Made to Stick by Dan and Chip Heath

Automating & Optimising Software Delivery during challenging economic times

David Brauneis (brauneis@us.ibm.com)
Chief Architect, Rational Automation Framework for WebSphere
Senior Architect, Rational Build Forge
IBM Rational Software, Software Delivery Automation

Rational. software

Agenda

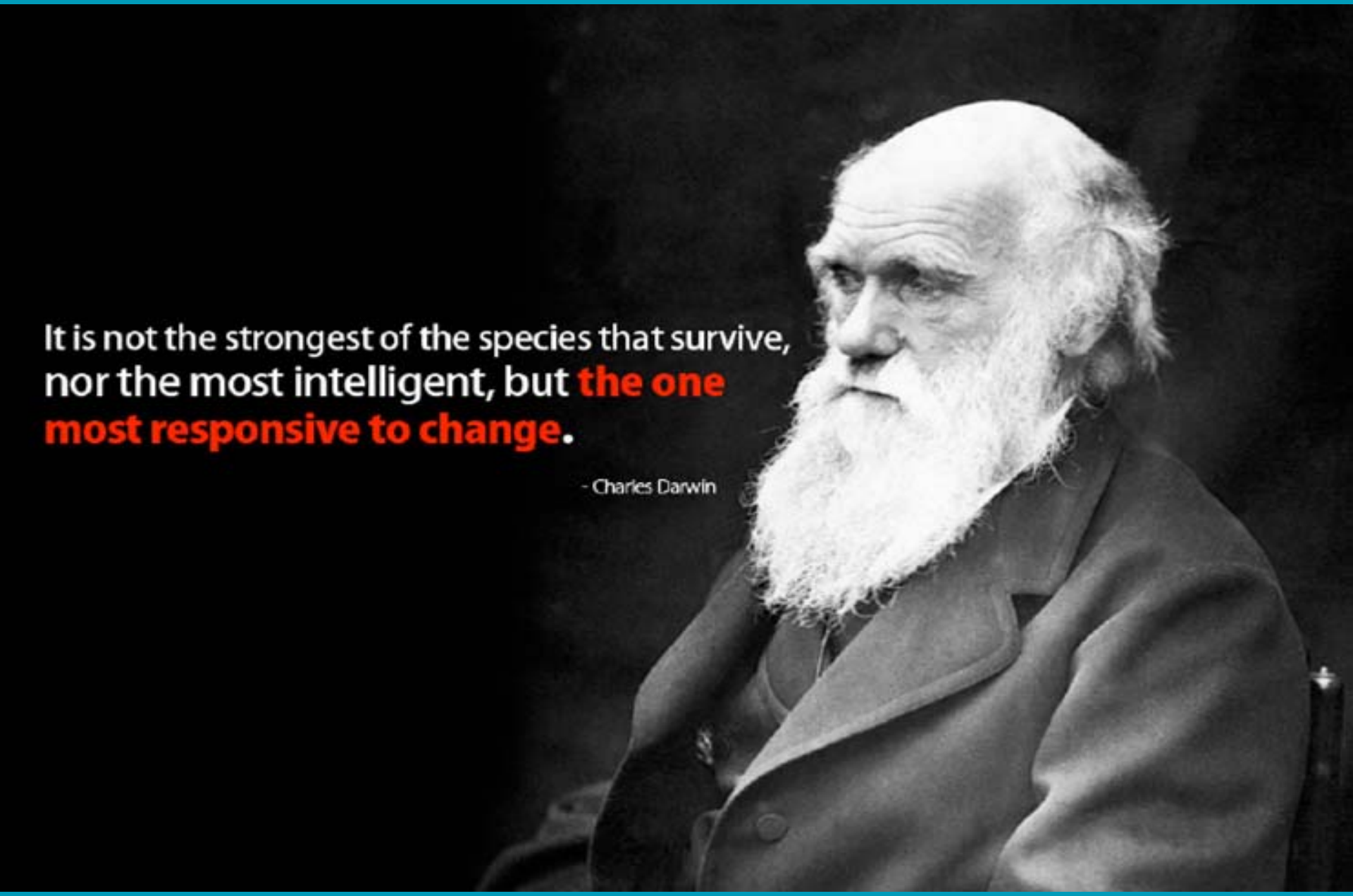
- Recession Factor
- Software Automation Concepts
- Software Automation Benefits
- Automation Examples
  - SDLC Automation and Consolidation
  - RTC and Build Forge
  - WAS / Portal Deployment s
  - Governance
- Wrap-up and Questions

Tough Times Ahead

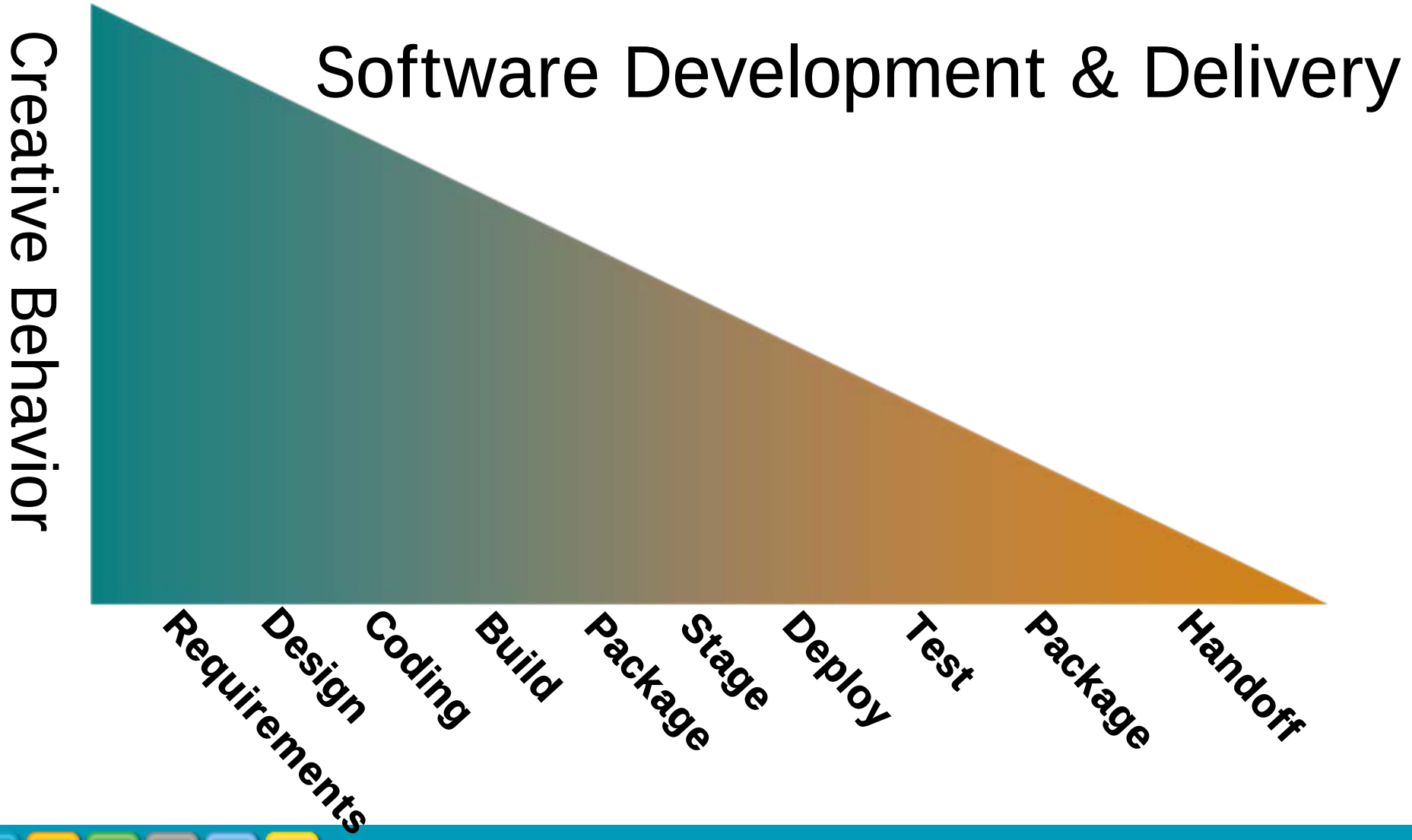
Tough Times Ahead

Sequoia Capital: "RIP: Good Times"

