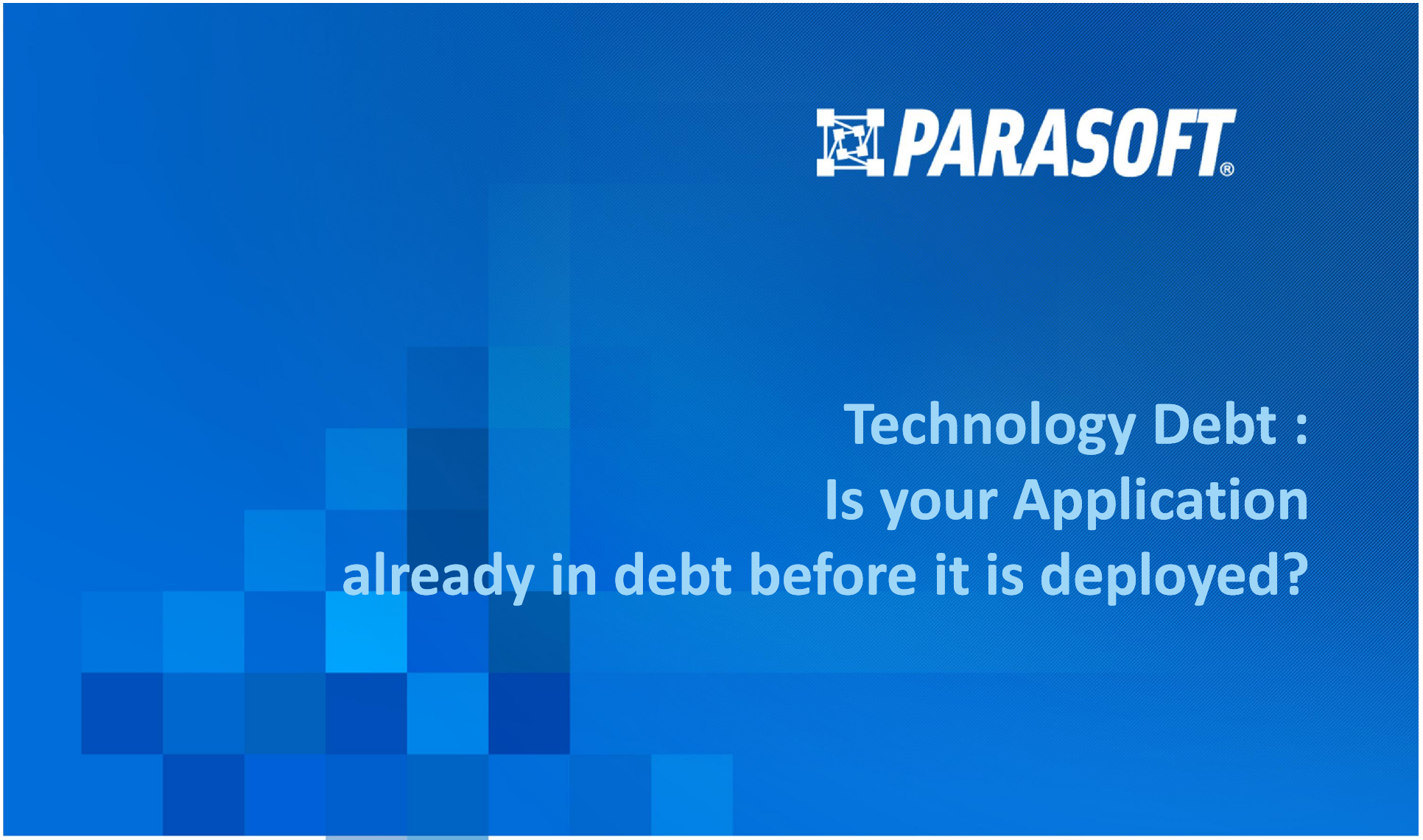


Stanley Eu
Regional Director
Parasoft Singapore (& ASEAN)

And they drew hope again
from their schoolfellow's teaching lesson

TORSTEN ZELGER

http://www.simply-the-test.blogspot.sg

Tech Debt

PARASOFT

Forbes

The term was coined by Ward Cunningham, a programmer who is also known for developing the first wiki.

- Engage in poor programming practices, followed by patches
- Fail to maintain technical assets
- Isolate their technology or product from compatibility with the rest of the world
- Fail to keep up with universal product and technology trends.

David Williams, Jan 25, 2013

PARASOFT

Why banks are likely to face more software glitches in 2013 : By Leo Kelion

BBC

Business software is becoming increasingly **complex**, composed of sub-systems written in different programming languages, on different machines by disparate teams….

The idea is that IT bosses have allowed a certain amount of "unfixed" code to accumulate in order to roll out new facilities on schedule. But as the debt has grown, so has the **risk** of systems becoming "gummed up".

(coined as Technology Debt)

Examples

The Tremors From a Coding Error (NYTimes, 2010)

Investment firm AXA Rosenberg shelled out $217 million last year to cover investor losses from what it called a "significant error" in the computer code for one of its investment models.

Is Knight's $440 million glitch the costliest computer bug ever? (CNN Money, 2012)

When it comes to lethal bugs, the computer glitch that set fire to $440 million of Knight Capital Group's funds last Wednesday ranks right up there with the tsetse fly.

Examples

PARASOFT

Software to blame for Prius brake problems
February 5, 2010

Software glitches leave Navy Smart Ship USS Yorktown dead in the water in 1998

USS YORKTOWN
suffered a major computer crash in September last year.

Ariane5 : One Software Bug caused a $7B project to explode in 1996

Question

What are your current Policies, Processes, People, Infrastructure and Technology that is in your Organization– especially in Software Development or Outsourcing?

How are they Enforced

What do you do when you see an AMBER light at the traffic junction?

