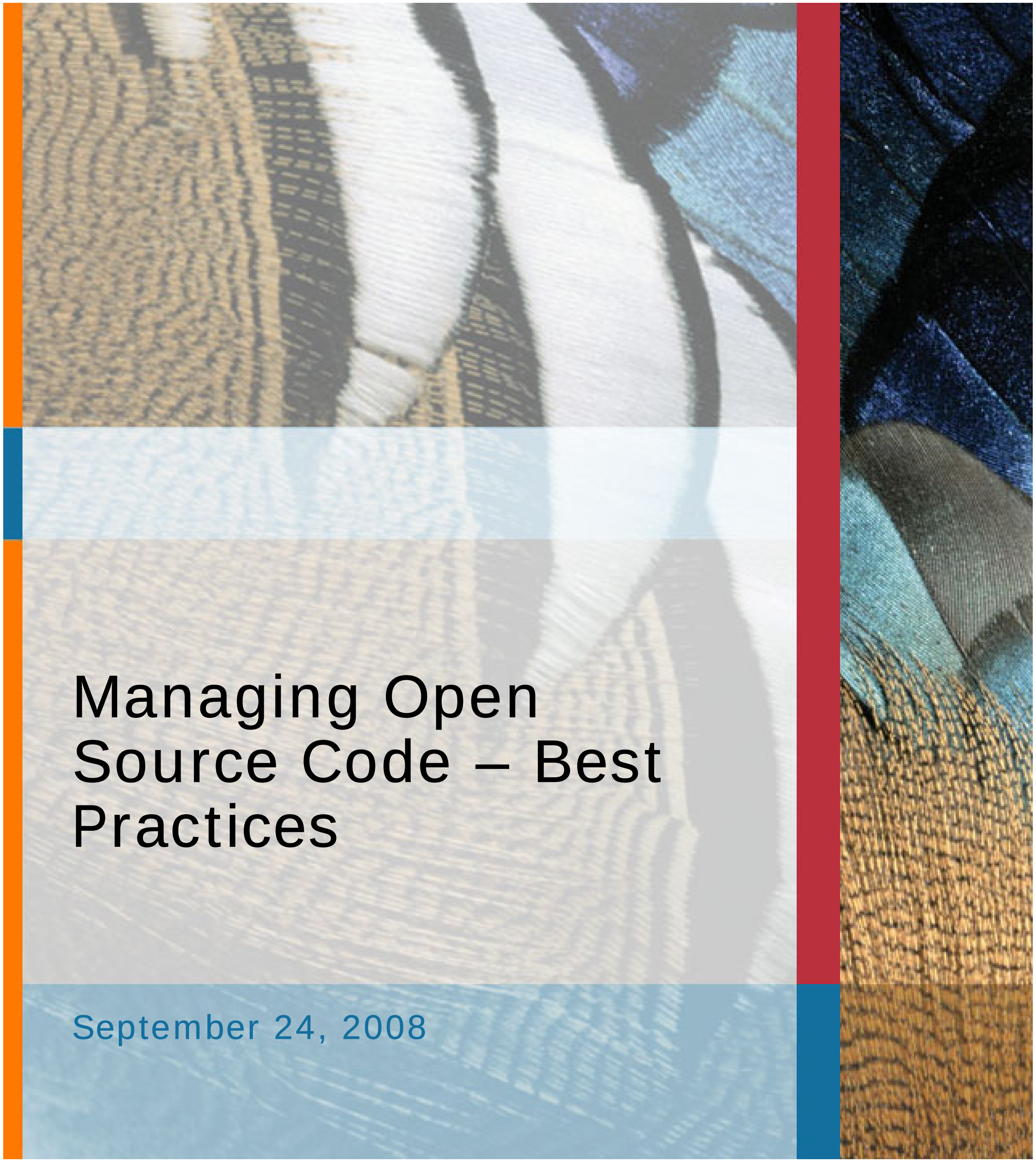
blackduck™



Managing Open Source Code – Best Practices

September 24, 2008

Agenda

- Welcome and Introduction – Eran Strod
- Open Source Best Practices – Hal Hearst
- Questions & Answers
- Next Steps

About Black Duck Software

Accelerate time-to-market and reduce development costs by providing products and services for finding, managing and confidently deploying open source software.

