IT Project Management Challenges with Open Source

George A Pace

Tonight's agenda

- Two parts to the Presentation
  - What is Open Source ?
    - A background primer on the key elements of Open Source .
    - A specific focus on the risk / benefit profile of Open Source which talks to a critical role of project management – managing risks
  - Impact in a typical Lifecycle
    - Focuses on the key elements talked about in the introduction section, and what it means during the normal SDLC.

About the presentation

- This is NOT a technical presentation - Really !!!
  - An overview of what Open Source really is (in PLAIN ENGLISH)
    - A definition of what makes something “Open Source”
    - The key conditions and terms you should know
    - A little bit about the business value
- With a high degree of certainty, I can tell everyone in this room
  - Most of you have encountered Open Source in at least one project within the past 5 years
    - You just might not have been aware of it !!
  - You are probably going to encounter MORE Open Source in future projects
    - So you really need this presentation !

About the Audience

- Just to make sure we are all on the same page, the amount of technical detail covered in this presentation consists of:
  - The Software Development Lifecycle
  - Source Code
  - Application Library
  - Application executable

- And for my benefit, where the audience is in terms of:
  - Who has heard of the term “Open Source”
  - Who thinks they know what Open Source is ?
  - Who thinks they have used Open Source on a project ?
  - Who thinks the development team tells you everything they are doing ?

So what is Open Source ?

- As the name suggests, the underlying source code is “open” for all to see.
  - But there is much more to it than that
  - There is a formal definition of Open Source
    - http://www.opensource.org/osd.html
      - It defines 10 attributes that make something Open Source that focus on
        - » Access to the Source Code
        - » Independence, Licensing

- Open Source is different than Freeware or Shareware

- While free, “Open Source” is *distributed under a specific license.*

- For all practical purposes, there is an Open Source alternative for most every commercial software product
  - Just because its available doesn’t mean its going to fit your needs
    - Requirements and functionality still matter
      - Lets take a second and talk about 80/20 or 60/40 for that matter

  - You will find Open Source across a wide range of functionality
    - Infrastructure components (Linux, Application Servers, Databases)
    - Business applications (OpenOffice, CRM, ERP)
    - General utilities (Graphics, Editors)
    - Application libraries (Struts, Spring, Log4j)

Why Open Source is important

- It's a change agent
  - It has had a major impact on commercial software vendors
    - Some vendors have changed product strategies
      - Express versions of products (Microsoft & IBM)
    - Some vendors have gone out of business (or were acquired)
      - Remember Borland International ?
      - Remember Sun Microsystems ?
    - Some vendors are purchasing commercial Open Source companies
      - Citrix purchased XenSource
      - VMWare purchased SpringSource

- What is driving this change
  - The current economy
  - The maturation of Open Source in general

- It offers companies financial opportunity AND flexibility
  - Think about the value of "commoditization"
    - Microsoft Word / Band aides
  - Think about the long term annuity you pay to commercial vendors

- Thought: It if its not a commodity today, where will the technology be in 3-5 years ?
  - Gets back to the 80/20 (or 60/40 idea)

Where can you find Open Source?

- In a word .. EVERYWHERE
  - Linux is a term/ technology you may be have heard ..
    - And if you haven't – its an Operating System similar to Unix

- But did you know it was used in these products ?
  - Consumer Products
    - The Android telephone
    - Dell Streak (New tablet)
    - Linksys Router
  - Commercial Products
    - IBM Web Sphere products
  - Financial Institutions
    - Chicago Mercantile Board of Exchange
  - Healthcare
    - https://medsphere.org/download/
    - http://www.patientos.org/
  - Automotive Industry
    - http://www.linuxtoday.com/news_story.php3?ltsn=2009-10-30-001-39-OS-BZ-DV
  - Government
    - http://goscon.org/

- And the list goes on … and on … and on ….

