

The Justification of Enterprise Web 2.0 Project Expenditures

By Dennis D. McDonald, Ph.D.
Sept. 19, 2006

This is the third in a series of articles related to Enterprise Web 2.0 project cost analysis. The first was [How Much Will Your Enterprise Web 2.0 Project Cost?](#) The second was [System Integration Costs and Enterprise Web 2.0 Projects](#). The author invites comments by email to ddmcd@yahoo.com. His website is located here: <http://www.ddmcd.com>.

Introduction

Given the [difficulty of doing Return on Investment](#) (ROI) analysis for IT projects, how do you justify an Enterprise Web 2.0 project?

In a comment he left on my [How Much Will Your Enterprise Web 2.0 Project Cost?](#) post, [Vinnie Mirchandani](#) suggested that one place to start would be to look at the criteria used in the past for evaluating large IT investments.

Reviewing his list "Ten Ways to Justify Acquiring Packaged Applications," based on work he did for Gartner in 1997, shows how much things have changed and how much they have stayed the same. Here is Mirchandani's list:

1. *Savings from the technology automating what was previously manual*
2. *Savings from data rationalization - e.g. through Master Data management facilitated by the technology*
3. *Savings from business process changes the technology facilitates*
4. *Savings from structural changes (e.g. moving to shared services enabled by the technology)*
5. *Savings from optimized information (back then constraint based decision tools were starting to allow for smaller physical asset bases)*
6. *Savings from not having to invest in Y2K fixes*
7. *Savings from disbanding other legacy systems (e.g. through data center consolidation. Key word is disbanding, not keeping it around)*
8. *Revenue increments clearly attributable to the new technology (rare back then - and even today)*
9. *Infrastructure investments - it was my cop out for Investments that "have to be made". My suggestion was not to burden a specific project with all those costs but to hold them in a corporate account and amortize them over other projects which also would leverage that infrastructure.*

10. *Intangible or soft justifiers - no monetary calibration*

First I'll mention a potential strategic difference, then I'll discuss each of these individually.

Packaged Applications

Consider the title of Mirchindani list, "Ten Ways to Justify Acquiring Packaged Applications." Are we still talking about "packaged applications" the way we were in 1997? No. Delivery mechanisms, business models, and technical architectures have changed.

Sophisticated applications are now available as "services" via remote hosting. Entire business processes, including both staff and technology, can now be outsourced. Open source alternatives are also available for many applications where technical skills are available for customization, maintenance and support.

Applications that emphasize collaboration, information sharing, and relationship management are no different. If you want to establish RSS enabled blogs as part of your corporate HR and project management intranet infrastructure, for example, you can opt for a variety of productized solutions, some self-hosted within the firewall, others remotely hosted. Or you can base your applications on open source applications.

In other words, the options for obtaining the benefits of packaged applications are now available via a variety of channels. That is a good thing. It does, though, make the justification exercise more complex, and this includes the comparison of costs among alternatives that are potentially very different

Justification 1: Savings from Automation of Manual Processes

Past paradigm shifts in computing (to mainframe, to client server, to corporate Intranet) were often justified through cost savings, quality, and consistency improvements delivered through automation of previously manual processes. Classic cases were reductions in the paperwork needed to administer large scale clerical-intensive processes such as insurance claims and bill processing, and improvements in call routing within call centers to optimize call handling and reduce wait times.

Do Web 2.0 collaboration and social networking based applications offer similar automation advantages? I'm not so sure. Partly this is a question of what we mean by "manual" and partly what we mean by "Web 2.0."

Many of the processes enabled by Web 2.0 applications appear to be of the "knowledge worker" variety as opposed to the "repetitive clerical" or "manual labor" variety. I will be the first to grant the contribution that "knowledge workers" make to the GNP, but much

of the work done by "knowledge workers" has never been classified as "manual." I think this type of a justification will need more work.

If we also include in the "Web 2.0" category the shift in technical architectures to include Rich Internet Applications ([RIAs](#)) as well as a move towards the flexibility offered by Service Oriented Architecture (SOA) development approaches, we might hear claims concerning costs savings through the reduction of application development times. These costs savings extend from programmer time through the reduction in the time via Agile project management techniques to get users up and running on newer, more productive applications.

In summary, using "automation of manual processes" as a justification may require a rethinking of what we mean by "manual processes" as well as a rethinking of how we address changes in technical architecture that enable more rapid delivery of benefits.