Mission

Founded in 2002 and backed by industry leaders

SAP

FLAGSHIP VENTURES

GENERAL CATALYST PARTNERS

redhat

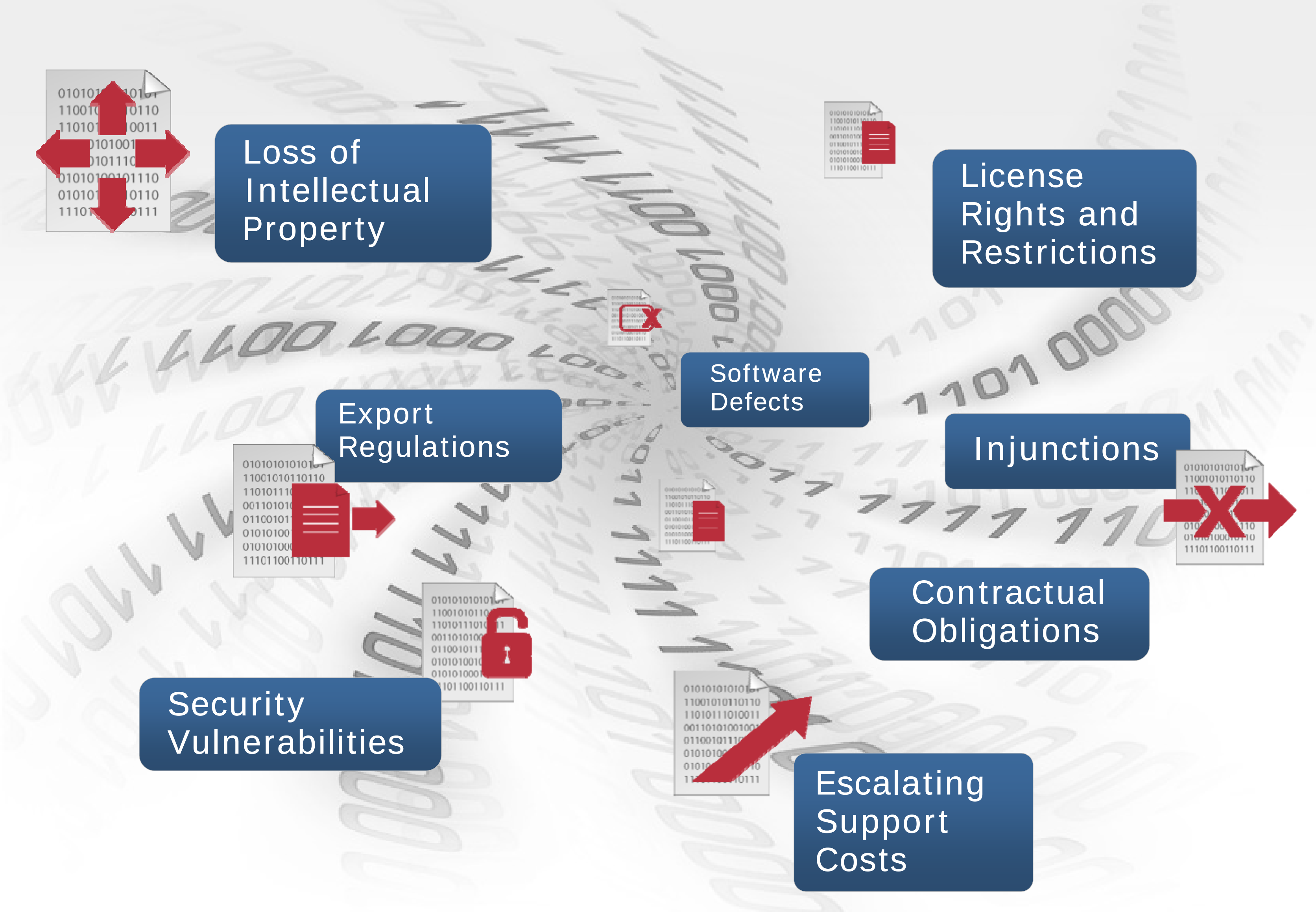
intel

FOCUS VENTURES

FIDELITY VENTURES

Black Duck Global Distribution

Mixed Code Development Adds Risk



Loss of Intellectual Property

License Rights and Restrictions

Software Defects

Export Regulations

Injunctions

Contractual Obligations

Security Vulnerabilities

Escalating Support Costs

Know Your Code.™

Speaker

Hal Hearst

Sr. Director of Professional Services

Black Duck Software

Background

- Responsible for delivering software assessment and implementation services to Black Duck Software's customer base
- Help Black Duck customers design and implement processes to manage the use of open source software
- Been with Black Duck Software Since October 2004
- 20 year career in professional services with companies such as Accenture, SAP and others

The Golden Rule for Managed Use of Open Source

Treat the management of open source software as an integrated, cross functional **business process**, and not simply as a development process.

