

Re-Platforming with IB-ARM

The Need



Practically every organization has one or more applications where a current and proper understanding of the application is lacking or the application's platform no longer fits the corporate architectural strategy. As time goes on, these applications are treated more and more as 'black boxes' and many are targeted for (eventual or immediate) re-platforming.

Reasons to keep a system include:

- It works, why change it (and, possibly, break it)
- Resources are busy on other business initiatives
- Re-platforming costs

Reasons to re-platform include:

- Current platform costs
- Withdrawal of vendor support for the platform
- Expected shortage of resources in outdated technologies
- Opportunity to bring the system in line with current corporate architectural strategies
- Opportunity to decrease application maintenance costs by allowing the easier addition of new functionality

At the end of the day, re-platforming is the appropriate course when, on balance, the reasons to change significantly outweigh the reasons to keep the status quo. IB-ARM provides a key component in shifting that balance.

Re-platforming Challenges

Scoping, Estimation and Analysis

Understanding the cost and risks of re-platforming requires a complete and accurate analytical inventory of the current system. Key cost factors include: the amount of real vs. obsolete code, the functionality involved, the relationships within and to other systems, to what extent code can be reused, and data migration requirements. At this stage, estimates typically have an accuracy of plus/minus 100%, sometimes even more.

Once the decision is made to go ahead with re-platforming the application, the first and most critical step is to create a further, detailed understanding of the components of the current application and its functionality.

The commonly available aids for estimation and analysis may not be cost-effective for this purpose. Spending a significant amount of money on application mining tools (including the cost of the tools, and investments to learn and maintain the tool) may not be feasible, especially when the very purpose of the project is to move off that particular platform. Why would an organization spend a significant amount of money and time on an application mining tool that supports the very platform it is trying to move away from? Payback on such investments is very questionable, at best. Consequently, most organizations tend to rely on highly manual processes.

Design and Coding / Re-platforming

Manual analysis often produces loopholes that may, or may not, be noticed and corrected during design and construction. When in doubt, programmers will check back into the old application, again manually. Typically, the few resources who fully understand the old application become true bottle-necks in the process.

Data Migration

Many times the old data structure is manually migrated into a new format. Even if a tool is available to perform the migration, the understanding and inventory of old data structures is performed manually so that it can be fed into the tool.

Testing

The creation of a proper (regression) test, with thorough and reliable test scripts, again requires a good understanding of the old system. The quality and completeness of this analysis will directly affect the effectiveness of the testing phase. Incomplete analysis hinders testing in at least two ways: Coverage is not correct (either too much or too little) and typically testing uncovers issues that should have been caught during analysis. Fixing these issues costs much more at this stage.

Implementation and Production

To meet deadlines, delays during the project typically shorten the time allocated for testing. An improperly tested system will take many months to stabilize in production, adding significant costs to the overall re-platforming project (even if the costs are only realized at the corporate level).

Knowledge creation for the new system

The learning curve for proper understanding of the new environment can be significant. Organizations must spend significant amounts to transfer knowledge from the re-platforming project team to maintenance teams. Almost always, documentation becomes outdated as soon as it is delivered. Many months, if not years, pass before system knowledge is built up.

In summary, re-platforming projects are known to be among the most risky, error-prone, costly and lengthy undertakings — *until now.*

Benefits of using IB-ARM

IB-ARM implements a revolutionary new approach to re-platforming projects. It provides a complete Roadmap for the re-platforming effort, with accurate views of the application before, during and after re-platforming. These views are continuously and dynamically updated with changes as they happen.

