

Business goals + pragmatism = middleware success

Martin D. Fincham
Managing Director
Alchemy Computing Ltd

Introduction

Many analyses that attempt to provide a blueprint for success are in fact little more than a cookbook of the pitfalls to avoid, assuming that the act of simply avoiding failure will automatically translate into project success. To Mr. Fincham, achieving success is meeting — and ideally exceeding — business expectations. The wise know that the most effective way consistently to meet and exceed expectations is to set them correctly in the first place. Therefore perhaps the most valuable information he has to share in this analysis is how to set expectations for a middleware project/implementation.

What credentials does Mr. Fincham have? In 1993 he founded Alchemy Computing, one of the first UK companies to make a living exclusively from selling and supporting middleware technology. Its success has been determined solely by its ability to implement middleware. It has no other streams of revenue as a crutch. This analysis, therefore, distils four years of practical experience into a form which others can use for middleware project planning and implementation.

Setting expectations

Business users and IT staff often have different definitions — and therefore expectations — of project success. IT can deliver a system where everything works in accordance with the functional specification. Yet when you visit the end user department the application will just be ‘sitting on the shelf’. Such a scenario can be regarded as a technical success but it is clearly a business failure.

Unless your company’s line of business is developing dream technical architectures, anything less than meeting the business users’ expectations is failure, plain and simple. I will expand on how to set expectations correctly as this analysis unfolds. Nevertheless, I do want to make the point, right-up-front, that this gulf — between what constitutes a technical success and a business success — is the primary contributor to middleware project failure.

Building a case

One of the most compelling themes running through our projects is that an increase in employee productivity is central to building a solid business case for middleware. It is rare to find a middleware implementation that can be cost justified on direct operational savings alone.

An effective approach (which both we and our customers have adopted) is to look for opportunities to provide seemingly small — but real and measurable — productivity enhancements to as large a group of users as possible. Rather than attempting to deliver a ‘blockbuster’ system that revolutionizes working practices for a select few, consider instead the organizational impact of raising the productivity levels of hundreds — or even thousands — of ‘factory floor or office’ workers by just a few percentage points. Since the payroll is still the largest operating cost for many organizations, delivering a broad increase in user productivity is a highly effective way of winning support for middleware projects.

By the way, even as we are advocating this modest, incremental approach we are not suggesting that large gains cannot be delivered by your first middleware project. For instance we have known organizations completely eliminate paper from a previously paperwork intensive function with their first middleware project.

The key is to ‘think outside the box’ when it comes to identifying the opportunities for middleware.

Do not become constrained by your current methods for setting project expectations.

Examples

Two examples of business functions that readily lend themselves — in our experience — to this incremental approach are:

- **customer service**
- **materials management (procurement and requisitioning).**

Recently, a mid-size electric utility embarked on a major restructuring program so it could compete effectively in a new, deregulated market. Customer service was quickly identified as a critical factor in this success.

Given the need for urgent results it chose to enhance — rather than redevelop — its existing Customer Information System (CIS) — by deploying middleware and a new graphical front end for its Customer Service Representatives (CSRs). This case proved to be an excellent example of correctly managing the risk/reward ratio inherent in implementing new computer systems. The company had two choices:

- **either significantly modify its existing CIS application, by implementing a three-tier client/server architecture**
- **or embrace and extend the CIS application via the use of middleware and personal computing.**

This is where, in our experience, the 80:20 rule kicks-in. You should first implement the 20% of the technology that delivers 80% of the business value. By following this rule you will not always win support from the technical purists. Yet you will win friends in the executive suite.

Back at the electric utility, management (from the Chairman down) were focused on making sure this middleware project was a business success. To make the meaning of ‘success’ clear and measurable, management set a single key expectation: increase the customer grading rate of CSRs (on how knowledgeable they were in handling a customer call) from 86% to 88%.

To orient the whole organization toward project success, management linked this service improvement goal to a corporate-wide employee bonus scheme. For meeting this key expectation all employees received an additional $\frac{1}{2}$ of 1% salary bonus. After a matter of months the Phase 1 deliverable was deployed and by the end of that year the customer grading rate had risen to 89%, thus exceeding the project's expectations.

Management was more than satisfied. In consequence, the middleware approach won credibility across the organization. Already it is planning other middleware projects, including:

- materials management
- payroll
- human resources.

The middleware divide

I have never found any printed material which attempts to quantify the effort required to implement different types of middleware. Unfortunately, I do not have sufficient hard data to compile such a league table myself, but I have been involved in numerous evaluation exercises from which it is clear that an enormous spectrum exists.

Today a fundamental element of Alchemy's mission is to deliver rapid business value. As part of this the middleware technology which we implement should be as 'non-invasive' as possible. Quite simply, the less you have to meddle with a customer's existing environment:

- the quicker you can deliver a solution,
- the less politics you have to overcome, and
- the less risk you introduce.