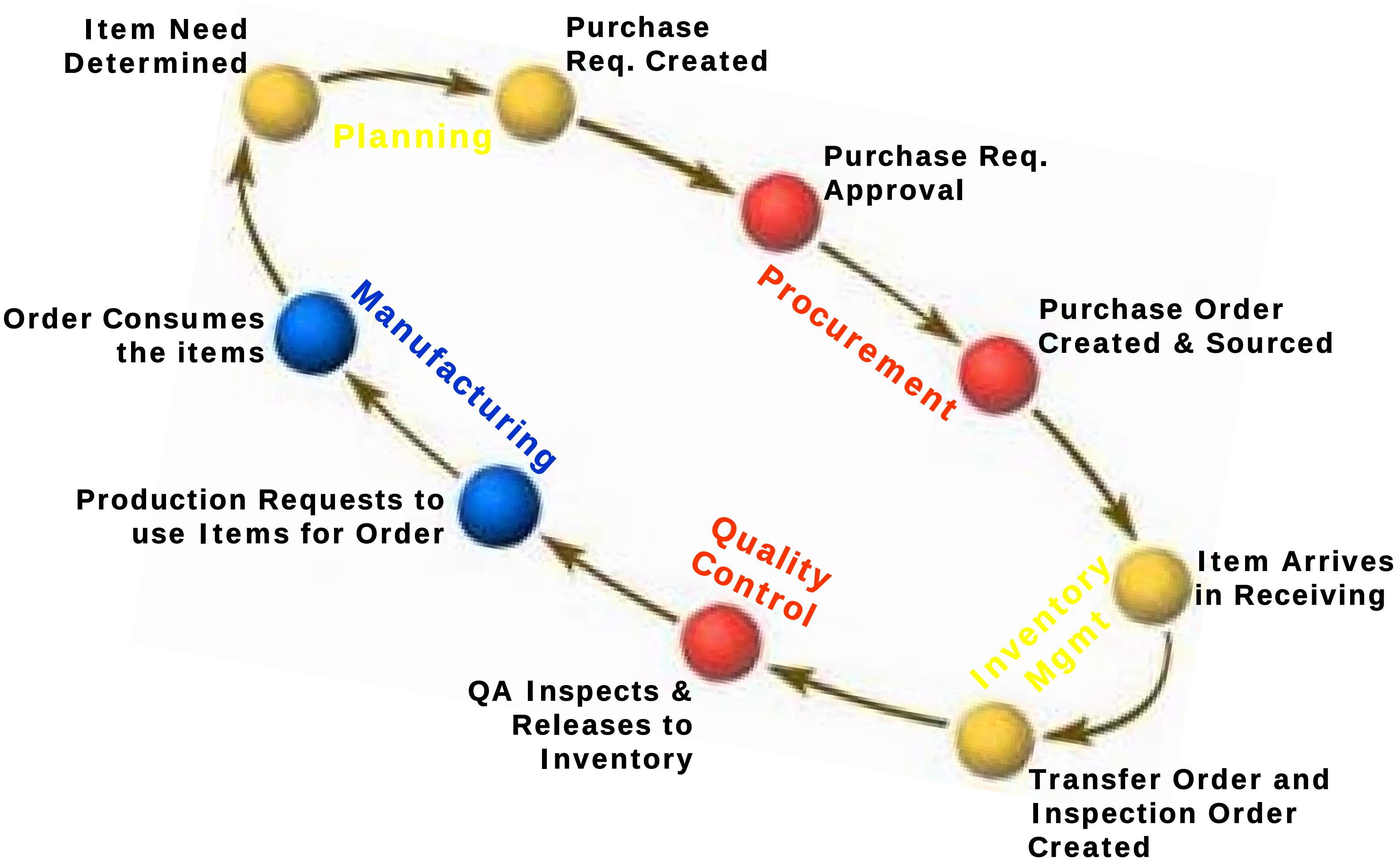
Golden Rule Details

- Cross functional
  - Product Planning/Management
  - Legal, Security & Export Compliance
  - Engineering
- Integrated Processes
  - Component Management
  - License Management
  - Release Management
    - Release Planning
    - Release Delivery
  - Security Review
  - Export Compliance Review

Golden Rule Details - Continued

- Systemic
  - Baked in to the culture & workflow
  - Event Driven
    - Component approval request
    - Planning a release
    - Accepting a code drop from a vendor/outsourcer
    - Performing a build
    - Creating a release
- Embrace Supply Chain Techniques
  - ERP systems brought together different users and processes
  - Workflow automates task creation
    - Notifications
    - Process Monitoring
  - Central repositories of data
  - Business Process Integration is the key

