

---

# Policy Driven Continuous Software Intellectual Property Management

DETECT – LOG – IDENTIFY – REPORT

**Session # 227**  
**Tuesday 13:30**  
**Room 207**

**Richard Mayer**  
**Protecode Inc.**  
[mayer@protecode.com](mailto:mayer@protecode.com)

**Come see us at Pedestal 105**  
or [www.protecode.com](http://www.protecode.com)



# Introduction and Agenda

---

- **Background and problem → Software IP management**
- **Policy → approved vendor list**
- **Pedigree → software BOM (bill of materials)**
- **Integration into development environment**
  - **Remove impediments to the use of Open Source**
  - **Identify and track external IP within a software project**
  - **Free developers from worrying about license issues**

**Practicing Safe Software Development**



# Software Licenses

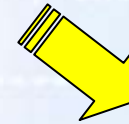
- **Commercial**

- Evaluation
- Run time
- Export restrictions
- Application restrictions
- Fees \$\$
- Typically only object code released

- **Open Source**

≠ **FREE**

Permissive & Restrictive



- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>– use as needed</li><li>– limited rules</li><li>– Eg. BSD</li></ul> | <ul style="list-style-type: none"><li>– copy left</li><li>– derivative works</li><li>– Eg. GPL</li></ul> |
|---|--|
- 
- Many incompatible licenses
  - Most rules associated with modification and distribution
  - Growing # legal actions
  - GPL Aferro has restrictions even for delivery of hosted service



