

Migrating Legacy Code to an Agile Architecture

Creating an Agile Ecosystem



Matt Anderson, PMP

Director, Program Management

Pratik Nath

Knowledge Architect, Uplift Center, Bangalore



September 28, 2011

Objectives

- **Using Cerner's experience, provide strategies for managing legacy code in an Agile world**

Cerner Corporation

- **Started in 1979, based in Kansas City, MO**
- **Leading global supplier of healthcare solutions, healthcare devices and related services**
 - Focus on creating a safer and more efficient healthcare system
 - Key solutions include:
 - *Computerized Physician Order Entry (CPOE)*
 - *Electronic Medical Records (EMR)*
 - *Personal Health Records (PHR)*
- **Clients in 25 countries serving more than 9000 unique facilities**
 - Hospitals
 - Physician Practices
 - Retail Pharmacies
- **NASDAQ – CERN**
 - \$1.8B revenues in 2010

Cerner Development Profile

■ 8 Global Development offices

- England, Ireland, France, Germany, India, US (MO, VA, CA)
- Majority of development in Kansas City, MO and Bangalore, India

■ 70+ unique solutions

■ Core technologies

- Java, C++, C#, VB6, JavaScript
- Oracle
- AIX, Linux, Windows Server



Migration Strategies and Challenges

Refresh, Migrate, Rewrite or Deprecate

Legacy Migration Guidelines

■ Dept. of Defense (DoD) paper on legacy migration guidelines:

- Guideline #1: Develop a comprehensive strategy with achievable and measurable milestones for each reengineering project.
- Guideline #2: When outside systems engineering services are needed, carefully define and monitor their roles.
- Guideline #3: If new technology is used for a project, provide adequate training in both the technical content and the motivation for change.
- Guideline #4: Establish and maintain configuration management control of the legacy system.
- Guideline #5: There should be a carefully defined and documented process for the elicitation and validation of requirements.

Legacy Migration Guidelines (continued)

■ Dept. of Defense (DoD) paper on legacy migration guidelines:

- Guideline #6: Make software architecture a primary reengineering consideration.
- Guideline #7: There should be a separate and distinct reengineering process.
- Guideline #8: Create a team-oriented reengineering plan ... and follow it.
- Guideline #9: Management needs to be committed for the long haul.
- Guideline #10: Management edicts should not override technical realities.

Major Migration Strategies

■ Refresh

- Recompile code in a newer compiler

■ Migrate

- Move code from one language to another

■ Rewrite

- Recreate features in a new language

■ Deprecate

- Allow solution to move into unsupported model and eventually remove from portfolio

Migration Considerations

■ Business Value of current solution

- Market penetration
- Client satisfaction

■ Functionality

- Add new
- Like for like only
- Simplify

■ Cost of options

- Duration

■ Deployment

- Big bang or phased roll-out

■ Support strategy

- Legacy and new

Activity – Define Your Key Drivers

Potential Drivers

- **Supporting ongoing innovation**
- **Improving application security**
- **Creating a software architecture that scales**
- **Improving solution maintainability**
- **Leveraging open standards/interoperability**
- **Avoiding support gaps for legacy systems**

- **List the key drivers at your organization for a migration**

Cerner's Drivers

- **Migration of viable legacy solutions to updated technology platforms**
 - Support continued innovation
 - Improve Solution Maintainability
 - Extend lifecycle

- **Investigate addition of new capabilities afforded by newer technologies for new development**
 - Create agile ecosystems

- **Maintain Cerner as an attractive destination for top development talent**

Activity – Define Your Key Challenges

Potential Challenges

- **Low value return to end users**
- **Innovation constrained by legacy tooling**
- **Substantial developer training needs**
- **Impact on developer and team productivity**
- **Resistance to change across development group**

- **List the key challenges you may face at your organization**

Cerner's Challenges

- **Incremental rollout required due to large, complex client install base**
- **“Big Bang” solution would present significant risk to all stakeholders**
- **Low client value return for rewrite strategy**
- **Innovation and maintenance constrained by legacy tooling**
- **Create repeatable process for technology upgrades**
- **Maintain team flexibility based on their unique markets**
- **Potential retraining of developers proficient in legacy technology**
- **Large segments of shared code leveraged by teams that may migrate at different time periods**
- **Change impacts to development community**

“The skills of the software development team are crucial for project success and can enable or doom the adoption of any new architecture or technology.”

