

Streamline application delivery



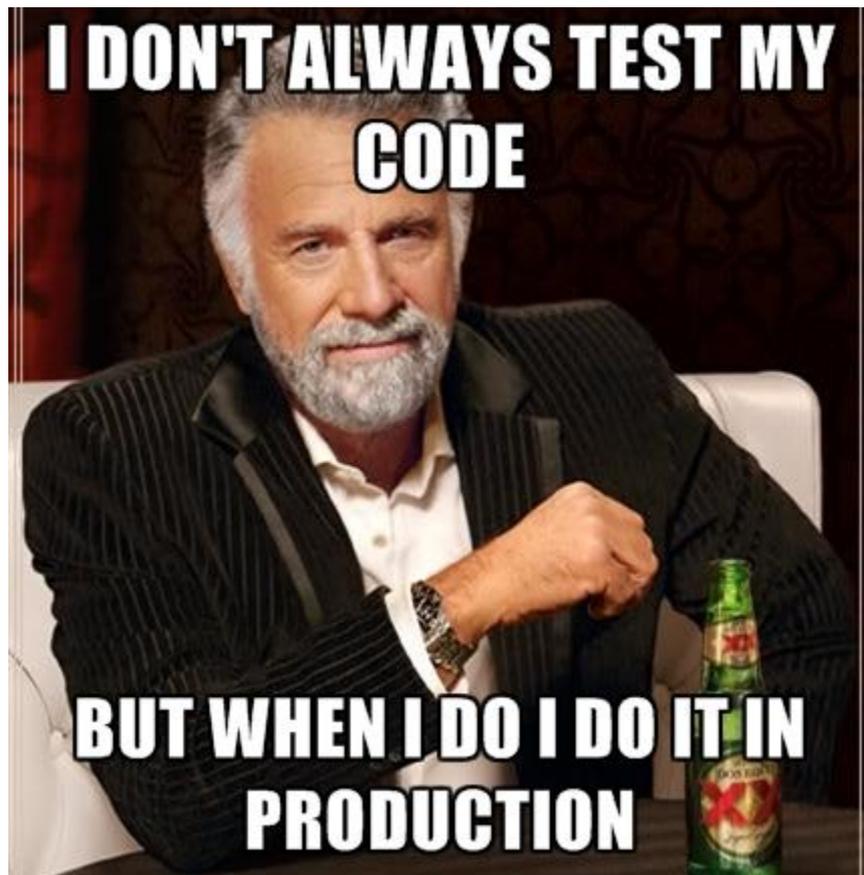
SOA Strategy Pre-Lockdown Vendor VA utilized capabilities briefings-- Group 2

October 6, 2013

agility
made possible™



“Shift Left” streamline application delivery



Agenda

- LISA implemented with SOA ESB
- What is LISA
- What happens when you use LISA
- LISA in the VA
 - VA TRM
 - SEDR Accredited, AITC, Bay Pines (TBA)



Definitions

- Service Virtualization (SV) “Not your Grama’s Hypervisor!” is the practice of capturing and simulating the behavior, data, and performance characteristics of dependent systems and deploying a Virtual Service that represents the dependent system without any constraints, thus allowing software to be developed and delivered faster, with lower costs and higher reliability.
- Release Automation (RA) automating the complex workflows required for agile software movement between development, test and production environments, reducing deployment errors and time-to-market through continuous application delivery.



One-VA TRM Home

Search One-VA TRM

Technology/Standard List

TRM Reports

Submit Idea to Improve TRM

TRM Glossary

Site Map

ONE-VA TECHNICAL REFERENCE MODEL v13.9



CA LISA Service Virtualization

Health Segment Entry

General

Reference

Component

Category

Forecast

Analysis

General Information ?

Website: [Go to site](#)

Description: CA Lisa Service Virtualization product provides capabilities around testing, validation, defect discovery, and simulation of services in a distributed multi-tier component-based infrastructure. It provides comprehensive testing in Service Oriented Architecture (SOA) or other component-based development environments, including heterogeneous distributed environments. It enables regression, performance, and functional testing of individual services or complex composite applications, including testing of composites where some services are not yet available. It provides the ability to conduct functional, integration, load, and performance tests of the solution using virtualized services to emulate end points for the many interface components in a Service Oriented Architecture (SOA) environment without having to access actual end point systems. End point systems can be rapidly simulated for tests. CA LISA provides virtualization of external systems providing consistent responses and reactions when testing Service Oriented Architecture (SOA) applications. Mock services are deployed when a system under test requires interaction with unavailable systems.

Decision: **Approved. This Technology has been approved for general use.**

Decision Constraints: No constraints are specified.

Decision Justification: CA LISA is a mature and stable technology, highly interoperable and portable to all VA approved Unix and Windows platform.

Decision Source: TRM Mgmt Group

Decision Process: One-VA TRM v13.9

Decision Date: 09/30/2013

Current Status: Analysis Completed

Introduced By: TRM Request

Vendor Name: [CA](#)

Accelerate innovation ... time-to-production is everything



Faster delivery:

- New applications
- Changes to existing apps
- Bug fixes

Higher reliability:

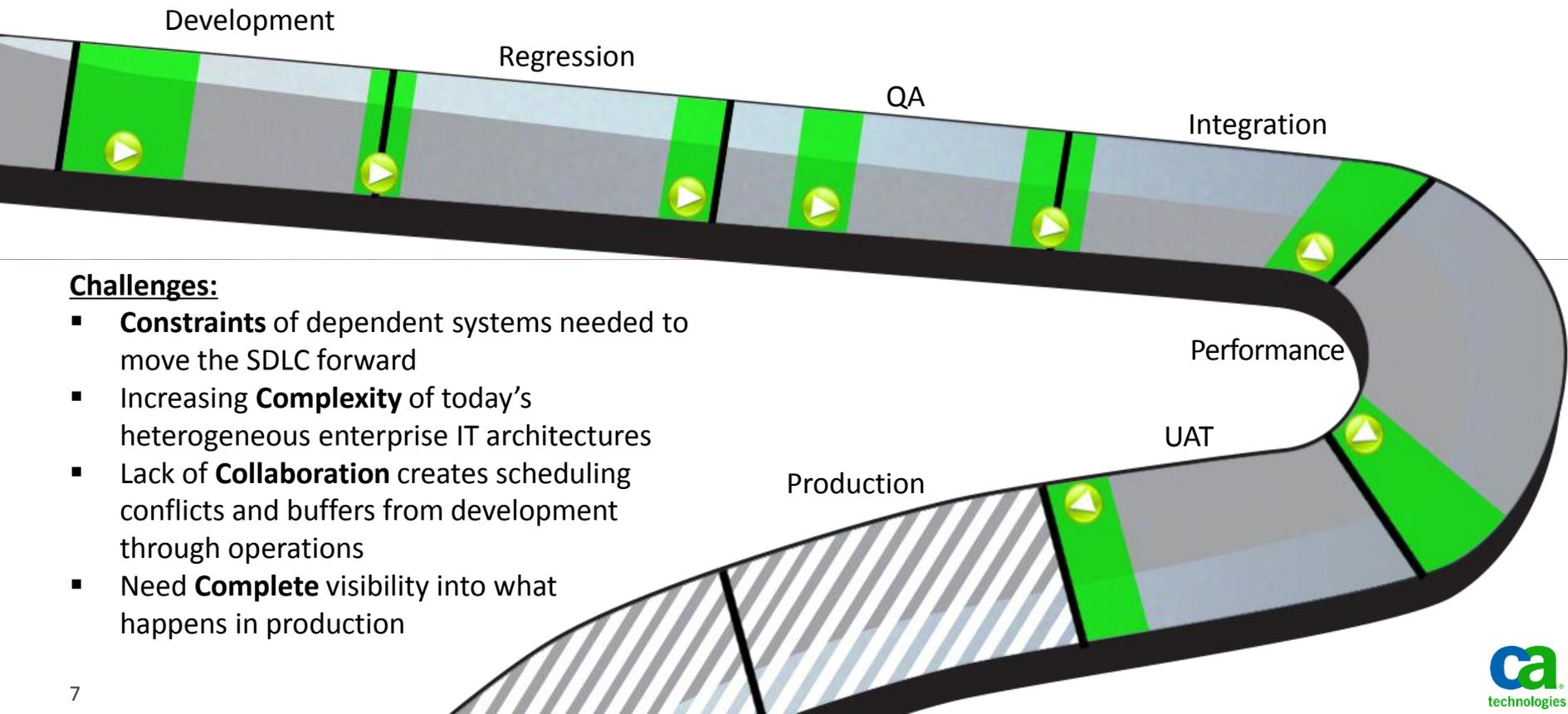
- Software is the customer experience
- Quality and Performance are critical

Against complexity:

- Composite services and heterogeneous systems
- Constant new releases atop existing ones
- Distributed development teams and IT partners

Accelerating application time-to-market

how do we increase “touch time” and reduce “idle time”?



Challenges:

- **Constraints** of dependent systems needed to move the SDLC forward
- Increasing **Complexity** of today's heterogeneous enterprise IT architectures
- Lack of **Collaboration** creates scheduling conflicts and buffers from development through operations
- Need **Complete** visibility into what happens in production

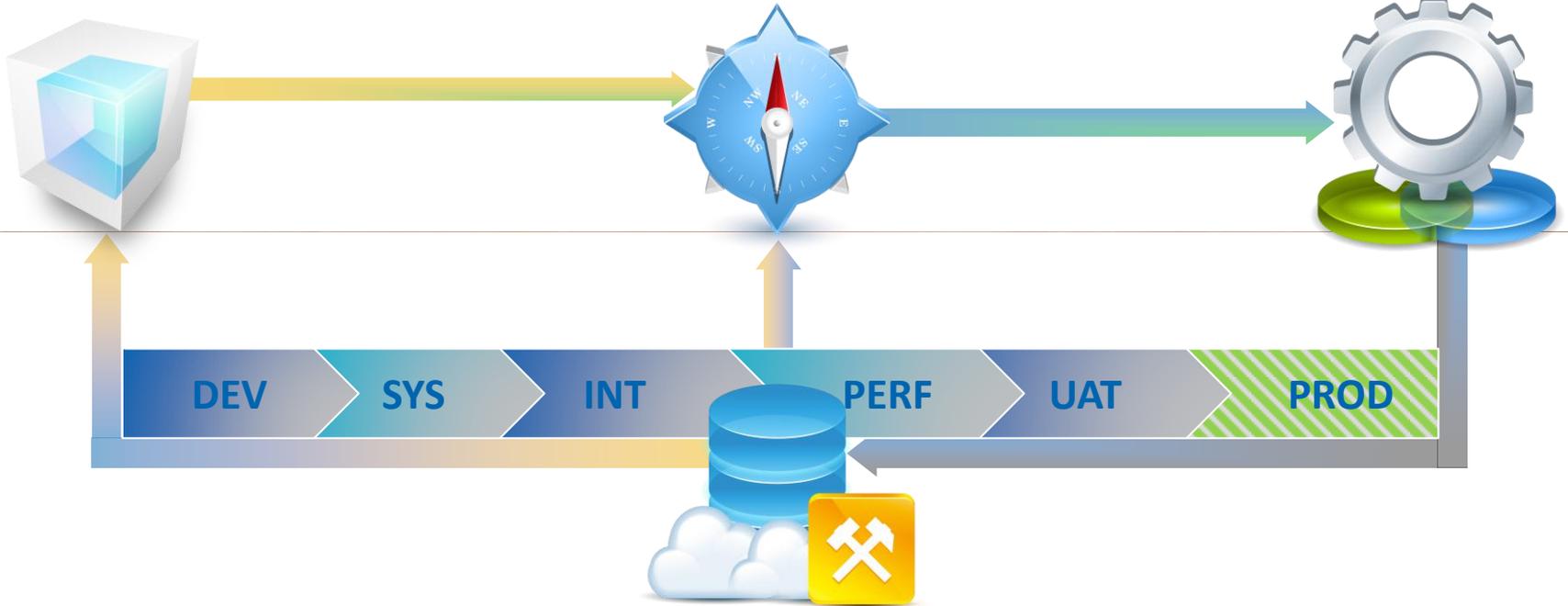
capabilities for high-velocity application delivery