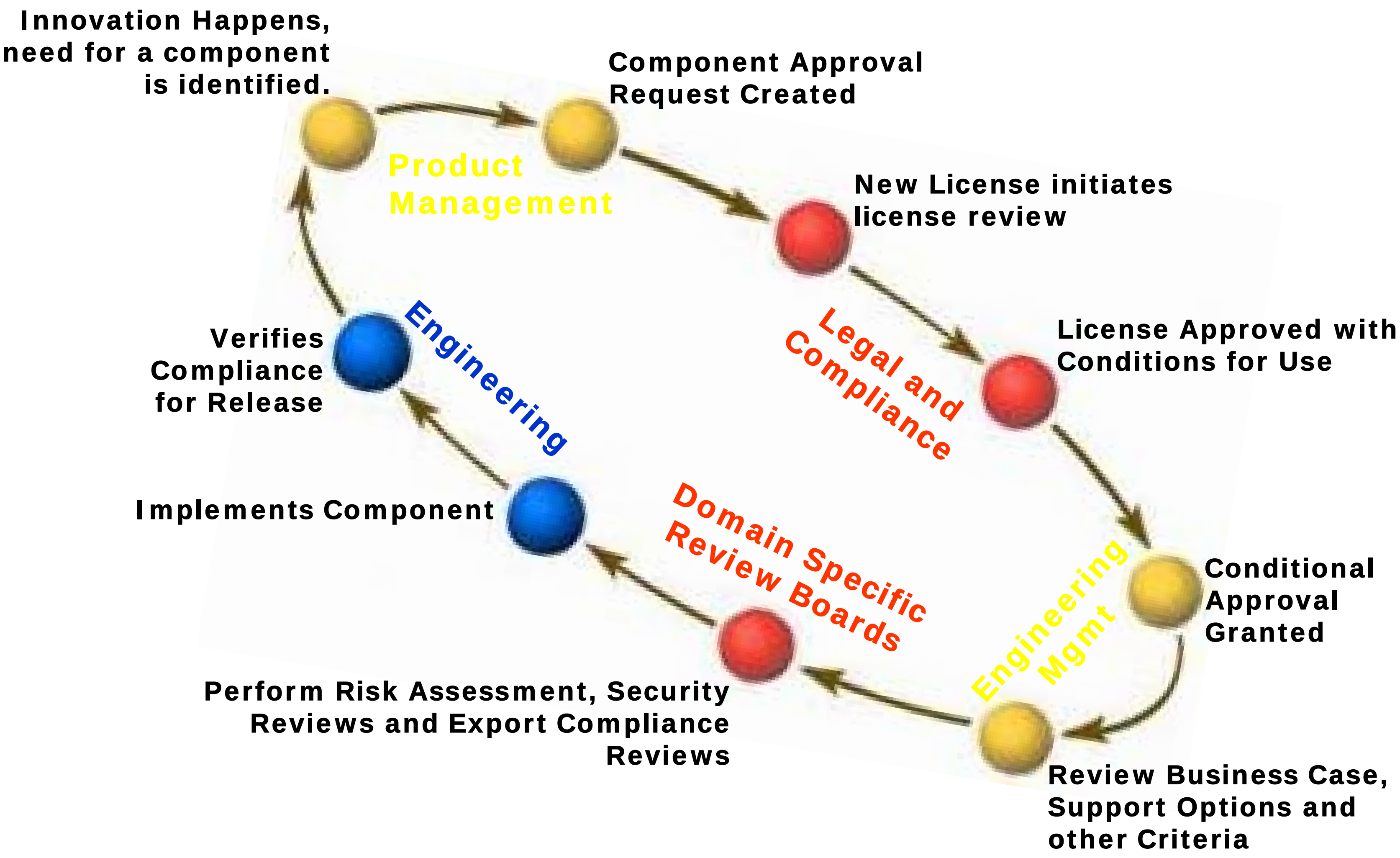
Example Supply Chain Business Process


Item Need Determined
Purchase Req. Created
Planning
Purchase Req. Approval
Procurement
Purchase Order Created & Sourced
Item Arrives in Receiving
Inventory Mgmt
Transfer Order and Inspection Order Created
Quality Control
QA Inspects & Releases to Inventory
Production Requests to use Items for Order
Manufacturing
Order Consumes the items

Supply Chain Comparison

- Technology companies have software supply chains
- Software products have bill of materials (BOM's)
- Tech. companies have similar roles and events
  - Materials Planner = Product Management
  - Purchase Req's = Component Approval Request
  - Warehouse = Source Code Management / Asset Management
  - Quality Assurance = Numerous types of code analysis
  - Procurement Approvals = Legal & Compliance Approvals
  - Shop Floor Production = Engineering

Example Software Development Business Process



Innovation Happens, need for a component is identified.

