



blackduck™

Know Your Code.™

Developing an Open Source Governance Strategy

SSTC

Tampa, FL

June 18, 2007

John Smith
Director Product Management
jsmith@blackducksoftware.com
781.810.2082

Premise of this Briefing

- Open source is a technology **process** rather than a technology
- Disruptive technologies historically present challenges to IT management
- With open source:
 - No one entity controls the process or its products
 - It potentially introduces large quantities of unknown code
 - Licensing, legal obligations & security risks can arise
- The open source component re-use model is attractive
 - It's available to all organizations
 - But, can my organization use the same model internally?
- First, a need for a Software Component Governance strategy
- Governance = Policy + Compliance
 - Policy is what to do
 - Compliance is monitoring and reporting



*Businessmen go down with their
businesses because they like the old
way so well they cannot bring
themselves to change. ...*

Henry Ford, My Life and Times, 1922



Definition: Disruptive Technology

- A quantum change, not an incremental step, that finally affects mainstream operations
- Often under-performs established products at first
- A technology that fringe customers value highly
- Most companies do not realize the impact of this technology until it is too late and others have taken over their field/product
- Characteristics:
 - Markets that at first do not exist and therefore cannot be analyzed.
 - Products are: cheaper, faster, simpler, more convenient to use.

Source: *Technology and Innovation*
Henry C. Co, Technology and Operations Management,
California Polytechnic and State University



The Disruption Cycle

1. A “New Disruptive Technology” (NDT) enters the market
2. Early adopters bring NDT into business settings to solve problems either not currently or better addressed by NT
3. Many unconnected pockets of NDT emerge in workplaces
4. IT management notices NDT has gained a foothold
5. IT & Executive management acknowledge the business value provided by NDT
6. IT & Execs acknowledge the need for NDT management tools
7. Management tools integrate NDT into business processes and create increased value

Examples: Photo-Copier, Fax Machine, PCs, Spreadsheets, LANs, WWW, Open Source



Open Source: Disrupting the IT World--Again

- PC's: CPUs everywhere
- LANs: Connectivity everywhere
- WWW: Collaboration everywhere
- Open Source: Development everywhere

Common stages of Disruptive Technology adoption:

- Creeps in from many directions
- Evades and threatens IT control
- Stays because of business value
- **Policy, governance & funding** ← Open Source is here
- Becomes a standard practice/technology
- Revolutionizes business



Component Reuse: Federal Government



Federal Enterprise Architecture

FEA Program Management Office

Office of Management and Budget
Executive Office of the President

February 2004



Services and Components Based Architectures
A Strategic Guide for Implementing Distributed and Reusable
Components and Services in the Federal Government

Version 3.5 Chapter 1: Executive Strategy
Last Updated: January 31st, 2006



Architecture and Infrastructure Committee,
Federal Chief Information Officers Council
January 2006



Copyright © 2007 Black Duck Software, Inc. All Rights Reserved.

Definition of Reuse (SCBA)

“Any use of a preexisting software artifact (component, specification, etc) in a context different from that in which it was created.”

“Government leaders should use the resources and guidance provided by the CIO Council, FEA, and other government-wide efforts, as well as their own agency resources, to establish service component reuse programs in their agencies.”

“Senior leaders must champion reuse by expecting that assets be reused, recognizing projects and individuals that successfully reuse assets or publish them, and by making reuse a priority....and Rewarding individuals and projects who successfully publish Service Components or have high reuse rates”

Acquisition recommendation (SCBA):

RFI, RFP, or RFQ processes should change to embrace reuse

(e.g., by integrating reuse concepts into questionnaires and decision criteria). Possible questions to add to decision criteria include:

- Does a vendor or contractor's technical approach embrace re-usability?
- Can the requirements for this project support any other organizations?
- Will the outcome result in new Service Components that can be registered in Core.Gov?

http://www.whitehouse.gov/omb/egov/documents/SCBA_Whitepaper_Chapter_1.pdf

Know Your Code.™

DoD Open Technology Development Roadmap



Roadmap Plan

April 2006

Prepared for:

**Ms. Sue Payton
Deputy Under Secretary of Defense
Advanced Systems & Concepts
www.acq.osd.mil/asc/**

Prepared by:

**J.C. Herz
Mark Lucas
John Scott**

Version 3.1 (Final)

