

Mature Legacy Seeking Sexy New Technology for Fun and Profit

Extracting Gold from Legacy Code

Dave Thomas

Bedarra Research Labs,
Queensland University of Technology Australia, Carleton
University Canada

dave@bedarra.com
www.davethomas.net

©2010 Bedarra Research Labs. All rights reserved.

Once Upon A Legacy

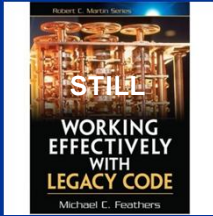
1. Dirty Jobs or Fame & Fortune?
2. No Fear - 1,2,3 Charge!
3. Agile in a Legacy Environment?
4. Time Something Completely Different?
5. Thinking Out of The Legacy Box – Case Studies of Legacy Innovation



©2010 - 2011 Bedarra Research Labs. All rights reserved.

“Legacy Code is code that has **No tests”**

- has insufficient tests
- is older than 3 years
- technical debt written by others . . .
- Is written in a language, tool, platform we don't know or like
- is difficult to understand or change
- . . .



New Improved 16th Edition!
(c) 2020

70 – 90 % of Software Budgets are spent on Legacy Systems!

©2010 - 2011 Bedarra Research Labs. All rights reserved.

Legacy – The Big Ball of Mud*

DILBERT © United Feature Syndicate, Inc. Redistribution in whole or in part prohibited.

* "The Big Ball of Mud" Design Pattern – Brian Foote

Legacy = Big Ball of MUD which has significant business value!!!

Oh My! It is our mess (debt)!?

Our framework adds so much value

Just add a little glue code and some SOA wrappers

Look we will just have to refactor it later!

Legacy ? No Fear ... 1, 2, 3 Charge!

Our vendor! Our consultants! Our outsourcer! Our team has the solution !!!

It will take years and lots of money but we can climb legacy mountain

©2010-2011 Rederra Research Labs. All rights reserved.

Rewrite It Using Modern Technology

Rewrite in my language/platform
high risk - most major rewrites fail
rewrites seldom provide sufficient business value
rewrites can be more difficult to maintain than the legacy

Automatic Migration – Rewrite Magic
Migration to equivalent runtime **perhaps**
Translation to another language is **seldom worth it**

Second Systems Effect: Read Mythical Man Month AGAIN!

© 2010 - 2011 Rederra Research Labs. All rights reserved.

Outsource It!

Out Source: Make it someone else's problem

- may work for a short time
- risks of **loss of skills**, knowledge of **code base** and domain **knowledge**
- increased investment in **requirements and testing**
- Stress fractures **where local systems touch outsourced systems** compounded by different processes and tool chains

© 2010 - 2011 Rederra Research Labs. All rights reserved.

Just SOA It!

Just Use SOA: Wrap It Into Services

- Assumes you have **interfaces**
- Assumes you have **tests**
- Assumes the service is **easily surfaced** and **encapsulated** when in practice the it may be **buried in several modules**
- Assumes that ESB/XML/Process Server **Performance, RMA** is acceptable when it often is **unknown**
- Analysis **paralysis** between **enterprise architects**, **vendor architects** and **development teams**
 - Which ESB, Process Server ...
 - SOAP, REST ...
 - XML ... JSON ...
 - BPM/BPEL ...

© 2010 - 2011 Rederra Research Labs. All rights reserved.

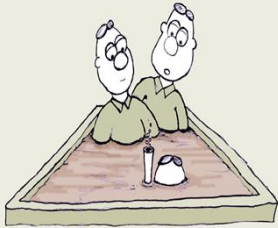
Agile It !

Incrementally Refactor It

Do you believe in Magic ?

- Wizard's work
- Absence of tests significantly **increases** risk
- Tools are **not** up to refactoring **legacy** code bases or **databases**
- Off the shelf tools may **not** be available for your **legacy** language
- Takes **too long** for the business to **sustain** it

Refactor THIS Baby!!



"I don't know about you... but I find these 'legacy' refactoring tools leave a lot to be desired . . .
Yeah, Let's try some dynamite"

©2010 Rederra Research Labs. All rights reserved.

But Can Help!

- Archaeology
- Agile Team
- Tool Chain
- Best Practices

©2010 - 2011 Rederra Research Labs. All rights reserved.

