Capturing Business Benefits from Process Improvement: Four Fallacies and What to Do About Them

Paul L. Bannerman
NICTA
Australian Technology Park,
Locked Bag 9013, Alexandria, NSW 1435, Australia
+61 2 8374 5465
paul.bannerman@nicta.com.au

ABSTRACT
A basic assumption underlying any process improvement initiative is that it will have a positive impact on the organization. Therefore, it can become easy to assume that process change will in fact deliver benefits to business. This paper takes a practice-based look at some fundamental assumptions about process improvement that can be as fallacious as they can be true. The argument is supported by case scenarios. Also, some ways are suggested to manage around these fallacies to achieve net benefits rather than no impact or negative impacts on the business. Four basic fallacies are considered: that process improvement leads to business improvement; that process change equates to process improvement; that software processes are non-lethal; and the vision of the enterprise as an automated process. The paper concludes that the future success of process improvement as a management strategy is dependent upon the capability of organizations to capture material gains.

Categories and Subject Descriptors

General Terms
Management, Measurement, Performance, Human Factors.

Keywords
Software process improvement, process change, business benefits.

1. INTRODUCTION
An implicit assumption of process improvement is that it benefits business. Why do it otherwise? Empirical studies suggest that this can, indeed, be the case [8] [12] [20] [21] [26]. However, this can make it easier for managers to assume that business improvements will flow from process change initiatives, without adequately considering and putting in place mechanisms and practices to ensure that the impact is not actually negative. The empirical evidence and experience also suggest that no improvement or even a negative impact are also possible outcomes [12].

Process improvement is emerging as a viable alternative to the more traditional dependence on technology as the basis on which business value is created and competitive advantage achieved [22]. Efficiency in the operational execution of business processes and competency in the adaptation of processes for the purposes of improvement and to meet new opportunities is emerging as a powerful option for generating business benefits and creating competitive advantage. Processes are central to most things that organizations do. Indeed, early writers on process-centric business improvement argue that process is the business, and that products and services are mere byproducts of processes [25].

However, empirical evidence suggests that the realization of benefits from process improvement is neither automatic nor guaranteed. This is supported by data on the impact of process improvement on business improvement that reports high variance and uncertain returns [12] [24] [21]. For example, a study of 138 CMM users found that, while positive results were achieved, 26% agreed that “nothing much has changed” after an appraisal and 49% were disillusioned with the lack of results [12]. More recent SEI data confirms that, for organizations engaging in process improvement that report a performance improvement, the variance is high [24]. This suggests that when an organization decides to invest in process improvement, it cannot be certain what the result will be.

Therefore, it is critical to understand the enablers and inhibitors of process improvement initiatives that might influence the resulting business benefits. Empirical studies have found a range of factors that suggest these enablers/inhibitors fall within management’s direct control or influence [12] [20] [21] [26]. For example, key enablers have been found to include investing in adequate resources, making goals very clear, and actively monitoring the goals [12]. Others are management commitment and support, staff involvement, providing enhanced understanding, tailoring improvement initiatives, managing the improvement project, change agents and opinion leaders, stabilizing changed processes, encouraging communication and collaboration, and setting relevant and realistic collaboration [26]; reviews, standards and procedures, training and mentoring, and experienced staff [20]; and executive support, experienced staff, internal process ownership, metrics, procedures, reviews and training [21].

Conversely, factors found to block benefits include characteristics of organizational culture such as turf guarding and organizational politics [12], organizational learning [7], and lack of vision, short-
term focus, lack of a sense of urgency, organizational politics, hunting certification, staff resistance, lack of change agent buy-in, lack of effective communication, neglecting to anchor change in corporate culture, lack of project management, and low quality of process professionals [17]. These factors typically fall within broader themes of process, people, skills, management and leadership [20] which are within management’s direct control or influence.

