

The Forrester logo consists of the word "FORRESTER" in a white, serif, all-caps font, followed by a registered trademark symbol (®). The text is centered within a dark green, horizontally-oriented oval. The background is a dark blue gradient with faint, light blue wavy lines that sweep across the frame from the bottom left towards the top right.

FORRESTER®

# Open Source And Its Role In Lean Software

**Jeffrey S. Hammond**

Principal Analyst

Forrester Research

March 17, 2010

**Open source software  
(OSS) is a practical  
enabler of Lean software  
production.**

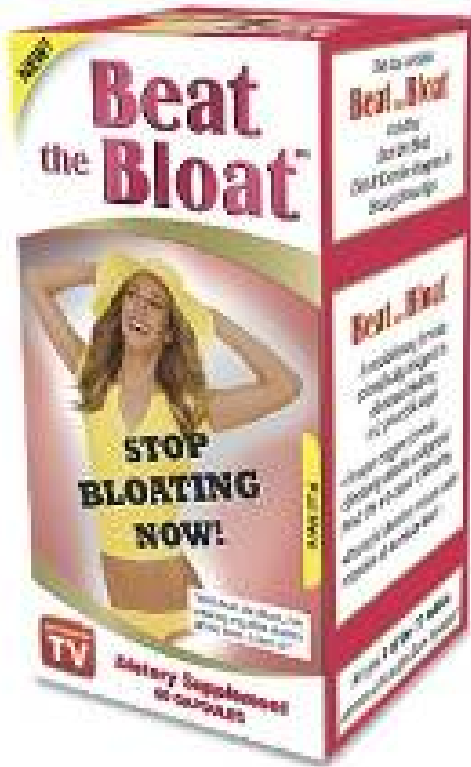
# Agenda

- What is “Lean software” - why does it matter?
- OSS and the lean software movement
- Evolving to a lean, mixed-source dev model

# Agenda

- **What is “Lean software” - why does it matter?**
- OSS and the lean software movement
- Evolving to a lean, mixed-source dev model

# Lean software is the response to bloat



An approach to building, delivering, and running software that **values fit-to-purpose, simplicity, and time-to-results** above all.

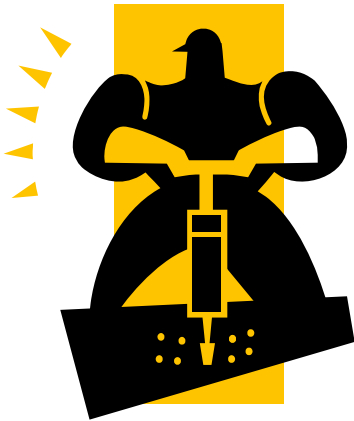
# In the broadest sense

Lean is a management framework, which aims to:

1. Support the development of better concepts for satisfying customer needs.
2. Improve processes and information flows.
3. Support the development of a culture of continuous improvement.
4. Reduce waste.



# Lean Software takes root when ...



... maintenance  
eats 80% to 90%  
of IT budgets.



... business  
innovations  
require  
responsive IT.

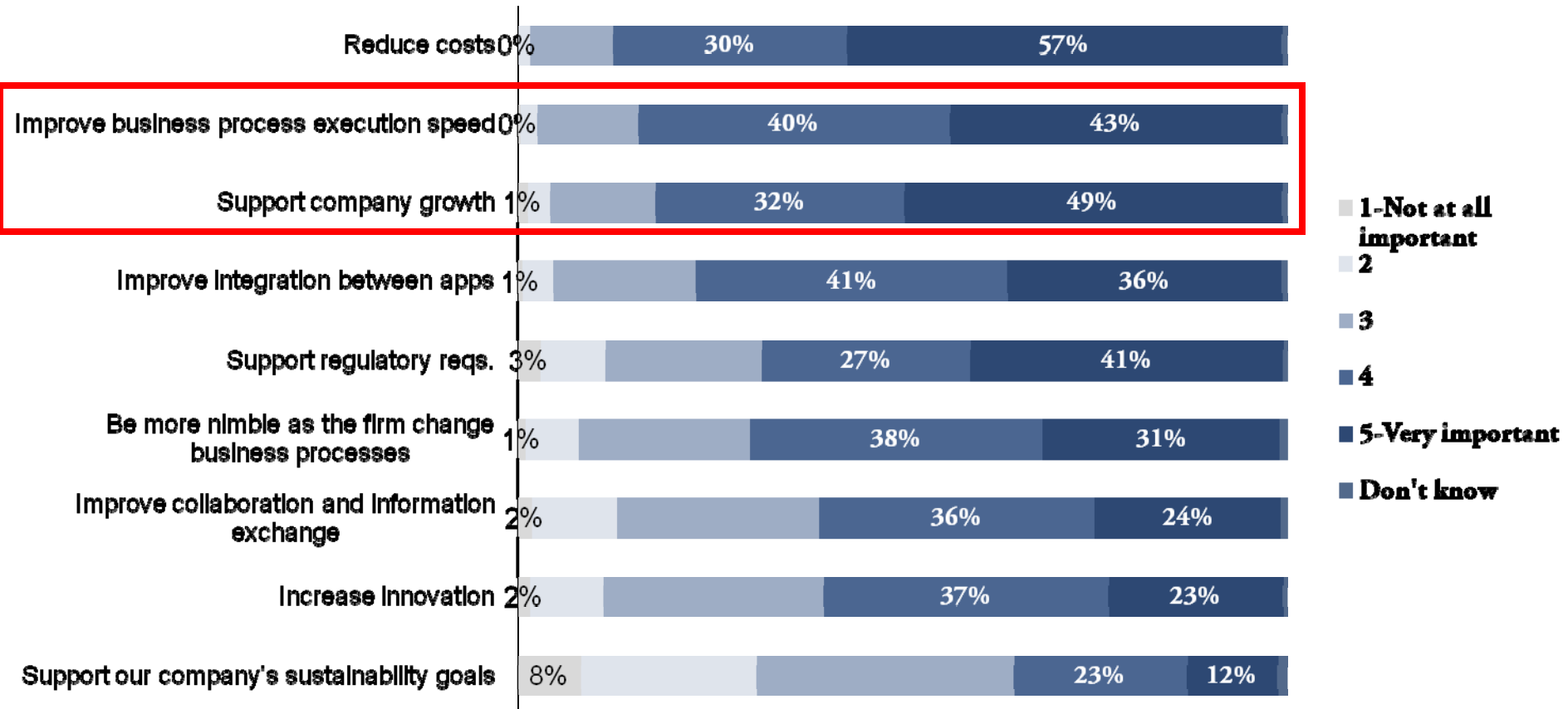


... time to market  
trumps all other  
factors.



# IT is leaning out for growth at speed in 2010

“How important are each of the following business goals to your internal IT organization when making software decisions?”



Base: 1,900 software decision-makers at North American and European enterprises and SMBs  
Source: Enterprise And SMB Software Survey, North America And Europe, Q4 2009

# Enter Lean thinking...

## Principles of Lean thinking

Add nothing but value.

Flow value from demand.

Center on the people who add value.

Optimize across organizations.

Measure and respond.

# “Lean” developer behavior we’ve observed

## Developer pain

EJB containers require heavy configuration.

WS\* specs are broad now and complex to use.

Java EE, .NET development is hard — not fun.

Projects take too long and too often miss the mark.

## Antidote to the pain

Spring, EJB 3.0 containers are easy to configure.

REST pattern is easier to create interface, straightforward to use

Dynamic languages, frameworks produce results fast -- are fun

Agile methods link developers and users to hit the mark quickly.

Source: December 12, 2008 “Lean Software Is Agile, Fit-To-Purpose, And Efficient” Forrester report

# 5 reasons you should care about Lean software

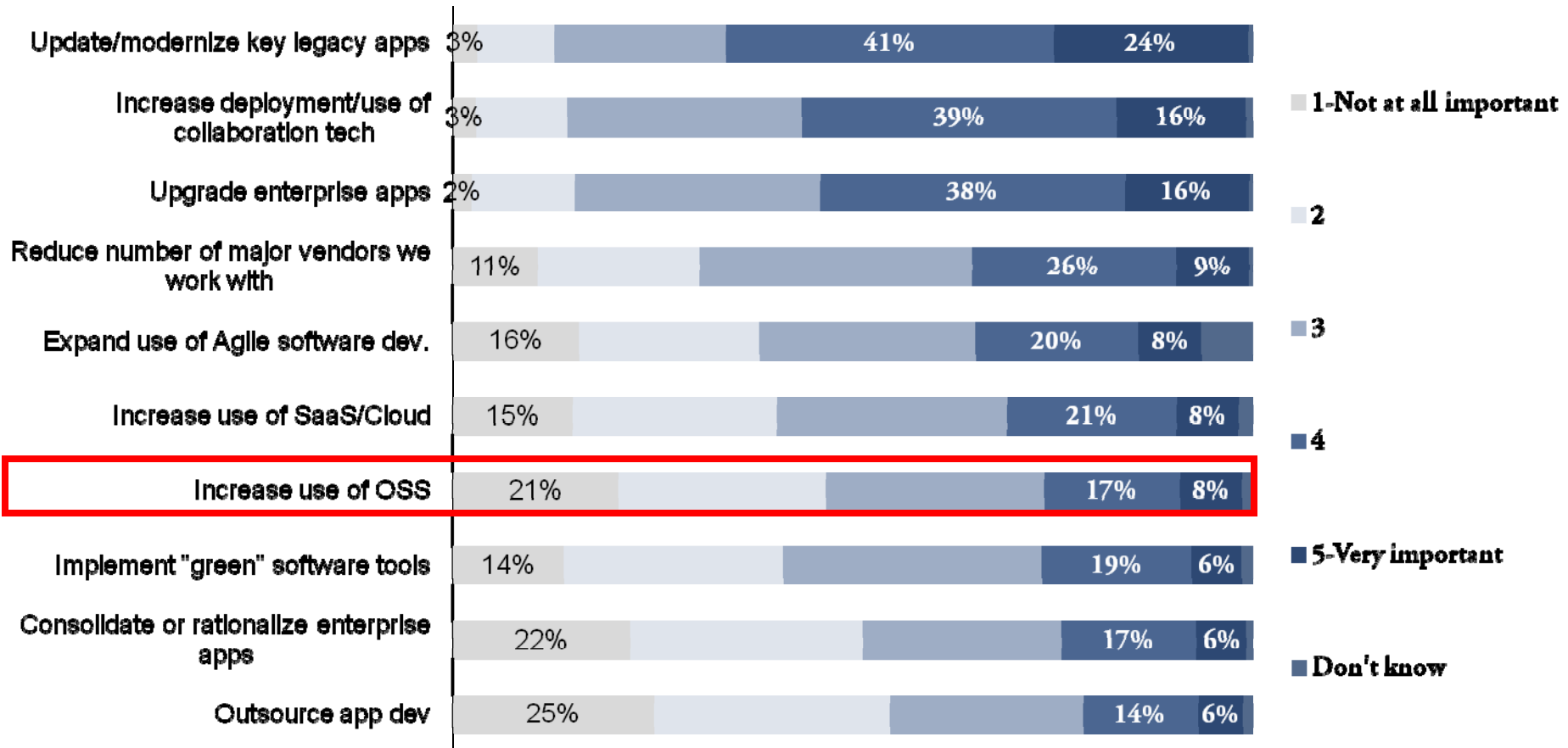
1. It's changing decision making process (and power) in IT
2. It's affecting which technologies are gaining/losing share
3. Business leaders are responding to lean software advocates
4. It's reorganizing how teams are structured and work together
5. It's reinforced by next generation technologies like Web 2.0, scale-out and cloud (and OSS)

# Agenda

- What is “Lean software” -why does it matter?
- **OSS and the lean software movement**
- Evolving to a lean, mixed-source dev model

# Urgency to adopt OSS is for it's own sake is fading...

“How important are the following software initiatives in supporting your firm's current business goals?”

