Journal of Information Technology (2004) 19, 4–20
© 2004 JIT Palgrave Macmillan Ltd. All rights reserved 0268-3962/04 $25.00
palgrave-journals.com/jit

State of the art

Technochange management: using IT to drive organizational change

M Lynne Markus

Trustee Professor of Management, McCallum Graduate School of Business, Bentley College, Waltham, MA 02452-4705, USA. Tel.: 781-891-2312; Fax: 781-891-2896; E-mail: mlmarkus@bentley.edu

Abstract

Using IT in ways that can trigger major organizational changes creates high-risk, potentially high-reward, situations that I call *technochange* (for technology-driven organizational change). Technochange differs from typical IT projects and from typical organizational change programs and therefore requires a different approach. One major risk in technochange—that people will not use information technology and related work practices—is not thoroughly addressed by the discipline of IT project management, which focuses on project cost, project schedule, and solution functionality. Organizational change management approaches are also generally not effective on their own, because they take as a given the IT "solutions" developed by a technical team. Consequently, the potential for the IT "solution" to be misaligned with important organizational characteristics, such as culture or incentives, is great.

Merely combining IT project management and organizational change management approaches does not produce the best results, for two reasons. First, the additive approach does not effectively address the many failure-threatening problems that can arise over the lengthy sequential process of the typical technochange lifecycle. Second, the additive approach is not structured to produce the characteristics of a good technochange solution: a *complete intervention* consisting of IT and complementary organizational changes, *an implementable solution* with minimal misfits with the existing organization, and an organization *primed to appropriate the potential benefits* of the technochange solution. With hard work and care, the combined IT project management plus organizational change approach can be made to work. However, an iterative, incremental approach to implementing technochange can be a better strategy in many situations. The essential characteristic of the *technochange prototyping* approach is that each phase involves *both* new IT functionality *and* related organizational changes, such as redesigned business processes, new performance metrics, and training.

Journal of Information Technology (2004) **19,** 4–20. doi:10.1057/palgrave.jit.2000002

Keywords: IS implementation; IT project management; organizational change management; resistance to change; system development life cycle

Overview

When organizations adopt new information technologies, the potential for significant transformations in people's work, in organizational business processes, and in organizational performance outcomes is sometimes – but not always – there. Using IT in ways that can trigger major organizational changes creates high-risk, potentially high-reward, situations that I call *technochange* (for *techn*ology-driven *o*rganizational *change*). Technochange situations differ sharply from other organizational uses of IT, and managing technochange successfully requires a different approach to solution design and implementation. This paper is about technochange and the deliberate *technochange management* strategy of using IT to drive organizational change.

Technochange: unintended outcome or deliberate strategy?

Organizations can do many useful and valuable things with IT *other than* changing organizational activities and performance results. Technical and economic benefits can be had through simple upgrades and technology substitutions that do not require much effort on the part of IT 'users'. Think of upgrading a departmental web server: it all goes well, it happens overnight and if users notice at all they enjoy somewhat better response time. The payoffs of IT projects like this can be significant, but they often do not make an appreciable difference in how the organization works.

Other new uses of IT, however, have the potential for big improvements in organizational performance – and the potential for all kinds of havoc and disruption for employees, customers, and other stakeholders. An example is a new accounting system that eliminates many specialist accounting jobs and requires business unit managers to master new accounting codes, new approval routines, and new ways to request and analyze financial data. Another is a web-based 'self-service' diagnostic tool that changes the relationship between customers and service providers. These are examples of technochange.

Managers sometimes get involved in technochange without really intending to. For example, enterprise resource planning (ERP) systems promise major strategic benefits and process improvements from cross-functional integration and process streamlining, but they are now notorious for their implementation challenges and problematic organizational consequences.[1] Many organizations implemented them, not for their potential performance improvements, but to address year 2000 concerns or to replace aging legacy systems that were running out of capacity. In one such organization, managers were unable to identify the business benefits of their ERP implementation, because they had approached effort as an IT project rather than as a business initiative:[2] managers stepped aside and gave the IS specialists a free hand. In many other ERP-implementing organizations, managers have been surprised or disappointed when the implementations proved difficult, did not have the desired outcomes, or even failed. By treating technochange situations as if they were IT projects, these organizations risked being blindsided by implementation problems and unintended consequences. In addition, they missed significant opportunities to benefit from IT-enabled organizational performance improvements.

Not all technochange situations 'just happen' to organizations. Some managers deliberately use IT as a strategy to drive organizational change. An executive at a health products manufacturing company recently explained how his company was 'entering a new era where *we're using technology as a catalyst to stimulate business changes*.' As one example, the company installed an ERP system to achieve procurement savings that had a noticeable improvement on the company's bottom line. Using IT that way required alterations in the procurement process and the work of people involved in purchasing. When asked whether the procurement savings could have been achieved *without* the ERP system, the executive replied:

> You know, this is an age-old argument we have here all the time. … The answer is in theory, yes. In reality, the culture of this company is that *it never would have happened without a big thing like an ERP out there forcing us to do it.*

Similarly, a financial services company executive explained that his company was going through a painful process of converting every operating unit to the same technology platform in order to achieve big improvements in how his organization works.

> I wish we were done with this, quite honestly, but *there's nothing better to get people on the same page*, to get your arms around them in terms of brand and culture, *than to have everyone on the same [IT] operating platform*. When you have different products, technology, pricing, and so forth, it is really – it's a challenge to bring everyone together. We view this deployment as *a very important strategic advantage* as we acquire new companies, because we'll have a single platform that will cover all channels, all products – you know, the whole universe here for us.

Just as unintentional technochange can have undesirable outcomes, deliberate technochange management can also turn out badly. Therefore, it is vitally important for both business managers and IT specialists to understand the dynamics of technochange and how to manage it well.

Different approaches to deliberate technochange

The premise of this paper is that using IT strategically to drive *organizational* performance improvements is fundamentally different from both IT projects and organizational change programs. Unlike IT projects, which focus on improving technical performance, technochange involves great potential impacts on 'the users' (people, processes, and organizational performance). Technochange also differs sharply from traditional organizational change programs, because information technology, information technologists, and technical methodologies are so prominently involved in technochange. As a result of the big differences between technochange and either IT projects or organizational change programs, tried-and-true approaches to managing IT projects or managing organizational change are not sufficient (alone or together) to ensure successful technochange. Effective technochange management requires a different kind of attention to the features of the '*solution*' and a different change *process* from those prescribed by either IT project management or organizational change management.

Not everyone believes that using IT to drive organizational change needs a special approach. Many technical specialists and consultants sincerely believe that good IT project management is *the answer* to technochange success. Yet experts estimate that as many as 75% of organizational change efforts involving technology fail (even when the technology performs acceptably) because of people's negative reactions to changes in their work, organizational business processes, and the technology they use.[3] Although the discipline of project management is a very important contributor to successful technochange, it has very little to say about how to manage the risks associated with people's 'resistance to change'.[4]

Organizational change management experts believe they have the solution to the problem of resistance. They argue that change efforts should focus on the *people* affected by the change. They recommend tactics such as assessing people's readiness for change, training them and initiating cultural changes, redesigning jobs or organizational structures, devising new ways to manage and reward people, or involving them in planning the implementation of change. But organizational change management has very little to say about how IT alters the problem of organizational change. Much of the extensive literature on how to change organizational behavior does not even mention IT,[5] not even as an example of something that can prevent desired organizational change.[6] Although organizational change management activities can play a very helpful role in successful technochange, they are not sufficient, because they do not address the unique aspects of IT-driven organizational change.

Many organizational change experts would go so far as to argue that using IT as a driver of change is not *the right way* (or even an acceptable way) to bring about big improvements in organizational performance. (The focus should be on people.) But organizations increasingly pursue technochange anyway, often with great success. Why is technochange such an attractive strategy for bringing about organizational change? Two reasons stand out.