In a review of the shape of the software process improvement field, Hansen et al. [8] argue for a “more reflective, theoretically oriented” perspective on SPI research. This paper contributes to this call by taking a practice-based look at some fundamental fallacies in assumptions about the business impacts of process change through the lens of a socio-technical perspective of the organization. It also considers how managers might avoid the risks, inherent in these false assumptions, of not realizing the potential business benefits of process change and innovations.

2. PROCESS IMPROVEMENT FALLACIES

In embarking on a process improvement initiative, managers can greatly improve the likelihood of a positive business impact if the following four false assumptions are avoided.

Fallacy 1: Process improvement leads to business improvement

It is both easy and common to assume that, by definition, process improvement leads to some form of tangible or intangible business benefit. After all, it is only logical that improving how a business operates must improve the organization. However, there is no research evidence to suggest that process improvement necessarily leads to business improvement. Research identifies no determinism or direct causal link in the relationship between IT-related process improvement and business improvement (see, for example, [27]). Indeed, if anything, the evidence points to a quite complex, even paradoxical, relationship between the two [13]. This applies to all forms of process improvement including business process improvement and software process improvement. Instead, the literature suggests that business improvement only accrues from process improvement if business benefits are explicitly sought and captured by the organization [5] [14]. Realizing benefits from innovations such as process improvement requires that the improvement is made in conjunction with corresponding changes to complementary assets with which there exists a dependency of some kind [28]. Changing only some of the complementarities, however, may not achieve the benefits that could be expected from a coordinated move, and may even generate negative returns [15].

The business literature suggests that three complementary integrating mechanisms are critical to realizing benefits from process improvement: aligning process changes with business goals and objectives; incorporating process management in governance mechanisms, and; establishing ownership of key processes. Each is briefly considered following.

Alignment

Just as the software engineering discipline denies the notion of a ‘silver bullet’ solution to the challenges of software development, organization theory denies the existence of a preeminent driver of business performance improvement. This can be explained, for example, by viewing the organization as an integrated socio-technical system, as illustrated in Figure 1. This view conceptualizes the organization as a configuration of five interrelated complementary components: strategy, structure, processes, technology and people (roles and skills). The notion is that each component can be determined independently, but the performance of the whole organization is optimized only if each component integrates consistently with (that is, complements) the others in a coordinated move. Optimal performance also requires this internal configuration to be consistent with the organization’s external environment. That is, the organization performs at its best when it is internally and externally aligned. Understanding the dynamics implicit in the model helps understand the uncertainty surrounding the outcome of change and improvement initiatives.

Why is alignment critical? When an organization conforms to a consistent design logic, it can operate more efficiently, without the need for artificial mechanisms to overcome misalignments in the configuration. Artificial coordination mechanisms can be very costly, wasting effort and resources that should be more productively employed serving customers and other stakeholders. If the components fit poorly because they do not conform to a consistent logic, the organization is likely to under-perform due to the overheads associated with these artificial devices.

For example, when a $2 billion per annum turnover organization decided to outsource its data processing operations, it overlooked the specialized needs of its main business division, whose systems accounted for over fifty percent of the data center’s activities. The division’s systems were highly dynamic, often changing on a daily basis, while the other systems were stable legacy systems that were easy to operate. The central and divisional IT units had a history of acrimony, stemming from the divisional IT unit’s autonomy. No specific provisions were built into the outsourcing agreement for technical resources to manage the constant flow of changes (usually daily) to the division’s systems. To maintain acceptable service levels, the divisional IT unit had to separately negotiate with the data center service provider, and fund, a customer relationship manager and additional technical resources, as well as establish a multi-representative operations management review committee and various technical task groups to handle their needs. The direct cost was over half a million dollars a year, ignoring management time to maintain the arrangement and
relationship independently of the central IT group, who managed the main contract. A review of the main agreement after three years found that the cost of the service had exceeded expectations, quite apart from the additional costs incurred by the division.