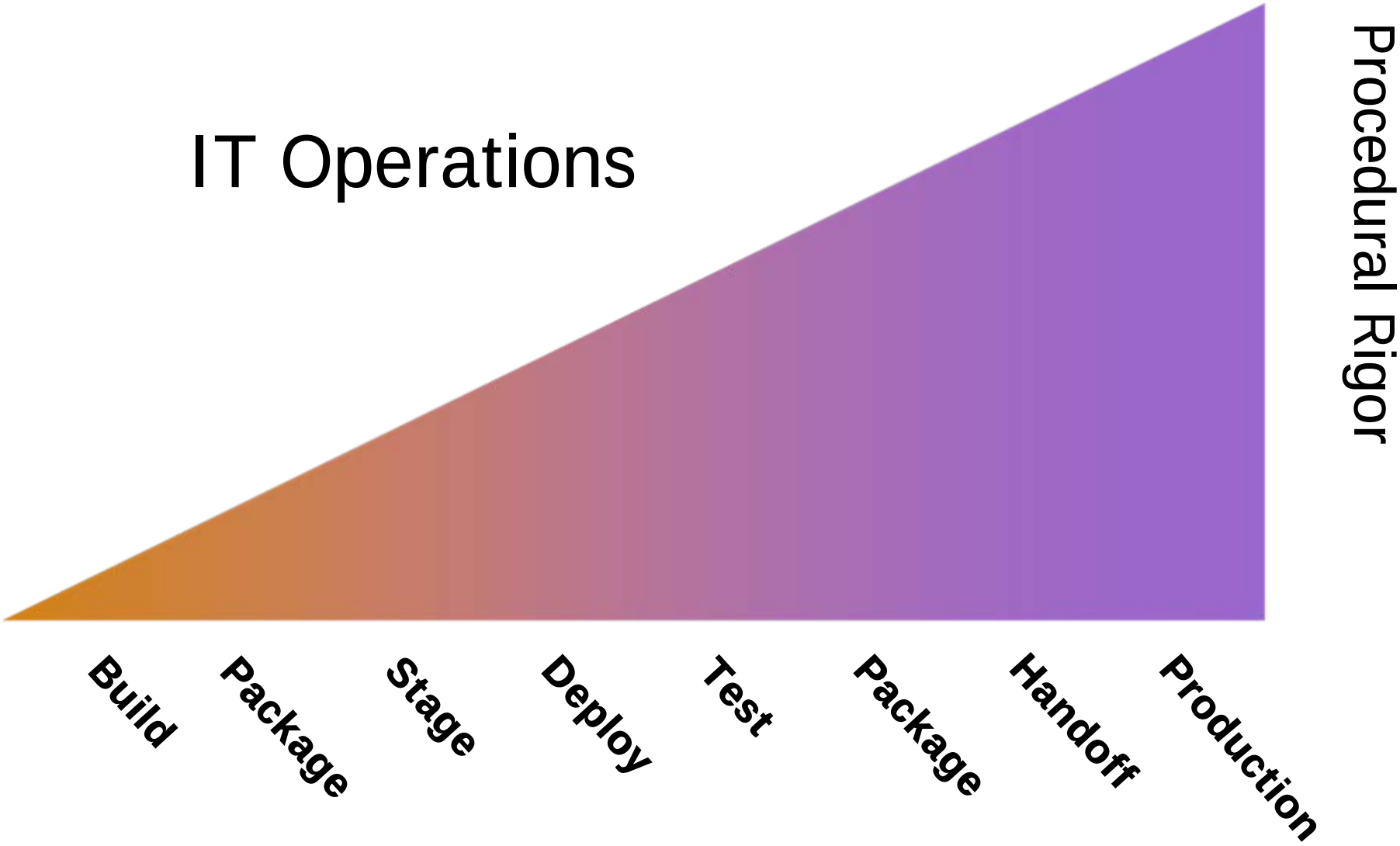
- Survival of Quickest: Cut Deep and Fast
- Engineering: Decrease Headcount for Next Version
- Product: What Features are Absolutely Necessary?
- Spend Every Dollar As If It Were Your Last
- Get Cash Flow Positive (Cash is King)
- Focus on Quality
- Lower Risk


It is not the strongest of the species that survive,
nor the most intelligent, but the one
most responsive to change.
- Charles Darwin

