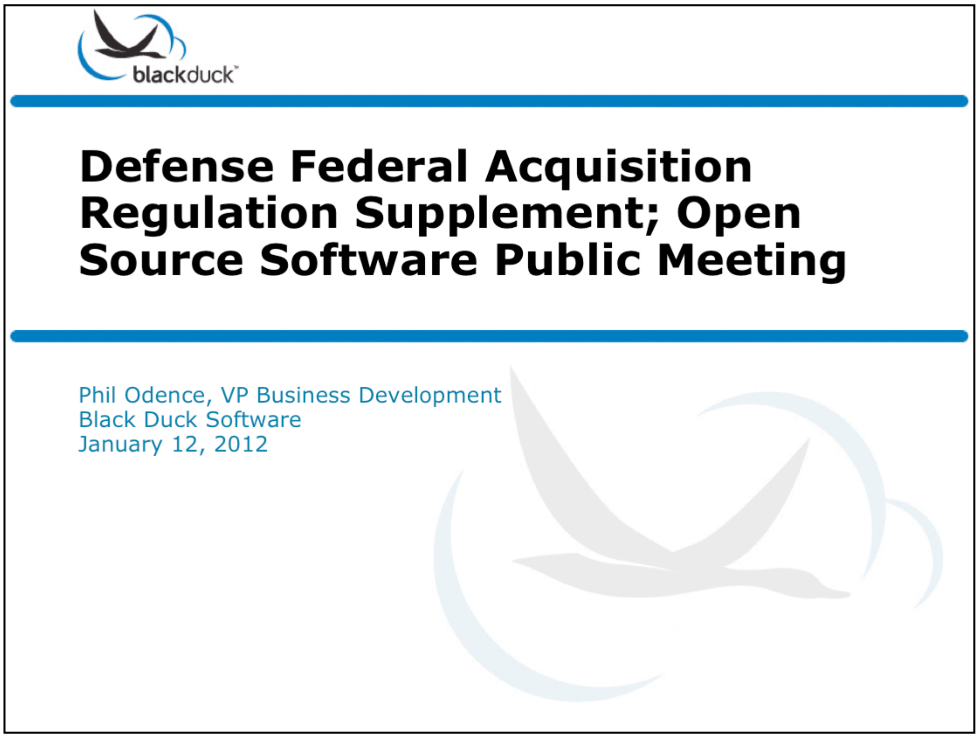
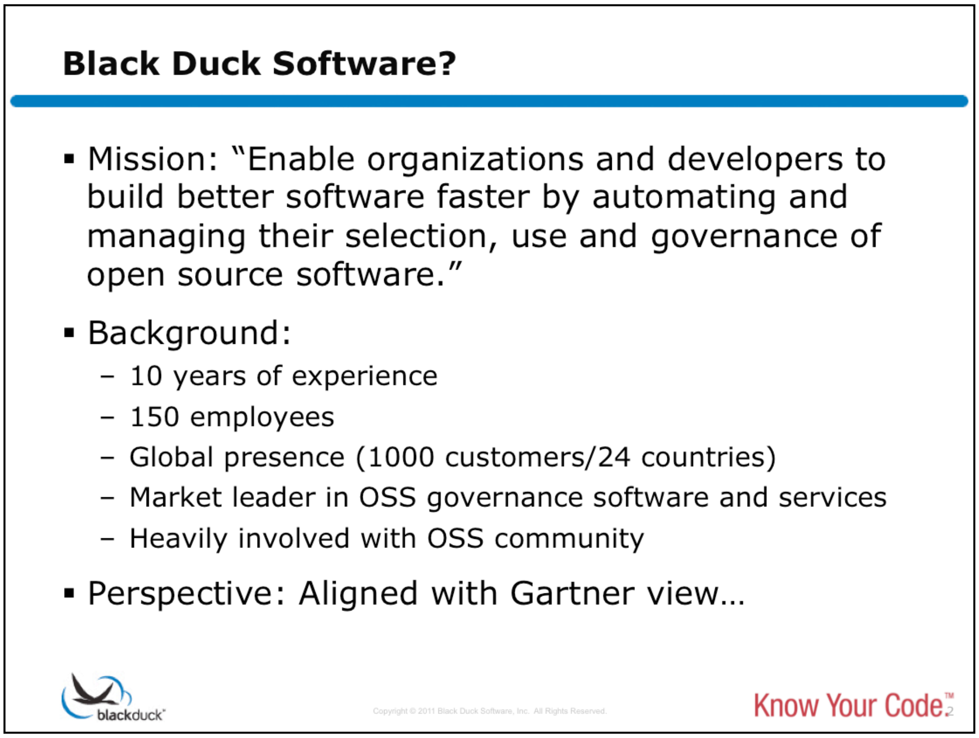