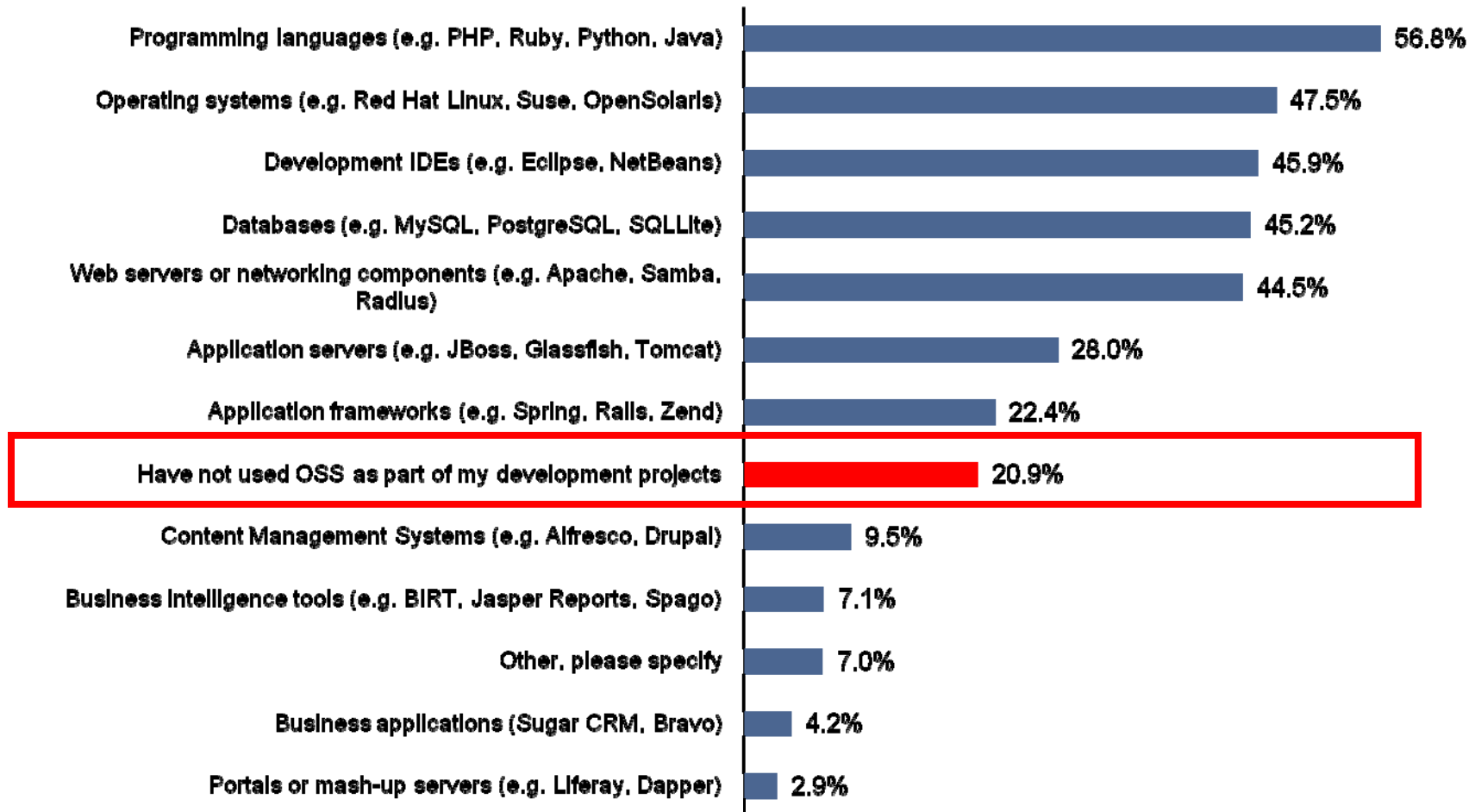


Base: 1,900 software decision-makers at North American and European enterprises and SMBs

Source: Enterprise And SMB Software Survey, North America And Europe, Q4 2009

# ...because it's now widely adopted...

“Which of the OSS infrastructure tools have you included as part of your development activities or deployed an application or software project to? “(Select all that apply.)

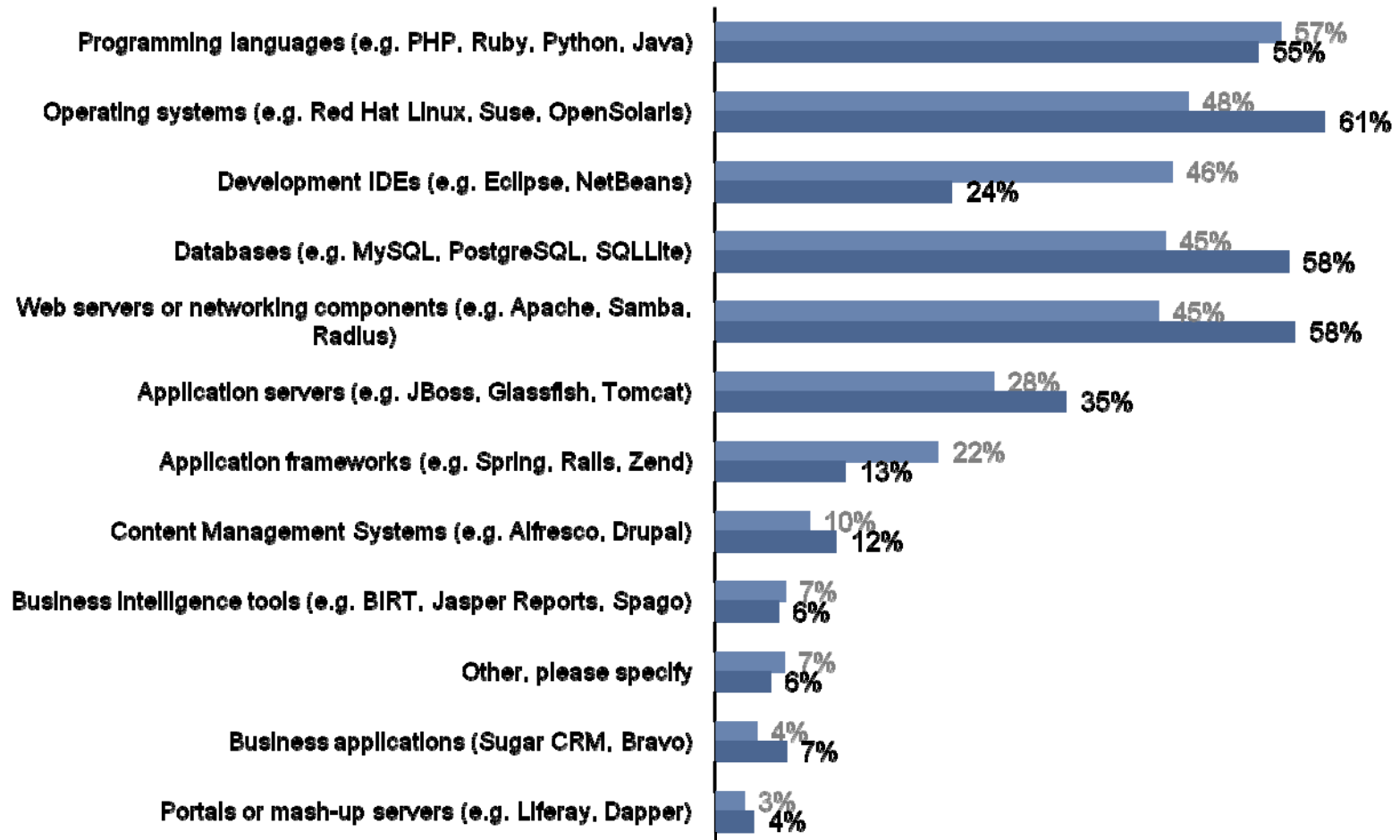


Base: 1,298 development pros at North American and European enterprises and SMBs

Source: Forrester -Dr. Dobb's 2009 Developer Technographics Survey, Q3 2009

# ... and awareness of OSS usage is growing.

“Which of the following OSS infrastructure tools does your firm currently use?”



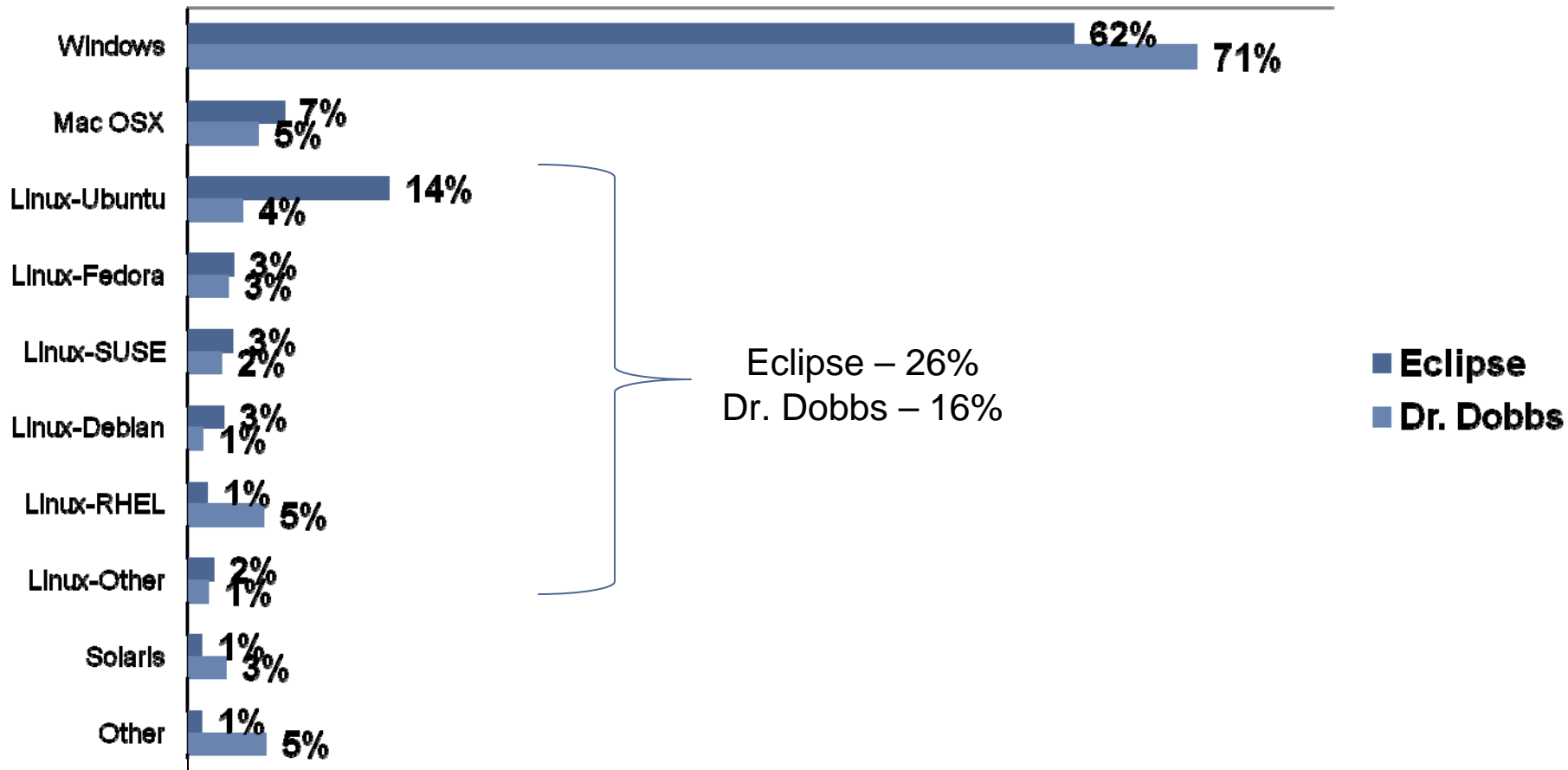
Base: 1,900 software decision-makers at North American and European enterprises and SMBs

Source: Enterprise And SMB Software Survey, North America And Europe, Q4 2009



# OSS At the OS Level - Development

“What is the primary operating system you use for development” (Choose one)