First, some organizational changes simply cannot happen *without* IT. For example, AlliedSignal Aerospace needed to change the way its salespeople related to customers and could not do so without new IT.[7] The company had 40 different product lines and four independent business units – each calling on customers and not sharing information with each other. Some large customers had to deal with as many as *50* contacts in AlliedSignal, and they were not happy about it. The company had to change or risk losing sales. Part of the answer would surely be changing salespeoples' attitudes and behavior around sharing information with people in other units. But without new IT, that change would be unsuccessful. The company had no common technology platform for sharing information about customer contacts and sales opportunities – each unit had its own. Therefore, AlliedSignal initiated organizational change by deploying customer resource management (CRM) software.

A second reason that managers use IT to drive organizational change is that starting major cross-functional changes *without an IT focus* does not work in many organizational cultures. The health products company executive quoted earlier stated that this company would not have undertaken change without being forced into it by a large IT implementation. People in his company were busy (who is not!) and very focused on the performance of their own business units. Improving the procurement process affected the entire organization and required the efforts of people in many business units, but people found it difficult to initiate and sustain interest without a highly visible project to galvanize attention. In the health products company, as in many others, the major expense of enterprise-wide IT implementations serves this function well.

A less creditable explanation for the popularity of the deliberate technochange strategy is the belief, widespread among both business managers and IT specialists, that IT alone is enough to create significant improvements in organizational performance. I call this 'magic bullet thinking'[8] and it is sadly mistaken. IT is *not* a magic bullet, and using IT to drive organizational change is a strategy that can drive organizations (and their drivers!) straight into failure, unless it is done well.

Preview

Well-designed and well-implemented technochanges can produce significant improvements in organizational performance.[9] But the qualifiers 'well-designed' and 'well-implemented' are key. Not all ideas for IT-driven organizational change are sound. When the ideas are sound, they are not always successfully translated into good IT solutions. Even when the IT solutions are good, they may not be integrated into good *technochange* solutions, and they may not be introduced into the organization well. Successful technochange is not just good IT project management or good organizational change management (or both together). It requires an entirely different approach.

The next section spells out more fully how technochange differs from both IT projects and organizational change programs and why successful technochange management requires a different approach. The following section outlines the characteristics of good technochange solutions and an effective process for designing good solutions and getting them to produce results.

Technochange is different and needs a different approach

Technochange situations are different from IT projects on the one hand and organizational change programs on the other (see Table 1). Technochange initiatives are usually expected to produce significant improvements in organizational outcome measures such as process efficiency or cycle time. Some organizational change programs also have performance goals, but many are more diffusely expected to enhance organizational culture or 'effectiveness'. IT projects, on the other hand, usually have narrower goals than deliberate technochange initiatives: they focus on improving technical performance (e.g., reliability, speed, functionality) and the costs of technical operations (e.g., total cost of IT ownership, system maintenance costs, etc.).

The three approaches also differ in the means by which the target outcomes are achieved.[10] The organizational change management approach relies on interventions that focus on people, organizational structures, and human resource management policies, whereas both IT projects and technochange initiatives focus heavily on IT. In effective technochange, IT solutions are complemented with related organizational changes to achieve an appropriate fit between IT and the organization. (Unfortunately, many technochange situations are treated like IT projects, with less than satisfactory results.) Some examples will clarify the differences among the three approaches.

Technochange *vs* IT projects and organizational change programs

IT projects contribute to organizational success by improving technology functionality, reliability, and cost of operations without significant changes in how the organization operates. For example, an organization might remove computer capacity constraints that slow business activity and frustrate employees by adding processors or memory to a computer system. Or, a business might try to reduce the maintenance and support expenses associated with 'thick client' software by web-enabling core transaction processing systems.

In another example of an IT project, one hightech organization installed a data warehouse to make it easier for people in the IS department to prepare ad hoc accounting reports for business managers.[11] The company's basic recordkeeping systems were outdated, unintegrated, and did not supply needed management reports. Consequently, managers turned to the IS department to write reporting programs for particular needs. Although the IS

Table 1 Technochange *vs* IT projects and organizational change programs

	IT projects	*Technochange situations*	*Organizational change programs*
Target outcomes	Technology performance, reliability, cost of operation and/or maintenance, within project schedule and budget parameters	Improvement in organizational performance	Improvement in organizational culture and/or performance
The solution	New IT	New IT applications, often in conjunction with complementary organizational changes	Interventions focused on people, organization structure ad culture, or human resource management policies
Example	Replace outdated management reporting software with data warehouse and analysis tools to reduce time that in-house IT personnel spend preparing ad hoc reports at the request of business managers	Achieve significant savings in procurement through a restructuring of the procurement function (centralizing at headquarters the process of contracting with vendors of key supplies and consolidating purchases to achieve deep discounts) in conjunction with the adoption of a new procurement software package that will allow headquarters to monitor business units' compliance with the consolidated purchasing contracts	Transform a mature organization that is underperforming its competitors by making people more innovative, customer-focused, and empowered to take initiative and make decisions
Basic approach	The 'project' – a temporary organizational structure led by a project manager who is expected to produce an outcome (e.g., a working system) that meets stated specifications on time and within budget	Typically, an IT project followed by implementation efforts; in effective technochange management, a 'program' of change initiatives of which an IT project is one; others may include organizational or business process restructuring, change in reward systems, job redesign, training, etc.	Organizational development – the umbrella term for a collection of change methodologies that target one or more of the following: managers' attitudes and behaviors, human resource development and training, organization culture, reward systems, job redesign, organizational structure, etc.
Role of organization's managers	Oversight – to approve the project, to provide funding for the project, possibly to initiate the project by identifying need, sometimes providing input for requirements specification	Leadership – to initiate the project, to act as sponsors and champions of change, to explore process options enabled by the new technology, to design and implement non-technology changes, to change their own management systems and behaviors as required to ensure benefits, to provide key design inputs and oversight for the IT project	Leadership – to initiate the change effort, to change their own management styles and behaviors in ways to lead by example, to reward the desired new behaviors and the achievement of objectives, etc.
Role of IT specialists	Central – to perform the project management role and most of the project labor; to coordinate with business managers and with vendors and external consultants	Central – to work together with organizational managers and other specialists to design a technochange in which the IT part meshes with other changes to achieve desired objectives; to lead and staff the IT project	Negligible
Role of other specialists	Technology vendors and consultants may perform various tasks	Internal staff specialists (human resources, industrial engineering, strategic planners, etc.), external management and technology consultants, and technology vendors may all play key roles	Internal human resource management and organizational development specialists and external management and organization development consultants often perform key roles

Table 1 Continued

	IT projects	*Technochange situations*	*Organizational change programs*
Key success factors	Project manager performance, technology performance, vendor performance	Performance of organizational managers, performance of internal and external organizational change consultants; project manager performance, technology performance, vendor performance; tight ongoing coordination between people involved in the organizational change program and the IT project	Performance of organizational managers, performance of internal and external organizational change consultants

department was happy to continue in this role, the demand for *ad hoc* reports was so great that a backlog had developed. Business managers were unhappy not to be getting timely reports, but the company did not want to add more IS personnel. Replacing the legacy systems to eliminate the need for special reports would have been expensive and would have disrupted business operations. So, the organization found another, also expensive, solution that did not affect the business. By building a data warehouse, IS personnel were able to integrate and reformat the data from legacy information systems in a way that substantially reduced the time required to program management reports. They also coded new routine reports that reduced for managers' requests for special purpose reports.

Unlike this data warehousing IT project, which involved the rest of the company in only minimal ways, technochange alters organization behavior and outcomes significantly. An example of technochange involving the same data warehousing technology concerns a consumer products company that wanted its brand managers to take a more 'scientific', data-driven approach to marketing its products through the use of sophisticated analysis programs and a data warehouse. Although this company deployed the same technology as the hightech company, its goal was very different. Instead of trying to reduce the cost of IS operations, as at the hightech company, the consumer products firm wanted *to increase sales*. And, whereas in the hightech organization it was the IS people who used the new technology, in the consumer products organization it was *brand managers* who had to use the data warehouse and change the way they made decisions about marketing promotions. Fortunately, the person in charge of data warehousing at the consumer products company clearly understood that the goal was organizational change; he described a painstaking process of one-on-one coaching by which he introduced the marketers to linear regression (without ever calling it that!) and changed the culture of the marketing organization.