Defense Federal Acquisition Regulation Supplement; Open Source Software Public Meeting

Phil Odence, VP Business Development
Black Duck Software
January 12, 2012

Contact Information:

L. Philip Odence
Vice President of Business Development
Black Duck Software, Inc.
8 New England Executive Park, Suite 211, Burlington MA 01803
Phone: 781.810.1819, Mobile: 781.258.9502
Skype: philip.odence
podence@blackducksoftware.com
http://www.blackducksoftware.com
http://twitter.com/podence
http://www.linkedin.com/in/podence
http://www.networkworld.com/community/odence (my blog)



Black Duck Software?

- Mission: “Enable organizations and developers to build better software faster by automating and managing their selection, use and governance of open source software.”
- Background:
  - 10 years of experience
  - 150 employees
  - Global presence (1000 customers/24 countries)
  - Market leader in OSS governance software and services
  - Heavily involved with OSS community
- Perspective: Aligned with Gartner view…

Know Your Code™ 2

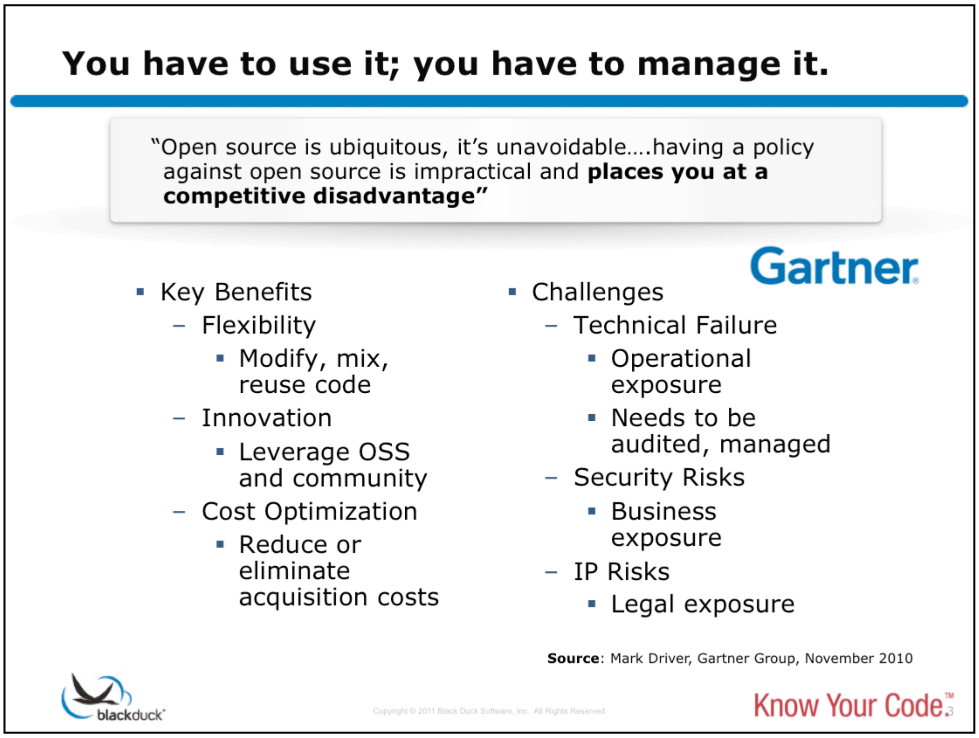
Quick background for context:

Black Duck provides products and services that help development organizations gain the benefits of open source while managing the risks.

