Modernizing Complex Legacy Applications

Pranjal Bathia
Mike Moore
9-May-2019
Legacy Application Modernization

**legacy**  adjective

of, relating to, or being a previous or outdated computer system  
(Merriam-Webster)

denoting or relating to software or hardware that has been superseded but is  
difficult to replace because of its wide use   (Dictionary.com)
Should you modernize?

There are reasons to change (some better than others)

- Improved functionality
- Expansion of system capabilities (scalability, performance, maintainability, flexibility, etc.)
- Opportunity and Risk Management

There are also reasons for caution

- Cost
- Business disruption
- Strategic drift and feature decay
Legacy CWE system

Certification Workflow Engine

- Use cases for solution certification
- Used by partners and Internal associates
- Decade old app, organically developed
- Undocumented. Ambiguous feature set
- Supportability Gaps
- Performance issues
- Restriction on developing more features
How to start?

1. Audit
   - Analysing the system

2. Evaluate
   - Compare technologies

3. Prepare
   - Get cloud ready

4. Plan
   - Identify milestones

5. Deliver
   - Final deployment
Audit

Analyzing the system

Understand
Understand business and the use case of application

Identify
List down discrete applications and infrastructure

Assess
Examine access logs to list down functions being used and which one can be retired

Map
Map relationship between applications and check which one can be combined

Dependency
Understand business dependency to make decision on disruption it can handle
## Evaluate

Compare against different parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Option 1</th>
<th>Option 2</th>
<th>Option 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>✓</td>
<td>☒</td>
<td>✓</td>
</tr>
<tr>
<td>Database</td>
<td>☒</td>
<td>✓</td>
<td>☒</td>
</tr>
<tr>
<td>Application Server</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Programming Language</td>
<td>✓</td>
<td>☒</td>
<td>✓</td>
</tr>
<tr>
<td>Monitoring tools</td>
<td>☒</td>
<td>✓</td>
<td>☒</td>
</tr>
<tr>
<td>Storage</td>
<td>✓</td>
<td>☒</td>
<td>✓</td>
</tr>
<tr>
<td>Cloud</td>
<td>☒</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cost</td>
<td>☒</td>
<td>✓</td>
<td>☒</td>
</tr>
</tbody>
</table>
We used open source technology
Prepare
Get cloud ready

- Rehosting
- Replatforming
- Repurchasing
- Retain
- Retire
- Rearchitecting

#redhat #rhsummit
Microservice Based architecture

Prepare
Plan

Identifying milestones allows minimal disruption to business and most effective method for migration

**Users**
Release MVP for set of users to minimize the business impact

**Business Functions**
Chose simpler functions which increases chances of success

**Sensitivity of Data**
Pick up function that deals with less sensitive data to start

**Size of Application**
Opt for a smaller application which helps to deliver & learn faster

**Complexity of Data**
Go for a service which has less complex structure of Data
Plan

Phase based approach

**PHASE 1**
OSP
- Simplest business case,
  well defined use cases and stakeholders

**PHASE 2**
Admin
- Admin functions, limited stakeholders

**PHASE 3**
Cloud Program
- Conservative stakeholders and open ended use cases

**PHASE 4**
Hardware
- Complex and business critical

#redhat #rhsummit
Plan
Phase based approach

Change Management

Process Transformation

Execution

Develop Champions
Align Teams
Communication

Front load Impacts
Skate to where the puck will be

#redhat #rhsummit
Deliver

First release should be MVP (Minimal Viable Product)

Plan  Build
Learn  Measure
Tips

1. Do a thorough **impact analysis** as part of your audit
2. **Containerization** allows you to deploy your application anywhere
3. Building **smaller services** helps efficient decoupling of system components
4. Focus on the **business processes** and **users** in your transitional states
5. A **phased migration** can help mitigate unforeseen pitfalls
6. Identify opportunities to include **automation** in every phase
THANK YOU

linkedin.com/company/Red-Hat
youtube.com/user/RedHatVideos
facebook.com/RedHatInc
twitter.com/RedHat