Editor: Robert R. Rowntree, M. Eng., MBA, PMP, CISSP, P.Eng.   *Spring 2009*

Also, what **Kmart** did wrong, the ideal team, the best predicator of high performance individuals and getting people to change.

WebSphere SOA Performance Engineering

Tracking down tough performance issues across complex systems and applications is likely not an easy task. For new applications, it seems projects are always 98% complete for a long period of time. Often, the last 2% of the project includes the toughest part of the project that requires a specialized skill set - performance engineering.

Click here for more info

High Performance SOA Part IV – Business Process Re-engineering Recap

In 2009 budgets for BPM (Business Process Management) and re-engineering are **expected to grow** while many other IT activities will stay level according to Information Week Analytics Outlook survey for 2009. And with Service Oriented Architecture (a.k.a., SOA) as the principle framework that is driving almost all vendors' product strategies, understanding its value proposition, risks and gotchas is critical.

SOA has garnered attention because **it is the first technology framework that delivers key ingredients requisite in streamlining complex and often disconnected activities into high-speed business processes.** Business people have reasonably mature financial (a.k.a., income, balance and cash flow statements) and strategy frameworks (e.g., Kaplan's Strategy Maps). However, until recently organizations were missing a high fidelity IT process framework that could effectively link the 10,000's of its activities to the highest levels in the business. SOA can help deliver this capacity while reducing or eliminating the lack of accountability that often occurs in the "**IT Twilight Zone**" – that warped space between business and IT operations. Assuming SOA can deliver these capacities, three major areas of business operation would be well formulated – strategy, finance and information related operations.

Unfortunately, a core requirement of SOA is that most of the people involved are going to have to be **disciplined** and willing to do some **"heavy thinking".** If those conditions are met then you should see linear effort payoff with geometric results.

So the big question - where are you going to get all those smart and disciplined people? In this article, along with SOA are some tips on what to look for in building teams to make SOA happen.

The line up for the SOA series of articles includes:
- **Part I – Business** - **The Business of SOA**
- **Part II – Technical – Built-in WAS SOA – Web Services**
- **Part III – Technical – ESB Advantages and Options**
    WESB - WebSphere Enterprise Service Bus
    WMB - WebSphere Message Broker
- **Part IV – Business – Business Process Engineering – The Recap**

In Your Business, What's Slowing You Down?

Research overwhelmingly has demonstrated that the fastest moving markets where a small edge can mean dramatic revenue gains include the finance, capital markets, distribution and insurance industries. What slows you down is processes that are not well understood that can neither work nor change fast. Compounding this is **expensive activities that are "baked in" with low value activities** that impede the re-allocation to either lower cost IT resources, lower cost labor or both. Think outsourcing low value activities, but only if you can break the negative "baked-in" approach. Currently, one of the best ways to "un-bake" your enterprise for high performance is through SOA. SOA process re-engineering success factors include tech savvy management, engineering versus just coding SOA, increased centralized SOA management and enterprise versus LOB based profitability incentives. The key question to ask is: Will LOB's pay for the services or will access be free? Free access will likely result in over use of your services and associated resources. In addition, there is added cost for accounting related to charging for services. **Bottom line is either** *you move faster* or *your customers will move* to faster competitors.

Complex Project Success – The Conditions for High Performance
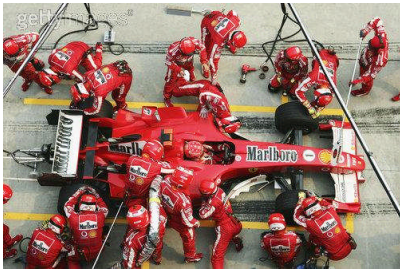
Almost all major enterprise IT projects can be classified as complex with risk profiles similar to some R&D projects. For these types of projects Rizova (MIT Sloan, spring 2006) concluded that 4 elements are critical to complex project success and high degrees of innovation:

1. **Management star(s)** – the management star is well connected inside and outside the organization so that critical resources can be mobilized in a timely fashion.
2. **Technical star(s)** – you would think that technical knowledge in itself was critical and sufficient. Contrary to popular believe, technical star's major contribution over other team members is the **facilitation of communication**. People with deep expertise are able to translate technical language with greater ease. The result, dramatic reductions in misunderstandings in the form of reduced duplication of work effort across the entire project including design, construction and testing of deliverables.
3. **Strong and sustained project sponsorship –** top management is involved and willing to make fast decisions to move the project forward.
4. **Open communication patterns and low degree of formal reporting –** successful projects have fast and fluid networks of communication that reduce delays in progress.