1. Slow Down
2. Go Faster
3. Jam Brakes

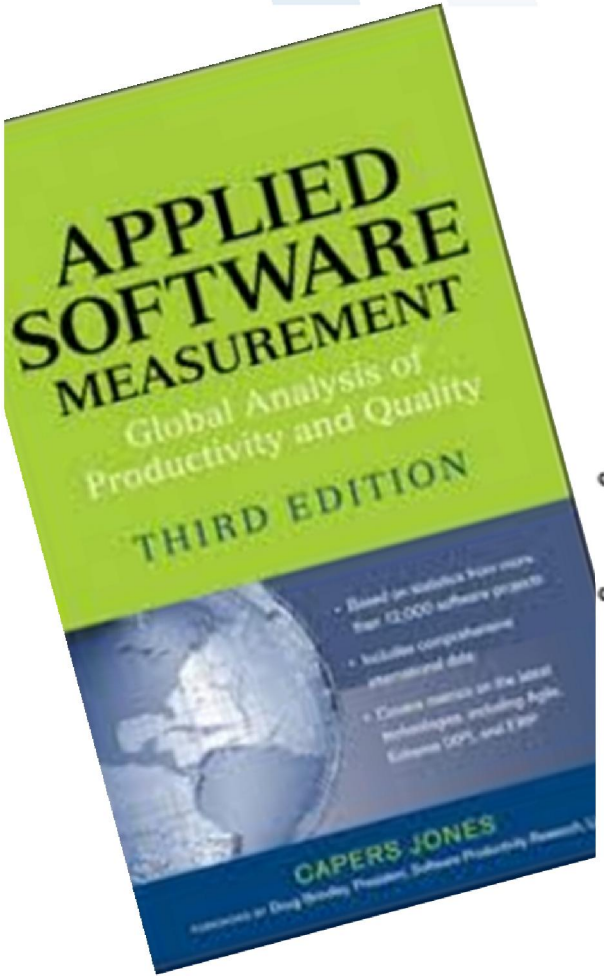
Mitigating Risk

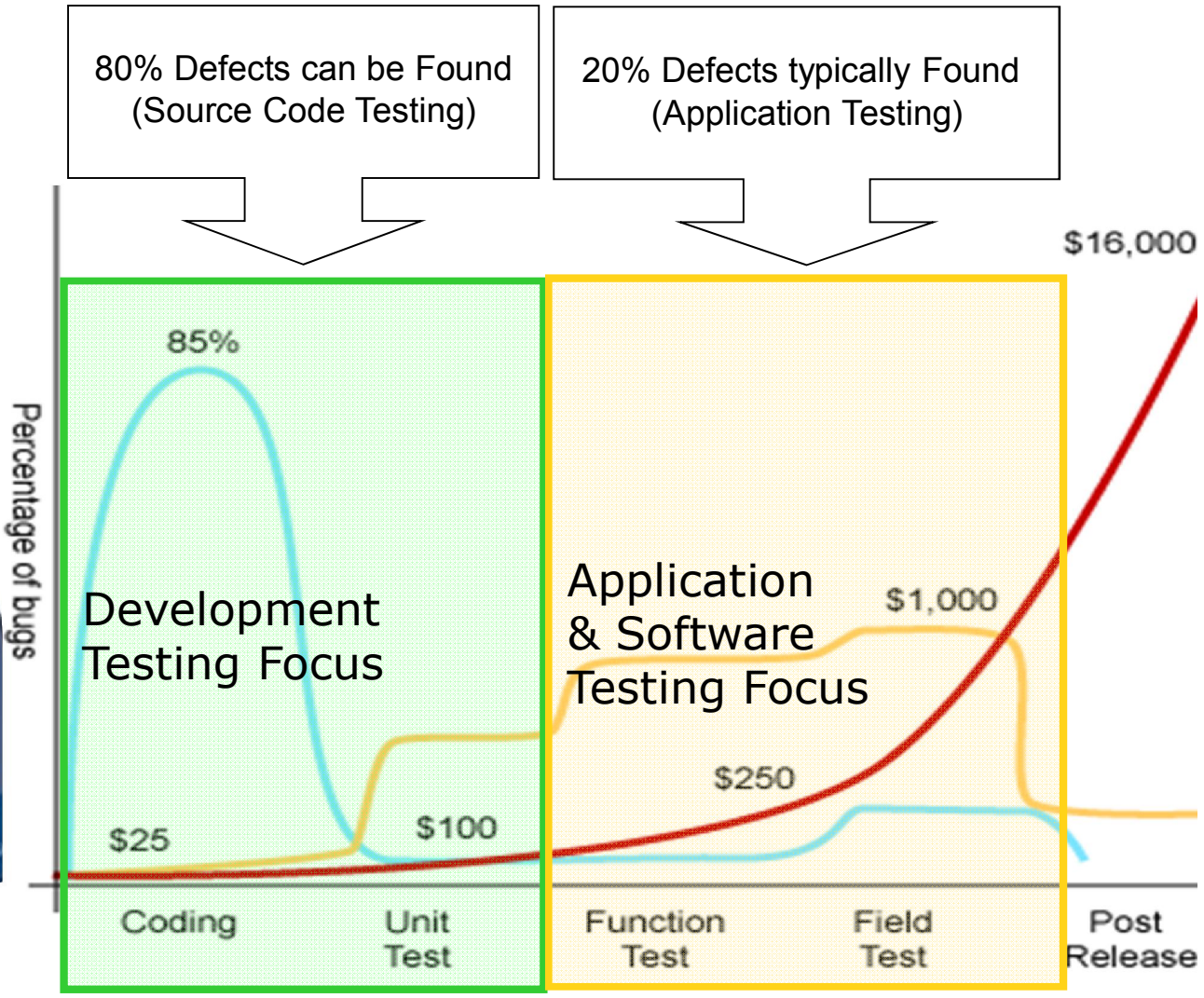
PARASOFT

Governance	Monitoring	Compliance	Enforcement
Coding Guidelines, Test Requirements, ALM	*Requirements Management, Code Analysis & Review, Functional and Load Testing*	*Industry, Company, Project*	*Carrot or Stick?*

Mitigating Risk

PARASOFT





Source: Applied Software Measurement, Capers Jones, 1996

Prevent Tech Debt

PARASOFT

White Box Testing

Find the defects in source code
Agile Development, Test Driven Development, Xtreme Programming

Black Box Testing

Find the defects in application
No source code needed

Development Testing (White Box)

Static Analysis

1. Pattern-based
2. Flow-based
3. Metrics-based

Dynamic Analysis

1. Debug/Trace
2. Unit Tests
3. Mock/Stub
4. Runtime Error Detection (Memory)
5. Binary Testing (No Source Code)

Coverage Analysis

1. Test Cov
2. Runtime Cov
3. Manual Cov

Application Testing (Black Box)

Functional Testing

Load Testing

Security Penetration Testing

Service Virtualization

Static Analysis - Security

PARASOFT

PCI DSS

OWASP

CWE/SANS

NIST SAMATE

Customize/Company Security Rules

PCI Security Standards Council

NIST National Institute of Standards and Technology

OWASP The Open Web Application Security Project

CWE Common Weakness Enumeration
A Community-Developed Dictionary of Software Weakness Types

Development Testing (White Box)

Static Analysis

1. Pattern-based
2. Flow-based
3. Metrics-based

Dynamic Analysis

1. Debug/Trace
2. Unit Tests
3. Mock/Stub
4. Runtime Error Detection (Memory)
5. Binary Testing (No Source Code)

Coverage Analysis

1. Test Cov
2. Runtime Cov
3. Manual Cov

Application Testing (Black Box)

Functional Testing

Load Testing

Security Penetration Testing

Service Virtualization

Coverage Analysis

PARASOFT

DISA
DEFENSE INFORMATION SYSTEMS AGENCY
DEPARTMENT OF DEFENSE

Application Security and Development STIG, V3R2
28 October 2011

Code Coverage

An important aspect of all testing methods, including fuzz testing, is code coverage achieved through the testing process. Code coverage is the percentage of the application code exercised during the testing process. Security flaws often occur in areas of the code not regularly executed, so it is important to keep track of how often a code branch is executed and tested to ensure thorough testing is performed. If code coverage is low, then the tests must be evaluated to determine why code coverage is low, and the tests changed to increase the percentage of code covered by the test cases.

(APP5070: CAT III) The Test Manager will ensure code coverage statistics are maintained for each release of the application.