The company has been in business for about 10 years. We are growing at about 35% and now approximately 150 employees. Headquarters are in Burlington, MA, but we have employees across the US and in Europe and Asia. We’ve done business with about 1000 organizations in 24 countries.

By any measure we lead the market for the types of products and services we offer and certainly have more experience than any company in helping organizations with OSS governance.

From this experience we have developed a view on the benefits and risks of using open source components in development.



You have to use it; you have to manage it.

> "Open source is ubiquitous, it's unavoidable….having a policy against open source is impractical and **places you at a competitive disadvantage"**

Gartner

- Key Benefits
  - Flexibility
    - Modify, mix, reuse code
  - Innovation
    - Leverage OSS and community
  - Cost Optimization
    - Reduce or eliminate acquisition costs

- Challenges
  - Technical Failure
    - Operational exposure
    - Needs to be audited, managed
  - Security Risks
    - Business exposure
  - IP Risks
    - Legal exposure

Source: Mark Driver, Gartner Group, November 2010

blackduck

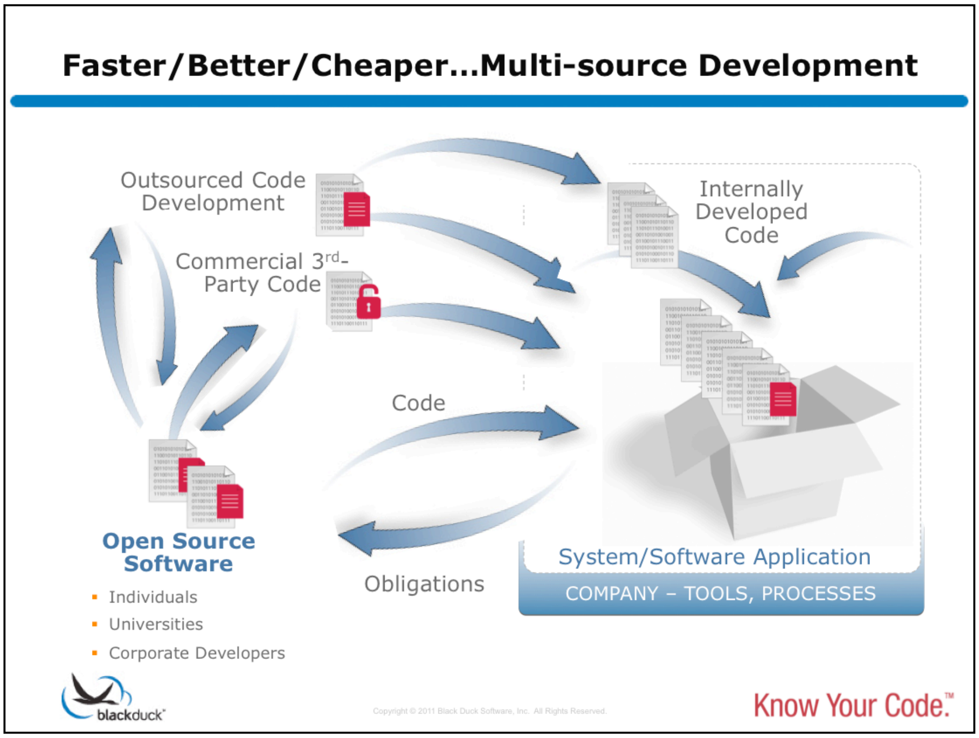
Know Your Code™ 3

Gartner Group's lead analyst on open source, Mark Driver (data in the slide is from November 2010), summarized the benefits and key challenges of using OSS. This position completely aligns well with Black Duck's experience: You have to use open source, but there are challenges/risks that require management.

