Migrating to Red Hat JBoss Middleware

Marc ZOTTNER
Senior Domain Architect and
Red Hat JBoss Middleware
mzottner@redhat.com

Tobias HARTWIG
Regional Product Manager, EMEA
Application Platforms Business, Red Hat
thatwig@redhat.com

June 26th, 2015
Definition: “Middleware Migration”

Major technical upgrade of an application landscape, its runtime environments and life cycles without functional change.

Different vendor and / or major version

AS - IS  Migration  TO - BE

#redhat  #rhsummit
Agenda

• BENEFITS – Why migrating to JBoss Middleware?

• APPROACH – How to do it best?

• EXAMPLES – Customer success stories!
BENEFITS

Why migrating?
WHY CHANGE?
Business Expectations become IT Challenges

- Deliver **VALUE** to the Business
- Go to market **FAST**
- Build and manage with **LIMITED RESOURCES**
WHY CHANGE?
Business Expectations become IT Challenges

70%

30%
WHY CHANGE?

Business Expectations become IT Challenges

- Complexity
  - More
    - Faster
      - Change Barrier
  - For Less

Rising Cost of Status Quo
There is hope: migrate to Red Hat JBoss Middleware

RE-BALANCE MAINTENANCE AND INNOVATION

REMOVE TECHNICAL DEBT & RISK

BECOME MORE PRODUCTIVE
Balance Maintenance cost and Innovation

JBOSS MIGRATION BENEFITS

- Lower TCO
  - Customer-friendly Subscription model, simple and flexible
  - Standardize, simplify, automate
  - Lower operational cost, higher productivity.
- Cloud ready – at your pace
- Enable business innovation beyond the Application Server with the Red Hat JBoss Middleware Portfolio

Shift budgets from “keeping lights on” to innovation
Balance Maintenance cost and innovation

JBOS MIGRATION BENEFITS - COST

Use the JBoss EAP platform cost comparison calculator:

- Available at http://www.redhat.com/promo/eap_calculator/
- 3 year cost comparison – license and support
- Fully customizable for your situation
Balance Maintenance cost and innovation
JBoss Migration Benefits - ROI

IDC Study - Financial benefits of moving from a proprietary platform to JBoss EAP

- IDC interviewed 6 large, US-based companies using JBoss EAP to develop and run custom applications for at least 12 months.
- Financial benefits experienced moving from IBM WebSphere and Oracle WebLogic to Red Hat JBoss EAP included:
  - 39% shorter development time for new applications (avg. 16 weeks!)
  - 24% less developer hours per application
  - 65% less infrastructure costs supporting development
  - 89% less cost to manage
  - 51% more applications developed in 1 year
  - 569% ROI over 3 years, payback of initial investment in less than 6 months after deploying the platform

- Study available at https://engage.redhat.com/content/jboss-eap-businessvalue-s-201401170935
Technical Debt and Risk

JBOSS MIGRATION BENEFITS

• Eliminate lock-in: embrace Open Source and open standards
• Reduce technical and business risks
  – Technology and framework updates, standard components
  – Security, audit-ability, maintainability, modularity, scalability
• Spend less time dealing with technical debt, focus energies on new initiatives

Standardize and modernize applications for lower maintenance, integration cost and lower risk.
Productivity to develop, deploy and operate

JBOSS MIGRATION BENEFITS

• Simplify and streamline application environments
  – Reduce operational complexity
  – High degree of automation
  – Infrastructure to support better process, not get in the way

• Enable business agility, align IT and business
  – Boost developer productivity
  – Cloud / DevOps / PaaS readiness

New apps and features faster, focus creative energies on innovations that make a difference.
Calculate the business benefits and impacts of PaaS:

- Accelerated application development
- Automated application provisioning and config
- Web-scale application operations
- Increased hardware utilization efficiency
Productivity to develop, deploy and operate JBOSS EAP ON OPENSSHIFT BENEFITS

Calculate the business benefits and impacts of PaaS:

- Accelerated application development
- Automated application provisioning and config
- Web-scale application operations
- Increased hardware utilization efficiency

Customize to your own situation and download report:
Migrations to JBoss Middleware
Red Hat's Catalyzers

Red Hat JBoss Migration Toolkit
Migrations to JBoss Middleware
Red Hat's Catalyzers

Red Hat JBoss Migration Toolkit

Collaboration, Sharing, Knowledge
Migrations to JBoss Middleware
Red Hat's Catalyzers

Red Hat JBoss Migration Toolkit

Collaboration, Sharing, Knowledge

Methodology and Approach
PROVEN APPROACH

How to?
Expect answers today!