enabling the “4C’s” of DevOps

Constraint-free Development

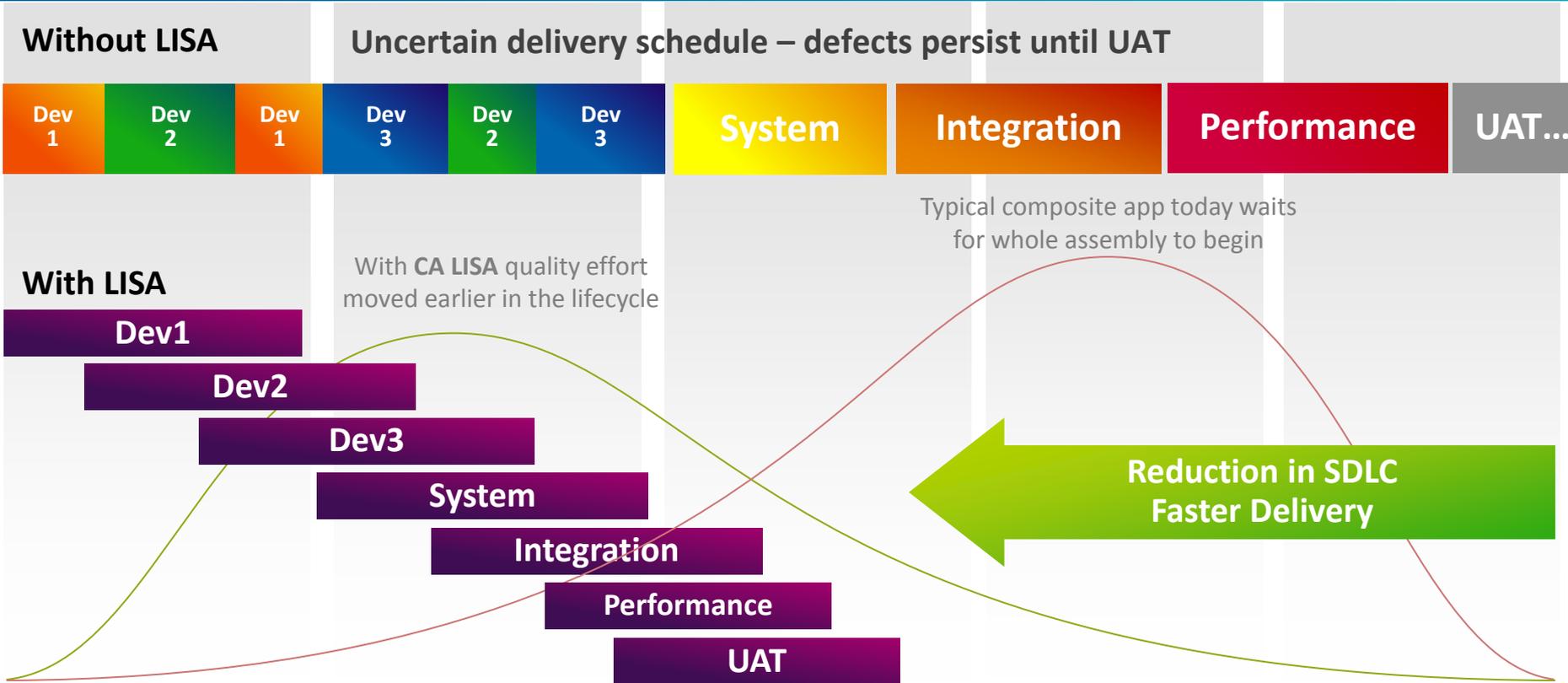
Continuous Application Delivery

Complete Monitoring

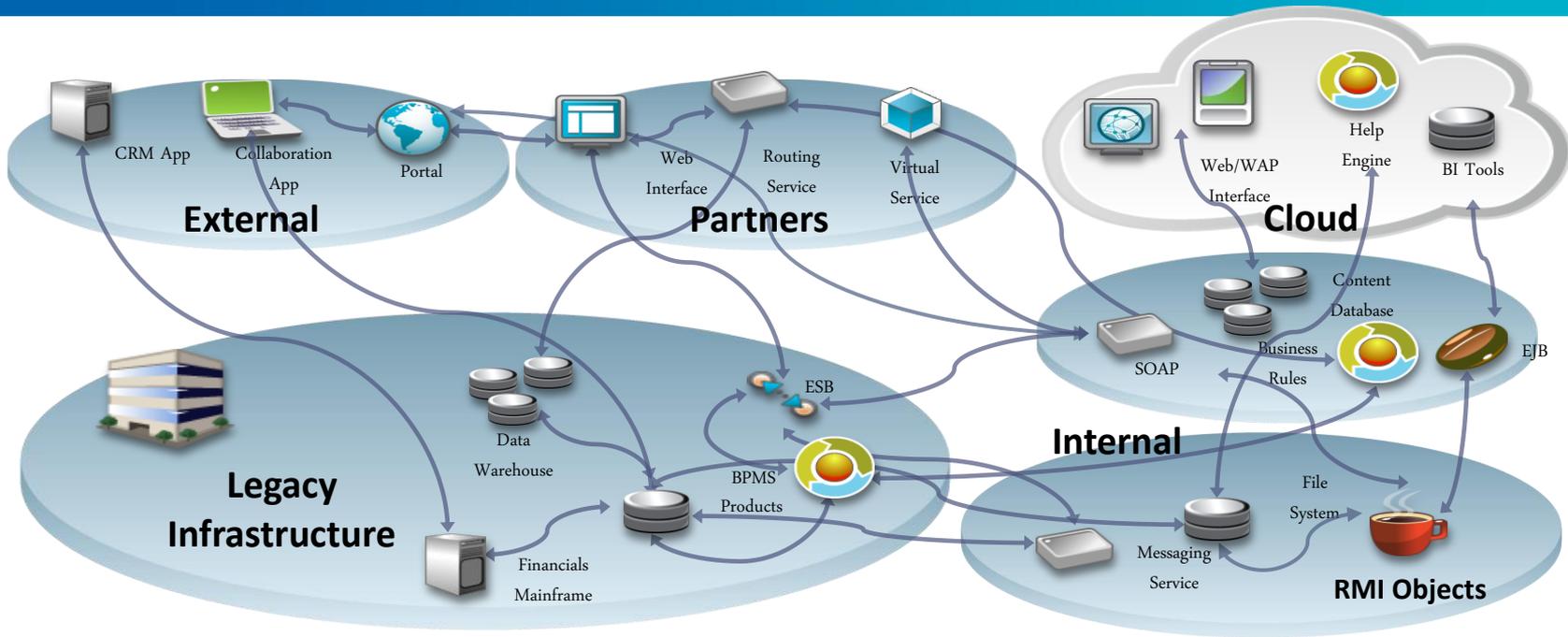


Collaboration between Dev and Ops

Constraint-free development solution: service virtualization, “shift-left” the SDLC