Justification 2: Savings from Data Rationalization

I believe that partly what Mirchinandi is referring to here is "data standardization." It's hard to argue against that. Is there anything about Enterprise Web 2.0 based solutions that would argue against the need to perform some kind of data standardization?

I don't think so. One of the Holy Grails of Web 2.0 for example is the "mashup" that seamlessly combines on the desktop data from two or more sources. For that to work, though, the data has to be compatible or easily transformed. If I want to be able to show a home customer a Google Map with an overlay of where he or she can buy replacement parts for the gas range that just failed, for example, I need to be able to coordinate Google's geographic information with my customer records and parts inventory system. If my company has three different methods for geocoding customer locations, do I create a middleware application (that has to be maintained) that handles transformation and standardization in the background while providing an easy to customize interface to the programmer? Or do I "bite the bullet" and consolidate my three systems into one so that, not only is a common interface available to the programmer and the user but also a common set of database maintenance processes are supported?

Justification 3: Savings from business process changes the technology facilitates

Savings in business process changes are a Good Thing -- as long as the goals and operations supported by the business process change are valuable to the enterprise. Here I don't see much difference in the justification between 1997 or 2007.

Nevertheless, this justification does require a consideration of what we project managers call scoping, i.e., deciding what is to be in and out of scope of the analysis we are using to justify the change. Web 2.0 applications adapted for enterprise use typically involve

changes or additions both to technology as well as changes to business processes. Depending on the business processes that are affected, these changes could be profound and potentially categorized into two areas:

1. Addition of new business processes that did not exist before
2. Changes to existing business processes

If the business process change is a “Type 1” business process – one that did not exist before – who is going to manage and perform the process? Additional people using hours that were not previously paid for? Or people whose time (and cost) will need to be shifted from something else they are doing? If the latter is the case, will we need to take into account the cost of losing their time to old processes?

In practical terms, this is sometimes what happens when blogs are added to the mix of customer relations media and methods. Blogs don't write themselves. To be taken seriously they need to be regularly updated and monitored. That takes time and money especially if a goal for rapidly responding to incoming comments is established. Will new people need to be hired; will the effort need to be outsourced? Or can staff be shifted from an existing effort to the new blog – at what cost?

One might argue that, in the long term, addition of a blog to the existing mix of customer relationship management processes is not a change in strategy but rather a change in tactics. His argues for a strategic view to be taken to the costs o adding new services, but this type of consideration is no different now than it was in 1997. Relating costs to strategic objectives was just as valid then as it is now.

But back to the justification as stated – savings from business process changes. If the technology supports a change or replacement in a business process, this is a reasonable area of analysis. If, on the other hand, the technology requires an addition to costs due to its support of a process that did not previously exist, we may need to look elsewhere for cost justifications.

Justification 4: Savings from structural changes (e.g. moving to shared services enabled by the technology)

It's hard to argue with this justification, though it's interesting that the concept of "shared service" may have changed quite a bit since 1997 when "middleware" was still a popular buzzword and "SOA" was not as pre-eminent an architectural concept as it is today. Architecturally, re-usability is a Good Thing whether we are talking about hardware, data, application components, or functionality.

Re-use of services across delivery channels can ultimately impact marginal costs and marginal pricing. Re-use implies cost-savings when considering, for example, how much it costs to maintain an application centrally via a service supplier as opposed to locally at multiple locations behind individual firewalls. That's one reason why custom coding of

key corporate systems has to a great extent gone by the wayside since so much powerful functionality is available off the shelf and is now available as a remotely supported services.

The concept of "sharing" and "reusability" can also be extended to include impacts on business processes and to the sharing of knowledge, skills, and expertise. Assuming the key user-oriented element of Web 2.0 applications include collaboration and sharing, the ability to locate and consult with an expert outside an employee's immediate workgroup, facilitated by a searchable expertise location database that allows for location of corporate expertise, could be an example of a shared service where the sharing is not of functionality but of knowledge and expertise.

Is this a structural change? In one sense it is a structural change that potentially impacts the way the enterprise is organized. If in an expertise location or expertise management system it is found that requests for expertise are constantly flowing across one particular organizational boundary into another, is there reason to think that a re-organization is called for?

