

Andrew Patterson

E-Mail : apatterson@electric-cloud.com

Electric Cloud Solutions



www.electric-cloud.com

What is the biggest issue?

Topics that have been discussed this week:

- **Well Java, yes**
- **Software complexity?**
- **Agile?**
- **Distributed development?**
- **Developer productivity?**
- **Time to market pressure?**
- **Underutilization of hardware?**
- **Software quality?**
- **Outsourcing?**
- **Centralization?**
- **Control?**
- **Compliance?**
- **Visibility?**

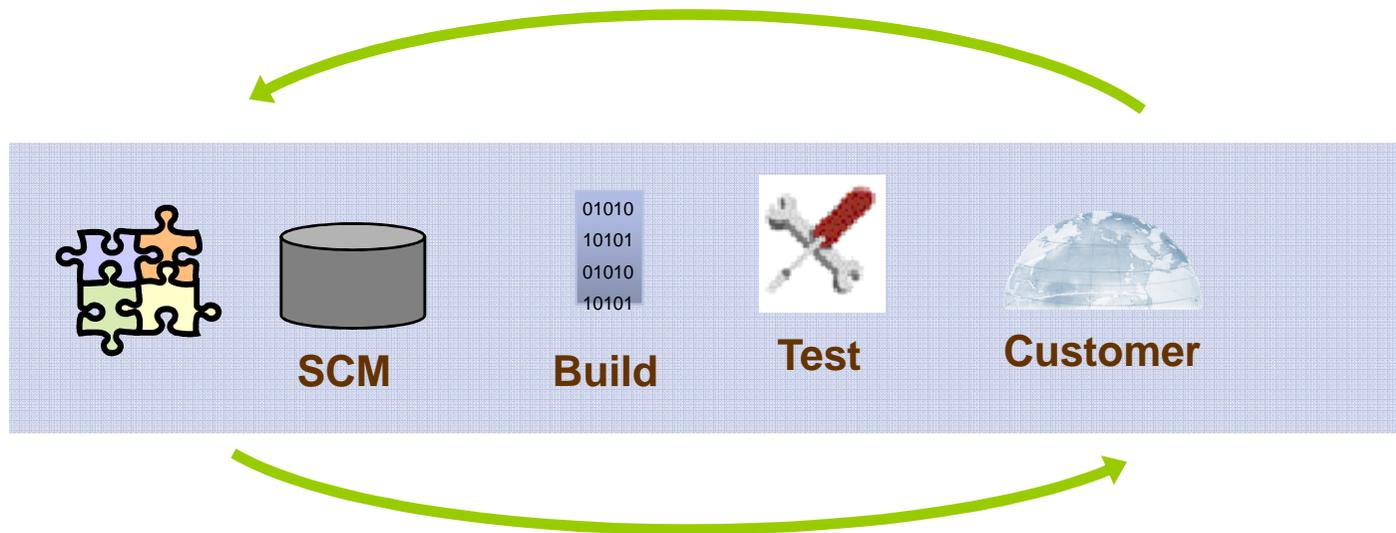
Answer: All of the Above

- **The stakes have never been higher...**
- **So, how do you measure the health of your software projects?**
 - Number of check-ins?
 - Bugs fixed? Bugs outstanding?
 - Specs completed? On-schedule? On-budget?
- **The clearest indicator of project health =**
Working Software

Without built software, there's no way to test to see what is done.

- Carey Schwaber, Senior Analyst, Forrester Research

The goal is fast, high quality development



- Each cycle round this loop can take **DAYS/WEEKS**
- Reduce iterations to reduce “time to market”

- High speed builds
- More right-first-time builds
- Automation (Build, Test, Deployment)

- Reduce “bad code” check-ins
- Test early (*by developers*)

Move problem detection upstream

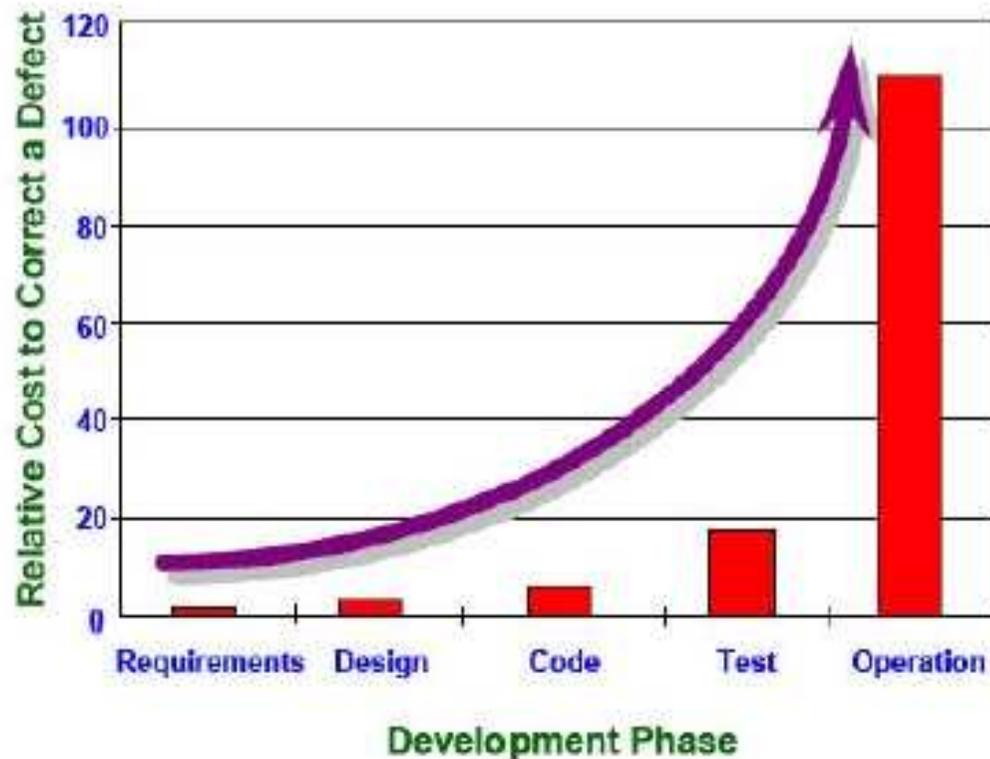
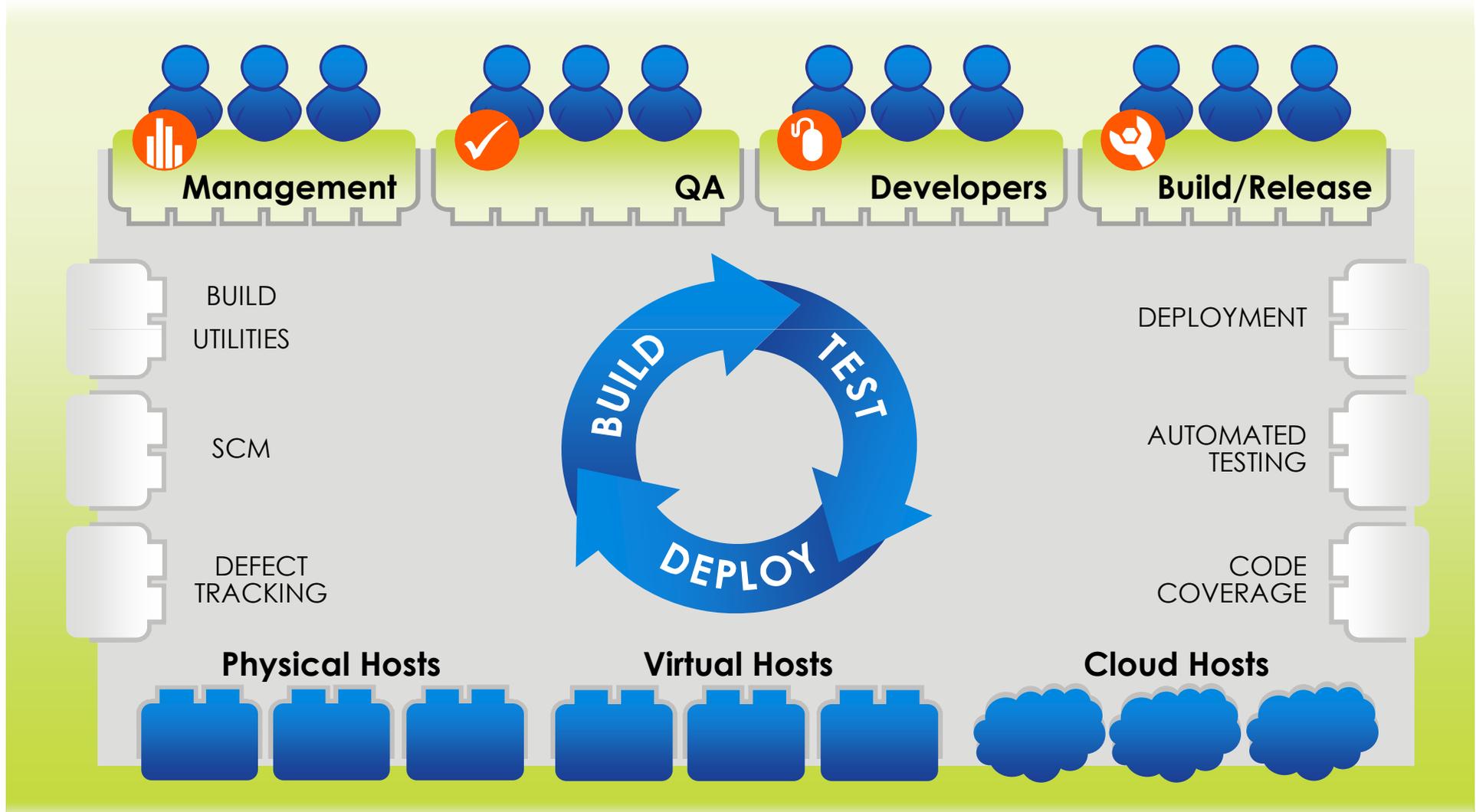


