

Birds Do It: Migrating Forms to Java EE Web—A Case Study



Government
Technology Conference
(GTC) East 2005

Peter Koletzke
Technical Director &
Principal Instructor

QUOVERA



Agenda

- The migration requirement
- Technology decision process
- Challenges, successes, lessons learned



Government
Technology Conference
(GTC) East 2005



QUOVERA

2

The Legacy Application

- The organization
 - New York State agency: Office of Alcoholism and Substance Abuse Services (OASAS)
- The application
 - Federally required reporting system for all patient admissions and discharges into substance abuse treatment programs
 - Statewide application involving 1,200 government, private and nonprofit programs
 - Approximately 400,000 transactions per year
 - Character-mode Forms 6.0 run in a telnet client session; Oracle 8i database



QUOVERA

3

Legacy Application's Shortcomings

applprod - TNVTPlus

10/29/2003 OASAS - CLIENT
10:46 CLIENT A
Provider: 87171 PRU: 00032 Client ID:

Sex:	1	Employment St
Birth Date:	01011978	Primary Sourd
Last 4 SSN:	9999	Marital Statu
L Name 2 Char:	00	Child Alchoh
Adm Date:	10292003	Number of Ch
Transaction Type:	1	Living with
Number Assessment Days:	00	Criminal Jus
Admission Disposition:	01	For 6 Month
Signif-Other Admission:	2	Number Time
Race:	4	Days Incar
Hispanic Origin:	5	--Problem S
Veteran Status:	2	Type
County of Residence:	99	Primary: 01
ZipCode of Resid:	99999	Secondary:00
Type of Residence:	01	Tertiary: 00
Living Arrangements:	1	Prior Treatm
Principal Refer Source:	01	Last Alc/SA
Highest Grade Compl:	01	Time Since L

Count: *0 <Replace>

- Non-GUI interface with resulting poor navigation
- Extensive on-site training
- Required client side software and hardware
- Difficult to modify
- Limited features
- Business rules in the application

QUOVERA

4

Migration Objectives

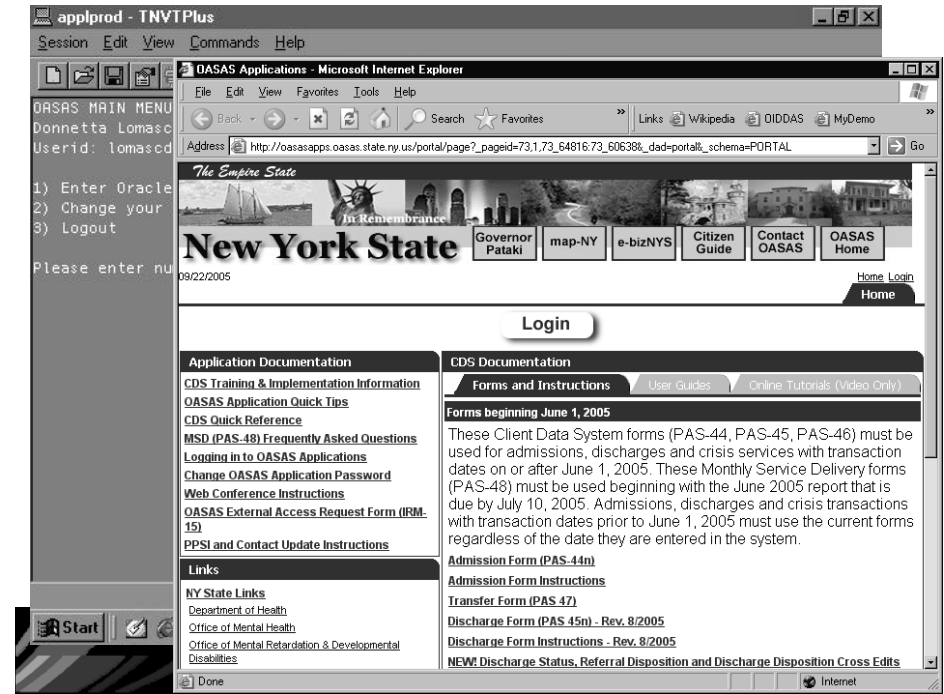
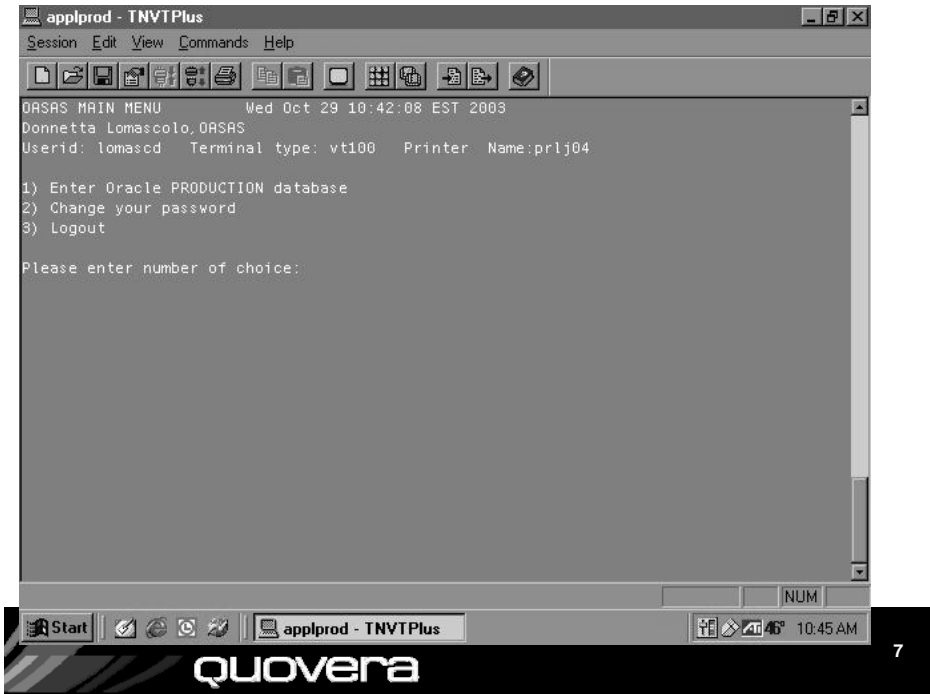
- Use industry standard technologies
- Leverage existing staff's knowledge base
- Mix mentoring and training into development work
- Retain the Oracle RDBMS
- Modernize the user interface
 - Pulldown descriptions instead of code values; easier navigation
 - Eliminate need for telnet servers and client software
 - Improve online help and provide online training