Product Management

Component Approval Request Created

New License initiates license review

Legal and Compliance

License Approved with Conditions for Use

Conditional Approval Granted

Engineering Mgmt

Review Business Case, Support Options and other Criteria

Domain Specific Review Boards

Perform Risk Assessment, Security Reviews and Export Compliance Reviews

Implements Component

Engineering

Verifies Compliance for Release

Application Development Lifecycle

Supporting Open Source-based Development

Select

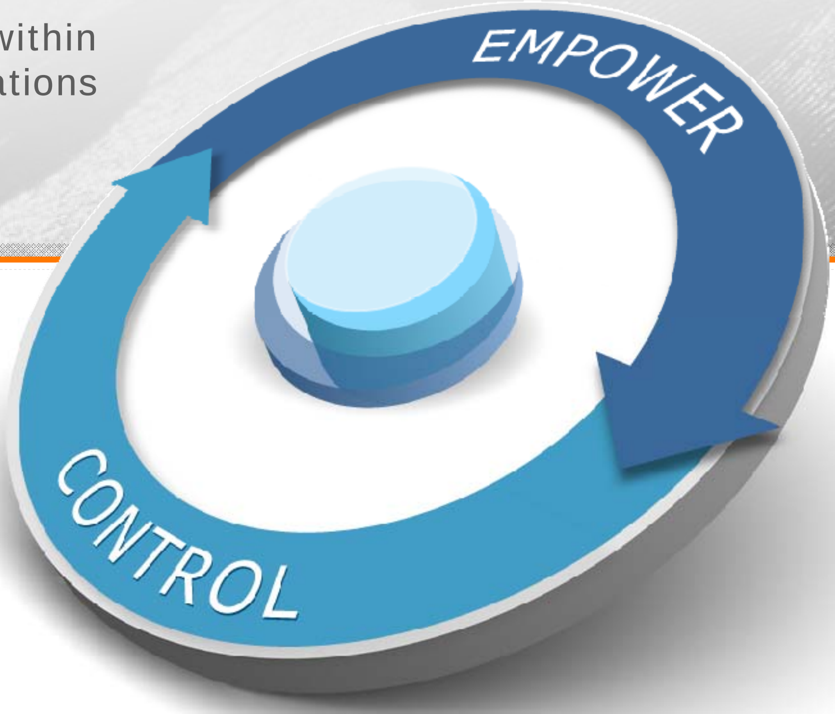
Code based on policies and key metadata

Search

For code to use within applications

Approve

Code based on organization's control polices



Monitor

Code usage and impact across complex applications

Build

The approved code using preferred tools

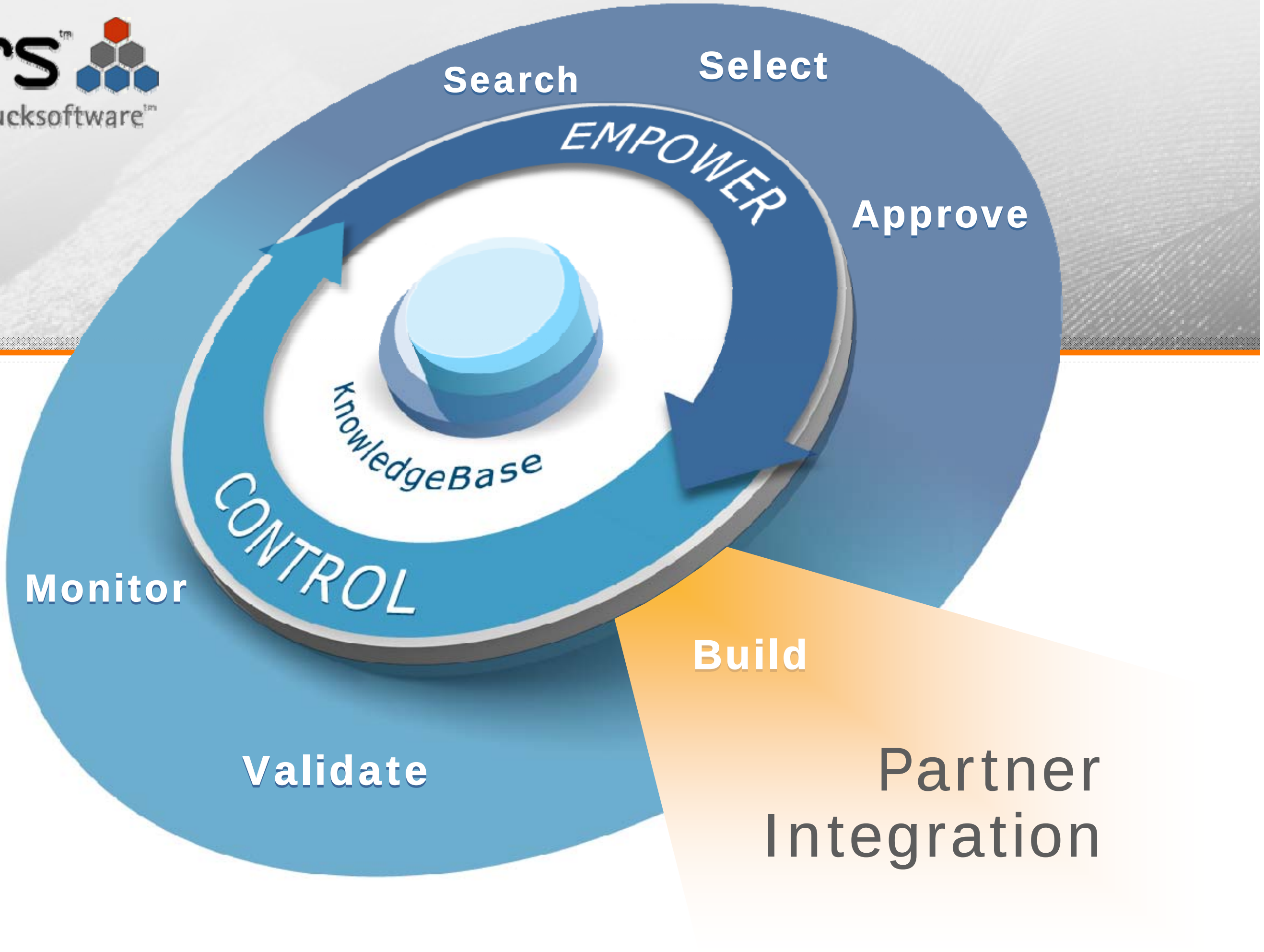
Validate

That only authorized code is used

Black Duck's Portfolio

koders powered by blackducksoftware

Code Center



Protex

Export

Open Source Programs Elements

1. Published Policy
   1. Created via Cross Functional Team
   2. Organization is educated on the policy
2. Open Source Process Owner
   1. Keeps the wheels running
   2. Grant certain types of approvals
3. Approval Processes
   1. Component Review & Approval
   2. License Review & Approval
   3. Release Plan Review & Approval
4. Monitoring & Tracking Process
   1. Component Verification
   2. Security Notifications
   3. Component Upgrade Notifications
5. Obligation Verification Process
   1. Ensure using approved components… and…
   2. Meeting the license and business obligations

Determine Policies

- Software development supply chain management
  - Open source, vendor, partner, contractor, outsourcers, other internal organizations, …
- Define criteria for approved software
  - Licenses
  - Sources
  - Support
  - Other
- Define criteria for unapproved software
- Define conditions for participating in the Open Source Software development
- Employee Education
  - No compliance without education

