



Risks and Rewards of Open Source Software

Deloitte.

13 July 2011

David Sisk Deloitte Consulting Bio

Current



Joined Deloitte in 2005

- Designed and code Java Application Development Framework
- Authored Deloitte Point of View on Open Source Software
- Technology Blackbelt for Java Enterprise Development

Currently work in several sectors

- **Commercial** – Lead architect on Life sciences and Health Care, Consumer Retail
- **State Public Sector** – Lead architect on state HHS project
- **Federal Public Sector** – Lead architect on IC personnel security and supply chain risk management projects

Education



University of Virginia

MS Management of Information Technology

University of MD University College

BS Computer Science/Finance



Previous Experience



• 15 Years Industry

AT&T
Convergys

• 6 Years Boutique Consulting

Magnet Interactive/AKQA
Dante Consulting
NumberSix Software/ATSVA



Open Source Takeaways

Rewards



- Open Source can have up-front cost savings
- Provides alternatives to commercial software
- Acquisition without cost is significantly easier
- More choices can be available
- Open Source can be much more transparent

Risks



- Uncontrolled adoption will cause problems
- Support and training can be challenging
- Integration is additional burden on consumer
- Very small percentage of Open Source Software is successful

Vendor Fees Do Not Guarantee Success

Two commercial software case studies, because vendors do not advertise these kinds of things

- Vendor A was so confident in their messaging product, they bet the client the costs of professional services to fix the problem.
- Vendor B responded to significant performance problem in their ecommerce product that no other client had the issue and therefore it could not be a problem with their product.

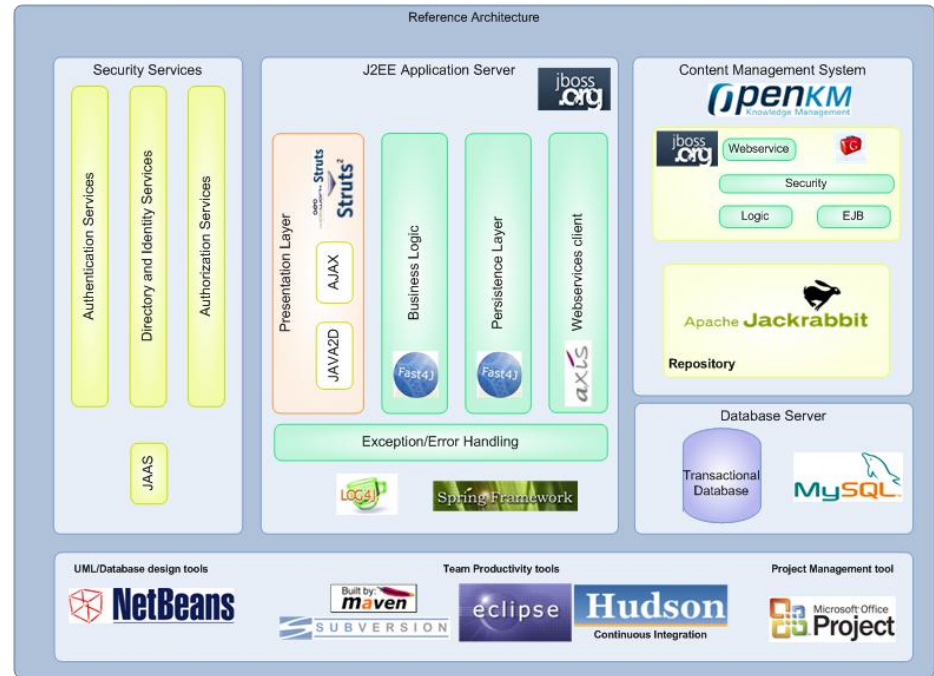
Replatforming Case Study

Up-front Software Costs

| | |
|-----------------|------------------|
| Graphics | \$ 24,000 |
| App Server | \$115,000 |
| <u>Database</u> | <u>\$144,000</u> |
| | \$283,000 |

Annual Software Costs

| | |
|-----------------|-----------------|
| Graphics | \$ 4,000 |
| App Server | \$17,000 |
| <u>Database</u> | <u>\$22,000</u> |
| | \$42,000 |



| Product | Year 1 | Year 2 | Year 3 | Year 4 | Year 5 | Total |
|-------------|-----------|----------|----------|----------|----------|-----------|
| Vendor | \$283,000 | \$42,000 | \$42,000 | \$42,000 | \$42,000 | \$451,000 |
| Open Source | \$0 | \$0 | \$0 | \$0 | \$0 | \$0 |

Deloitte ported application to Open Source JavaEE stack for Pharma client

Open Source Can Be a Negotiating Tool

State Public Sector Project

- We suggested Mule for Integration and Pentaho for Data Services.
- Open Source alternatives were used as a negotiating strategy against commercial product suite.