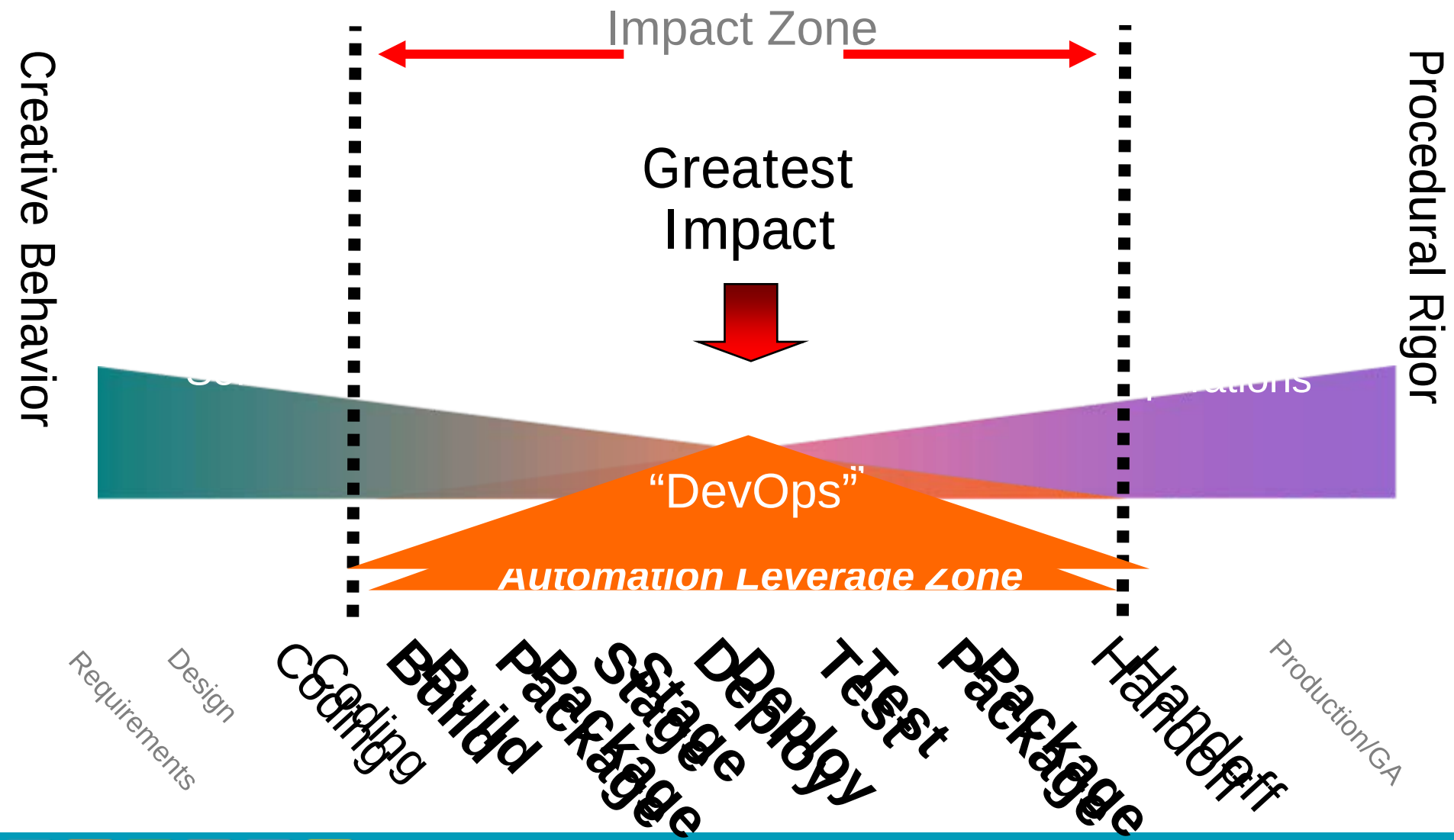
Two Worlds Collide



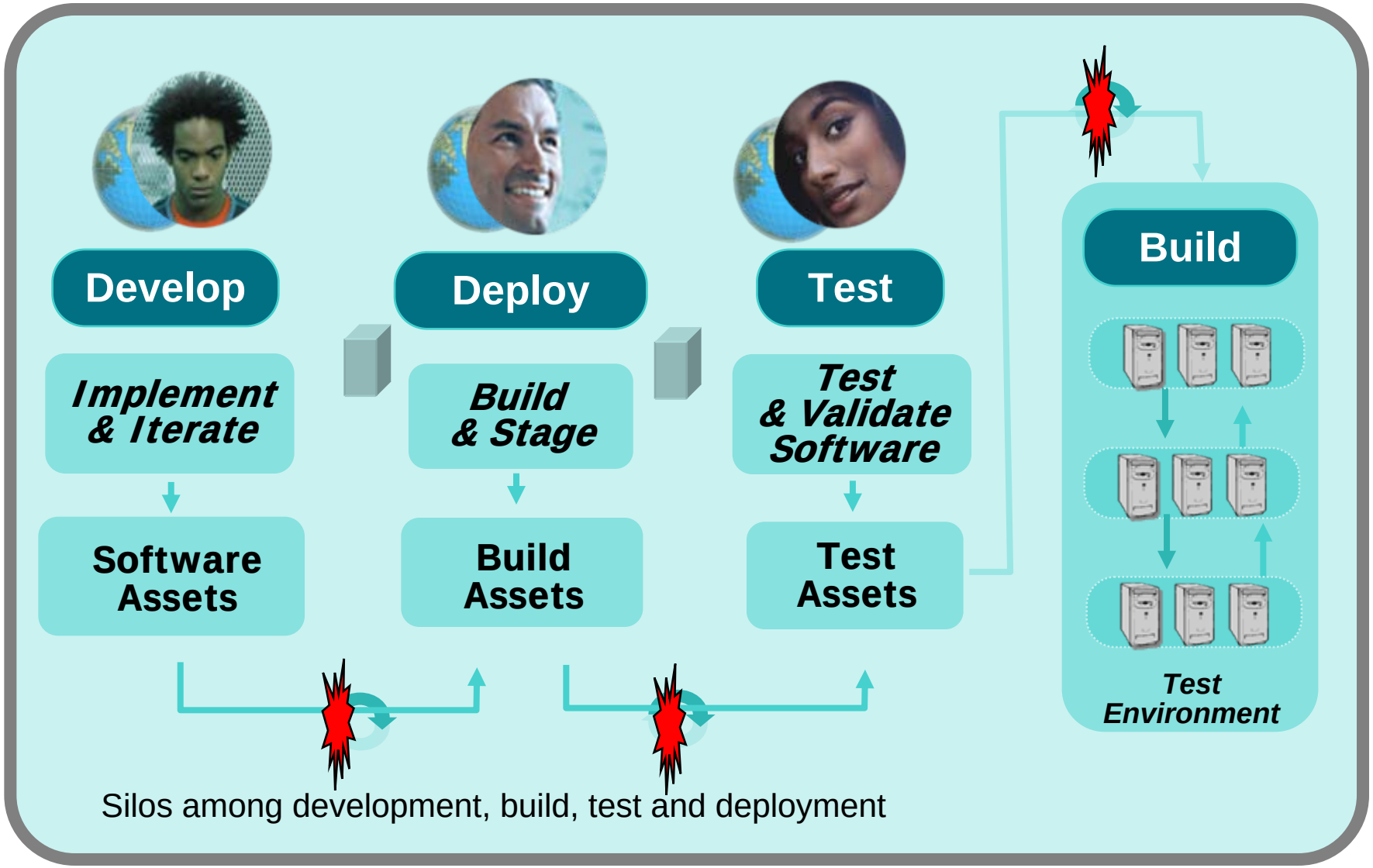
Two Worlds Collide



Two Worlds Collide: "The Clash"



Software Delivery Challenges in the Impact Zone

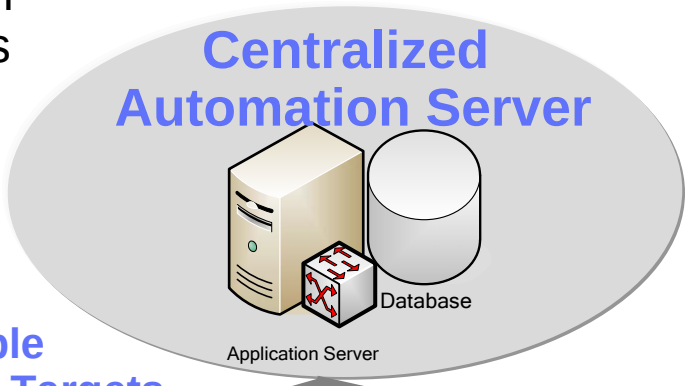


Software Delivery Challenges in the Impact Zone

- **Lack of control and visibility** due to multiple tools, methodologies and environments
- **Manual / Repetitive error-prone** tasks and processes
- **Proprietary, internally-developed** scripts and hand-off
- **Inconsistent processes** for different products & platforms
- **Separation environments: developer to** production systems
- Difficult and **time consuming to resolve problems**
- **Lack of audit trails** to satisfy **compliance** requirements

Centralized & Standardized Automation

Distributed automation system supporting variety of platforms **Windows, Linux, Unix, Mac, z/OS, i5/OS, zLinux, Tandem, proprietary….**



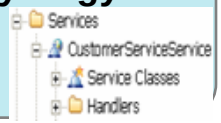
Example Automation Targets

White Box Testing / Code Scanning

RSAR AppScan DE Black Duck

Agent worker

ClearCase Synergy etc.



Agent worker

RAM

Agent worker

Build Automation

Build Forge

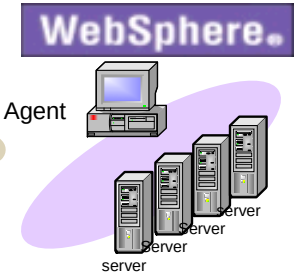
Agent

Agent worker

Rational AppScan

Agent worker

System z & i



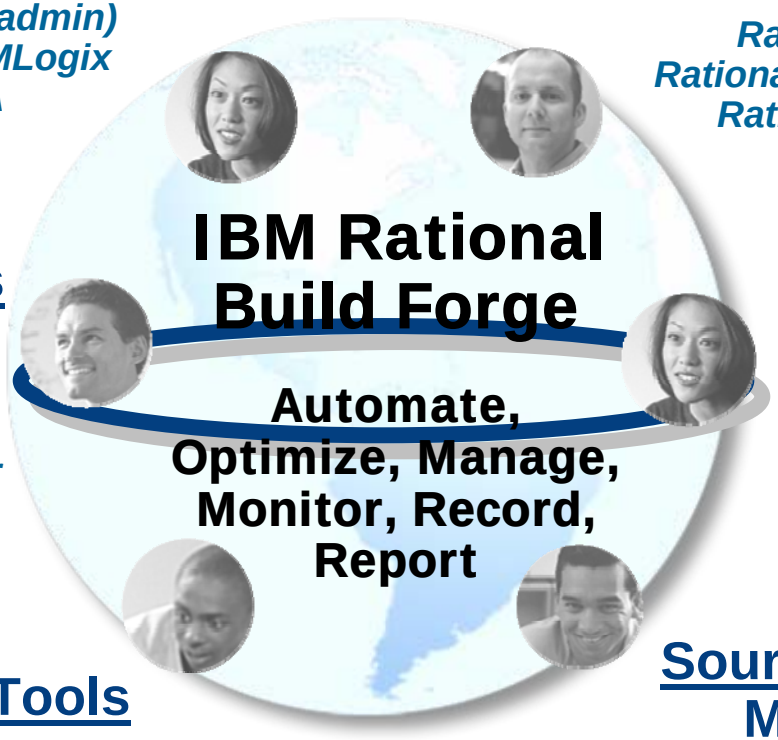
Build Forge Extensive Tool Integration

Deployment & Provisioning Tools

Tivoli Provisioning Manager
WebSphere Admin (wsadmin)
Phurnace, VMWare, VMLogix
Ascendant WICA

Development Tools

Rational Application Developer
Eclipse, VisualStudio
Rational Software Analyzer
Rational AppScan Developer Edition
Rational Team Concert (Jazz)
Black Duck



Software Quality Tools

Rational Test Lab Manager,
Rational Functional, Performance & Manual Tester, Mercury Quality Center, LoadRunner, TestDirector
WinRunner, Junit

Governance Tools

Rational Team Concert
Rational ClearQuest
WebLayers
Rational Asset Manager

Compile/Assemble Tools

Compilers (any)
Linkers (all)
Ant, Make, NMake
Maven, Rational ClearMake
Jar, rpm, zip,

Source Control & Change Management Tools

Rational ClearCase,
Rational ClearQuest
StarTeam, Perforce, CVS,
PVCS, VSS, Synergy
Subversion

Key Aspect: Optimizing Software Development Teams

- **Automation**
  - Eliminate manual activities in all stages of software processing
- **Consolidation**
  - Centralized visibility and control
- **Integration**
  - Link together individual tools used across the entire lifecycle
- **Virtualization**
  - Reduce physical compute resources required for every task
- **Quality Improvement**
  - Eliminate manual / repetitive tasks & give more cycles for testing
- **Cost Reduction**
  - Optimize processes and remove repetitive/manual steps
- **Governance**
  - Institutionalize compliance requirements through automation

Business Benefits of Automation….