The key elements you need to know

- What is an Open Source project
- How Open Source enters an organization
- How Open Source can be used in a project
- Open Source license types
- The Benefits of Open Source
- The Risks associated with Open Source

What is an Open Source project?

- Where you get Open Source software from
  - Can have multiple elements to it
    - Copyright Ownership (i.e. Linux)

- What makes up an Open Source project
  - Sponsors of the project
  - Individuals that participate with the project
    - Some work for free
    - Others are sponsored by a company
  - The source code repository
  - The underlying processes that are in place
    - Meritocracy
      - How people are promoted (key word – COMMITTER)
    - Build / Testing
    - Support provided by the project

How Open Source enters an organization

- The first step of the journey is understanding how Open Source enters an organization
  - Downloads from the Open source project (or foundation)
    - 2 common examples are
      - www.apache.org
      - Sourceforge.net
  - "Commercial" Open Source
    - A "product" that is created from one or more Open Source projects
      - Redhat Linux
  - Packaged with Commercial products
    - Consists of both proprietary and "open source" components
      - IBM WebSphere and Rational Application Developer products
- Why is this important ?
  - It is one of the first risk elements you encounter
  - Some organizations believe they are preventing the use of Open Source
    - Because they may not understand all of the ways Open Source can appear

How Open Source can be used in a project

- Infrastructure Components
  - Core elements applications run on top of
    - Operating Systems, Application Servers
- General Business Functionality
  - Functionality specific to a business or business function
    - Business Intelligence (Pentaho / Jaspersoft)
    - CRM (SugarCRM)
- Utilities
  - Tools and utilities used on a workstation
    - Office Suites / Developer Utilities
- Application Libraries
  - Code that gets included as part of your application

Open Source license types

- Depending on who is counting (and how!) there are over 200 Open Source licenses
  - Early days of Open Source led to various flavors of existing licensees
  - In some cases, there are multiple versions of a license
    - Apache
    - GNU Public License (GPL)
- Fortunately, licenses tend to fall into 2 camps
  - Viral and non-Viral licenses
    - Viral – Usage of components with a viral license can "taint" proprietary source code
    - Non-viral – Does NOT "taint" other proprietary components
- Terms YOU NEED TO KNOW
  - Modification
  - Distribution
  - Derivative Works
- Note: This where your legal team can help understand the differences / nuances

Benefits of Open Source

- Financial Benefits
  - Open Source provides a flexibility that you do not have with commercial software products
    - Initial licensing costs
  - Support flexibility
    - You have the option to pick the support option that best suits the enterprise / project needs
      - You have several support options to consider / choose from
- Opportunity Costs
  - Reduce implementation time
    - As contract negotiation is alleviated, you can implement solutions quicker
    - Product is already developed and ready to use
  - Staffing
    - There is a growing community of developers that are well versed in Open Source products

The risks of Open Source

- Balancing the Benefits are a series of Risks – and how those risks apply depends on your usage
  - Companies that develop / sell / license software for a living (ie IBM, Microsoft etc)
  - Companies that USE software as part of what they do (everyone else !)

- For a native Open Source project, there is no company to enter into a "contract with.
  - No warranties of any kind are available
  - No indemnification is available

- The concept of "One Throat to Choke" doesn't exist

- If you sell product that uses Open Source you need to be careful what you are using and what you are doing:

  - Linksys Case (http://www.wi-fiplanet.com/tutorials/article.php/3562391/The-Open-Source-WRT54G-Story.htm)
    - Linksys open sourced the WRT54G firmware in July 2003

  - Dell Streak (http://www.extremetech.com/article2/0,2845,2369687,00.asp)
    - Dell has published the source code behind the the Dell Streak, after some online protesters claimed that it was violating the terms of the GNU Public License

End of “part 1”

- The items discussed should give you a general idea of what Open Source is
  - How it works
  - The Risk / Benefits of using Open Source
- Any quick/pressing questions before continuing?