I remember one project for a large UK manufacturing company which illustrates the magnitude of this divide. After a nine month middleware evaluation process the decision came down to a two horse race between us and a strong competitor. It was clear to everybody that both middleware approaches could deliver the desired application functionality, and therefore meet the business requirements.

However, the implementation approaches differed radically. The client had an IBM mainframe environment running CICS and 20 million lines of

COBOL. The alternative approach to our proposal was RPC based and required each CICS transaction be modified before it could be 'called' from the new GUI desktop application:

- the host functionality required for this project was spread across 45 different CICS transactions
- the RPC vendor quoted 7 man days per CICS transaction for development and testing of modifications
- in other words, the RPC implementation would have required 315 man days of changes to the host environment before the new GUI desktop application could even be built and tested.

In contrast, our proposal for delivery for the whole project — including host integration, design, building and testing of the new GUI desktop application — was less than 315 man days. So, in the time it might have taken just to prepare an environment for the installation of one type of middleware it was possible to have an entire project delivered using a different type of middleware.

This spectrum is representative of how marked are the implementation divergences between different middleware approaches. Although technical purists could have argued that, in this example, the RPC approach was more 'technically elegant' or even 'more modern' than our approach, the business found the extra cost and risk (particularly the regression testing) impossible to justify. It was not surprising that it chose the more pragmatic approach.

Fundamentals of (middleware) project management

Firstly, I should differentiate between a middleware project and more 'traditional' development projects, irrespective of whether the latter use modern RAD/OO methodologies or classic waterfall lifecycle development. When middleware is involved it usually forms a component of the overall application architecture and is thus exposed directly to the application developers.

However, middleware is typically an area of confusion and uncertainty for developers, particularly those who have become accustomed to 'rolling their own' communications code. Additionally,

infrastructure specialists — such as DBAs or Network Managers — normally have limited experience or understanding of middleware. Traditional skills and skill boundaries, therefore, need to be re-evaluated if you are to manage effectively a middleware project.

Of course, the fundamentals of good project management generally apply to middleware projects as well. Experienced project managers will tell you that there are only three things necessary to make any project successful — communication, communication and communication. In a middleware project you will likely be sailing in uncharted waters, with a mixed crew — internal IT, end users, external consultants, contractors, etc. Good communication systems and skills will be more important than ever.

Courage, conviction and competence

It is no coincidence that every middleware project that we have ever been involved in has included supplemental expertise from one or more of the following:

- **ourselves (as the middleware vendor)**
- **a consulting firm or systems integrator**
- **specialist contractors.**

In fact, we have walked away from projects where organizations have resisted external help, because we have rated their chances of success as very low. That said, while buying-in expertise, do not forget that you will live with the middleware product long after the initial project is completed. Once working, middleware will become a part of your infrastructure. Ensure that proper knowledge transfer takes place between vendors and external consultants and your internal staff.

In traditional software engineering circles a ‘quick win’ may be synonymous with ‘quick and dirty’. In fact you may even be skeptical that real business value can actually be delivered in a matter of a few weeks or months. Other key project qualities that we have identified are courage, conviction and competence. It is clear from our own business activities that the first use of middleware within an organization takes courage. No matter what case studies are available — or what proof-of-concept

has taken place — it is inevitably an act of faith to implement a new middleware product. This is where our key principle of the ‘quick win’ comes to the fore.

User confidence is earned, plain and simple. There are no short-cuts. By finding some quick business gain, or by alleviating some immediate business pain, the risks associated with implementing a new middleware product can be balanced with the anticipated rewards. The ‘quick wins’ that we have experienced in the field are measured in weeks — sometimes a few months — and deliver rapid business value.

During this first ‘quick win’ it is imperative that the project sponsor and manager maintain absolute conviction in meeting the project’s expectations. As an infrastructure component, middleware often spans — or ‘touches’ — different departments, companies, skill sets, systems, etc. Politics and individual agendas are rife. Conviction is needed to ride (rough-shod, if necessary) over the political ground to ensure that project success prevails.

This leads on to our final quality, competence. It is important to build the right specialist team to work with your chosen middleware product. People experienced in middleware are still thin on the ground. They are difficult to find — and yet vital to a successful project (nobody ever said it was going to be easy).

I have the evidence in the form of a brief case study. I have selected this particular one as I feel that it epitomises a well run middleware project.

Case study: new system in 15 weeks? ‘Oil be surprised’

Recently, a pan-European oil company needed to implement a new procurement strategy. Its existing global purchasing system — running on Oracle Financials — had a complex, character-based interface that, while acceptable to procurement professionals, was regarded as too complex for the occasional user.

A key component of its strategy was to enable on-line requisitioning and purchase order processing by ‘occasional users’, extending the use of the purchasing system beyond the central purchasing organization. Thus the Company sought a technical solution to replace the complex, character interface with an intuitive Windows GUI.

The Company's in-house team advised that such a system would take at least 12-18 months to deliver. This time scale was unacceptable: the business needed a production system in just three months.

Previously the oil Company had used a consulting group (PA Consulting) to deliver a Web-enabled data warehouse. It was so impressed with this system that it invited PA back to assess its new requirement.