What we are faced with



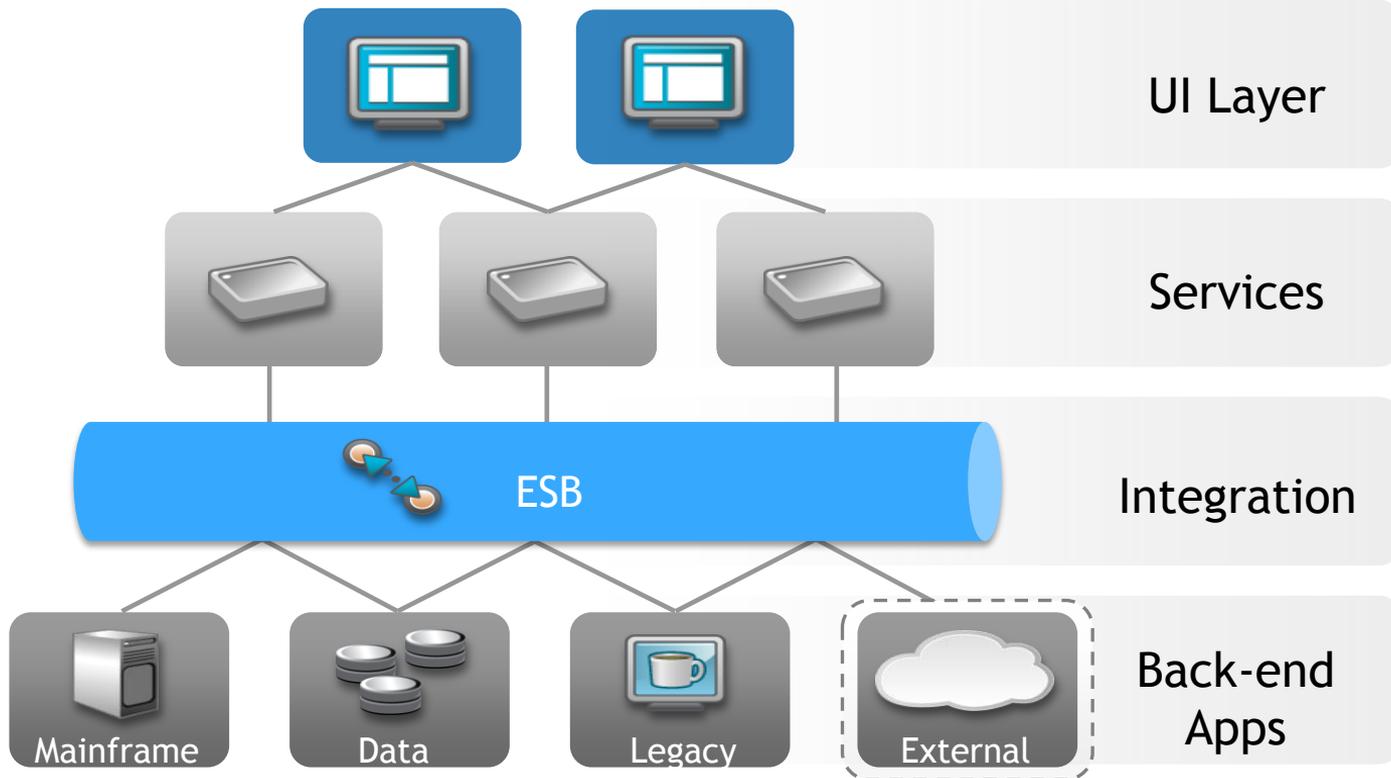
of Interconnected Components

of Inter-dependent Teams

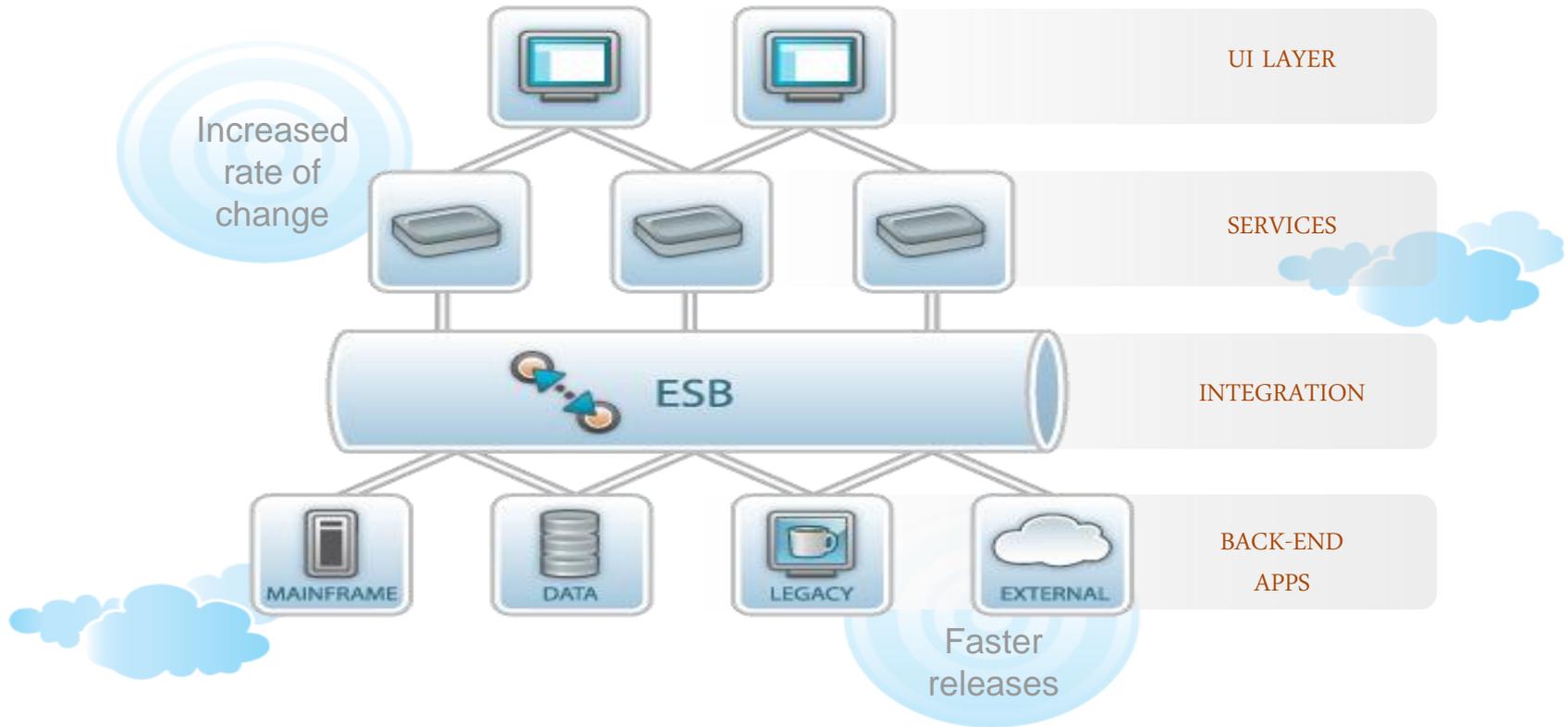
of Heterogeneous Technologies

Rate of Change

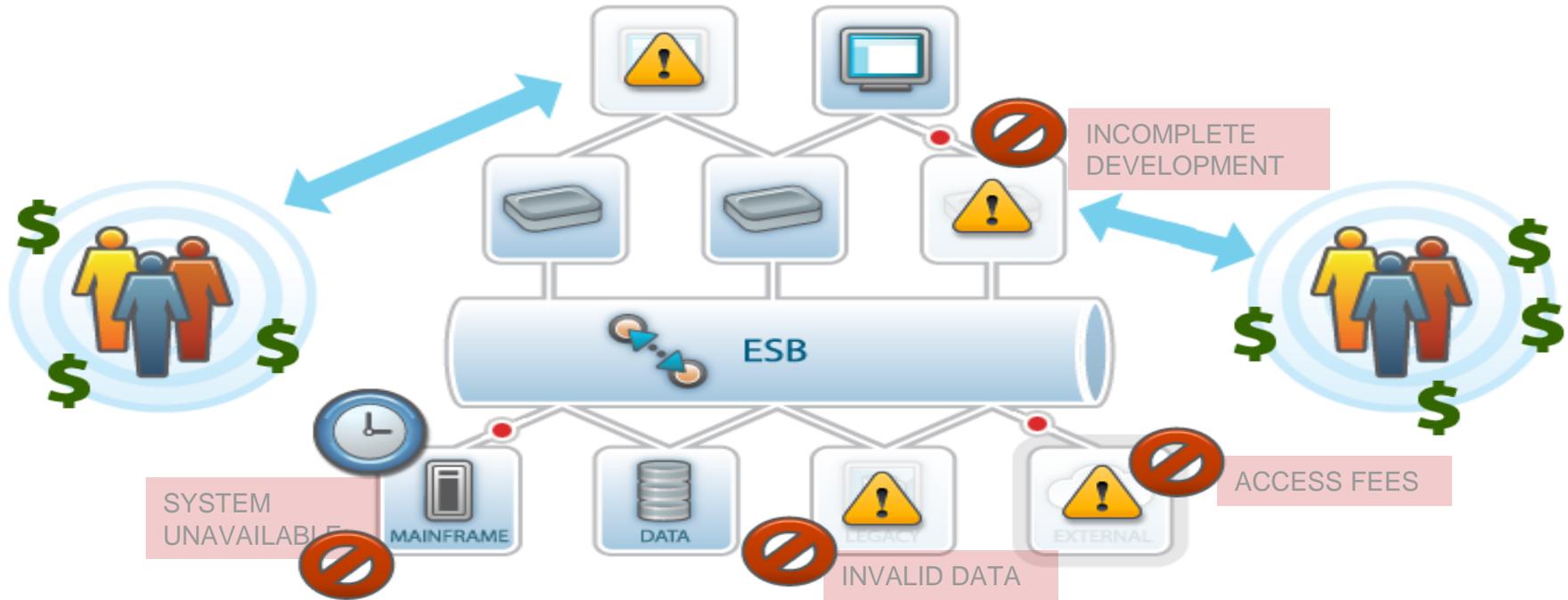
A Typical Application



Challenges: Change and Complexity

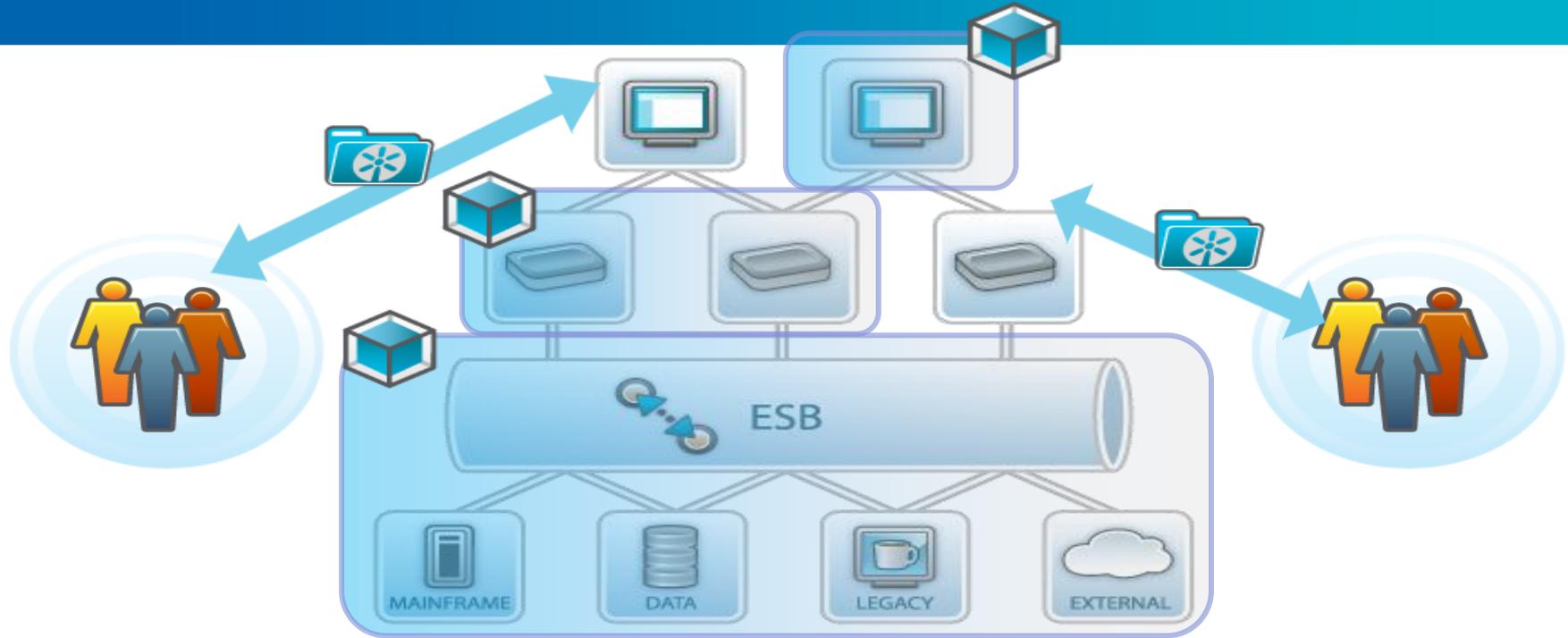


Problem: Constraints



*“I can’t do anything until I have everything...
and I never have everything!”*

Solution: Service Virtualization



24/7 availability

Saved \$30M

Reduced Timeline by 50%

What can be virtualized?

- HTTP/S
- Web Services
- JDBC
- MQ
- JMS
- TCP
- ICD 10
- HL-7
- Java
- DRDA
- CICS Link
- And many more:
 - Sockets
 - Restful web services
 - TIBCO EMS
 - FTP and SFTP
 - etc., etc.



Examples of LISA Virtualized Applications



- Mainframe



- Risk Solution Providers



- Credit Bureau



- SMS Gateway Providers



- eBill Services



- Card Processing Services



- Motor Vehicle Records (MVR) System



- Teleconference Service Provider



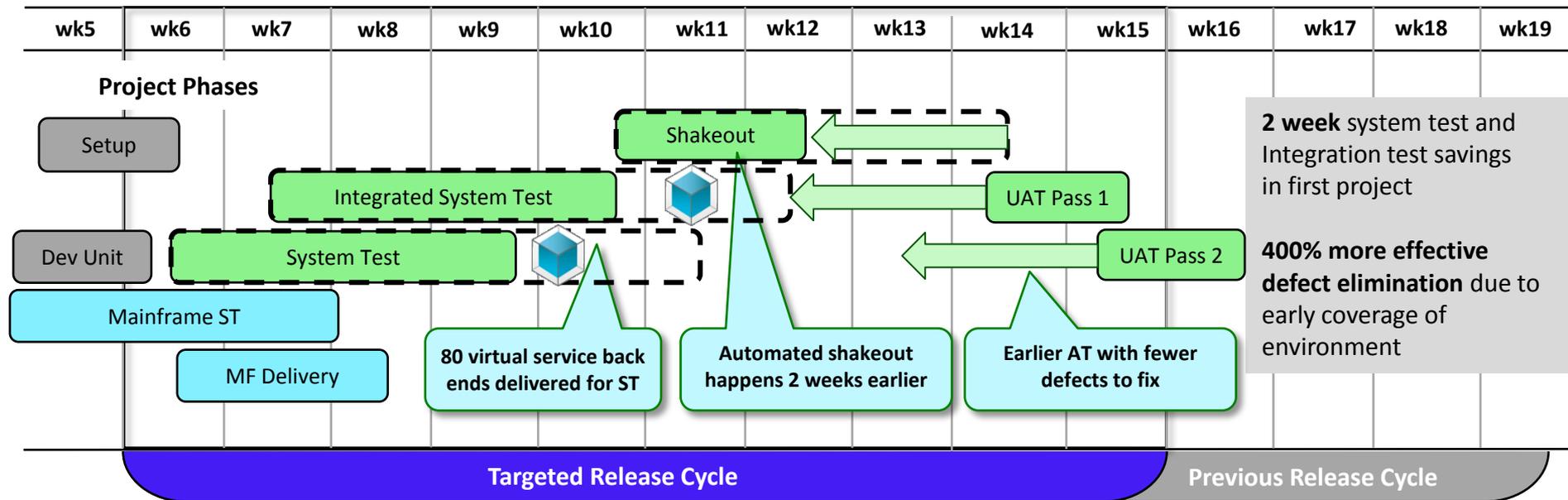
- VOIP Phone / Conference Bridges



- Global Distribution System (GDS)