IB-ARM is used to develop a repository of the application components, including source code and documentation. This repository is built to:

- Assist in the estimation of the effort for re-platforming
- Be the Roadmap during re-platforming (providing a live and dynamic source of the 'current picture')
- Create a repository of business rules and corresponding code
- Enable reuse of source code (where possible)
- Provide the base documentation for the new system
- Provide an *on-going knowledge asset* that allows cost savings during each future systems change

Projected savings are realized in the following areas:

- Scoping
- Identifying and analyzing current business rules
- Estimation, Analysis and Design
- Testing and QA of the New System
- Creating documentation for the new system
- Current maintenance costs for the application (until re-platforming completes)
- Future maintenance costs of the application

In addition to these savings, the re-platforming project is delivered **faster**, with **higher** and more reliable **quality** and the new system will have a dynamically updated maintenance support base, providing on-going yearly savings.

The Process

1) *Streamlining of the current application*

By creating a solid, connected inventory and view of the application, the system can be made 'lean and mean' before any re-platforming effort commences. In a typical implementation, 15-20% of code is found to be unused or for some other reason not worth re-platforming (such as outdated functionality). To reduce the costs of creating the new system, it is essential that the starting ('from') view be as compact as possible.

2) *Assistance in current maintenance of the application*

By loading source code objects (such as COBOL, copybooks, JCL, Natural and Assembler programs) into the Repository and creating a solid analysis environment, current maintenance efforts can be improved. While this is a temporary cost saving, the system will need to be maintained and even upgraded from time to time while the new system is being built. Using IB-ARM results in immediate and direct cost savings as well as freeing up resources for the re-platforming effort.

3) *Creating an inventory of Business Rules*

Once the Repository is built, business rules can be extracted from the code and stored in the same Repository, linked to existing code. This inventory of business rules can act as the roadmap for the new system, providing a clear and concise base for estimation, development, testing and implementation purposes. Identified code may then be reused or re-platformed in a much more effective manner.

4) *Including current / creating new documentation*

Existing documentation can be updated and streamlined to correspond to the availability of the IB-ARM Repository. In our experience, the need for conventional documentation is significantly reduced once the IB-ARM Repository is available. After all, most IT documentation is simply restating what is already in the source code – and becomes outdated once the code changes. With IB-ARM, such documentation is dynamically and continuously available from the code itself.

As a result, the content for much of these manuals can be generated automatically and made available in the Repository. Additional documentation can also be generated in the form of diagrams and process flows.

5) *The Roadmap*

While the focus of the IB-ARM Repository is on the presentation of dependencies between various objects, an intelligent rules-based engine can be added to guide the usage of the objects contained in the Repository. The information is presented in an easy to follow, graphical interface so that the Repository will assist in answering questions such as:

- What source code can be reused?
- What data needs to be migrated?
- What are the business rules and how are they implemented?
- How do the business rules relate to the models?
- How do the models relate to architecture objects?
- How do new development objects relate to architecture objects?
- How do test plans relate to development objects?

As the new system is being built, its objects are added to the Repository, clearly showing the transformation process from old to new. In essence, the Repository acts as the Roadmap for the application – before, during and after re-platforming.

6) *Continuous Knowledge Base*

Once the new system is implemented, the IB-ARM Repository becomes a complete and actionable knowledge base, providing a dynamic and cost-effective foundation for future training and maintenance. The benefits of IB-ARM long outlive the re-platforming project itself.

About Information Balance, Inc.

Information Balance Information Balance, Inc. was founded in 1988 to provide consulting and training services to the IT industry. Over the years and through continuous growth, the company's field of expertise has significantly broadened and now includes all aspects of Systems Development and Integration, covering all platforms including mainframe, client/server and the Internet. Many of these areas are supported by formal training curricula.

Today, Information Balance, Inc. is a well-established firm with offices in Canada and Europe. Sustainable, controlled growth has been the mainstay of Information Balance's success, with over 30% growth year over year. Information Balance has been awarded the following accolades supporting its corporate excellence.

- Financial Post Fast 50 ('98, '99, '00)
- Profit Top 100 ('99, '00)
- Andersen 50 Best Privately Managed ('99, Regional Finalist)

Information Balance maintains on-going professional relationships with many large Fortune 500 corporations and government organizations.

For more information, please call 416-962-5235, e-mail infobal@infobal.com, or visit our website at www.infobal.com.