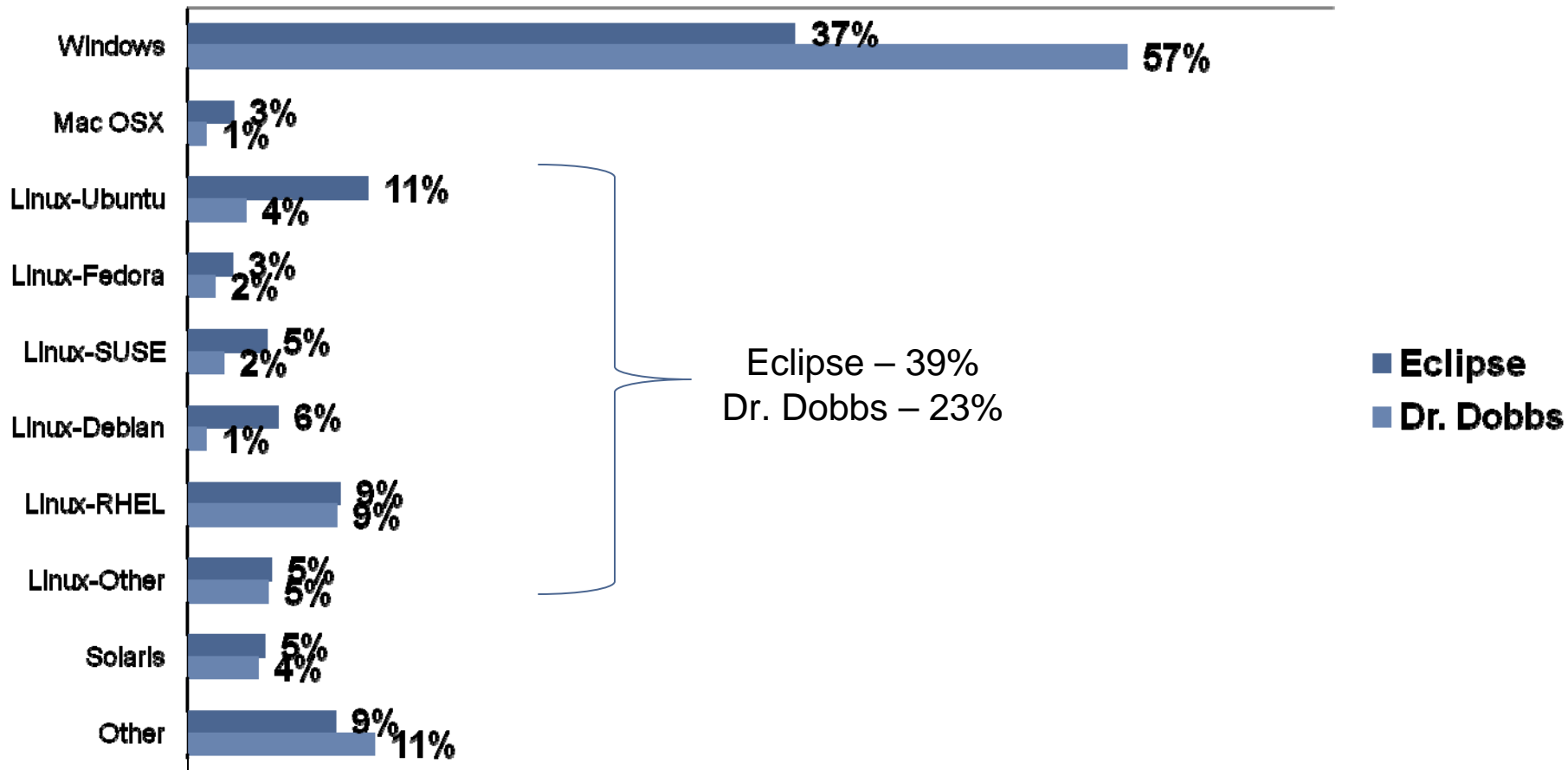
Base: 1481 application development professionals Source: 2009 Eclipse Community Survey

Base: 1298 application development professionals Forrester – Dr. Dobbs Developer Technographics

Q3 09

# OSS At the OS Level - Deployment

“What is the primary operating system you use for deployment (Choose one)”

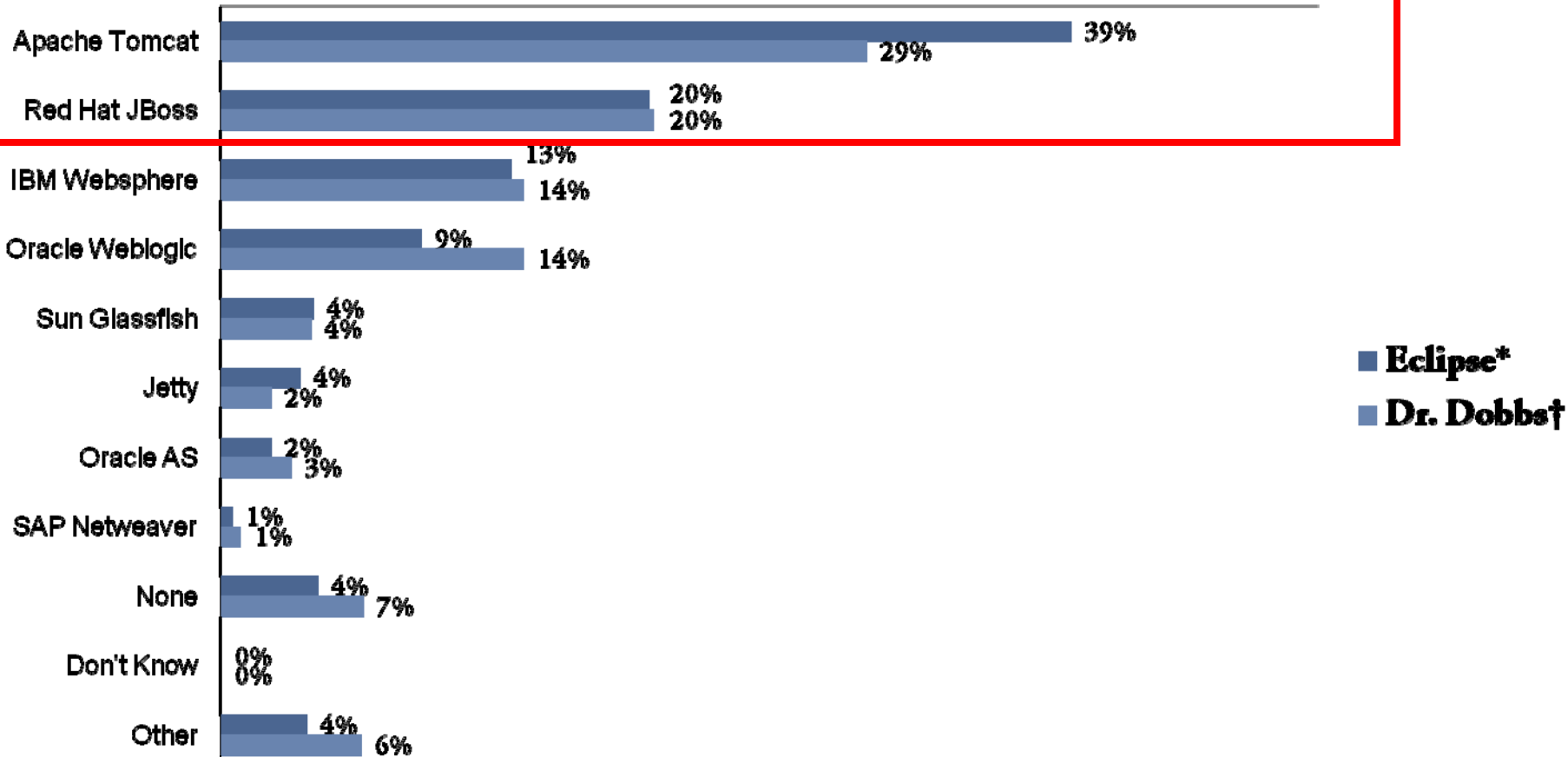


Base: 1481 application development professionals Source: 2009 Eclipse Community Survey

Base: 1298 application development professionals Forrester – Dr. Dobbs Developer Technographics Q3 09

# App server adoptions favor lean (and OSS) deployments

“What is the primary app server you typically use for deployed applications?” (Choose one.)



\*Base: 385 app dev professionals building server apps and programming in Java

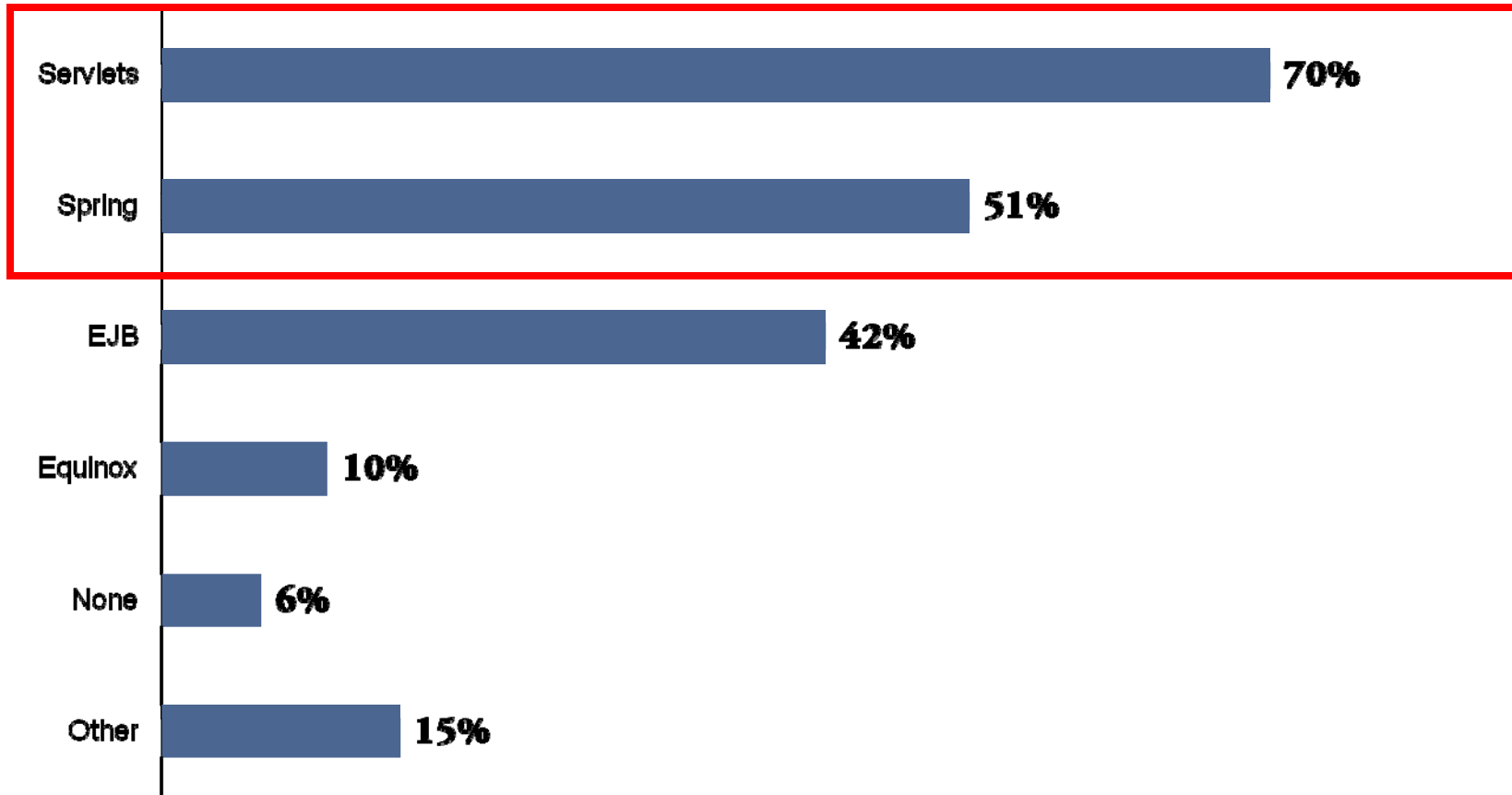
† Base: 218 app dev professionals building server apps and programming in Java at least 50% of the time

\*Source: 2009 Eclipse Community Survey

† Source: Forrester -Dr. Dobbs 2009 Developer Technographics, Q3 2009

# Framework choices favor lean frameworks

“What platforms are you using to create server-centric applications?” (Multiple choice OK.)



