

Java Development

– Why is it so difficult? –



Don Friedlander, CCP, CQA
Systems Development Manager

FTA Technology
Conference 2004

10/25/04

MINNESOTA • REVENUE

Agenda:

- Introductions
- DOR's Architectural Needs
- Is Java Development so Difficult?
- What's Hard or Easy and what are some Mitigating Strategies
- Short and Long Term Support Issues
- Questions?

10/25/04

MINNESOTA • REVENUE

Minnesota Department of Revenue

- **Our mission is to make the revenue system work well for Minnesota.**

- **We:**
 - Administer Minnesota's 28 Individual and Business Taxes.
 - Collect and Distribute almost \$14 billion in annual revenue
 - Service approximately 4 million Taxpayers.
 - Collaborate with legislature, citizens, taxpayers, and other state, local and federal agencies.

10/25/04

MINNESOTA • REVENUE

Examining Needs

- **We Needed:**
 - *Improved functionality*
 - *Faster Turnaround*
 - *Services exposed to our Taxpayers*
 - *Distribution to other locations with simpler distribution methods*
 - *Scalability*
 - *Lights out processing (24x7_[mostly])*
 - *Better applications monitoring and separation*
 - *Supportable and common future architecture*

10/25/04

MINNESOTA • REVENUE

Legacy Architectures

- **Integrated Mainframe Applications:**
 - *Supra/Mantis/COBOL/MVS/CICS/Tabular Datastream bridges/FTP*
 - *Individual Taxpayer Registration, Income Tax Processing, Accounting, Refunds, and other tax types.*
- **Client/Server Applications:**
 - *PowerBuilder Clients/Sybase Databases/AIX Servers/C and C++/Tabular Datastream bridges/FTP*
 - *Business Taxpayer Registration, Sales and Use Tax Return Processing, Collections, and others tax types.*
- **FoxPro Applications:**
 - *Flat file loads/bridges*
 - *Various small tax types and federal filing history*
- **Lotus Notes Applications**
- **Windows Clients and Netware Servers**
- **Various Security Applications and Related Data**

10/25/04

MINNESOTA • REVENUE

Assessment

- **Mainframe applications were robust, but not scalable, and had no GUI**
 - *Screen Scraping was not an option*
 - *COBOL Did Not Have Necessary Features*
 - *Skills base for legacy software/hardware were aging and hard to find*
- **C/C++ Too Cumbersome**
- **FoxPro, VB, or Access were either not Robust or couldn't be easily distributed**
- **PowerBuilder & Sybase had aged (badly)**
- **Best Practices pointed to:**
 - *Browser Based GUI*
 - *Server Side Java*

10/25/04

MINNESOTA • REVENUE

New Architectural Direction

- IBM
 - *DB2 Databases*
 - *MQSeries Asynchronous Queues*
 - *Content Manager for Image storage/retrieval*
- BEA:
 - *Web Logic Server*
- Sun
 - *Java*
 - *iPlanet Web Servers*
- Internal Security
 - *DOR's Global Control*
- Query and Reporting Tools
 - *Brio*
- Scanning and Imaging Software and Hardware
 - *Kodak, IBM, Syscom, Captiva, FormWare*
- P.S. We Still Have Everything We Discussed Earlier–
(Just a few thousand lines less COBOL and Mantis Code)

10/25/04

MINNESOTA • REVENUE

What We Did

- Hired Specific Java Expertise
- Placed an Internal Lead Technologist with an External Expert
- Used a modified JAD methodology
- Built Numerous Design Models
- Used *Some* Tools
(But not enough of them)
- Acted as Developer and Integrator

10/25/04

MINNESOTA • REVENUE

Is Java Development so Difficult?

It is, and it isn't

- Does the Java community try to make it as difficult as possible?
 - *The Java 1.4 Tutorial is 1542 pages long!*
 - *The IBM COBOL Manual is 640 pages.*

- It Just Doesn't matter... There are lots and lots of "things" you need know...
 - *Components*
 - *OOConcepts*
 - *Objects*
 - *Messages and Parameters*
 - *Interfaces*
 - *Classes*
 - *Abstraction*
 - *Encapsulation*
 - *Inheritance*
 - *Polymorphism*
 - *API's*
 - *Beans*
 - *Containers*
 - *IDE's*
 - *Java Foundation Classes*
 - *Deployment (builds)*
 - *Factories*
 - *Java Server Pages*
 - *JARs*
 - *JNDI*
 - *Libraries*
 - *Memory Leaks*
 - *Models*
 - *Multi-Tasking and Multi-Threading*
 - *Packages*
 - *Patterns*
 - *Refactoring*
 - *Servers*
 - *Servlets*

10/25/04

MINNESOTA • REVENUE

What's Hard? What's Easy?

10/25/04

MINNESOTA • REVENUE

What are some Mitigating Strategies?

- Not Just A Language
- Evolution
- Lot's Of Features
- Roots in C/C++
- Steep Learning Curve
- Object Orientation
- Open Source/Freeware
- Deployments
- Integrating
- Writing Code
- IDE's
- Database Connectivity
- Portability
- Memory Management
- Exception Handling
- Debugging

10/25/04

MINNESOTA • REVENUE

Short and Long Term Support Issues

- **Short Term**
 - Transitioning Staff from Procedural to Object Models
 - Education in J2EE Classes and Libraries
 - Understanding The Technologies
- **Long Term**
 - Standards and Code Walkthroughs
 - Survivability
 - Evolution
 - Dynamic Environment
 - Ownership and Maintenance
 - Significant Increase in Support Resources
 - Maintaining Skills

10/25/04

MINNESOTA • REVENUE

Questions?