Results

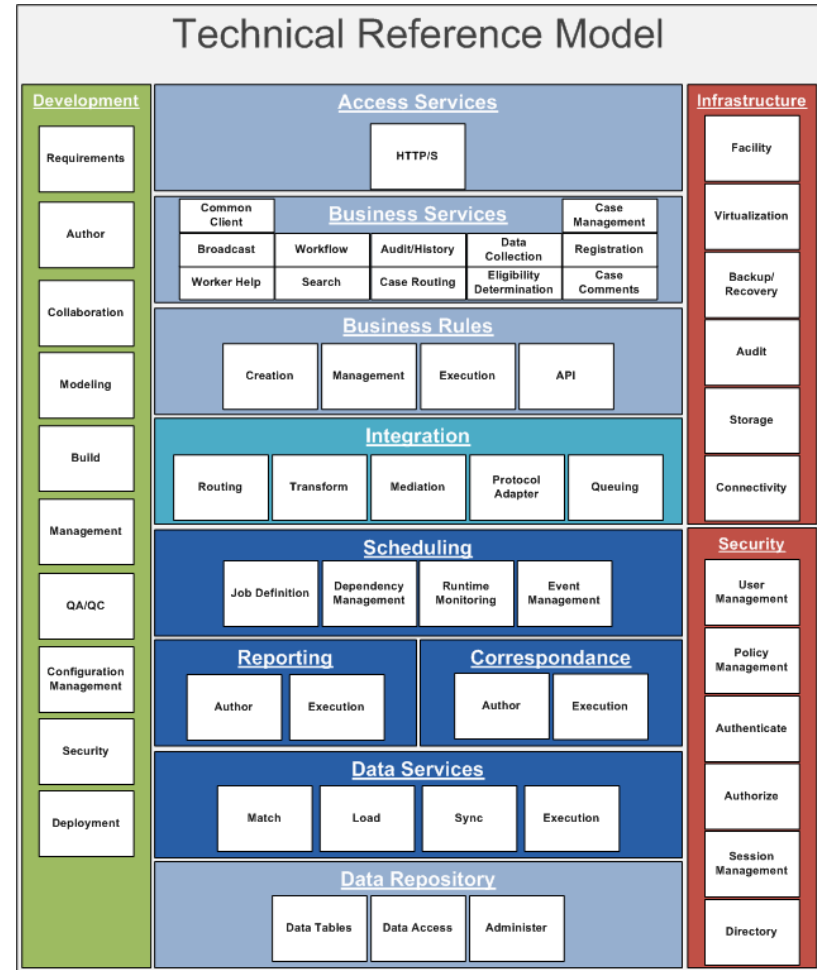
Proposed Software Costs

\$750,000

Final Software Costs

\$225,000

Year 1 Savings \$525,000



Deloitte negotiated software license agreements for State HHS client

Open Source Has Less Beans To Count



- For many agencies, having no up-front cost means no procurement process. Support contracts can be negotiated separately and not delay the beginning of software development.
- The internet has accelerated software distribution from traditional physical models. Downloading software is free and easy.

These conditions have driven significant Open Source adoption in the Public Sector

But This Is Not Just About The Cost

Open Source Software Is More About Choices Than Cost

- Software authors can **give away** software, but **it is not** Open Source Software
- Software authors can **charge** for software but **it is** Open Source Software
- Software authors can **use** Open Source Software in a commercial product

...but if they do, they need to be aware of the license

Increased Transparency and Flexibility

Open Source provides a very different experience than commercial software

- Open Source Software can be modified to purpose
- Obscurity is not Security, Open Source code can be inspected for Certification and Accreditation activities
- Open Source support can be purchased and scaled to individual needs in an ala-carte fashion
- At least 10 choices of LINUX to choose from, so less vendor lock-in issues
- With insight into the source code, confidence is higher in end product



While I'm indeed a big fan of the open-source approach to software development, there are definitely situations where an open-source approach would not benefit the parties involved

Open Sources: Voices from the Open Source Revolution Brian Behlendorf

Brian was a primary developer of the Apache Web server, the most popular web server software on the Internet, and a founding member of the Apache Group, which later became the Apache Software Foundation

Architectural Oversight is Critical

- Without procurement controls, anyone can (and will) bring new software into the environment
- Uncontrolled adoption results in duplicated solutions
- New dependencies are created every time software is introduced
- Certification and Accreditation becomes quickly more difficult with lots of unanswered questions
 - Who owns remediation?
 - Who submits for ISO or DoD certification?
 - How do we reconcile requirement to make software available when deployed on classified networks?