Another reason why alignment is critical is that it can be a significant barrier or facilitator of effective innovation and change. In terms of impact, alignment can work for process change, producing net benefits, or against it. Alignment and misalignment act as powerful reinforcing mechanisms of the established organizational configuration (the status quo), for better or for worse. Changing a tightly integrated organizational design requires the cooperation and participation of all impacted components. As the experiences of IT-led change projects attest, a major change initiated in any one of the five organizational components will have limited impact and success unless it is concurrently matched with corresponding complementary changes in the other related components.

A well-known example is the all-too-common tendency for IT-based projects to ignore the human side of technology-based innovation. Resistance to change by the people the project is ultimately dependent upon to deliver the benefits of the change has blocked or significantly diminished value from many projects. By contrast, people can be major enablers of process innovation if managed by techniques such as stakeholder analysis, ‘rules of the game’ analysis, change history analysis and current initiative analysis to identify, plan around, and manage potential barriers. For example, when a completely new newsroom system was to be implemented into a major radio and television network, resistance was overcome by ‘converting’ the most respected journalist to the new way of preparing and presenting news programs. He became an ‘evangelist’ for the new system, encouraging others to embrace the change, thereby dissolving the resistance.

An attempt to introduce a major new IT-based application (via the technology component) will immediately encounter the organization’s cultural status quo (the vertical oval in Figure 1), a recognized inhibitor of successful process change. Such a change will only be effective if it is matched with corresponding changes in the organization’s skill sets and responsibility assignments (the people component), the business processes surrounding the application and related work structures, to enable the new system to be adopted and used as intended. Only then will this revised configuration of components represent a new organizational capability that can provide a platform for new strategy options.

Similarly, a process change will only be effective (that is, generate a business improvement), if it is matched by corresponding changes in complementary components that will enable a new state of alignment to be reached. For example, teaching new staff how to use a new enterprise system will have minimal effect if they are not taught how to use it to perform their redefined roles. If the change program ignores or fails to ensure that the new process is implemented with corresponding changes to impacted structures, technologies and people, or is inconsistent with the organization’s strategic objectives, then the desired process improvement may be little more than a localized process change that does not deliver any particular business improvement. Indeed, according to the logic of Figure 1, the change effect may even be negative by creating further misalignment.

Governance

As a second integrating mechanism, corporate governance is an interrelated set of formal structures, arrangements and practices that aim to ensure that business activities are aligned with business goals and objectives; value is maximized from investments; resources are used effectively; risks are managed responsibly; and the organization complies with regulations and other external stakeholder requirements. That is, that business benefits are identified and captured from the activities of the firm. Within this framework, IT and process governance can be effective ways for the organization to coordinate software-related activities and ensure that business and enabling processes contribute positively to the firm.

Empirical data suggests this practice may not yet be widespread. A recent industry survey of 274 people with an interest in process management found that 36% of respondents’ companies had no formal process group or center of excellence. For those that did, 18% were located at the divisional level, 16% in the IT function, and only 13% at the executive level (the remainder was within quality control, finance or some other function) [11].

Another industry survey found that organizations with a dedicated process team reported twice the ROI of companies without a dedicated team [18]. More significantly, organizations with a process center of excellence were six times more likely to report a successful outcome from an enterprise-wide process initiative and reported five times greater ROI than companies without a centre of excellence or those with just a dedicated process team.

Governance can enable business improvement from process-related activities in several practical ways:

1. Governance provides formal linking mechanisms between key enterprise responsibility domains that can facilitate integration and alignment between the components of the model in Figure 1. These domains may include the Board, the strategic management framework, business operations, and various technical bodies such as the architecture committee, data management committee, IT management committee, change control board, project board, process councils and centers of excellence. Through integrating mechanisms such as overlapping membership, multi-directional reporting, assigned decision rights and accountabilities, and monitoring and control, this integrated framework overlays the model in Figure 1, facilitating co-alignment of the organizational components and activities in a manner that is both consistent with the evolutionary path of the organization and delivers measurable improvements in business performance.

