Rational. software

IBM Rational Build Forge

Adrian Daniels
Technical Consultant, IBM Rational
adrian.daniels@uk.ibm.com

IBM Rational Software Development Conference UK 2007

What keeps me **Rational**?

Agenda

- Introductions
- Build and Release Challenges
- Build Forge Customers – what do they see
- What Build Forge Does
- Technology Overview
- Demonstration
- What next ?

Build and Release Challenges

Current Conditions

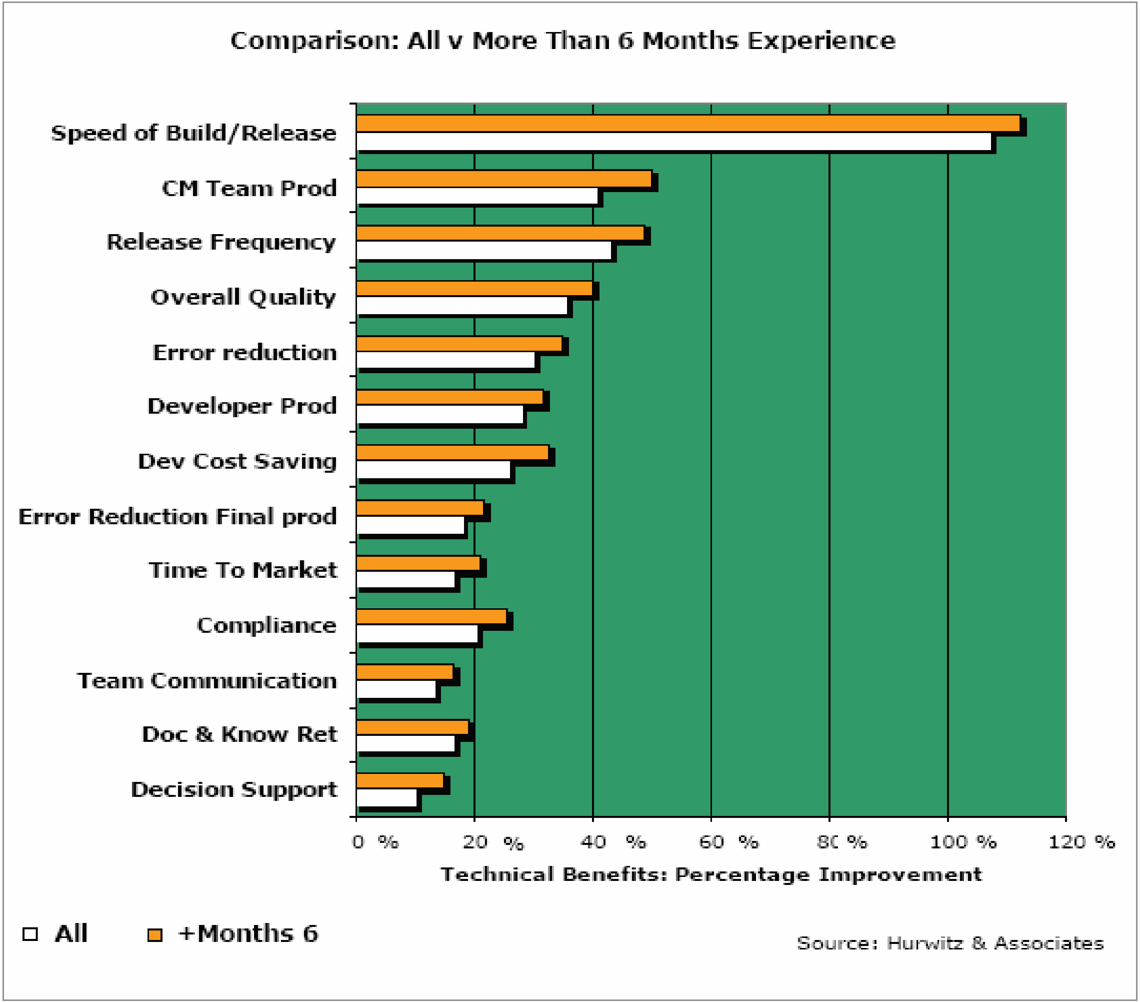
- **Manual**, error-prone build processes
- Proprietary **internally-developed** systems
- **Inconsistent processes** across products and platforms
- **Dependence on build staff** to execute, troubleshoot, and provide feedback
- **Inconsistency** between developer environments and production systems
- **Time consuming** to detect and resolve problems
- Lack of **traceability and compliance** readiness