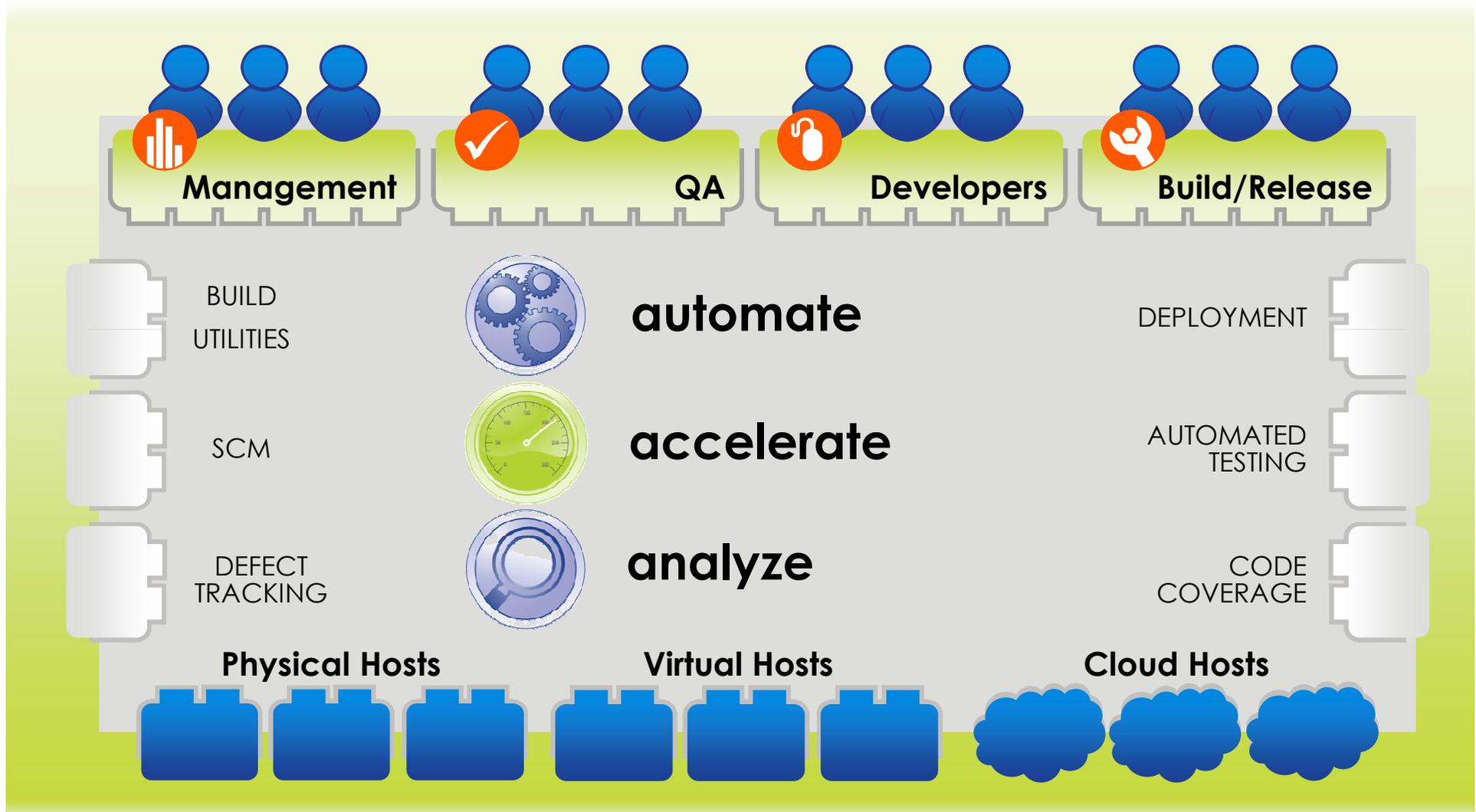
Figure 1 - Increasing cost of defect correction

- Software Build “early and often”
- Test earlier in lifecycle
- More upstream code validation
- Improve Code Quality
- Reduce Project Times

Software Production Management



Where Electric Cloud Helps



Electric Cloud Solutions

- **Build Management**
- **Build Acceleration**
- **Geographically Distributed Development**
- **Agile Development / Continuous Integration**
- **Virtualization**
- **Software Quality**
- **Centralization and Control**
- **Compliance**

Build Management

Challenges with Today's Systems:

- Script-based systems work, but they are hard to maintain
- Scaling to multiple platforms, multiple targets, multiple teams is complex
- Only the experts who built them can run and modify them
- They are slow and not designed to run tasks in parallel



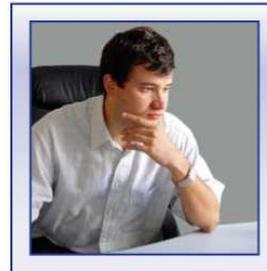
Solution Requirements:

- Flexible, easy to use framework for software production
- Re-use of processes and components
- Parallel execution of build and test tasks to reduce cycle time
- Scalability to handle a large number of projects, builds, tests

Managing Global Teams



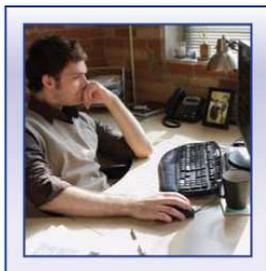
SW DEVELOPERS
Copenhagen



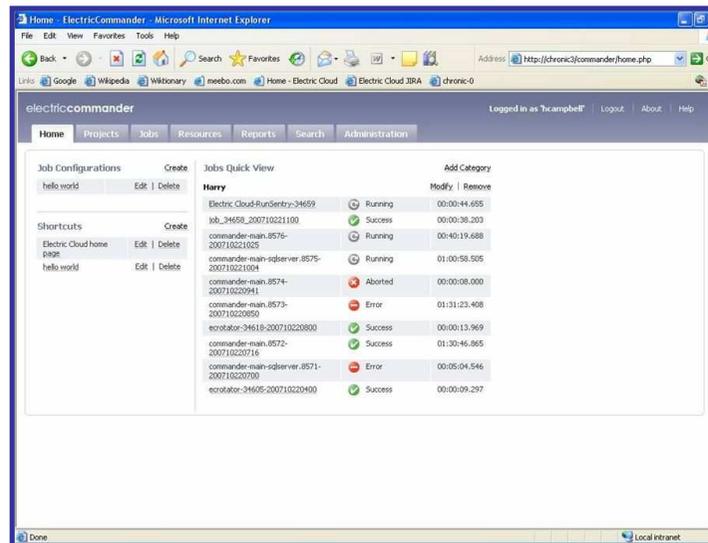
ENGINEERING MGR
Aarhus



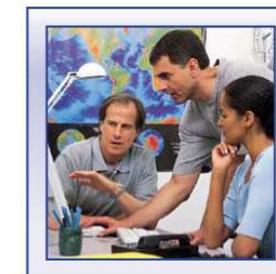
BUILD TEAM
San Francisco



SW DEVELOPERS
San Francisco



Job Configurations	Create	Jobs Quick View	Add Category
hello world	Edit Delete	Harry	Modify Remove
		Electric Cloud-RunSentry:34659	Running 00:00:44.655
		job_34659_200710221100	Success 00:00:38.203
		commander-main.8576-200710221025	Running 00:40:19.688
		commander-main-observer.8575-200710221004	Running 01:00:58.505
		commander-main.8574-200710220941	Aborted 00:00:08.000
		commander-main.8573-200710220850	Error 01:31:23.408
		ecrotator:34618-200710220800	Success 00:00:13.969
		commander-main.8572-200710220716	Success 01:30:46.865
		commander-main-observer.8571-200710220700	Error 00:05:04.546
		ecrotator:34605-200710220400	Success 00:00:09.297