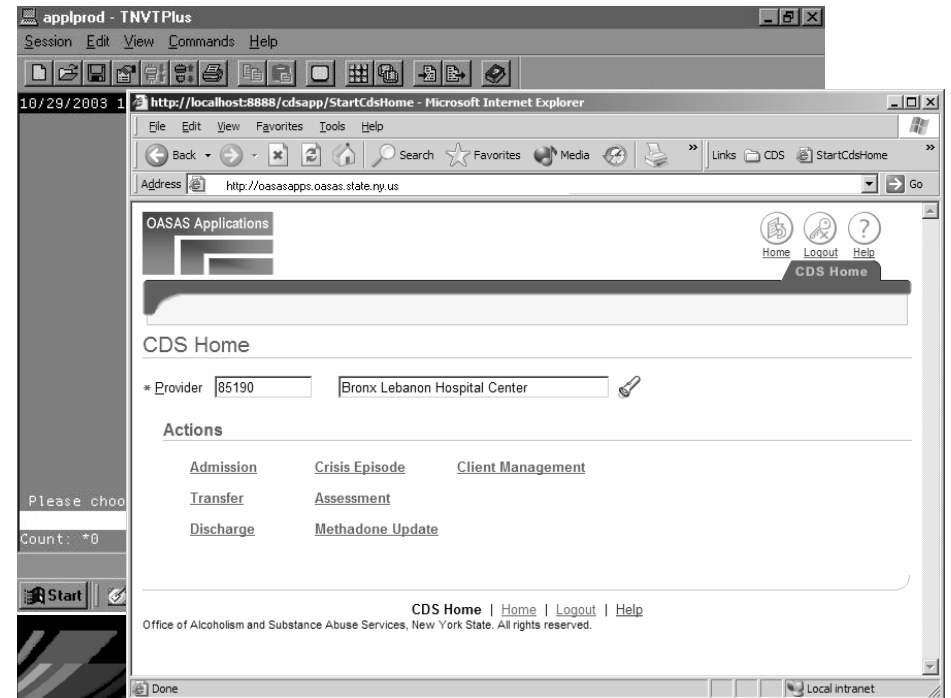
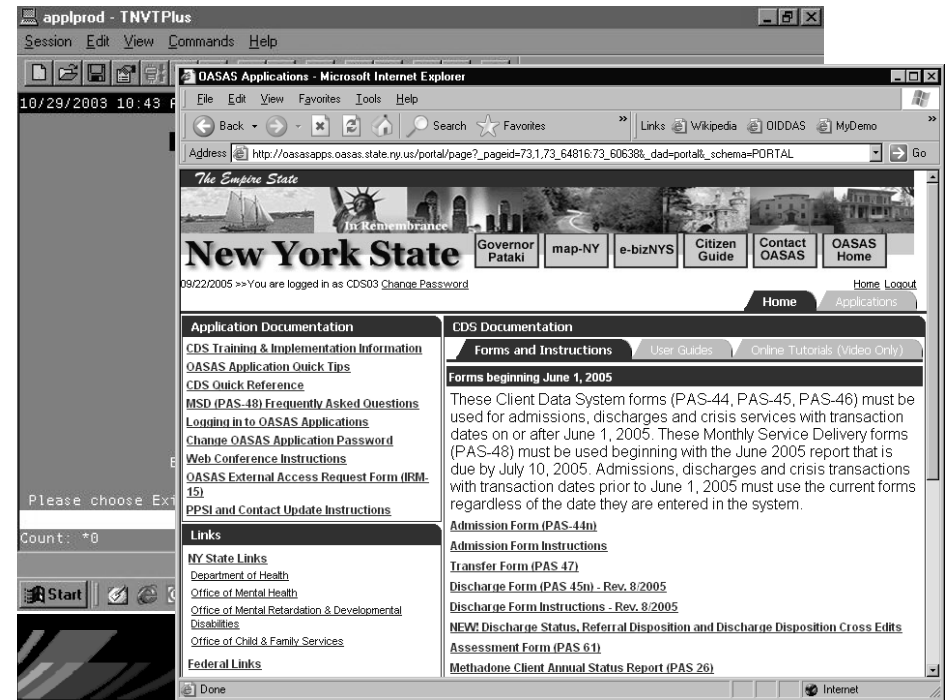