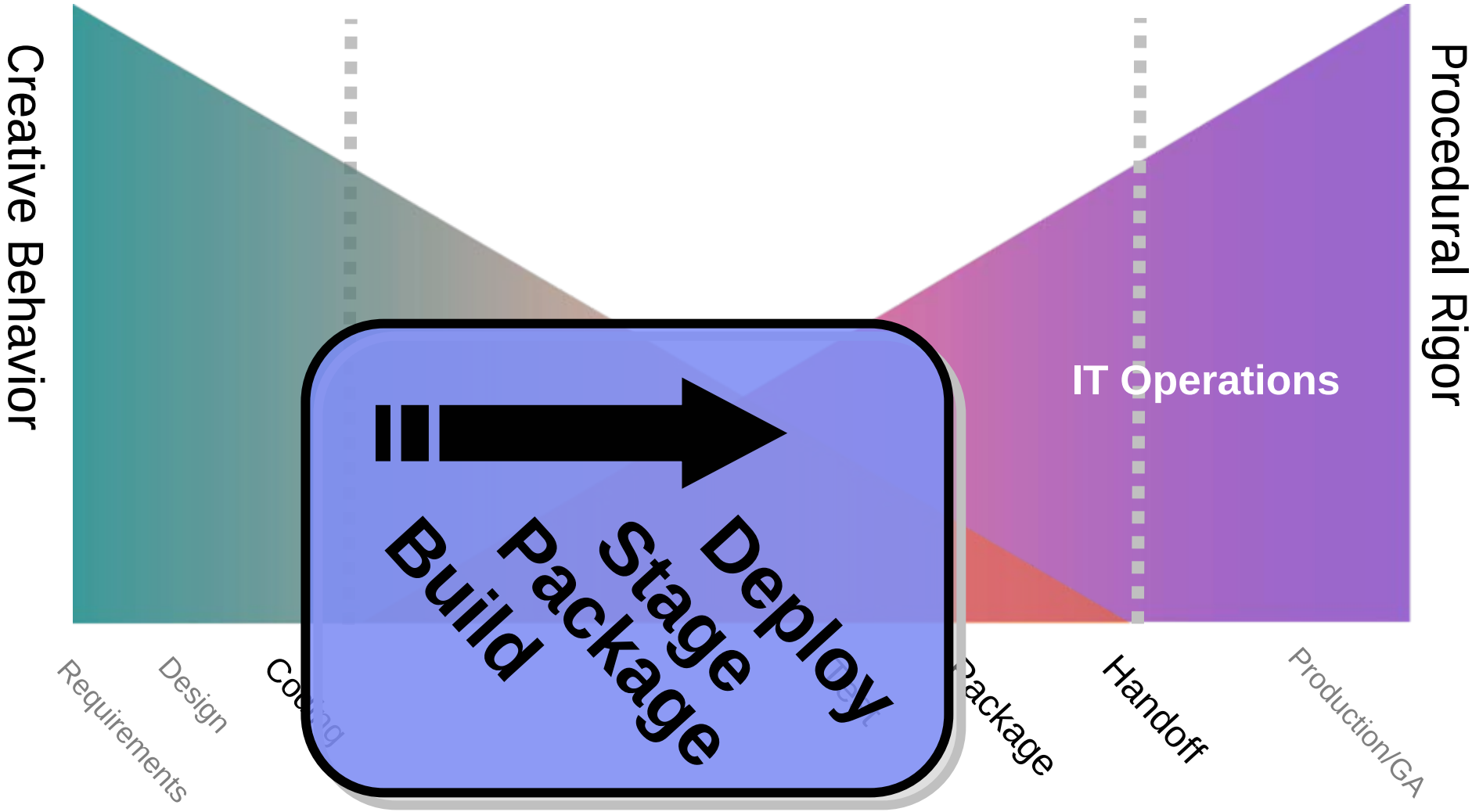
Benefit	Average Improvement	Highest Improvement
Reduction in Costs	**25%**	**50-70%**
Improved Quality	**30-40%**	**70-80%**
Return on Investment (ROI) ▪ 3 to 9 months ROI	**80%**	**Over 200%**
Increased Productivity ▪ Speed of builds/releases ▪ Configuration Management ▪ Developer Productivity	**110%** **42%** **28%**	**500-2000%** **90% or greater** **81-90%**

Source: The Evolution of Build and Release Management for Effective Software Delivery: *A Customer Survey with Case Studies*
By: Hurwitz & Associates – October 2007

Development / Delivery Team Benefits

Team	Benefits
Development	▪ "10% improvement from coordinating with deployment and testing teams."
Configuration Mgmt. / SCM	▪ 30% improvement from automated scheduling ▪ 50% improved Response times to dev / test / etc
Quality Team	▪ "15% improvement due to testing better quality code"
Deployment	▪ "40% improvement automation & real time alerts ▪ More cross functional, removed guru bottlenecks ▪ Less Time satisfying audits
Managers	▪ "50% improvement in audit time ▪ Information more accessible and transparent

Rational Software: SDLC Example



Rational. software

"Build Forge helped us improve our turnaround times, quality and overall process by giving us a continuous integration system that allows us to notify developers of project status"

Before

- 47 active projects /3 locations
- 8 platforms /124 build machines
- Windows, All Unix Flavors