OUTSOURCED QA
Bangalore

**Specific Access and Permissions Based on Role
Anywhere in the World**

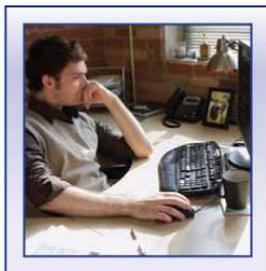
Managing Resources



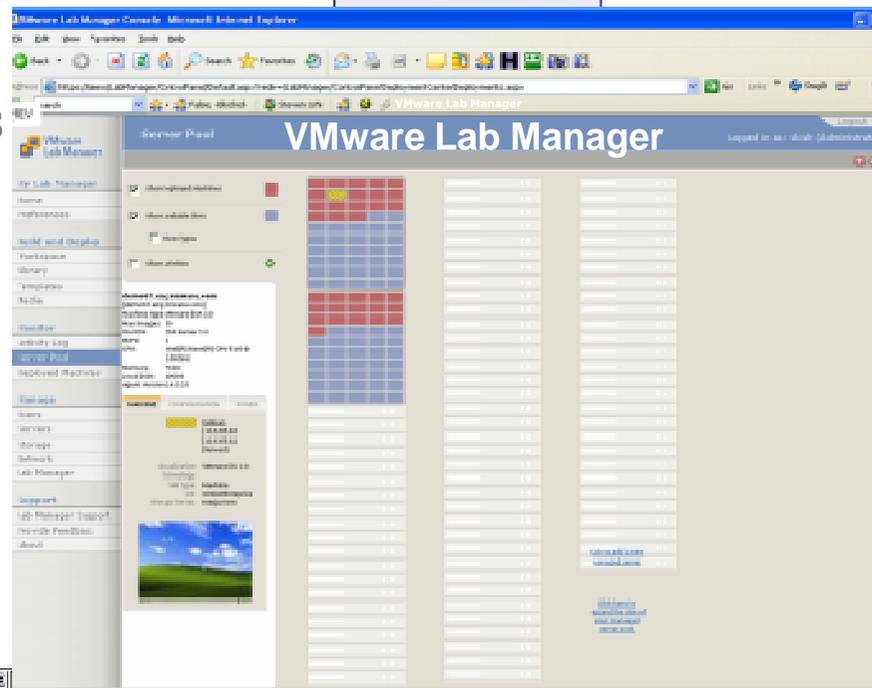
SW DEVELOPERS



BUILD TEAM



SW DEVELOPERS



VMware Lab Manager

Server Pool

Resource	Capacity	Usage	State
Server 1	100%	80%	Running
Server 2	100%	60%	Running
Server 3	100%	40%	Running
Server 4	100%	20%	Running
Server 5	100%	10%	Running
Server 6	100%	5%	Running
Server 7	100%	2%	Running
Server 8	100%	1%	Running
Server 9	100%	0%	Running
Server 10	100%	0%	Running

Dynamic Provisioning

OUTSOURCED

QA



Build Servers



Test Servers

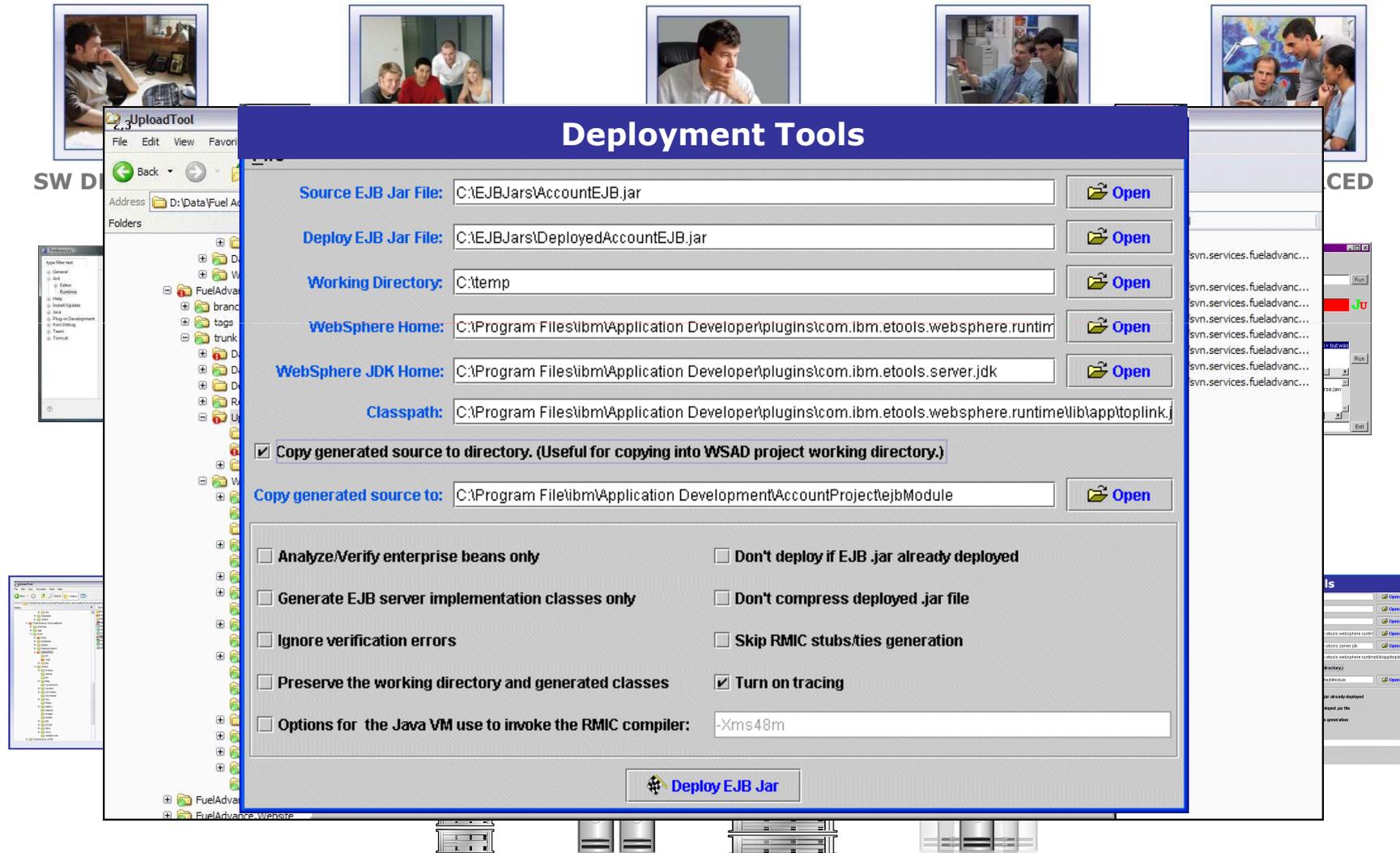


Production Servers



Virtual Servers

Integrate Tools and Processes



Deployment Tools

Source EJB Jar File: C:\EJBJar\AccountEJB.jar **Open**

Deploy EJB Jar File: C:\EJBJar\DeployedAccountEJB.jar **Open**

Working Directory: C:\temp **Open**

WebSphere Home: C:\Program Files\IBM\Application Developer\plugins\com.ibm.etools.websphere.runtime **Open**

WebSphere JDK Home: C:\Program Files\IBM\Application Developer\plugins\com.ibm.etools.server.jdk **Open**

Classpath: C:\Program Files\IBM\Application Developer\plugins\com.ibm.etools.websphere.runtime\lib\app\toplink.jar

Copy generated source to directory. (Useful for copying into WSAD project working directory.)

Copy generated source to: C:\Program Files\IBM\Application Developer\AccountProject\ejbModule **Open**

Analyze/Verify enterprise beans only Don't deploy if EJB .jar already deployed

Generate EJB server implementation classes only Don't compress deployed .jar file

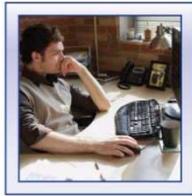
Ignore verification errors Skip RMIC stubs/ties generation

Preserve the working directory and generated classes Turn on tracing

Options for the Java VM use to invoke the RMIC compiler: -Xms48m

Deploy EJB Jar

Tie it all Together



SW DEVELOPERS



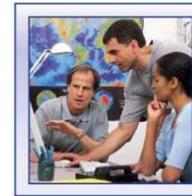
SW DEVELOPERS



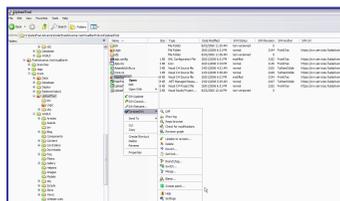
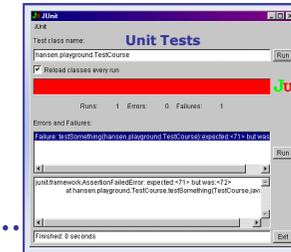
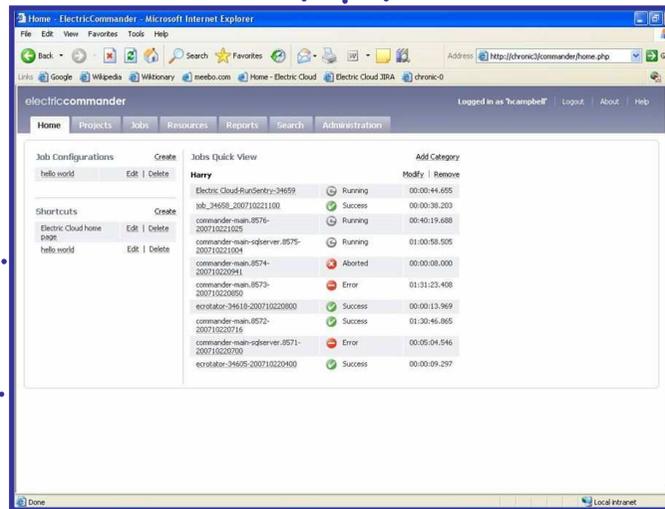
ENGINEERING MGR