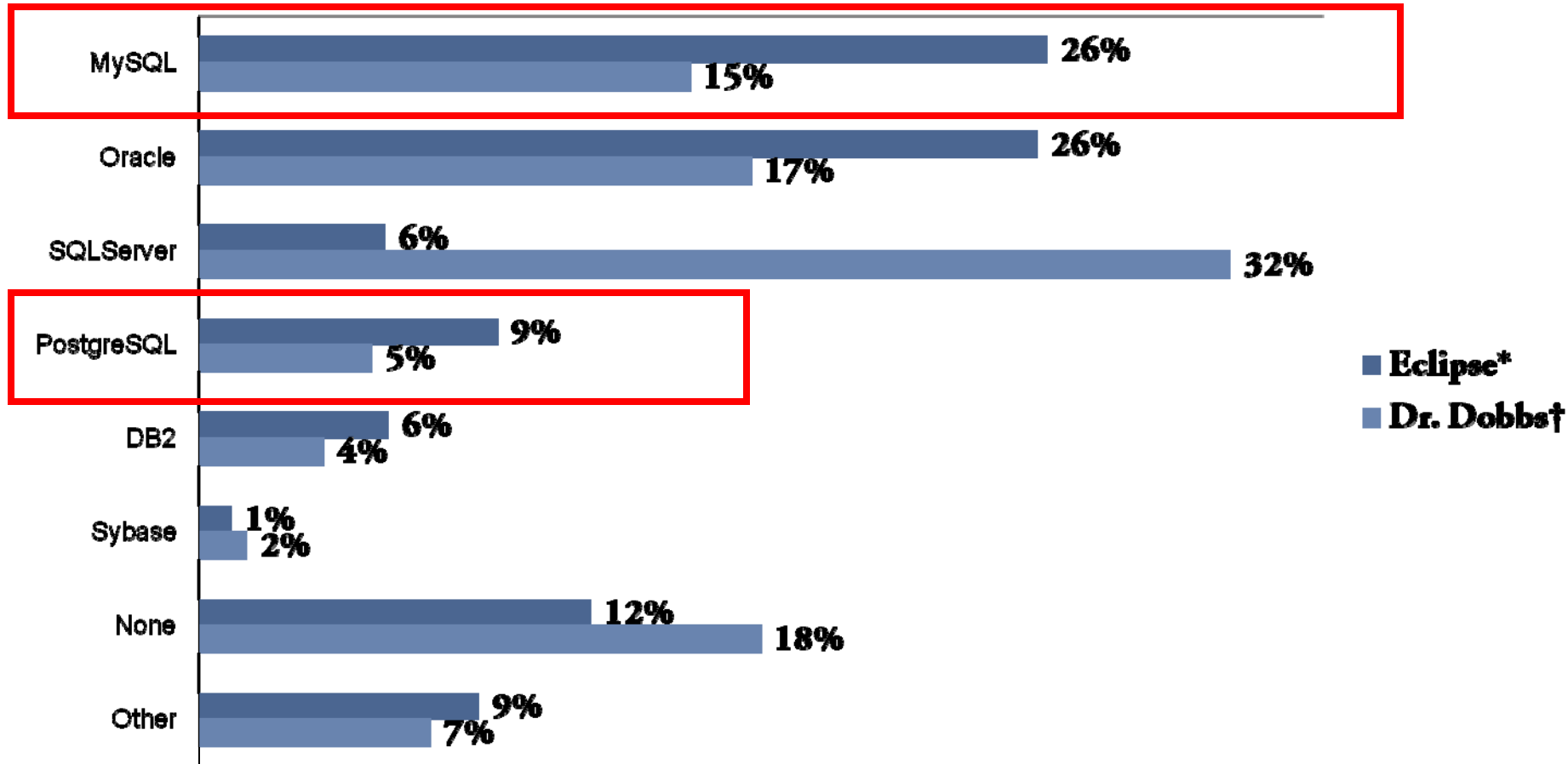
Base: 385 app dev professionals building server apps and programming in Java

Source: 2009 Eclipse Community Survey

Entire contents © 2009 Forrester Research, Inc. All rights reserved.

# OSS and DBMS use

“What is the primary app server you typically use for deployed applications?” (Choose one.)



\*Base: 1,481 application development professionals

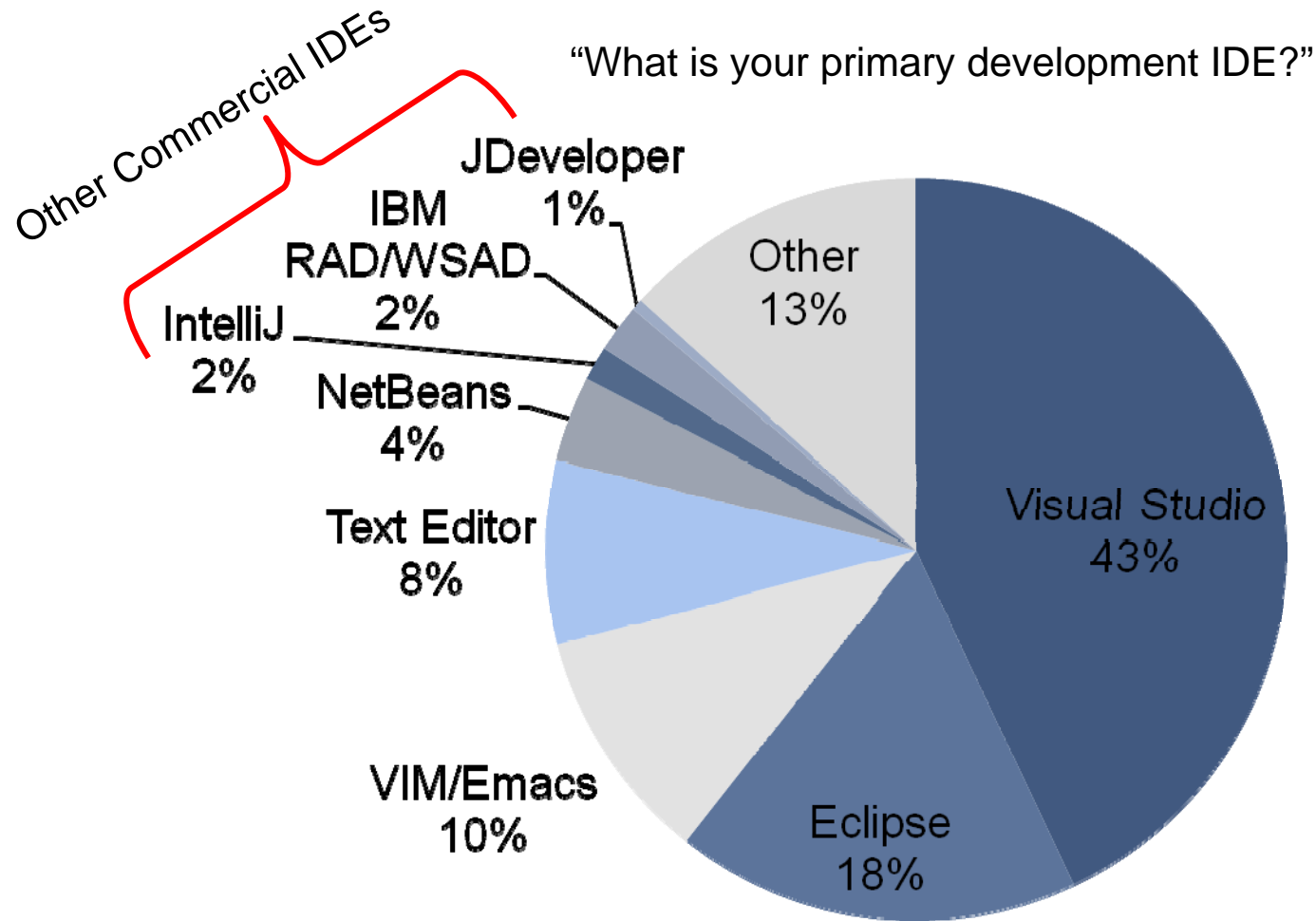
† Base: 1,298 application development professionals

\*Source: 2009 Eclipse Community Survey

† Source: Forrester-Dr. Dobb's 2009 Developer Technographics, Q3 2009

Entire contents © 2009 Forrester Research, Inc. All rights reserved.

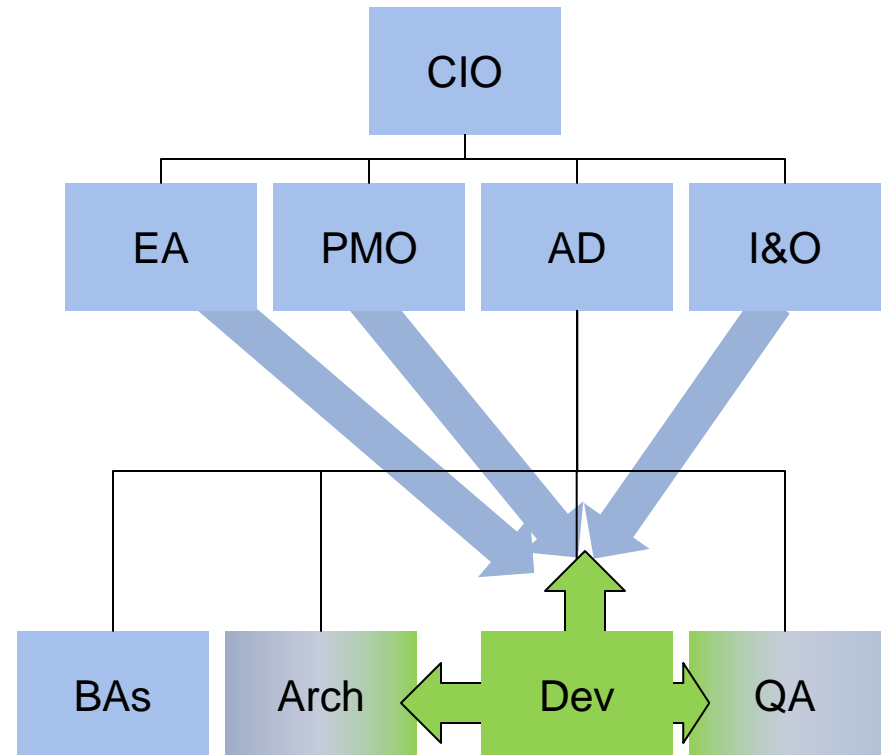
# OSS Adoption In the IDE Segment



Base: 1298 App dev professionals Forrester – Dr. Dobbs Developer

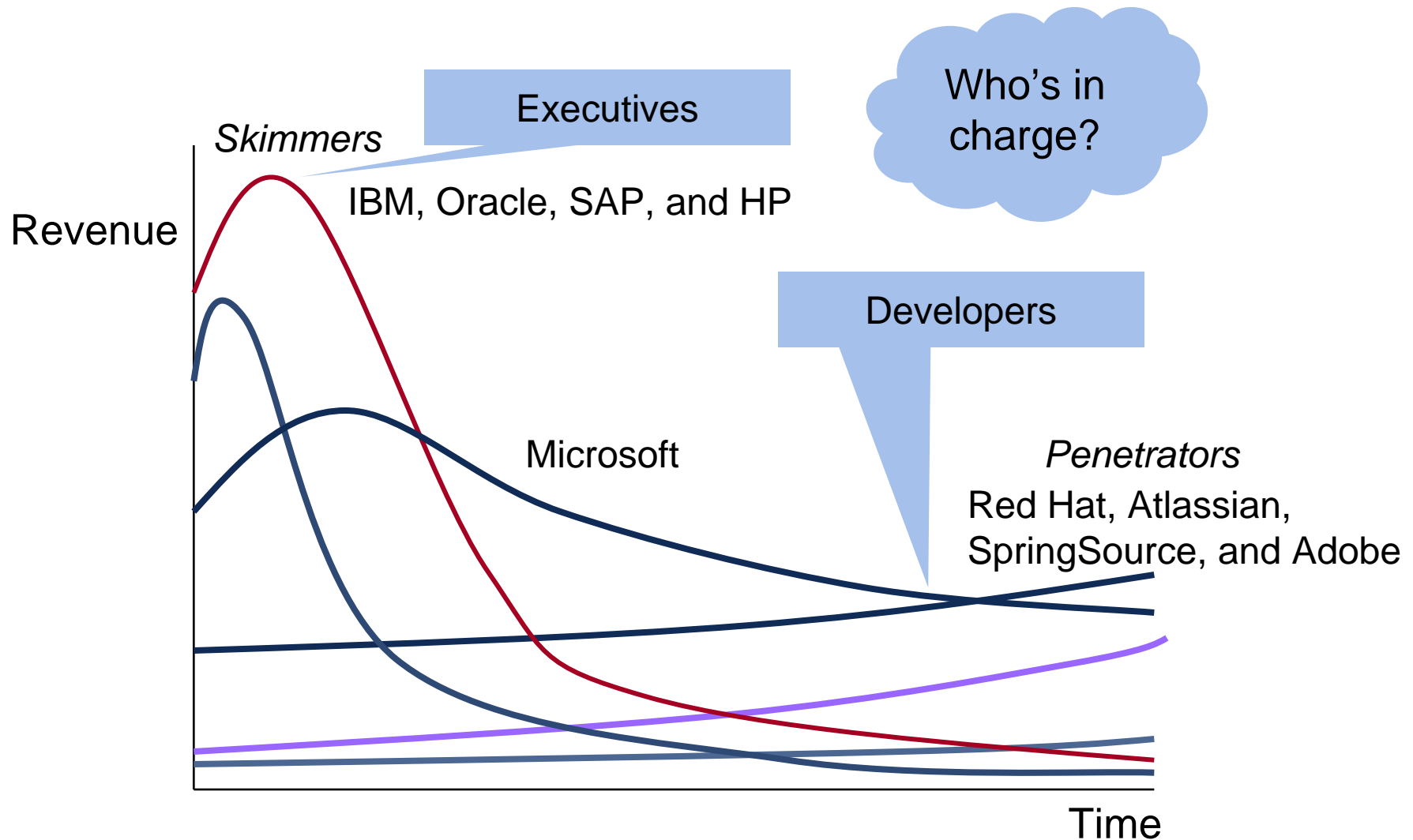
# Adoption paths are shifting toward devs

- Developers resist technology that doesn't meet their needs
- Traditional financial controls are of limited value
- LOBs defends dev teams that produce value
- Management is willing to yield when a “win-win” results
- The path from developer to customer is getting shorter
- Developer productivity is no longer the problem



More than ever: Developers can block – or significantly aid the adoption of software!