In addition, Driver has commented on the ubiquity of open source, that it is unavoidable and should be embraced as part of normal development process. Mark also made the following predictions:

-- By 2016, OSS will be included in mission-critical software portfolios within 99% of Global 2000 enterprises, up from 75% in 2010.

-- By 2014, 50% of Global 2000 organizations will experience technology, cost and security challenges through lack of open-source governance.



Faster/Better/Cheaper…Multi-source Development

Outsourced Code Development

Commercial 3rd-Party Code

Internally Developed Code

Code

Open Source Software

- Individuals
- Universities
- Corporate Developers

Obligations

System/Software Application

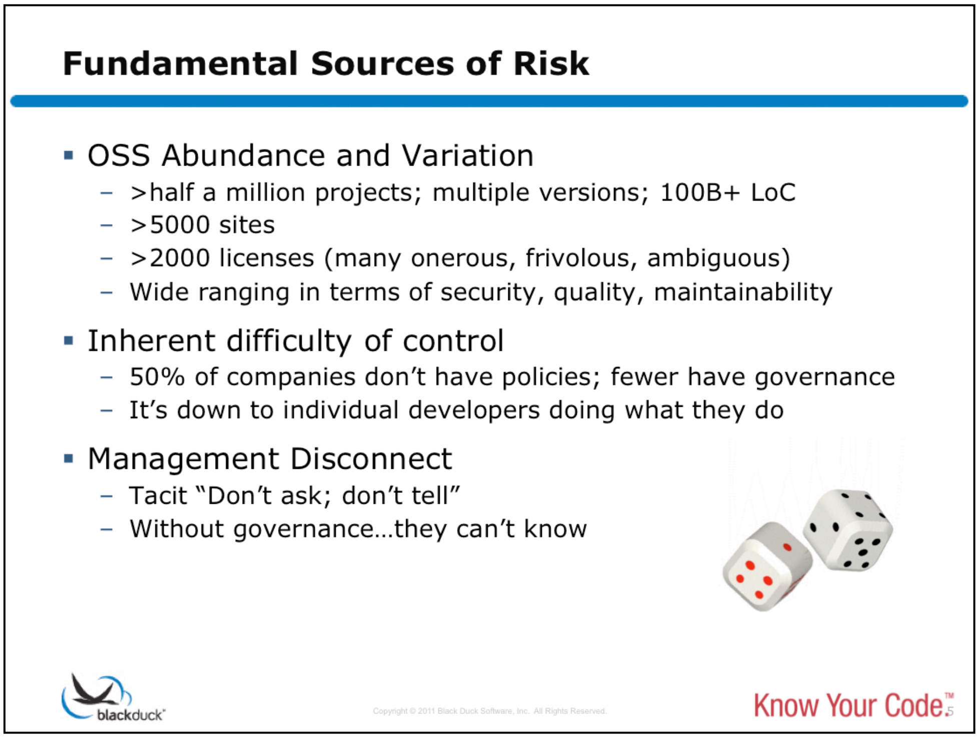
COMPANY – TOOLS, PROCESSES

Know Your Code.™

Whether DoD personnel or contractors, and whether its open source software or otherwise, this is how most software is developed today. Pressure on software developers has lead to a process of assembling components from a variety of sources and increasingly open source. We call this "Multi-source Development" and believe it is the "new normal" for software development.

The real picture is actually much more complex because every piece of code used in development likely comes from multiple sources as well. So there are many complex paths by which unidentified components can find their way into a code base.

The real point is that today it is not easy to know what components are actually in your code and to therefore identify the associated risks.