CA LISA service virtualization customer success

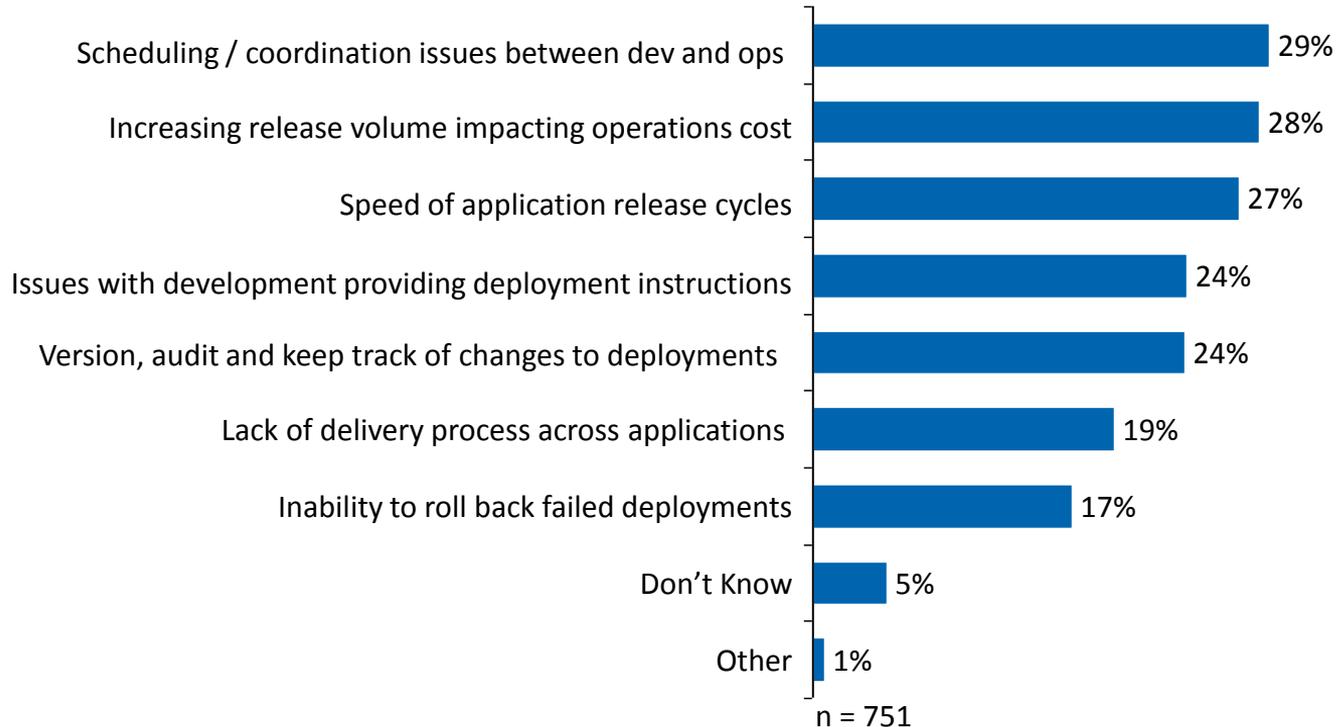


Large US Telco – Popular cell phone launch

- Reduced software release cycle time by 33%
- 400% increase in defects identified
- 4 weeks to achieve 100%+ ROI – \$1.6M

you told us you are struggling to releasing applications

Release Management Pain Points



..and are looking for Continuous Delivery to solve the problem

Transforming/Speed →

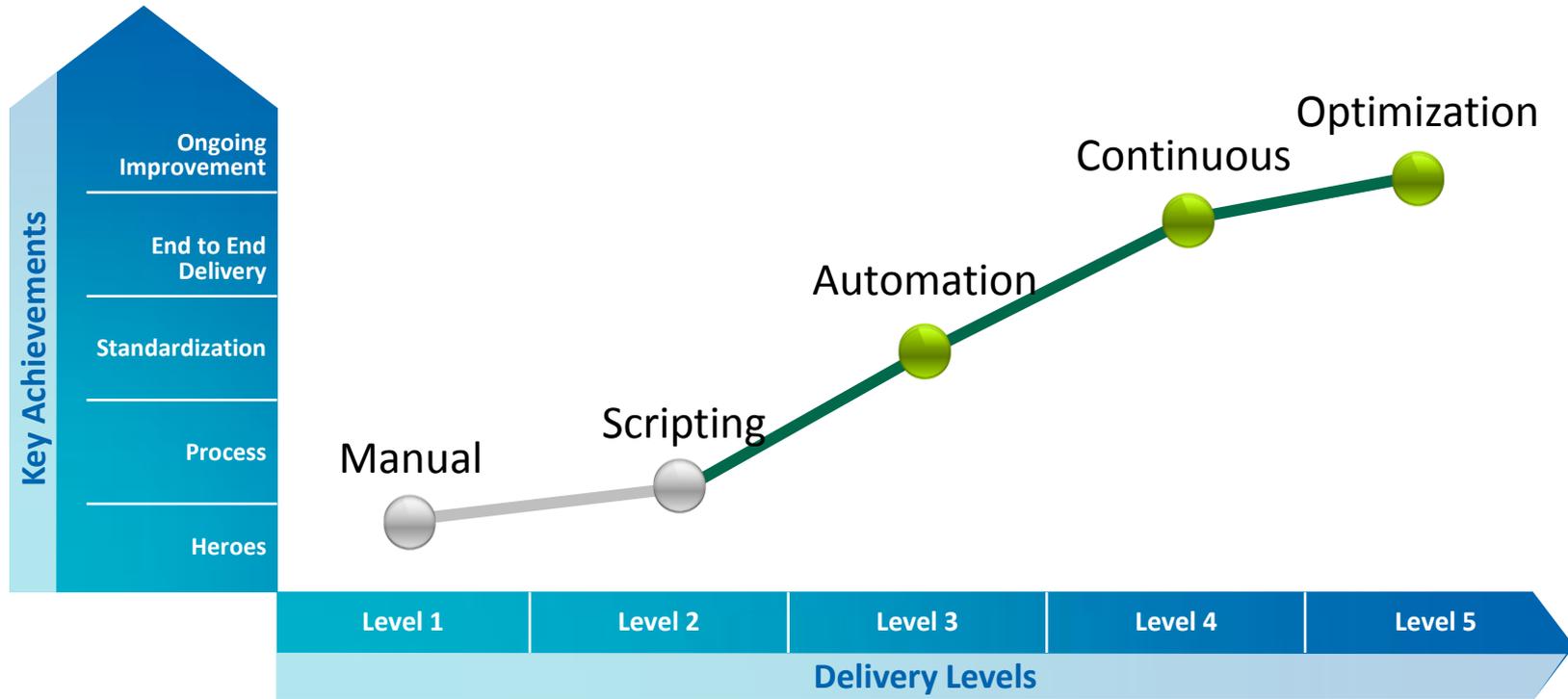
← Streamlining/Control



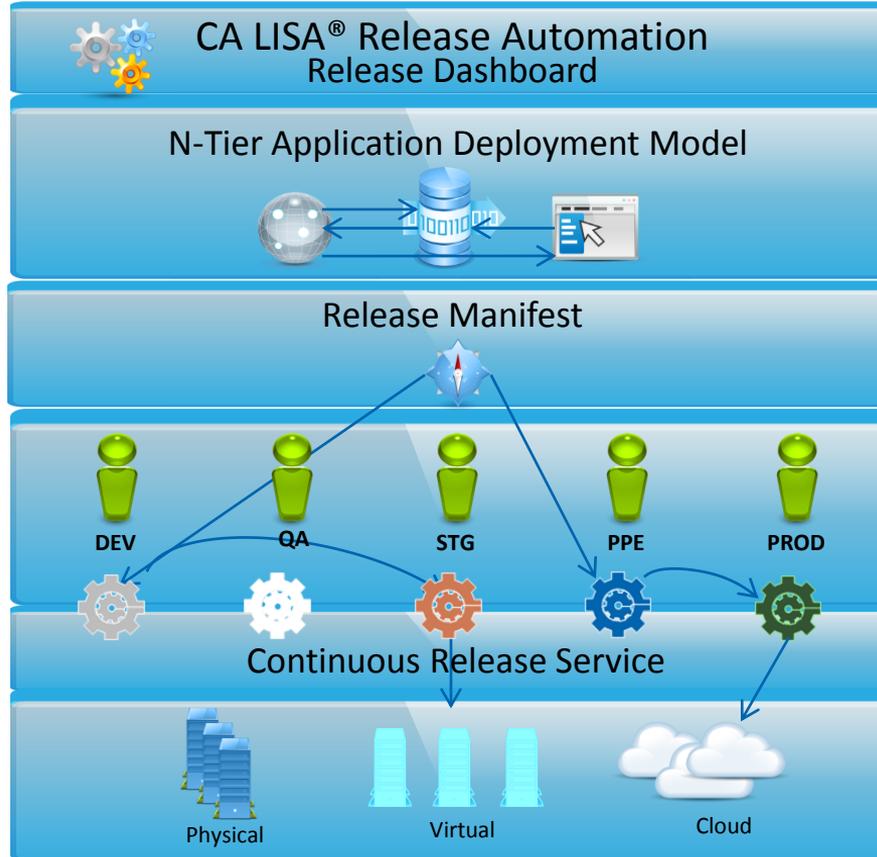
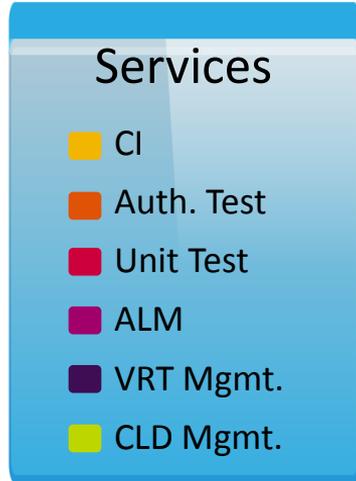
- Agile – more frequent releases with lighter feature payload
- Test automation – higher quality, more efficient releases
- Fewer stages – access to more production-like environments to reduce cycle time

- Virtualization – self-service system for standard environment requests
- Audit – summary reports for changes and compliance
- Repeatable process - want to ensure consistency and high quality