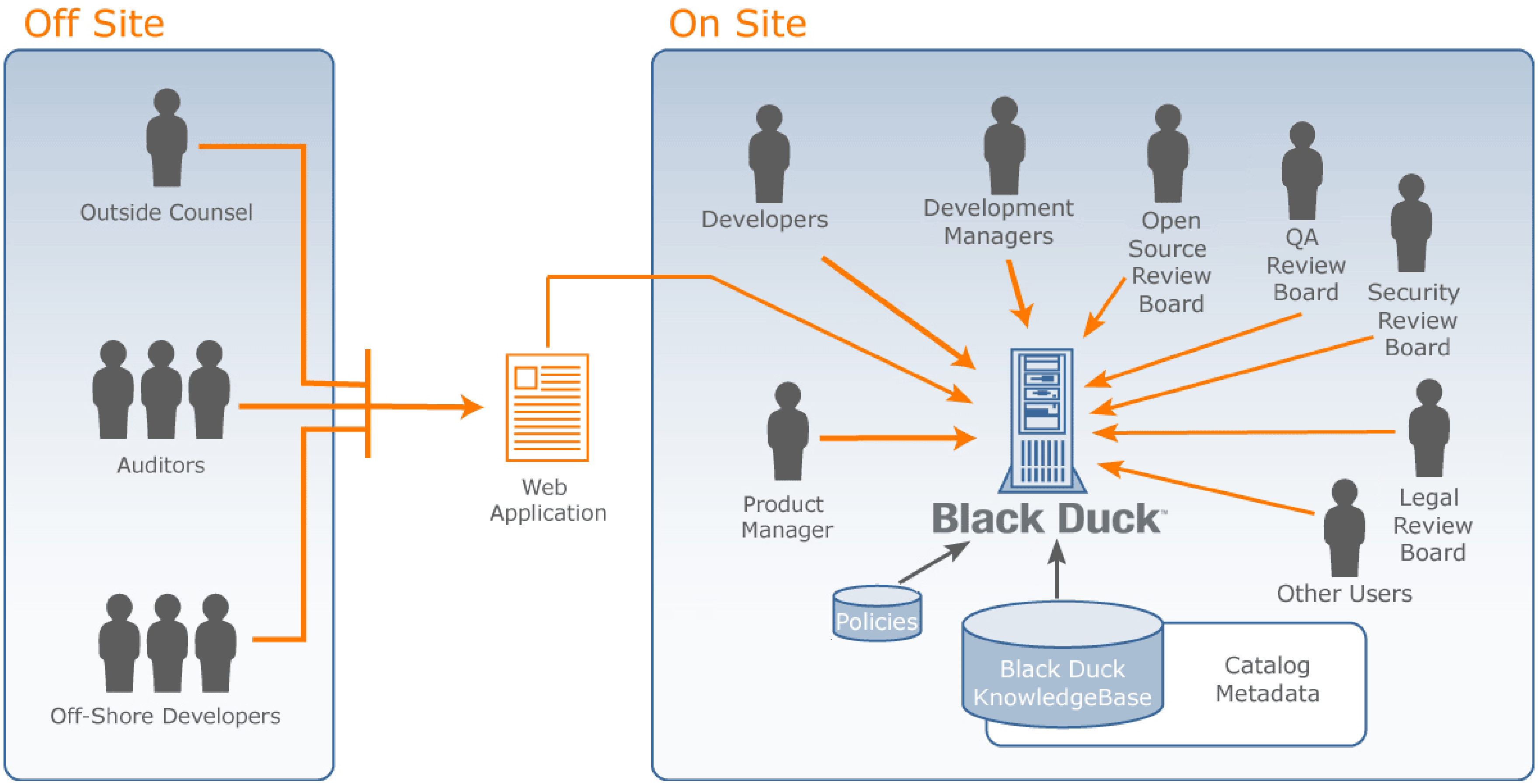
Select a Compliance Core Team

- Legal
  - Perform iterative review of identified components
- Open Source Process Owner
  - Appoint a person with overall responsibility
- Business / Product Perspective
  - Prioritize products (by release) for analysis
- Technical / Lead Architect
  - Integrate analysis and review with the development process
  - Identify code based on automated discoveries
- Project Management
  - Coordinate resources
  - Drive the project plan
  - Resolve issues

Establish Process

- How development teams and other functions
  - Search, select, approve, track, validate, track & monitor
- Inbound approval processes
  - Code from internal teams, external sources
- Outbound compliance processes
  - Distributed code
- Create a Baseline
  - Prioritize
  - Perform code analysis
  - Plan remediation
  - Document the origins of the code base
  - Determine all components and licenses in use
  - Verify usage is approved
  - Create a catalogue of approved components and licenses
- Validation processes

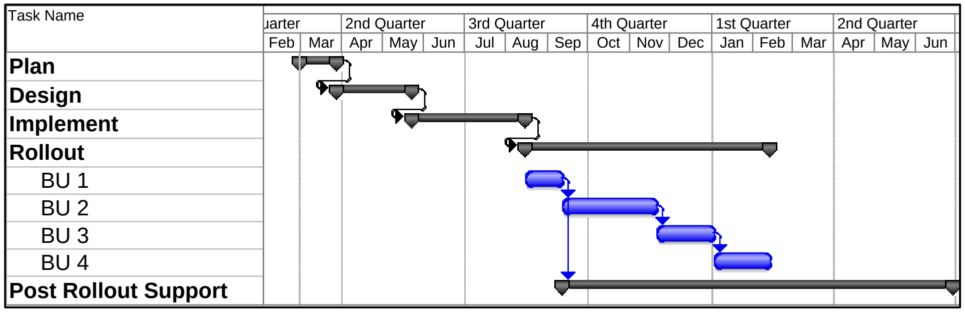
Enterprise Collaboration



Global Rollouts Require a Project & Sponsor

Many methodologies work, but typically they have:

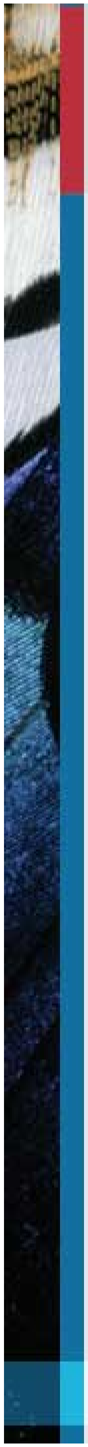
- •Plan
- •Design
- •Implementation
- •Rollout



Task Name
Plan
Design
Implement
Rollout
BU 1
BU 2
BU 3
BU 4
Post Rollout Support

May require a pilot and stakeholder approval:

- •Global Process
- •Implemented in Multiple Business Units



Implementation Deliverables and Phases

1 Design Phase

- Identify server topology
- Create deployment plan and articulate integration points
- Define test plan

2 Development Phase

- Deploy to pilot group
- Customize the application and the reporting features
- Present test results

3 Deployment Phase

- Deploy client application to end-users
- Connect external applications through integration points
- Disseminate policies company wide