Coverage Analytics

PARASOFT

- Unit Tests – unit testing framework
  - Java – Junits
  - C# – Nunit
  - C++ - CPPUnit
  - Python – PyUnit
  - PHP – Phpunit
- Runtime Coverage
  - Code coverage based on execution of application
- Manual Coverage
  - Code Walkthrough

Coverage Analytics

PARASOFT

- Types of Coverage
  - Line
  - Path
  - Block
  - Decision
  - Branch
  - Condition
  - Modified Condition / Decision Coverage (MC/DC)

Coverage criterion	Minimum number of different inputs required for 100% coverage	Example of a minimum set of test inputs
Statement coverage	2	0, 1
Branch coverage	2	1, 2
Path coverage	4	0, 1, 2, 3
Full regression coverage	256	-128, ..., 127

- One of the many methods of Quality Analysis
  - Don’t be too fixed on it
  - Depends on the industry

Development Testing (White Box)

Static Analysis

1. Pattern-based
2. Flow-based
3. Metrics-based

Dynamic Analysis

1. Debug/Trace
2. Unit Tests
3. Mock/Stub
4. Runtime Error Detection (Memory)
5. Binary Testing (No Source Code)

Coverage Analysis

1. Test Cov
2. Runtime Cov
3. Manual Cov

Application Testing (Black Box)