Cleared for Open Publication, June 7, 2006
Office of Security Review, Department of Defense



Calculating Component Re-Use Value

$$NPV = \sum_{k=0}^M \frac{[C_{c_{wrp_k}} - C_{c_{rp_k}} + P_k - C_{p_{r_k}}] \rho_k}{(1+i)^k}$$

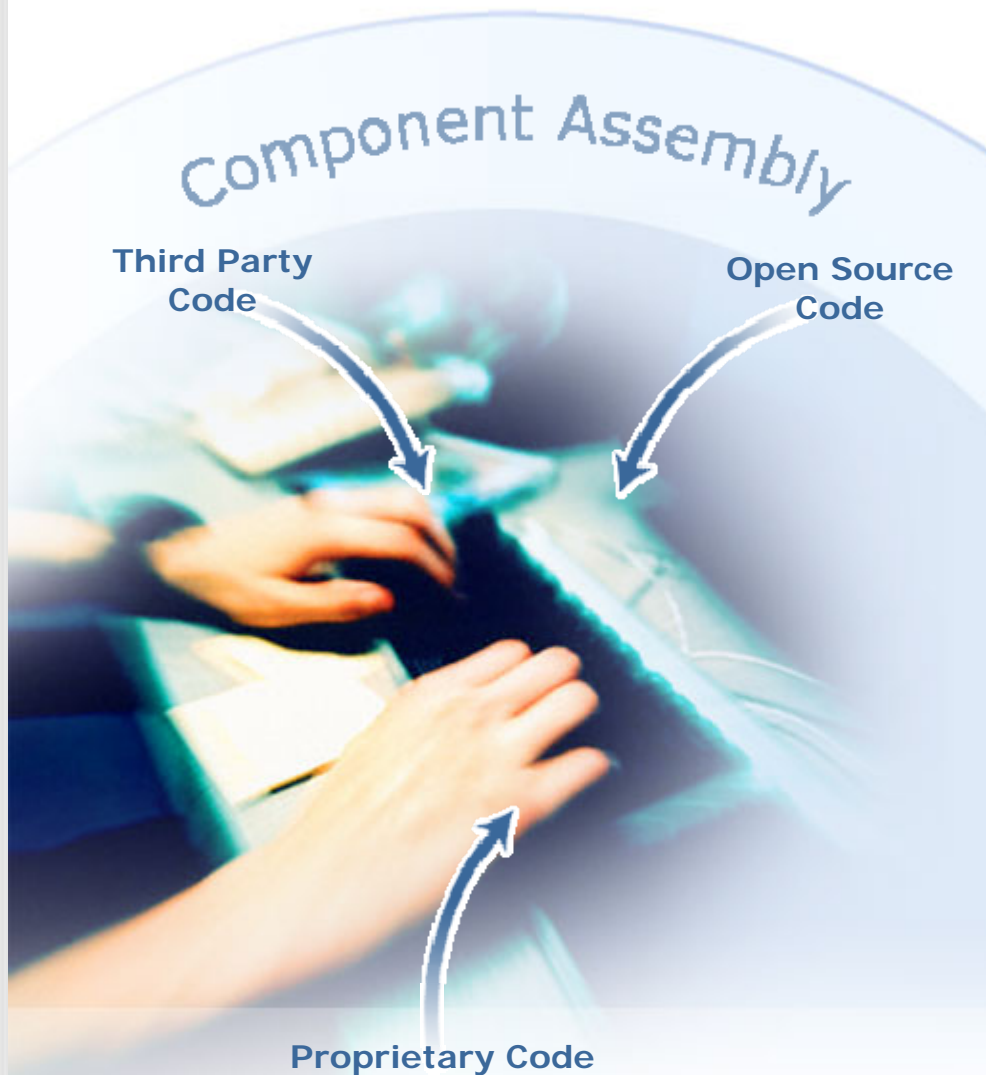
Lim, Wayne C.,
"Managing Software Reuse"

Calculates the value from reuse by taking the sum of the consumer costs reduced and avoided and the increased profit from reuse, less the producer costs. This figure is then multiplied by a probability which accounts for risk, and finally discounted to take into account the time value of money.