Fundamental Sources of Risk

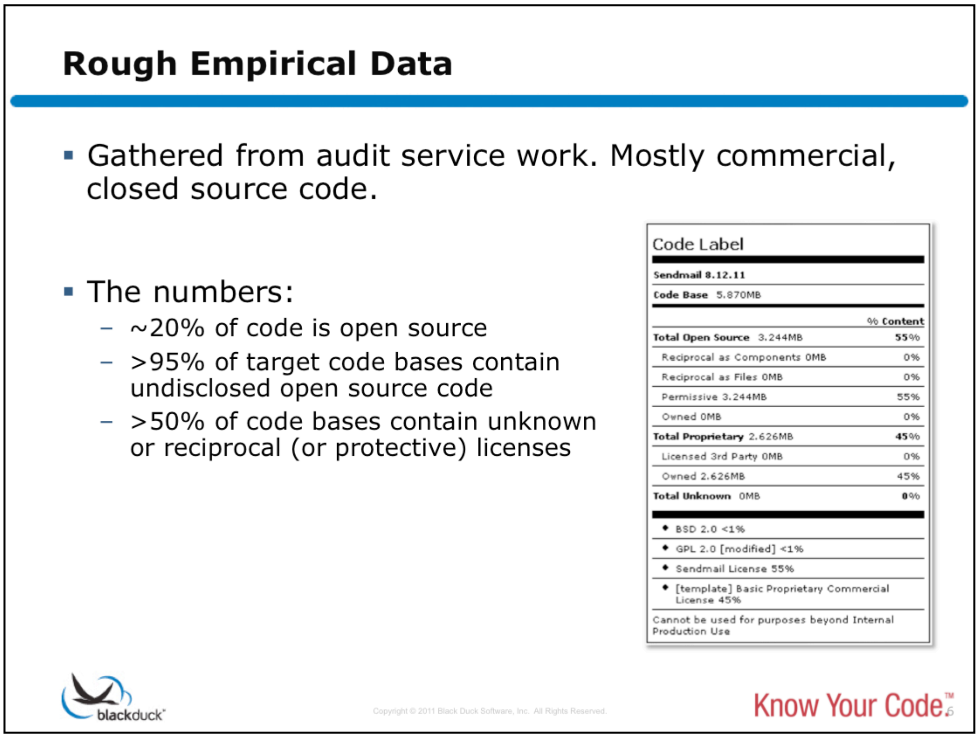
- OSS Abundance and Variation
  - >half a million projects; multiple versions; 100B+ LoC
  - >5000 sites
  - >2000 licenses (many onerous, frivolous, ambiguous)
  - Wide ranging in terms of security, quality, maintainability
- Inherent difficulty of control
  - 50% of companies don’t have policies; fewer have governance
  - It’s down to individual developers doing what they do
- Management Disconnect
  - Tacit “Don’t ask; don’t tell”
  - Without governance…they can’t know

Given the multi-source style of development and the increasing use of open source, there are fundamental forces that increase the potential of code risks:

-There’s an enormous amount of code out there freely available to anyone with a browser. Some of it is great code, some of it has problems with respect to security vulnerabilities, quality, documentation, support, maintainability, and licensing.

-This wealth of code is highly attractive to developers, but inherently difficult to control, and few companies have near the requisite controls in place. Without proper controls in place, decisions about what components end up in software are being made by individual developers.

-Supplier personnel who are making assertions about code content, typically don’t know. Software development has changed so much over the last few years that the folks in charge are generally not in touch with what developers are doing. And, even if they are in conceptual touch, without governance in place, they literally can’t know the details of what components are being used where.



Rough Empirical Data

- Gathered from audit service work. Mostly commercial, closed source code.
- The numbers:
  - ~20% of code is open source
  - >95% of target code bases contain undisclosed open source code
  - >50% of code bases contain unknown or reciprocal (or protective) licenses

Code Label

Sendmail 8.12.11

Code Base 5.870MB

	% **Content**
Total Open Source 3.244MB	**55**%
Reciprocal as Components 0MB	0%
Reciprocal as Files 0MB	0%
Permissive 3.244MB	55%
Owned 0MB	0%
Total Proprietary 2.626MB	**45**%
Licensed 3rd Party 0MB	0%
Owned 2.626MB	45%
Total Unknown 0MB	**0**%