Functional Testing

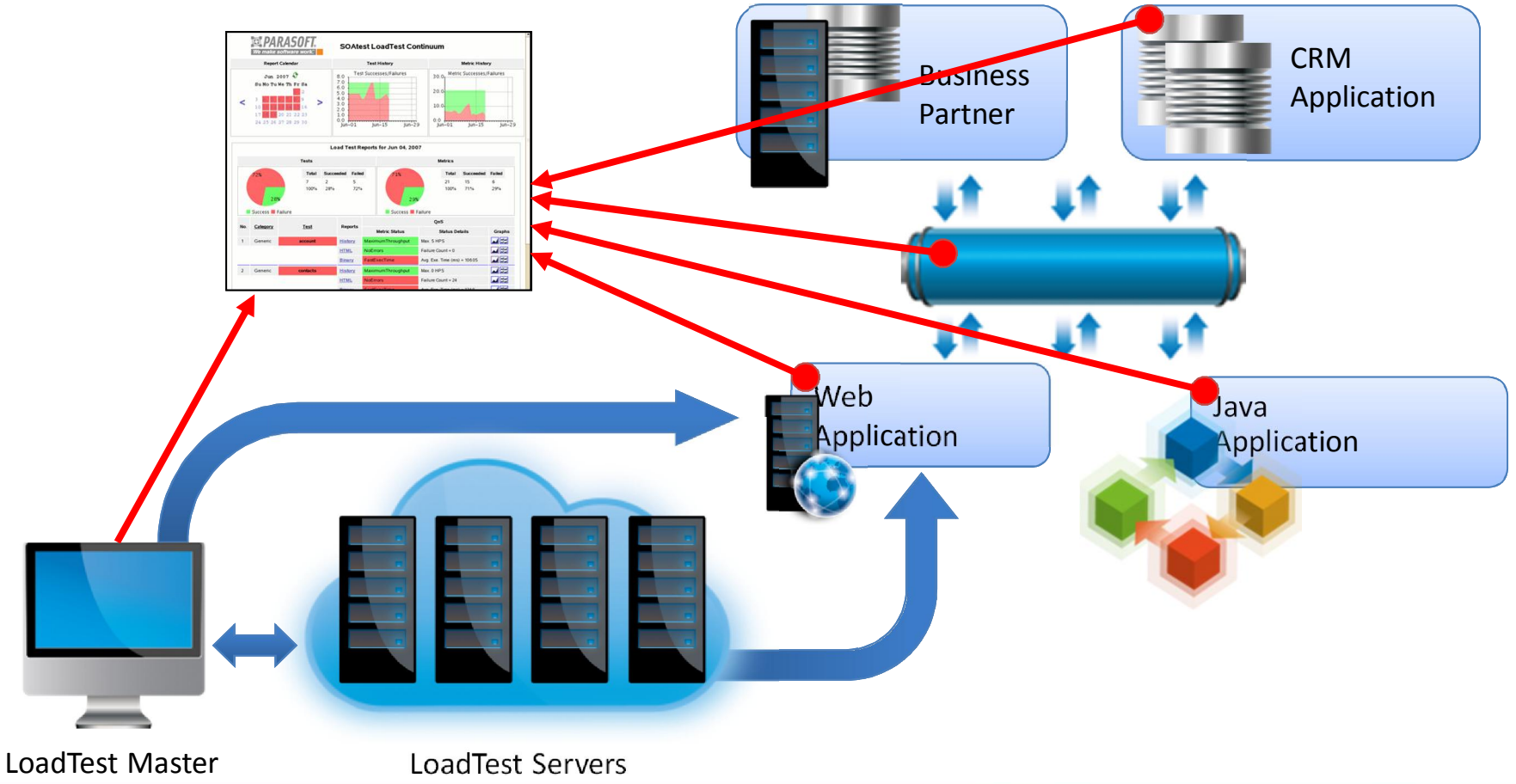
Load Testing

Security Penetration Testing

Service Virtualization

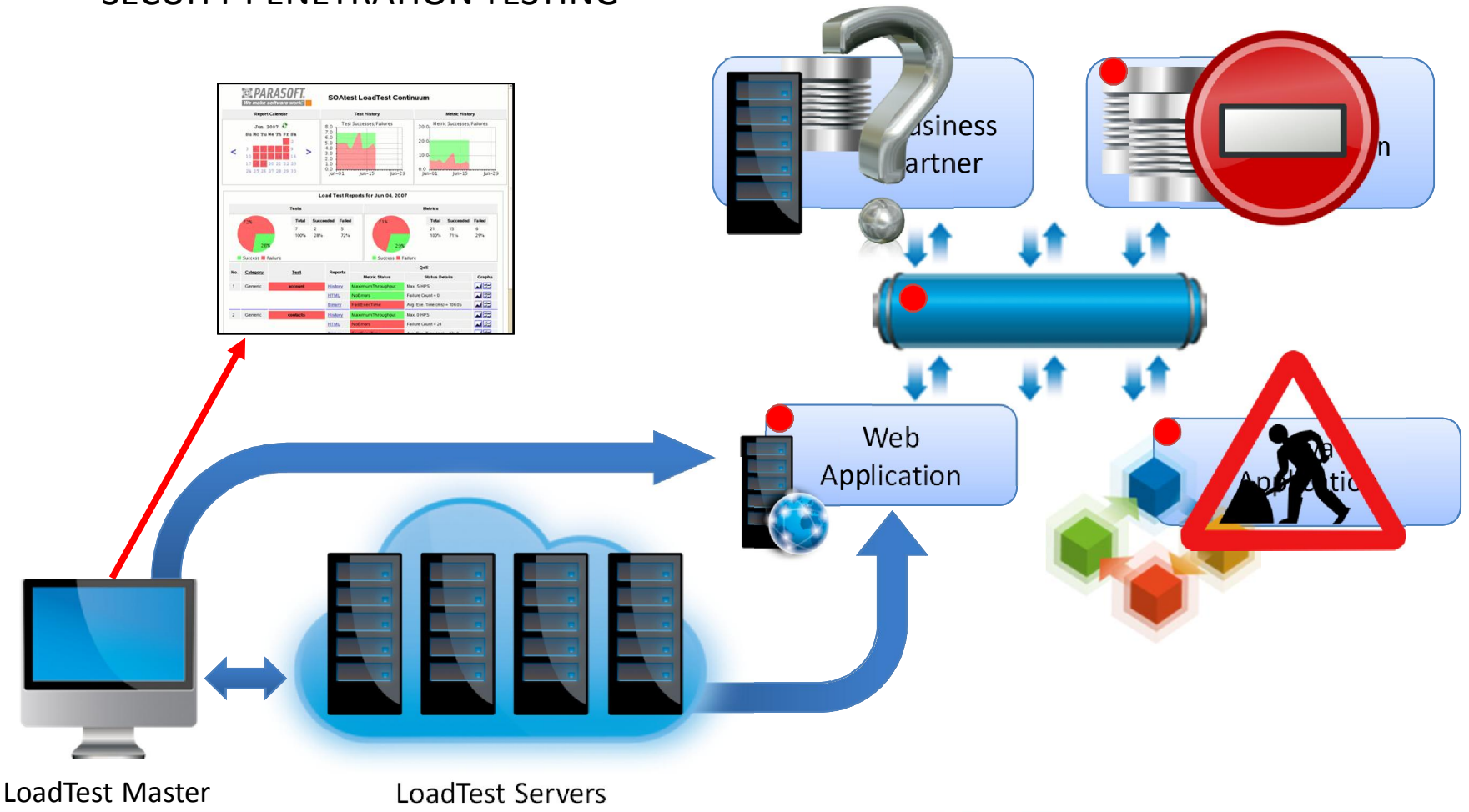
Black Box Testing on Application

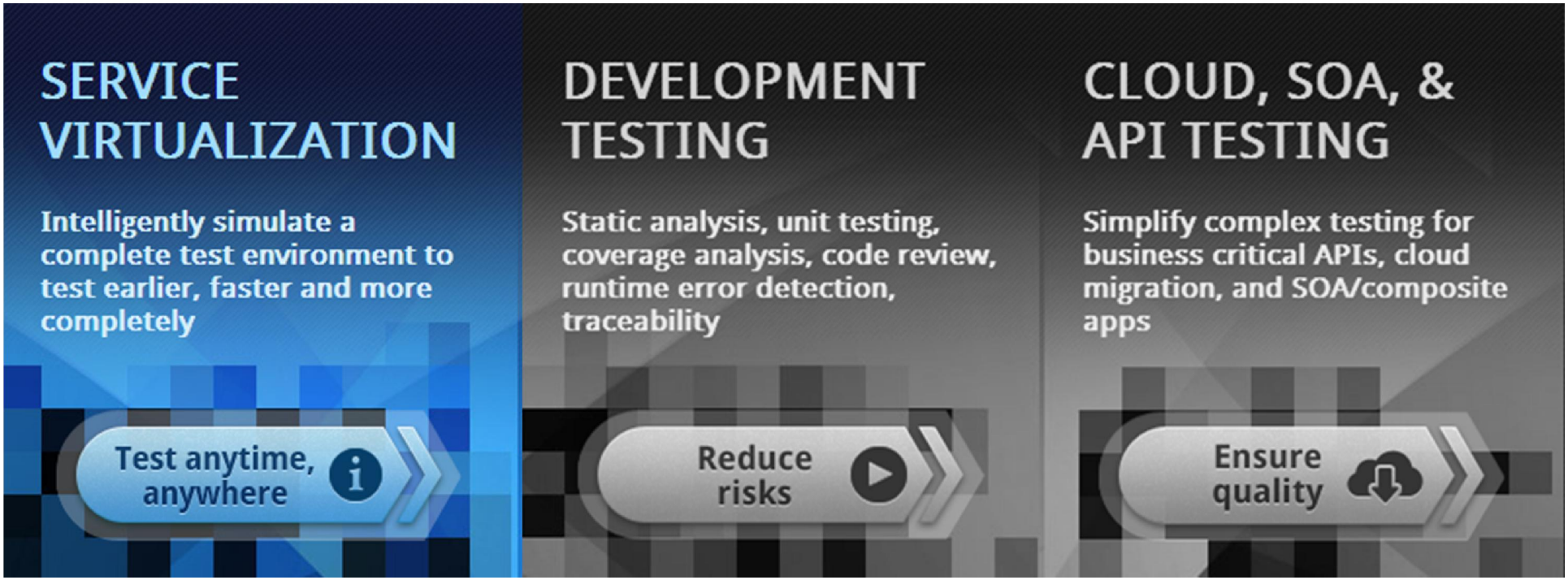
- FUNCTIONAL TESTING
- LOAD TESTING
- SECUITY PENETRATION TESTING



Black Box Testing on Application

- FUNCTIONAL TESTING
- LOAD TESTING
- SECUITY PENETRATION TESTING



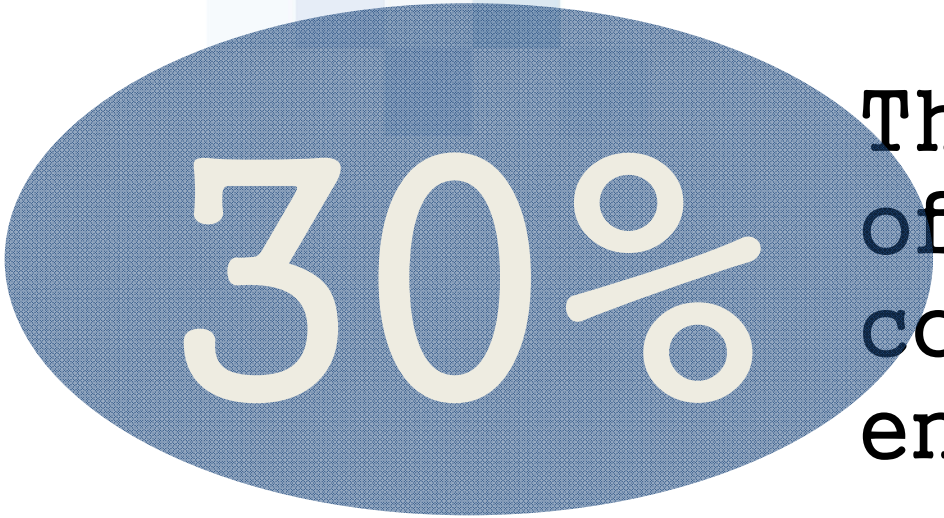


New Paradigm, Revolutionary System

Service Virtualization delivers a simulated development / test Environment allowing an organization to test anytime or anywhere