The AlliedSignal Aerospace CRM project mentioned earlier provides another illustration of technochange. This effort was kicked off like an IT project, but it turned into technochange when it became apparent that IT alone would not achieve the desired results. When the CRM system was first implemented, salespeople did not use it. Investigation revealed that getting the benefits from CRM software required a new sales process in which members of different business units shared information and coordinated their sales efforts. But the salespeople were familiar with their old tools and processes, did not know what better processes based on the new system would look like, and probably did not have the time or motivation to figure it out on their own. Recognizing that the change effort was stalled, CRM sponsors changed their approach. They appointed a CRM manager who performed business process mapping for the affected sales units, recommended software-supported process changes that would increase efficiency, and provided one-on-one coaching. Eventually, the old sales information was migrated to the CRM software, legacy systems were terminated, and, more importantly, the business goals were achieved. At the end of the day, there was a lot more to this success story than a successful IT project.

Coming at the AlliedSignal situation from an organizational change management approach would have looked very different from what AlliedSignal did. Organizational change management experts might have tried to identify individual and organizational barriers to information sharing and attempted to break them down through communication and training. They might have identified representatives of the groups affected by the change and asked them to design the new processes. Only if the change team recommended new IT would an IT project have been launched under the direction of IS specialists.

The organizational change management approach is sometimes combined with IT project management. The challenges of large ERP implementations led experts to recommend running both IT project and organizational change management processes in parallel.[12] While one team configures the software and tries to get it up and running, the other team assesses people's readiness for change, tries to improve their ability to change through training, and attempts to increase the organization's acceptance of the IT solution through communication. Although this combined approach can produce better results than an IT project approach without organizational change management, it suffers from lack of integration between IT development and organizational change management activities.

Why technochange needs a different approach

Technochange situations call for big improvements in organizational performance. These improvements cannot happen unless tasks, jobs, and organizational processes all change along with IT. As we have seen, achieving a coordinated sales approach requires not just CRM software, but salespeople willing to sharing information with people in other units. It also demands new processes and behaviors, such as new rules about who gets to go on a sales call and who gets credit for making a sale. In the best case, it takes time and hard work for people to agree on what these new processes should look like. In the worst case, people do not agree, usually because the desired new ways of working conflict with the organization's reward systems. Why should a salesperson share information with someone in another unit if it means that the other sales rep gets the bonus? Expecting that new IT will solve these problems for you is magic bullet thinking – and it often leads to failure.

Unfortunately, the discipline of IT project management does nothing to eliminate this source of failure. Project management techniques and methodologies evolved to control different threats to project success – the threat that the *technology* will not work as expected and the threat that the budget and schedule will not be sufficient for project completion. These are indeed serious threats, and they should be controlled. But controlling them does not control the major risks in technochange. The major risks in technochange are the risks that 'users' (employees, customers, suppliers, etc.) will *not use* the technology, that they will *misuse* it (that is, use it in ways project sponsors did not expect), or that they will use the technology without capturing the expected benefits.

These *organizational* risks are major threats in technochange, but they are not major threats in IT projects. The impacts of IT projects are usually quite narrowly focused. Direct effects are often confined to the technical specialists who support the technology. 'Users' do not have to do anything differently; often they may not even notice the change. In true IT projects, benefits are almost automatic once the new technology is in place: 'down time' decreases, or maintenance bills fall.

In technochange situations, however, the risks of nonuse, misuse, and failure to capture benefits are very high. And IT project management approaches do not control them. Treating a technochange situation like an IT project does not work.

Treating technochange situations solely as organizational change programs also does not work. Organizational change programs evolved to address the very serious risks that people may not be motivated enough, not be skilled enough, or not allowed by their managers and organizational practices to behave in new ways. Organizational change programs address these risks by training managers and subordinates and creating conditions conducive to new ways of working. The risk that an organization lacks readiness for change is serious, and it must be controlled. But controlling that risk does not control all the important risks in technochange. In particular, it does not fully control the risk of technology taking precedence over human concerns or the risk of a bad IT solution with negative human and organizational consequence.

As mentioned earlier, new IT is necessary for some desired organizational changes. Legacy information systems can be a serious drag on an organization's ability to move in new directions. Consequently, new IT, IT specialists, and IT development methodologies are prominent (even dominant!) in many organizational change efforts. In addition to whatever needs to be done to prepare people and the organization for change, considerable time and effort is required to specify, select, purchase, customize, test, and install IT. Even when software packages and external services are used instead of in-house development, the effort, cost, and problems associated with new IT can take over the change effort. When this happens, the focus on technology overwhelms concern for human and organizational issues (if it was there in the first place).

Even more important in technochange is the risk of a bad solution. By 'solution', I mean the features of the change itself. Technochange solutions can be bad in two major and related ways: they can be incomplete, and they can be misaligned with the organization. Briefly (since we will return to these points later), *incomplete* technochange solutions are simply *IT* solutions; they lack the other supportive changes required for use of the IT solution to yield the desired results. For example, getting the desired results from CRM software may require reallocated sales territories and new salesforce incentive systems. *Misaligned* technochange solutions may or may not be complete, but they conflict so seriously with the existing organization that they are likely to be rejected. For example, health information systems that require doctors to key enter orders rather than handwriting them or dictating them to nurses sometimes offend doctors' sense of status and fail to gain their acceptance.

These two characteristics (completeness and alignment or 'fit') of good technochange solutions are important, and we will return to them. The point here is that combining a traditional organizational change program and an IT project does not eliminate the risk of a bad technochange solution. When IT projects and organizational change management programs are combined, the efforts are loosely coupled. The IT project team designs the IT solution, and the organizational change team designs organizational changes and tries to get people ready to accept the IT solution. The problem is that the 'organizational solution' is expected to conform to the 'IT solution', when a better outcome would be an integrated technochange solution.

The reasons are several. First, when an organizational change team gives an IT team requirements for a technical solution, their requirements are not always informed by full awareness of what IT can and cannot do. Representatives of a business unit may know they need a new system, but may not know that there are vast differences between systems of the same type offered by different software vendors. They also may not know how much flexibility there can be in the way business processes can be redesigned around a software application. If they were better aware of the options, they might give the IT design team a very different charter.

For example, a company might decide that it needs to do a better job of coordinating its buying from its major supplier. But there are many technochange options under the heading of better 'supply chain management' (SCM). Here are just a few:

- The company could buy a software package to improve its own demand forecasting in order to place smarter orders with the supplier.
- The company could give the supplier online access to its sales and forecast data so that the supplier would be better prepared to anticipate and fill the company's orders.
- The company could require the supplier to manage the inventory on a consignment basis (with penalties for stockouts) by accessing the company's point-of-sale data; in this case, the company no longer has to place orders.

A particular SCM package may support only one of these possibilities. Even so, it may support several quite different variations on the same theme.

Unless decisions about new IT and business processes changes are integrated, the likelihood of an incomplete or misaligned solution is great. The real payoffs of technochange come from creating new products and processes that exploit the potential of IT, rather than just fitting IT to the way processes are currently done. It is very difficult to make this happen when those requesting an IT solution are unaware of the possibilities and decoupled from making the technical decisions.

Second, even when the architects of organizational change know exactly what they want in the way of IT, there is no guarantee that an IT solution developed independently will actually fill the bill. Although 'user representatives' on IT project design teams are common in enterprise system implementations, their involvement may not be sufficient to guarantee a good IT solution.[13] People involved in participatory design processes often produce incremental solutions, even when more radical change is needed. In addition, technical personnel and methodologies often unduly influence the outcomes of collaborative design processes. Intentionally or unintentionally, IT design decisions are often made on the basis of technical criteria, even when participants voice alternative business needs. If the result of technical design decisions is a misaligned system, organizational acceptance may be impossible, no matter what the change team does.

The problems that arise when organizational change design and IT development are decoupled are best understood by examining the lifecycle of a typical large-scale enterprise IT project. It will be clear that, no matter what kind of change program is carried on in parallel with the IT project, the IT project often sets its own course. When this happens, bad technochange solutions, disappointing performance outcomes, and unintended negative consequences often result.