- How many man-days do I need to migrate all my applications?
- What is the best approach to identify and mitigate migration risks?
- What are the biggest time-savers and catalysts for a migration?
Common Migration Challenges

**MoNSTeR application**
- Huge and monolithic
- Historically grown
- Tightly coupled
- Poorly documented
- Customized standards
- Horrifying tech. debt
- Highly tuned
- Company critical
- “Do not touch” sticker

**stAff**
- Dr. Frankenstein (SME) no longer in-house
- Missing knowledge of the target platform
- Technical and business releases combined
- Weak automation across the app. life-cycle
- No automated tests, low coverage

Image source: https://www.flickr.com/photos/cleomorgause/3887178183/
Enterprise Class Challenges

- Very heterogenous implementations across enterprise
  - Java 3, 4, 5, 6, 7
  - No standard: any version of any imaginable framework
  - Customized version of the above

- Too many tightly coupled applications
  - Hard to define a migration strategy (big bang vs. pragmatic)
  - Effort to estimate itself takes too much time
  - Unknowns creates fear to get started

- Impact on the whole application life-cycle
  Development → Build → Test → Operations

Image source: http://www.moebiusnoodles.com/
Scope of a migration

Application code

Infrastructure
hardware + virtualization + OS + JVM + application container

Processes
application life-cycle, build, configuration, deployment, provisioning, DevOps, environments, test, integration, continuity *, monitoring ...

Knowledge
Core recommendation

Involve Red Hat JBoss expertise before it gets purely technical ... 

... to make your migrations low-risk, predictable and efficient.

Migration Good Practices – Top 3

**Information sharing** based on a central collaborative platform

*Least effort*: no issue solved twice, no question asked twice

**Reuse, automate, standardize** as much as possible

- infrastructure, environment, dependencies, processes, operations
- strategy for software versioning and revision control
- functional and non-functional tests

**Minimalist and pragmatic methodology**

*As few changes as possible* to get a running *functionally identical* application
Migration Good Practices – Top 3

1. Information sharing based on a central collaborative platform
   - Least effort: no issue solved twice, no question asked twice

2. Reuse, automate, standardize as much as possible
   - infrastructure, environment, dependencies, processes, operations
   - strategy for software versioning and revision control
   - functional and non-functional tests

3. Minimalist and pragmatic methodology
   - As few changes as possible to get a running functionally identical application
Establish one living collaborative documentation platform:

- central entry point for the migration
- significant catalyst and time-saver
- exhaustive, concise, comprehensive and accurate documentation
- easy to browse and to search (tagged content, lean structure)
Collaboration Content – Example

- **Step-by-Step Migration Guide**
  Comprehensive and pragmatic approach to migrate an application from scratch

- **Migration Cookbook**
  Thematic collection of “How-to” and “Known-solution” recipes
  Recipe = article (“issue”, “resolution”, “learn more”)

- **Platform FAQ**
  Learning fast and more about the new platform

- **Pilot changes**
  Description of all changes done to specific projects
Migration Good Practices – Top 3

Information sharing based on a central collaborative platform

Least effort: no issue solved twice, no question asked twice

Reuse, automate, standardize as much as possible

• infrastructure, environment, dependencies, processes, operations
• strategy for software versioning and revision control
• functional and non-functional tests

Minimalist and pragmatic methodology

As few changes as possible to get a running functionally identical application
Automation: Wind-up your applications

Windup = open source reverse-engineering tool generating top-down HTML report foreseeing migration changes

• Shows a holistic picture of the Level of Effort in story points needed to migrate or upgrade applications to Red Hat JBoss EAP
  → Application complexity not taken into account.

• Decompiles and analyzes specific packages and files
  → No source code required. Only the JAR / WAR / EAR files.

• Rule-based estimation, listing potential changes for the migration
Use-cases

- **Migration assessment**
  - Identify critical issues
  - Do estimates
  - Build application clusters
  - Select pilots

- **Exhaustive migration support**

- **Extend with new/custom rules**
  - Linking collaboration platform
  - Specific pattern detection

Some generated HTML reports ...
Migration Good Practices – Top 3

Information sharing based on a central collaborative platform

**Least effort:** no issue solved twice, no question asked twice

Reuse, automate, standardize as much as possible

- infrastructure, environment, dependencies, processes, operations
- strategy for software versioning and revision control
- functional and non-functional tests

Minimalist and pragmatic methodology

**As few changes as possible** to get a running **functionally identical** application
Migration Methodology

Phase 1 – Preparation

Assess → Proof of Concept → Pilot → Plan → Execution
Migration Assessment

AS - IS  Migration  TO - BE

Assess  Proof  Pilot  Plan  Exec
Migration Assessment

Analysis

• infrastructure, architecture, technologies and application landscape
• dependencies, interfaces, non-functional constraints
• knowledge, processes and life-cycles

Definition

Report and management presentation

• Feasibility and early identification of potential risks (PoC)
• Migration plan draft and gross estimates
Assessment Time Plan – Example

Estimated assessment duration: 2 weeks

- Preparation
  - Initiation
    - Architecture & Dependencies
  - Processes
  - Application analysis
  - Documentation