2. The governance framework provides a practical mechanism through which corporate and business unit executives can have visibility of, and direct involvement in, the business processes and activities of the organization. This has a two-way integrating effect, increasing the likelihood of business benefits from the organization’s activities and processes. First, it informs these activities and processes of the strategic and operational goals and objectives of the organization as they form and change over time, through the direct participation of the executives. Second, it provides feedback to the strategy makers and planners on progress towards the defined goals and on bottom-up developments and opportunities that may further shape those objectives. Furthermore, at the business unit level, the participation of business executives brings
knowledge of industry trends and competitive behaviors that supports integration with the external environment.

3. One of the key functions of governance is to assign decision rights, responsibilities and accountabilities over critical corporate assets, including its processes. This determines who has what authorities to act on behalf of the firm. This mechanism enables organizational processes such as low level technical processes (e.g., software engineering) and higher level business processes (e.g., managing customer relationships) to be recognized as critical assets worthy of explicit proprietorship and ensuring that appropriate responsibilities are put in place. Without this mechanism, the realization of business benefits from processes and process improvements may be more a matter of good fortune than good management or design.

4. Another function of governance is to provide mechanisms for ensuring compliance with company and accounting standards and regulations, as well as fulfill the requirements of external stakeholders such as shareholders, the stock market, and members of the public. This function facilitates integration of the organization with its formal external environment.

5. Finally, governance facilitates the adoption and monitoring of best practice standards and behaviors. Effective governance benchmarks operational performance against best of class standards as well as promotes internal innovation to build and develop competitive advantage. Building an organizational capability in process management can be a significant source of business value for the firm.

Consider further the case of the organization that outsourced its data centers (discussed under Alignment, above). Following the review after the first three years, major changes were enacted to realign the organization’s IT services. These included integrating all IT units into one corporate IT function and renegotiating the outsourcing contract to incorporate fee-for-service and resource-based costing components where they were most applicable. Also, the operations management review committee, which met weekly, became a pivotal mechanism for resolving operational issues and smoothing relationships with the service provider, as well as integrating with other internal governance bodies such as the executive management committee, the IT steering committee, and the ongoing stream of system change projects. The membership of the committee included the data centre manager, customer relationship manager, contract manager, system manager, technical support manager, quality control manager, help desk manager and a user representative. The turnaround was such that one year after the reorganization, a consultants’ review of the relationship with the service provider found it to be the best they had seen in an outsourcing arrangement.

Ownership

Extending the discussion on governance as an integrating mechanism to enable process change to be translated into business value, proprietorship, in the sense of ownership and stewardship of a valuable resource, is a fundamental device for capturing derived value. Ownership was found to be a critical success factor in the literature reviewed earlier. Process owners have a detailed understanding of, and stake in, the processes they own. Within the governance structure, they have the authority and responsibility to determine and model the process characteristics, and monitor and measure process performance over time (as deemed appropriate by the organization for each process).

A key success factor in the case of the operations management review committee discussed above was that each member manager actively claimed ownership for their separate domains of responsibility within the committee’s charter of activities and, together, they owned and fostered the relationship with the service provider.

In addition to formal governance-based process ownership, recent research suggests that less formal responsibility arrangements can also be highly effective in ensuring that business benefits are captured from process improvements [1].

These include arrangements in software projects, for example, where the business/process owner works directly with the project manager within the project to jointly devise solutions to any technical or business issues that arise in delivering the software-based innovation. As issues arise relating to the software engineering processes employed, for example, or to embedding the system in appropriate business processes, this team works together to determine any changes that might be necessary to achieve the project’s objectives. These are then advised (and ratified if necessary) through the formal steering committee and any other relevant governance bodies. This approach enables the project to operate efficiently ‘on the ground’, without unwanted delays or distractions, while maintaining essential ‘line’ reporting and executive access for matters that need to be escalated.