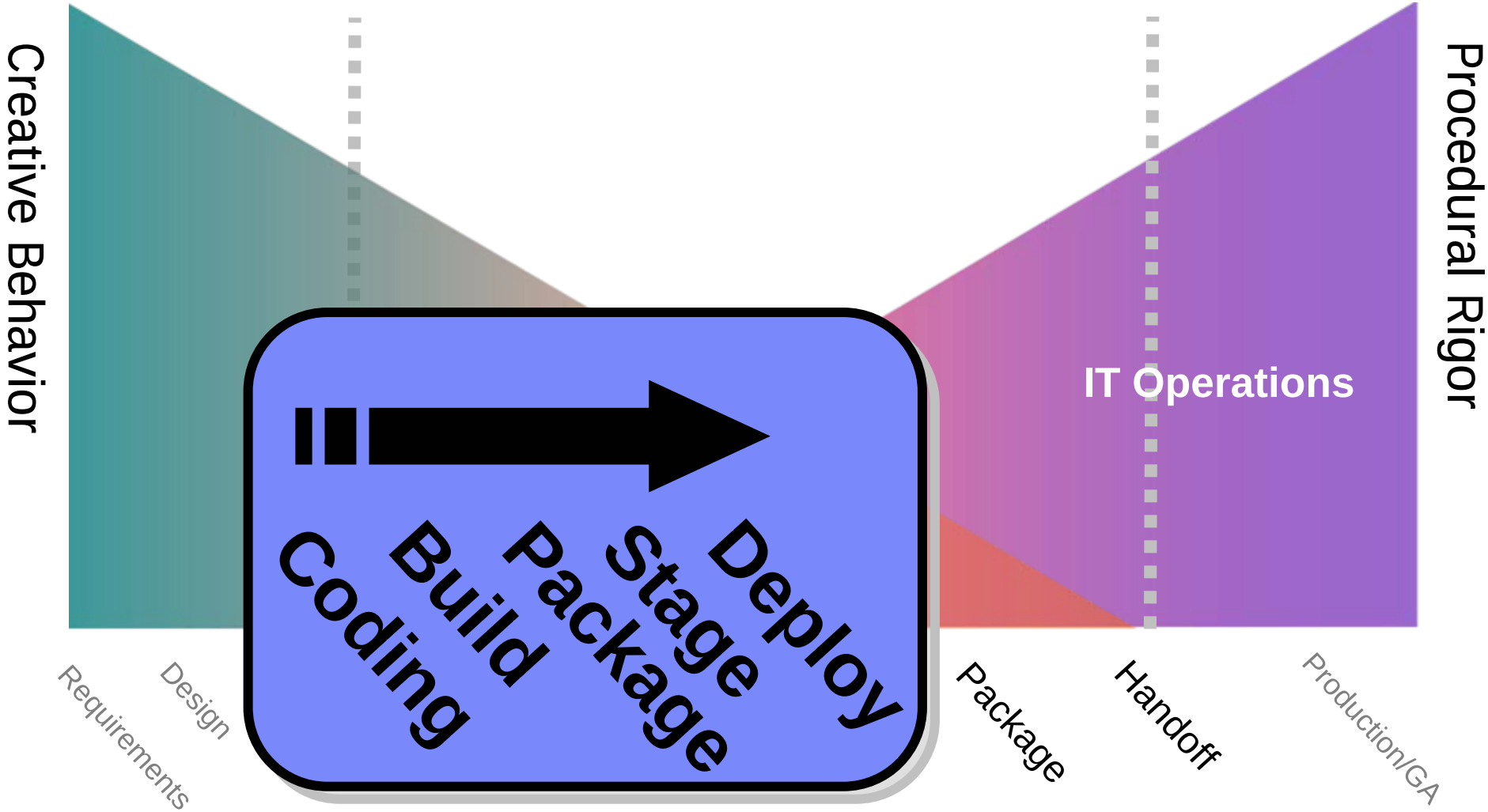
~

- 24hr "Suite" build & 14hr product build
- Release Team bottleneck. No developer self service
- Serial / manual work effort

After

- Fully Automated Build Orchestration & Product Image Creation
- Implemented Developer self-service in 3 months
- "Suite" AND point product builds reduced to 3 hours up to 800% improvement
- Automated packaging 85+% efficiency gain

US Airline Co. : Enterprise SCM Consolidation Ex.



US Airline Co

"Build Forge allows us to standardize our software development process across the organization and do more with fewer resources."

Before

- ClearCase, ClearQuest, Build Forge

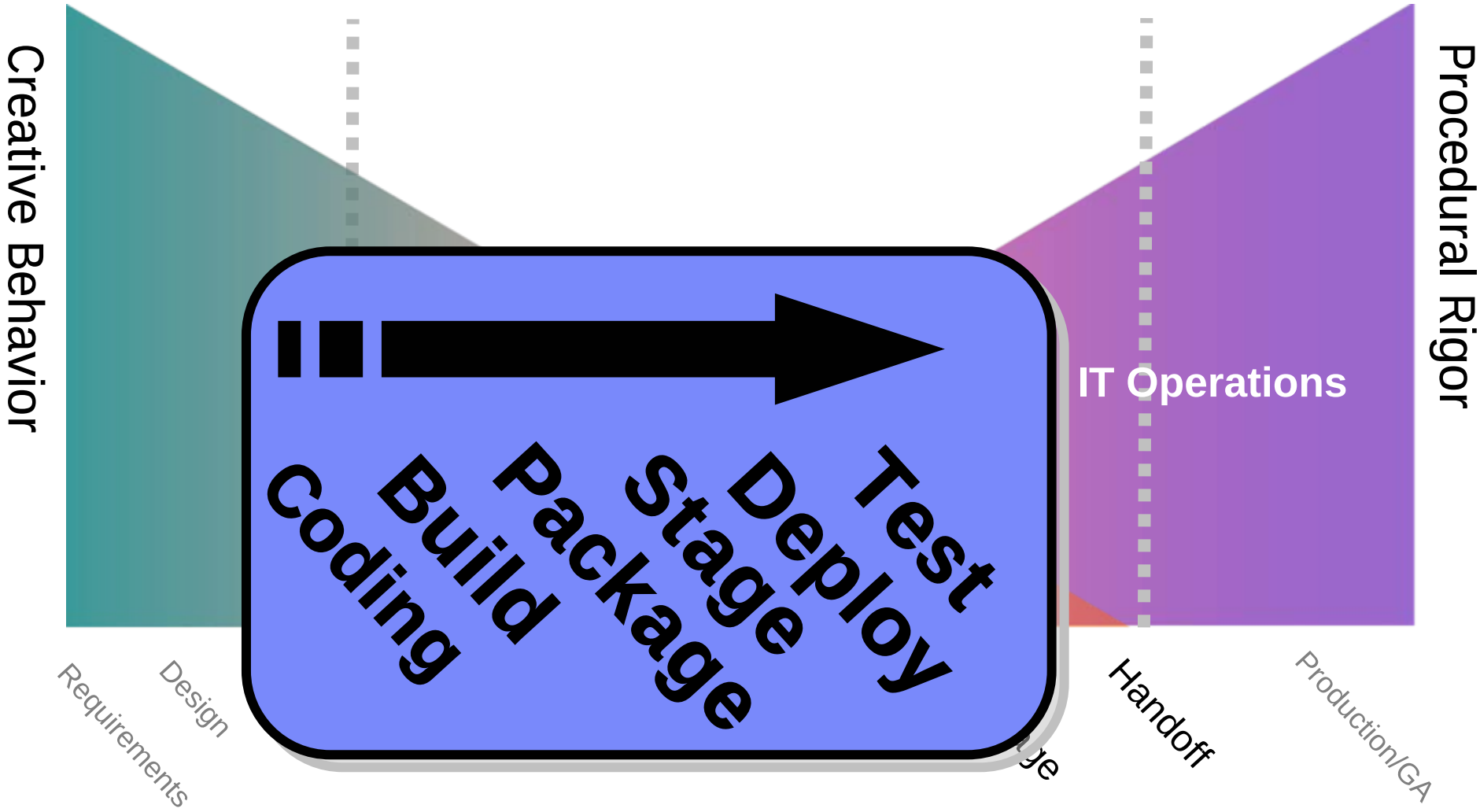
~

- Centralized team formed to consolidate and standardize build practices across for 150+ applications.
- Processes were varied & manual
- Goal to support Continuous Integration and Developer Self Service for Builds

After

- **Consolidated** builds into a single console for entire org.
- Converted projects with **minimal impact** and w/out extensive training.
- **Optimized** build steps across a server farm
- **Improved Predictability** via automation and continuous integration

Financial Company: SDLC Enterprise Example



Financial Co.

Centralized software delivery mgmt while still providing flexibility for integrating front and back office

Before

- 2000+ developers
- 100's of projects with varied processes.
- Team Concert, ClearCase, ClearQuest, ReqPro and Build Forge

~

- Ad hoc processes, chaos
- No Visibility
- No Reproducibility

After

Faster throughput on builds, testing and deployment.

Improved quality due to catching defects earlier.

Better ability to maintain and deploy 3rd party projects

Projects choice (CC/CQ or RTC) keeping teams agile and efficient.