Many trade-offs and choices → need education and policies

# Software Development

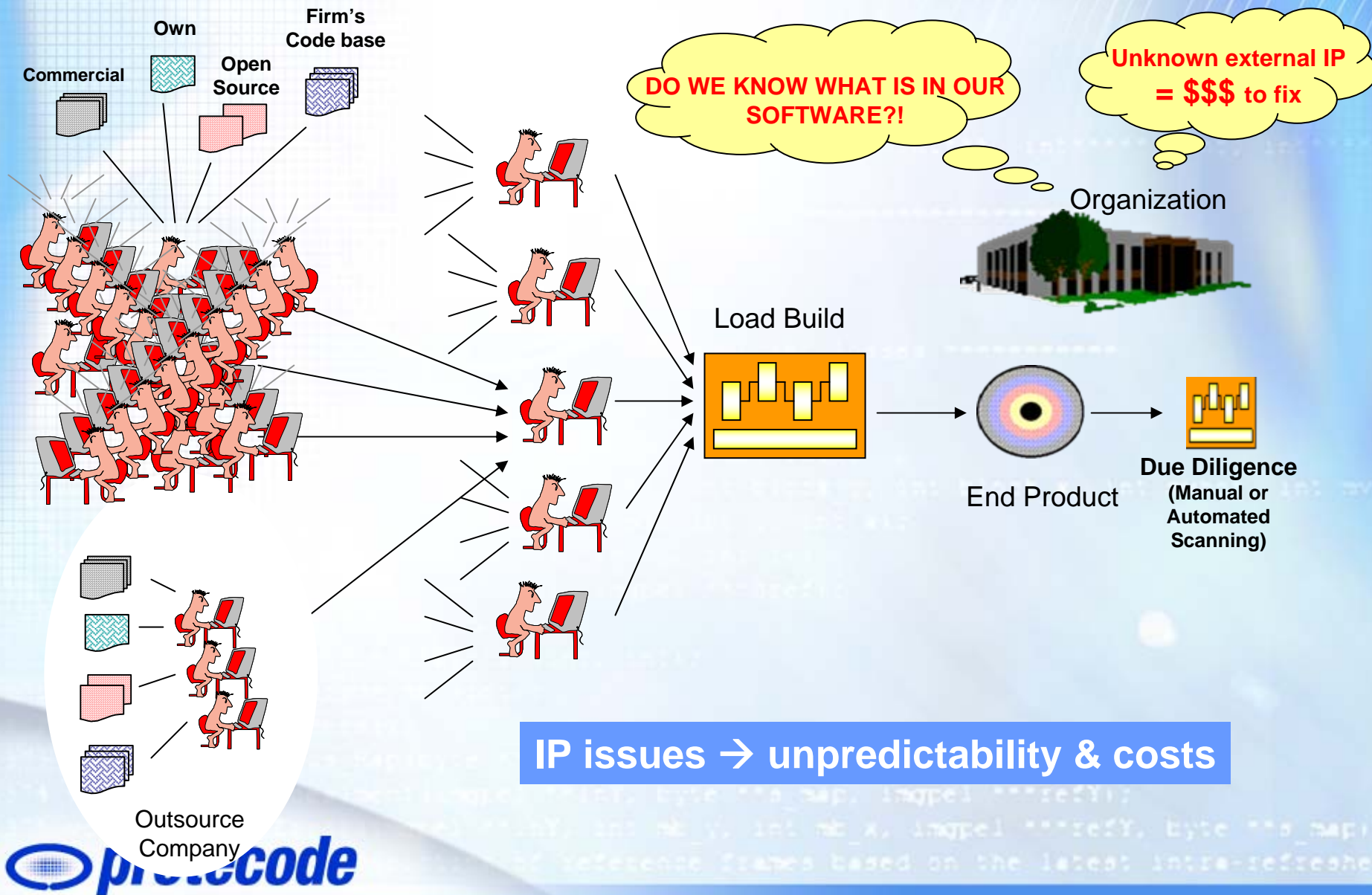
---

- **Software Development is more than writing code**
  - Combination of packages
  - Open source
  - Proprietary
  - Commercial
  - Creation
- **Re-use is important for profitability and efficiency**
- **Distributed Teams**
  - Multi-site
  - Outsourcing
  - Collaboration
  - Barriers – geographical – time – language

**There are more than enough challenges for developers today  
without worrying about IP license issues**



# Problem illustrated



# Open Source Success

---

- **Market trends and transition**

- Open Source is one of Gartner's top 10 trends for 2008  
[+http://www.gartner.com/it/page.jsp?id=593207](http://www.gartner.com/it/page.jsp?id=593207)
- 70% of large firms are seeing moderate or major benefits from open source  
[http://www.cio-weblog.com/50226711/the\\_growth\\_of\\_open\\_source\\_software\\_in\\_organizations\\_optaros.php](http://www.cio-weblog.com/50226711/the_growth_of_open_source_software_in_organizations_optaros.php)
- Open Source usage is growing at 43% CAGR (Gartner Feb/07)
  - By 2011, at least 80% of commercial software will contain open source code  
Gartner: <http://www.networkworld.com/news/2007/092007-open-source-unavoidable.html>
- Eclipse success and growth
  - 66% Enterprise developers use Eclipse (BZ Research Nov/07)
  - Eclipse downloads growth

**Need for Software IP management  
grows with Open Source Success**



# Why is Software IP important?

---

- **Intellectual Property Management impacts enterprise value**
  - Software is significant source of effort and investment
    - Exposure to third party rights
    - Protecting own rights
- **Impact on commercial transactions**
  - Representations and warranties, IP indemnity clauses
  - Real or *perceived* IP risks:
    - Could delay or lose software sale
    - Could lower M&A value
- **A company's existence could depend on it**
  - Litigations, law suits, penalties, out of court settlements
    - Dlink, Cisco-Linksys, SCO-IBM-Novell-Autozone, Verizon

**Growing industry awareness of issues**



# Code IP Contamination

---

- **How does software become contaminated?**
  - Code repository of organization or outsourcer may have impure artifacts
  - Open Source components do not satisfy the policy
  - Derivative works may be tainted by previous restrictive licenses
  - Collaboration partner adds undesirable code
  - Outsource partner has cross project contamination
  - Improper license (evaluation, expired, restricted)
- **Extent of software contamination is increasing :**
  - 70% of software developers carry code from one gig to the next  
(ZD Net <http://news.zdnet.co.uk/software/0,1000000121,39156544,00.htm> )
  - Google code search makes it simple to find software
  - Growth of outsourcing practices - offshoring, e-bidding
- **Growing up with the culture of plagiarism**
  - “rip, mix, burn”, “cut & paste”
  - Instant information access, blurring what’s yours & what’s not ...