BUILD TEAM



OUTSOURCED
QA



Build Servers



Test Servers

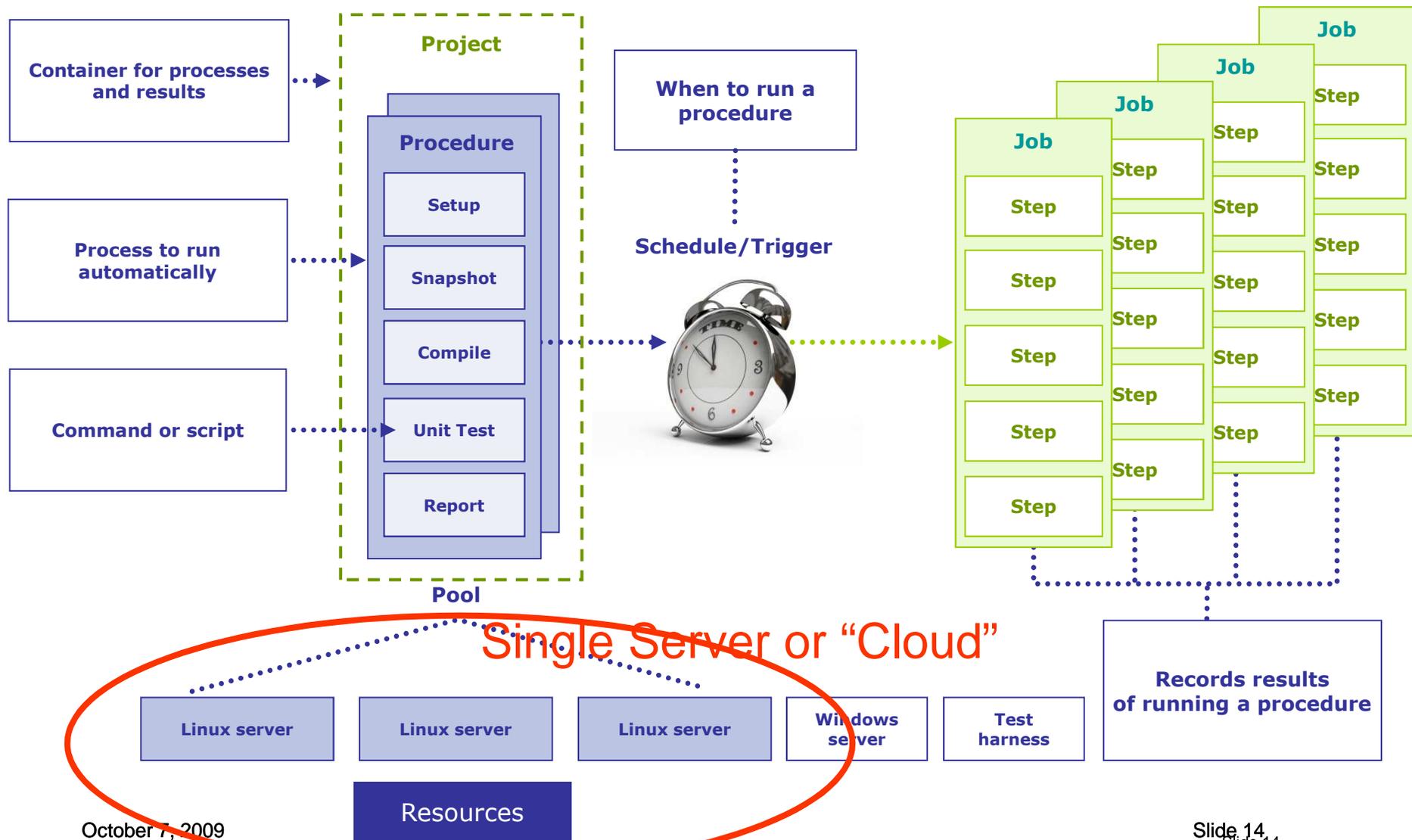


Production Servers

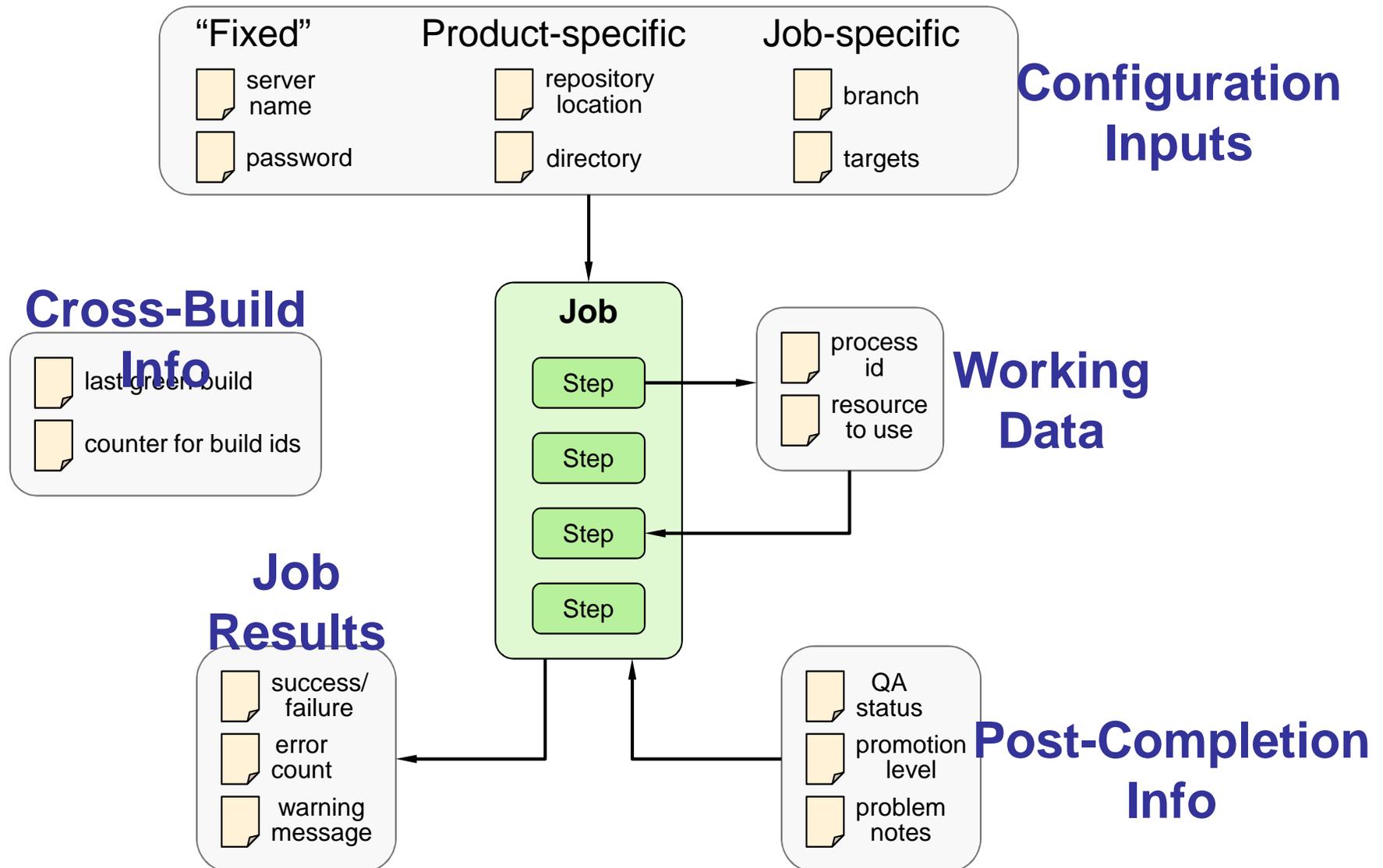


Virtual Servers

ElectricCommander Architecture



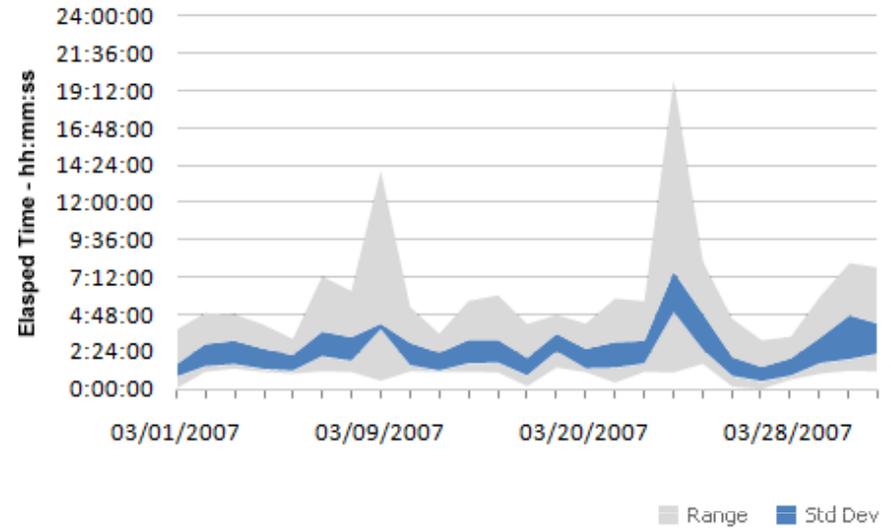
Managing Build and Test Data



Outcome Trend

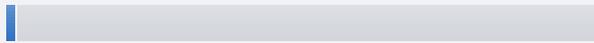
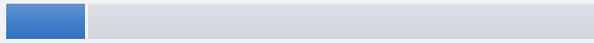
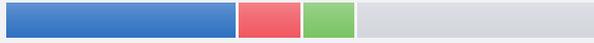
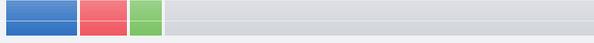
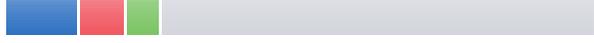
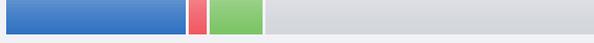
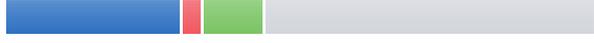
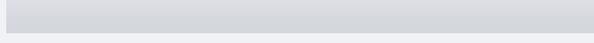