Some Feedback…

PARASOFT

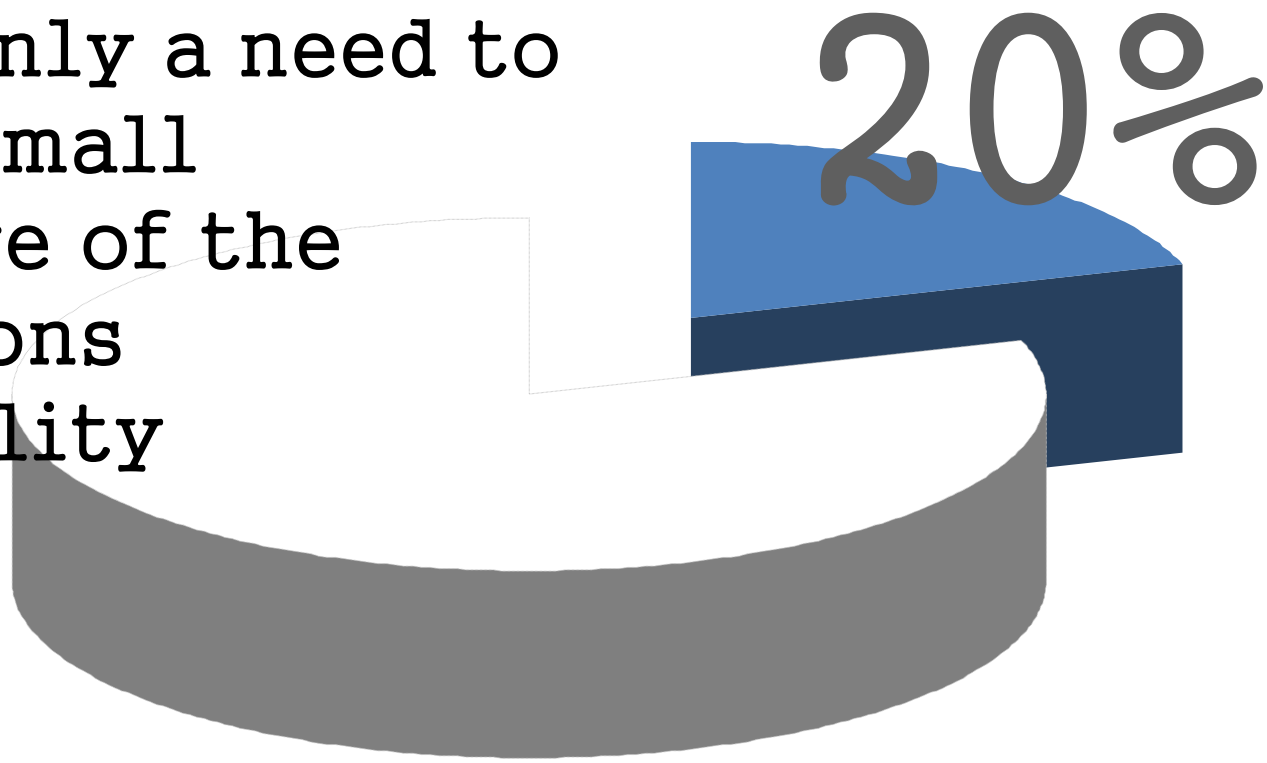


The average percentage of time spent configuring the test environment.

The average percentage of the test plan able to complete once configured.

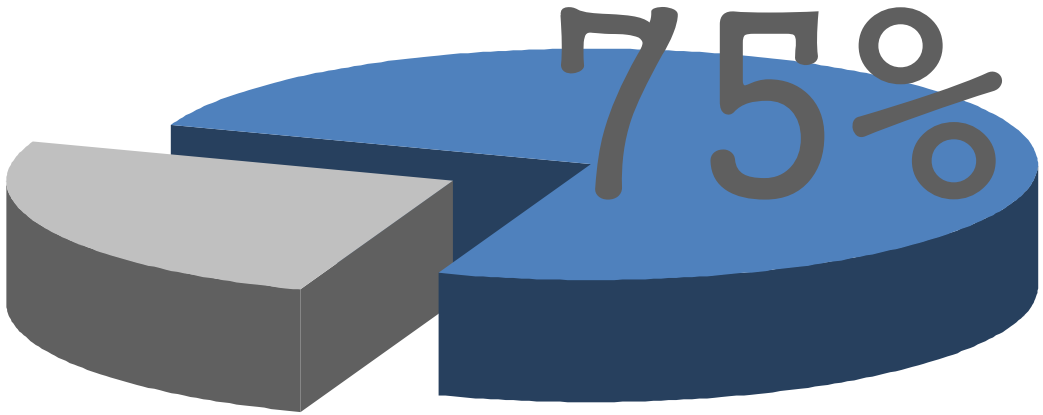
Some Feedback…

When testing with a **dependent** application, there is only a need to access a small percentage of the applications functionality



Some Feedback…

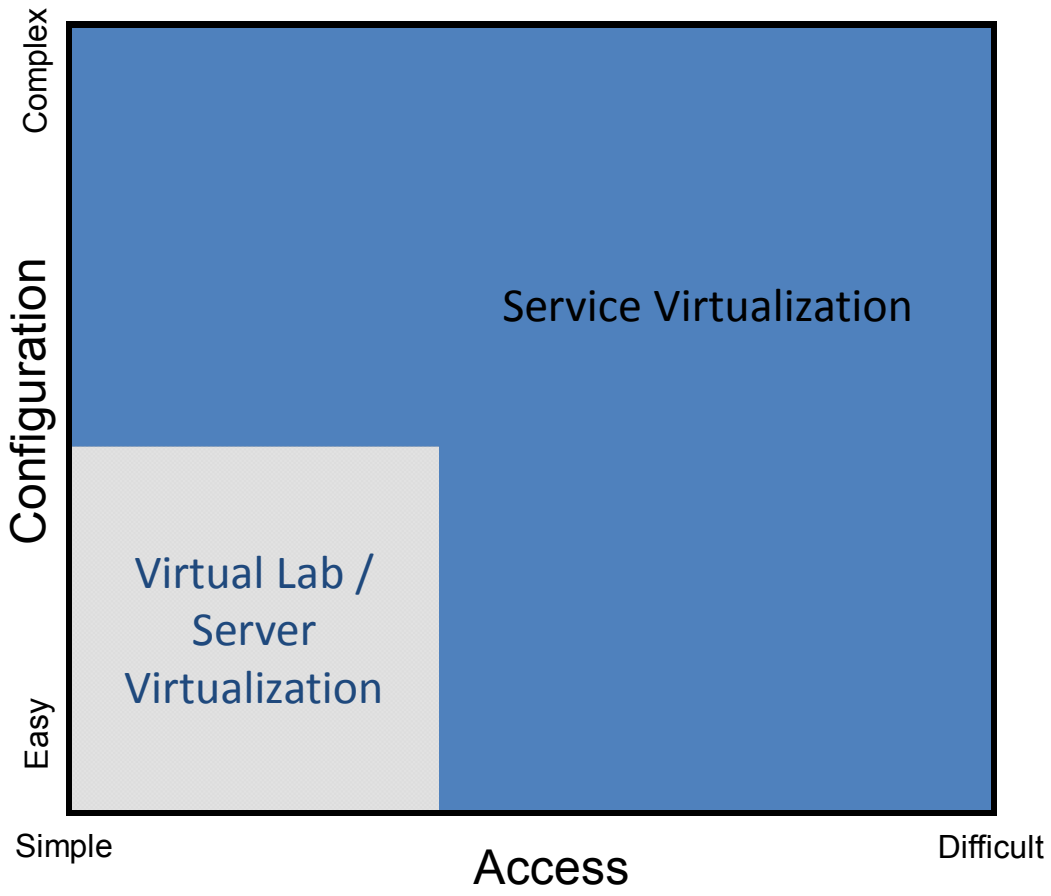
75% of organizations must schedule time in order to access a test environment