application delivery maturity model



continuous application delivery solution



Release Reporting

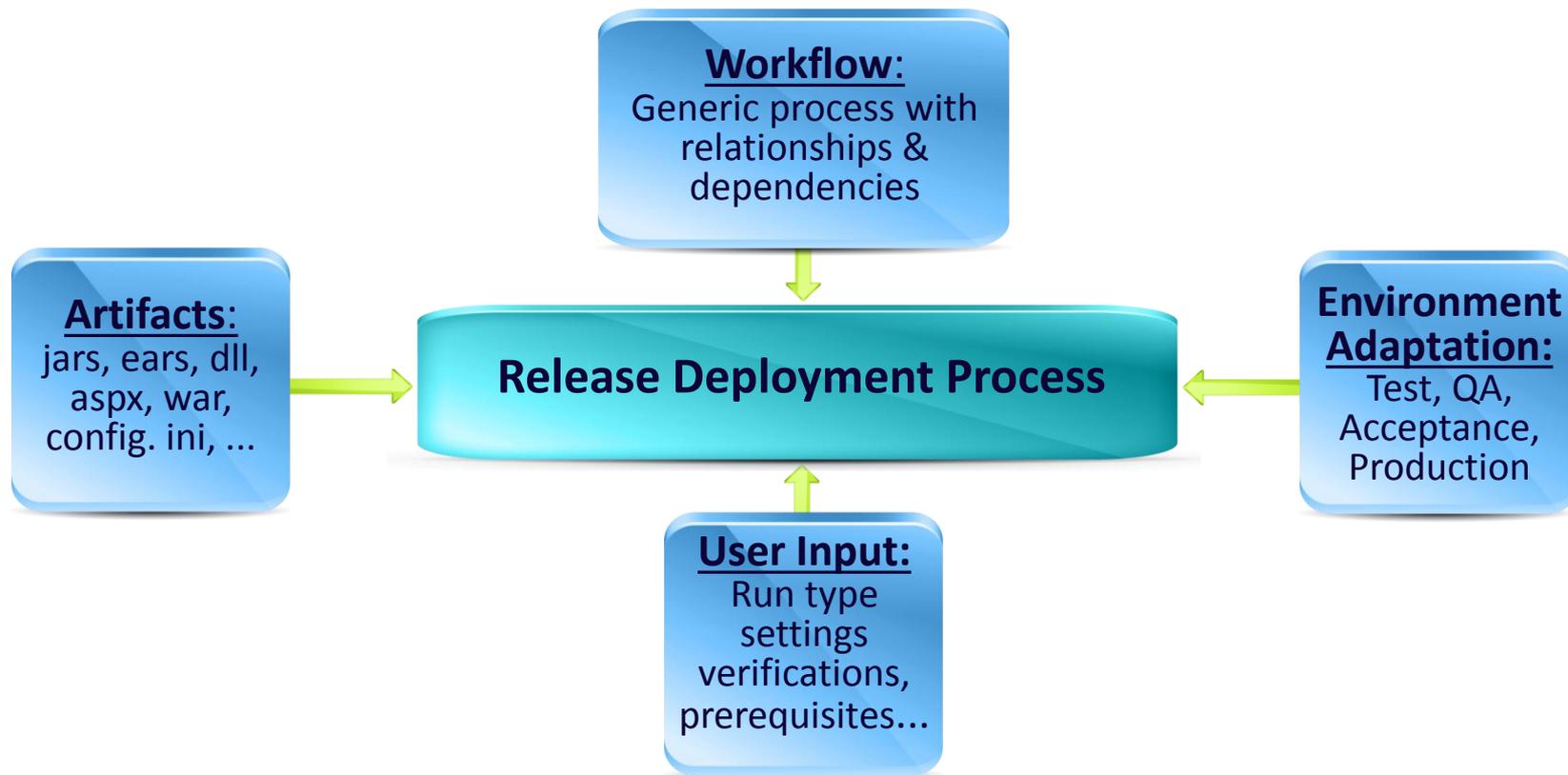
**Generic Application
Release Model**

**Releases are built
dynamically**

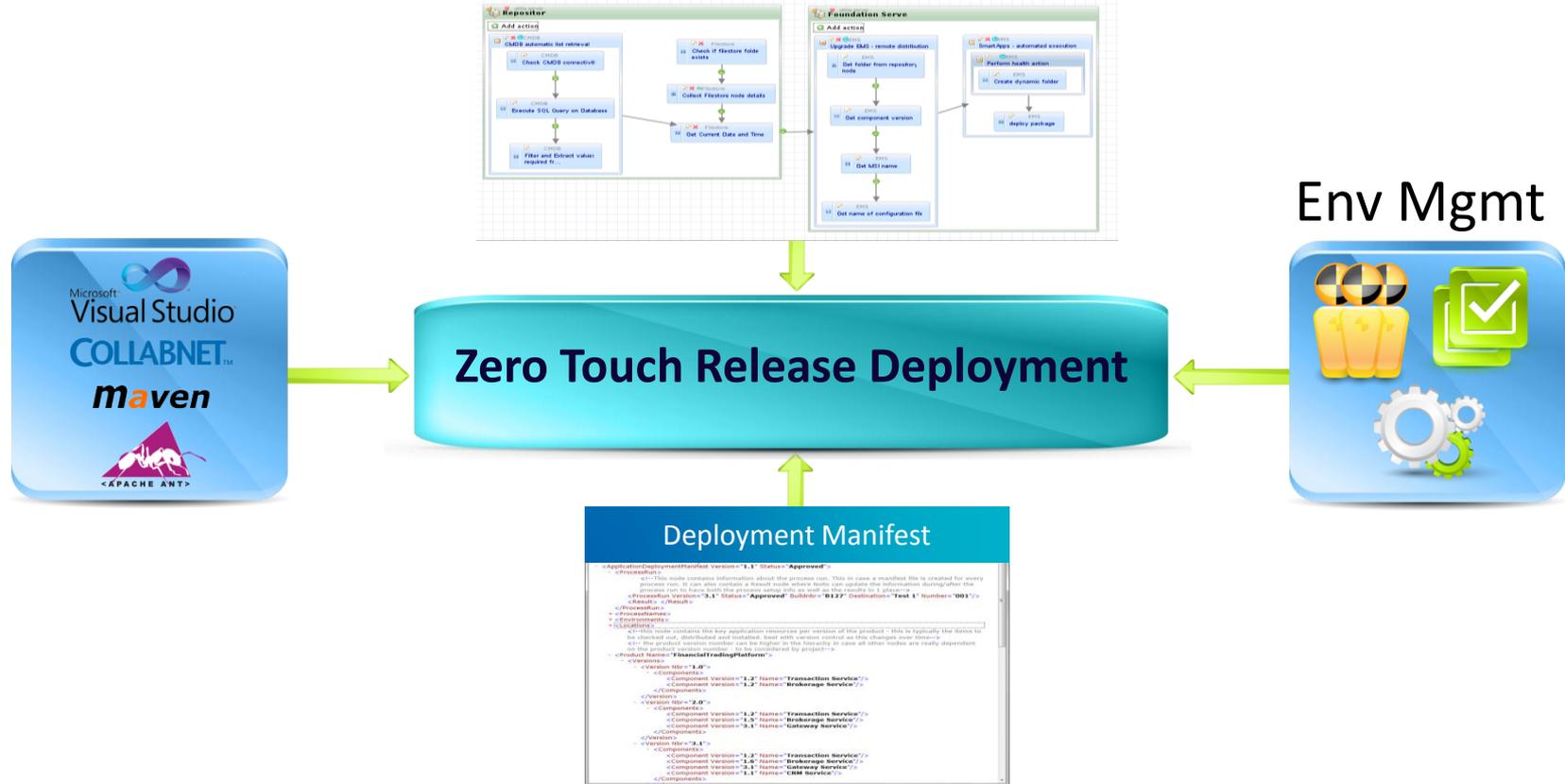
**Automated &
Continuous Release**

**Hybrid Environment
Deployment**

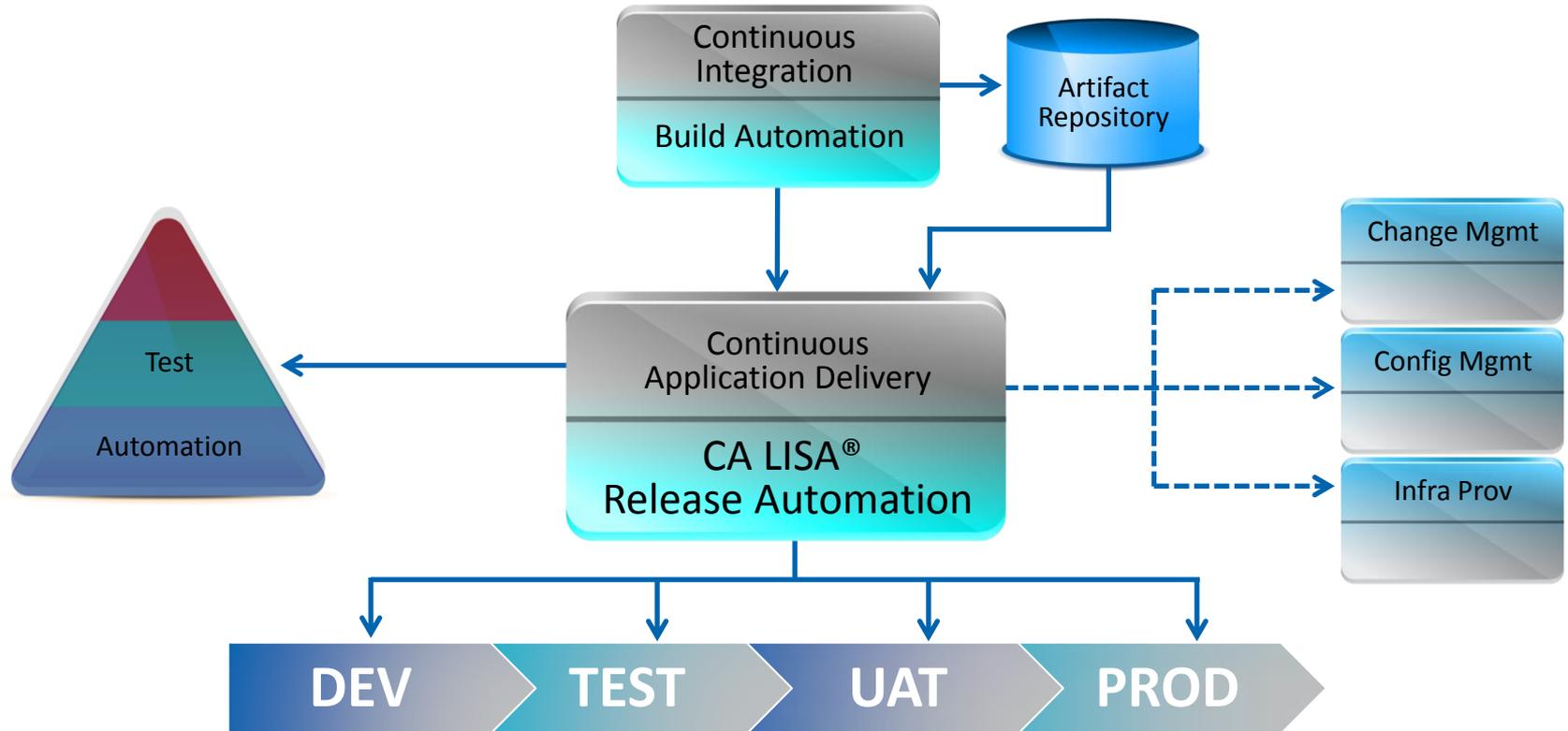
application delivery automation



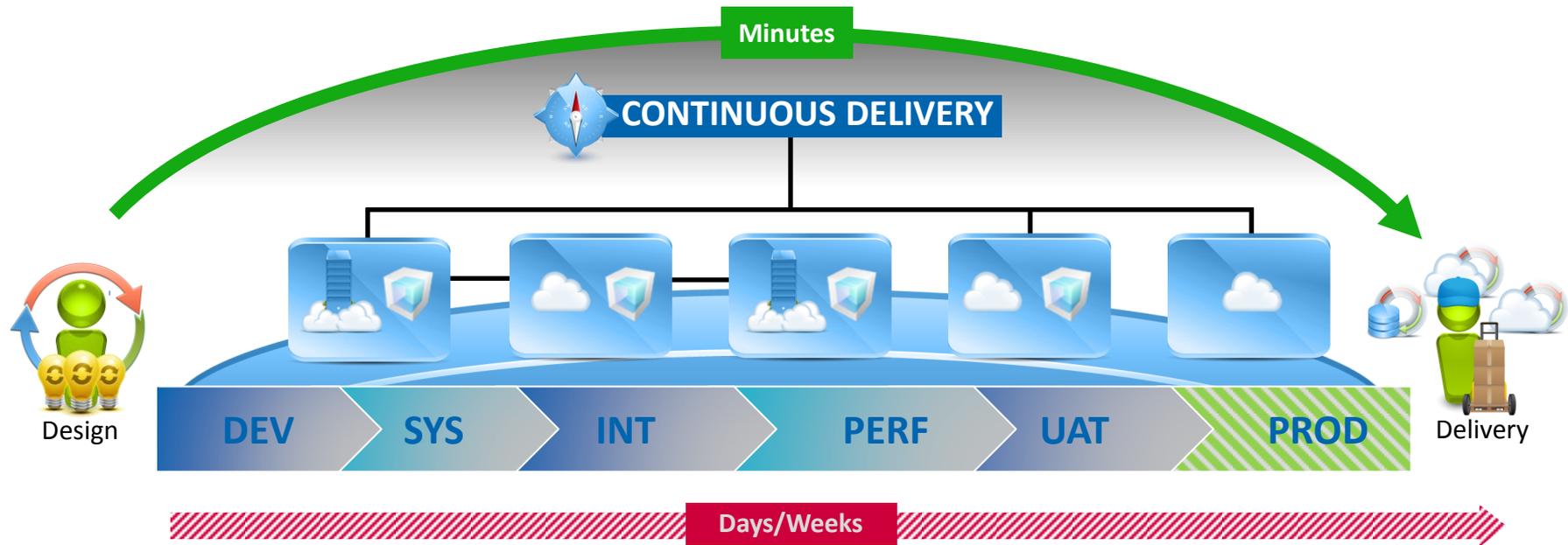
level 3 – application release automation



level 4 – continuous application delivery



continuous application delivery solution: Release automation +



Challenge

- Ensure rapid and consistent application
- Move to agile dramatically increased the volume and frequency of releases
- Eliminate manual release processes and bottlenecks
- Accelerate its time-to-market, increase efficiency and reduce errors