The Industry Has Even More Questions

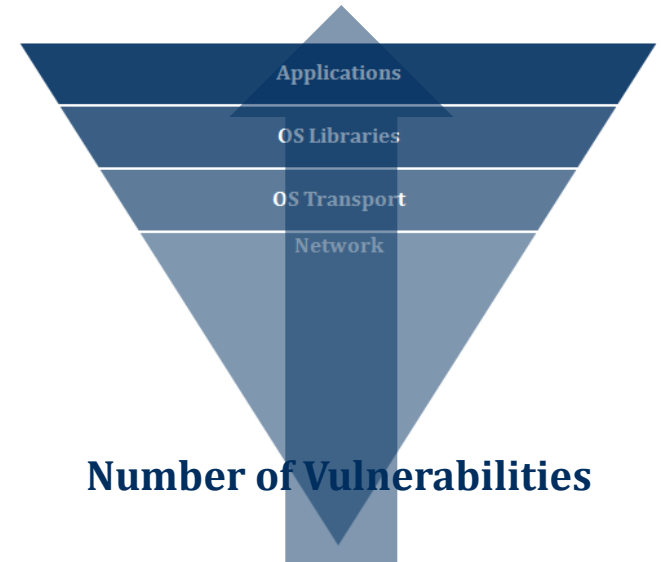
Potential consumers of Open Source Software have raised many valid questions about Open Source model

1. Is Open Source Software secure?
2. Who provides support and training?
3. Does Open Source provide enough features?
4. What if the developers abandon the project?
5. Who will integrate the products?

Open Source Is Not Inherently Secure

1. Is Open Source Software secure?

- Open Source anything is not inherently more secure, bad developers do not just work for commercial vendors.
- Widespread security attacks have always been a function of the number of targets available, not vulnerability of specific software



Recently situation where a client's servers were used for Denial of Service attacks, because developer left default password on the Open Source Application Server

Support and Training Can Be Sparse

2. Who Provides Support and Training?

- Open Source documentation quality and quantity can vary widely and is often out of date
- With a small set of volunteer core developers, support might not be a top priority
- Relatively few Open Source projects provide training courses. This can change due to investment or incorporation



Very popular products will attract enough interest for training and support, less so for the rest

Missing Features Can Be Showstoppers

3. Does Open Source provide enough features?

- Most common gap between Open Source and commercial is intuitive UI*
- Developers might not think new features are interesting enough to work on. With no carrot, or stick consumers have little influence
- GOS is an option, but development pool will be smaller if USG is only consumer



**Very popular products will attract interest and a UI will be created but it might not be timely*

Software Without Financial Interests

4. What if the developers abandon the project?

- 100's of thousands of abandoned software projects*
- Ventures dependent on just the enthusiasm of a few participants without financial interests can easily fail
- Commercial software vendors have structure in place to ensure ongoing life of software



**375,000 Projects on SourceForge haven't had a new release in 2 years*

Open Source Is Not Open Standard

5. Who will integrate the products?

- Product lock-in is no less a problem than vendor lock-in
- Integration between products is almost never a priority and usually non-existent and is the consumer's problem
- Even when Open Source is widely used, it might not be standard
 - Struts was not based on a JavaEE specification and could not be swapped out for another choice
 - Spring core framework was created as a way to avoid the JavaEE specification complexity
 - Prior to transparent persistence API, ORM solutions could not be substituted easily



Recommendations

| Question | Mitigation Response |
|---|--|
| Is Open Source Software secure? | Start C&A process early and make someone accountable. Educate and get buy-in from Information Assurance |
| Who Provides Support and Training? | Identify and understand the costs associated with training staff and ensure enough knowledge is in-house to support the technology |
| Does Open Source provide enough features? | Introduce Open Source like any other software, perform software portfolio rationalization and fit/gap analysis |
| What if the developers abandon the project? | Create contingency plans for abandonment, check in source code each release to your repository. Be prepared to own any Open Source Software you use. |
| Who will integrate the products? | Enforce architectural process and standards for Open Source introduction, upgrade and deprecation |

Bottom line is to be prepared to own any Open Source Software you use in your solution

Thank You

David Sisk
dasisk@deloitte.com
703-251-1188

Deloitte.