Next steps
- Go/No-Go Decision
- Results Presentation to the Management

Assessment phases:
- Assess
- Proof
- Pilot
- Plan
- Exec
Proofs of Concepts

• How
  • Isolate and solve previously identified technical risk
  • Focus on feasibility and documentation – no (pilot) application migrated

• Why
  • Risk mitigation – Find and crack the hardest nuts first!
  • Prepare and seed the target infrastructure

• Examples
  • Replacement strategy for proprietary code and libs (OS native, WebSphere, Weblogic…)
  • Integration with 3rd party components (WebSphere TAM, TAI, LogFaces…)
  • Infrastructure, management, provisioning, monitoring and security implementations
Pilot

• How
  • Finalize the infrastructure (CI/CD/PaaS)
  • Define application clusters / types
  • Select representative applications
  • Technically migrate and document
  • Move applications through their life-cycle

• Why
  • Sharpen estimates and minimize risks
  • Prepare large-scale migrations (infrastructure, processes, documentation)
Plan and Prepare

- Define the migration execution strategy
  Phases / Order / Teams / Documentation / Code / Infrastructure / Processes

- Make estimates based on ...
  Assessment / Windup reports / Proof of Concept / Pilot

- Refine the project plan

- Finalize all preparations
  Structured documentation / Guidance / Knowledge transfer
Execution

- Developer and administrator enablement courses
  1. Understanding the new Application Platform
  2. Using the new infrastructure (PaaS)
  3. Migrating your applications

- Step-by-step application migration

- Dedicated technical support

- Continuous improvement loop: infrastructure, standards, documentation (DevOps)

Image source: https://www.flickr.com/photos/sidereal/76724710/
Step-by-Step Migration

1. Initial server configuration
2. Pre-emptive changes
   ... use Windup report, migration guide, Java standards ...
3. Iterative deployments and modifications until the application runs
   ... do as few changes as possible, document all changes
4. Optional changes
   a. Standardization / Cloudification
   b. Framework updates / Refactoring / Optimizations
5. Staging, quality and functional as non-functional tests
6. Go-live
EXAMPLES

Customer successes...
Some JBoss Migration Customers

- **NYSE EURONEXT**: 50-60% cost savings; increased performance, deployment speed, streamlined processes; faster development cycles, improved security.
- **Lufthansa Cargo**: Saved 50%+ by migrating. Gained flexibility, accommodate new customer requests faster. Free from HW and SW lock-in.
- **UnionBank**: Replaced aging and costly IT infrastructure. Scale to growth and respond agilely to changing market dynamics. Improved reliability and scalability, cut costs, new financial services and products to market faster.
- **Sprint**: Saves $4 million annually in licensing and maintenance fees, improved developer productivity, reduced time to market.
- **CenturyLink**: More flexibility at lower cost, better separation of Dev and Ops considerations, save time deploying, provisioning, managing. WebLogic to JBoss EAP.
- **Roche**: Reduced cost, developer workload and time-to-market, faster provisioning, optimized operations, high automation. Approx. 600 Java applications. EAP, Data Services, OpenShift (POC).

http://customers.redhat.com/
Customer Story: Dienst Uitvoering Onderwijs (DUO)

- Executive agency of the Dutch ministry of education
- Finances and informs all participants in education
- Manages student grants and loans
- Organizes examinations
- 2700 employees, over 500 IT specialists
Migration motivation

- IBM WebSphere costs too high

- Update to Java EE 6

- Unsatisfied business needs: quality and time-to-market
Selection Criteria

• Functional
  Java EE 6 certified full profile with management tooling

• Support
  Enterprise-class support with a long life-cycle

• Experience
  Proven platform with many relevant public references

• Cost
  Lower cost than WebSphere, free version available
Scope

Processes

Applications

Application Platform

JDK

Migration

OS

Migration

OS
Migration Roadmap

- **Phase 1 (2014) – Preparation Phase**
  - Migration assessment
  - Proof of Concepts
  - Pilot projects
  - Preparation, planning and documentation

- **Phase 2 (2015/2016) – Large scale migration**
  - Enablement workshops for developers
  - Application migration and dedicated support
  - Improvement loop: infrastructure, standards, documentation

- **End of 2016 – IBM WebSphere decommissioned**
Experience so far ...

- Catalyzed migration, high-quality outcome
  - Standardized and automated everything
  - Paradigm shift to DevOps and PaaS
  - Early involved Red Hat knowledge and expertise

- Financially scalable model

- Flexible choice of products and suppliers (no lock-in)

- Business department is excited!
SUMMARY
Wrap-up

Benefits

- Early engage Red Hat JBoss expertise
- Central collaborative platform
- Reuse, automate, standardize
- Minimalist and pragmatic methodology

Challenges

Approach

Scope

Assess → Proof → Pilot → Plan → Exec