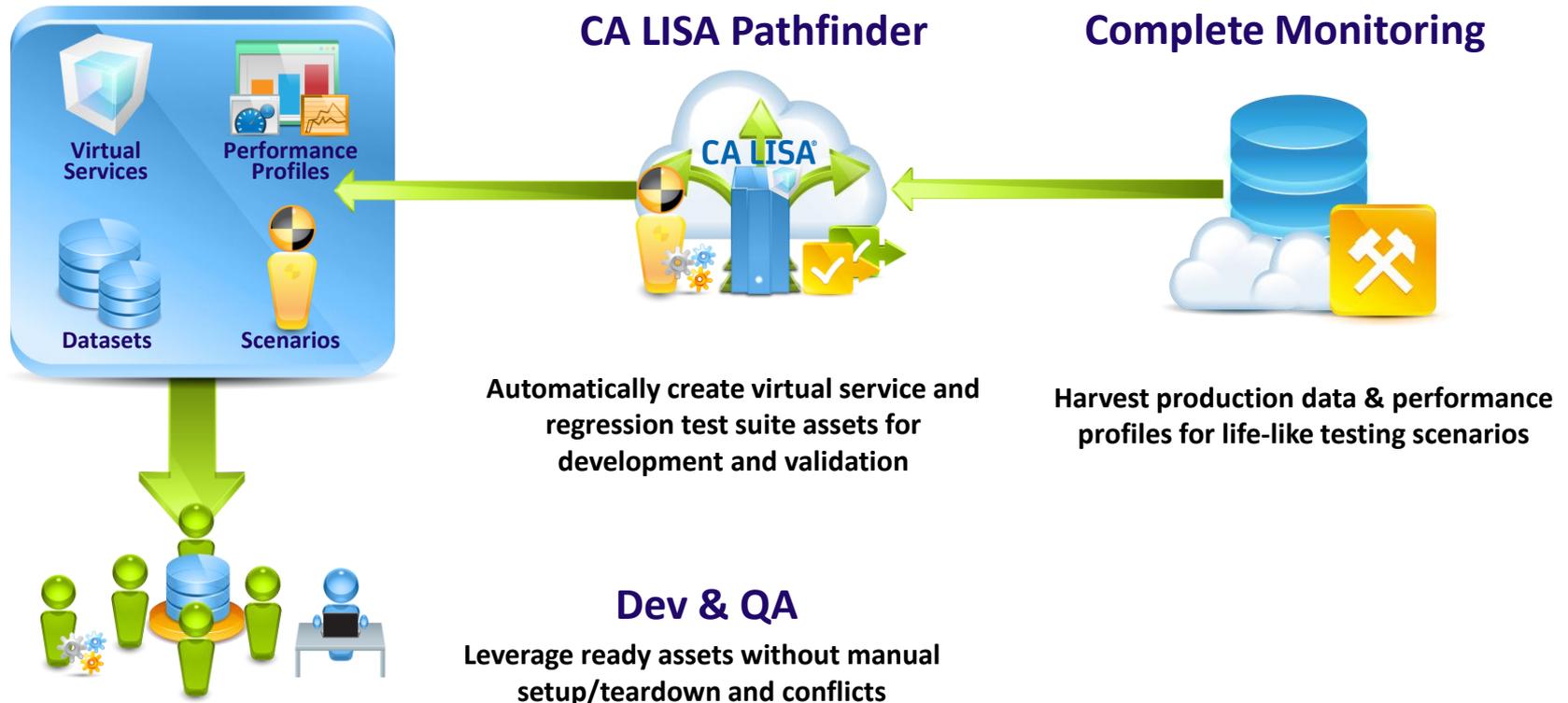
Solution

- Cut release times from days to mere minutes
- Enable application agility needed to meet customer demands, beat the competition and rapidly expand their international footprint.

Results

- 67% of applications were deployed into staging by the end of the first week
- 95% of applications were deployed into staging by the middle of the second week
- Deployments could then be released to pre-production and production environments, reliably and predictably, without the need to modify the deployment process
- Reusable release processes which took just two weeks to build, can be applied to unlimited regions across five environments and cut release times from days to minutes

collaborative data mining & complete Monitoring solution: better realism and data with less effort



customer outcomes with CA LISA[®] enabling the 4Cs of DevOps

Constraint-free Development

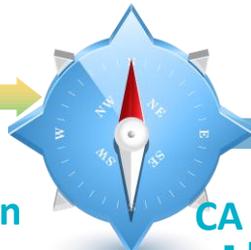
Continuous Application Delivery

Complete Monitoring



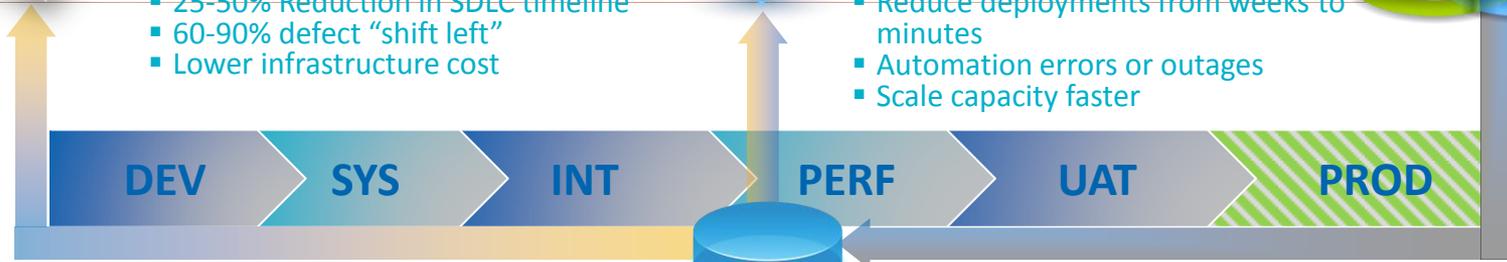
CA LISA Service Virtualization

- 25-50% Reduction in SDLC timeline
- 60-90% defect "shift left"
- Lower infrastructure cost



CA LISA Release Automation

- Reduce deployments from weeks to minutes
- Automation errors or outages
- Scale capacity faster



*Collaboration between
Dev and Ops*

CA LISA Data Mining

- Shift defects left with automated coverage
- Reduced time and effort to create live-like scenarios

SIGNIFICANTLY REDUCE TIME TO MARKET

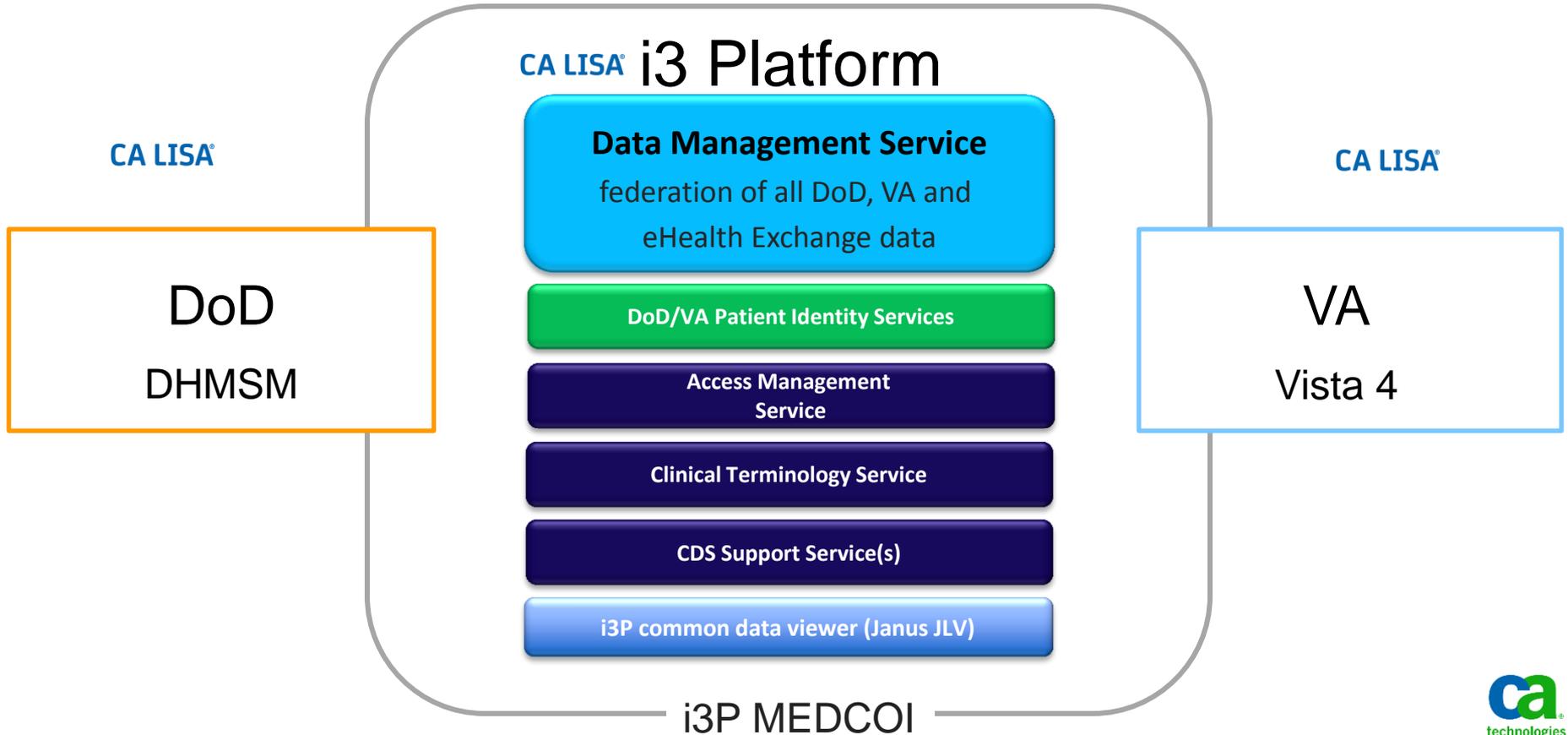
- **Touch time increases:** Continuous Delivery without constraints that cause idle time
- **Shift-Left:** Quality happens up to 2 phases earlier, and deployment happen in minutes for faster overall delivery by 25-50%
- **Reduce infrastructure costs:** \$10M+ per large environment

Value Analysis: Ask Yourself...