Business Impact

- **Unpredictable** product release cycles
- **Costly** systems to support & maintain with knowledge held by a few
- **Limited repeatability or portability** for new projects and platforms
- **Staff burden, delays, lost productivity**,
- **Unforeseen errors surface** later in the release cycle
- **More people required** to do more work
- Extensive **ad hoc effort** spent in audit preparation

"Software build management increasingly impacts successful software deployments, business and IT productivity and is becoming a focus for IT organizations." **- IDC**

Build Forge - Business Benefits



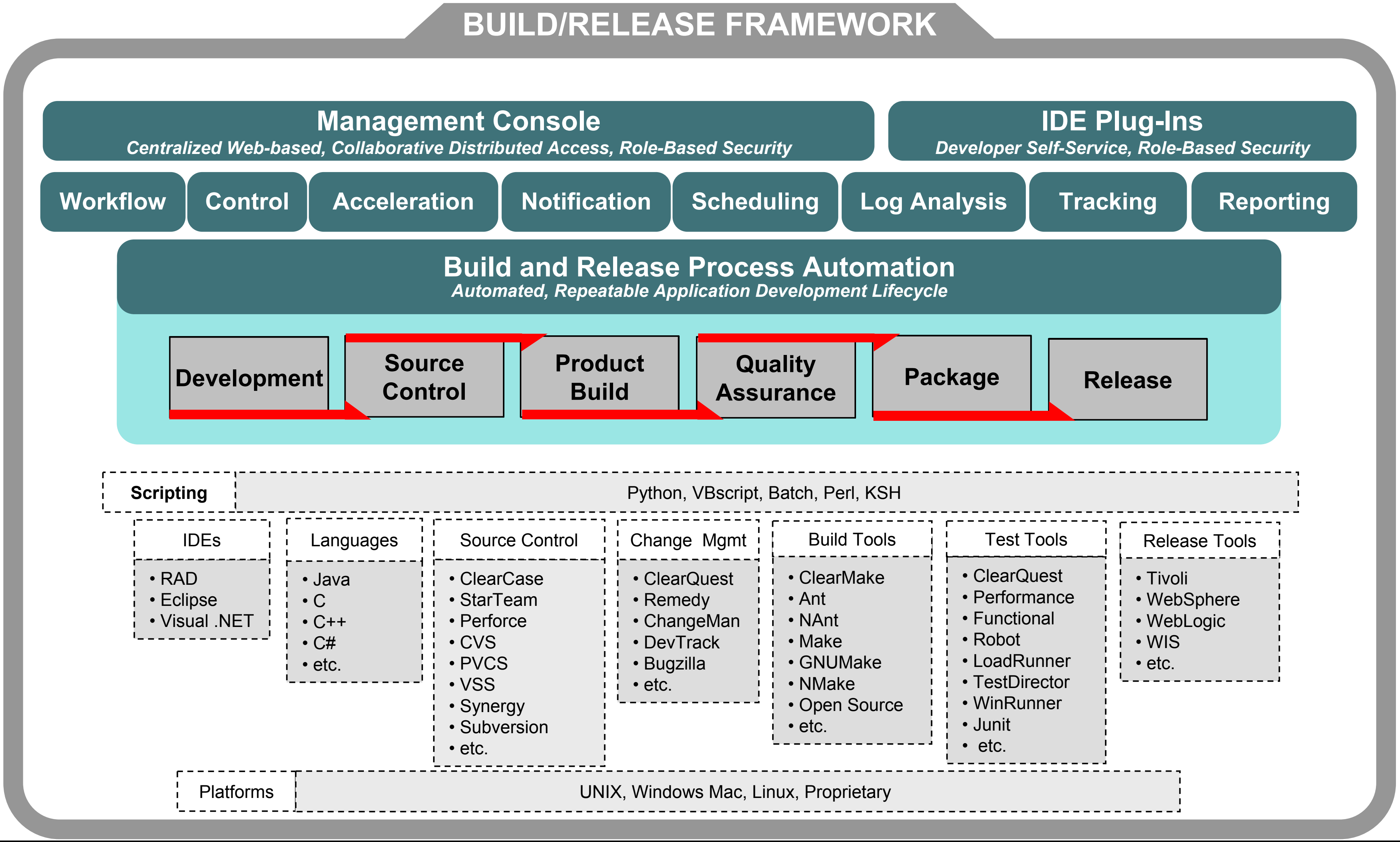
"We discovered that new customers were able to achieve results similar to long-time customers. This validates BuildForge's claims that the product can be implemented and deployed in a relatively short period of time."