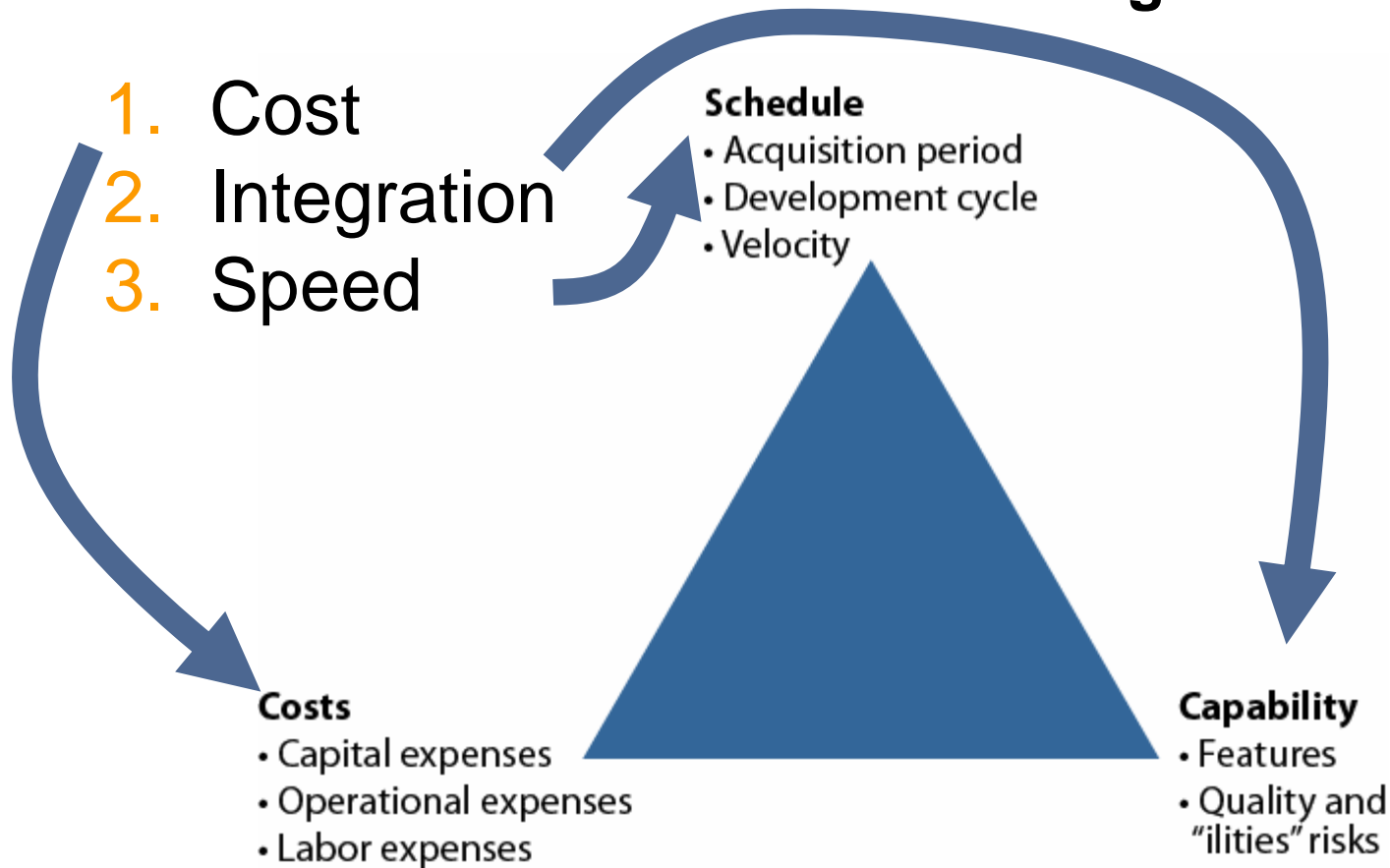
# A shift in decision makers affects adoption





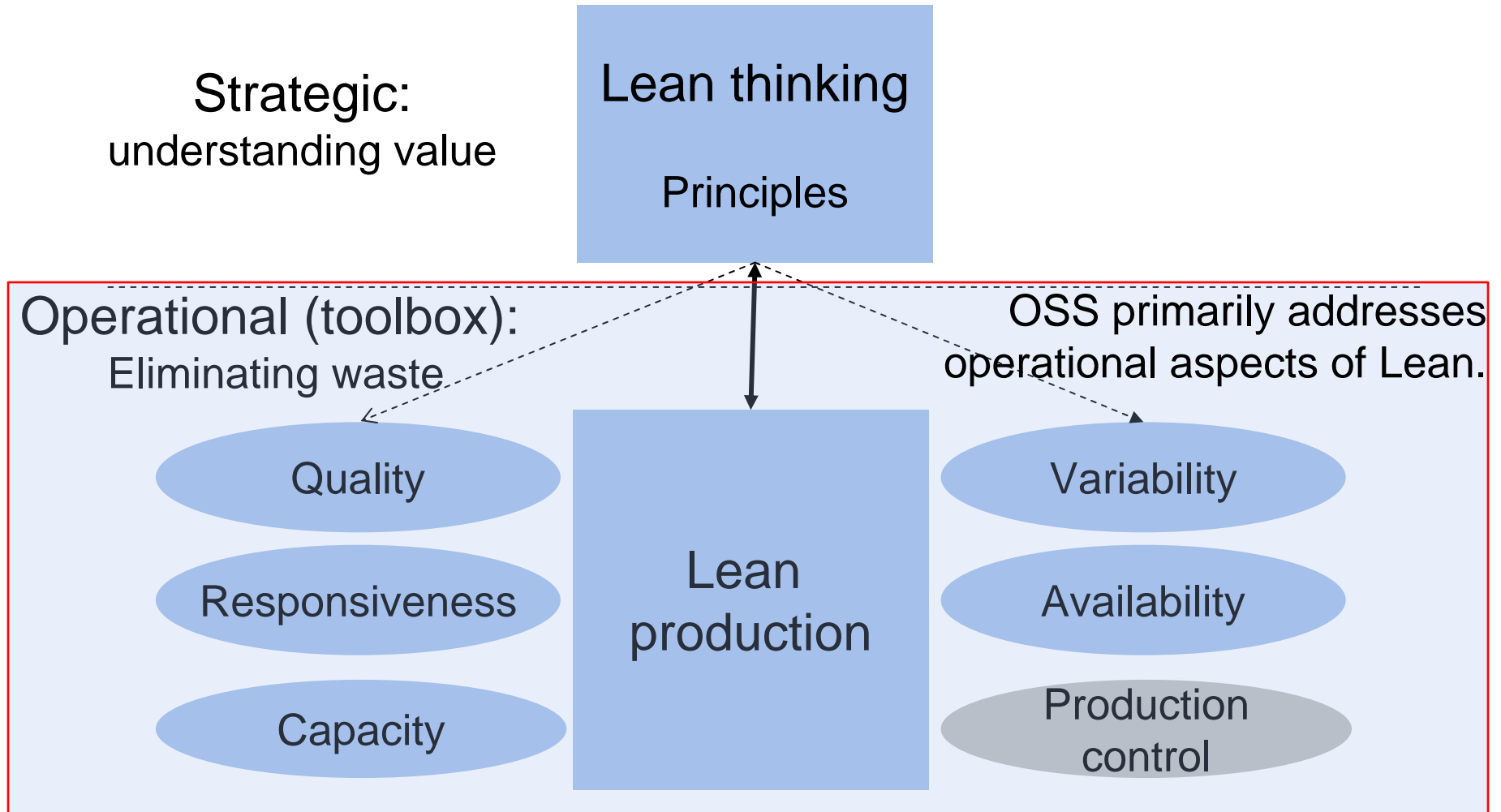
# Developers adopt OSS when it's Lean

## The software “iron triangle”



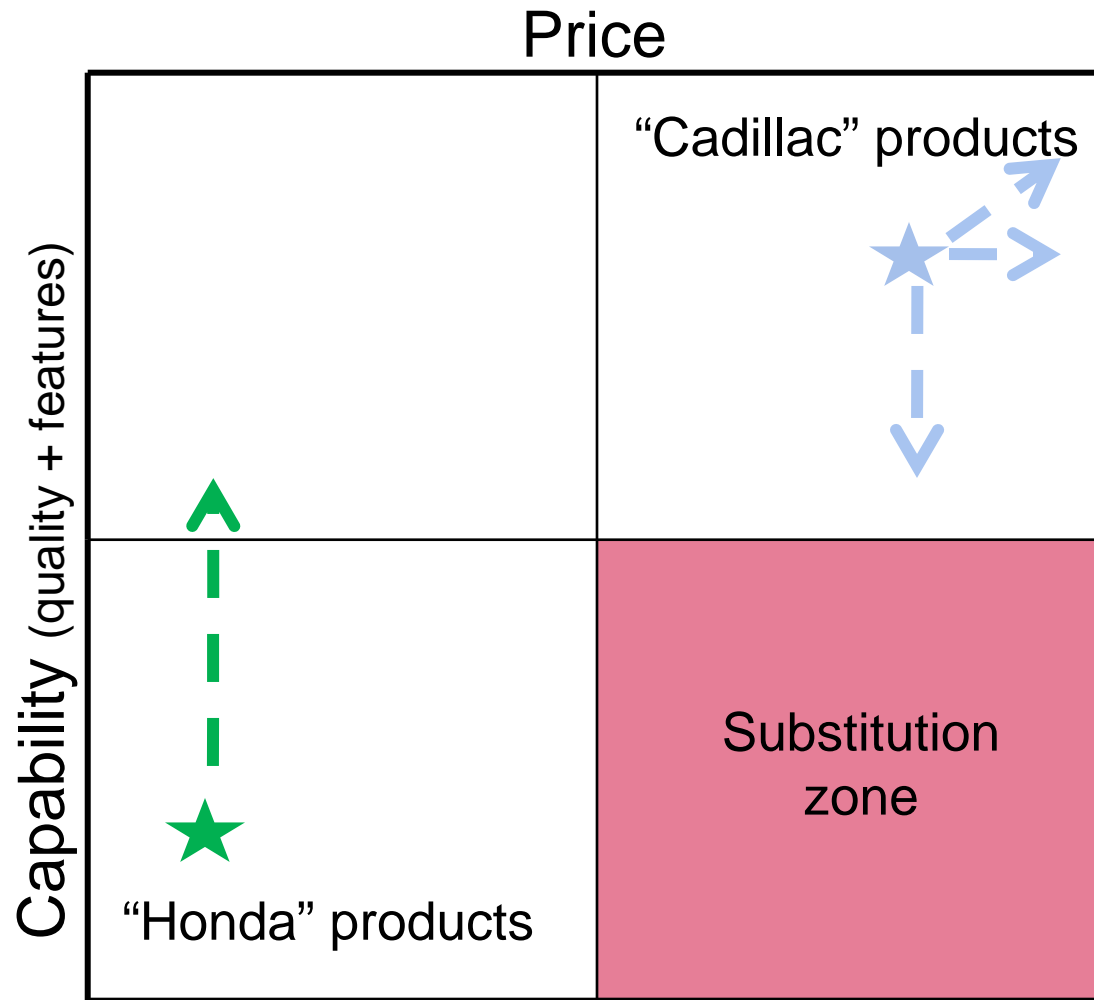
Source: February 2, 2009, “Best Practices: Improve Development Effectiveness Through Strategic Adoption Of Open Source” Forrester report

# In practice, there are two levels of Lean



Source: Peter Hines, Matthias Holweg, and Nick Rich, "Learning To Evolve: A Review Of Contemporary Lean Thinking," *International Journal of Operations & Production Management*, 2004

# “The Innovator’s Dilemma”: solve for Quality



# OSS business models are Responsive

Uh . . . Houston: We've got a problem . . .

## Traditional commercial

- I need a support number . . .
- Support is mandatory . . .
- Single source model . . .
- “We’ll get back to you on a fix ETA . . .”
- One throat to choke
- Development in the cathedral

## Open core model

- Let me search the bug DB . . .
- Support is optional . . .
- Variable support sourcing . . .
- “We temporarily patched it so we’re back up for now.”
- Whose throat to choke?
- Development in the bazaar

# OSS supports “ultra-modular” Capacity

- Ultra-modular architectures create new licensing challenges.
- Stateless servers push CPU cycles to the edge of apps.
- Forecasting demand with new business ventures is a tricky proposition.
- With commercial software, capacity has financial constraints as well as technical constraints.
- What’s the option value of tunable, transparent source code?