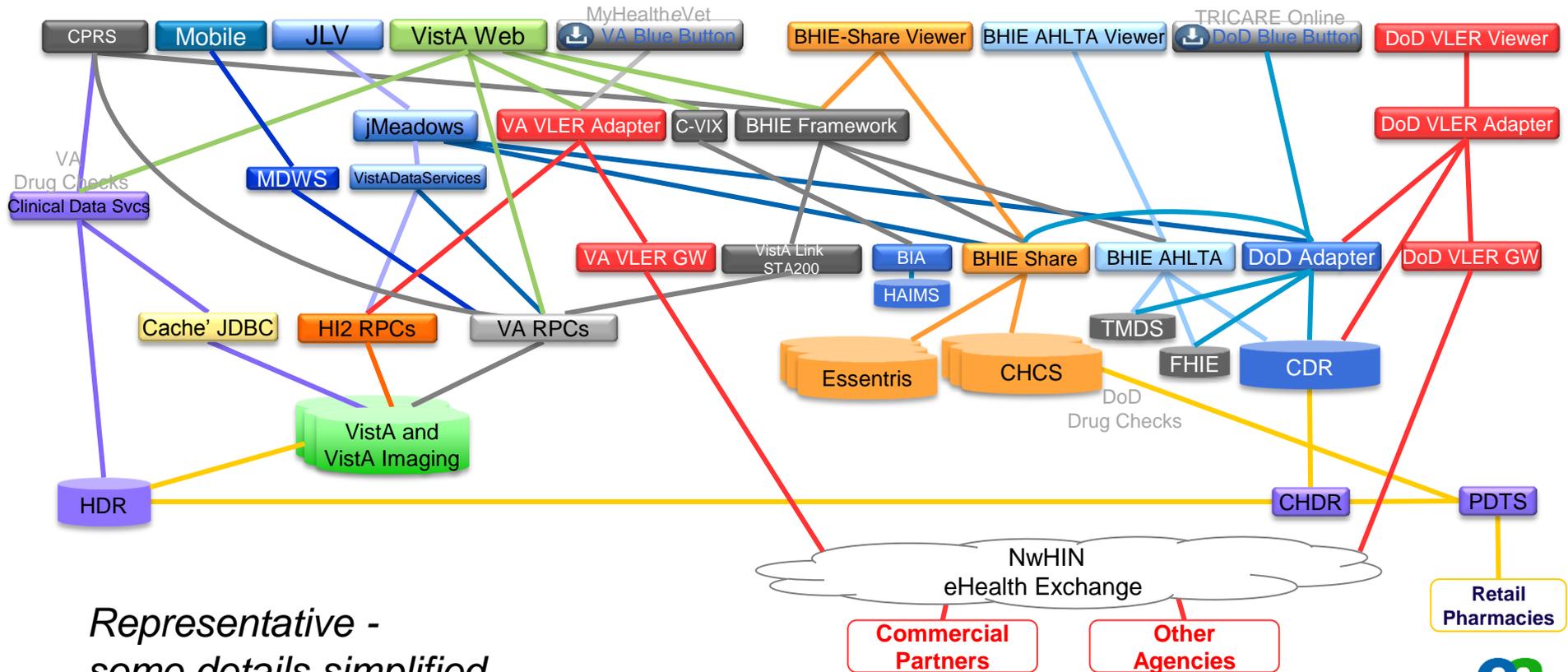
-  How many requirements are tested manually vs. automated? “90% manual”
-  What % of QA’s reported defects are rejected as “not reproducible“ by Dev? “At least 35%”
-  Time spent “proving” an issue that’s been rejected as not reproducible? “3-4 hours each”
-  % of resolved cases from Development re-opened by QA as still failing? “25% or more”
-  Avg. Dev & QA time recoding and retesting each unresolved issue? “4-5 hours avg.”
-  **The big one:** % of Total Test time in the next release spent retesting what you are testing in the current release? **“80%+”**

Typical Results: Cost Savings > 25% | Releases > 20% Faster

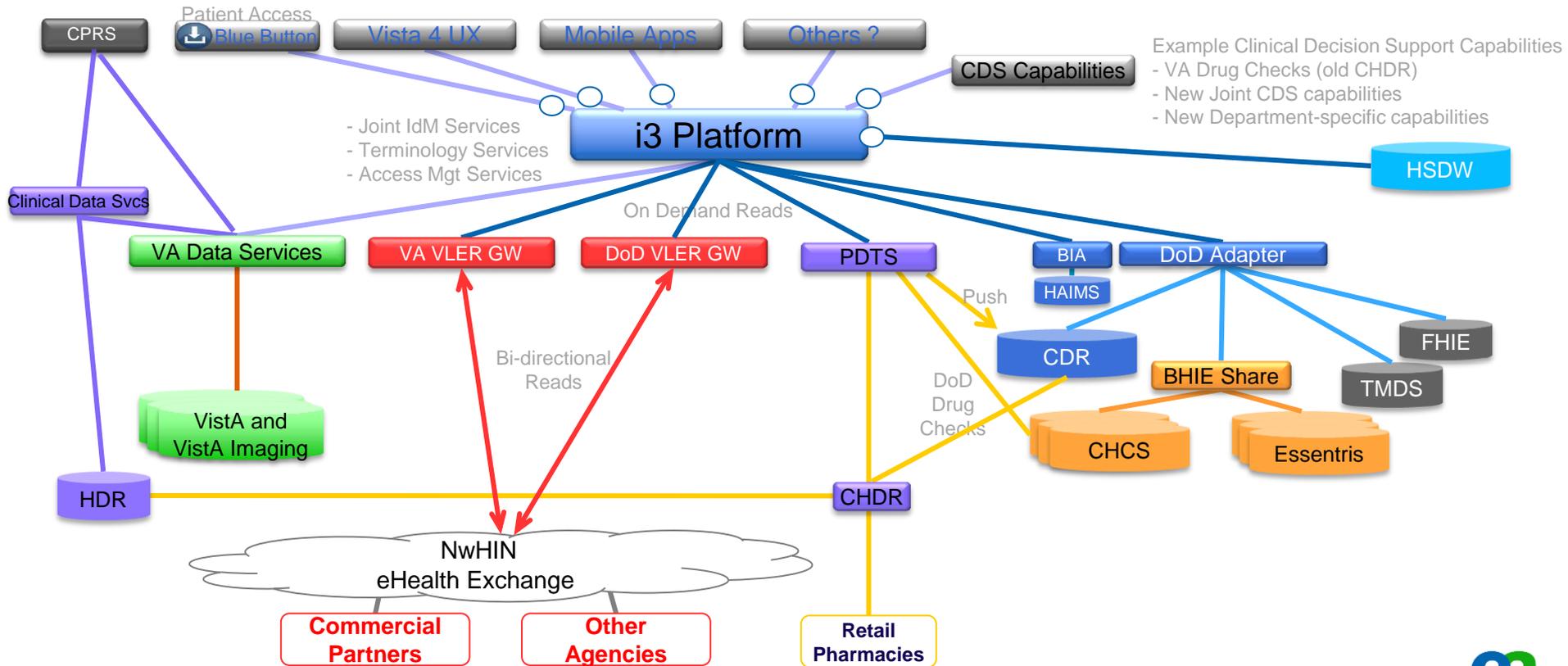
iEHR Informatics Integration and Interoperability Platform (i3P)



2013 Data Sharing Components & Interactions



End of CY2014

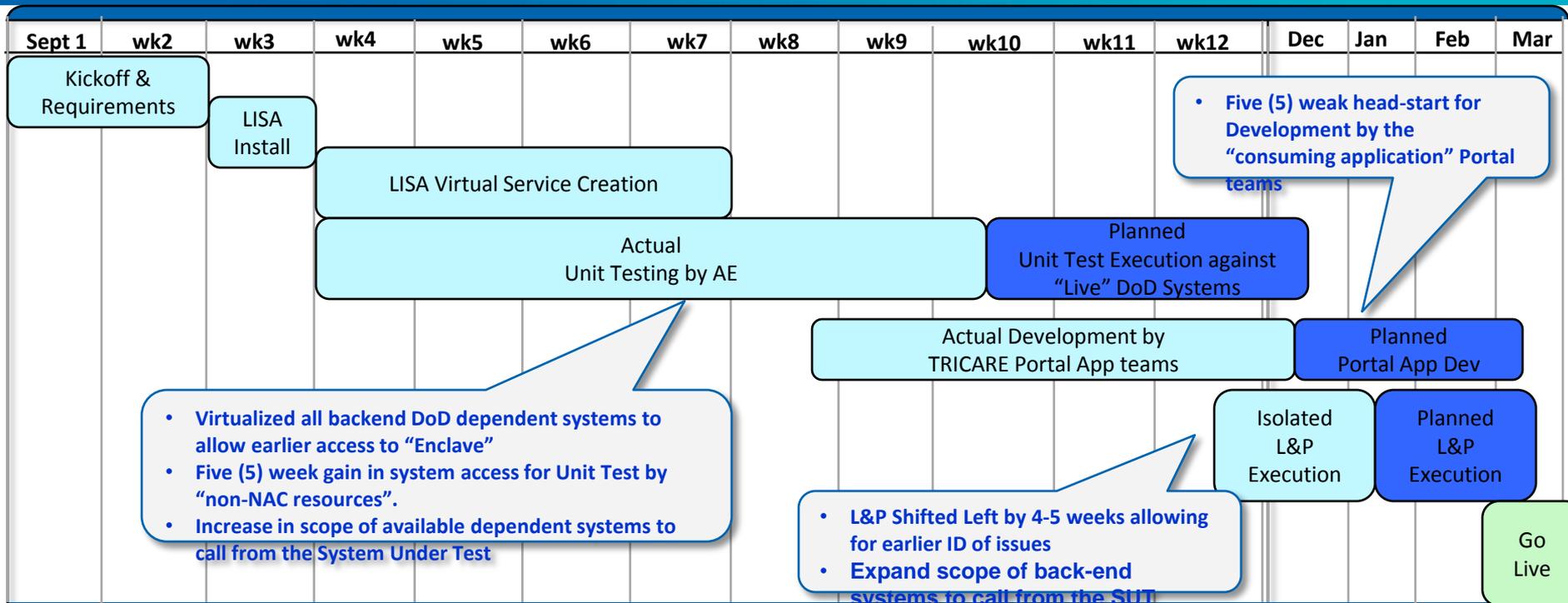


UHG: TRICARE Project Scorecard (November 2012)

Measure	Before	After	Assumptions	Improvement / Savings	Total
Efficiency					
Ability to start development before environment is ready	5 weeks delay	No dependency	Assume 25% gain in elimination of idle time for 40 developers @ \$50 / hour	5 weeks savings	\$100k
Resources to maintain data for development & testing	40k	10k	Assume 16 hours per week in live and 4 hours per week using LISA @ \$50 / hour	75% reduction in data setup time	\$30k
Downtime of live system impacts development	50k	10k	Assume 4 hours per week downtime impacting 20 developers causing 25% idle time and 0.5 hours weekly downtime with LISA @ \$50 / hour	80% savings due to reduced downtime	\$40k
Quality					
Improved quality due to better unit testing by developers using LISA		20 fewer bugs	Assume 20 fewer bug using LISA per developer per year at \$10k per bug	20 fewer bugs	\$200k
Improved quality due to better system testing by QA team		10 fewer bugs	Assume 10 fewer bugs escape from QA due to using LISA at \$40k per bug	10 fewer bugs	\$400k
Infrastructure & Resources					
Cost of pre-prod environment			LISA replaces multiple pre-prod servers, associated software costs, maintenance costs & administration costs	Est. 500k	\$500k
Cost avoidance of hiring NAC resources for development phase (assume 40 resources for 6 months)	\$75/hour 3MM / year	\$45/hour 1.8MM	Offshore resource do not have access to secure Tricare environment, but have access to virtual environment	1.2MM (40%)	\$1.2MM
Cost avoidance of hiring NAC resources for maintenance phase (assume 10 resources)	\$75/hour 1.5MM / year	\$45/hour 0.9MM	Offshore resource do not have access to secure Tricare environment, but have access to virtual environment	0.6MM (40%)	\$600k
Total Savings for TRICARE Project					\$3.07MM

UHG TRICARE Deployment Timeline- "Shift-Left"

LISA Value Release for TRICARE



- Virtualized all backend DoD dependent systems to allow earlier access to "Enclave"
- Five (5) week gain in system access for Unit Test by "non-NAC resources".
- Increase in scope of available dependent systems to call from the System Under Test

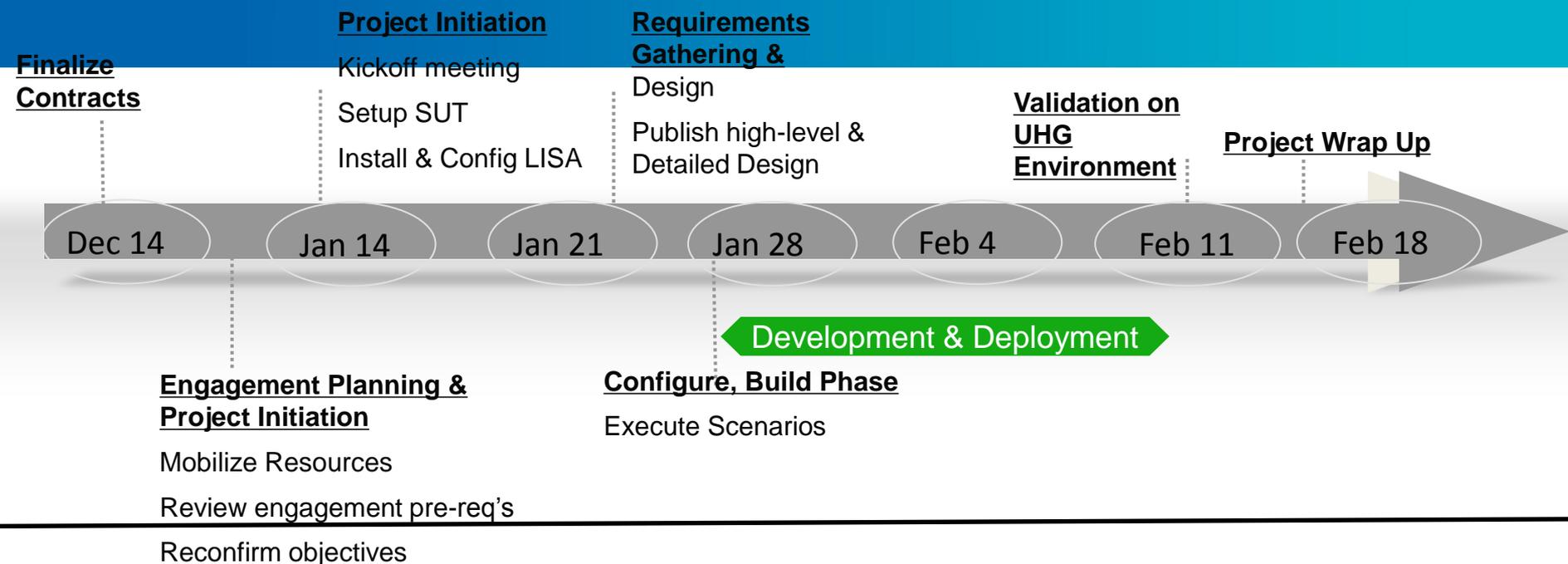
- Five (5) week head-start for Development by the "consuming application" Portal teams