What Conditions Make SOA Desirable?
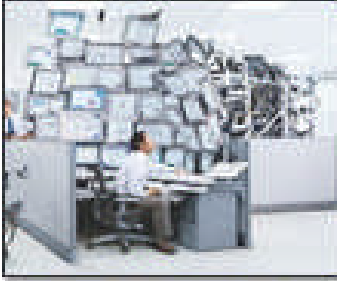
Three key conditions make SOA desirable: One, the high complexity of both communications between and within all the enterprises applications; Two, the need for speed in modifying processes and three, the need for pure processing speed.

1. **You have complex processes that are not well understood resulting in increased costs and lower throughput.** Complexity increases the cost of maintenance and process operations because complexity begs for expertise – expertise required for critical components in the value chain are often scarce and expensive. It also slows processes down because human intervention is required more frequently to referee complex decisions. Typically, larger companies have complex processes that should be better understood. Using SOA can be a good way to drive and sometimes push the organization to understand their processes better.

2. **Processes cannot be modified fast enough to keep up with changes in your business environment.** If the organization does not have to change its processes often then they may not be in need of SOA.

3. **Your customer demands speed or your industry is fast and super competitive.** For example, if FedEx promises next day delivery, then it has to be there the next day. This is an industry characteristic. A bakery has to get the bread out that morning because it is a characteristic of the product – otherwise it does not meet its freshness quality attribute.

Get the Best Team in the Company - Hire for Intelligence and…

In the article "Hiring for Smarts" (November 2005, Harvard Business Review), Menkes indicates that all of the current research shows that both psychological (PI) and emotional intelligence (EI) evaluation are weak predictors of executive success. Many articles and books appear advocating the importance of PI and EI predictors, which may indeed play a role. However, significant evidence has yet to be confirmed by the top tier of the research community. **The only evaluation that shows a strong correlation to a high job performance today is evaluation of intelligence**. This alone is not best practice. It needs to be combined with past behaviour evaluation with the goal of determining whether good intelligence has been well implemented by the person. Jim Collin's, "start with disciplined people" finding in his book, Good to Great supports this practice. And of course, behaviour includes being able to trust.

"It has been shown that the single best predictor of job performance tends to be general intelligence -- that's fairly widely accepted these days," says Colin Cooper, a psychologist at Queen's University in Belfast. "A big meta-analysis done last year looked at the size of the relationship between test scores and a huge range of job-related behaviours. It found that for a huge variety of jobs -- from office work to van driving to management -- the higher your test score, the higher you scored within that particular job."

Source: The Guardian

What's the Value Proposition of SOA (a.k.a., ROI, NPV, CVA)?

The valuable arguments in our view consist of three concepts. One, SOA's inherently intense focus on the business; Two, potential to leverage economies of scale through service reuse; and three, the ability to act as a vehicle to drive out improved process knowledge that will lower cost while delivering faster process execution.

ROI - FOCUSED

1. **A business focused framework first, technology as an enabler**. Past waves of innovation focused substantially on improving technology that often lost its direct traceability to business needs and subsequent ROI. Although SOA does not force a focus on the business, it does push implementers to think services closest to the business domain. Business versus technology domain services differs in that the business services correlate directly with business goals. For example, "Get Checking Account Balance" versus "Encrypt Data" would each be part of business and technology domain services respectively.

2. **Efficiency with scale – leverage the cost across many users.** In tech talk, this is called reuse. Economies of scale are important because it is a basic tenet of economic theory that can help maximize profit. For software products, adding the first user is very costly; in fact, it's the whole cost of the software. Adding subsequent users is essentially free other than the incremental resource utilization costs. Lastly, businesses should build services throughout the enterprise that can be reused within the company and potentially by its business partners.