Problems in the technochange life cycle

It is useful to think about the technochange life cycle in terms of what happens before, during, and after an IT project. (If an organizational change management program is also part of the technochange process, it usually runs in parallel with the IT project and subsequent life cycle phases.) *Projects* are temporary organizational structures, usually managed outside of the chain of command of the operating units in which the IT will be used. The project structure is intended to enable project managers to concentrate on developing the IT solution, leaving operating managers free to focus on operations until the solution (and related organizational changes, if any) is ready to go. The downside of the project structure is that project teams members are spatially and temporally distant from the rest of the organization and this causes certain predicable problems.

Before the IT project, a technochange idea is proposed, reviewed, approved, and funded. After the IT project, activities can be divided somewhat arbitrarily into two phases. In the first, the target operating units start up with the new IT (and ideally with redesigned organizational processes) and attempt to 'shake down' the problems that occur when, as many consultants put it, the organization tries 'to change the tires on a moving car'. The goal of this phase is to reach a state of stable operations, in which technology use and new ways of working become routine. Not all technochange initiatives reach that point: quite a few are terminated when shakedown problems prove to be severe.

During the second post-project phase, the organization (should) try to capture the expected (and the unanticipated!) benefits of the technochange. Research suggests that most of the benefits of organizational innovation are realized considerably after the shakedown phase. They occur as a result of people learning better how to use technology and fine-tuning it, continuously improving processes, finding new uses for the technology, and other 'benefit capture' processes.[14] Again, not all technochange initiatives reach that point: organizations sometimes 'declare victory' for a technochange but stabilize operations at a much lower level of performance improvement than expected.

Each phase of the technochange life cycle – chartering, the IT project, startup and shakedown, and benefit capture – involves different actors and activities and has characteristic problems[15] (see Table 2). The sections below discuss two important dynamics of the technochange life cycle – time and distance effects and exported problems – and the major negative consequences of those dynamics – failure, diminished ability to change, and other unintended consequences.

Time and distance effects

The first important technochange life cycle dynamics is *time and distance effects*. The project phase can take a long time, during which project team members are organizationally and psychologically distant from the ongoing operations of the organization. This distance occurs even when the project is staffed with representatives from the target operations. Almost from the moment operations personnel are assigned to a project, they cease to be 'representative' both in worldview and in the perception of people in the operating units. In my own research, I have heard operating personnel refer to their former colleagues as 'those IS people' just 2 weeks after the colleagues joined an IT project team. In other research, 'user representatives'

Table 2 Problems in the technochange life cycle

Phase	*Chartering*	*Project*	*Shakedown*	*Benefit capture*
Description	'Ideas to Dollars' – phase during which the technochange idea is proposed, approved, and funded	'Dollars to Solution' – phase during with the technochange solution is developed and technology is acquired or built; end when technochange starts up or 'goes live'	'Solution to Usage' – phase during which the organization starts operating in a new way with technology and the organization troubleshoots problems associated with technology and new processes; the goal of the phase is 'normal operations'	'Usage to Dollars' – phase during which the organization systematically derives benefits from the new way of working; may involve continuous improvements, 'upgrades', and 'conversions' of various kinds
Key actors	Organizational executives, operations managers, consultants, vendors, IS specialists	Project manager, project personnel (staff specialists, operations representatives, consultants), vendors, steering committee and/or project sponsors	Operations manager and operations personnel, some project personnel, IT operations and support personnel, vendors	Operations managers and personnel, others as invited
Prescribed activities	• Proposals for change • Evaluation and approval • Identification of solution constraints • Selection of project manager • Allocation of resources	• Development of project plan • Selection and training of team members • Analysis and design activities • Technology selection/building • Infrastructure development • Development of complementary changes • Implementation planning • Communication and change management • Testing • Data conversion • Documentation and development of training materials • User training • Rollout and startup	• Problem identification and analysis • Rework activities including technical fixes, procedure changes, additional training, adding personnel to handle backlogs, etc.	• Evaluation of technochange outcomes • Benefit capture • Continuous improvement • Skill building and retraining • Technology upgrades and/or conversions
Problems commonly experienced	• Uncertainty about costs, benefits, and risks • Disagreements about how to proceed • Pressure and lobbying from vendors, consultants , investment analysts, media, peers, others • Resistance • Insufficient funds availability	• Inadequate resources including operations representatives and specialized technical skills • Turnover on the project team or among project sponsors • Project team conflicts • Inadequate technology or vendor support • Business changes leading to scope changes, budget cuts, or schedule advances • Time pressures • Resistance from operating units • Project cancellation	• Premature disbanding of project team • Training proving inadequate • Technology and/or new procedures not operating as expected • Operations disrupted • Backlash from customers or other business partners • Pressure to revert to old systems and procedures • Termination of technochange	• No auditing of technochange performance • No benefit capture • Lack of learning and continuous improvement • No skills assessment • Unwillingness to upgrade or convert owing to painful implementation experience
Problems commonly exported	• Inadequate analysis • Magic bullet thinking (approaching technochangeas an IT project) • Underfunding of change costs (e.g., training) • Lack of communication about need for change and solution constraints • No real incentives or ownership by affected operations managers • Unintended consequences	• Scope cuts that affect promised technochange functionality • Shortcuts, particularly involving data conversion, testing, and user training • Failure to design complementary changes • Inadequate implementation planning • Design errors attributable to inadequate participation or lack of socio-technical analysis • Unintended consequences	• Few operations personnel knowledgeable about technology • Premature stabilization of operations with inferior 'workarounds' • Failure to establish expected new patterns of behavior (e.g., use of data for decision making)	

of a design team were found to lose their ability to think like a 'user' within a matter of weeks.[16]

Time and distance effects can create 'drift' between the project and the business needs that led to it. An example involves an industrial product manufacturer that had doubled in size as a result of several mergers. As there were no common systems, executives had difficulty in forecasting revenues and making information-based management decisions. They consulted the CIO, who realized that integrating the myriad existing systems was impossible and that the year 2000 problem loomed. With executive support, he launched what turned out to be a major ERP system implementation. Review of the project plan shortly after launch revealed that it did not include provisions for revenue forecasting capability. This was not exactly an oversight: the CIO and project team members had concluded that revenue forecasting capability should be provided by means of a data warehousing project to be started *after* the ERP project was successfully completed. But the executives were never informed of this decision and probably would not have agreed. In the best case, the ERP project would have taken at least 2 years, so it could have been 4 or more years before the executives had what they had asked for. As it turned out, the CIO and the project team never got the chance to provide it: after months of development effort, executives cancelled the ERP project. Experts familiar with the project concluded that starting with a data warehouse would have been feasible and may actually have reduced the complexity of the ERP system implementation.

The longer IT projects continue, the more likely they are to drift away from their initial business objectives. The longer IT projects last, the more likely it is that the initial business needs and organizational processes will have changed, resulting in misaligned IT solutions. The longer IT projects go on, the more likely they are to lose management attention and be terminated. The longer IT projects drag on, the longer it is before the organization can recoup project expenses and reap net benefits. Sometimes, project delays coupled with changing business conditions entirely prevent organizations from capturing the intended benefits of their IT implementations. Time is money; therefore, time is the enemy of IT projects.

Exported problems

A second life cycle dynamics that contributes to the challenge of technochange is *exported problems*.[17] Exported problems are problems that arise during one phase of the life cycle but either are not recognized as problems or are not remedied at that time. Instead, they show up in later phases, when it may be too late or too expensive to fix them. The classic example of an exported problem is failure during the chartering phase to recognize that a proposed IT implementation is actually a technochange situation. No money is set aside for change management; not enough is set aside for communication and training. Some affected groups are not consulted or involved in the IT development process. The need for complementary organizational changes is overlooked. Only when the IT solution is completed and turned over to operating units who resist it does it become apparent that the solution is inadequate and that operating managers lack motivation to adopt it.

Exported problems occur during the project phase as well as during project chartering. A common project phase example concerns the project manager who runs into schedule or budget pressures (usually both). The project manager may decide to cut project scope rather than slip the schedule, leaving some important features to be delivered as 'future enhancements'. Unfortunately, the parts of the project that remain undone are those that deliver the business benefits or the features that would motivate users to use the system. Alternatively, the project manager may decide to scrimp on software testing, resulting in buggy software, or on training, resulting in users who make too many mistakes.