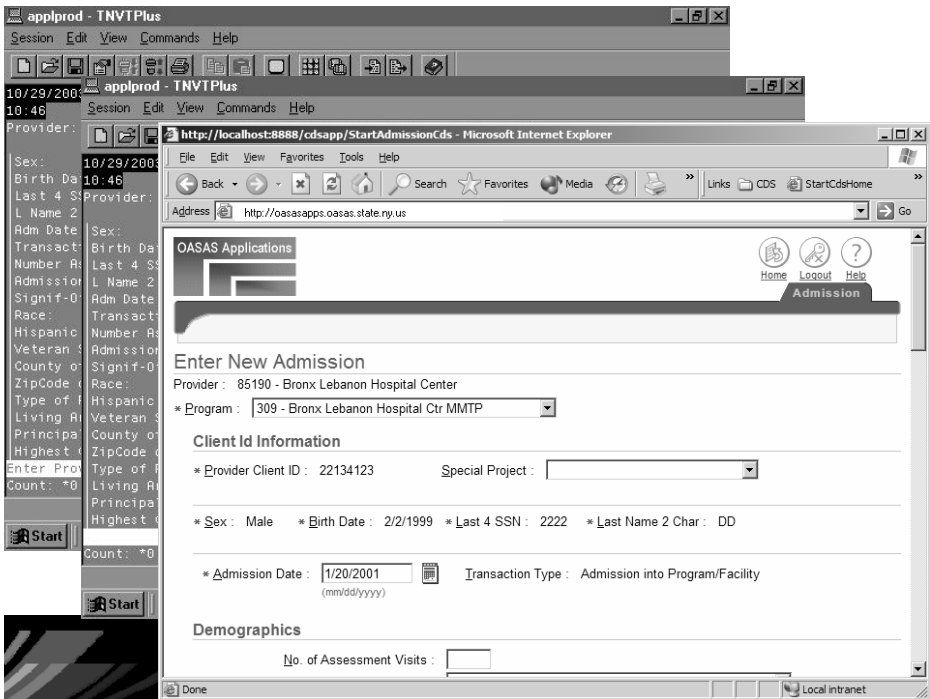
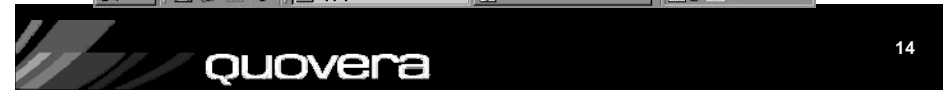
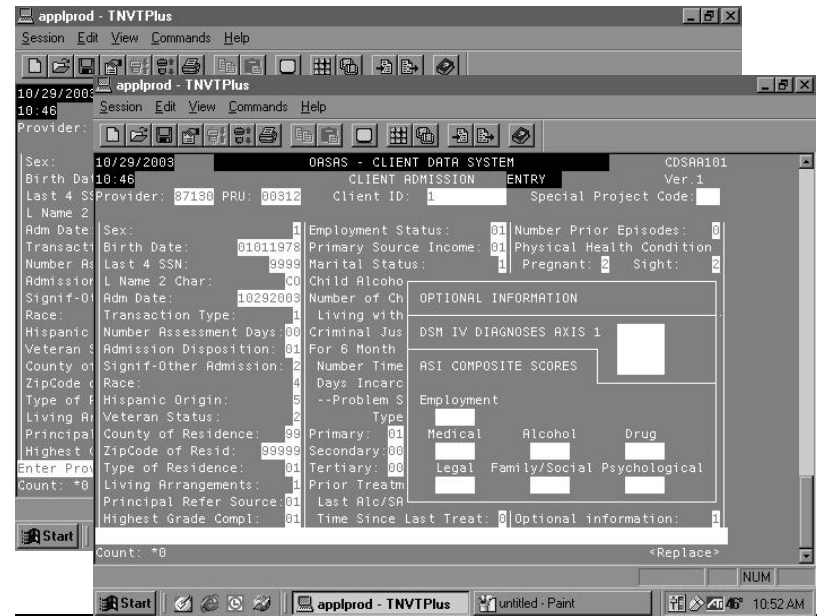
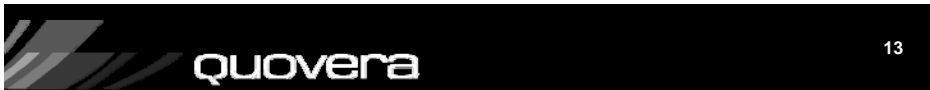
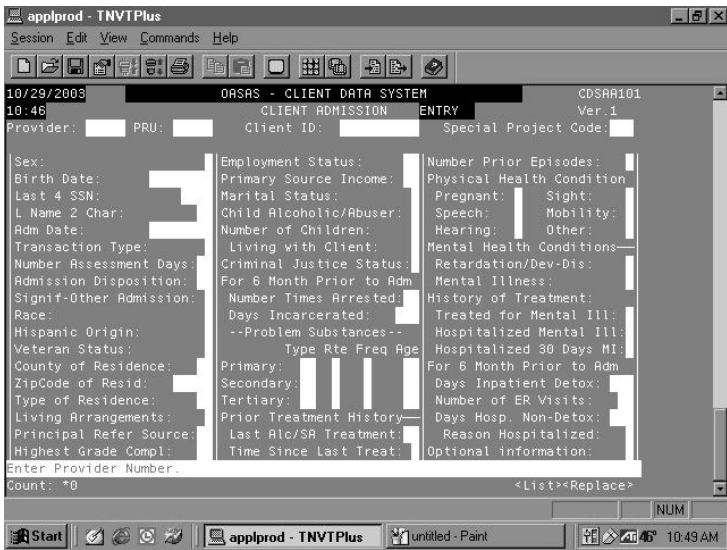
User Interface Changes