Open Source in the project lifecycle

- Taking the concepts discussed in the previous section lets apply them to the Project Lifecycle
  - Plan Phase
    - Management Approvals / Legal Approvals
    - Conflicts with enterprise standards
    - How products are selected / acquired
  - Build Phase
    - Modification, Developer “religion”
  - Run Phase
    - Support, End of Life, Security Patches, Maintenance

Planning phase

- Before starting with the Business Requirements
  - Is the company open to Open Source ?
  - What are the company policies / standards specific to Open Source ?
  - Does any type of Open Source office and/or central contact person exist ?
  - What Open Source may already be used in the organization?

- Reviewing the Business Requirements
  - What are the budgetary constraints?
    - Are there possibilities to leverage Open Source solutions for any part of the project ?
  - What is the prevailing “religion” of the technical teams
    - Some people have very strong beliefs as to the tools they are going to use
  - Do you need anything special from the Operational Teams
    - Do they know anything about Open Source
    - What lead times do they need to get your environments up and running

- Related to-do’s if the team is going to use Open Source:
  - Find out what resources are available for reviewing the underlying Open Source licenses and how the tools will be used.
  - Who decides what versions to use, which products are acceptable
  - If you are going to use Commercial Open Source
    - Who reviews the license
    - How will you acquire
    - How many do you need

Development phase

- Opportunities abound to use Open Source during the development phase

- Development Tooling (Low Risk)
  - This is usually the domain of the development team and few if any people get involved at this level
    - Tools built on top of the Eclipse Framework are VERY common and VERY popular
  - Risk Mitigation Company Approvals / Company Standards
  - Benefits: Cost savings / Time to Market

- Team environment (Low Risk)
  - Build processes, Bug Tracking, Continuous Integration (Low Risk)
  - Risk Mitigation Company Approvals / Company Standards
  - Benefits: Cost savings / Time to Market

- Application Functionality (High to Highest Risk)
  - Application Libraries
  - Risk Mitigation Company Approvals / Company Standards
  - Risks: How do you make sure the Open Source code being used hasn't been modified ?

Production phase - Part I

- Now that you are ready to go “live” ….
  - What are the preferences for support ?
    - You have 4 options
      - Depend on support from the native Open Source project
      - Purchase a “commercial” version of the product
      - Use a 3rd party support company
      - Internal expertise
    - Which option(s) you select depend on application requirements
      - What a concept !!!!
      - You can usually start with
        - » Mission critical vs. non-mission critical and go from there

Production phase - Part II

- The next set of decisions get back to the concept of “maintenance”
  - How will the team maintain Open Source components
    - Remember, Open Source projects can change *frequently*
      - You need to develop an approach for the project
        - » How do you track, when do you modify
  - What does “End Of Life” when it comes to Open Source components
    - It is easy to get old versions – so what does “End of Life” mean to your project
    - It is unlikely this is tracked at an enterprise level
  - How will you identify Security vulnerabilities?
    - Open Source components have vulnerabilities just like commercial software
    - Vulnerabilities are tracked at :
      - » http://osvdb.org/

Summary

- Expect to see more Open Source – not less !!!
  - Don’t be surprised to see resistance from management
  - Don’t expect all companies to understand the risks
- Open Source has risks that need to be managed just like any other project
  - It you don’t have the proper controls for the right use case, it can cause larger issues ..
- Understanding Open Source and its applications can differentiate YOU from your competition
  - Every little bit helps ……
- And just one more thing ..

What you can do next

- Consider getting involved with an Open Source project
  - Projects look for people other than developers
    - Project Managers, Documenters etc
- Recommended projects
  - FOSSBazaar is a good resource for keeping abreast of current Open Source trends / topics / events
    - www.fossbaazar.org
  - Not an "Open Source" project per se - but likely to use a number of Open Source components
    - Fundamentally, it's all about helping American cities use web technology to do a better job of providing services to citizens.
    - www.codeforamerica.com