Elapsed Time Trend



Date	Job	Schedule	Status	Elapsed Time	Start Time
03/01/2007	mckinley-main.13190-200703010103	sentry_ea_main	error	3:20	2007-03-01T01:03:18.519
03/01/2007	mckinley-main.13192-200703010112	sentry_ea_main	error	3:37:04	2007-03-01T01:12:18.500
03/01/2007	mckinley-main.13195-200703010451	sentry_ea_main	error	2:29:42	2007-03-01T04:51:19.889
03/01/2007	mckinley-main.13196-200703010936	sentry_ea_main	error	1:53:34	2007-03-01T09:36:19.60
03/01/2007	mckinley-main.13197-200703011130	sentry_ea_main	error	45:58	2007-03-01T11:30:19.963
03/01/2007	mckinley-main.13198-200703011221	sentry_ea_main	error	33:47	2007-03-01T12:21:24.439
03/01/2007	mckinley-main.13199-200703011327	sentry_ea_main	success	1:43:18	2007-03-01T13:27:18.764

Resource Summary - 1 Day 3/25/2007

Host	Resource	Concurrent Steps Running as % of Day	Total Steps	Load Factor	Total Time
jotest	jotest		21	0.02	22:31
jo-linux	jo-linux		5	0.01	9:11
installer-win2	installer-win2		9	0.02	28:31
installer-win1	installer-win1		39	0.08	2:00:54
eng	eng		170	0.05	71:55
ecbulid-win2	ecbulid-win2		260	0.52	12:21:58
ecbuild-win1	ecbuild-win1		243	0.46	10:63:11
ecbuild-sol2	ecbuild-sol2		144	0.18	4:20:55
ecbuild-sol1	ecbuild-sol1		145	0.19	4:38:50
ecbuild-lin2	ecbuild-lin2		209	0.35	8:18:59
ecbuild-lin1	ecbuild-lin1		209	0.33	8:01:17
chronic3	chronic3		0	0.00	0

■ 1 Step
 ■ 2 Steps
 ■ >3 Steps

Reported on March 26, 2007

Customer Spotlight: Build Management

Intuit Software

Problem:

- Building 1200 product variants (SKUs) on existing build system was unmanageable
 - Large matrix of product versions X platforms X tax jurisdictions
- Small process changes resulted in hundreds of manual changes to the individual procedures for each variant
- Needed to integrate with existing tools

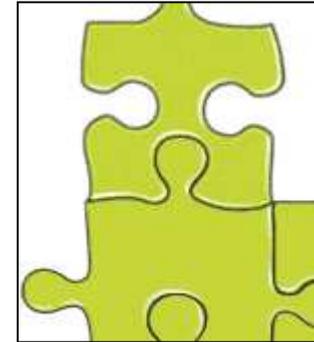
Solution: ElectricCommander

- Scalable, high-performance build automation
- Property mechanism allows process changes to be immediately inherited across a large number of product variants
- Complete API eased integration with existing tools
- Initially deploying for production builds, plan to expand to developer “pre-flight” builds

Agile Development/Continuous Integration

Challenges:

- Build process often requires manual intervention (setting parameters, copying files, etc.)
- Failures due to build infrastructure prevent regular, reliable builds
- Build and test cycle time is too long to build frequently



Solution Requirements:

- SCM integration
- Automated, reliable build and test execution upon developer check-in
- Fast build and test cycles to allow frequent integration

Virtualized Build and Test

Challenges:

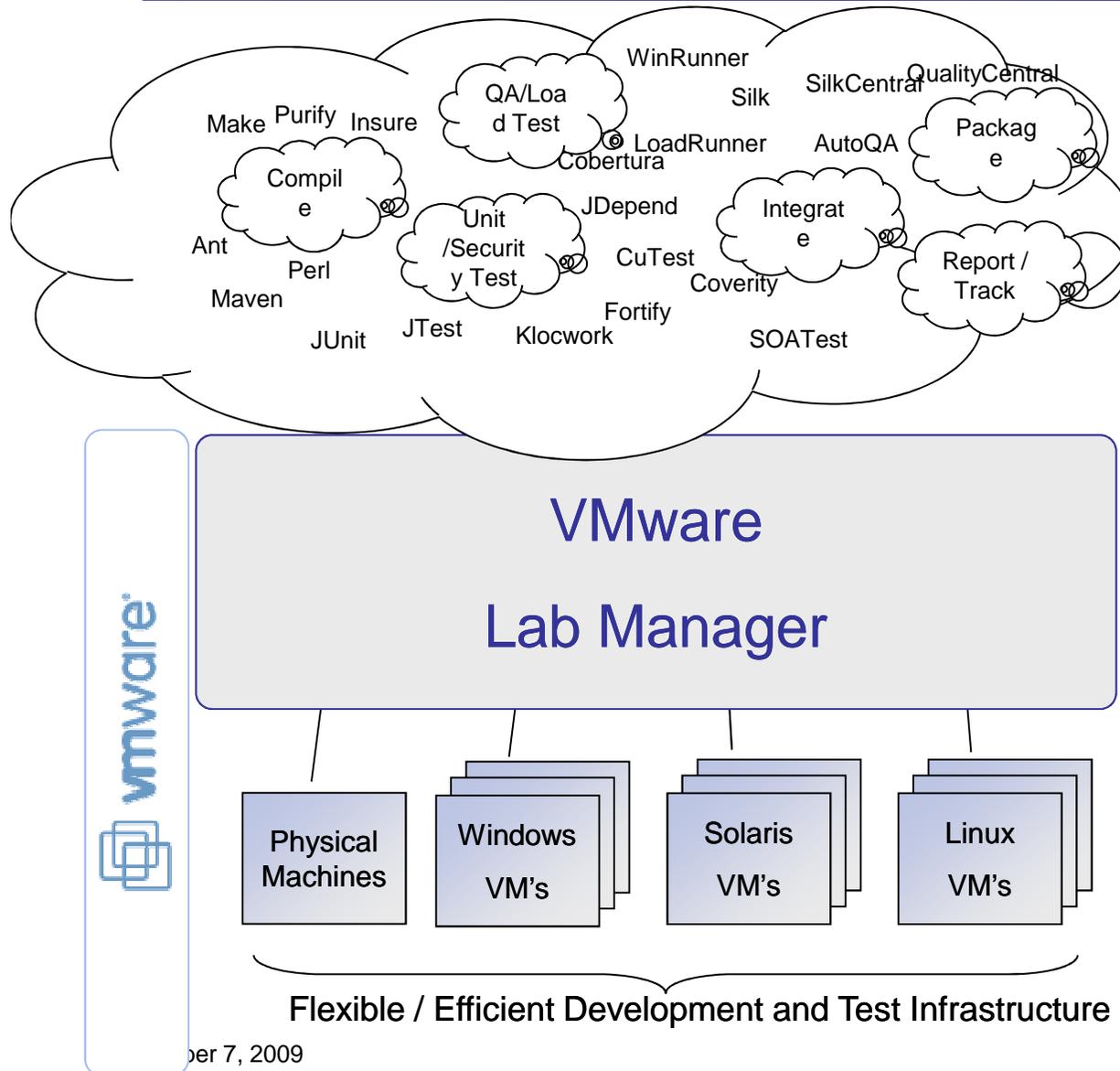
- **Software development infrastructure is underutilized and costly to manage**
- **Hours or days to set up each new environment**
- **Difficult to run frequent testing on all target architectures**



Solution Requirements:

- **Rapid, self-service provisioning of virtual environments**
- **Ability to support new/changing test configurations quickly**
- **Leverage library of pre-built configurations**
- **Flexible, adaptable process engine that can take advantage of a flexible, adaptable virtual infrastructure**

Completing The Solution



Remaining Problem:

- Can't quickly build up/tear down process / application
- Scripts not fast, flexible, automated or self-service
- Disparate tools not integrated

Lab Management:

- Quickly build up/tear down any lab

Automation helps organizations get the most out of a virtualized development infrastructure

Virtualization:

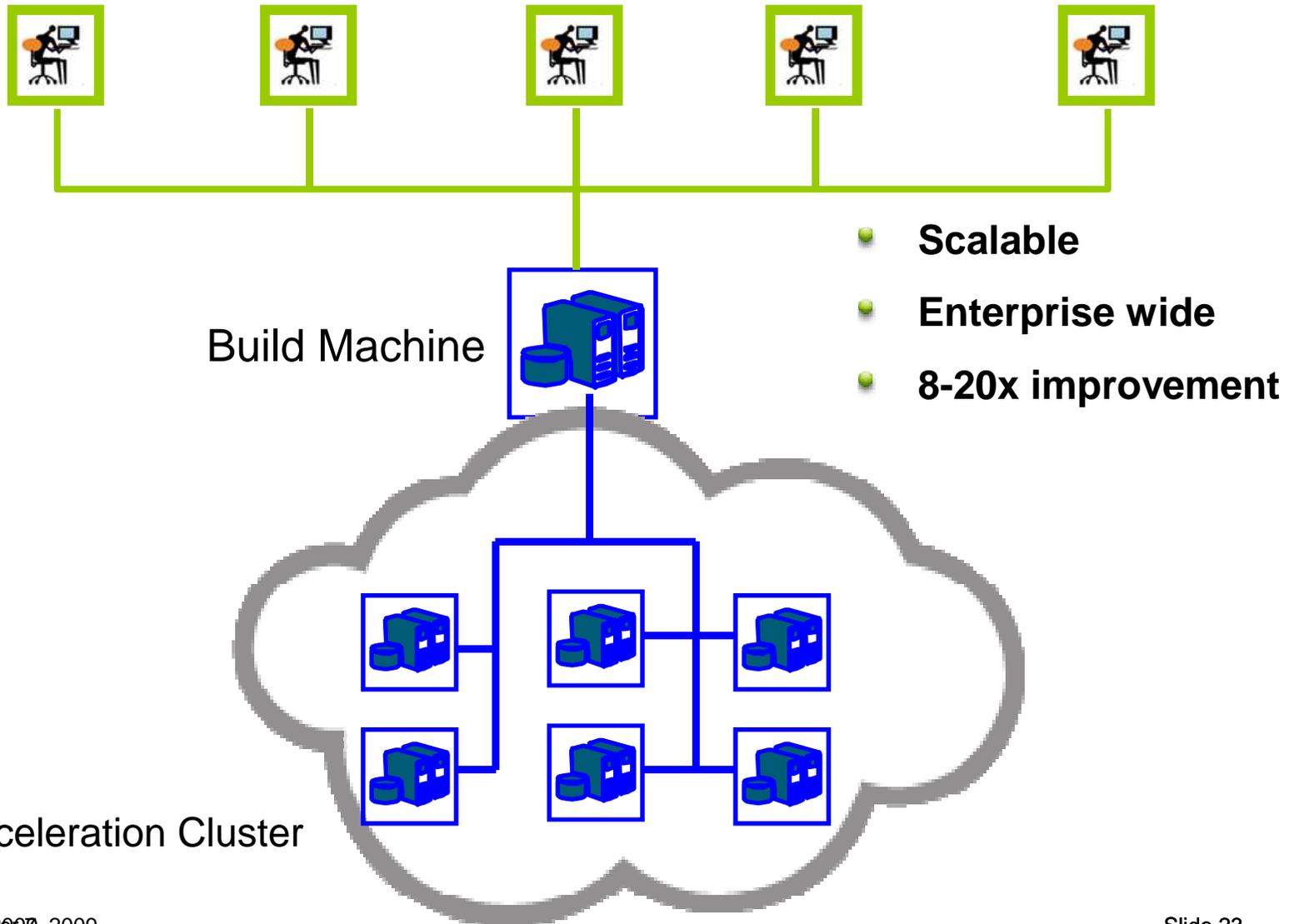
- Quickly build up/tear down any lab
- Configuration, restore and reuse
- Production environment cloning
- Simplify IT Management