In larger projects, other specialist owners might be added to the project management team to take care of other areas that are critical to the success of the project such as change management, infrastructure, or third party integration.

Fallacy 2: Process change equates to process improvement

This fallacy refers to the phenomenon sometimes referred to as premature optimization or process sub-optimization through sub-process optimization. The key issue is the perspective or level at which a process change is considered to be an improvement. The software development process has, historically, been a rich ground for process change-based sub-optimization.

Take the common case of testing. When should software testing be done and how much is enough? Traditional development processes require the programmer to unit test code before it is passed on for integration. Depending on the nature of the system, this can be a very complex and time-consuming exercise. Code execution may have many data and interface dependencies with other sub-systems and systems. The effort required to create an elaborate and comprehensive single-use test harnesses to validate an individual program may significantly impact the performance of the programmer as a coder (especially if his or her performance is measured in lines of code written and ‘tested’). Also, the test harness may be so artificial that the validation value of the process, at this stage and if rigorously performed, may be marginal at best.

From the perspective of developers, therefore, the rationalization might go something like this. Our performance would be improved if we spent less time on artificial testing processes and more on coding. After all, the downstream integration and system test world have a realistic test environment established so it is easier for them to pick up any bugs in our code that we miss. If bugs are
found, the code unit can be quickly recycled back through us to fix the problem. This change in process is likely to result in a much more efficient development life cycle overall. However, we won’t say anything about this change in case the testers object and insist we perform even more testing than we’ve done in the past. We have good testers so there should be no problem with this.

Therefore, from the perspective of coders, this sub-process change is a process improvement.

Now consider the issue from the perspective of the testers. In a typical plan-based large system development environment, system integration, integration testing, regression testing (if needed) and final system testing processes are usually performed by people who are independent of the original programmers. This ensures objectivity in the integration and testing processes. The role of the testers is to ensure that all components of the system fit and operate together and interact with other internal and external systems as intended. Much of the focus here is on the operational and functional integrity of the system, in preparation for client or user acceptance testing. Testers may not have access to the design specifications for individual units of code because many of them are non-technical. Rather, they work from test scripts that were prepared from the original functional specifications concurrently with the technical design processes. Most of these scripts are written from the perspective of a business or user specialist, not a system designer.

A frequent problem encountered in such projects is pressure on the time available for system testing before delivery is due, caused by upstream project delays, under-estimation of effort, or scope creep. If this happens, the testers sometimes ‘streamline’ testing processes by focusing on the client/user view of the system, to ensure that it passes user acceptance. In changing the testing process, they may rationalize, for example, that this is not ideal but it is better than being late or failing the acceptance test (both of which may attract high penalties under the contract). Also, since the developers have already thoroughly tested the code at a detailed technical level, any issues that arise are likely to be in the integration or user interfaces. However, we won’t make this explicitly known. After all, everybody knows we are being pressured into testing on a ‘best endeavors’ basis to deliver the system on time, so let’s not make an issue out of it.

A possible outcome from this scenario is that each quarter after the system is implemented, the client/business unit achieves sales targets more easily than expected, so they sail through the year on a wave of improved performance with the exertion of minimal effort. Early in the next year, however, an industry survey reveals that the business has lost market share to its competitors, despite apparently achieving its sales growth targets. This is eventually confirmed by the company’s own year-end financial results. Suspicion falls on the new system but no one is sure if anything is wrong. Six months later, a new programmer finds a bug in the code unit discussed above whilst applying changes for a routine function upgrade. It appears that a data item was misinterpreted from the specification, inflating the value of a basic sales performance metric. The program code ran perfectly, but it was generating incorrect data.