Consumer	C_{cwrp}	Cost to consumer to create product/system without reuse
	$C_{c rp}$	Cost to consumer to create product/system with reuse
Producer	$C_{p r}$	Cost to producer to create an asset for reuse
Profit	P	Profit from increased revenues enabled by reuse
Risk	ρ	Probability of receiving cash flow
Time Value	i	Interest rate by which cash flows are discounted
	M	Number of time periods under consideration
Output	NPV	Net Present Value



Inevitability of “Mixed-Source” Environment



- Virtually all organizations that develop software are now working in a “mixed-source”, mixed intellectual property environment
- Reuse of internally and externally available source code is inevitable
- Uncontrolled re-use introduces substantial risks



Uncontrolled Reuse Introduces Risk

Component Assembly

WARNING
Restricted
Open Source
Code

- Loss of software integrity
- Costly recalls, code reviews, redesign and/or re-work
- Inadequate due diligence and/or compromised assets
- Partner or customer relationship issues
- Loss of intellectual property
- Unexpected legal and/or financial obligations



Open Source Governance

- Allows for informed and appropriate entry and use of open source in the enterprise
- Enables organizations to leverage and adopt the best features of the open source process:
 - Continuous improvement development model
 - Collaborative development by distributed workforce
 - Wide peer review
 - Control of product
 - Leaner applications
 - Quicker and fewer bug fixes
 - Large support network
- Provides enterprises a management structure to begin implementing open source processes internally
- First comes policy



The Open Source Policy Lifecycle



- Examine open source use cases
- Evaluate open source license compliance requirements
- Assess risk exposure
- ACTION: Create checklist of open source concerns and license requirements



The Open Source Policy Lifecycle



- Work with all key organizations to ensure their open source requirements are captured
- Define legal requirements for third-party software
- Document all legal and open source use requirements
- Define how internal personnel may interact with open source
- ACTION: Create open source/third-party approval process
- ACTION: Create open source use policy



The Open Source Policy Lifecycle



- Understand how organization is currently using open source
- Review use against Open Source Policy
- Define any necessary remediation to ensure compliance with Open Source Policy
- ACTION: Develop inventory of company open source use
- ACTION: Create remediation plan and track progress



The Open Source Policy Lifecycle



- Develop training materials to educate all appropriate employees in Open Source Policy
- Ensure all appropriate employees are educated regarding policy
- Include Open Source Policy training as part of new-hire orientation
- Include suppliers and contractors
- ACTION: Create training program
- ACTION: Deliver training to all existing and new hires



The Open Source Policy Lifecycle



- Develop organization to serve as repository of open source knowledge and reference point for Open Source Policy
- Ensure organization is involved as part of project management process
- Incorporate organization into open source policy training and approval mechanism
- ACTION: Create Open Source Program Office/Review Board
- ACTION: Monitor code prior to acquisition
- ACTION: Monitor code prior to release/production



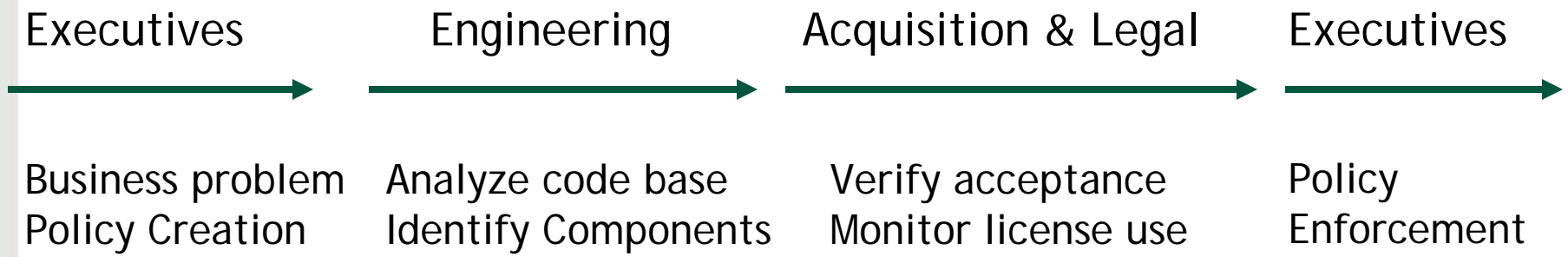
Putting a Governance Plan into Action

Two areas needing Governance:

1. License management
2. Component re-use



Open Source Software Governance

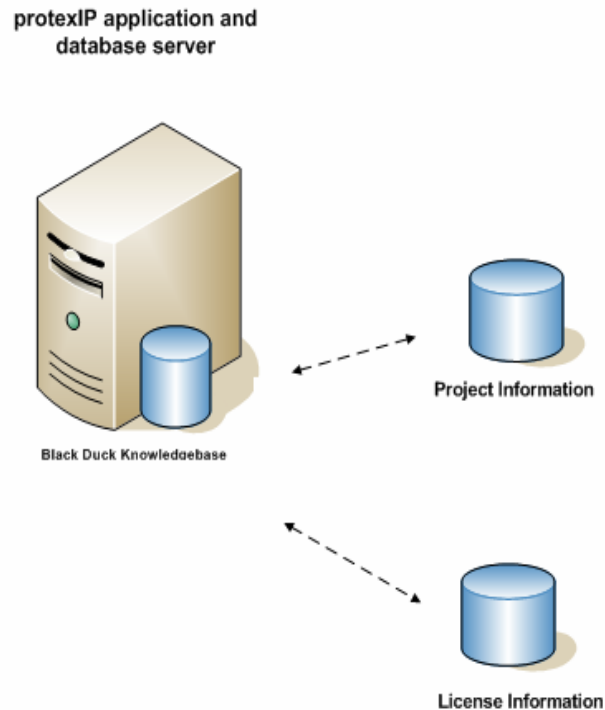


Involve all relevant stakeholders in the policy setting and governance process



Black Duck Knowledgebase

The Knowledgebase is the core of Black Duck Products



- The code is rendered into "Code Prints": digital representations of a file.
- Two types of Code Prints
 - Files, allowing for the identification of exact matching.
 - Snippets, allowing for the identification of similar files.
- Open Source, OEM, Commercial Code Prints in Knowledgebase
- Extensible Code Printing of legacy or delivered code bases for enterprise implementations



Black Duck KnowledgeBase: 1,200+ licenses identified to date

Licenses
Apache License Version 2.0

Licenses
BSD 2.0

Licenses
Common Public License

Licenses
Creative Commons Attribution 2.0

Licenses
Eclipse Public License - v 1.0

Licenses
GPL 2.0

Licenses
LGPL 2.1

Licenses
MIT License V2
Mitre CVW License version 1

Licenses
Sun Industry Standards Source License - SISSL
Sun J2EE Version 1.3.1 Binary Code License
Sun J2RE Binary Code License v. 1.4.2_x
Sun License for J2SDK
* Sun Public License v 1.0
Sun XML Demo License
Sundial License (similar to BSD)
SunW Drv8 License
Supadup License

Apache
BSD
CPL
Creative Commons
Eclipse
GPL
LGPL
Microsoft
MIT
Sun



Black Duck KnowledgeBase: Code Prints from 3,500+ sites, Apache to Zope

The screenshot shows a web interface with a list of projects. Each project name is enclosed in a blue-bordered box with a 'Projects' header above it. Green arrows point from the project names to the corresponding website names listed on the right. The projects listed are:

- Apache-Jakarta POI
- Asterisk
- CLISP - an ANSI Common Lisp
- Eclipse Project
- FSF.org
- Linux kernel
- MySQL Database server
- PHP Nuke
- SourceForge.net
- Sun JDK
- Zope
- Zope development at Chalmers-DBQueryPool
- Zope development at Chalmers-DBQueryTypes
- Zope development at Chalmers-ExternalAuthentication
- * Zope development at Chalmers-KerberosIdentification
- Zope development at Chalmers-LDAPAuthorization
- Zope development at Chalmers-RAMCacheCrumbler
- Zope development at Chalmers-StaffList
- Zope File System Folder

Apache.org
Asterisk.org
Cons.org
Eclipse.org
FSF.org
Kernel.org
MySQL.com
Phpnuke.org
SourceForge.net
Sun.com
Zope.org



Set policies

Tools > License Modifier

Search:

Show: All

Name	Type
Apache License Version 2.0 [modified]	Modified
apcd License	Standard
APL\11 License	Standard
Apple Disclaimer	Standard
Apple Public Source License 1.2	Standard

Prohibited licenses

General Attributes Obligations

Approved:

Comment:

Approved licenses

General Attributes Obligations

Approved:

Comment:



License data allows automated license compatibility calculation

GPL 2.0 License

License Conflict

Your Right to Distribute Source Code / Enforced Sharing of Source

<input checked="" type="radio"/> True	You may distribute source code.
<input type="radio"/> False	
<input checked="" type="radio"/> True	You are required to distribute source code.
<input type="radio"/> False	

You are required to distribute the source code of the Work Based on the Licensed Code.