# **Variability** is also well-served through OSS

## 1. Variability of *demand*

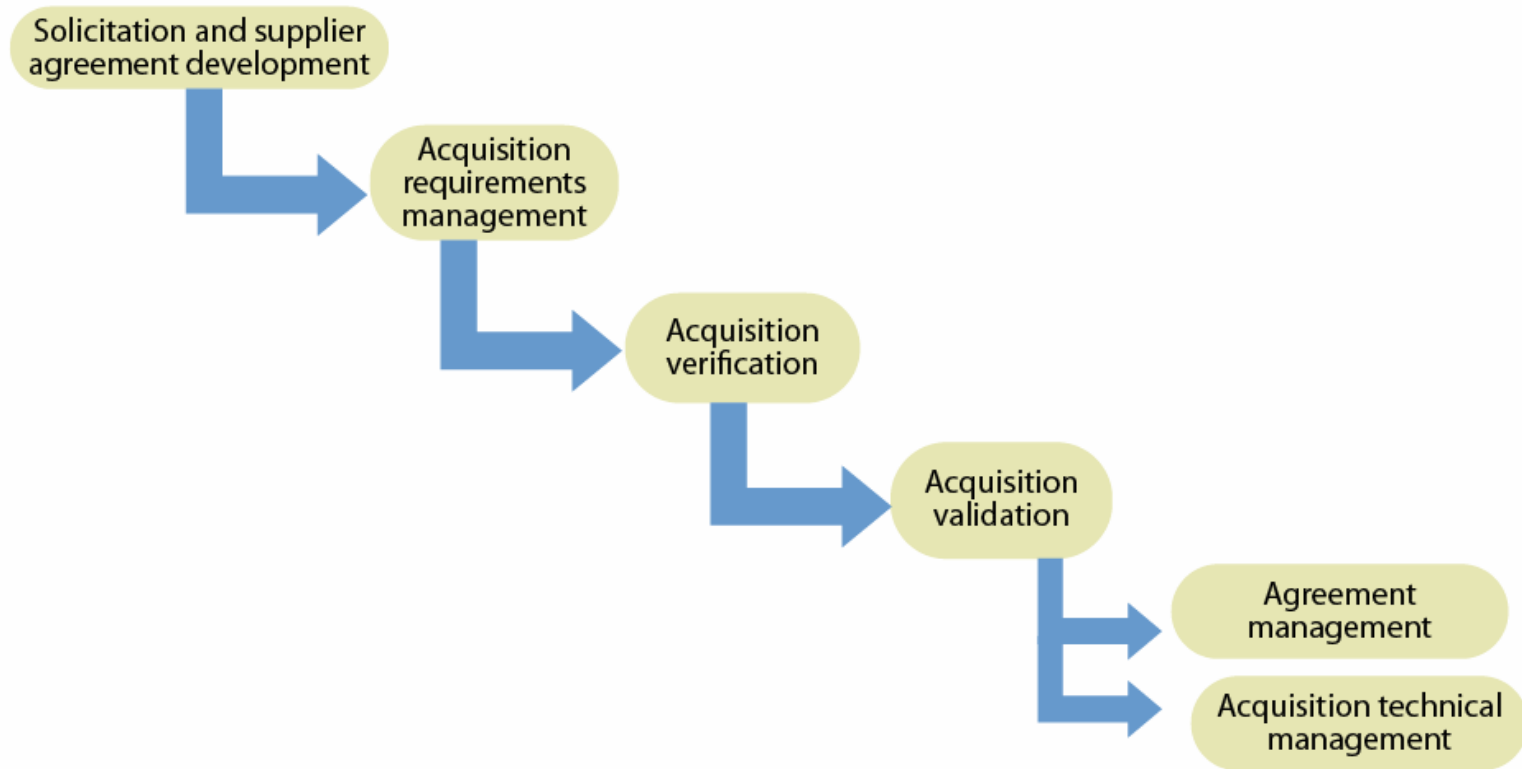
- The “cloud” scenario — scale up/scale down.
- Decoupling of support and upgrade rights

## 2. Variability of *technology*

- The early adopter scenario
- The “home brew system” integration scenario
- The virtual appliance scenario

# Acquisition processes retard Availability

## Traditional acquisition process



*This is a primary reason why OSS adoption is often developer-led!*

Source: February 2009, "Best Practices: Improve Development Effectiveness Through Strategic Adoption Of Open Source"  
Forrester report

# OSS is well suited to lean software production

- Developers will choose tools they can use to quickly prove viability of projects, prototypes
- An increase in cloud, virtualization and scale-out architectures will make OSS ever more appealing
- Supporting app integration is a strength of OSS model
- The OSS model tends to drive out waste
- World class shops will embrace self-support and contributions to key OSS projects
- Over time Hondas sometimes become Acuras



# Agenda

- What is “Lean software” -why does it matter?
- OSS and the lean software movement
- **Evolving to a lean, mixed-source dev model**

# Looking to manufacturing as an inspiration

Software production processes can be rich in waste . . .

## Manufacturing sources of waste

1. Overproduction
2. Waiting (time on hand)
3. Unnecessary transport or conveyance
4. Overprocessing or incorrect processing
5. Excess inventory
6. Unnecessary movement
7. Defects
8. Unused employee creativity

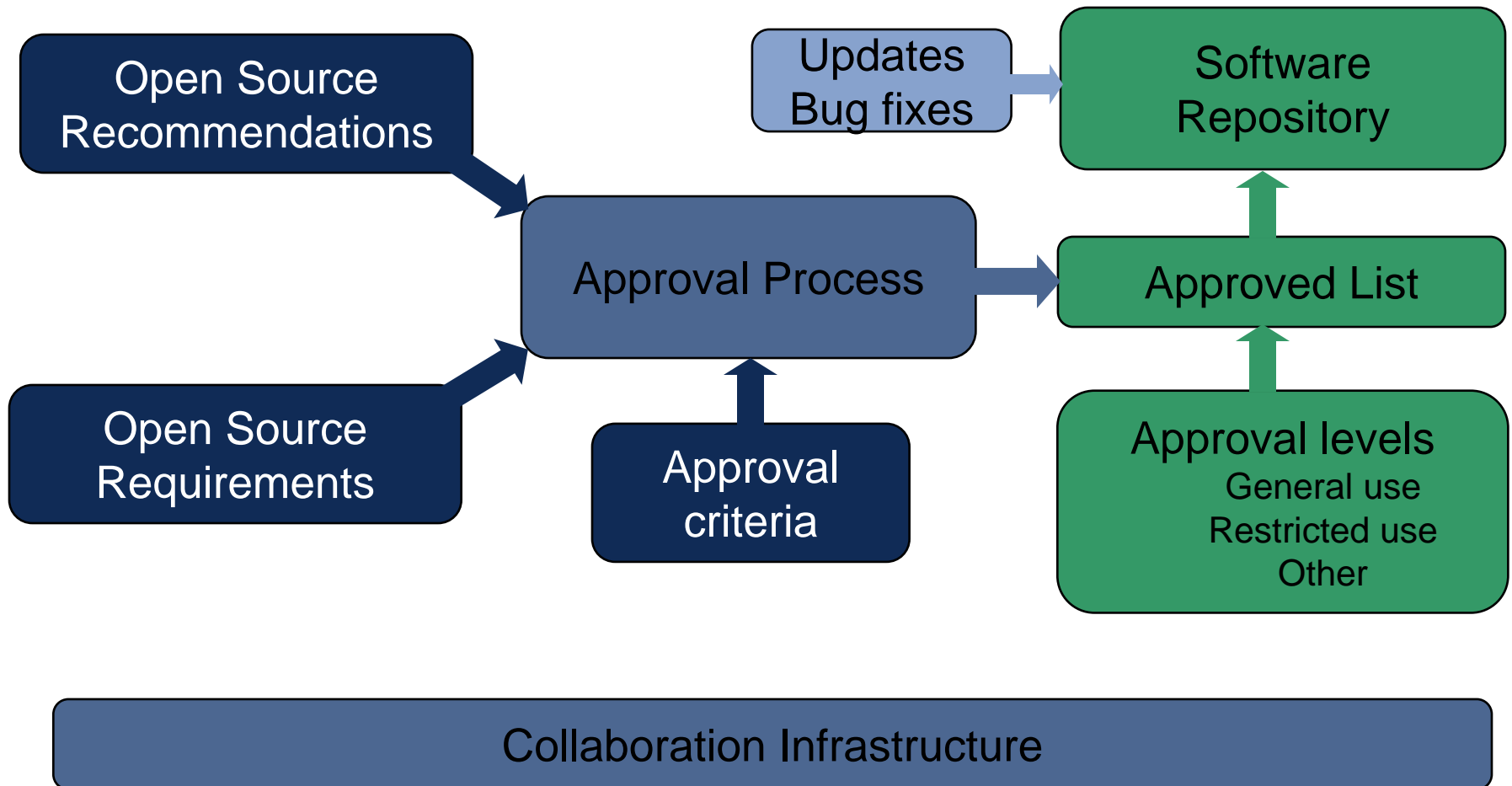
## App dev equivalents

- Too many superfluous artifacts
- Broken builds
- Too many tool transitions
- Rigid architectures
- Analysis paralysis
- Late discovery of defects
- Rising downstream labor costs
- Polluted SCM streams
- High null-release ratios
- Measures of effort, not results

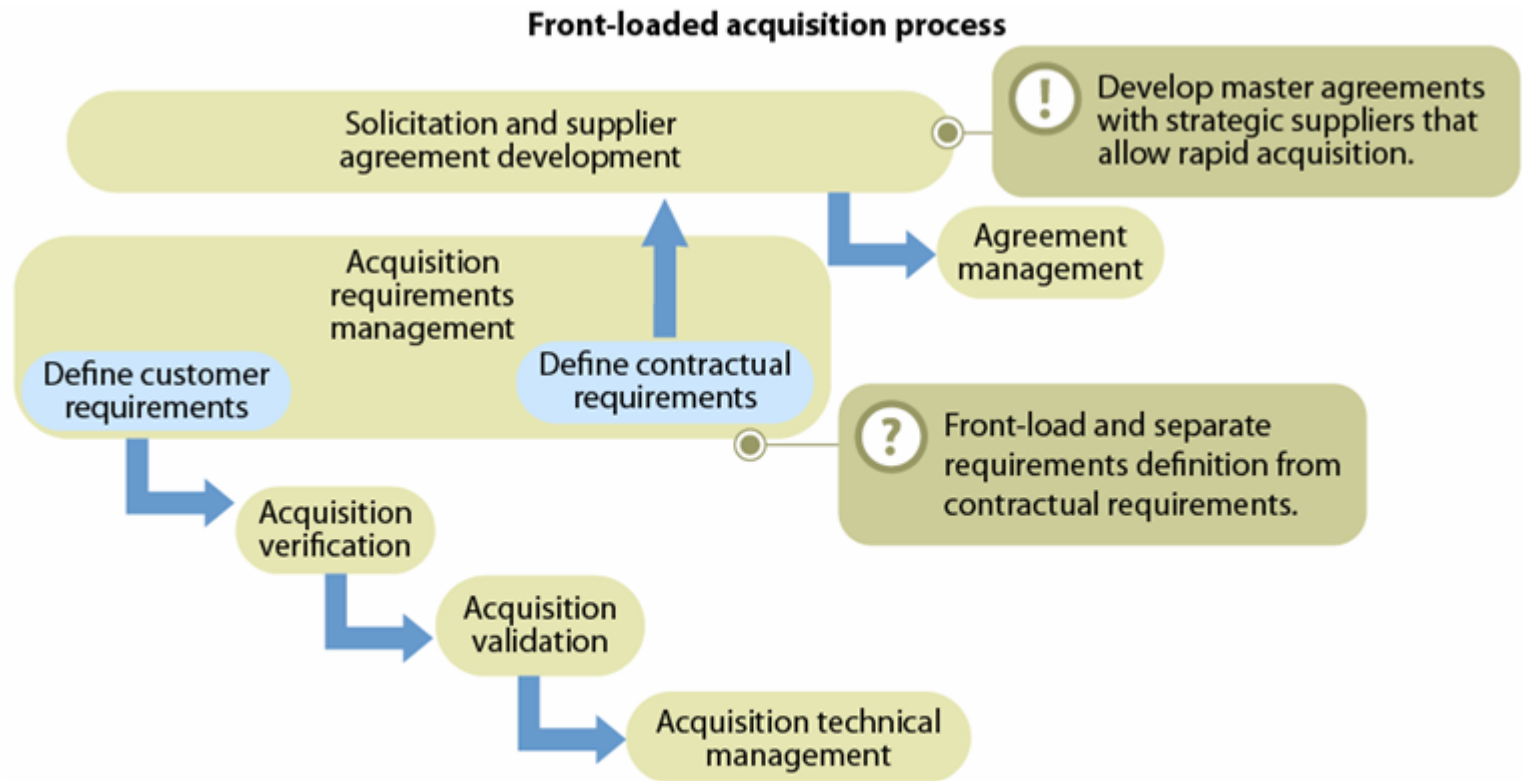
# Governing lean, mixed-source development