What is the message from this scenario? A process change is unlikely to translate into a process (and, therefore, business) improvement if optimization occurs only at the immediate process level. For net benefits, sub-processes must be optimized against sub-process goals and objectives as well as higher level process goals and objectives. In practice, this means that a sub-process change should not be made by the sub-process owner acting alone. Consultation with, and the concurrence of, the overall process owner are also necessary to ensure optimization at the level that can significantly impact the business. This also suggests that for intentional process improvements, measurement of the performance impact must extend up to the highest process level for an accurate reading, to avoid falling victim to myopic optimizations.

In terms of Figure 1, the scenario illustrates at a micro level how a configurational change made without consideration of other parts of the organizational ‘system’ may not work for the benefit of the whole. A localized move, albeit well-intentioned, may be out of alignment with other parts of the configuration, leading to a compromised or even negative performance outcome.

**Fallacy 3: Software processes are non-lethal**

Extending the previous case, in the situation where a negative organizational impact can arise from a well-intended process change, it is possible for seemingly ‘innocent’ process changes to have more dire consequences. Most readers will be familiar with cases in defense, aerospace and commercial airline industries, for example, where software and software processes have been key factors in systems failures, some leading to loss of life. Typically, in such cases, a series of interlinked process failures occur which, by themselves are innocuous but, in sequence, lead to disastrous and even lethal consequences. This phenomenon can also occur in otherwise mundane business system processes.

Take the case of a tragic road traffic accident.

A maintenance release was planned for a Department of Motor Vehicles’ (DMV’s) vehicle registration and licensing system to fix various long-standing defects. The DMV processed 20 million registration and licensing transactions a year through 130 customer service offices and an online website. It had an aggressive system development program with 4 to 6 major system enhancement releases per year and several smaller incremental releases. Each release was managed as a separate project. The maintenance release was fitted into this tight schedule to clear a number of accumulated software bugs and operational problems in the system to consolidate the system platform for several planned functional enhancements that were being driven by new, time-critical government legislation.

The administration system was a monolithic CASE-based system comprising over 140 executable load module components. Changes were implemented by replacing only the individual load modules in production that were affected by the change. The number of load modules impacted by a system change varied substantially. A major functional upgrade may be isolated to just a handful of load modules whereas a simple one line code change could impact 60 or 70 load modules. This resulted in substantial disparities in the ratio of development versus testing effort for any particular software change. Small bug fixes usually only involved changing one or two modules, so they could be implemented in production at anytime, independently of a release.

The business owner for this particular maintenance release project was the help desk manager, because the help desk carried the burden of assisting customer service officers in navigating around
these problems in the system when they processed registration transactions for customers. As this was an atypical release (that is, not the usual new feature upgrade), the business owner thought it would be more efficient to sit with the programmer ‘agile-style’, and make the system changes directly on the programmer’s screen. This would also mean that no formal systems analysis and design specification would be done for the changes, contrary to the usual practice for release projects.

While making changes to one software module, the pair noticed a piece of code that made no sense to them, so they changed it to what they thought it should be. Fortunately, the quality control process picked up the unauthorized change during subsequent testing. The change had interfered with an obscure but important registration function. On quarterly registrations of heavy vehicles over three years old, a roadworthiness certificate would not be required on the main transaction screen. However, it would still be prompted for on a later confirmation screen.

A group of system stakeholders met to discuss the issue and the system owner decided that the release could proceed and be implemented in production that night. Other time-critical releases were backed up, waiting for the large collection of load modules in the maintenance release to clear the testing environments. The erroneous change would be reversed the next day in the affected load module and updated in the system that next evening. The risk was considered to be low because the bug would be present for only one day; this transaction type was rare; the secondary screen still prompted for the certificate; and an alert would be sent out to customer service officers informing them of the issue and what to do about it.