Justification 5: Savings from optimized information (Mirchandani says: "back then constraint based decision tools were starting to allow for smaller physical asset bases")

I call this the "getting the most bang for the buck" justification. Just as you want to optimize the number of physical assets in relation to cost and utility (e.g., locate warehouses to minimize transportation costs, locate trained customer service staff to optimize training, CRM, and telecom infrastructure costs, etc.) a similar case can be made for optimizing the mix of social and collaborative technologies you implement. If you value the collaborative and interactive effects a wiki has, you may want to minimize the number and type of platforms you support within the organization, just as you might want to minimize the number of different tagging methodologies you support.

An interesting question arises when we apply the concept of "asset management" to the number and type of subject matter experts we support within the organization. Just as we want to optimize the number and location of physical and software assets so that we do not "overspend" on infrastructure, does the ability to share expertise across organizational and time zone boundaries impact the way that knowledge- or expertise-intensive activities are organized?

Justification 6: Savings from not having to invest in Y2K fixes

We don't have a Y2K situation to contend with now. We do have externally imposed requirements that are regulatory in nature that might impact – and be impacted by – addition of Web 2.0 technologies to the enterprise.

Consider Sarbanes-Oxley regulations and its spawning of systems, business process changes, and consulting services designed to formalize review and approval of certain types of corporate information. Sensitivity to Sarbanes-Oxley reporting requirements might explain some of the reluctance voiced during a [series of interviews I conducted earlier this year concerning corporate adoption of Web 2.0 technologies](#).

It makes sense to think there are certain types of corporate information that should not be subjected to the type of sharing and collaboration processes that some Web 2.0 technologies invite. Security, accuracy, and validity of certain types of data is a clear requirement for stockholder, management, and SEC reporting.

At the same time, can Web 2.0 systems actually lead to savings if, say, business processes and systems are being modified to conform to Sarbanes-Oxley requirements or, say, the kinds of reporting required by regulatory agencies related to the pricing of regulated goods and services such as electricity or natural gas? Would the log files of a shared bookmarking or file tagging system be sufficient to document that certain types of oversight or review processes have taken place? Would a wiki that integrates workflow or process management automation techniques be useful for supporting and documenting a financial report's review?

I don't know the answer to that question but it might be worth examining in more detail, starting with considerations of being able to improve the answer to questions such as, "Who knew what, and when did they know it?"

Justification 7: Savings from disbanding other legacy systems (Mirchandani says: "e.g. through data center consolidation. Key word is disbanding, not keeping it around")

Put another way, this justification raises the question of whether a "web 2.0 application" could actually replace a legacy application, where "replace" means "the business processes supported by legacy system X are supported by web 2.0 system Y."

A lot of the answer to this justification question is based on what we mean by the Web 2.0 application being considered. Two things are clear, though, based on my own experience with projects that either planned or implemented legacy system replacements:

1. Conversion costs are killers.
2. Business processes inevitable must change when a legacy system is changed (which leads us back to Point Number One)

There's nothing inherently "web 2.0" about these types of considerations. Let us say, for example, that the legacy system being considered for retirement or replacement is client server based content management system that emphasizes document retrieval based on manually created indexing. Right away we have a comparison issues since the essence of Web 2.0 based content management functionality is sharing and collaboration, functions which may not be supported at all by the legacy system.

Justification 8. Revenue increments clearly attributable to the new technology (Mirchandani says: "rare back then - and even today")

Ah, the Holy Grail of Return On Investment analysis: revenue increments! I well remember the times I've been able to populate spreadsheet cells assessing potential IT initiatives not with COST SAVINGS but with INCREASED REVENUE.

Such justifications typically relate to the ability to develop, sell, or service more products or services for a given outlay of resources. Since process streamlining and operational efficiencies can be linked to revenue increments, cost savings can have impacts that go beyond cost avoidance.

The kicker in this analysis is that many potential IT investments are more infrastructure related than product- or customer-specific. When this happens it is difficult to link infrastructure investments to specific revenue programs. Justifications must come from elsewhere and are often related to cost savings.

Many Web 2.0 types of technologies are like this; they tend to be infrastructure oriented and difficult to link to a specific revenue program. Do you agree with this last statement? If not, please let me know!

Consider the technologies potentially related to [expertise management systems](#) (the link is to a .jpg page image from a [slide presentation](#) on expertise management systems):

1. Analysis of email & communication patterns
2. Tagging
3. Subscription (e.g., RSS)
4. Network based search
5. User feedback
6. Web based communication tools

Taken individually these technologies have multiple applications and can support many different processes, not just those processes linked directly to developing, selling, or servicing more products.