3. **Untangle expensive from cheap activities to lower process cost dramatically while increasing process speed.** When processes are not well understood it seems that all activities need to be completed by the same resources whether it is structural (e.g., computers, networks, software) or human capital. Low cost activities cannot be moved to cheaper locations or be done by lower cost IT equipment or labor. For example, do all the process activities need to be done on high availability hardware or do all X-Rays need to be reviewed by a local and perhaps expensive radiologist? During the SOA effort, physical and information components may be identified as separable. This is important because high intensity information activities are more location independent and easier to outsource or distribute geographically than high physical presence activities. A key caveat is that human and structural communication bottlenecks are more likely to arise when information activities are dispersed geographically. To untangle activities, process discovery will uncover what needs to be done close in proximity to the business and which activities are independent of locality. SOA can drive the discovery process.

Kmart looked over their shoulder and said, "Hey, if Wal-Mart can do it we can too". Kmart went bankrupt.

Kmart made massive investments in IT. As I mentioned in previous articles that just copying **best practices** and not understanding the details of why it worked for a best of breed companies is likely a disaster about to happen. Turns out while Kmart spent a substantial portion of their money on technology, they had few people that could manage and understand it. Kmart did not spend the money hiring and/or training the people to use it effectively. We encounter many instances like this where we see customers fixated on the technology solving a problem to the detrimental exclusion of the people that will support and use it. In fact, if you have to choose where your company's money is allocated, invest in your people first, technology second.

What are the Risks Associated with SOA- The Truth Comes Out?

Most of the risks in our view are related to people, not technology issues. The negotiation effort and skill amongst the LOB's management will likely determine the success of SOA. The next important predictor of success is the presence of managers and architects that can "broker" and execute the planning and delivery of the projects.
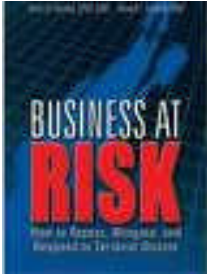
1. **As with most IT projects, SOA initiatives are hard to quantify in terms of ROI with details down to the spreadsheet level**. Risk of seeing low ROI increases with limited tech savvy management, few real Architects/Engineers, a highly decentralized organizational structure and poor LOB inter co-operation dynamics combine to conspire against the successful implementation of SOA.

2. **Who has been successful with SOA - big budgeted, very disciplined tech savvy management**. Companies with success in SOA to date have always had success with technology. Typically, they are big companies with big budgets and businesses that are technology intensive. These companies also have supportive and technology savvy managers. Many are in the financial services industry.

3. **A LOB versus enterprise mindset stifles agreement on cost allocation to build services**. Additional upfront costs include extra management, increased negotiation efforts and the enhanced robustness required in offering services to a wider client base. However, there is a large potential for the organization to recapture that cost over the life of those services. A key problem is most organizations incent managers so they are judged on performance at the LOB level not at the enterprise level. A major component of the SOA change cost will be how organizations measure their managers, much more as a co-operative team than individuals.

4. **Reuse talk is easy but takes intense discipline to execute.** Reuse of services is the cornerstone of SOA but reuse has been "hocked" before as the key value of Object Oriented software – that effort largely failed. Object orientation is a concept that has been promoted since object oriented development became popular in the early 1990's. Few enterprise users outside of software product companies received a good return on the approach, if any. Tools were not inherently available to make it easy to discover components so they were very often duplicated. The good news is the industry has learned from past deficiencies in standards and built into the SOA conceptual framework a discovery mechanism that can find services dynamically and easily.

5. **SOA requires a significant centralized effort.** Without proper management at the enterprise level, SOA will lose traction. If your organizational structure is significantly centralized already then this is a mute point. To gain maximum benefit, management must look across the enterprise to understand what services are needed and how they need to be used. Generalization of those needs must be done so that all users are satisfied. Not an easy task. This is where centers with top architects and people with superb negotiation skills are critical.

