



## WHITE PAPER

# Realizing *Agile IT*

## ...for IT Leaders

### **Abstract**

*“Agile IT” represents the ultimate goal for IT transformation – enabling IT to respond rapidly to changing business needs. Most large IT organizations have determined that business as usual won’t produce the needed agility and are adopting modern approaches. However, as we have learned, adopting new methods on a piecemeal basis won’t get you there either. An integrated approach is needed. Agile IT combines best practices from agile methods, service oriented architecture and model-driven development under a management framework known as portfolio transition engineering.*

*Adopting Agile IT within the typical IT environment requires additional capabilities and adjustments to current IT management practices. Agile methods provide rapid delivery of capability increments. However, to scale agile methods, project teams must be able to work independently. Service Oriented Architecture (SOA) provides the organizing paradigm that allows smaller teams to achieve parallel development on a massive scale. This creates pressure on release management and development coordination disciplines. Model-based development facilitates precise communications and synchronizes products across teams. Portfolio transition engineering enables the organization to implement and assimilate the rapid changes within a coherent framework.*

*This paper explains what Agile IT is, why it is needed and provides insights into how to prepare your organization to become more responsive to business challenges today.*

May, 2012

# Realizing *Agile IT* for IT Leaders

By Dave Mayo, President of Everware-CBDI, Inc.

## Introduction

Across the government and private sectors, IT managers are struggling to meet their stakeholders' needs and expectations. We are facing a near-perfect storm of increasing demands, *real* budget pressures and ineffective procurement and solution development processes. Combining this with the failure rate for large IT programs, the need for a new IT paradigm is clear. In addition, corporate business model transformations are increasing the pressure on IT organizations to come up with viable approaches to manage and modernize IT.

*Agile IT*, a term coined by Gartner, represents the goal of IT transformation. It consists of managing a portfolio of IT assets, along with the supporting infrastructure and governance, that enable an organization to compose solutions to address issues on a near real-time basis. This IT capability is the foundation for aligning IT resources with business imperatives and allows IT to keep up with business change, if not lead it.

Agile IT requires a significant process improvement in architecture and delivery practices. In the Agile IT world, existing applications must be rationalized, restructured, service enabled or replaced to reduce dependencies and risk in major programs. This should not be attempted in yet another large program, but rather managed through effective architecture that governs smaller work packages to progressively transition the portfolio to higher levels of separation. This approach allows continuous evolution on a rapid delivery basis that is capable of quickly yielding new solutions.

The new process should combine best practices from agile methods, service oriented architecture, application modernization and model driven development into a transition engineering approach that allows transformation to begin immediately – with low risk, incremental solution development to meet new challenges. This approach, pioneered and proven by Everware-CBDI, is referred to as *Service Oriented Application Modernization* (SOAM).

## IT Challenges

Transforming IT throughout industry and government is imperative. Large organizations face many significant challenges with respect to IT. Some of the most important ones include:

- Major IT program failures – billions of dollars lost due to the cancelation of failed programs and the under-performance of many others
- Inability of applications to interoperate or share data – due to organizational, semantic and technical issues
- Majority of IT budgets spent on O&M, leaving little to fund development of new capabilities – resulting in a large backlog of enhancement and new functionality requests
- Traditional development lifecycle takes too long – driving large contracts and large contractors – and requirements change during development
- Application redundancy (same capability is built into many applications) and inconsistency (enforcement of the same business policy differs in multiple applications)
- Redundant infrastructure – significant funds wasted
- Reuse is difficult to achieve even when it appears to be common sense.

## The Need for IT Transformation

While most organizations recognize the need to modernize their IT business model, few are undertaking the holistic set of steps necessary to bring it about. The high level objectives for IT transformation include:

- Make IT more cost effective (reduce the per unit cost of delivering capabilities)
- Provide better support to the business – increase IT agility to enable functional agility
- Reduce the risk and cyletimes associated with IT development programs
- Reduce redundancy and improve consistency through reuse

A variety of actions is required to achieve these objectives. They include:

- Rationalize IT platforms and infrastructure (virtualization, cloud)
- Update the IT process (acquisition through deployment/maintenance; modular, incremental SDLC)
- Improve roles and skillsets (CIOs, acquisition specialists, program managers, etc)
- Improve communications and collaboration (business – IT, internal – external, providers – consumers) using semantic approaches and integrated product teams
- Institute evaluation and feedback mechanisms to evaluate efficacy

## Overview of Service Oriented Application Modernization

As should be apparent from the above, a new paradigm is required to get IT organizations out of this quagmire. Everware-CBDI has successfully applied SOAM for commercial and government clients. SOAM contains the core components of Agile IT: best practices from *Service Oriented Architecture* (SOA), *Model-Driven Development* (MDD) and *Agile Methods*. At first glance, these might seem to be incompatible, even diametrically opposed; however, we have found that each provides a balance to the others and the result is highly synergistic.

### Agile Development Methods

**Agile Methods** are adept at quickly identifying requirements and creating software to serve specific business purposes. However, Agile teams often begin with little or no formal documentation (e.g., requirements or architecture) – the emphasis is on producing functionality (executable code) quickly that can be observed and assessed by knowledgeable and empowered users. Therein lies the beauty of Agile: the solution evolves based on *rapid feedback from users* into the development process. But what if you don't have access to knowledgeable users who are empowered to make decisions about the solution design for the organization (which is typically the case for cross-organizational projects)? Moreover, Agile Methods work best with small teams that include highly-skilled developers who are well versed in the tools across the lifecycle and technology stack. The result is that Agile is more difficult to scale. You can't just grow the team size – it becomes unwieldy. And if you organize as multiple teams, then you have communication/coordination issues. That is where SOA comes in.