-- Hurwitz & Associates

Selected Customers

VERITAS
bp
Adobe
CAMPBELL & COMPANY SINCE 1972
LEHMAN BROTHERS
Google
FIDELITY NATIONAL FINANCIAL
IBM
Morgan Stanley
hp OpenView
TEXAS INSTRUMENTS
SIEMENS
MetLife
Scotiabank
paymetric
EMC² smarts
DIGITAL INSIGHT
CLEARCHANNEL
invensys
Intuit
Fidelity Investments
agile
Reynolds & Reynolds
MULTIMEDIA GAMES
EA.com ELECTRONIC ARTS
LSI LOGIC
Autodesk
EMC² where information lives
VITAL
PERVASIVE
VITAL PROCESSING SERVICES
GE
AVAYA
symantec.
OVERWATCH SYSTEMS
John Hancock
Vicarious Visions
CITADEL
BetweenMarkets
INTERWOVEN Enterprise Content Management Solutions for Business
CITRIX online
Manhattan Associates
Stratus Technologies
EXCEL Where Applications Perform Best
Canon
CONEXANT
OUTCOME POST-MARKETING EXCELLENCE
hp LaserJet
Medplus
WYSE
freescale semiconductor

What We Do



BUILD/RELEASE FRAMEWORK

Management Console
Centralized Web-based, Collaborative Distributed Access, Role-Based Security

IDE Plug-Ins
Developer Self-Service, Role-Based Security

Workflow | **Control** | **Acceleration** | **Notification** | **Scheduling** | **Log Analysis** | **Tracking** | **Reporting**

Build and Release Process Automation
Automated, Repeatable Application Development Lifecycle

Development → **Source Control** → **Product Build** → **Quality Assurance** → **Package** → **Release**

Scripting	Python, VBscript, Batch, Perl, KSH

IDEs
- RAD
- Eclipse
- Visual .NET

Languages
- Java
- C
- C++
- C#
- etc.

Source Control
- ClearCase
- StarTeam
- Perforce
- CVS
- PVCS
- VSS
- Synergy
- Subversion
- etc.

Change Mgmt
- ClearQuest
- Remedy
- ChangeMan
- DevTrack
- Bugzilla
- etc.

Build Tools
- ClearMake
- Ant
- NAnt
- Make
- GNUMake
- NMake
- Open Source
- etc.

Test Tools
- ClearQuest
- Performance
- Functional
- Robot
- LoadRunner
- TestDirector
- WinRunner
- Junit
- etc.

Release Tools
- Tivoli
- WebSphere
- WebLogic
- WIS
- etc.

Platforms	UNIX, Windows Mac, Linux, Proprietary

Rational Case Study

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"

Environment

- 47 Active Projects / Products – 3 Locations
- 8 Platforms, 124 Build Machines
- Rational Products
- Windows, All Unix Flavors

- Release Team is bottleneck. No developer capabilities
- Serial and manual work effort
- 24 Hour "Suite" Build, 14 Hour Product Build

Solution

- Implemented Developer self-service in 3 mos.
- "Suite" and point product builds reduced to 3 hours
- Parallel processes implemented.
- Automated packaging

Siemens Medical Case Study

SIEMENS

"We were interested to adopt Agile Development, but were limited by an inflexible, non-standard build process. Each team did their own thing, and there were multiple points of failure on each project."