Concurrently with this activity, a man was retrenched from his job after driving a delivery truck for a company for twenty years. He decided to go into business for himself so he bought a second-hand heavy vehicle. Because his operating funds were limited, he decided to register the truck for an initial period of three months to preserve his cash flow and see how the business developed. He was not familiar with the procedures for registering a truck so he did not get a roadworthiness inspection done on the vehicle. He went to his local DMV office to register the truck on the day the system was exposed. The customer service officer processed the transaction and thought it odd that the system asked for a screen. This would also mean that no formal systems analysis and design specification would be done for the changes, contrary to the usual practice for release projects.

The system was repaired that night.

A week later, the truck failed to stop in a line of traffic, slamming into the back of a small car, killing the two occupants. The truck was found to have defective brakes.

How could such an unfortunate sequence of events be avoided? Many books have been written on this phenomenon (e.g., [2], [3], [16] and [19]), but the risk persists. Certainly the initial software process change and associated unauthorized program change initiated one stream of the set of coincidences that led to tragedy. But it was not obvious anywhere along the chain, until the very end, what the consequence of the inter-linked series of events might be. So it is not obvious how to avoid this possibility other than to be aware, as in the previous fallacy, that the implication of a particular process action can extend way beyond and above the immediate process level. Developing a learning culture that learns from past mistakes, builds in extra safeguards when expensive or dangerous process changes are uncovered, and develops a capability for early warning detection of high risk trends and patterns of activities are also positive counter measures to runaway chains of mistakes.

By repositioning the organizational components, we can use the socio-technical model from Figure 1 to conceptually point to two alternative approaches to process improvement, shown in Figure 2, that conform to a consistent path of realignment following an intentional change to the established configuration.

The top-down path is the traditional planning-based approach to change that starts with a strategy and matching structure, and proceeds to process improvement, followed by the development of matching roles and skills (people) and technology to support the change. The main problem with this approach is that development of supporting competencies tends to lag the other changes because they take time to build. This top-down planning-based approach was the established practice in software engineering at the DMV in the case above. However, enthusiasm and inexperience in software development best practice led to a hastily conceived change in process.

An alternative approach is to lead from the bottom-up through learning-based competency development (people and technology), followed by process changes to build on this new capability, which then opens up new strategic options for the organization (strategy and structure). This is what the outsourcing organization did (discussed in Fallacy 1) after the initial contract term. They used an operations level management committee to build new capabilities from the bottom-up to realign the organization after the bumbling start to their new sourcing strategy. In the DMV case, however, the business owner and programmer tried to be more agile in the software engineering process they followed but did not base this change on prior learning and competency building in the functionality of the code they were changing, resulting in a misalignment.

Fortunately, the occurrences described in this case are extremely rare. However, ‘stuff’ does happen, even with software processes. The case illustrates in an extreme way that otherwise innocuous commercial processes can have serious impacts beyond just business performance.
Fallacy 4: The enterprise as an automated process

A view has emerged in the process marketplace, under the banner of BPM (Business Process Management), of the enterprise as a set of modeled business processes that can be directly executed and continuously monitored, changed and controlled by business managers without conventional software development or intervention by the IT function. This is different to the traditional view of process improvement as the deployment of a methodology and tools for discrete process changes. It is made possible by a combination of technologies packaged into a business process management system (BPMS) (see, for example, [25] and [9]). It aspires to enable autonomous process optimization or system-guided process improvement with minimal human involvement.

In general, this process-centric view presents a powerful vision of a future possibility rather than a description of present reality. Academic researchers and commercial product developers are still discovering the form and boundaries of IT-enabled process automation as a practical business capability. As one proponent notes, at present, “there are so many vendors pushing a wide variety of tools that users can be forgiven if they are uncertain whether BPMS is really just Workflow or EAI ([Enterprise Application Integration]) or Process Modeling with a new name, or if there is something else there” [10].

BPM is a departure from traditional thinking. Its stated promise is not to speed up or automate applications development as a basis for increasing organizational agility and adaptability. Rather, it is to eliminate software development by enabling business processes to be directly and immediately executable from process models. In conjunction with the ability to monitor, continuously improve and optimize the whole value chain, this could provide flexibility and responsiveness to business changes unmatched by current IT-based capabilities.