The startup phase can also export problems, when resources needed for shakedown are not available or prematurely removed. Cigna had that problem with its $1 billion IT overhaul and CRM initiative.[18] Facing financial losses and customer pressures, Cigna tried to accelerate the startup of its new systems. As the systems were rolled out, the company began laying off customer service personnel, because the technochange idea involved consolidating service centers. When Cigna moved 3.5 million customers to the new systems all at once, major problems occurred. Customers' complaints were so pronounced that analysts downgraded the company's stock. Eventually the problems were resolved, but similar rollout problems have forced other companies into bankruptcy.[19]

Not all problems in the technochange life cycle are exported. Some are 'experienced' and corrected right away. For example, the first time Microsoft tried to implement an ERP system, Bill Gates rejected the proposal during the chartering phase, because the rationale for the project was purely a technical one (to upgrade system architecture) rather than a business case.[20] Problems can also be experienced and corrected during the technochange project phase. A project manager may discover, for example, that the existing IT infrastructure will not support the purchased software and initiate an infrastructure upgrade.

In contrast to these 'experienced problems', exported problems do not go away, because they remain unrecognized or untreated. Instead they are passed on to the people, usually different, involved in the next life cycle phase. (See Table 2 for the actors and activities typical of each phase as well as common experienced and exported problems.) Since these new people have different knowledge and skills and are expected to perform different activities, they may not recognize the exported problems or be able to take appropriate action. Even when the new people suspect they have inherited an exported problem, they are under powerful pressure to ignore it. Trying to correct it by themselves is politically dangerous (they are taking action without authority), and the alternative – handing the situation back to the people of previous phase for rework – is also problematic. The people who pick up the ball at the 'handoff' from an earlier phase are not supposed to second-guess what the people before them did; their job is to take what they are given and complete their own work. No one wants the sequential lifecycle process to get off track. Project managers do not like for their projects to be rescoped, because it reflects poorly on their ability to handle the job and it could affect their performance measures (completing the project on time and on budget) and their bonuses. Executives do not like it when operations managers reject an expensive IT solution or demand that it be fixed. Consequently, there is a strong tendency for people to pass exported problems on.

The old adage about computers applies here better than anywhere else: garbage in, garbage out. If the prior phase exported problems, the current phase is likely to export them too, until they have nowhere to go. When the unresolved problems finally surface, they are much bigger problems than they would have been earlier. Unfortunately, the final resting place of many exported problems is in technochange failure, in an organizational culture that is hostile to future changes, and in unintended negative consequences.

Technochange failure, the culture of failure, and unintended consequences

The longer technochange efforts continue with exported problems, the more difficult and expensive it is to correct them. If exported problems show up as 'resistance' during the shakedown phase, the political costs of trying to 'overcome' them can be great. Often, the best course of action is terminating the effort, blaming the failure on unspecified 'technical problems'.

Canceling a failing project rather than sinking more time and effort into the attempt to overcome resistance may be a sound *business* decision. Unfortunately, failing in this way can have long-term negative side effects on the organization's ability to succeed in *future* change efforts. Cynicism ('here it comes again') and pain over failed change efforts can cripple people's willingness and ability to attempt another change.

Cynicism and defeatistism are unintended consequences of technochange failure. But even completed and apparently successful technochange initiatives can have unintended consequences.[21] *Unintended consequences* are outcomes that can be attributed to the technochange solution but that were not expected or intended by executives or project personnel.

Some unintended consequences are positive. For example, American Hospital Supply developed a computer-based ordering system for one of its key customers that had experienced persistent problems with ordering errors. The problems were solved, and the company soon recognized the potential of extending the new ordering system to other customers. Faster, more accurate ordering enabled customers to keep less inventory on site, reducing their costs, and reinforcing their choice of American as supplier.

However, many technochange efforts have unintended *negative* consequences. For example, in recent years, many companies have implemented telework programs, where employees do most of their work at home or on the road. These changes have the planned benefits of reducing space and administrative costs; they have the unplanned costs of reducing employee commitment to the organization. Similarly, service-oriented companies that install web-enabled self-service technology for customers sometimes experience erosion in their customer relationships.

Some unintended negative consequences may be inevitable. But others have their roots in problems exported from the chartering and project phases. In principle, therefore, they can be foreseen and prevented, or at least caught early enough that the damage can be contained.

Summary

Technochange initiatives are different from IT projects and organizational change initiatives and they have different risks. The key risks of technochange are the risk of IT non-use, misuse, and non-benefits and the risk of a bad IT solution. These risks arise from problems exported from phase to phase in the technochange lifecycle. Neither IT project management nor organizational change programs can control those risks, although they do control other risks that also apply to technochange situations. Therefore, neither IT project management nor organizational change programs, alone or together, fully address the risks of technochange. Successful technochange management requires a different approach to solution design and implementation.

The product and process of successful technochange

In successful techochange, both the solution and the process of arriving at the solution are important. If the solution is a good one, but the process of designing and implementing the change is poor, people may reject the solution. If the design and implementation process is good but the solution is poor, the business results will be disappointing. The quality of the solution and the quality of the design and implementation processes are certainly interdependent to some extent, because the solution is a product of the design and implementation process. Nevertheless, it is possible to have one without the other, and both contribute independently to the ultimate results. Therefore, savvy technochange managers attend to both the features of the technochange solution and the technochange design and implementation process. The next two sections outline the characteristics of each.

Characteristics of a good technochange solution

Successful technochange has three conditions. The first is a technochange solution that is capable of yielding the desired results if it is properly implemented. The second is that the solution is actually used effectively. The third is that the benefits of the solution are actively captured. Although these conditions are conceptually distinct, in practice it may be difficult to separate them, because certain features of the solution can support (or detract from) more than one condition.

A workable solution – completeness

For many years, organizations have invested heavily in IT, and for at least some of those years, people have wondered whether their investments have paid off. Early research reported a 'IT productivity paradox' – in which IT apparently showed up everywhere *except* in the productivity statistics.

Recently, however, enough studies have accumulated to make the picture clearer.[22] IT can, and often does, make significant contributions to business value. But in many cases (IT projects being a notable exception), the benefits are *only* realized when organizations reorganize work in new ways to take advantage of the capabilities of IT. The research also shows that, when organizations fail to make complementary changes, they often *lose* business value from their IT investments. If you automate a bad business process, you get a faster, more expensive, bad business process.

The additional changes required to make IT productive can be called *complementary changes*. Among the complementary changes that can be needed to transform IT into a complete technochange solution are the following:

- Changes in business processes and workflow
- New job designs

- New skills training
- Restructuring departments or business units
- Management changes
- Changing HR policies such as those concerned with hiring, performance evaluation, compensation
- New computerized or manual 'management systems' to monitor performance and support taking corrective action
- Redesigned spatial layouts
- Reallocated resources
- New metrics and incentives

For example, when companies implement ERP software with the objective of getting worldwide inventory visibility or putting 'one face to the customer', business process redesign, new measures and rewards systems, and changes in organization structures are sometimes necessary to complement the ERP software. SCM software may need to be complemented by changes in the frequency and type of scheduling, increased information sharing with partner organizations, changes in the nature of procurement practices, etc.

The AlliedSignal Aerospace CRM example is a good illustration of the need for technochange solution completeness and what it takes to make an IT solution complete. Initially, IT was envisioned as the solution to the problem of an uncoordinated sales process. CRM software was not a complete technochange solution, and without complementary changes, the software was not used. Redesigning processes and coaching sales managers were all that it took to complete the IT solution in that case, but other situations may require additional measures such as reallocation of sales accounts and new incentives to share sales leads.

Complementary changes are needed because IT alone is not enough to deliver the expected benefits of technochange. Only installing new IT in an established organization in the hope of changed organization behavior and performance is magic bullet thinking, and it leads to disappointment. Without supportive organizational changes, one of three negative outcomes is more likely to happen than achieving the desired results: the technology may not be adopted and used, the technology may be used in ways that reproduce old working patterns, or the technology may be used as expected without yielding the desired benefits. Although new IT can be a potent force for change, existing organizational conditions are often more powerful still. To ensure successful change in organizational performance, the good technochange solution complements new IT with supportive organizational changes.