Environment

- 1000+ users
- Build machines around the world (US, EMEA, India)
- Continuous unit testing (Cactus and Junit)
- ClearCase, ClearQuest, Test Director

##

- No standards
- No global access
- Multiple points of failure
- Low developer productivity
- No continuous integration

Solution

- # of build cycles increased 3X
- Build times reduced by 65%
- Secure developer self-service established
- $6M savings over 3 years

Electronic Arts Case Study

"The environment necessary for a successful build is very complex, and is different for every product. This information must be carefully maintained and consistently used."

Environment

- 500 Developers – 30 CM's
- 20 Products
- C++, .NET, Perl, Python
- Perforce, DevTrack
- Windows, Xbox, Playstation

- No centralized release mgmt.
- Underutilized server farm
- 60,000+ graphic files built daily that take 30 minutes to 60 hours

Solution

- Build times reduced by as much as 20X. from 60 to 3 hours.
- Machine usage improving – reduced HW buys.
- Management has new intelligence with dashboards.

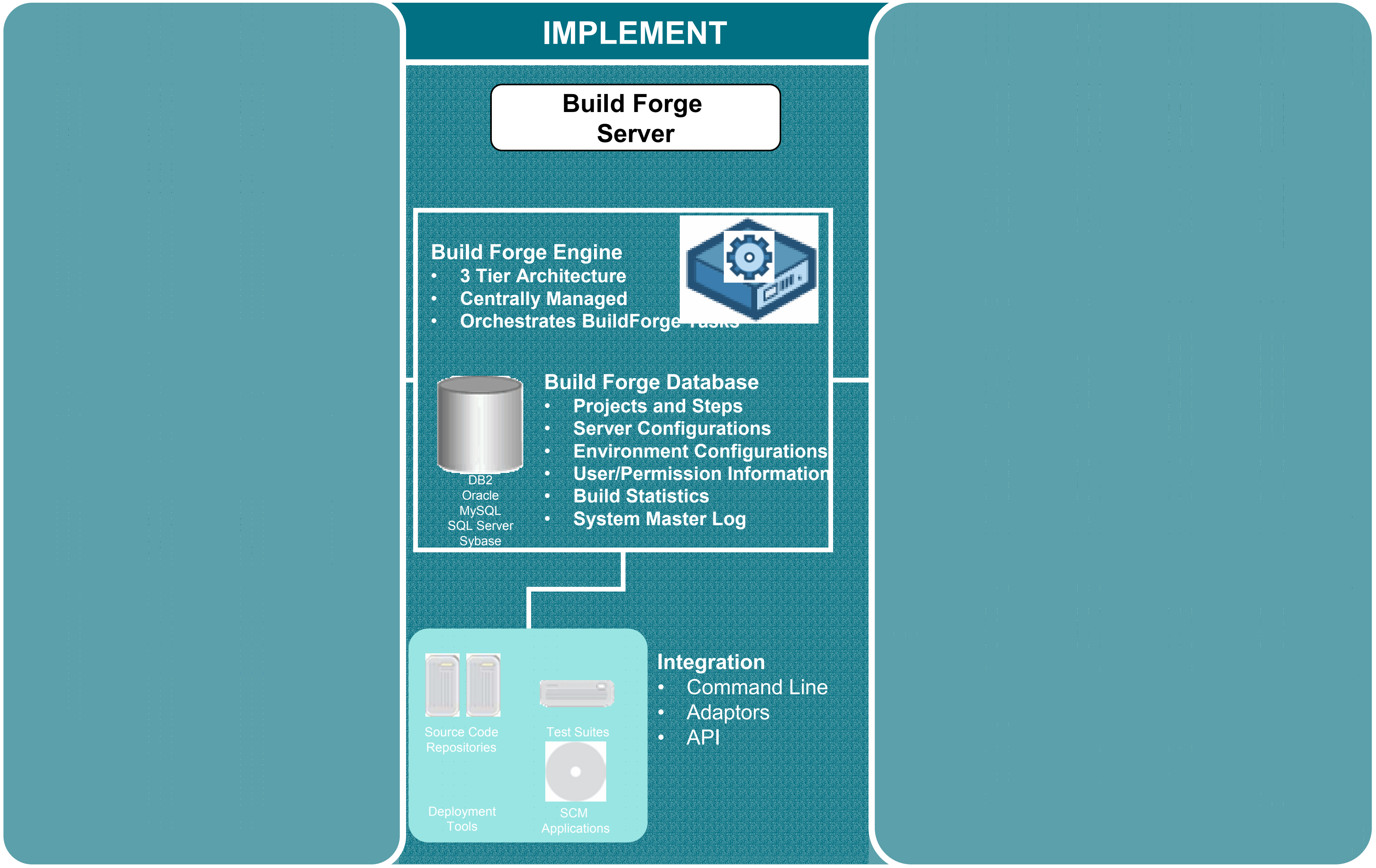
Rational Build Forge – Driving Customer Value