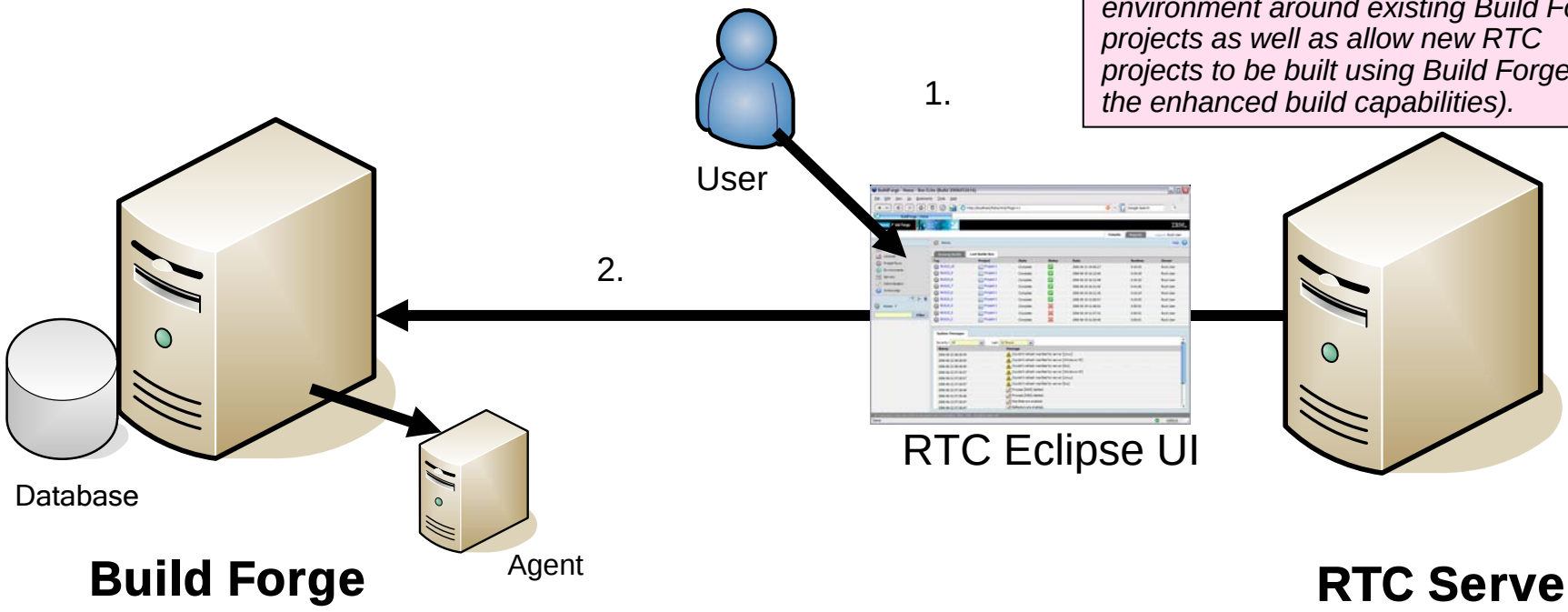
Background **compliance**

Build Forge / RTC Integration Use Case

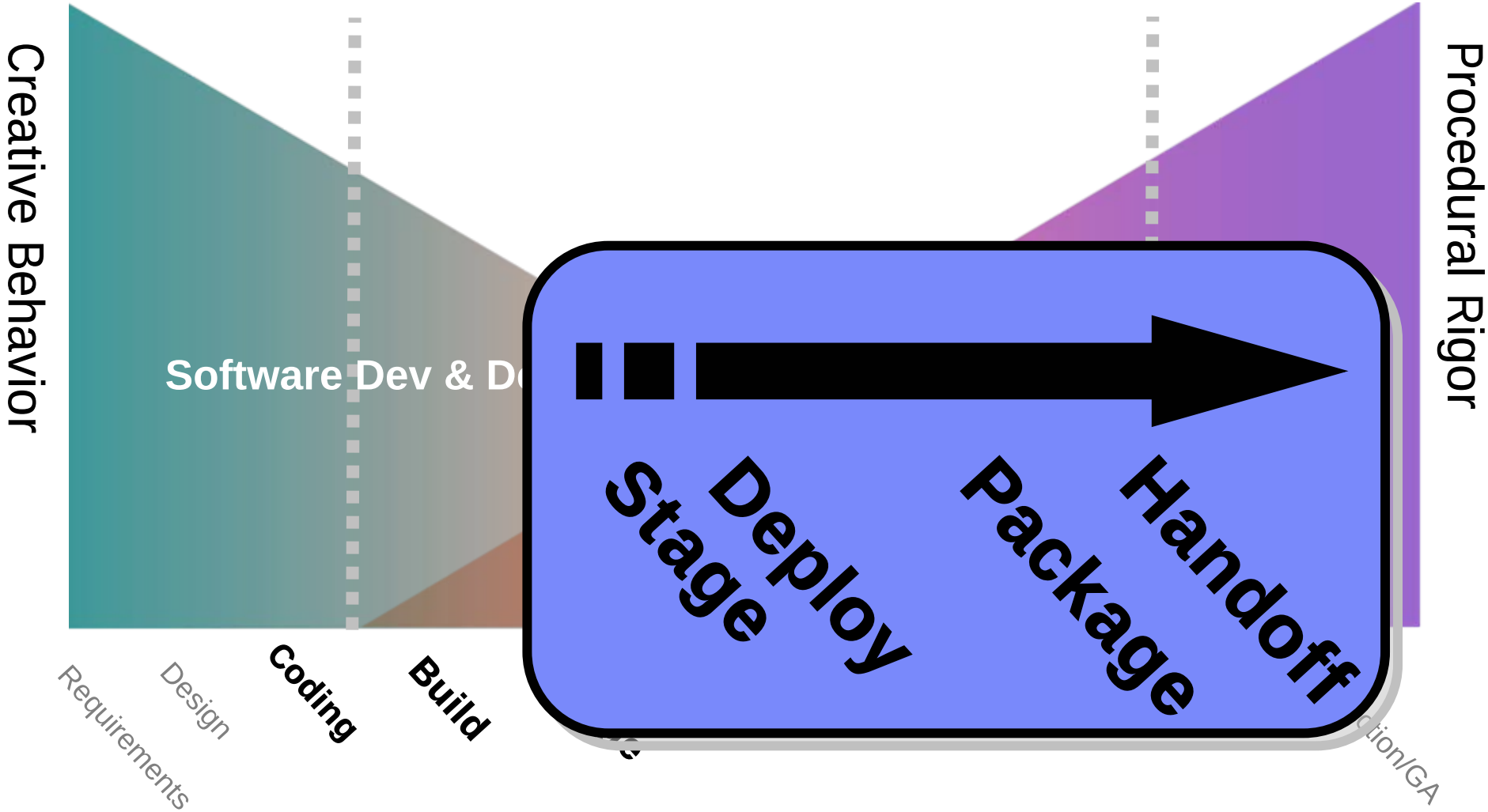
Build Forge configured as build provider for RTC

1. User interacts with RTC Eclipse UI to request a build which will be serviced by Build Forge
2. Build Forge executes the build (including multi-platform, multi-step, and threaded)
3. Results appear in the RTC Eclipse UI (in addition to the normal Build Forge web UI)

Allow RTC users to have the collaborative environment around existing Build Forge projects as well as allow new RTC projects to be built using Build Forge (with the enhanced build capabilities).



Networking Company: Build/ Deploy Example



Networking Co.

Reduced time of standing up a WS Portal cell from 2 weeks to 1 day with no extra resources.

Before

- **Centralized team supporting Websphere Portal for many lines of business**

~

- **Standing up each Portal cell consisted of 100's of manual, error prone steps.**
- **2-4 weeks of effort for ea.**
- **Missed delivery schedules**
- **Backlog of additional requests**

After

- 52 man hr weeks to 20. Future requests from 138wks to 38wks.
- **Savings of 400k due to automated install and config. 1.2M in the future.**
- **Reduced time diagnosing environment issues**
- **Higher customer satisfaction and greater predictability**

The Need: Common Challenges in WebSphere Environments

- Lack of consistency and/or repeatability
  - Staff taxed by environment synchronization efforts
  - Configuration changes hard to manage
  - Challenge to connect disparate groups under single, enforceable process
- No ability to manage WebSphere environments beyond the cell level
  - Most WebSphere admin performed today using home-grown solutions.
    - Cost-center and a burden to the business.
- Costly automation of configuration changes and deployments
  - Requires custom coding
  - Time consuming to deploy without a framework
- No change control or audit trail for WebSphere administration

What Is the Rational Automation Framework for WebSphere?

- Customizable Framework for the WebSphere Family of products that delivers
  - WebSphere product installation & patching automation
  - Configuration change management
  - Application deployment automation

The framework's strengths are…

- **Accuracy** → "Data Driven" - RAFW applies the right data to the right environment
- **Speed** → Eliminate manual administration with pre-built automation
- **Consistency** → Apply data in repeatable manner to WebSphere environments

Rational Build Forge – Core of Rational Automation Framework

End-to-end Software Delivery Automation

Cut the Cost of Software Development

- Rational brings industrial production techniques to your development lifecycle that reduce time-to-market and deliver higher quality, consistent results

Automate for Agility

- Automate your existing tools and processes, gain rapid ROI, and then fine-tune your cloud development for increased efficiency and savings

Improve Control in WebSphere Environments

- Over 450 field-proven automated tasks for configuration and application deployment to Application Server and Portal Server targets

Head for the Clouds for Greater Resource Utilization

- Rational Automation Framework for WebSphere integration with WebSphere CloudBurst delivers on-demand, easily customized middleware appliances for all purposes

Simple Scenario of Steps…

Total Steps = 150+

1. Perform pre-installation tasks (5 steps)
2. Create, configure, and verify deployment manager profile (20 steps)
3. Create, configure, and verify application server profile (12 steps)
4. Create, configure, and verify custom profile (10 steps)
5. Federate nodes (both app server and custom profiles) (14 steps)
6. Install, configure, and verify IBM HTTP server (14 steps)
7. Install the distributed remote plug-in (20 steps)
8. Create and configure the horizontal cluster (High Availability) (17 steps)
9. Enable and configure HA persistent service (9 steps)
10. Configure HTTP session persistence (41 steps)
    1. Memory-to-memory (20 steps)
    2. Database (21 steps)
11. Create and configure SIBus and messaging engine (5 steps)

