

# Open Source Software in an Industrial Setting: Why Open Source, and the Issues

Richard Huntrods, P.Eng.

17th IFIP World Computer Congress  
Montreal, P.Q.  
August 29, 2002

## Agenda - 1

- Introduction
- Background
- Problem Description
- Possible Solutions
- Chosen Solution
- Discussion of Solution Components

© 2002, Richard S. Huntrods

2

## Agenda - 2

- Development Environment
- Development Methodology
- Why Open Source?
- Demonstration
- Conclusions
- Lessons Learned
- Web Sites of Interest

© 2002, Richard S. Huntrods

3

## Introduction

- Who is Richard Huntrods?
  - P. Eng.
  - Working in IT since 1980
  - huntrods.com
- What is Huntrods Consulting Inc.?
  - Systems consulting since 1989
- What is Quick Test International Inc.?
  - quicktestinc.com

© 2002, Richard S. Huntrods

4

## Background

- The Quick Test System
  - a competency assessment and validation system
  - a time sensitive information system
  - a "Validated Resume" resume builder
  - a Prior Learning Assessment system

© 2002, Richard S. Huntrods

5

## Problem Description - 1

- The client wishes a robust client-server application that can run on the Internet, Intranets and Stand-alone PCs
- The client's existing application consists of forms talking to a database (classic C/S computing)
- Existing application uses ASP and MS SQL Server on Windows NT 4

© 2002, Richard S. Huntrods

6

## Problem Description - 2

- Existing application is Internet available, but cannot be ported easily to the other environments (especially standalone)
- Initial development costs for system is fixed.
- New system must be expandable, both in terms of users and capability

© 2002, Richard S. Hammonds

7

## Possible Solutions - 1

- Continue ASP / SQL Server / NT development
  - pros: known technology, common to many systems, available expertise
  - cons: Locked into Microsoft products and upgrade path, products are costly.
  - A “safe” solution.

© 2002, Richard S. Hammonds

8

## Possible Solutions - 2

- Java servlets / database / JDBC connectivity
  - pros: select from almost any hardware / OS / database choices, control over cost, vendor independence
  - cons: may be “bleeding edge”, support may not be the best, solutions may disappear
  - A possibly higher risk solution

© 2002, Richard S. Hammonds

9

## Possible Solutions - 3

- Other technologies
  - still need something to store the information (database)
  - still need technology to communicate with users (C/S model)
  - many possibilities for C/S model: Perl, PHP, VB, etc.
  - all have various pros and cons

## Chosen Solution - Hardware

- Sun SPARC systems (20's and E250)
- Intel systems (Pentium & PII)
- Multiple servers / firewalls
- ADSL internet connection

## Chosen Solution - Software

- Java Servlets (J2EE)
- MySQL database (opensource)
- "MM" JDBC driver (opensource)
- Apache Tomcat Servlet server (opensource)
- Apache Web Server (opensource)
- Solaris O/S (proprietary)

## Why Sun / SPARC?

- Sun is major Unix vendor with significant history and industry presence
- Solaris was available for Sparc and Intel
- Hardware is less expensive than other non-intel Unix solutions
- Calgary has significant Sun presence

© 2002, Richard S. Hammonds

13

## Why Solaris / Unix?

- Secure\*
  - weathered over 10,000 Code Red hits/day during July and August 2001
- Chosen software has supported Solaris versions
- Java / Solaris developed at Sun
- Other "Unix" solutions "Not ready for prime time"

© 2002, Richard S. Hammonds

14

## Why Multiple Servers / Firewalls?

- Balance server load
  - experience shows Web load is significant
  - can move database to own server if necessary
  - 100BT backbone
- Firewalls are now necessary
  - security (block non-used ports)
  - can redirect IP/port to internal servers
  - Black Box / OpenBSD - BSDWall

15

## Why ADSL?

- *Cost*
  - less than half cost of other vendors
- Current solution allows:
  - up to 7 static IP addresses
  - up to 5 email accounts
  - they \*KNOW\* I'm running "industrial" servers
- Excellent "Uptime" and support

© 2002, Richard S. Hammonds

16

## Why Java?

- Java is a mature OO development technology
- Java is portable if properly designed
  - code on W2K, classes run on Solaris
- Java has rich API and extensions
  - JDBC
  - Mail
  - many others

© 2002, Richard S. Hammonds

17

## Why Servlets?

- Applets suffer from security and support problems
- JSP was not mature when starting
- Servlets are server side code
  - avoid issues with portability and security
  - avoid vendor 'lock in' and shortcomings
- Servlets serve "pure" HTML 4.0

© 2002, Richard S. Hammonds

18

## Why Apache / Tomcat?

- Apache is the leading Web server solution
  - secure
  - robust
  - NOT virus/worm friendly
- Tomcat is Apache's servlet engine.
  - Supports Servlet standards
- Opensource solution

© 2002, Richard S. Hamrods

19

## Why MySQL?

- MySQL is an "up and comer"
  - smaller than commercial products
  - faster than many commercial products
  - very easy install on all platforms
  - supports ANSI SQL-2
  - nearly as feature-rich as commercial products (and changing monthly)
- Opensource solution

© 2002, Richard S. Hamrods

20

## Development Environment

- W2K Laptop
- Commercial editor
- Ant build environment
- Junit unit testing (integrated with Ant/Xerces)
- Other tools as time permits (HTTPUnit, Jmeter)

© 2002, Richard S. Hamrods

21

## Development Methodology

- Agile Methodologies (Crystal Light)
- Object Oriented
- Frequent Releases
- Test Early, Test Often
- Refactor as Required
- Transparent Development Environment

## Why OPEN SOURCE? - 1

- Free!
- Not Proprietary
  - not locked in to one vendor
  - upgrade path always present
- Can talk to the developers
  - can influence development
  - can become developer if desired

## Why OPEN SOURCE? - 2

- Can obtain source code if necessary
  - can modify product\*
- Fixes available at “internet speed”
- huge user community
  - email lists, forums, etc.
  - rapid solutions to problems
  - used by major colleges

## Live Product Demonstration

## Conclusions - 1

- Robust, scalable Client/Server system using:
  - Solaris with Intel & Sparc servers
  - ADSL with Firewalls
  - Java and Servlets with JDBC
  - MySQL database
  - Tomcat servlet engine
- Open Source is a viable solution

## Conclusions - 2

- Initial development was on time and on budget
- Open Source still performing 12 months later (2000 users, 18 MB database)
- Development is still ongoing - no intention of leaving Open Source

## Lessons Learned

- User community is most valuable resource
- Why “Everyone else is doing it”
  - means as much as it did when you were 7 (or when your kids were 7)
- upgrades aren’t always “up”
- version numbers don’t tell the whole story

## Web Sites of Interest

- [www.huntrods.com](http://www.huntrods.com)
- [www.quicktestinc.com](http://www.quicktestinc.com)
- [www.openbsd.org](http://www.openbsd.org)
- [www.bsdwall.org](http://www.bsdwall.org)
- [www.apache.org](http://www.apache.org)
- [www.mysql.org](http://www.mysql.org)

## Questions?