- BSD 2.0 <1%
- GPL 2.0 [modified] <1%
- Sendmail License 55%
- [template] Basic Proprietary Commercial License 45%

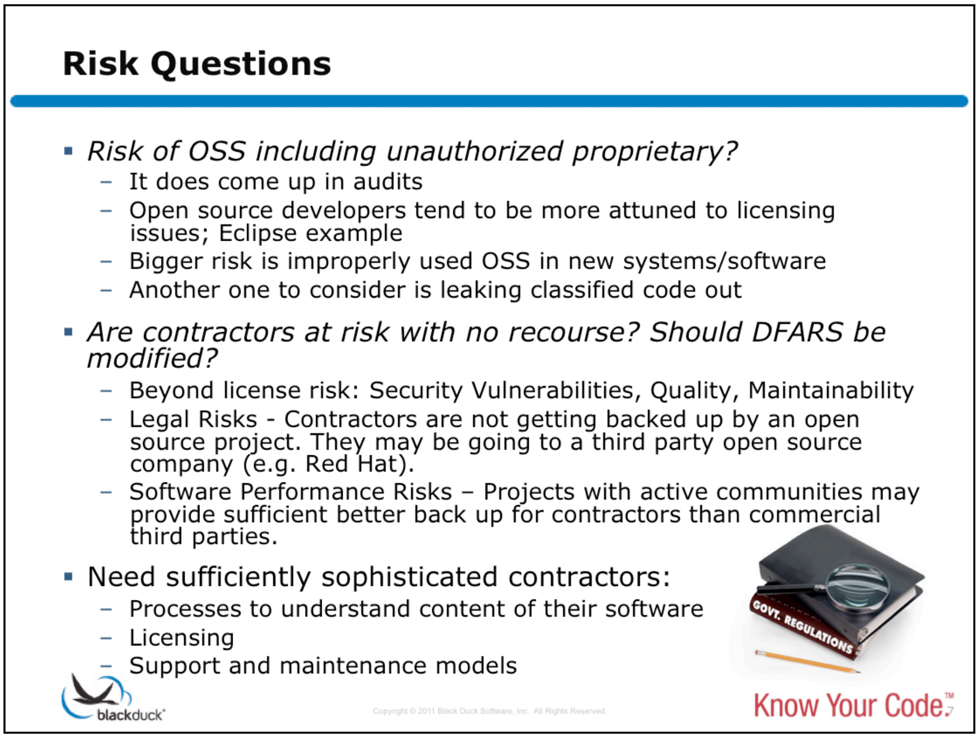
Cannot be used for purposes beyond Internal Production Use

Know Your Code.™

There is no way to comprehensively analyze how much of what components are used where out in the wild. However, Black Duck has performed 1000s of audits of code, typically closed-source commercial code and so we have some sense for the state of the system.

Of the code bases we scan, it is typical that 20% of the code is open source. (We've seen as high as 90%.) Often we are doing these code content audits in the context of a company being bought and we are comparing to a declared software Bill of Materials that a company has generated at the request of the buyer. Almost every time we find code that the code contains open source components that were unknown to the code owner. And, more than half the time, these components are licensed under licenses that are GPL-style or for which the licensing can not be determined.

The bottom line is that even companies that make an effort to determine what is in their code are generally unable to do so with any accuracy.



Risk Questions

- *Risk of OSS including unauthorized proprietary?*
  - It does come up in audits
  - Open source developers tend to be more attuned to licensing issues; Eclipse example
  - Bigger risk is improperly used OSS in new systems/software
  - Another one to consider is leaking classified code out
- *Are contractors at risk with no recourse? Should DFARS be modified?*
  - Beyond license risk: Security Vulnerabilities, Quality, Maintainability
  - Legal Risks - Contractors are not getting backed up by an open source project. They may be going to a third party open source company (e.g. Red Hat).
  - Software Performance Risks – Projects with active communities may provide sufficient better back up for contractors than commercial third parties.
- Need sufficiently sophisticated contractors:
  - Processes to understand content of their software
  - Licensing
  - Support and maintenance models