4 Post-Deployment Phase

- Manage support issues
- Poll end-user experience

Phased Deployment Plan

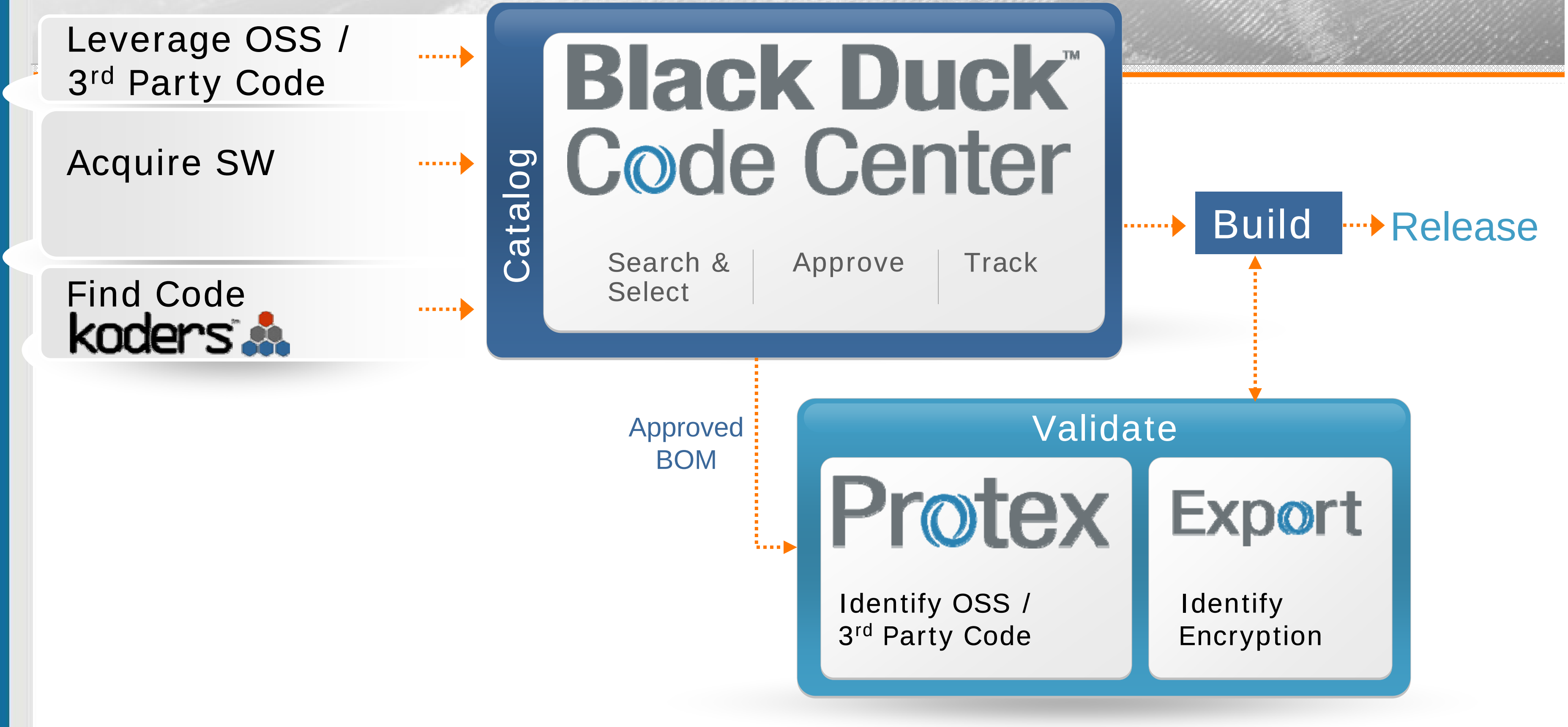
Depending upon need, you can designed phased deployment plans to quickly yield value

- Code Baseline
- Remediation Work
- Ongoing Analysis
- Hardware Topology
  - Scaling across sites, users, products
- Training
  - Technical, legal
- Policies, Process, and Resource Definition

Quick way to yield value

Audit what you have to help determine policies

Effective Management of Components



The Black Duck KnowledgeBase

Product Portfolio Foundation

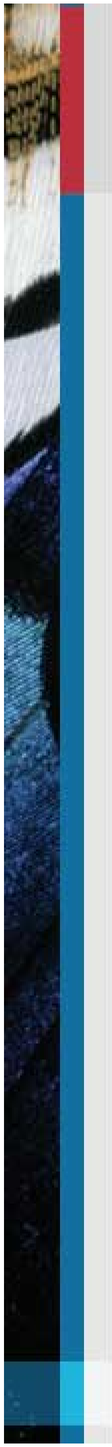
Comprehensive open source database

- 170,000+ OSS projects
- From 3,600+ sites
- Spanning 560+ million files
- Tens of billions of lines of code
- Released under 1,400+ unique licenses
- 31,000+ security vulnerabilities

Extensive metadata

- Name, description, versions, URL
- License, programming language, OS
- National Vulnerability Database
- Cryptography
- Code Prints of source/binary
- Other information

- Continuously expanded
- Custom Code Printing to add proprietary code
- Daily security vulnerability alerts
- Updates issued 1-2 times per month



Black Duck Professional Services

Deployment Services

Enablement Driven

- Software Implementation Services
- Strategic Planning
- Project Implementation
- Custom Development
- Training
- Integration

Assessment Services

Event Driven

- M&A Due Diligence
- Funding Event
- OEM Agreement
- Internal Audit
- Vendor Assessment