What that looks like in RAFW

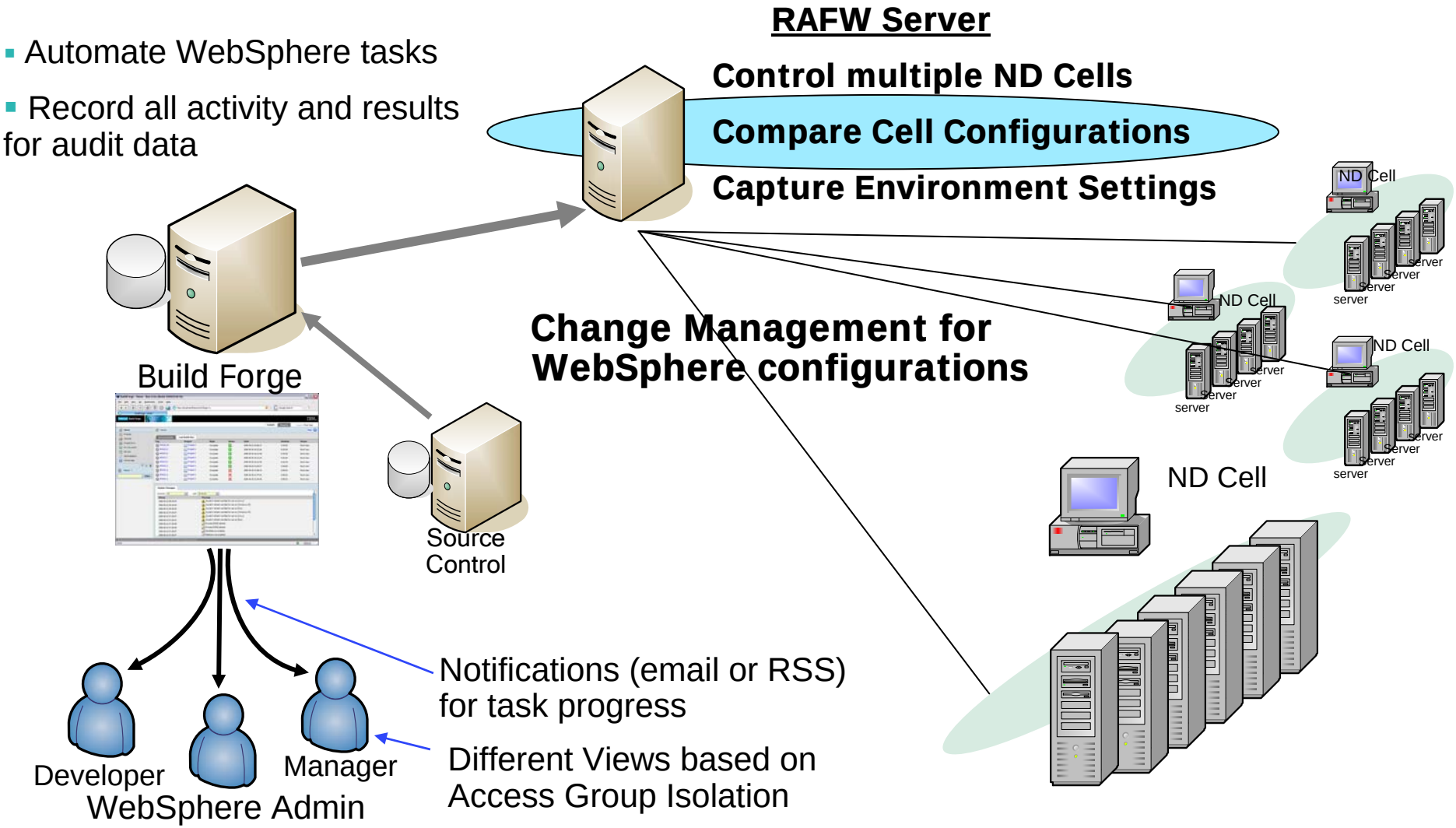
Complete WebSphere Cell Build out!

Build Out New WebSphere Clustered Environment

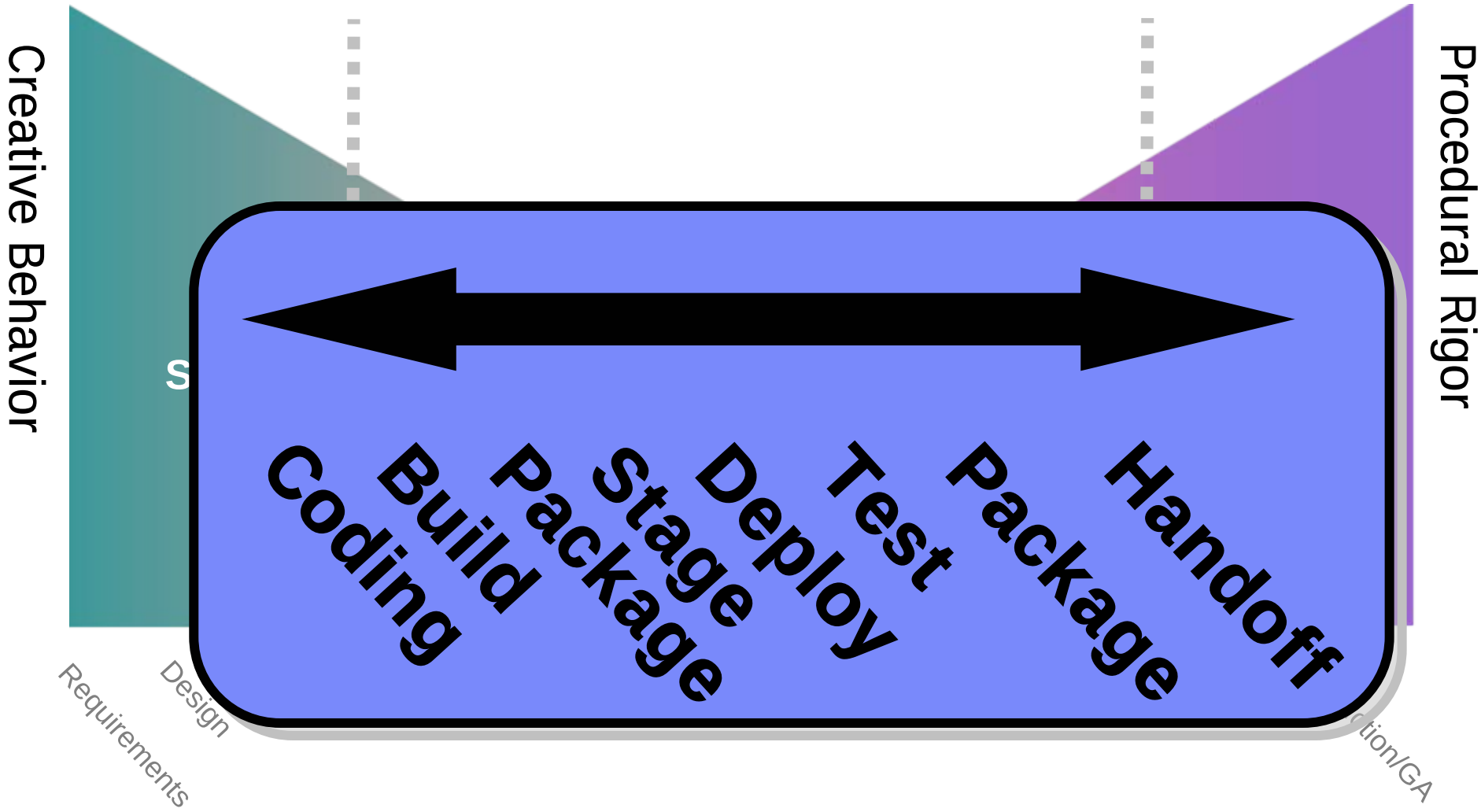
1. Generate new Environment using Framework Wizard
2. Click on New Project to launch build of new Environment
3. Automatically Notify interested parties upon completion
4. Add steps in project for Configuration elements (JDBC, JMS, JAAS etc)
5. Schedule unattended jobs
6. Rebuild environment as Needed!

Rational Automation Framework for WebSphere

- Automate WebSphere tasks
- Record all activity and results for audit data



Insurance Co. : Enterprise Governance Example



Insurance Co.