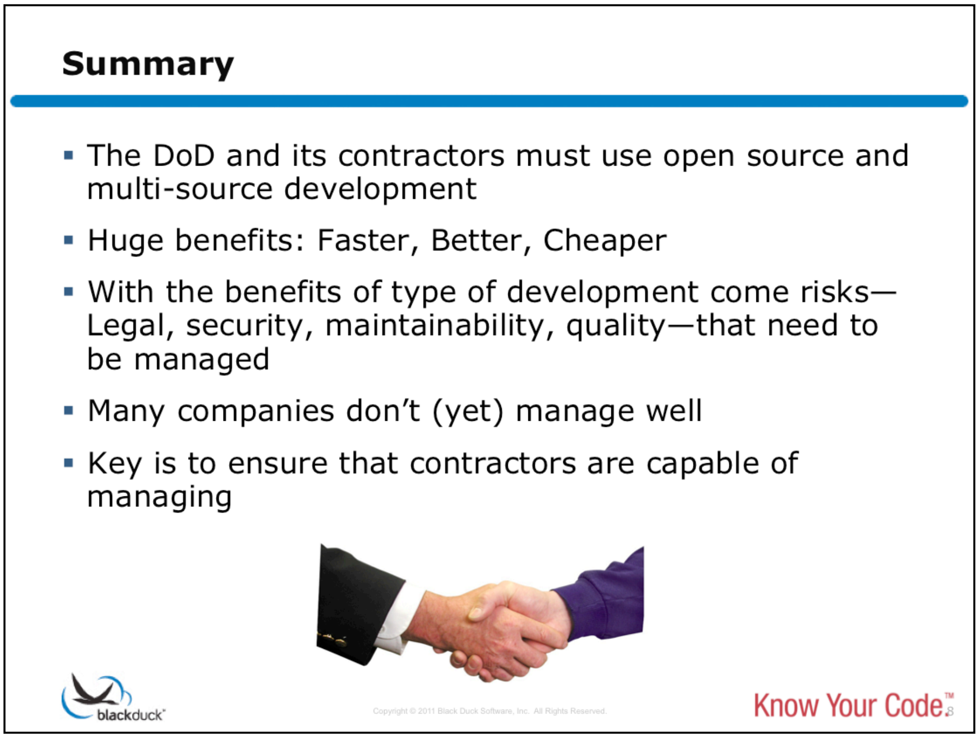
blackduck

Know Your Code.™

We have certainly seen proprietary code turn up in open source code. In one case, we informed a company that their code matched closely to an open source project and they discovered that a disgruntled employee had stolen their proprietary code and made it available to the world as open source. However, organizations and developers are generally more sensitive to ownership of proprietary code, and therefore proprietary code is less likely to "wander" than is open source. So, it is more common for the issue to be incompatible open source licenses in an open source component or in proprietary code. The Eclipse Foundation, for example, scans and analyzes every piece of code that comes in the door for this reason.

Another risk worth considering is code leaking out into the open source world. There are great self-serving reasons to make contributions back to open source projects, but there need to be controls on what goes out the door.

It's important to broaden the perspective on risk beyond license/copyright risk. There are plenty of other reasons to want to know what's in your code, beyond the legal ones. Only by knowing the source of



Summary

- The DoD and its contractors must use open source and multi-source development
- Huge benefits: Faster, Better, Cheaper
- With the benefits of type of development come risks—Legal, security, maintainability, quality—that need to be managed
- Many companies don’t (yet) manage well
- Key is to ensure that contractors are capable of managing

Know Your Code.™

There’s little choice about using open source off the shelf as well as in the form of components in developed code. The benefits are just too great to ignore. However, along with the benefits come risks, not just legal, but also risks with respect to security, quality and future maintainability. Those risks need to be managed and it is imprudent to assume contractors are doing so properly. The key for the DoD is to ensure that their contractors are sufficiently open source savvy to manage the risks.