Page Style	Legacy	Web
Start up page	Telnet session menu	Home page with feeder forms, publicly available
Menu of all applications	Character mode list	Tab selections on home page after login
Menu of application functions	Multi-page character mode list	All links on one page
Application form	Character mode entry of codes (lists-of-values available to decode codes)	GUI pulldown lists, textual entry (no knowledge of codes required)

- Same as any char mode to GUI migration







Agenda

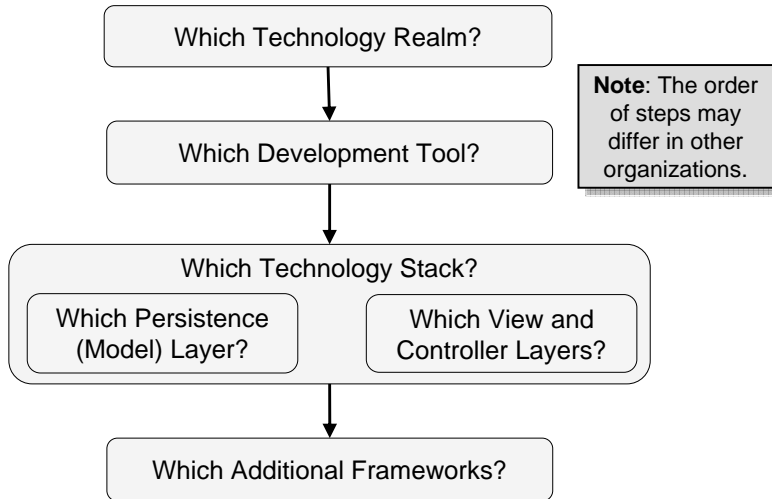
- The migration requirement
- Technology decision process
- Challenges, successes, lessons learned