“ Build Forge has allowed us to formalize our process and put in place controls to enforce the process through automation that has also streamlined our entire application delivery environment”

Before

- 100’s of developers
- Outsourced development
- Windows, Linux, All Unix

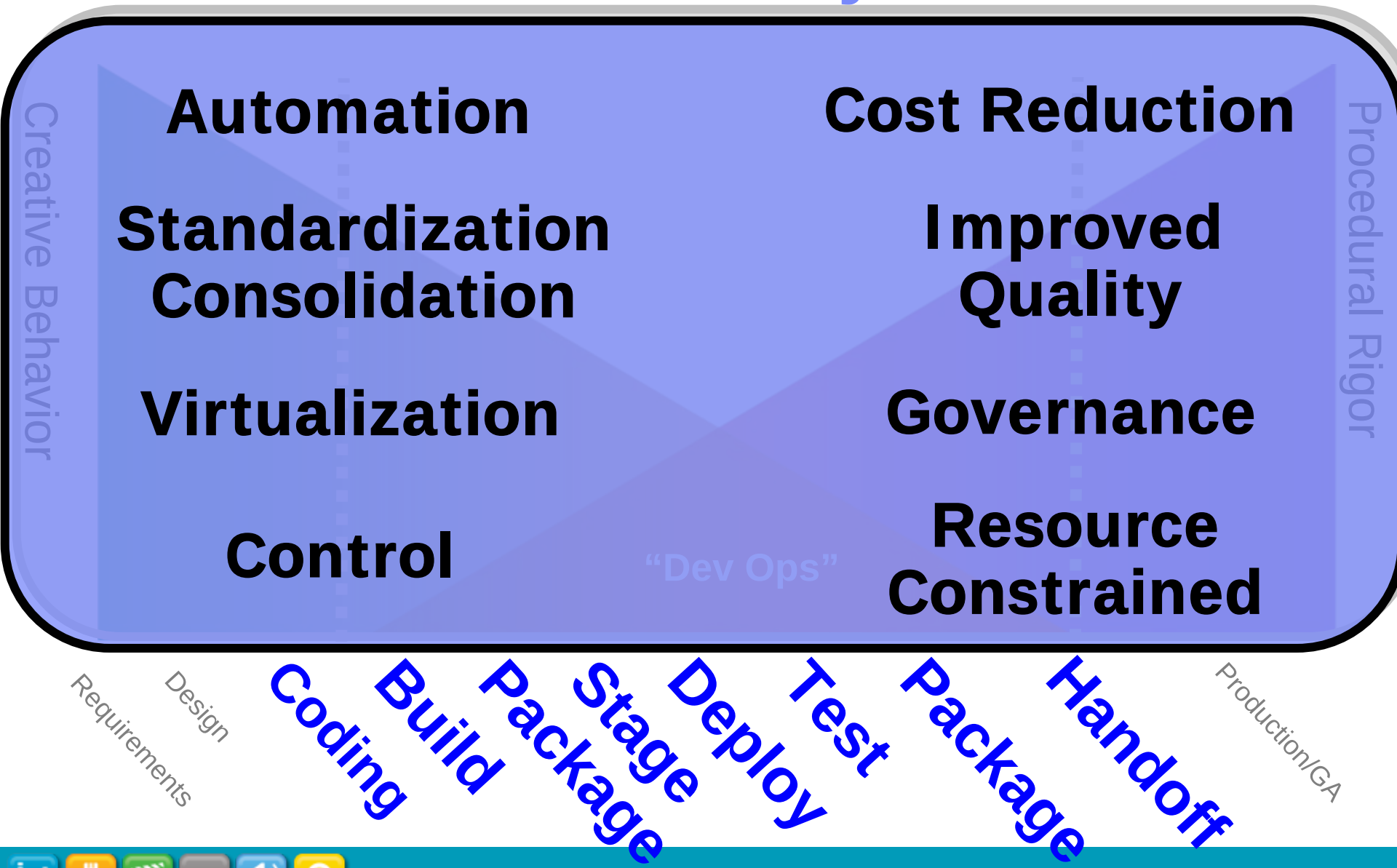
~

- Failed Internal Audit in prep for SoX audit
- Took days to find errors
- No reproducibility or production application
- Inability to document process

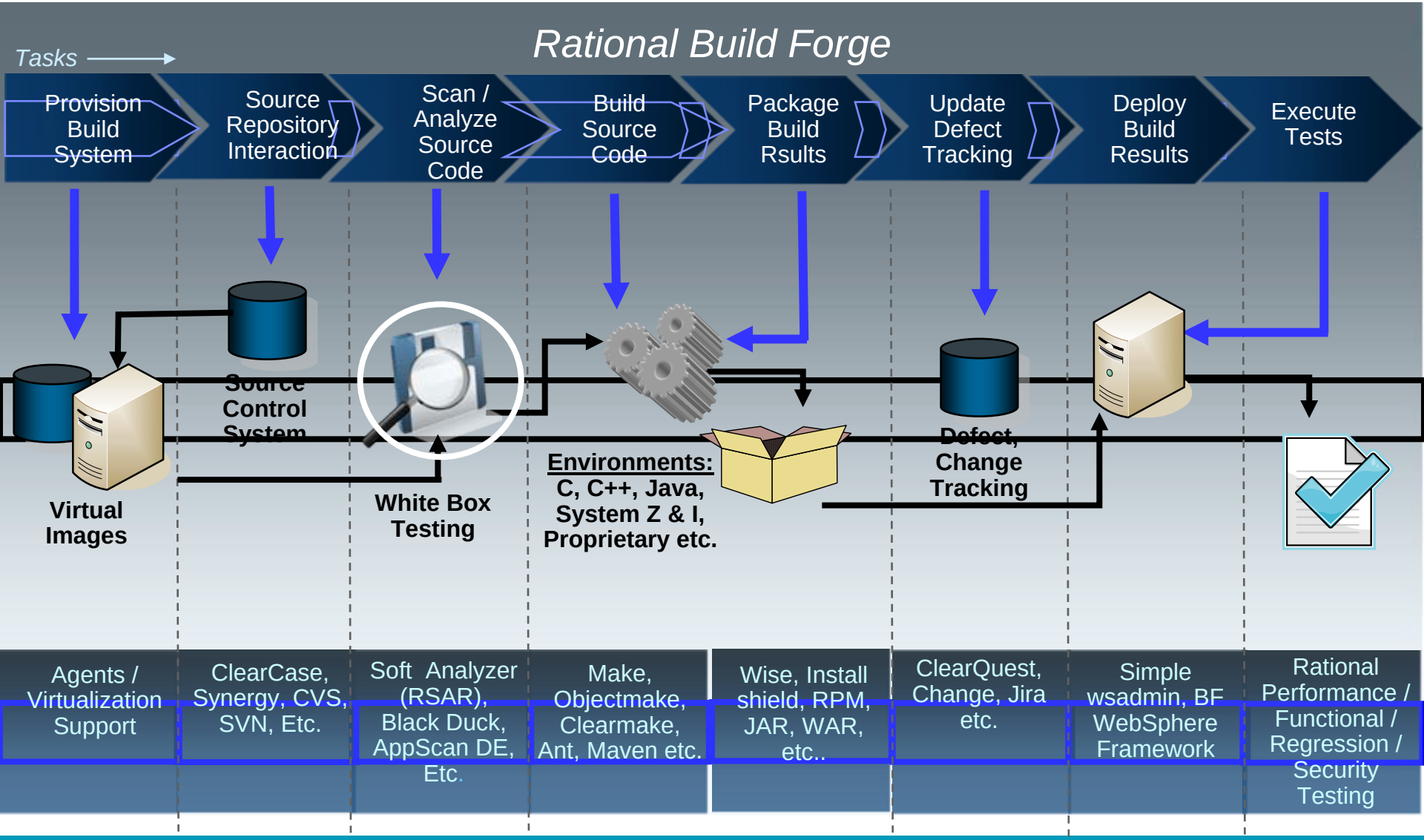
After

- Repeatable processes and deployments.
- Dramatic cost savings through improved speed and consistency. 3 Month ROI
- Self Documenting processes for audits.

Summary


Automation
Cost Reduction
Standardization
Consolidation
Improved
Quality
Virtualization
Governance
Control
Resource
Constrained
Creative Behavior
Procedural Rigor
"Dev Ops"
Requirements
Design
Coding
Build
Package
Stage
Deploy
Test
Package
Handoff
Production/GA

Example: SDLC Automation with Build Forge



Build Forge Core Components