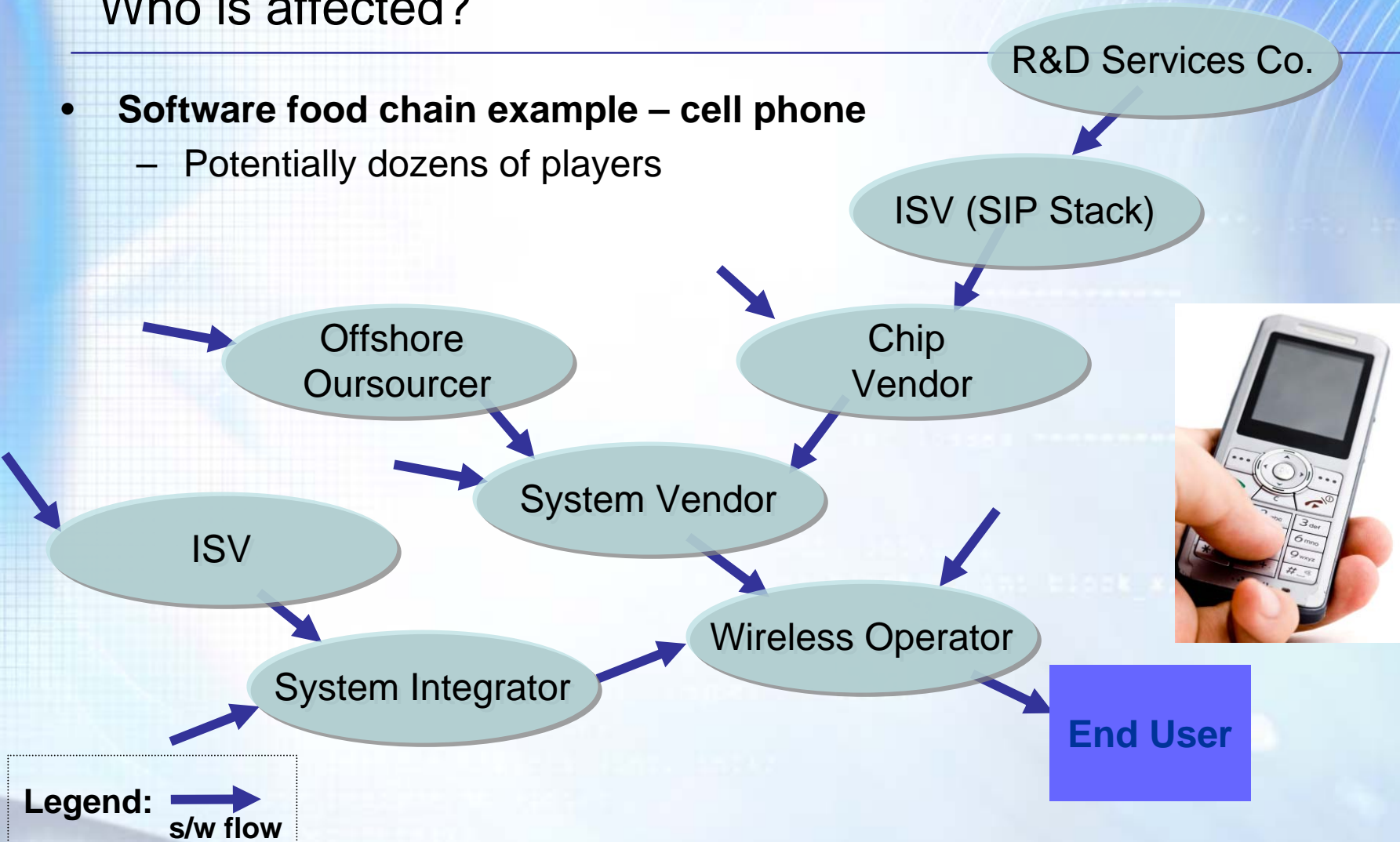
**Most code contamination is unintentional**





# Who is affected?

- **Software food chain example – cell phone**
  - Potentially dozens of players



Anyone who makes, buys or sells software is affected by software IP



# Current Solutions

- **Education**
- **Spreadsheets**
- **Code Reviews**
- **Prohibiting Open Source Use**
- **Post Processing Scans**
- **Don't think about it**
- **Integrate it into the development process/culture**



**The earlier IP issues are addressed → the better**

# Lawyers and Developers

---

- **High tech professionals**
  - Both highly educated
  - Both typically overworked
  - Both fun at social events or as golf partners
- **BUT they speak two different languages**
  - For software licensing issues it's best if lawyers and developers don't need to talk about it !
  - Corrective analysis requires lawyers to deal with developers



The best solution is a preventive policy-driven process



# IP Policies

---

- **Central focused team to define policies**
  - Legal, business and technical input
- **Define clear policies**
  - Approved Vendor List
    - What is allowed and what is restricted ?
  - Project relationships
    - Who can do what and how often?
  - Re-use parameters
    - Goals and guidelines
  - Export restrictions
    - Which software goes into which product sold into which country ?