- L&P Shifted Left by 4-5 weeks allowing for earlier ID of issues
- Expand scope of back-end systems to call from the SUT

16-Wk Value Release (including mentorship)



Test Harness Implementation Summary: Planned Approach for Delivery / Collaboration with UHG



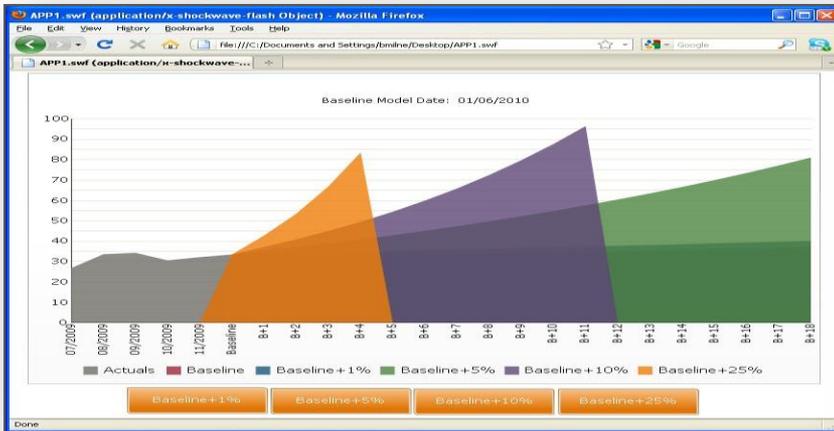
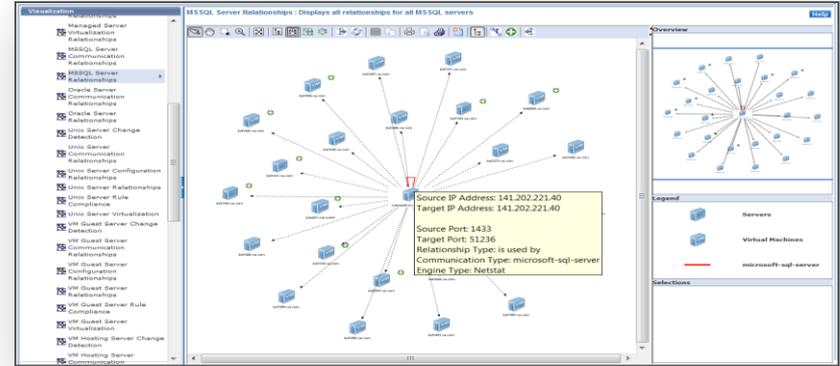
Benefits of early implementation of Service Virtualization:

- Service Virtualization environments will be accessible for Test Harness teams within 6 weeks of project initiation.
- Test Harness is the technology foundation for Optum Cloud and Environment Reduction efforts.
- CA Services will hand-off to UHG staff to run future Test Harness engagements as early as possible

Increasingly demanded application management capabilities



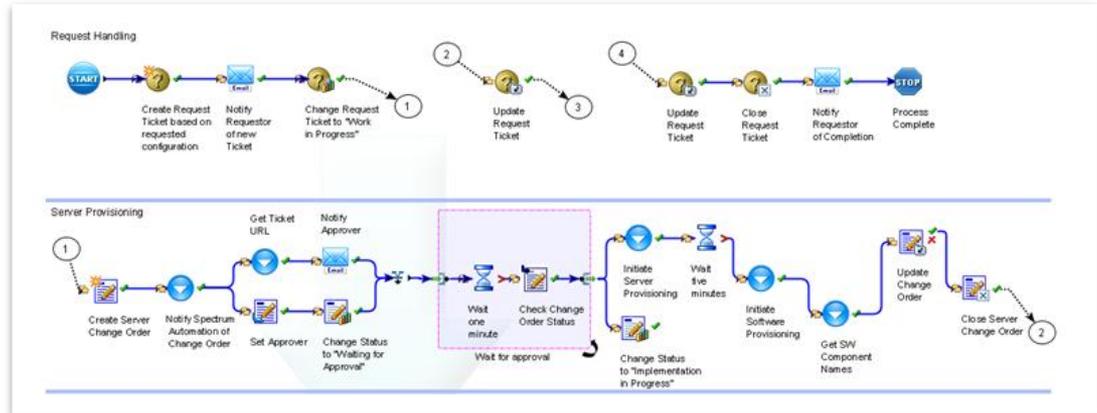
- Discovery & dependency mapping
- Capacity Planning
- Configuration tracking, mgt.



CA Process Automation

Service orchestration is at the center of most cloud reference architectures

- Design, engineer and automate processes
- Reliable, consistent, repeatable processes for visibility, quality, auditable control
- ITIL and other processes instantiated - incident, change, run-book, etc.
- Extensive content library, building a community

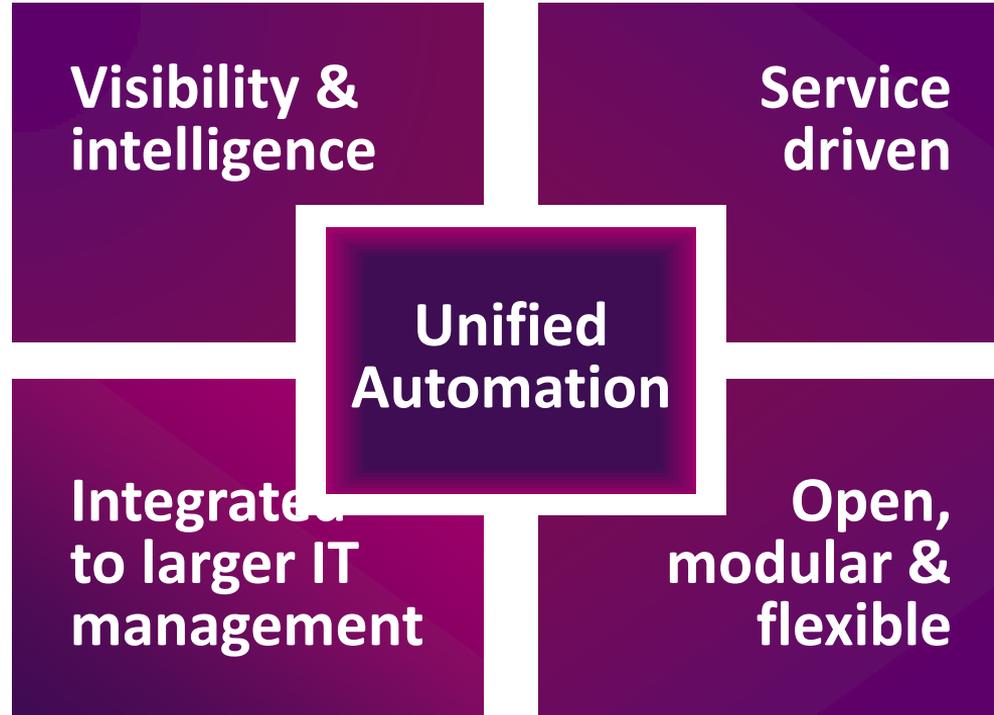


IT organizations with **IT Process Automation** are:



Source: EMA | *Operationalizing Cloud: The Move Towards a Cross-Domain Service Management Strategy* | February 2011

Tying it All Together - Key Success Factors



VA Enterprise Synergy

ITKOOLISA™
VA SEDR-approved for TRM

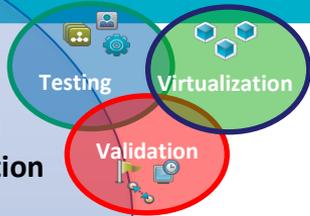
National Service Desk

Enterprise Testing

Metrics Monitoring
Trouble Ticketing Resolution
Service Operational Insight
National Visualization

Application Testing
SOA Testing
System Load Testing
User Testing
Application Validation

Automated & Distributed
Application Testing & Validation



Helpdesk & Service Support
Service Monitoring
Service Level Agreements
Technology Management
Change & Configuration Mgmt

Standardization
Common Language Normalization
Increased Customer Service
Enhanced ITSM
System Validation

Application Testing
SOA Testing
System Load Testing
User Testing
Thread Validation

IT-related Data Collection/Ingest
Data Automation/Analysis/Reporting
Business Intelligence Insight
Data Repository and CMDB
Virtualized Servers
Historical Reports Storage
Performance and De-duplication

Visualization
Connectivity
Accountability
Automated Testing
Infrastructure Compatibility

Compatible and Interoperable
Inter-Agency Information Exchange

EMF FDR

SOA ESB

Tagged Medical Asset Data Collection

Consolidated Help Desks

Consolidated Help Desks

IT-related Data Collection/Analysis

Asset Data Discovery/Aggregation

Storage Data Collection/Analysis

Performance Monitoring

VA Enclave
Joint Enclave

DoD Enclave

Best Practices for Faster IT Service Delivery

Key Challenges

- Many existing IT department processes, tasks and workflows are not representative of the needs of agile organizations.
- Life cycle or workflow synchronization across departments within IT organizations is often poorly performed and lacks clear ownership.
- Convergence points in workflows where information should be exchanged between departmental processes are not well-defined, resulting in delays in areas such as release management.
- Organizationally aligned release processes are often rigid, with little time spent optimizing associated processes and workflows.

Best Practices for DevOps for Faster IT Service Delivery

Recommendations

- Update departmental cycles to reflect the new requirements for velocity-oriented IT organizations.
- Coordinate cycles across departments to reduce process and workflow delays and information loss.
- Identify convergence areas that require multidepartment collaboration and technology integration.
- Perform continuous improvement of workflows to reduce waste and improve IT effectiveness

Notice

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. No unauthorized use, copying or distribution permitted.

Certain information in this publication may outline CA's general product direction. However, CA may make modifications to any CA product, software program, method or procedure described in this publication at any time without notice, and the development, release and timing of any features or functionality described in this publication remain at CA's sole discretion. CA will support only the referenced products in accordance with (i) the documentation and specifications provided with the referenced product, and (ii) CA's then-current maintenance and support policy for the referenced product. Notwithstanding anything in this publication to the contrary, this publication shall not: (i) constitute product documentation or specifications under any existing or future written license agreement or services agreement relating to any CA software product, or be subject to any warranty set forth in any such written agreement; (ii) serve to affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (iii) serve to amend any product documentation or specifications for any CA software product.

This document is for your informational purposes only and CA assumes no responsibility for the accuracy or completeness of the information contained herein. To the extent permitted by applicable law, CA provides this document "as is" without warranty of any kind, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or noninfringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, business interruption, goodwill or lost data, even if CA is expressly advised in advance of the possibility of such damages.

Any examples provided in this presentation are for illustrative purposes only and are not necessarily reflective of the results you can be expected to achieve.

Thank you

Steve.thomas@ca.com

(904) 806-4630

agility
made possible™

