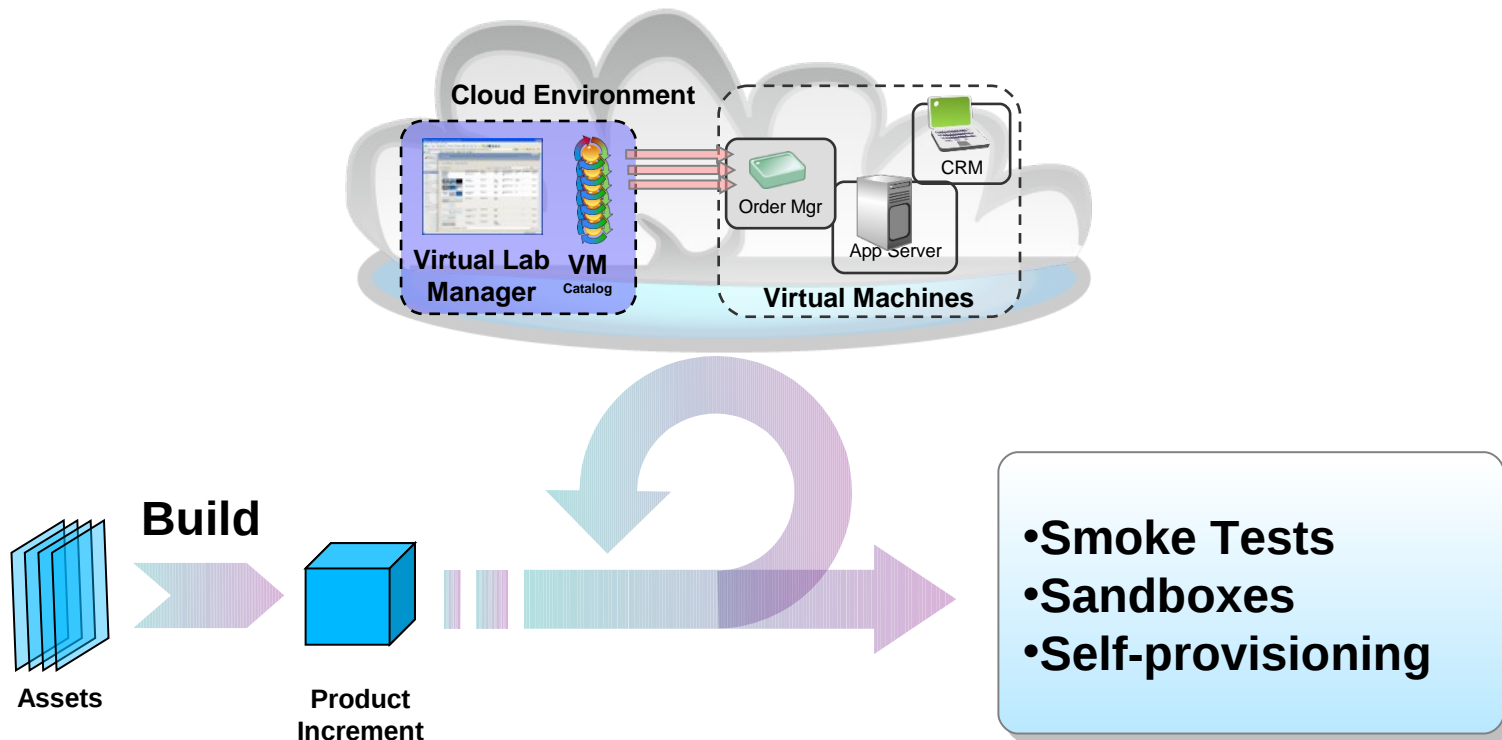
- Actual Component
- Virtualized Component

When the system is complete:

End to end testing can be carried out with fewer surprises and lower risk

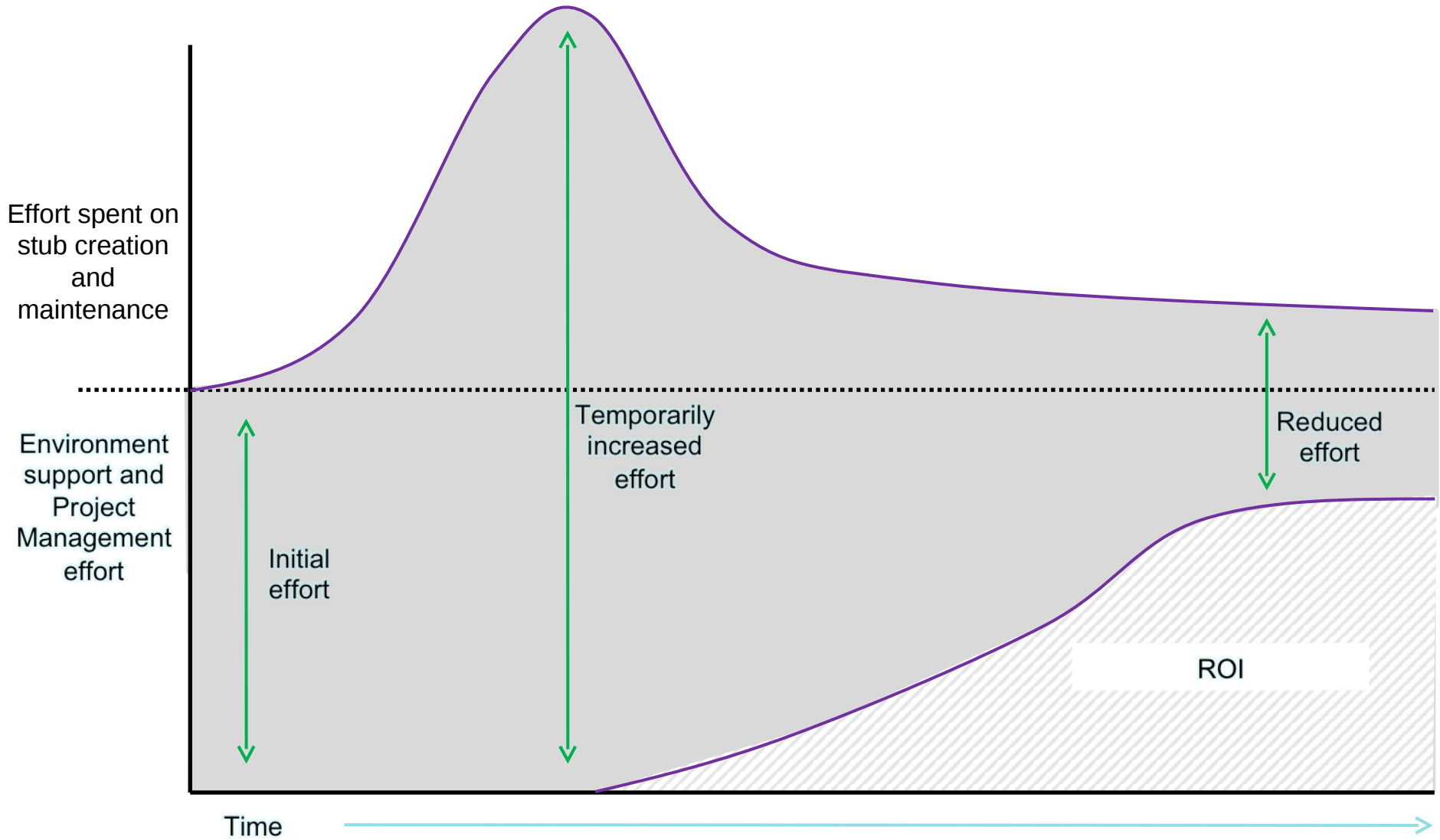
# Streamline your process with continuous integration & test

- Avoid time spent installing and configuring software just to discover basic build issues
- Smoke tests can be integrated as part of the continuous integration process through deployment automation and test virtualization
- Regression test results are made available to the entire team to shorten resolution