"We were able to improve from 18 builds per week to over 360 builds per week! Across 50 other projects, that will save us millions annually!" -- Adobe

Customer results: higher productivity, improved quality, faster delivery, reduced cost

- **Higher productivity/Reduced cost** – typical payback in less than 6 months, millions saved annually.
- **Increased quality of products** delivered through reliable, repeatable processes and rapid error detection – as much as 70% improvement.
- **Faster software delivery** through more frequent, iterative development cycles. As much as 3 to 20 times faster.
- **Better compliance and governance** with integrated audit trails, traceability, and IT controls for each release

Functional Architecture



Data Elements…Building Blocks…

Steps

```
1a. cleartool mkview -tag $BF_TAG \\views\$BF_TAG
1b. cleartool setcs -tag $BF_TAG config.spec
```

```
2a. gcc main.c -o main.o
2b. gcc main.c -o ui.o
2c. gcc main.o ui.o -o HelloWorld.exe
```

```
3a. testscript.sh -run -r $RELEASE -module HelloWorld.exe
```

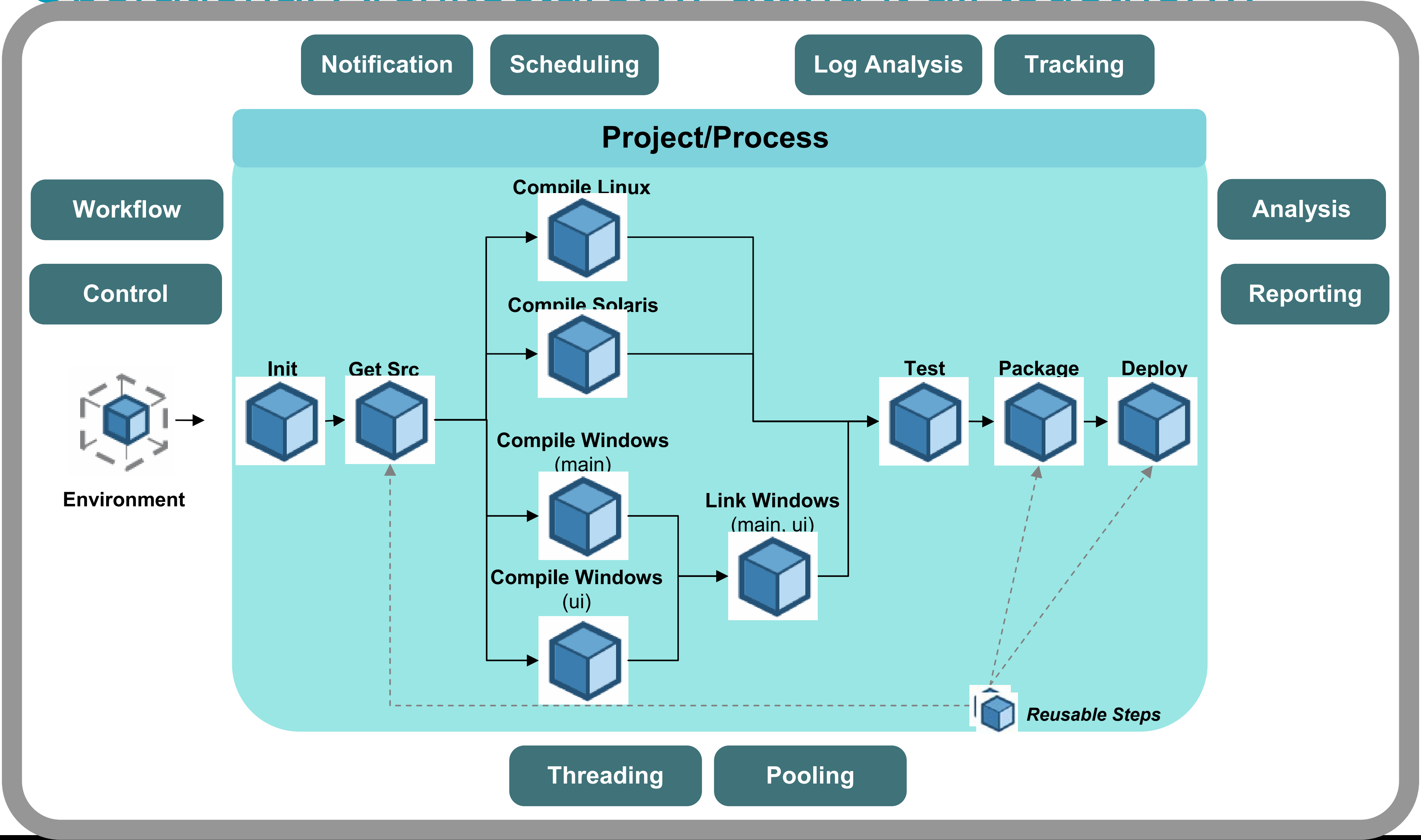
Project

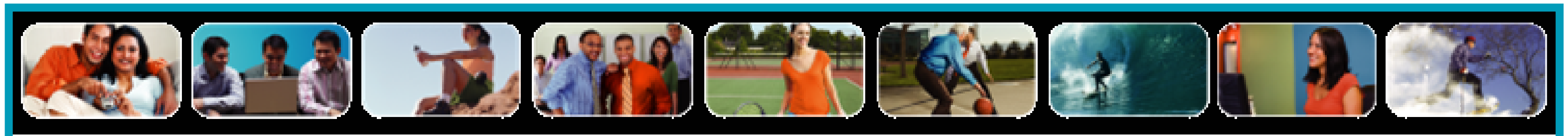
Source Build Test Package Deploy …

Environment

```
1. RELEASE=Release_1.1
2. JAVA_HOME=C:\Program Files\Java\jdk1.5.0_06
3. PATH=C:\:\Program Files\Java\jdk1.5.0_06\bin
4. …
```

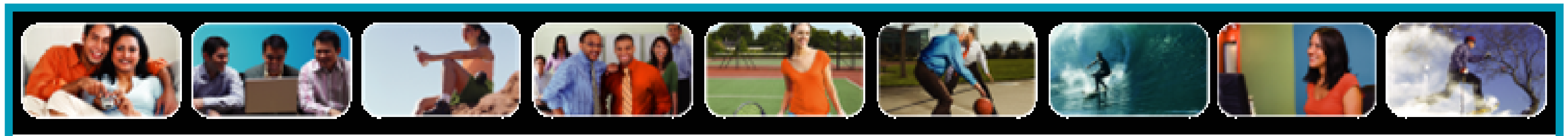
Operational Architecture…Putting it all together…





Demo

Questions



Thank You

Adrian Daniels

adrian.daniels@uk.ibm.com