Government
Technology Conference
(GTC) East 2005



Decision Process



Notes About the Decision Process in this Case Study

- Migration consisted of multiple subsystems
- First subsystem migration project started in 2003
 - Decision tree was followed with available technologies at the time
 - It was followed again for each subsystem within this application
 - The decisions had similar results but used technologies that were current at the time
- To simplify, this discussion merges these into a single process



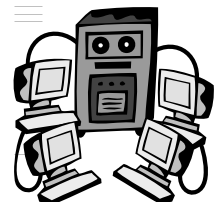
Which Technology Realm?

- Decision: Java Enterprise Edition (Java EE); Web
- Reasons for Java EE
 - Industry standard; lots of options exist for Java
 - Oracle is focusing on it
- Reasons for web interface
 - Available anywhere, any time
 - Browser-based user interface
 - Eliminates telnet software (cost and set up)
 - No applet downloads; JInitiator was rejected
 - Familiar GUI interface components
 - Eliminates arcane function key presses



Which Development Tool?

- Decision: Oracle JDeveloper (9i, 10.1.2, 10.1.3)
- Reasons
 - Works well with Oracle database
 - Flexible: can produce standard code using any Java framework – from Oracle or from elsewhere
 - Oracle uses it to build the EBS apps
 - “Standard” Oracle development tool
 - Single vendor
 - Parallel decision was made to also use Oracle Application Server
 - “Relatively familiar” environment for Oracle Forms developers



Which Technology Stack?

- Decision: Oracle ADF frameworks
- Reasons
 - They are well supported in JDeveloper
 - The development experience consists more of declarative programming than of writing lines of code
 - More familiar to PL/SQL and Forms developers
 - Oracle supports them
 - Vendor support vs. community support



Which Persistence (Model) Layer?

- Decision: ADF Business Components
- Reasons
 - ADF BC works better than EJB for an Oracle database
 - Friendlier than EJBs and TopLink for traditional Oracle developers
 - Project started before EJB 3.0
 - Reliance on Oracle-specific framework not considered to be a drawback
 - Again, the benefit of vendor support



Additional Benefits of ADF BC

- Simple to develop – less Java code
 - Assumes business rules code is primarily in the database
- Used by Oracle to build its prepackaged applications
 - A.k.a., Oracle Applications, E-Business Suite, Fusion Applications
 - We can rely on support, assistance, bug fixes, enhancements
- Does everything that other persistent frameworks do
 - Hibernate
 - EJB
 - TopLink



Even More Benefits of ADF BC

- Highly-evolved
 - Started when most Java developers were still coding JDBC
- In its earliest iteration, more features than most comparable frameworks
- Middle tier data caching done natively before other frameworks tried this
- Plays well with others in ADF
 - Narrows the decision for View and Controller layers



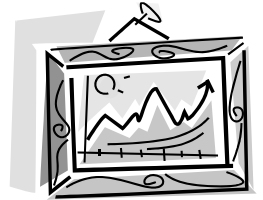
Which View/Controller Layers?

- Decision:
 - 9.0.3: UIX/MVC
 - 10.1.2: JSP/Struts
 - 10.1.3: ADF Faces/JSF
- Reasons
 - Similar: Oracle is/was using them to create its applications (except 10.1.2)
 - Excellent support in JDeveloper
 - Good IDE tools for Oracle Forms developers
 - They are all Oracle supported



Which Additional Frameworks?

- Decision: Oracle JHeadstart
 - Oracle Consulting JDeveloper plugin
 - Generates View and Controller code from definitions in an XML file
 - Generated code was customized in some subsystems
 - Later subsystems are 100% generated
- Decision: Drools
 - Rule engine used for page flow logic
 - File-based repository with logic
 - Complex navigation between pages
 - Multiple versions of each page (date-driven)



Agenda

- The migration requirement
- Technology decision process
- Challenges, successes, lessons learned



Government
Technology Conference
(GTC) East 2005



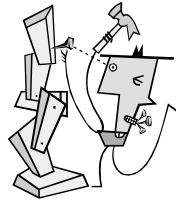
Did We Mention the Award?