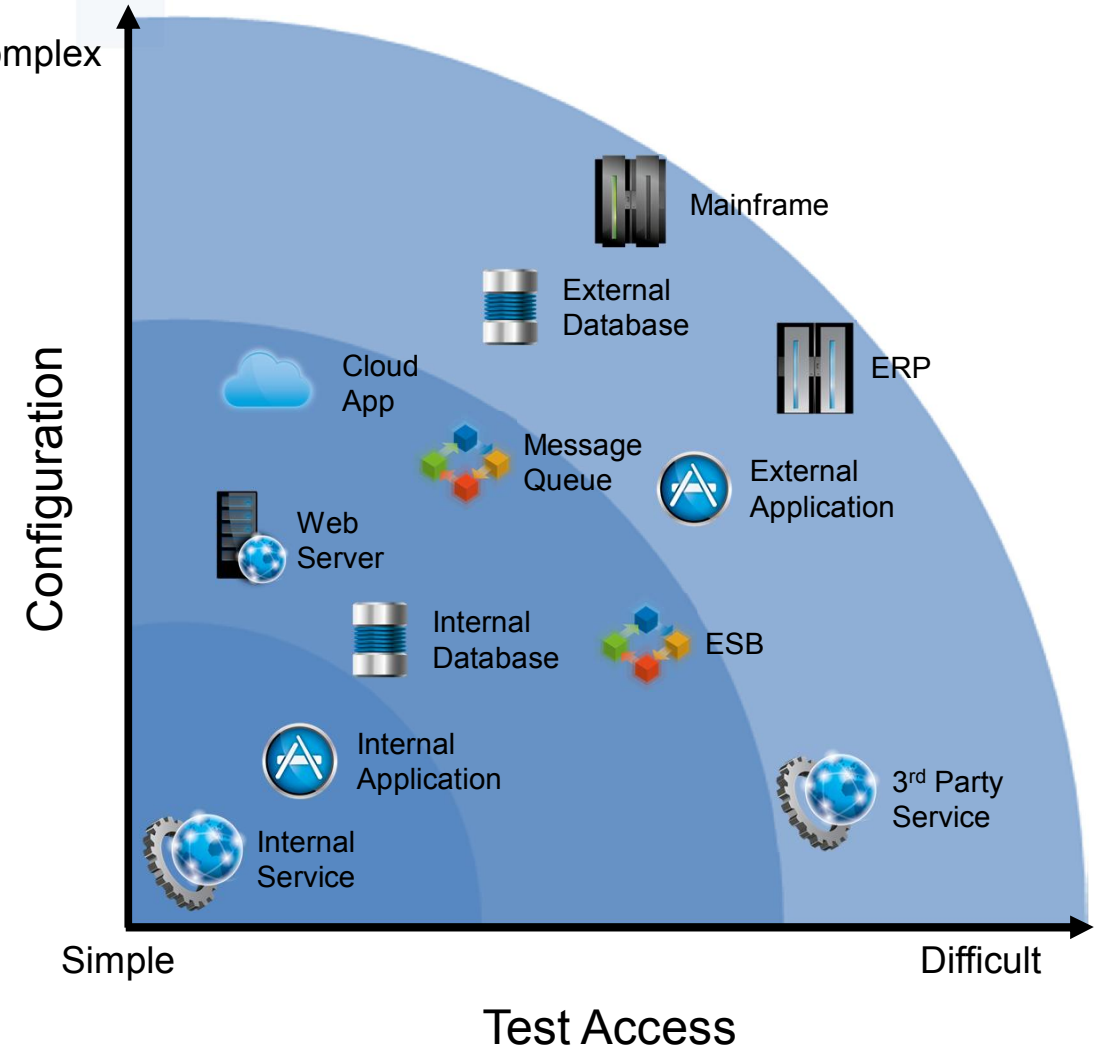
The average time block allowed to be schedule is 4 hours

The Pain

- Access
  - Dependent applications difficult
  - Scheduling conflicts
  - High access fees
  - Geo-political boundaries
  - 3rd party or partner applications
- Configuration
  - Complex to configure
  - No control
  - Limited variability
  - Consumes test time