A working solution – implementability

The second condition for a successful technochange is a solution that can be, and is, adopted and used. But many technochange solutions simply cannot be adopted and used easily or at all, because they conflict with existing organizational structures, cultures, or practices.

All technochange has the potential to provoke the human reaction often called 'resistance to change'. But not all technochange solutions are actually be resisted. And, given two technochange solutions that could accomplish the same objectives, one will often provoke much more resistance than the other.

Southern California Bizco provides a good example. The company was facing cost pressure. To the CIO, 'e-procurement' seemed like a good solution: across the many divisions of the company, six categories of purchases accounted for 40% of indirect spending (e.g., office supplies). By aggregating purchases and negotiating large volume discounts with suppliers, savings of 10–25% of the indirect spending budget could be realized. The CIO convinced the CEO to support a technochange with two key features. First, procurement contracting would be centralized at headquarters. (Purchasing against the contracts would still be done in the divisions.) Second, to ensure that the division purchasing staff did not engage in 'maverick spending', but instead made purchases against the new volume contracts, an expensive e-procurement system would be installed to monitor local purchases. The e-procurement system was justified, the CIO calculated, because every 3% increase in purchasing compliance with the volume contracts would drive $1.2 M to the company's bottom line.

The purchasing specialists rebelled, because they resented the elimination of the best part of their jobs – interacting with different vendors. The CEO sensibly decided that it was not worth the agony to implement the original solution over their opposition. (The potential payoffs looked good, but not good enough to justify intense organizational trauma.) Together with the CIO, he decided that the corporate volume purchasing contracts were still a good idea. But the e-procurement system was not necessary, because division personnel already had incentives to keep indirect spending low. The redesigned technochange solution was accepted by the divisions. Southern California Bizco captured most of the expected benefits *without* the completion of an expensive IT project. And the CIO learned a lesson about the dangers of relying on IT to change people's behavior.

The technochange idea at Southern California Bizco was initially complete: organizational restructuring, new procedure processes, and software to monitor compliance. But the e-procurement effort nearly failed because the complete solution conflicted so sharply with the existing organization. A slight scaling back of the radicalness of change reduced the resistance and allowed implementation to proceed. (In other instances, however, the way to eliminate resistance might be to make the solution more complete.)

For almost any organizational change goal, it is possible to design several complete technochange solutions that *could* accomplish the goal, *if the solution were adopted and used*. The challenge of successful technochange management is to design or select a complete solution that is *likely to be adopted and used.*[23] Such solutions are said to be *implementable.*

Designing technochange solutions for implementability is designing to avoid resistance to change (while still accomplishing organizational change goals). Therefore, it is important to understand where resistance comes from and what can be done to prevent it. (It is a much better idea to try to prevent resistance than to hope you will be successful in eliminating it *after* it arises.) One explanation[24] of resistance to change focuses on the degree of alignment or

Box 1 Technochange misfits – types and examples

Misfits are misalignments between a technology or a technochange solution and important dimensions of the organizational setting in which it is used. At least three types of misfits have the potential to cause technochange response failures.

- **Task or business process misfits.** A solution may be technically adequate but still not fit the ways people work in particular settings.
  - ERP systems designed for continuous production processes do not work well when applied in discrete part manufacturing plants; an ERP systems designed for discrete part manufacturing plants do not work well when used in the manufacture of products with dimensionality (e.g., the same shoe made in many colors and sizes).
  - Knowledge bases designed for use by experts often do not work well when they are made available to novices.
  - Systems that work well in one national context with particular business practices or legal frameworks do not work well in other countries with different norms and requirements.
  - Systems that require each repair item to be entered separately will not work well in a setting where workers are used to customers submitting repair items in large batches.

- **Cultural misfits.** A technically adequate solution may not fit particular settings for reasons that reflect organizational or national culture more than particular tasks. Organizational culture can be defined as ‘the way we do things around here’. It often reflects what has been successful in the past. Apparently ‘arbitrary’ differences between technochanges and organizational culture can create friction and contribute to ‘resistance’, as can misfits associated with certain aspects of national culture.
  - Systems aimed at increasing administrative efficiency are often resented by doctors and nurses who are committed to patient care.
  - Systems designed to promote teamwork may be rejected by people who typically work alone.
  - A geographical information system was not accepted in India, a country that does not have ‘a map culture’.
  - ERP systems are not much used in countries like China, in which managers do not trust their subordinates with access to organizational data.

- **Incentive misfits.** Technically adequate solutions may be misaligned with the authority and reward systems of an organization. These misfits are sometimes called ‘political’ misfits, because change managers who promote technochanges with incentive alignments often run into lots of destructive organizational politics.
  - When Lotus Notes was first introduced into a consulting firm to promote knowledge sharing, it was not used that way. The reason was that, in the company’s up-or-out promotion system, consultants got ahead by hording what they knew. The system only gained a foothold when the firm began making promotion decision in part on the basis of consultants’ contributions to the knowledgebase.
  - Corporate accountants introduced a new financial system to give them detailed visibility into what was happening in the divisions. The system was strenuously opposed by division managers who wanted to avoid headquarters interference.
  - A product configuration system was never used by sales people, in part because they were not measured on or rewarded for what the system helped them do: ensure configuration quality.

Situations involving task or business process misfits, cultural misfits, or incentive misalignments cannot be successfully dealt with by focusing on *technical adequacy* (IT functionality, ease of use and learning, reliability, availability of good technical and support infrastructures). Unfortunately, misfits often have the appearance of technical *inadequacy*, and those who experience misfits often claim that the system is inadequate. This can prompt technochange managers to waste resources on actions such as redeveloping the system or upgrading infrastructure, which will not resolve the misfit problem, because it has different causes. Therefore, it is important to take potential misfits carefully into account when designing technochange solutions and when dealing with apparent cases of resistance to technochange.

fit between the technochange solution and the existing organization. There are many ways in which technochanges can create ‘misfits’ when introduced into an existing organization. Three main types of misfits are task or business process misfits, cultural misfits, and incentive misfits (see Box 1). Which type of misfit is most important to avoid will vary from setting to setting, but incentive misfits are most likely to be ‘showstoppers’.

An example of an incentive misfit involves Config,[25] an expert system intended for use by computer sales personnel, with the objective of increasing computer system configuration accuracy. When the sales representatives (reps) did not use Config, the system was expensively redesigned to increase usability and relaunched with training for all the sales reps. Use of Config still did not increase. The problem was that sales reps (and their managers) were not measured or rewarded for configuration accuracy – that was not a goal that mattered to them. This might not have been a problem had the system provided benefits directly to the sales reps. But despite the costly redesign, the system was still very difficult for the sales reps to use, because it was not integrated with the work processes and software they had to use.

A worked solution – appropriation of benefits

The third condition for successful technochange is benefit capture. 'Capture' might seem like an odd word to use here. But the benefits of technochange, unlike those of IT projects, are not automatic. Effort is required to turn *potential* benefits into measurable organizational results.

Let us consider some examples. Say that the major benefit of a technochange is that it enables accountants to do their work in less time. How does the organization benefit? If the accountants have been completing their work in unpaid overtime, there is no *financial* benefit to the organization (just happier accountants). If the accountants can now do their work in 6 h instead of 8, does the organization benefit financially? To realize financial benefits, managers would have to reassign work and reduce the number of accountants. Would they actually do so? There are lots of reasons not to (unhappy accountants!). To make sure that managers actually capture these benefits, the managers must have *incentives* (such as rewards) to reduce their costs actively. If existing management practices do not provide rewards for cost reduction, achieving the benefits of this technochange will require incentive systems to be changed.

As discussed above, it is sometimes possible to 'build in' benefit capture by making changes in measures and rewards part of a complete technochange solution. In other cases, however, negotiations about how the costs and benefits of a technochange solution are to be shared must be conducted as part of the implementation process.