Government Technology Conference
GTC East 2005

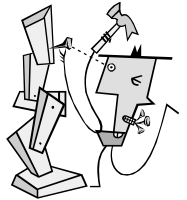
Challenges – Strategy

- Staff unaware of Java EE coding styles
 - Training on new tools and languages early in project
 - On site “mentors” and Java architect to answer questions (and do the “hard stuff”)
 - “Recipes” for specific tasks
- Users accustomed to character mode forms
 - Online training
 - Much advance warning



Challenges – Technology

- Redesigning navigation and security
 - Character mode menus needed modernization
 - Eliminating individual database user accounts
- Rewriting all application code
 - Moving Forms business rules logic to the database
 - Questioning and tightening up business rules
 - Rewriting character mode reports in Oracle Reports
- Installing new hardware for Oracle Application Server and Portal
- Database upgrade to 9i (now 10g)
- Integrating web application security



Successes – Strategy

- Gradual rollout to user base was a key
 - Over 400 user sites (multiple users each)
- User acceptance remarkably high
 - Most folks seem to know about Internet applications
 - No need to teach how a mouse works
- Moving business rules code to the database was a key to success
 - Business rules are independent of application
- You need a resident Java architect
 - Preferably one who knows ADF



Successes – Staff Transition

- Somewhat apprehensive staff was productive
 - New is always scary but in a fun way
- Mentoring strategy empowered them for future work
 - Pre-project training addressed project requirements
- Database code orientation side effect:
 - Leverage known skills
- Mentoring included “recipes” for frequently-performed tasks
 - Create a form
 - Create a report



Successes – Handling Users

- Online problem and issue tracking
 - Speeds problem resolution
 - Users enter issues online (no Help Desk needed)
 - Development staff processes the problem
 - Issues stored in the database
- Online user administration
 - Manages LDAP user repository



Successes – Documentation

- Online training videos
 - Streaming video
 - Virtually eliminates onsite classroom training
- Online documentation
 - Step-by-step user guides
 - Online paper forms
 - Lists of business rules



Successes – Technology

- Technology choices proved solid
- ADF Business Components presented no major surprises
 - JDBC layer worked
 - Published techniques for application context and user connections worked
- View and Controller technologies did their job
 - Used Browser Look and Feel
 - Slightly boring but consistent with EBS
- JHeadstart plugin helped
 - JDev 9i offered less declarative coding
 - Softened Java development learning curve
 - Provided default page flow code and patterns



Bonus Success

- Oracle is now “betting the farm” on these technologies to build Fusion Applications
 - JDeveloper is the development platform
 - ADF BC is the Model layer
 - ADF Faces/JSF is the View layer
 - JSF (plus ADF Task Flow) is the Controller layer
- Who woulda known?



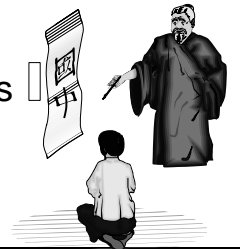
Lessons Learned – Strategy

- Full application rewrites cannot “leave everything the same”
 - Budget for some redesign time
- Database structures can and should change
- Be watchful of scope creep
- Plan for staff skills transition time
- Budget for a business expert on the team
 - You need quick decisions



Lessons Learned – Technology

- Don't use Oracle SCM with the JDev 9i front end
 - Enough said
- Design around native features
 - Frameworks work best when you leverage their strengths
 - Avoid overriding framework classes
- Stick with default page flow
 - Upgrades will be easier

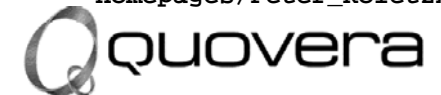


Summary

- You can do well with ADF BC and related frameworks
- Remember the successes and lessons learned
- This is a specific case study
 - Your mileage may vary
 - Decision points will be slightly different in each case
- You can do it!
 - Only if you need to, however



- Please fill out the evals
- Books co-authored with Dr. Paul Dorsey, Avrom Roy-Faderman, & Duncan Mills
- Personal web site: http://ourworld.compuserve.com/homepages/Peter_Koletzke



<http://www.quovera.com>

- Founded in 1995 as Millennia Vision Corp.
- Profitable for 7+ years without outside funding
- Consultants each have 10+ years industry experience
- Strong High-Tech industry background
- 200+ clients/300+ projects
- JDeveloper Partner
- More technical white papers and presentations on the web site