Proprietary Commercial License

Your Right to Distribute Source Code / Enforced Sharing of Source

<input type="radio"/> True	You may distribute source code.
<input checked="" type="radio"/> False	
<input type="radio"/> True	You are required to distribute source code.
<input checked="" type="radio"/> False	

You are not entitled to distribute source code.



License text

Complete license text available for review

License Text:

The GNU General Public License (GPL)

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.



Review – Obligations in Detail

Obligations:

Fulfilled	Obligation	Type
<input type="checkbox"/>	You are not entitled to impose a fee related to what Recipient may do with the code.	Legal
<input type="checkbox"/>	You are not entitled to place additional restrictions on what Recipient may do with the code.	Legal
<input type="checkbox"/>	You are required to disclaim warranties on behalf of others.	Legal
<input type="checkbox"/>	You are required to distribute the source code of the Dynamic Library.	Legal
<input type="checkbox"/>	You are required to ensure that the software displays a particular notice at runtime.	Legal
<input type="checkbox"/>	You are required to grant the right to reverse engineer the Dynamic Library.	Legal
<input type="checkbox"/>	You are required to include a copy of the license in distributions that you make.	Legal
<input type="checkbox"/>	You are required to license the entire Dynamic Library under the same terms as the original code.	Legal



Component Bill of Materials

“Does any of my code contain open source, freeware or shareware?”

“If yes, please provide the software’s name, origin, and your license for this software.”

Bill of Materials							
Approval Status	Violation	Component	Version	Comment	License	Usage	# Files
Approved		ANTLR, ANother Tool for Language Recognition	2.7.5		Public Domain	Snippet	211
Pending Approval		Apache Jakarta Commons Logging	1.0.3		Apache 1.1 [modified]	Snippet, File	245
Pending Approval	Declared License Violation	EMMA code coverage	maven-1.0-plugin-0.4		Common Public License	File	297
Pending Approval	Declared License Violation	JUnit	3.8.1		Common Public License	Snippet, File	93
Approved		Net Hack	3.4.3		BSD 2.0	Snippet, Released Component	23
Pending Approval	Declared License Violation	Shift2Ingres	1.0 Alpha	ignored	GPL 2.0 [modified]	Snippet, File	126
N/A	N/A	STZ-EA			[template] Basic Proprietary Commercial License	Snippet	165
Approved		Tomcat FAQ			BSD 2.0	Snippet	1

BOM: Component name, version, applicable license, usage, origin, approval status



Code Label

Code Label	
Rollup Project	
Code Base 67.785MB	
% Content	
Total Open Source 24.203MB	36%
Reciprocal as Components 4.154MB	6%
Reciprocal as Files 0MB	0%
Permissive 20.049MB	30%
Owned 0MB	0%
Total Proprietary 42.572MB	63%
Licensed 3rd Party 0.004MB	<1%
Owned 42.027MB	62%
Total Unknown 0MB	0%
<ul style="list-style-type: none">• Apache 1.1 3%• BSD 2.0 27%• Eclipse Public License - v 1.0 <1%• GPL 2.0 6%• LGPL 2.1 <1%• Public Domain <1%• Sun License for J2SDK <1%• Unspecified 27%• [template] Basic Proprietary Commercial License 64%	
Cannot be used for purposes beyond No Use	

- Provides a structured view of code contents
- Communicate code contents to customers and partners



Know Your Code™

- **Measure and Manage**
 - Assess the amount of open source and other software components in code base
 - Audience: SW Development, IT management, Acquisition
 - Use: Implement policy for governance of appropriate component use
- **Software Intellectual Property Assessment**
 - Licensing obligations and restrictions (open source, OEM, proprietary)
 - Audience: Legal, acquisition, IT management
 - Use: License compliance, code acceptance, SW supply chain audit
- **Code Pedigree**
 - Attribution of code origins, authentication
 - Audience: SW development, legal, IT management
 - Use: Software assurance
- **Software Bill of Materials**
 - Comprehensive listing of all software components in a project
 - Audience: SW development, acquisition, IT management
 - Use: Validation and verification of code contents (IV&V)
- **Component Catalog**
 - Software IP asset inventory (open source, OEM, internally or externally developed)
 - Audience: SW development, acquisition
 - Use: Component reuse of Software IP assets; within RFIs, RFQs, RFPs