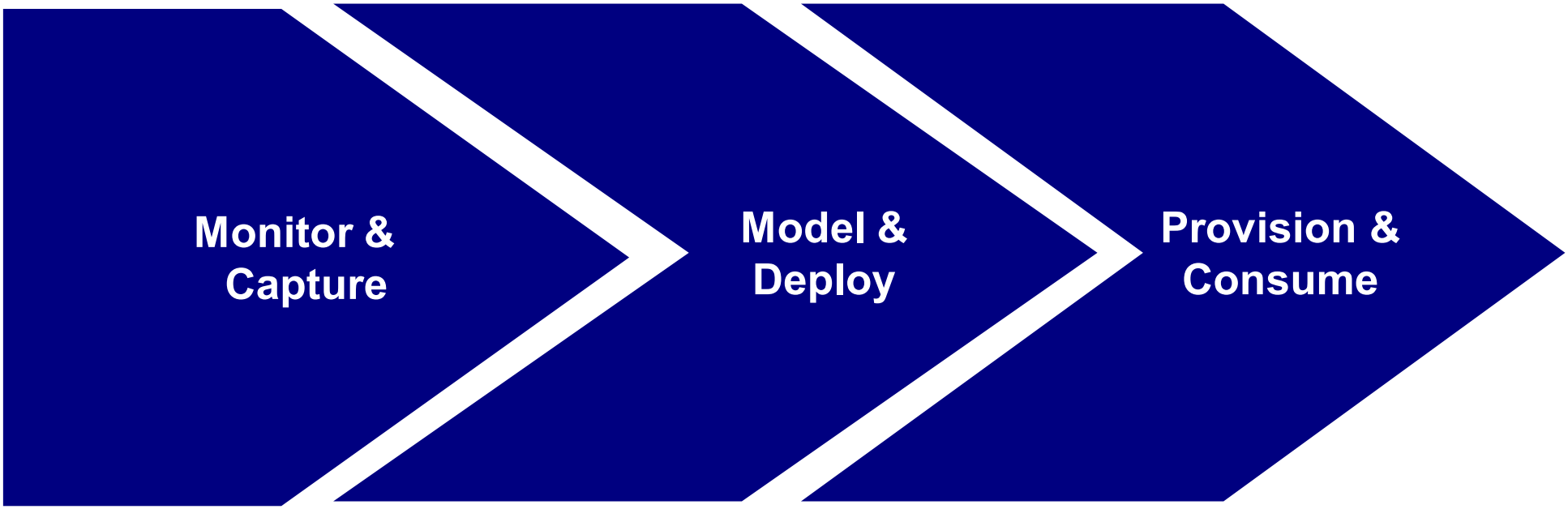
The Degree of Pain



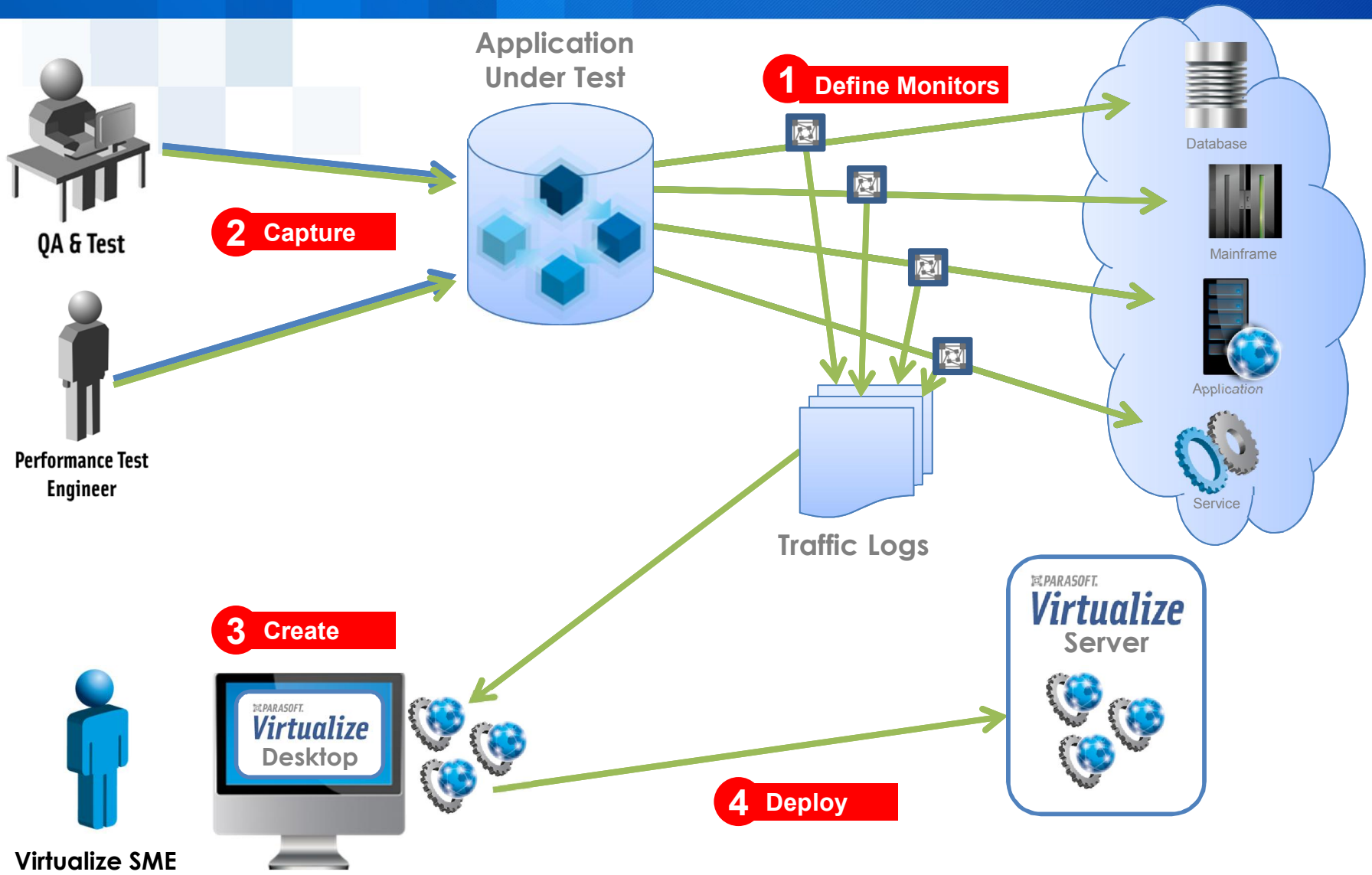
The Remedy : Service Virtualization

PARASOFT

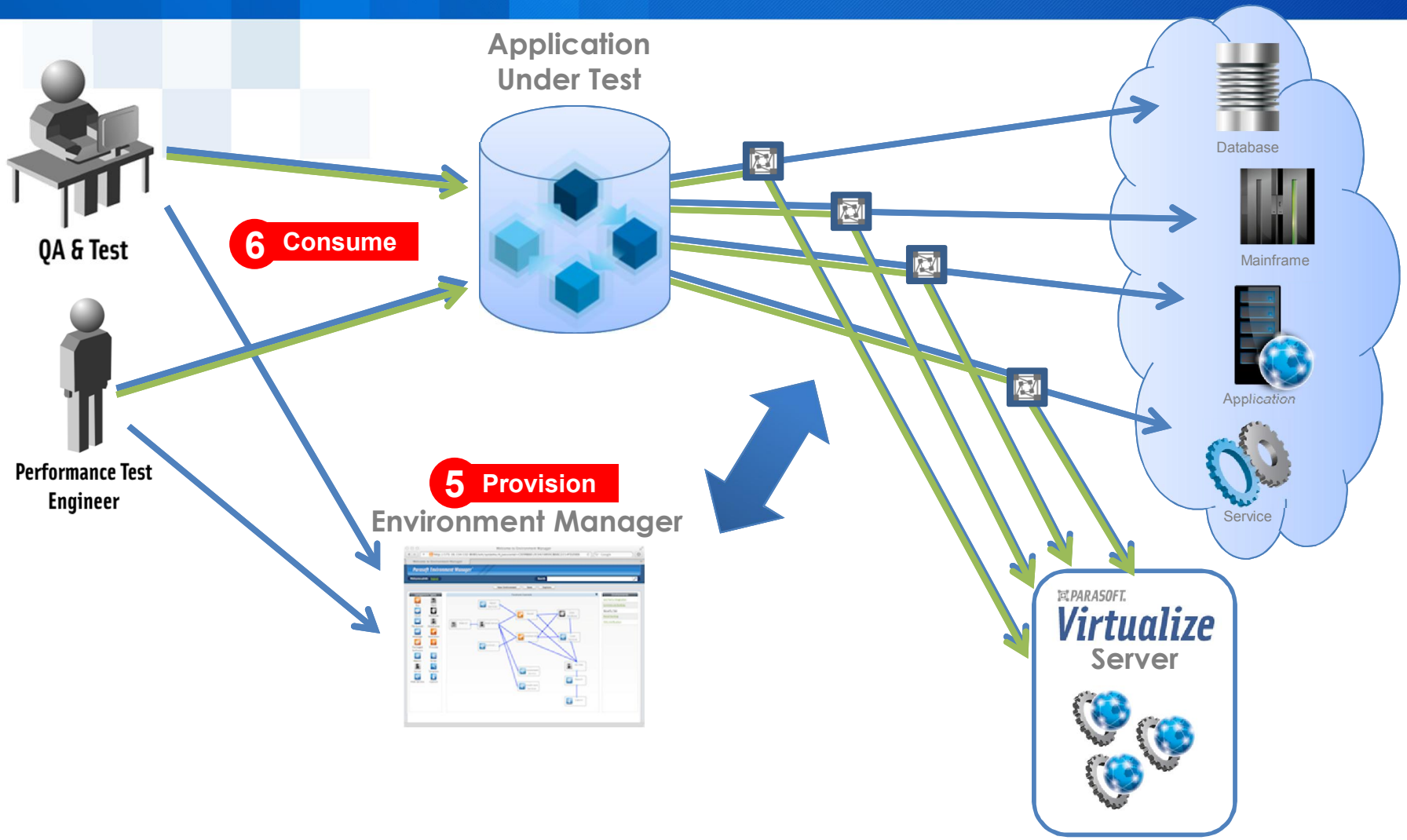
Leveraging application behavior virtualization the team can reduce the complexity and the costs of managing multiple environments while providing on-demand access for development, test and training



Service Virtualization : How does it work?



Service Virtualization : How does it work?



Parasoft Virtualize Customers

- Some customers report that Service Virtualization has:
  - “Eliminated 28 days of wait time “
  - “Cut over 85% of environment configuration time”
  - “Enabled 100% of the test plan to be executed” (this had never happened)
  - “Allowed us to finish 15 days faster than expected”