- Appoint an OSS steward
- Create a comprehensible policy
- Frontload acquisition processes
- Require project leaders to identify OSS dependencies
- Use architects to regulate exploitation and maintenance
- Trust teams - but verify with code-scanning utilities
- Maintain a repository of preapproved OSS components
- Focus on automating two control loops
- Focus on outcomes not dev processes and artifacts;
- Don't expect perfection, and plan for remediation

# Implementing an appropriate use policy for OSS



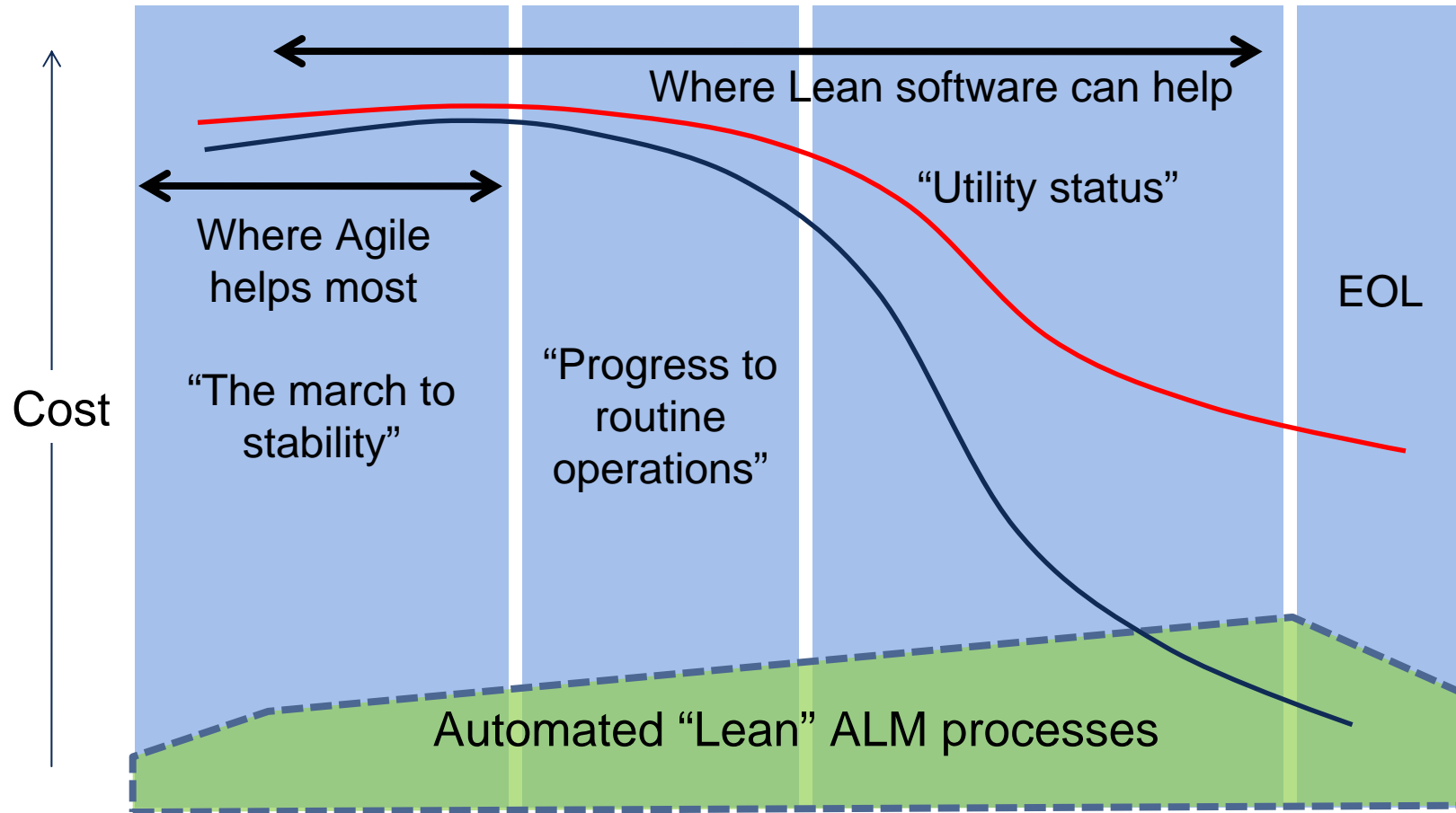
# Front-loaded processes improve availability



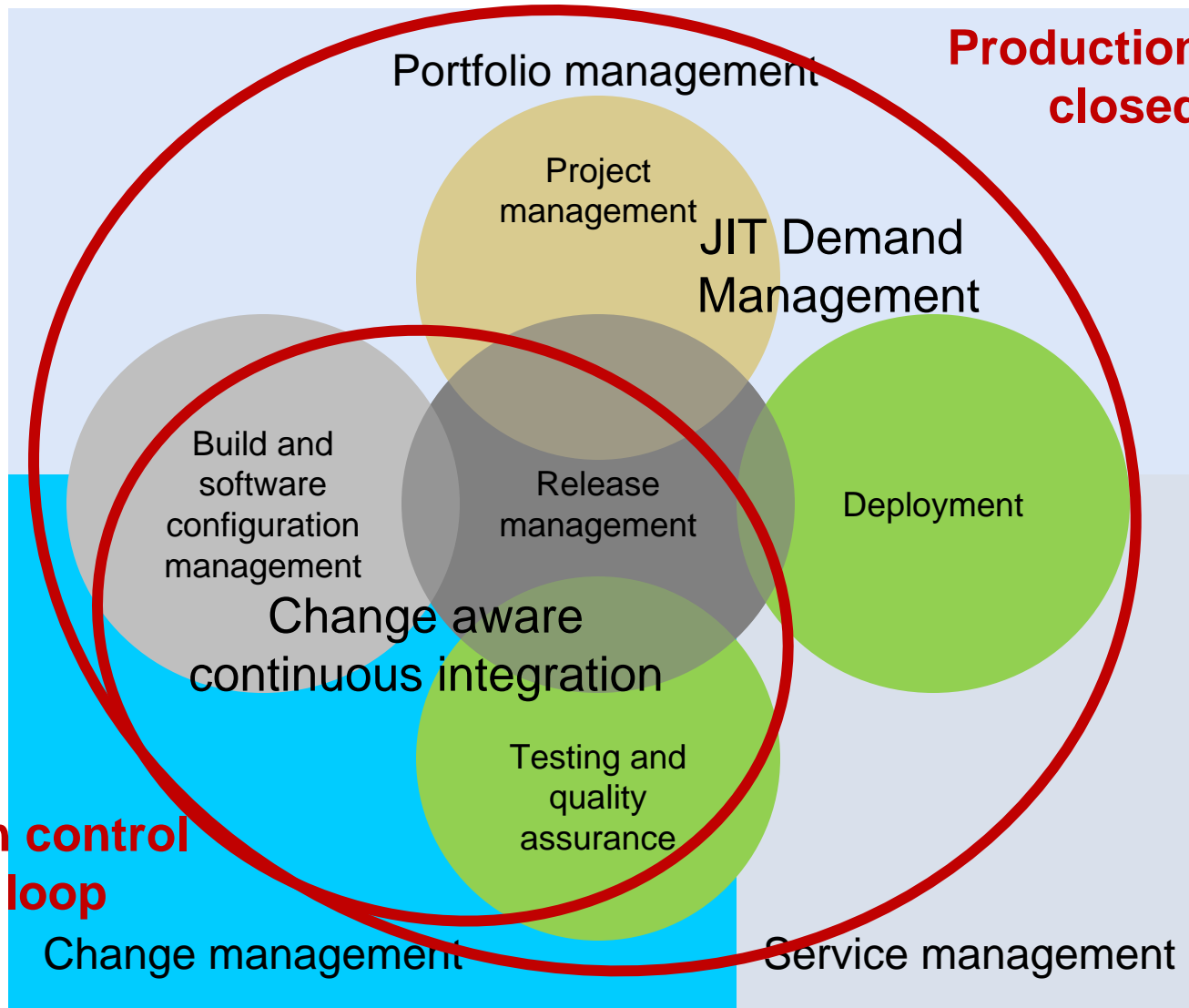
Source: February 2009, "Best Practices: Improve Development Effectiveness Through Strategic Adoption Of Open Source"  
Forrester report

# Consider a mixed process model

Cost of an application over time

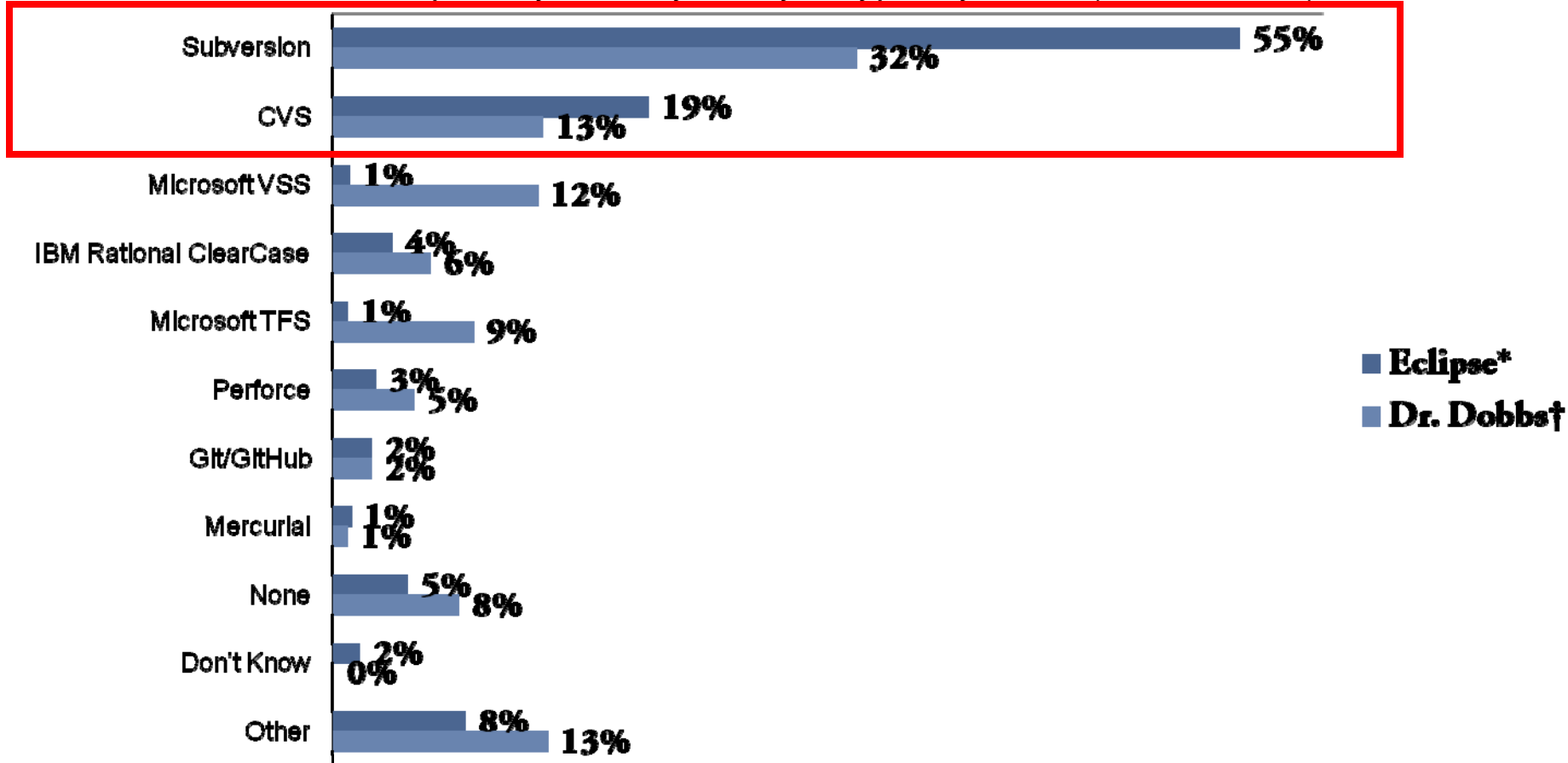


# Focus on automating two control loops



# OSS in the production control loop

“What is the primary SCM system you typically use?” (Choose one.)