For example, years ago, managers at Frito-Lay envisioned the potential marketing benefits of large-scale data mining.[26] The problem was that they did not have data in enough detail to support the desired analyses. Asking field sales personnel to collect more data was impractical – they already spent hours of their own time each week doing paperwork. Handheld devices for data capture is an obvious solution today, but at that time, the technology hardly existed: Frito-Lay had to work with a startup vendor to develop the technology. Obviously, this was a very expensive proposition, so Frito-Lay had to be sure there would be organizational benefits. The benefits *to field sales personnel* were immediate and obvious – less time doing paperwork, more accurate orders. But the benefits *to the organization* had to be captured. So Frito-Lay asked the field sales organization to pay part of the cost. District sales managers agreed to (and did) reduce costs or increase sales over a period of time[27] in return for the technology. As a result, Frito-Lay got benefits from this project well before they were able to implement their data-mining concepts.

Another example concerns Pellton International, a multinational chemical company that sought more efficient supply relationships with a key customer.[28] The technochange solution was for the customer to agree to order a reduced number of product variations (reducing Pellton's production and distribution costs) and for Pellton to manage the customer's inventory on a consignment basis, delivering products on an as-needed basis (reducing the customer's inventory carrying costs). It sounds like a win–win solution, but it did not start out that way. When the technochange was initiated, Pellton failed to negotiate a benefit-sharing agreement with the customer, possibly because Pellton's supply chain experts thought their benefits would be automatic. The benefits *were* automatic – but only for the customer. Once Pellton took over managing the customer's inventory, the customer's costs decreased. But the customer had no motivation to change its ordering practices, so Pellton did not get its projected benefits. Only when Pellton threatened to pull out of the arrangement did the customer see the need for, and agree to, changed ordering practices that lowered Pellton's costs.

Synthesis

The demands of successful technochange are significantly greater than those of successful IT projects. Successful IT projects have only two major requirements: first, the technology (and its support system) must work acceptably well – a condition that often requires vendors and other outsiders to perform as they promised. Second, the people tasked with getting the technology up and running have to manage the project to a schedule and budget, because otherwise the solution will not pay off as expected. Satisfying these conditions is challenging. But, successful technochange has additional requirements. Successful technochange requires a good (complete and aligned) solution that is actually used and actively managed for benefits.

These conditions for successful technochange can be satisfied partly through the solution's features and partly through the process of solution design and implementation. What this means is that no one technochange solution or implementation process can be right for every situation. When a technochange effort runs into difficulties, either the solution or the implementation process may be at fault, and opportunities for fixing the problems may lie in either. It also means that it is very difficult to get technochange right the first time. Neither design nor implementation is an exact science. However, as technochange project cycles get longer, the need to get the details right the first time escalates. (Managers' tolerance for problems decreases as project length increases, and big failures damage organizations' capacity for future changes.) Unfortunately, the longer the projects go on, the less likely it is that the details *will* be right the first time, because of the life cycle dynamics of time and distance effects.

The only way out of this trap is not to get in it. The next section describes an alternative strategy in which the traditional sequential technochange life cycle is replaced with a sequence of several (or even many) small, iterative design and implementation cycles, each designed as an learning experience or as an opportunity to capture specific business benefits.

Characteristics of a good technochange process

The technochange process encompasses the activities of idea generation, solution design, solution implementation, and benefit capture. This process can be structured as a sequence of big phases with no iteration – the typical life cycle presented earlier. Or it can be structured as an iterative series of many smaller cycles – a process often called 'prototyping'.

Prototyping approaches have evolved in both the technical disciplines and the organizational change literature as a way to cope with the limits on designers' foresight. A prototype is a model or an example for purposes of demonstration. The prototyping approach can be loosely understood as trying something and using the results as a basis for deciding what to do next – in a cycle some describe as 'plan, do, check, act' (or 'design, implement, evaluate, correct').

In the IS field, for example, transaction processing systems (e.g., accounting, inventory control, etc.) have long been developed by a 'waterfall' method in which system specifications are fully documented and 'frozen' before system building begins. But the challenges of specifying in advance how managers would use systems in unstructured decision-making processes led to the evolution of a new system development approach (prototyping) as well as to a new system type (decision support system). In decision-support system prototyping, developers rapidly build a few simple capabilities based on user input, then work closely with users, observing them as they apply the system to real business tasks. Observations lead to suggestions for enhancements and refinements, and the process iterates. In so doing, the design specifications and the solution evolve in tandem.[29]

Similarly, some organizational change experts have also advocated prototyping approaches to organizational change (under different labels). For example, pilot projects are often used to test organization redesign ideas in one or two locations before attempting to transfer them to the rest of the organization. Another approach to large-scale change encourages organizations to 'let many flowers bloom' and then attempts to grow the most promising innovations throughout the organization.[30] In the 'breakthrough strategy', managers are encouraged to set achievable, but challenging, result targets and to demand that employees meet them within a short, defined time frame without new resources.[31] Early successes are then expanded via additional experiments.

Technochange prototyping

The prototyping approach can also be adapted to technochange. Here what is to be prototyped is not just a technical solution *or* just an organizational change, but both together. For example, one expert describes an 'improvisational approach' to IT-enabled organizational change, in which organizations iteratively capitalize on the planned and unplanned organizational consequences of IT implementations.[32] Other experts advocate an 'incremental approach' to packaged software implementation,[33] which bears much resemblance to organizational prototyping.

'Results-driven incrementalism' divides potentially massive technochange efforts with major performance improvement goals into a series of short projects (2–3 months long), each of which is designed to deliver measurable organizational benefits.[34] This approach should be distinguished from simple phased IT implementations, in which there are project milestones and sequences of deliverables, but business benefits are not expected until the whole project is complete. In the results-driven incremental approach, the organization can achieve business benefits from early phases even if the process is terminated prior to the originally envisioned end.

There are numerous possible variations on the technochange prototyping idea. All focus on achieving behavior change and organizational results, rather than just a successful IT project. An example concerns the building of an information system designed to support the activity of organizational design.[35] The process of organization design occurs in all organizations, but no particular job category lays sole claim to this activity. Although a great deal of academic knowledge exists about how to perform organizational design, experts disagree on many points, and novices often do not believe they lack expertise. Requirements analysis revealed that, although people might use IT support for organization design, they would not accept training in how to perform the process or use the tool. In short, if IT were to be successful in improving the process of organizational change, the tool itself would have to motivate users to learn what they needed to know to perform the process well.

Amazingly, the developers of the organizational design support tool succeeded in their objectives. They developed a tool that, once users began to try it, encouraged users to perform complete socio-technical analyses of their workplaces. Furthermore, evaluation research showed that using the system changed users' *off-line* behavior. Users acted on what they learned from their analyses, by communicating with others in their organizations, by collecting additional data, and by implementing designs suggested by their analyses.

What did the designers do to achieve this success? Their radical prototyping approach involved over *70* design-build iterations in 18 months. In many of these cycles, the behavior of users was observed, not only when they were working with the tool, but also afterwards as they went about their work. When users demonstrated the desired off-line behavior change, the designers knew that they had accomplished their goal.

The pros and cons of technochange prototyping

Prototyping approaches of all kinds tend to encounter resistance – usually from those trying to bring about the change, rather than from the change targets.[36] Among their many objections is that a prototyping approach is much more time-consuming than a traditional 'design first then implement' approach, at least when the traditional approach goes according to plan. The reason is that prototyping requires much more involvement of 'users', who have to review and test the solution at every step of the way. Developers often have difficulty commanding the attention of users and fear that their lack of participation will jeopardize project completion.

Unfortunately, traditional large-scale 'design then implement' approaches, such as the traditional IT project life cycle, rarely work out exactly as expected. Much more frequently, the problems experienced and exported are so numerous or consequential, that large, complex IT projects founder. By contrast, technochange prototyping can achieve significant results in smaller increments, minimizing the risks of failure. And, as these results are achieved, people in organizations can grow in their *motivation and capability to undertake additional change.*

The traditional sequential technochange life cycle reflects the belief that a straightforward 'recipe' (a predefined sequence of steps with no iterations or improvizations) will yield significant organizational change. This is yet another version of 'magic bullet thinking'; I call it 'the myth of the methodology'. In some areas of human effort, good recipes and methodologies do sometimes produce nearly foolproof results. Examples include cooking and software coding. But in other areas – IT-driven organizational change being a prime example – methodologies codify only one important kind of knowledge. Methodologies designed by IT and organizational change experts represent attempts to avoid the major problems that, on a *statistical* basis, bring projects and change programs down. But they are not sensitive to the *unique* risk factors and opportunities in particular situations. Consequently, they fail to provide the diagnostic and intervention tools that technochange managers need to avoid and work their way out of difficult, situation-specific spots and lull technochange managers into a false sense of security based on the incorrect notion that such tools are not needed.