GOAL

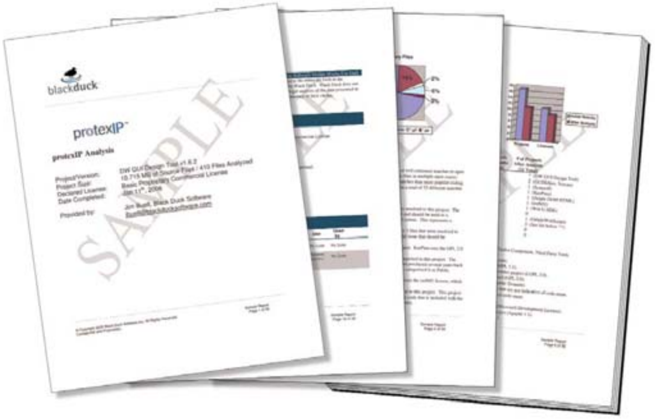
Customer Success

GOAL

Help Evaluate Risk

Assessment Services

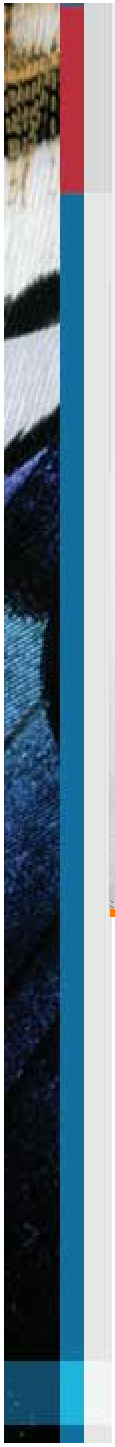
- Short-term engagements
- On-site or remote analysis
- Work with a variety of security, confidentiality and privilege requirements
- Report based
  - Executive Summary
  - Detailed Discoveries
  - Potential Risks, Conflicts
- Service is embedded in several customer's (M&A / OEM) processes



Example #1: Code match to DelphiWebScript project

File Location: [illegible]

In addition to matching to this project, the file also contained the copyright notice as well. This project uses the MPL 1.0 license. Developer(s) stated that they had used the code in the DW GUI Design Tool.

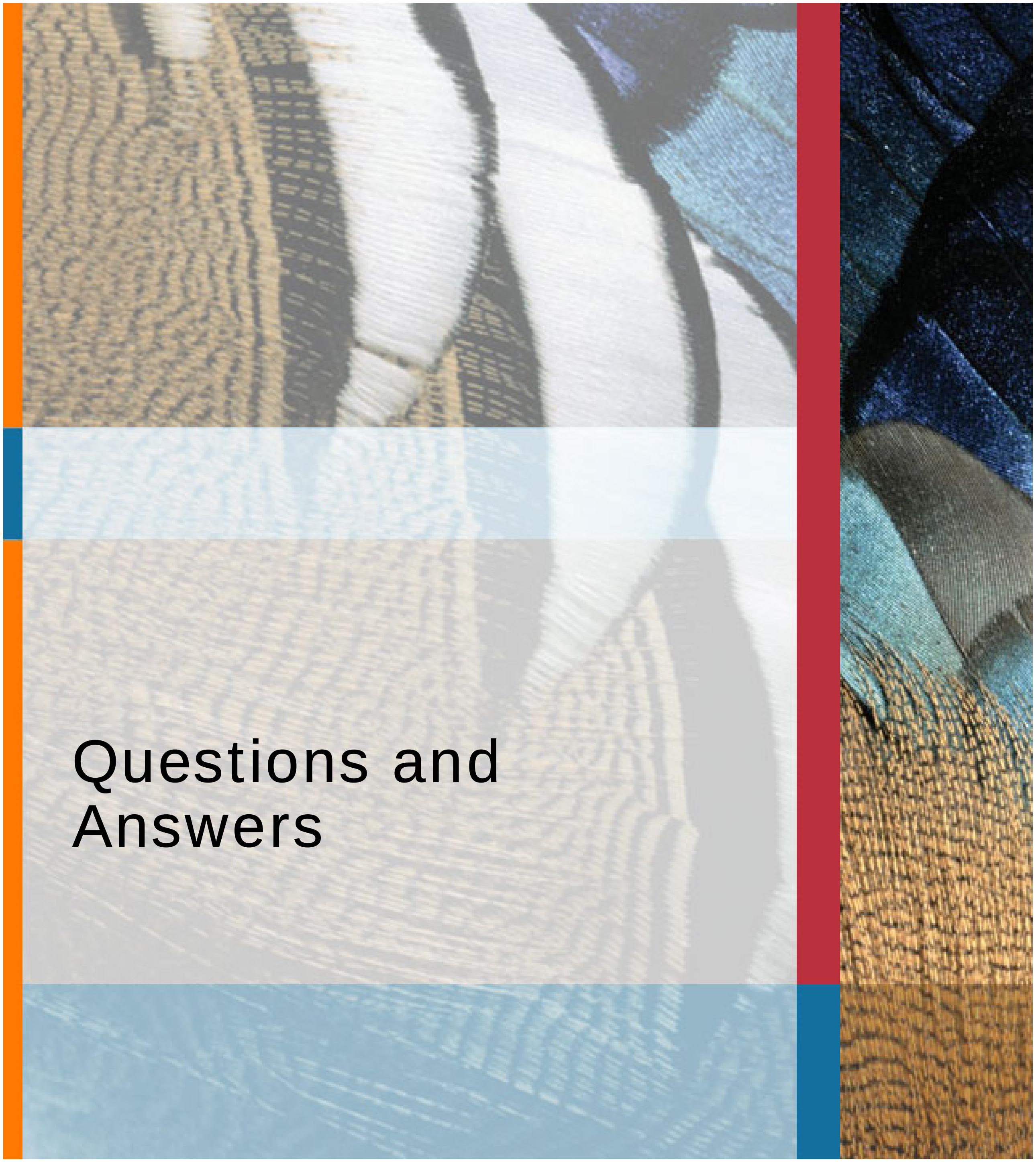


Put Black Duck Software to Work

- Accelerating software development by enabling you to better leverage open source
- Helping you avoid the pitfalls of mixed code development
- Managing your open source approval process
- Revealing the *unknowns* in your software

Know Your Code.™

blackduck™



Questions and Answers

Next Steps

- Black Duck Knowledge Center

  http://www.blackducksoftware.com/resources

- Black Duck Open Source License Resource Center

  http://www.blackducksoftware.com/oss

- For more information, email:

  info@blackducksoftware.com

blackduck tm

Thank You for Attending