6. **Enterprises that rather "code it" than "design it" are prone to SOA failure.** Efficiency of scale through degree of reuse is highly dependent on design – something that few companies invest time or money in these days. Organizations in our experience have not enforced the execution of application design in the past and likely will have a difficult time instituting design in the future. Often, companies say they have a design phase. But with time and hiring budget constraints, few real engineers surface that can make/enforce a systematic approach for "**the hoard**". This is especially true when it comes to abstract concepts. Systems and networks that are more concrete in nature than applications tend to have a significant design effort that is done well. Application development that should have a substantial design phase and use of abstractions often doesn't. The software industry is populated by 80% concrete thinkers and 20% abstract thinkers. Unfortunately, there are not enough

abstract thinkers to go around. *As a result, for all but the best managed organizations, software engineering and SOA maturity are likely to be of too low of rigor to be effective*.

What to "Hammer Out" first for successful SOA

The real question may not be **when but if**. Figuring out who is going to pay to get the services built and then once running, how costs are allocated to keep them running. This means access should be prioritized based on how much LOBs and its users are willing to pay for services.

1. **Services cannot be free - yet many companies are run like communist empires**.
   This may seem obvious that services cannot be free, but this runs counter to how many companies implement initiatives with IT. For example, we have seen Fortune 50 companies that implement a mission critical application such as a core insurance portal or an Internet banking application with little attention to prioritization of resource use. Often, the LOB applications access a data tier that allows any application to access resources and use as much as it wants – no priorities. What's mind boggling is that a sales clerk running a massive database query can cripple customer's throughput for online banking during peak hours. Essentially, the service owner's applications can easily be squeezed out by "customers" from other LOB's wanting services for "free". From a formal business perspective, free services go against the progressive costing accounting approach of **activity based costing.** Additionally, lack of costing neglects organizational **priorities of allocating scarce resources** to the most profitable initiatives. This leads to…

2. **Who is going to pay for the services?** Ed the VP of Business Development wants services - If they are free, even better. Bill the VP of Marketing who offers services cannot just allow all his servers to be saturated by other LOB's requests. Bill wants to charge for them not only to recover real-time costs but project, design and engineering cost associated with offering the service to other LOB's. Which leads to…

3. **If there is a charge to build it, then each LOB will want to get the data they want, their way.** Therefore, the requirements not only include the original interface for the service, but also it must incorporate all the requests of the LOB that plan to use the service. Companies need a mechanism to broadcast services to be offered like an RFP to get cooperation and feedback. There will likely be ongoing modifications required as the service sees more reuse. Once you get over this, the organization will need…

4. **More rigorous and centralized support in the form of portfolio management and architecture development**. For companies that are highly decentralized, especially on a global basis this may be a shock to the system. Then again, it might be a good excuse or driver for a CEO who wants to gain more control over a decentralized organizational structure. Which leads to…

5. **More overhead costs due to the increased centralized effort**. However, if the effort is successful, just like the overhead of creating a software product, this overhead cost can show worthwhile returns.

Which one are you? A Jimmy Carter who struggled as President or a Ronald Reagan who got a lot done.

There are two types of thinking patterns according to Leonard – **divergent** and **convergent** thinking. High capacity for convergent thinking aids in finding solution to analytical problems. Like IQ tests or accounting problems. Divergent thinking best suites situation where there is no clear answer. Like, do we help Africa, invade Iraq or do we enter a new market. Obviously, there is no equation to solve these questions. The less data the more divergent thinking helps.

Convergent thinkers in meetings want to keep things to a schedule. The upside to convergent thinking is in time-pressured situations it works well. Where it doesn't work well is when you need to generate innovative ideas. Innovative ideas come from free flowing and often extended thinking sessions. In these situations, convergent thinkers often want to be kept to schedules and often generate fewer ideas and likely not the best.

On the other hand, **divergent** thinkers value ideas and don't like to keep to schedules – it hinders their thinking. Since less analytical problems don't have clear answers, they get shoved up the organization to the top. This is where divergent thinking pattern helps the most, when there is no clear solution. So, while Jimmy Carter was great at analytical issues, it was not ideal for the role of a President. Conversely, Reagan's divergent pattern of thinking likely horned through days in showbiz and Hollywood, served him well.