Legacy Archaeology

- Legacy Story Telling
- Listen to the dev and ops Data
- Look at the Code and Data Assets
- Look at the development value chain – people, process and tools
- Publish the Stories, Data and Value Chain and put the essence on the wall

©2010 - 2011 Rederra Research Labs. All rights reserved.

Legacy Agile Team

- appropriate skills for all areas
- cross train /pair Legacy and non-Legacy Developers
- Use playing Coaches who know the territory



© 2010 - 2011 Rederra Research Labs. All rights reserved.

Legacy Agile Tool Chain

- Workstation IDE with cross development
- Modern SCM environment (in front of legacy SCM)
- Unit and Acceptance Testing tools
- Continuous Build and Test appropriate to the technology e.g. HW/SW Emulators
- Sufficient Disk space for logging and testing

© 2010 - 2011 Rederra Research Labs. All rights reserved.

10 Year Old Legacy Java Application

Challenge

- unsupported 10 yr old Java 1.3 Swing, EJB 1.2 application
- application sits in middle of existing core systems .
- urgent need to implement major regulatory

Solution

- Agile practices BDD, TDD ,...
- Major investment in automated testing provided an essential safety net
- Regulatory changes implemented to legacy platform
- In parallel re-platform code base & merge regulatory changes

© 2010 - 2011 Rederra Research Labs. All rights reserved.

A Few Practices for Legacy Developers

1. **Isolate** variability
2. **Make it data driven** – validation, decision tables, state tables, constraints . . . it allows changes at runtime!
3. **Invest in mirroring, sampling backups, caching, emulation** if you can't test live . . . Hardware is cheap!
4. If necessary consider **live database TDD Using Transaction Roll Back**

©2010-2011 Bedarra Research Labs. All rights reserved.

A Few Practices for Legacy Developers

5. **Screen Scrape** if you must, but **automate from data definition**
6. When you renovate a schema **ask for lots of space, add fields, blobs, extension tables**
7. Use **in-memory DB testing** or a **server-side ODBC proxy** to split production DB changes from test changes
9. Enable **continuous inspection** (remote inspector) and **monitoring** or using system logging via **web enablement RSS/ATOM**

©2010-2011 Bedarra Research Labs. All rights reserved.

~~We need a miracle Innovation!~~



©2010-2011 Bedarra Research Labs. All rights reserved.

Thinking Out of The Legacy Box Case Studies of Innovation

1. Insurance legacy + EAP + ISV
2. HR Benefits System Bottleneck
3. Enhancing a 1M+ assembler product
4. Telecomm Legacy Product Enhancement
5. Banking Platform Migration
6. Factory Process Control Modernization
7. Massive Legacy Data Base Migration
8. Real Models to Code

©2010-2011 Rederra Research Labs. All rights reserved.

Insurance Legacy + SOA + EAP

Challenge

Legacy L1, L2, L3,L4; Vendor EAP V1, V2,V3,V4,V5;
Integration Services I5, I6; Commercial Insurance V1, L1
Personal Insurance L2; Vehicle Insurance V2
Rating Engine L2, L3, V3; Billing System V4
Policy System of Record L4 + I5 + V5
Enterprise Software Bus I6

Solution

1. Outstanding BAs define all products in tables
2. Agile experts generate applications from BA tables
3. SI experts build simple interfaces to ESB + Interface Acceptance Tests
4. All vendors required to deliver acceptance tests

©2010-2011 Rederra Research Labs. All rights reserved.



HR Bottleneck


Challenge

- calculations very complex.
- analysts capture in Excel.
- 100 Devs in COBOL too slow
- Agile OO gave only 15% in productivity

Solution

1. Retain 2 Agile OO experts
2. Excel rule checker in java
3. Spreadsheet in Java on mainframe
4. Legacy team deployed on other systems

©2010-2011 Rederra Research Labs. All rights reserved.



Lots of Opportunities to Innovate!

- Use the Cloud for Testing
- Use Ruby, Clojure ... for Scripting Tests & DevOps
- Use F# or Scala for algorithms, C# and Java for middleware – popular in financial engineering
- Use NoSQL for speed => SQL for reporting
- OLAP => High Performance FP
- Continuous Release and Deployment – DeVOps Clouds
- Go Native – Provide access to the HW

© 2010 - 2011 Bedarra Research Labs. All rights reserved.

**Go for Gold!
Embrace and Extend Your Legacy**

**Be Agile! Think Lean!
Innovate! and Prosper!**

Thanks!

© 2010 - 2011 Bedarra Research Labs. All rights reserved.