Software Compliance Management

- Create and enforce policy in the software development process to meet:
 - Business goals
 - License obligations
 - Regulatory requirements
- Enable re-use of:
 - Open source software
 - Commercial components
 - Enterprise software assets
- Implement business process based on best practices
- Software Assurance
 - Code Authentication
 - Independent Verification and Validation (IV&V)



Best Practices

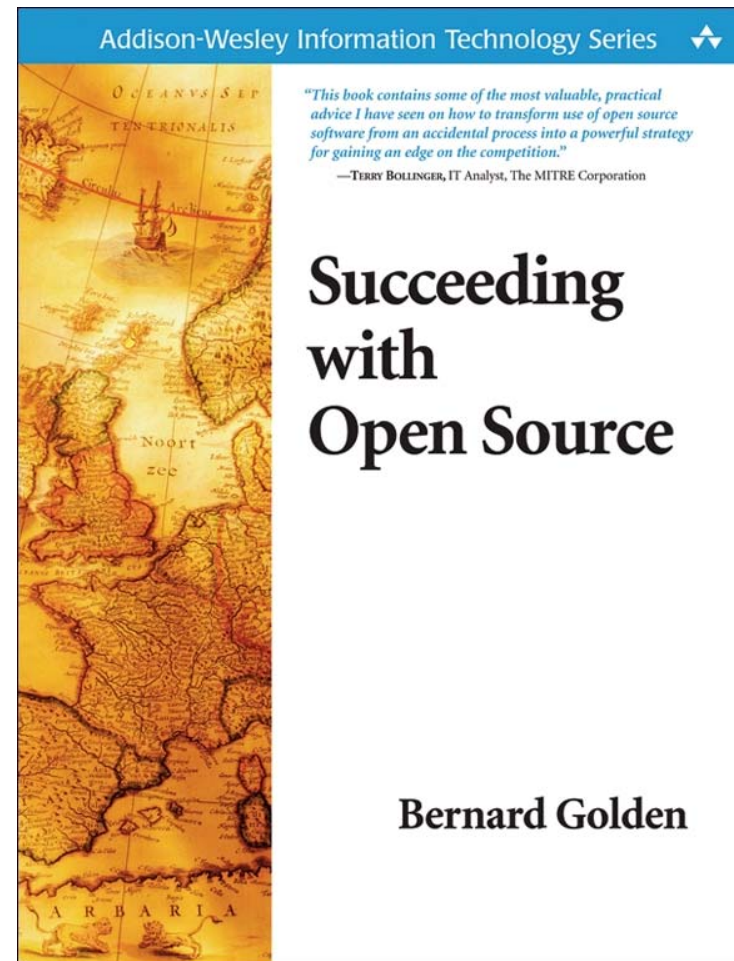
Managing Software Intellectual Property

1. Create Policy and implement a governance platform
2. Reuse existing components wherever possible
3. Track and control changes to internal components
4. Control re-use of sensitive or external components
5. Verify every build and release
6. Review compliance at critical project transitions
7. Control component contribution & disposition
8. Assess software components before acquisition



Succeeding with Open Source

- Leading open source strategy consulting firm
- Representative clients: SugarCRM, Compiere, OpenLogic, Red Hat
- Leading authority on enterprise open source use
 - Author of "Succeeding with Open Source" (Addison-Wesley, 2005)
 - Creator of Open Source Maturity Model
 - "The Open Source" blog for CIO Magazine



Bibliographic Information

- Many software reuse studies can be found at the Data Analysis Center for Software (DACS) which is a Department of Defense (DoD) Information Analysis Center (IAC) run by the AF Research Lab at Rome, NY.

<https://www.goldpractices.com/practices/arrc/index.php>





blackduck™

Know Your Code.™

Thank You

© 2007 Black Duck Software. Black Duck Software, the Black Duck logo, transactIP, exportIP, protexIP, and Know Your Code are trademarks of Black Duck Software, Inc. All other trademarks are the property of their respective holders.