A small team from PA's Information Systems Implementation Practice delivered a proof-of-concept in three weeks in order to build the confidence that a Web-based application could be developed to deliver the simple purchasing functionality required, with minimal roll-out effort. This team also proved that the Web-based application could be attached to the existing global purchasing system using off-the-shelf middleware, thereby:

- **obviating the need to change the existing system**
- **massively reducing the project cost and risk.**

In the solution, middleware connects a new interface — called the 'Simple Front End' (SFE) — with the existing Oracle Financials system and allows purchase requests to be passed into the underlying Oracle Financials databases. Two integration mechanisms were planned:

- **in the short term, middleware is used to integrate the SFE directly with the existing Oracle Forms-based applications; MitemView messaging middleware (supplied by us) provides the interface between the new Web front end and Oracle Financials**
- **in the long term, a standard interfacing object called an Interface Denormalised Entity (IDE) will be developed for Oracle Financials, which will allow external programs such as SFE to be connected through a common and standardized route similar to an API; this 'loose coupling' arrangement will allow SFE and new versions of Oracle Financials to evolve without affecting each other.**

Using PA's Rapid Systems Development (RSD) methodology, the team delivered a production system in a short time frame. Indeed, the first invoice

passed through the system within 15 weeks of starting the project (recall that the in-house team had quoted 12-18 months). The oil Company was delighted.

The system has now been rolled-out to 12 countries across Europe. The Web browser approach has allowed a GUI interface to be rapidly deployed in three country languages. Just as significantly, the middleware has provided a robust, high performance connection to four different versions of the underlying Oracle Financials transaction processing system.

Looking back, I regard the RSD methodology as an excellent model for delivering 'quick wins'. Five aspects of RSD shaped the success of this project:

- **iterative development was used to place procurement functionality in front of the target end users at the earliest opportunity**
- **business involvement was actively encouraged (this included demonstrations and workshops)**
- **a rapid delivery team culture was encouraged between all members of the team**
- **the project and solution were structured to decouple dependencies with external events**
- **the project produced the focused documentation necessary for rapid delivery of a solution.**

The policy of the SFE project was that the IDE mechanism represented the most elegant, long term integration method. However, in the short term, middleware was used to deliver rapid business value (read 'quick win'). Consequently an architecture was developed which caters to both integration methods. Given the requirement for the new user interface to be accessible by many business users within the oil company, the SFE used Web technology and the Company's existing Intranet.

It is also important to note that 'quick win' does not automatically translate to 'quick fix'. It is vital to choose your middleware and architecture carefully so that, as the acceptance and use of middleware in your organization grows, the technology should still deliver the performance and reliability levels demanded by real-time transaction processing.

In this context, your first middleware project is usually the start of a journey. It is not a destination in its own right. Plan, therefore, for that journey and not a day trip.

Alchemy's golden nuggets

Alchemy's golden nuggets are derived from its years of implementing middleware. I would like to see our 'golden nuggets' evolve, and to become further enriched by the experience of other practitioners. Therefore, if you have a comment, criticism, pearl of wisdom or simply an observation to make, please feel able to email me at mdf@alchemy.demon.co.uk; I should also pay tribute to the following who have already helped evolve these nuggets as well as this analysis: Dr. Aurel Kleinerman and Gale Aguilar of MITEM Corporation, Ian Evetts of Cap Gemini and Adam Hughes of PA. Our nuggets include the following:

- **find a 'quick win': be business led, not technology driven**
- **evaluate the alternatives: choose the middleware approach that fits your budget, time scale and risk ceiling**
- **commission a proof-of-concept: typically this should take 5-10 man days (and no more than 20 man days)**
- **build a solid business case, keeping increases in user productivity central to it; rarely can a middleware project be justified on direct operational cost savings alone**
- **make your own choice of middleware technology, as you will have to live with the technology long after the analysts and consultants have left your building**
- **build internal technical competence: and buy in experience and expertise to encourage initial project success, as well as to facilitate knowledge transfer**
- **follow a stepwise implementation process, with small increments where the end user can track and judge success**
- **identify and manage the politics: middleware projects span departments, skill sets, agendas and even whole companies — so expect to integrate people as well as systems**
- **measure results and communicate success: your first successful middleware project is a catalyst — exploit your early success to build confidence for future projects**
- **capitalize on your 'quick win': once confidence and competence have flourished do not lose the momentum (or the people), so start that next middleware project straightaway.**

Management conclusion

This analysis ranges far and wide over the delivery and management of middleware. What Mr. Fincham makes quite clear is that business purpose is all important if middleware projects are to be successful.

His 10 'golden nuggets' — perhaps better described as critical ingredients for a middleware recipe — make sense and are practical. In an environment — middleware — where technology can all too easily seem to rule triumphant, his pragmatism is a breath of fresh air. It might not be original — in the sense that Galileo, Kepler or Copernicus were. But being reminded of the obvious by a practitioner is valuable.