Acceleration of Builds

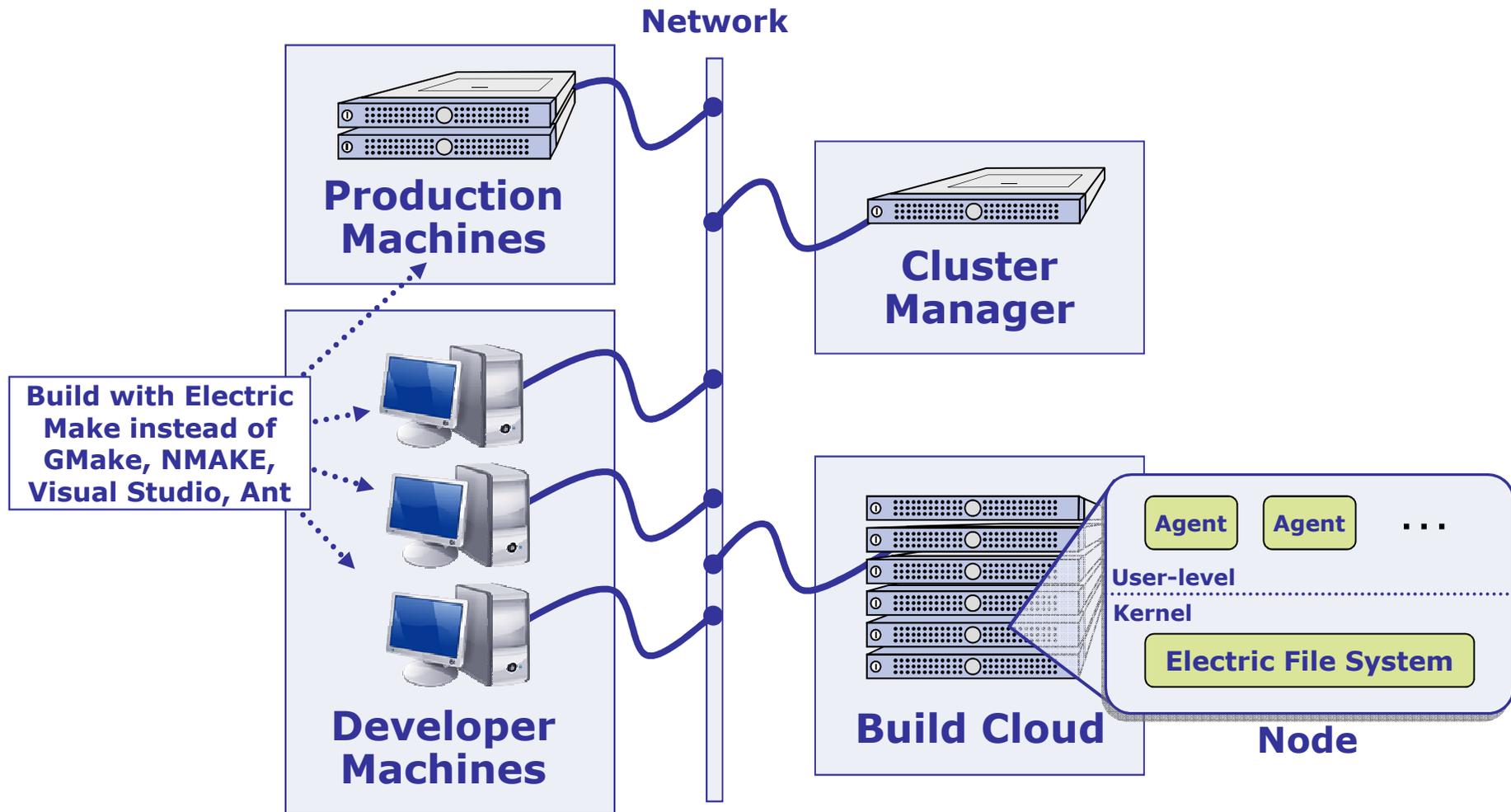


www.electric-cloud.com

Electric Accelerator : Centralized Cluster



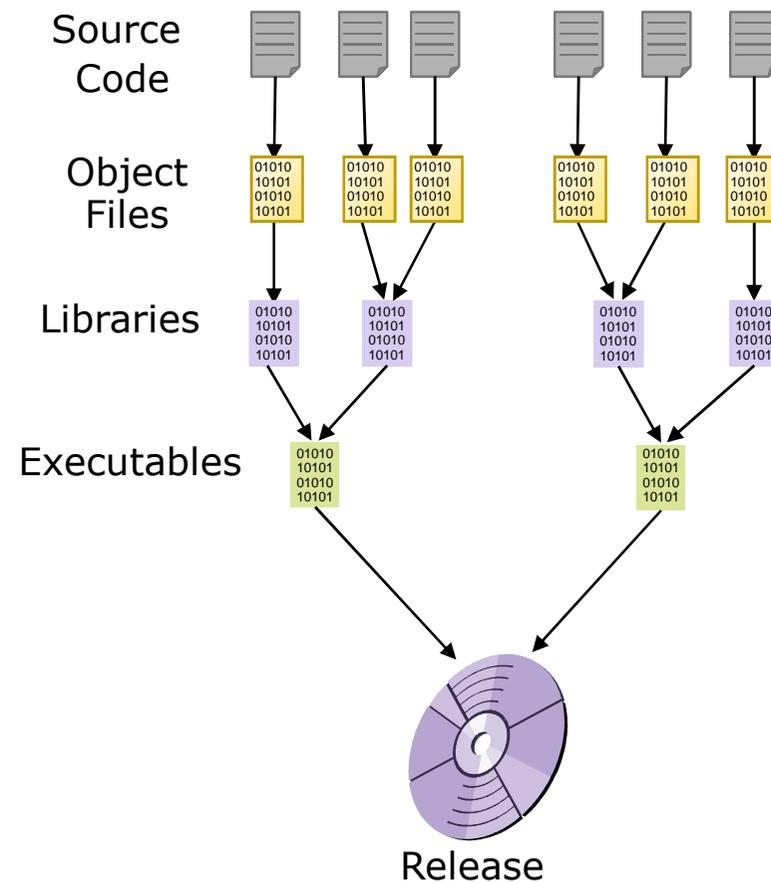
ElectricAccelerator Architecture



Developer builds, production builds leverage the same cluster

Managing Dependencies for Parallel Builds

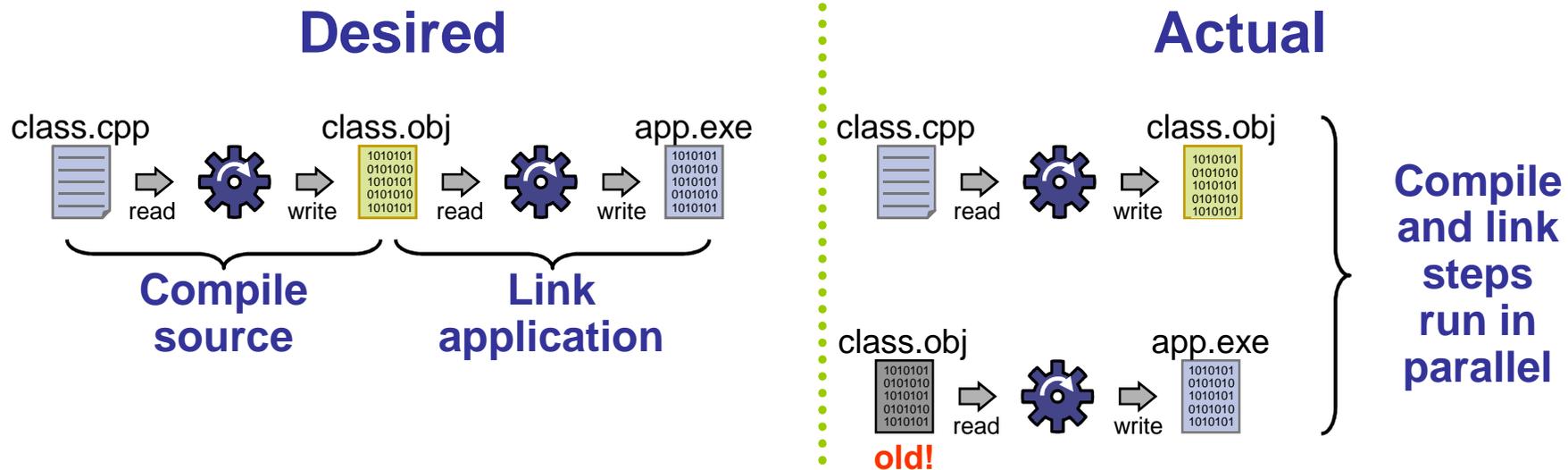
- **Builds have inherent parallelism**
- **Simple to run in parallel**
 - Large SMP Machines (gmake -j)
 - Distributed builds (distcc)
- **Traditionally yield small results due to dependencies:**
 - Incomplete or unknown
 - Implicit dependencies
 - Only understand compiler dependencies
- **Result: broken builds**



Solution for Fast, Accurate Builds

In-Flight Dependency Management

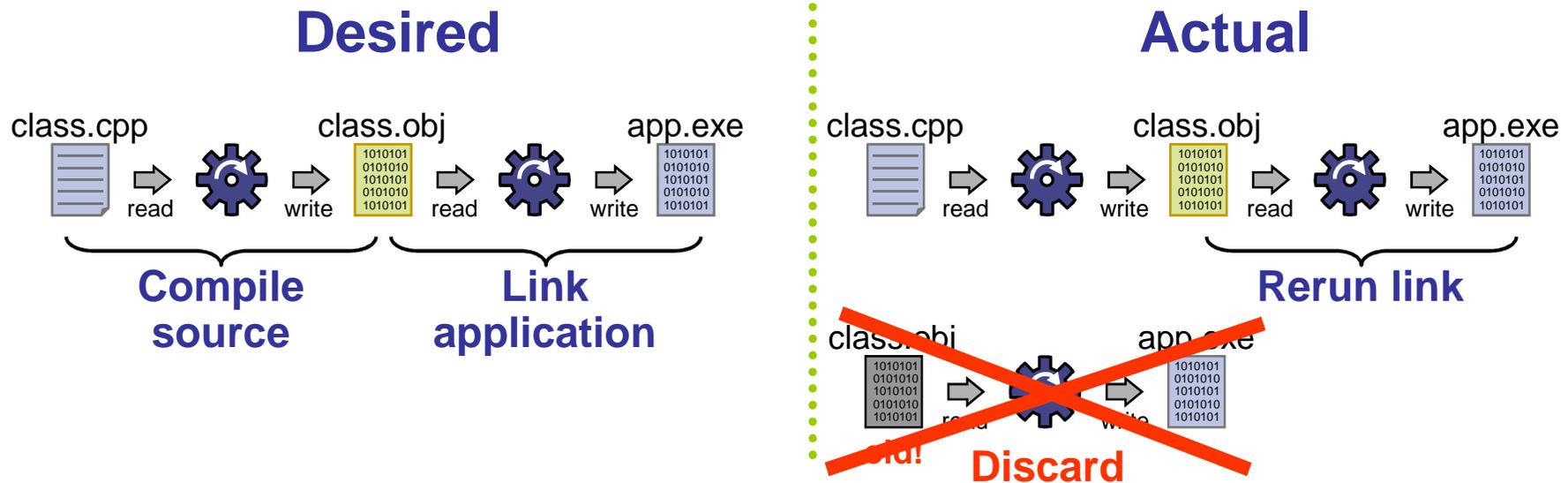
- Watch all file accesses: these indicate dependencies
- Automatically **detect and correct** out-of-order steps



Solution for Fast, Accurate Builds

In-Flight Dependency Management

- Watch all file accesses: these indicate dependencies
- Automatically **detect and correct** out-of-order steps
- Save discovered dependencies for future builds
- Result: **safely unlock high concurrency**



Works Seamlessly in Environment

Works seamlessly with existing infrastructure

- GNU Make, NMAKE, Visual Studio, Ant, and more
- No learning a new interface/tool
- Use existing build files and scripts
- Identical log files



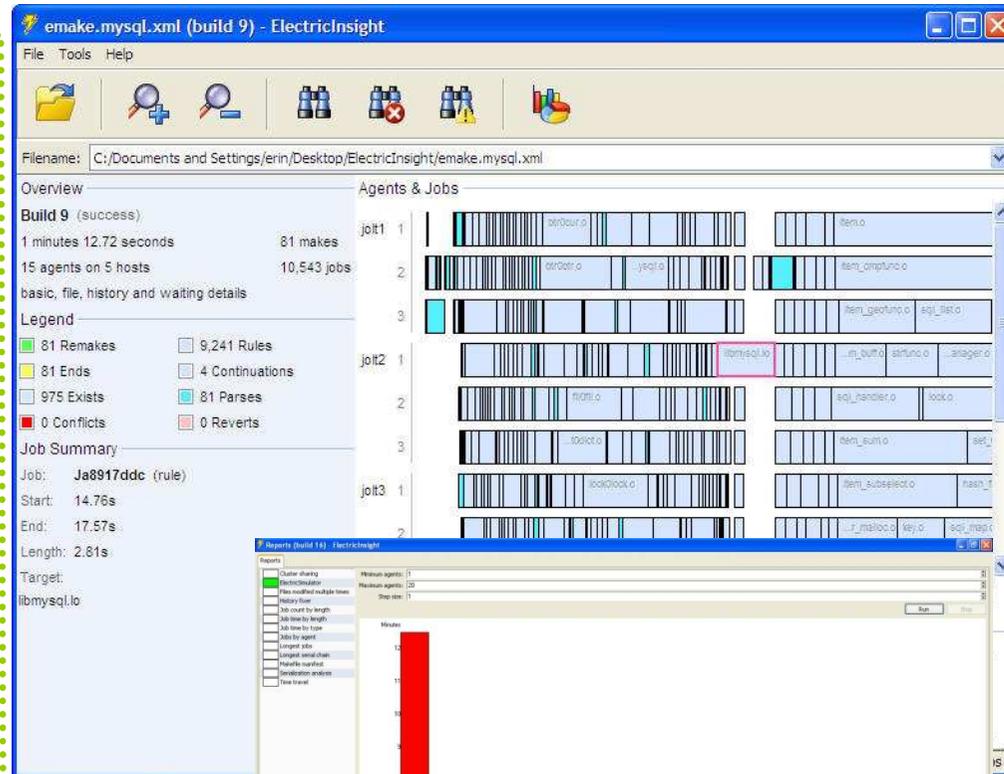
Works with grid and high performance computing environments

- Fully integrated with the Platform LSF grid computing solution
- Optimized for Microsoft High Performance Computing servers

.....
Minimal changes required for end users

ElectricInsight

BUILD



**Understand the details of
your build**

**Enable further
optimization**

Impact of Faster Builds: *Productivity*

Build Time	Production Builds	Developer Builds
14 hours	Build doesn't finish overnight	Impractical to rebuild before checkin
6 hours	Overnight build	Impractical to rebuild before checkin
2 hours	Same-day fix for a broken build	Less likely to rebuild before checkin
30 min.	Continuous integration	Full rebuild before checkin
5 min.	Continuous integration	Little need to switch context
1 min.	Continuous integration	No need to switch context

Customer Spotlight: Build Acceleration

Global Semiconductor Company

Problem:

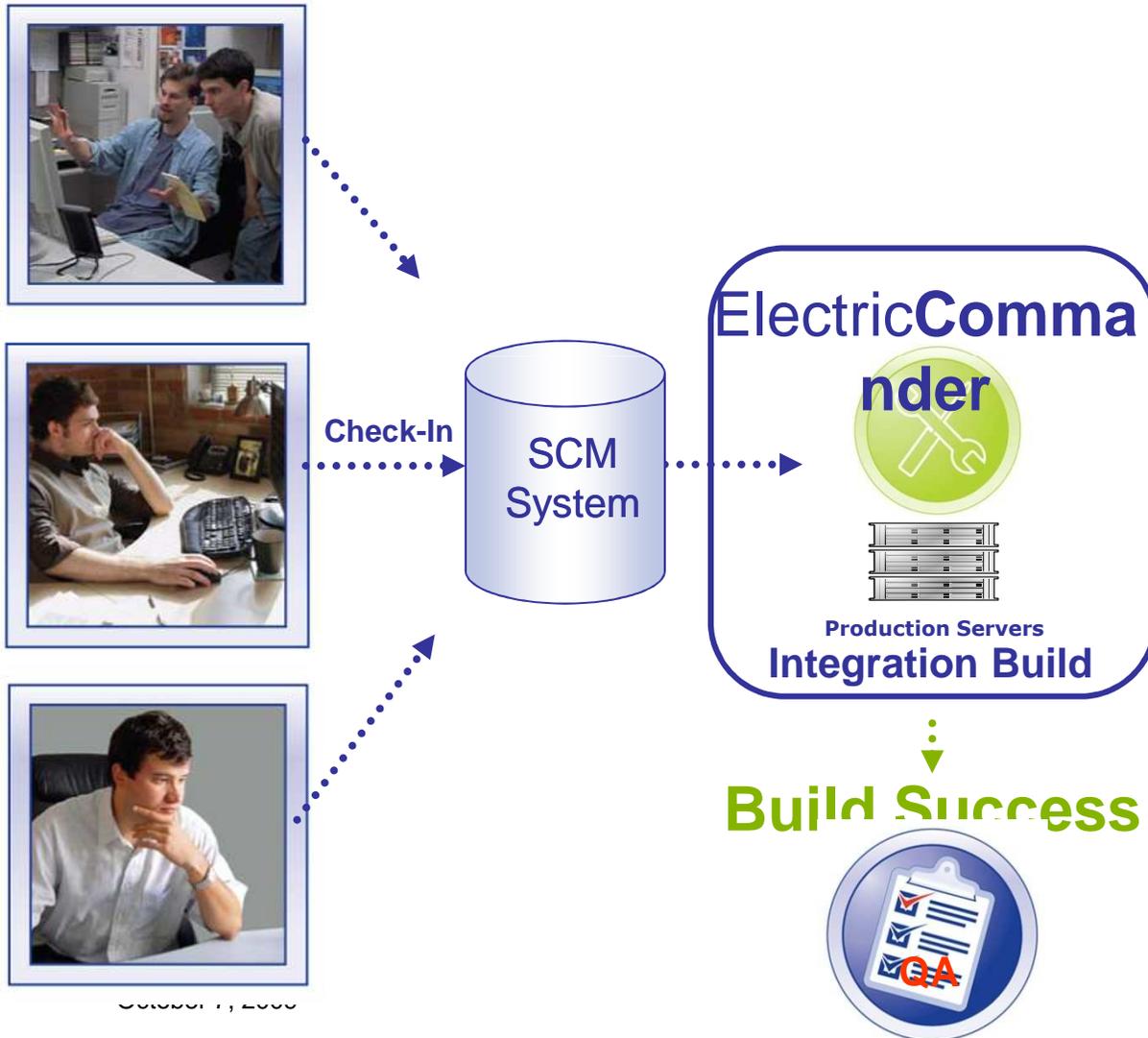
- Trying to implement continuous integration
 - Dozens of builds per day, 250+ tests each
- Length of cycle and broken builds led to delays in getting features and fixes to customers

	Before	After
Build	3 hours	25 mins
Test	18 hrs	2.5 hrs
Total	21 hrs	< 3 hrs

Solution: ElectricAccelerator + ElectricCommander

- Parallel builds significantly reduced build times
- Parallel test execution dramatically reduced test cycle time
- Continuous integration improved customer turnaround on features and fixes
- Payback measured in millions of dollars per year

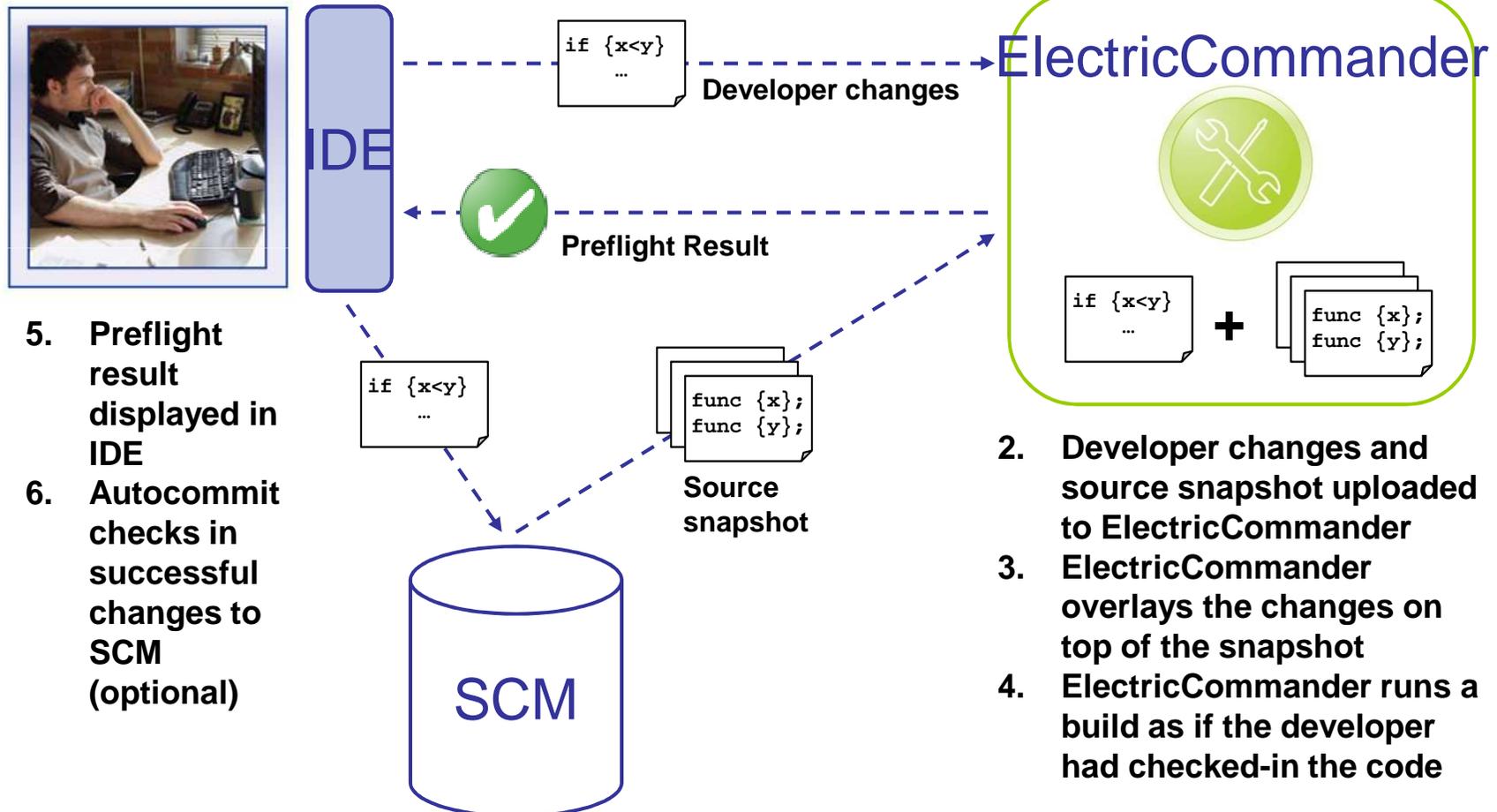
Continuous Integration Theory



- Developer runs local build and automated tests
- Developer checks tested code into SCM system
- Integration build run at frequent intervals or upon check-in

Preflight Workflow

1. Developer invokes preflight through Visual Studio or Eclipse



About Electric Cloud

- **Founded 2002, California HQ, strong presence in EMEA**
- **Market Leader in Software Production Automation**
- **100% year-over-year revenue growth 4 straight years**
- **Several hundred customer sites across 15 countries**
 - **More than 3+ million builds per day**
- **Deep customer penetration:**
 - 8 of the top10 mobile phone/PDA providers
 - > 50% of the Fortune 500 network and communications equipment providers
 - >50% of the Fortune 500 ISVs
 - >50% of the top 10 semiconductor companies
- **Strong financials**
 - **Top tier backing- USVP & Mayfield**
 - **EMEA : Ericsson, Motorola, SonyEricsson, Nokia, Siemens,**

Sample Customers

•Networking	•Semiconductor	•Cellular	•ISV	•Other
				
				
				
				
				
				
				
				

Thank You



www.electric-cloud.com