### Service Oriented Architecture

**SOA** is an architectural style in which solutions to user needs are composed from services that are accessed via well-defined interfaces. SOA allows solutions to be designed as a collection of interacting modules that can be reused in multiple ways. These modules may include wrapped legacy transactions or shared services from other organizations, perhaps offered via cloud computing. There are many benefits derived from SOA – such as lower cost from not having to (re)build functionality, greater consistency in the enforcement of business rules, flexibility from the plug and play nature of services, and better alignment with the business needs. But for the development process, the key advantage is the ability to divide a solution into independent modules that are well isolated from each other. Obviously, you need teams to integrate these modules into solutions (à la “twin track” development); but, with an overarching service oriented architecture, the objectives of each team and their interrelationships should be clear.

### Model Driven Development

Of course, with SOA the number of moving parts increases dramatically. Managing them is where **MDD** comes in. MDD uses models to document the requirements, design the solutions

and services, and generate the code or other executable artifacts (e.g., BPEL, business rules, etc.) that become the solution, along with much of the documentation demanded by the standard SDLCs. Using MDD, the solution is developed at an abstract level and refined to more detailed levels. For example, typically a model that describes the business solution, independent of technology, is created and then transformed to a platform-specific version by applying design patterns and platform conventions and constraints. From the platform-specific models, the code is generated that become the executables of the system. In the Everware-CBDI MDD environment, some amount of code customization or extension is required (typically 10-20%), but the generated code is intended for human consumption and hand modification.

There are three key advantages from MDD in the development process. The first deals with managing the IT outsourcing process. With MDD, requirements provided to the contractor are well structured – at various levels of abstraction – and can be developed in an agile fashion. This provides a mechanism for improving communications, as well as providing rigorous specifications that form a contract with the outsourcer. Secondly, much of the functionality is generated – accelerating development, reducing the coding effort and associated errors, and improving the consistency of the code base. This also allows the developers to focus on customizing the code that directly corresponds to the users’ requirements (value-added effort), rather than on the application “plumbing” that integrates all of the components of the solution. Third, MDD serves a coordinating role in synchronizing the efforts across all of the development teams. For example, changes in one module/model are automatically rippled across all related modules (avoiding the infamous ‘I fixed the error in 5 of the 6 places where it was found’ problem). This third benefit also implies more efficient application maintenance for increased agility and a lower total cost of ownership (TCO) of the software. In fact, a major part of the ROI is derived from the reduced cost of maintenance and enhancements – which typically accounts for the majority of the TCO.

### **Application Modernization – *Portfolio Transition Engineering***

All of this sounds great – for new system development. However, the vast majority of organizations face a legacy environment that constrains new development – creating a vast backlog of enhancement requests, despite the fact that most IT organizations devote upwards of 70% of their IT budget to application maintenance. Almost all IT organizations recognize the need to transform to an environment where most of the IT budget can be devoted to delivering new and improved capabilities driven by and supporting the business. The primary challenge is how to migrate the collection of legacy applications (some of which may be decades old – with a fragile code base and poor documentation) to a flexible IT environment.

Everware-CBDI’s Portfolio Transition Engineering is an over-arching management approach to evolve IT from a rigid legacy environment to solutions based on a flexible portfolio of loosely-coupled services. This approach leverages the iterative nature of SOAM to gradually evolve capabilities to the new paradigm. It is driven by business priorities and employs a “pull” strategy – identifying what capabilities are required and determining what legacy or newly developed assets can be used to meet the requirements – pulling them into the transition process. (Note:

typical IT modernization approaches exhibit a “push” strategy in which legacy applications that need to be replaced are pushed into a modernized, yet still stovepiped, form.)

Portfolio Transition Engineering combines business capabilities into releases that are provided to end users on a cycle that optimizes the assimilation of new capabilities. For example, although our Agile development methods produce new, tested functionality on a frequent basis (every 2–4 weeks), the functionality is released to users at a frequency optimized for user effectiveness (typically 3–6 month cycles) – to account for user training needs and to avoid functionality churn.

## IT Management Considerations

While the benefits of applying SOAM are considerable, adopting it amounts to a significant shift in IT management. The transformation from a waterfall lifecycle to agile, service-oriented development is a major paradigm shift and much preparation is required – e.g., roadmap planning, organizational change management, education, etc. To be successful, we recommend a just-in-time planning approach (focused on the next releases) within an over-arching program management framework.

## Conclusions

As you can see, at Everware-CBDI we have taken the agile approach and combined it with model-driven service architecture. The end result is a significant reengineering of the IT process. The focus changes from managing documentation and producing code to developing and refining models from which much of the code and documentation can be generated. This is not only more efficient, it enables the rapid turnaround associated with agile methods to be combined with architectural and engineering integrity that is critical for developing software solutions on a large scale. As should be clear from the above, each of these disciplines (SOA, MDD and Agile) augments the others to improve the overall approach. The result is greatly increased productivity, flexibility of solutions, and responsiveness to the user needs.

---

**Dave Mayo** is President of Everware-CBDI, Inc. His expertise includes strategic planning, business case development, enterprise architecture, service oriented architecture, and economics. He is a frequent guest lecturer and invited speaker at international conferences. He is a graduate of the University of California, Riverside and holds a Masters in Economics from the University of British Columbia.

**Everware-CBDI** is a specialized consulting firm dedicated to the transformation of business and IT – leveraging proven, advanced approaches such as SOA, agile and model-driven techniques. These techniques are combined with a business focus to evolve organizations from stove-piped, hierarchical structures to modern service-based, loosely-coupled structures that enable responsiveness and agility. We offer transition planning guidance, along with practical, on-the-ground IT development support, to assist organizations successfully accomplish their transformational objectives. If you are interested in applying SOAM in your environment, please use our [contact form](#) to get in touch.