These goals appear to have some resonance in practice. In a recent industry survey of 274 people interested in BPM (only 2% had “no interest” in BPM), 56% indicated that their companies were involved in BPM to save money or improve productivity, while 51% indicated they were driven by a “need to improve management coordination or organizational responsiveness”. Other drivers included “the need to improve customer satisfaction” (37%) and the “need to improve products” (36%) [11].

This is a space to watch, which offers great potential. Currently it is immature [11] [18], but growth is expected to be 40-50% in 2008 [11]. However, if the potential of BPM is realized, not all organizations are likely to benefit from this technology. Business models with a strong process orientation that can substitute automation for manual processes may be better placed to benefit [22]. Businesses with processes characterized by high levels of standardization, repetition, stability, manual effort, discrete operations and a high ability to be specified are more likely to benefit – at least in the early stages until the technology develops and matures. Businesses with processes that are executed on a one-off or infrequent basis, are characterized by novelty, have high variability, require specialized skill inputs, or are embedded in complex operational contexts may be less able to benefit from BPM, especially in its early forms. Industries currently experiencing the greatest success in adopting BPM are financial services and insurance [18]. A challenge for early adopters is to carefully filter the practical capability from the promotional ‘hype’ in their particular business context.

Adopters might also reflect on the implications of the socio-technical model in Figure 1. The model suggests that the components can work together to enable and reinforce a new organizational capability such as automated business processes or they can conspire to make it hopelessly ineffective. Apart from suitability of the technology for the business model of the organization, what will determine which effect wins is the strength of the initial state of alignment (or misalignment) and the way in which the change is introduced (that is, the change management strategy).

Critically, the model suggests that investment in BPMS by itself, as a process improvement initiative, is unlikely to deliver the level of positive business impacts expected. The greatest value is likely to come from the innovation being absorbed into a newly aligned configuration through the development of complementary assets, creating a new organizational capability. Important complements are likely to include the business model and strategic orientation and context of the organization (does it build to stock or to order, and is it a leader or follower in a stable or volatile market?), the commitment and capability of business managers to own and run the system (can they make the cultural and skill transitions to manage through a computer?), and the retention of skilled IT staff to support the system. Like other off-the-shelf enterprise systems (such as ERPs and CRMs), the operational, strategic and competitive value derived from the investment is likely to be a function of the competency accumulated in applying and using the product [6] [29]. Introducing a new technological capability – even a potentially very smart one – offers no certainty of a positive business payoff.

3. CONCLUSION

This paper has argued that while process improvement offers opportunities for improved business performance and competitive advantage, benefits will not flow automatically. Organizations are complex, multi-dimensional entities. There is no simple direct link between process change and performance outcomes. Rather, benefits must be sought and captured as an integral part of the improvement initiative or change program. This was illustrated by considering four fundamental assumptions about process improvement that can be as fallacious as they can be true. It is argued that key to capturing business benefits is applying corresponding changes to complementary organizational assets, such as those illustrated in the socio-technical model in Figures 1 and 2. A well-intentioned process change can have deleterious effects if it is idly implemented or not explicitly managed.

Ironically, the success of the process innovation as a management strategy is ultimately dependent upon the ability of managers to apply and extract benefits from it, as an organizational resource. More positively, process improvement offers the potential to incrementally or radically transform firm capabilities to enact new strategies and adapt to changing environmental circumstances. This applies equally to software processes, as the means of ‘manufacturing’ software-based products and internal systems, as it does to the business processes used in the manufacture of non-software-based products and services. As such, process improvement presents both opportunities and challenges for researchers to work out how to realize the potential as a business
option, and for managers to learn how to capitalize on the potential for business gain.

4. ACKNOWLEDGMENTS
NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

5. REFERENCES