**Clear policies can enable constructive use  
and adoption of open source**

# Preventive vs. Corrective

- **Methods of managing IP contamination fall into two groups**
  - **Corrective**
    - Try to detect contamination or policy violations in a piece of software
  - **Preventive**
    - Strive to stop unintentional penetration of desirable code in a project

	<i>Manual</i>	<i>Automated</i>
<i>Preventive</i>	<ul style="list-style-type: none"><li>❖ <i>Education, Ethics &amp; Practices</i></li><li>❖ <i>Use only known code</i></li></ul>	<ul style="list-style-type: none"><li>❖ <i>Commercial</i></li></ul>
<i>Corrective</i>	<ul style="list-style-type: none"><li>❖ <i>Commercial Due Diligence Service Companies</i></li></ul>	<ul style="list-style-type: none"><li>❖ <i>Academic</i></li><li>❖ <i>Commercial</i></li></ul>

**An ounce of prevention is worth a pound of cure**



# Four steps to manage external content

---

- **Detect external content** as much as possible
    - Libraries
    - Files – drag and drop, import, etc.
    - Code snippets – cut & paste
  - **Log**
    - Keep accurate records
  - **Identify**
    - Determine and annotate license info
  - **Report** → make info readily available
    - Different audience
    - Different stages
- } Provide complete information for decision making

**Detect – Log – Identify – Report**



# Steps for identification of external content

---

- **Compare to other known content**
  - Internal
  - Commercial
  - Open source
- **Annotate with additional information**
  - Some software can have multiple licenses
  - Source URL where software/license can be found
  - Metadata – about author, copyright etc.

**As much information as possible  
to help making decisions on risks & obligations**

# Integrated into development environment

---

- **Continually and automatically manage software IP**
  - Centrally define IP policies
  - Seamless integration with development process to maximize detection of external content
  - Record information in detailed logs
  - Identify code source and license in real time
  - Monitor and enforce or report policy violations
  - Provide timely reports and feedback
- **Benefits**
  - **Solve issues right as they show up to reduce costs**
  - **No wasted time on integration of contaminated code**
  - **Reduce risk of project or revenue delays**
  - Free developers and project leaders from requiring to understand license and policy details
    - **Reduce training**

**Unobtrusive policy driven continuous software IP management**

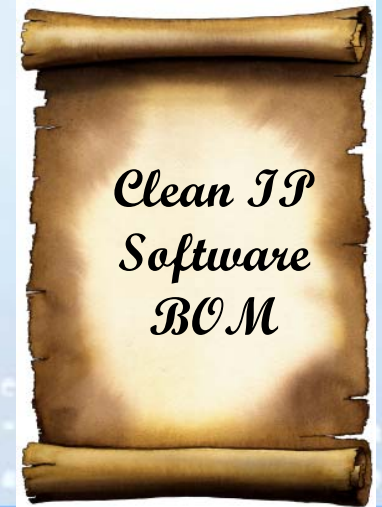




# Pedigreed Software – valuable for Open Source

- **What is a software pedigree?**
  - Software with a pedigree – is software that carries with it a certificate of it's IP history and bill of health
  - Keeps track of ownership, authorship and licensure
  - BOM – a software Bill of Materials
- **Software Pedigree enables applications:**
  - Committing software to an open source community together with it's pedigree makes submission/approval process streamlined and faster
  - Checking out open source software from a repository that includes the pedigree makes it easier to understand implications and increases confidence
  - Collaborative software development including software pedigree clearly defines background and foreground IP

**Know what's in your software – all the time**



# Summary

---

- **Exciting times**
- **Software intellectual property is important**
- **Practice Safe Software Development**
  - Software development best practices
  - Centralize policy management → approved vendor list
  - Software Pedigree → Bill of Materials
  - Minimize training effort
  - Keep projects on time
  - Maximize commercial results
  - Free developers and lawyers from interacting

**Policy Driven Continuous Software  
Intellectual Property Management**



---

# Thank You !

Want to continue the discussion?

Richard Mayer  
Protecode Inc.  
[mayer@protecode.com](mailto:mayer@protecode.com)

Come see us at Pedestal 105  
or [www.protecode.com](http://www.protecode.com)