# ROI Model



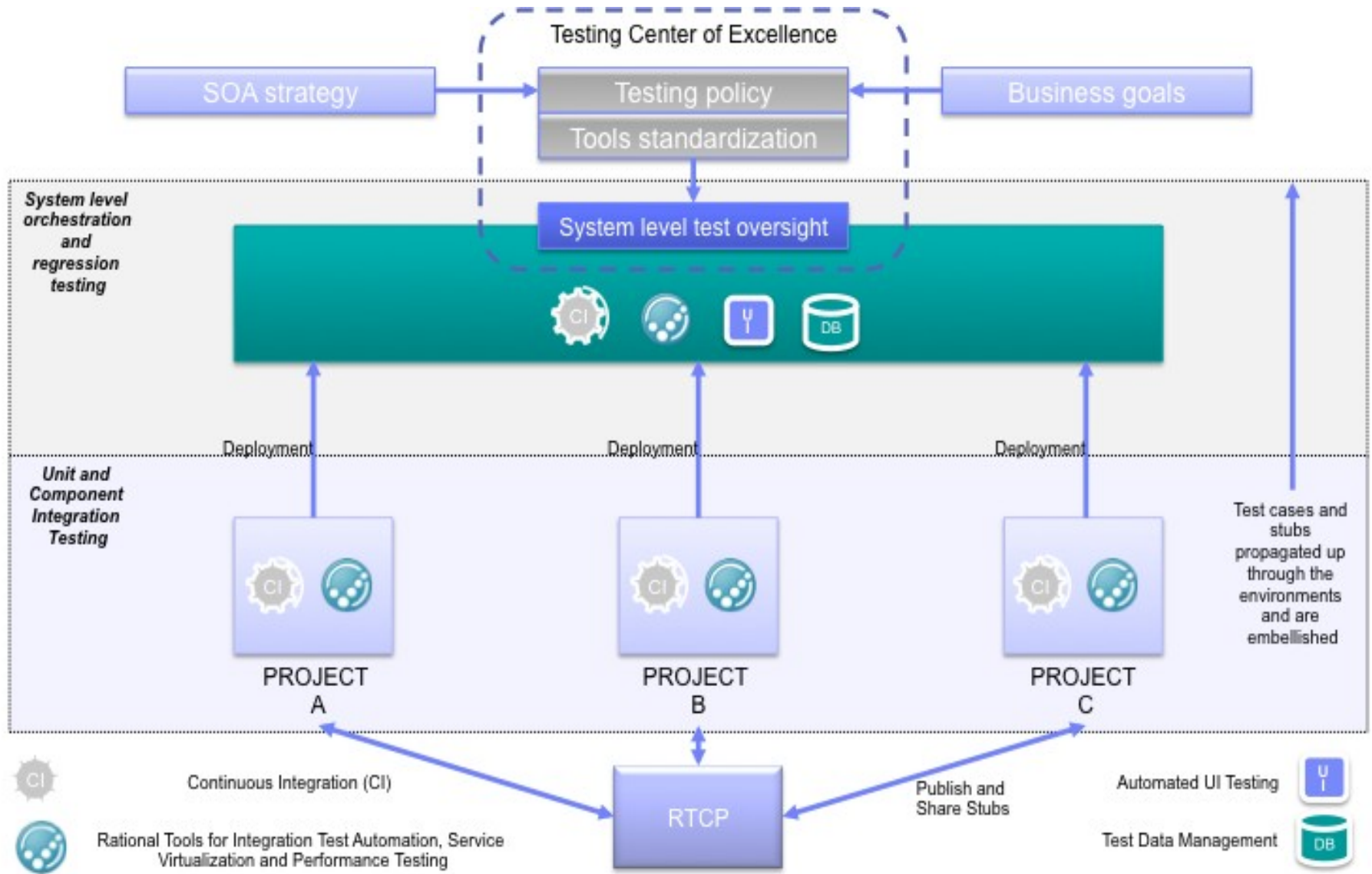
## Agenda

- Looking back at the past
- Planning for flexibility
- Avoiding the big bang
- **Wrapping up** – *realize the possible!*

## Service Virtualization allows you to...

- **Eliminate** critical development and testing **constraints** by virtualizing IT resources
- **Deliver** realistic simulated development and **test environments at a fraction of the cost**
- **Isolate defects** and drill into their root cause
- Create automated regression test suites to **reduce test cycle time** and **improve test coverage and quality**
- Leverage Cloud services with Service Virtualization to **minimize infrastructure set-up time** and provide **scalable, on-demand infrastructure**
- **Drive benefits across the entire application development and test lifecycle**

# The Vision



# IBM quality management and testing solutions have demonstrated measureable results and business value

## **Major telecom carrier**



Multiple test releases required additional test resources, increasing testing costs

- Solution: Virtualized services and created reusable test cases across multiple environments
- Results: Reduced testing time by 50%, saving \$800K over 3 years

## **A leading global financial services firm with assets of over \$2 trillion**



Many disparate legacy format to be migrated to next gen payments system

- Solution: Virtualized third-party systems otherwise unavailable for testing
- Results: Reduced manual testing from 10 days to 10 minutes, saving over \$7 million to date

## **Major U.S. insurer**



Recognized manual efforts insufficient for complex SOA and web services

- Solution: Agile middleware solution created to match the legacy systems' functionality
- Results: Reduced testing by 95% to 2 hours; reduced 'rate filing' validation by 94% to 320 hours

## **Global manufacturer acquires competitor**



Migrate off rented infrastructure onto company's standardized middleware platform

- Solution: Virtualized critical, unavailable systems during migration
- Results: Fully integrated in 6 months – 2 months ahead of schedule; saved significant rental costs and dependencies on third-party systems

# QUESTIONS

[www.ibm.com/software/rational/offerings/quality](http://www.ibm.com/software/rational/offerings/quality)



[www.ibm.com/software/rational/offerings/quality](http://www.ibm.com/software/rational/offerings/quality)

© Copyright IBM Corporation 2011. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.