Parasoft Service Virtualization Wins Dr. Dobb’s Jolt Award

Benefits Service Virtualization

PARASOFT

Increase software quality

Increases productivity

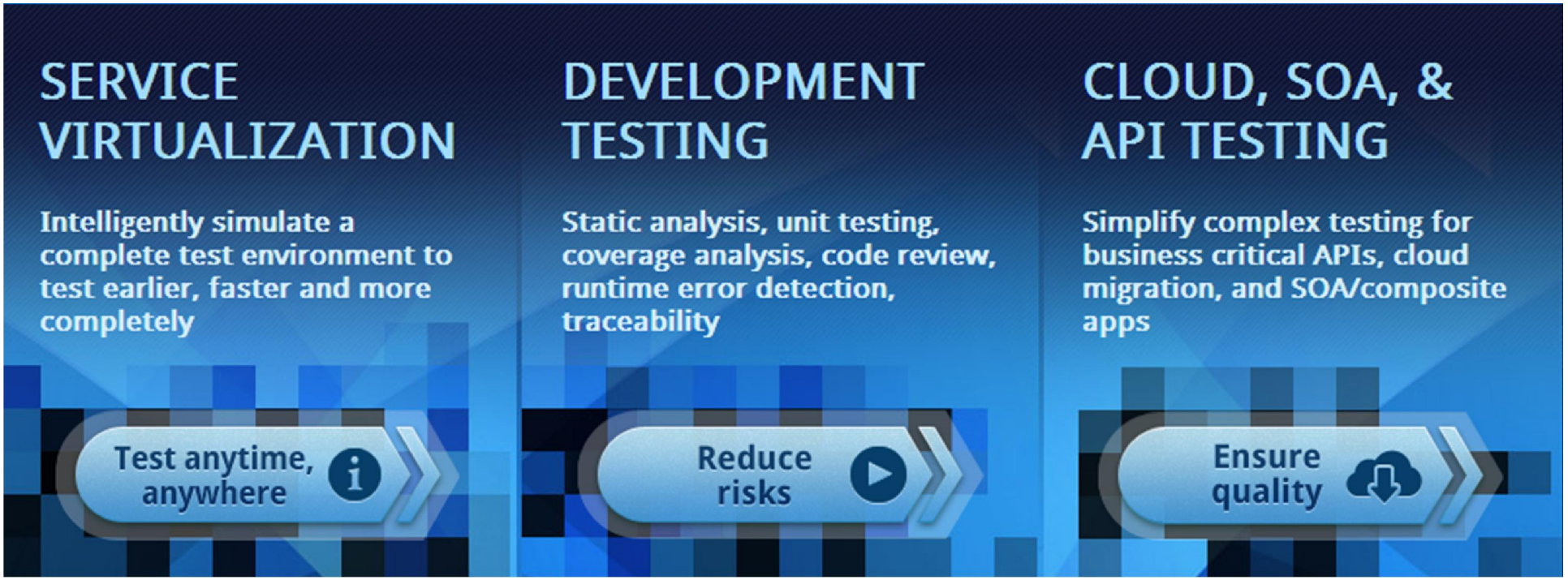
Reduces time to configure applications for test

Reduces hardware costs and system access fees

Reduces the cost for “Test/QA” licenses

The possibilities are stunning. Not only can you have repeatable automated tests, but Environment Manager enables testers to define and assign different response performance profiles to each virtual endpoint. Testers can generate, modify, and run tests including setting the virtual endpoint performance (such as timing, latency, and delay) to emulate peak, expected, and slow performance.

(Jolt Judge: Gary Evans)



- Over 25 years in Software Development & Testing
- HQ in Monrovia, California (USA)
- Asia : Singapore, China, India
- Over 2500 Customers world-wide

Mission Statement: To assist organizations to define and deliver defect-free software efficiently

SERVICE VIRTUALIZATION

Intelligently simulate a complete test environment to test earlier, faster and more completely

Test anytime, anywhere

DEVELOPMENT TESTING

Static analysis, unit testing, coverage analysis, code review, runtime error detection, traceability

Reduce risks

CLOUD, SOA, & API TESTING

Simplify complex testing for business critical APIs, cloud migration, and SOA/composite apps

Ensure quality

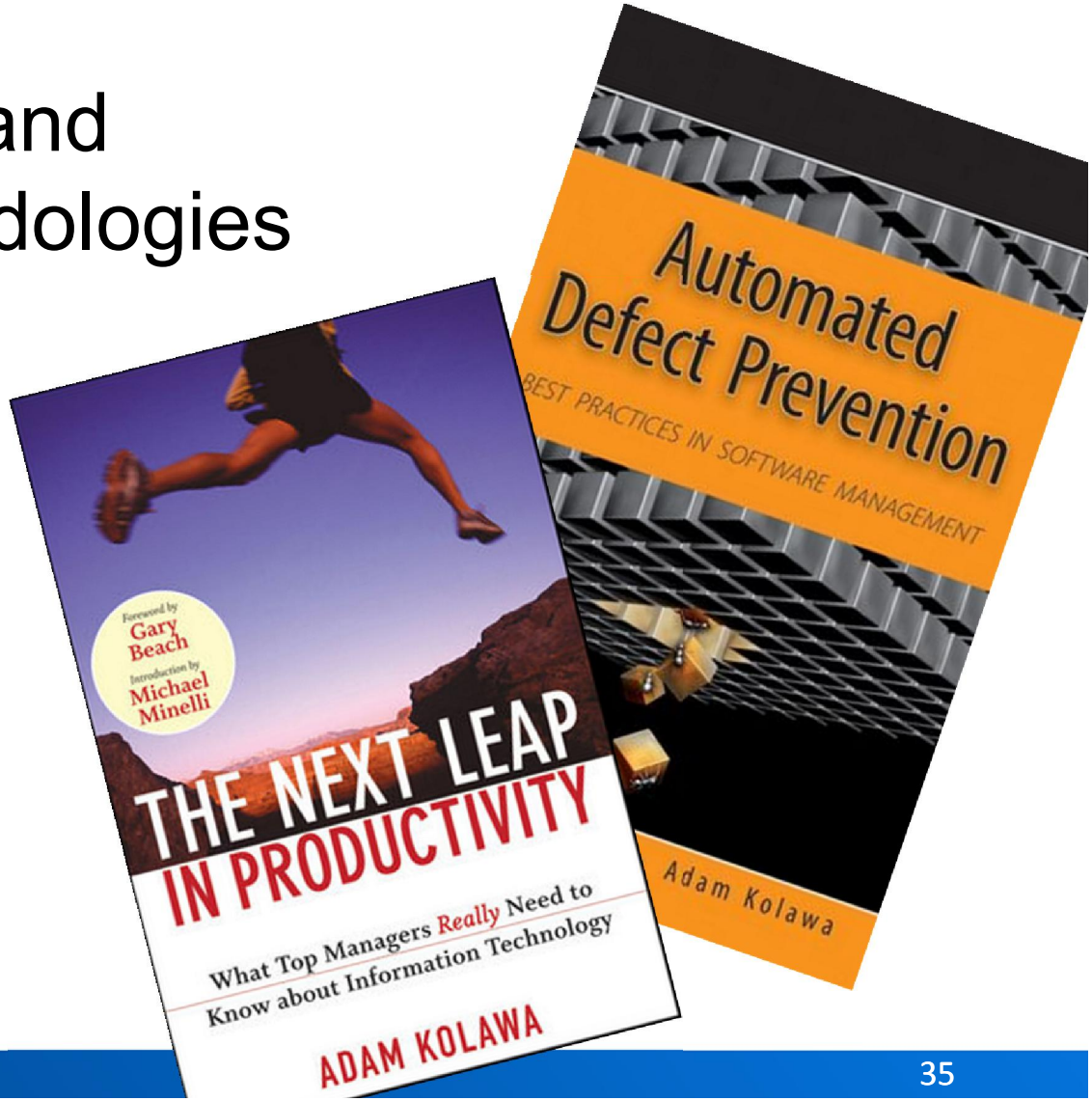
TO ENSURE QUALITY, RELIABILITY AND SECURITY IN APPLICATION DEVELOPMENT & TESTING

Adam Kolawa

PARASOFT

Founder, June 25, 1957 – April 26, 2011

- Introduce New and Effective Methodologies
- Technology is needed to improve
- Human Factors must be taken into account




The End