As with most analyses of this type, though, the devil is indeed in the details of how the technology is applied. After all, it is not the technologies that support revenue increments or cost savings, it's the business processes the technologies enable and support. If these business processes are highly generic, distributed across many different business areas, or, like email based communication, just fundamental to the operation of the organization, it makes sense to treat them as infrastructure expenses, in which case explicit revenue increment justification is not appropriate.

Justification 9: Infrastructure investments (Mirchandani says: "it was my cop out for Investments that 'have to be made'. My suggestion was not to burden a specific project with all those costs but to hold them in a corporate account and amortize them over other projects which also would leverage that infrastructure")

See the discussion of Justification 8 above.

Justification 10. Intangible or soft justifiers - no monetary calibration

To the modifiers "intangible or soft" I would add "or difficult to quantify." Let's consider, for example, a system that is designed to help people locate experts within a company and is based at least partially on applications of social networking technology as well as dedicated search methods that analyze email traffic to identify and tag "experts" and "expertise" in a searchable database. (One product like this is Microsoft's [Knowledge Network](#) software, currently undergoing beta testing.)

While it is possible to put numbers around a process such as "find expert," being able to relate this to overall costs and benefits will require a bit of analysis. While such a cost analysis not impossible, what is really needed is a quantitative picture of how frequently experts are sought out, how often they are found/not found, and what are the cost and other consequences of their being found – or not found.

That last item is tricky. Even though an expertise management system might shorten the time it takes to locate an acknowledged expert, the value of the system is ultimately determined by the benefits derived from the help provided by the expert. If that is the case, then the costs of using the system should also be considered, not just technology specific costs (including maintenance and support) but also the cost to prepare and

maintain the database, the costs of communicating with the expert – and the cost associated with providing a response.

What is the relationship of such costs to “justification,” which is really what our focus is here? At minimum, we need to be able to picture costs with and without the system. Admittedly, the operation of an expertise management system might alter communication and innovation patterns within a company enough to cause a rethinking of management techniques and organizational structure. Such concerns, while not impossible to model from a cost standpoint, are perhaps more realistically addressed in terms of intangibles such as improved communications, higher quality, and more opportunities for participation and innovation.

Conclusions

Much of Mirchindani's list is still relevant. Partly this is because many of the technologies we frequently associate with “Web 2.0” (e.g., see list in Justification 8) are identifiable as applications, and partly this is because it has long been the practice to define IT related justifications in terms related to costs and benefits.

There are, however, some issues that need to be addressed when approaching the justification of Web 2.0 expenditures in the enterprise:

1. Much of the focus of Web 2.0 applications is on “knowledge work.” Given the mental nature of knowledge work, a reasonable extension to an analysis focusing on cost would be a farther disaggregation of costs by the nature of the problems or issues being addressed. An example of this would be to understand the nature of the problems people use the system to address. For example, are the knowledge issues simple or fact based, or do they require involvement of multiple contributors and multiple iterations? If the latter, is the system also intended to support the iteration and communication process, which suggests that some form of collaboration functionality should also be included in the system (and in the associated analysis)?
2. Many Web 2.0 applications might be more appropriately thought of – and justified – as “shared infrastructure.” I think about this whenever I test out applications such as Cogenz, RawSugar, and Connectbeam. Applications such as these are, I find, extremely useful for managing, organizing, and communicating information, interests, and expertise. These are functions I regularly engage in in my consulting, writing, and research. I consider them to be basic tools, yet their value to me organizationally will be to a great extent determined by their use by others with whom I need to share and communicate. Having them available might be more appropriately thought of as “the cost of doing business,” yet I fear that simply assuming their costs are not allocatable to individual business processes might be a mistake, both politically and financially.

3. It's a mistake not to get IT involved. Some have suggested the attractiveness of end-running IT and going outside for externally-vended and supported solutions. This is an attractive option to those who feel IT is overwhelmed and slow to respond, but viewed from the vantage point of efficiency, security, and the need to guarantee integration potential, I think failing to involve IT right from the initial justification stage would lead to significant problems down the road, especially as it becomes appropriate to add or integrate collaborative or social network oriented features to other "non-web-2.0" applications.

Copyright (c) 2006 by Dennis D. McDonald.