***Dave West, Senior Analyst
Forrester Research***



Cerner's Strategies

Incremental Migration

Overall Strategy

■ Create a Center of Excellence – “Uplift Center”

- Work with each solution to determine best path
- Expedite technology migration by capitalizing on learning
- Orchestrated migration
- Manage vendor relationships
- Modified Agile process
- Global development effort with Kansas City and Bangalore centers

■ Open Forum for Q&A from Development Community

- Transparency in decision making
- Seek open feedback directly from community
- Create a top-down and bottom-up synergy
- Opportunity to “market” ideas and results to entire community to create positive momentum
 - *Utilized Web 2.0 Collaboration software*

VB Migration Strategy

■ Incremental

- Leverage COM Interoperability

■ Create vendor relationships to build and supplement internal expertise

- Technology
- Education

■ Focus on “Like for Like”

- Minimize client training impacts
- Minimize risk profile (scope, testing, etc)
- Speed to completion
- Variances evaluated on case-by-case basis

VB Case Study - SurgiNet

■ Snapshot

- Primarily VB6 with shared componentry leveraged by 2 other solutions
- 300K LOC
- 6 month timeframe to migrate and fully test internally

■ Migration Strategy

- Code Migration from VB to C#
- Leverage migration partner to assist with automation
- Upgrade COM OTS components to .NET
- Like-for-Like only
- Leverage existing test cases to create test scenarios
- Utilize COM Interfaces for interaction

VB Case Study - SurgiNet

■ Team

- Global team
- Solution knowledge
- C# knowledge
- Start project with entire team in Kansas City for 4 weeks to “build team” and set common expectations

■ Development Methodology

- 4 week iterations focused on migrating related components in each iteration
- Done = Code compiles and passes engineer component layer testing
- Integrated Testing and issue resolution completed in subsequent iteration

■ Communication

- Daily Stand-up with full team via teleconference
- Web 2.0 Collaboration software to facilitate shared documentation and discussions

VB Case Study - Results

■ Migration completed on schedule

- Needed to add additional capacity for the last half of the project to account for unplanned attrition and early learning curve delays
 - *Leveraged another team and used it as their knowledge transfer for their own migration*
 - *Knowledge gained delivered a 15% reduction in migration timeline for 2nd team*

■ Like-for-Like Target Met

- Business Owners assisted in testing
 - *Frequently did not realize they were testing the new application*

■ Live at multiple partner client sites

- Minimal issues reported post go live
- No additional training required
- Legacy version now completely replaced for new and upgrading clients

■ Associate Impacts

- Majority of development community eager to adopt new technologies
- Communication and collaboration strategies are a model for other large projects

VB Case Study - Results

■ Migration Best Practices Established

- Uplift Center retains lessons learned for continuous improvement
- 5+% Productivity gains for new teams
- Reduced learning curve for new teams
- Historical and collaborative documentation prevents decision rehashing

■ Current/Future Plans

- Continue with Incremental Migration by Solution
- Multiple teams migrating concurrently
- Strategies and timelines vary per team
- Timeline based on business strategy for each solution market

Activity – Identify Critical Success Factors

Potential Factors

- **Agile methodology**
- **Team collaboration software**
- **Executive management support**
- **Targeted training & skills development**
- **Integrated testing**
- **Software architecture reengineering**

- **Identify what your critical success factors would be for your migration**

VC Refresh Strategy

■ Refresh all solutions concurrently

- Moved from VC 6 to VC 10 compiler
- Plan for simultaneous release
- Perform bulk of refresh and testing in single 3 week iteration

■ Leverage Uplift Center migration experience

- Create internal tools to assist with recompile
- Provide initial versions of recompiled code with common problems automatically corrected
- Leverage Web 2.0 technologies to share proposed resolutions

■ Focus engineering effort on regression and issue resolution

VC Refresh - Results

■ Preparation

- 3 months for Uplift Center to create tooling to fix majority of compile issues
- Entire VC code base (19+M LOC) recompiled and new files provided to teams immediately prior to joint start

■ Time to complete

- 3 week target met for majority of issues
- Remaining minor issues completed before initial release

■ Regression testing

- Exposed value of automation and generated positive support to implement NUnit and other automation testing for the VC codebase



Additional Topics

Questions and Answers