\*Base: 1,481 application development professionals

† Base: 1,298 application development professionals

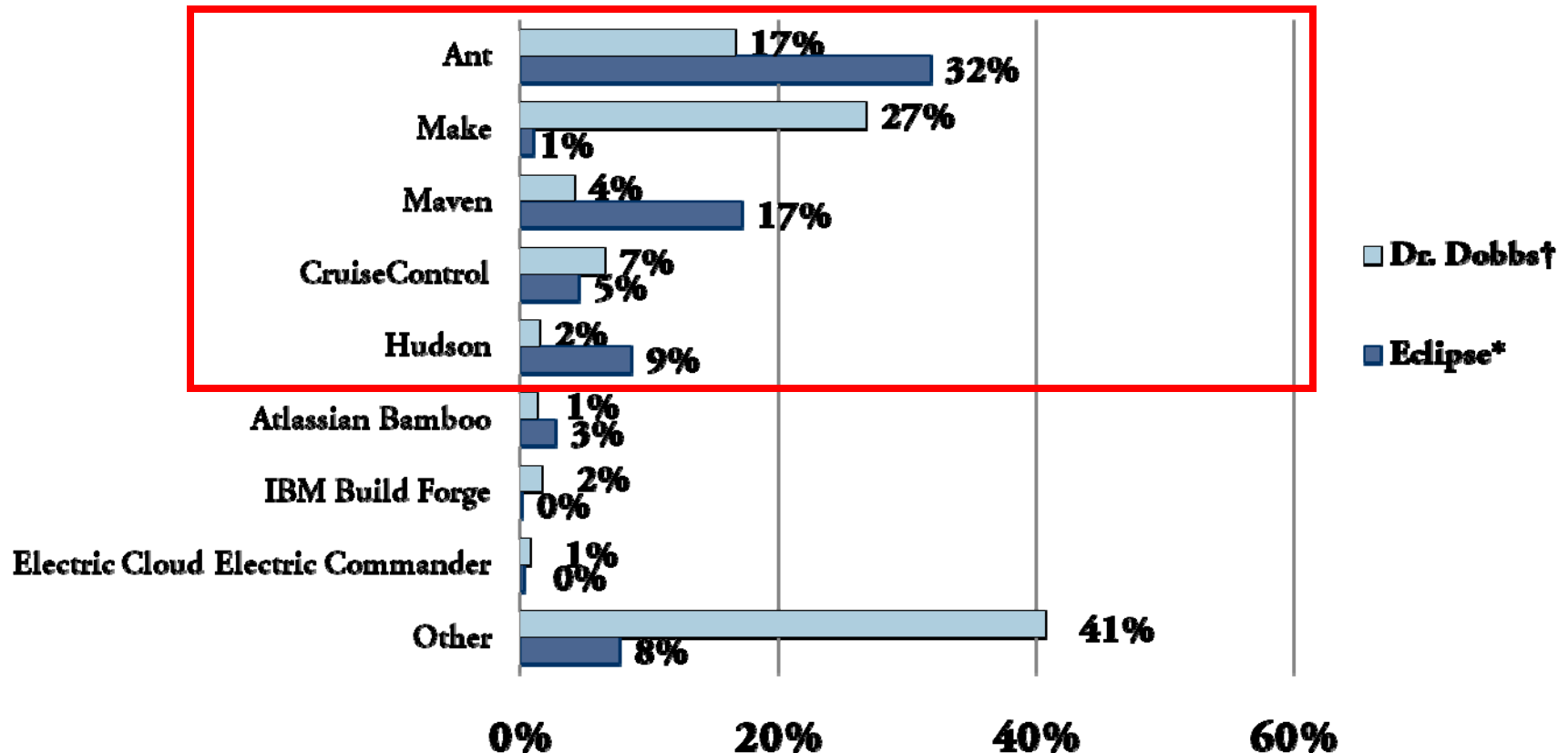
\*Source: 2009 Eclipse Community Survey

† Source: Forrester -Dr. Dobb's 2009 Developer Technographics, Q3 2009



# OSS in the production control loop

“What is the primary build and release management product you typically use?”



\*Base: 1,481 application development professionals

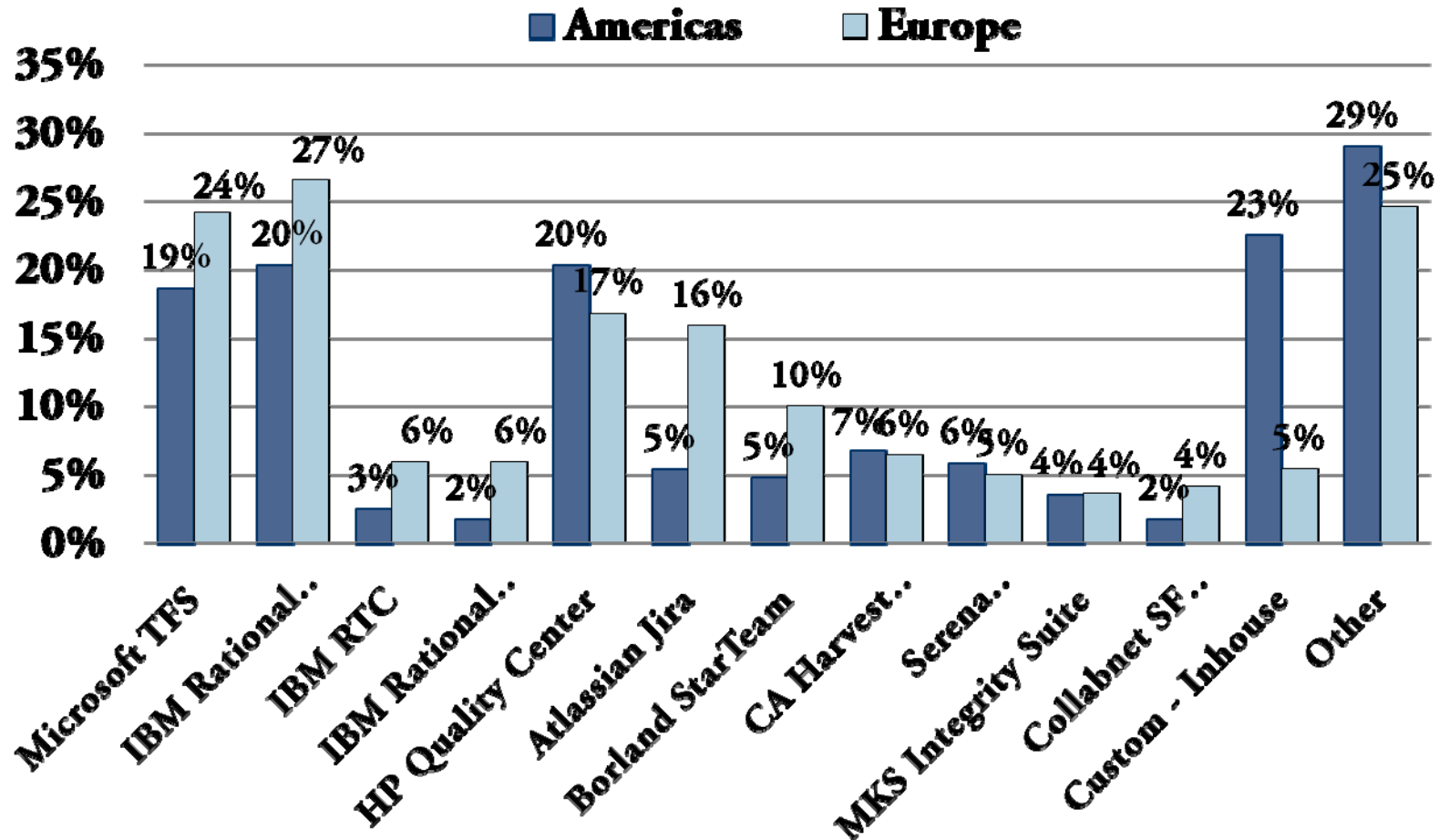
† Base: 1,298 application development professionals

\*Source: 2009 Eclipse Community Survey

† Source: Forrester -Dr. Dobbs's 2009 Developer Technographics, Q3 2009

# Where OSS is still nascent

“What CM tools are commonly used in you application development organization?”



Base: 211 European application development professionals

Base: 479 Americas application development professionals

Source: Q4 2008 European Application Life-Cycle Management Usage And Trends Online Survey

Source: Q1 2009 Americas Application Life-Cycle Management Usages And Trends Online Survey

# Add up your waste reduction opportunities

**Example of a simple three-year formula for annualized total cost of ownership (TCO)\***

$$\text{TCO} = (\text{capital expenses} + 3(\text{operational expenses}))/3$$

**Capital expenses =**  
***hardware acquisition + software license acquisition***

CAPEX considerations:

- Can commodity hardware suffice, or is specialized hardware required?
- Are license costs a lump sum, or do they increase as deployment scales?
- Are software licenses perpetual or fixed-term?

**Operational expenses =**  
***maintenance + power + labor + indirect costs***

OPEX considerations:

- Are recurring costs optional or mandatory: How much flexibility is there?
- Labor costs include developer and administrator salaries and training fully loaded costs.
- Indirect costs include vendor management, software configuration and administration tools, and the cost of migrating to new platforms.

\*For greater precision, use net present value (NPV) calculations for OPEX and break out indirect costs on a separate, annualized basis.

Source: February 2009, "Best Practices: Improve Development Effectiveness Through Strategic Adoption Of Open Source"  
Forrester report

# Case study: leading financial services firm

Provides asset management, investment banking, private banking, treasury and securities, and commercial banking

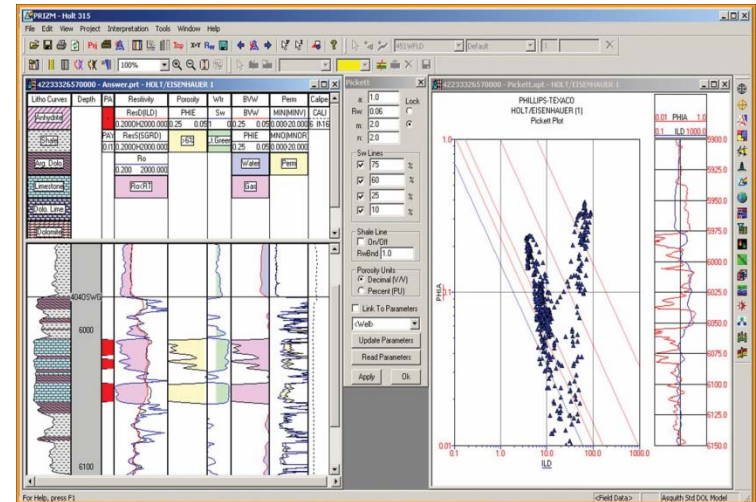
- Using Linux since 2001, OSS is now moving up the stack.
- Using OSS app servers keeps big ISVs honest.
- Needs good developers — not average ones
- Used OSS as an opportunity to refashion dev processes
- Now looking at BI, using a lot of Spring
- Needs to manage the support “fear factor”
- Dev community is very happy; I&O is mixed.

Result: Per project software costs savings range from 30% to 80%. For every \$10 put in, they get \$4 back.

# Case Study: Landmark Graphics

Landmark Graphics supplies software to Oil and Gas industry across a broad variety of application areas

- OSS Steward monitors policy compliance
- Prioritize standardization
- Restructured release process
  - Use BDS to monitor compliance
  - PM assumes responsibility for OSS
  - Remediate if/as violations are found
- Contributing back in limited cases



Result: Rapid adoption of the latest models and technologies, with accurate identification of OSS dependencies

# Case Study: Reliant Security

Reliant sells PCI compliant in-store systems that include many OSS subsystems.

- Set a clear policy for OSS use
- Tuned acquisition policies
  - OSS first mandate
  - Prioritized “ilities”
  - Loosely coupled design
- Adjusted dev processes
  - OSS use identified at design
  - Developer on the hook for provenance



**Result: Significant customer savings over commercial alternatives**



# If you remember nothing else . . .

1

The characteristics of many OSS projects are representative of lean production.

2

Adopting OSS will require adjustments to process, tools and people.

3

Focus on eliminating sources of waste in your software production processes.

# Thank you

Jeffrey S. Hammond

+1 617.613.6164

jhammond@forrester.com

Twitter: jhammond

[www.forrester.com](http://www.forrester.com)