In medicine, epidemiology and clinical treatment are understood as two separate ways of knowing. Knowledge about the statistical incidence of diseases and risk factors is useful in clinical treatment, but it cannot substitute for the ability to understand the patient as a whole. Similarly, knowing how technochange is supposed to be done in the abstract is only part of what it takes to make technochange succeed in particular organizational settings.

When the 'myth of the methodology' is taken into account, technochange prototyping appears much less risky than the traditional technochange lifecycle. It has the added advantage of being able to reveal unintended negative consequences early enough to allow for corrective action. However, in light of the fact that technochange prototyping creates its own anxieties, it is useful to consider when that approach is most needed and effective.

In general, the more closely a change initiative resembles technochange and the less it resembles an IT project, the more beneficial the technochange prototyping approach is likely to be. The sharp distinction between IT projects and technochange initiatives drawn earlier in this article is a bit simplistic. It is probably better to think of IT-driven organizational efforts in terms of a continuum. A change effort is better approached with technochange prototyping than with the traditional IT project life cycle when:

- It affects people outside the IS department.
- It affects people outside the organization.
- It affects more, rather than fewer, people, occupational groups (e.g., accountants, managers, front-line employees), organizational units, organizations, etc.
- It has larger effects (e.g., people have to learn new tasks and skills) rather than smaller ones (e.g., people have to learn new software, but their jobs do not otherwise change).
- It affects some people in ways that are likely to be perceived as negative (e.g., eliminates jobs; removes activities from jobs; reallocates key resources like personnel, projects, or accounts; changes the rules by which performance is evaluated and rewarded; restructures organizational units; changes relationships between supervisors and subordinates or between organizational units, etc.).
- It is very expensive, is projected to take a long time, and has the potential to disrupt organizational performance significantly during startup.
- It is revolutionary *vs* evolutionary.

In addition, the technochange prototyping approach is better used when:

- The organization is believed to be highly resistant to change. An organizational culture that supports experimentation and learning is highly favorable to the success of technochange,[37] no matter which approach is used. But if the organization is very conservative, the prototyping approach can help build the capacity to change incrementally, whereas the traditional life cycle approach is likely to overwhelm it.
- The technology or its application in the organization is unproven. The traditional life cycle approach invests too much upfront before results are known for this to be a good strategy with unproven technology.
- The technology is not monolithic or 'indivisible'.[38] Some units of IT functionality cannot easily be subdivided to support an incremental implementation approach – an example is a module of an ERP system. This characteristic inhibits use of the technochange prototyping approach.
- The organizational change is trialable, at least in certain locations. The prototyping approach is less feasible in situations where the change must be implemented everywhere at once and is irreversible. An example of an indivisible change is reorganizing a company to provide 'one face to the customer'.

If the technochange prototyping is *not* used where these conditions exist, it is all the more important to pay attention to the quality of the technochange solution. Technochange managers should analyze organizational processes, organizational culture, and organizational incentive systems using techniques outlined in the business process reengineering and organizational change literatures and attempt to design solutions that are complete, aligned, and enabled for benefit capture. Accomplishing this goal probably requires new types of partnership between IT specialists, organizational managers, and human resource management specialists during the solution design process.

Synthesis

Many technochange problems can be traced to the traditional sequential process by which IT solutions are usually developed and implemented in organizations. An alternative approach, in which complete technochange solutions are implemented incrementally in much smaller steps, can sharply cut the risks of technochange. However, technochange prototyping requires major organizational change in the way IT work is done, and consequently it is as likely to be resisted by managers and IT specialists as technochange is likely to be resisted by users. Nevertheless, the risks of unsuccessful technochange are so great that efforts to overcome resistance to technochange prototyping are definitely worthwhile.

Concluding remarks

Deliberate technochange is the use of IT to drive improvements in organizational performance. As technochange involves both IT and organizational changes, it differs both from IT projects and from organizational change programs. Attempting to manage technochange as an IT project or as an IT project combined with a traditional organizational change program is not a winning game. What is needed for successful technochange is an *integrated* technical and organizational solution.

Successful technochange is characterized by completeness (IT complemented with relevant organizational changes), alignment ('fit' between the technochange solution and organizational processes, culture, and incentives), and division of benefit arrangements that enable benefit capture. In achieving a successful outcome, both the features of the technochange solution and the technochange design and implementation process are important: either can create technochange problems and either can offer opportunities for avoiding or fixing problems. No step-by-step methodology will always produce successful technochange outcomes: technochange prototyping, an approach that offers advantages compared to the traditional sequential IT implementation lifecycle, is less an exact science than a lower risk 'try it and see' approach. Successful technochange involves a balancing act between careful upfront design and evolutionary implementation.

Achieving significant improvements in organizational functioning and performance is difficult, with or without IT. But it is not impossible. Better solution design skills and better implementation processes can go a long way toward reducing the risks of technochange failure and increasing the benefits possible from technochange success.

Acknowledgements

The author gratefully acknowledges the helpful comments of Ann Majchrzak, Chris Sauer, Peter Seddon, Mark Silver, and Alfred Waller on an earlier version of this paper.

Notes

1 Markus, Axline, Petrie and Tanis, Learning From Adopters' Experiences with ERP – Successes and probems,' *Journal of Information Technology* 15(4) (December 2000), 245–265.

2 Markus and Tanis, The Enterprise Systems Experience – From adoption to success, in Zmud (ed.) *Framing the Domains of IT Research: Glimpsing the Future Through the Past*, Cincinnati, OH: Pinnaflex Educational Resources, 2000.

3 Majchrzak, Management of Technological and Organizational Change, in Salvendy (ed.) *Handbook of Industrial Engineering*, New York, NY: John Wiley and Sons, 1991, pp. 767–797.

4 Check out what the Project Management Institute has to say about the processes of project management. 'Project communication' hardly scratches the surface of what an organization change expert would think is needed.

5 Pick up a sample of organizational behavior and change textbooks and see for yourself.

6 Laartz, Monnoyer and Scherdin, Designing IT For Business, in *McKinsey Quarterly*, 2003, pp. 76–87.

7 Levinson, Cleared for Takeoff, in *CIO Magazine*, 2002, p. online http:www.cio.com.

8 Markus and I, The Magic Bullet Theory In IT-Enabled Transformation, *Sloan Management Review* (Winter) 1997, 55–68.

9 Brynjolfsson and Hitt, Beyond Computation: Information technology, organizational transformation and business performance, *Journal of Economic Perspectives* 14(4) (Fall 2000) 23–48.

10 IT projects, technochange efforts, and change management programs also differ in their basic methodologies, the roles of the organization's managers, IT specialists, and other specialists, and in key success factors (see Table 1.)

11 Bashein and Markus, *Data Warehouses: More Than Just Mining*, Morristown, NJ: Financial Executives Research Foundation, Inc., 2000.

12 Harrison, Conner and Horney, *Project Change Management: Applying Change Management to Improvement Projects*, New York: McGraw-Hill, 2000.

13 Mao and Markus, Revisiting User Participation – A new look at an old problem, Bentley College, Waltham, MA, working paper available from the author.

14 Stinchcombe, *Information and Organizations*, Berkeley, CA: University of California Press, 1990.

15 See Note 2.

16 Markus, Majchrzak and Grasser, A Design Theory For Systems That Support Emergent Knowledge Processes, *MIS Quarterly* 26(3) (September 2002) 179–213.

17 See Note 2.

18 Bass, Cigna's Self-Inflicted Wounds, in *CIO Magazine*, 2003, online at www.cio.com.

19 Bulkeley, A Cautionary Network Tale: Fox–Meyer's high-tech gamble, in *Wall Street Journal Interactive Edition*, 1996.

20 Bashein, Markus and Finley, *Safety Nets: Secrets of Effective Information Technology Controls*, Morristown, NJ: Financial Executives Research Foundation, Inc., 1997.

21 Markus and Robey, Why Stuff Happens: Explaining the unintended consequences of using information technology, in Vendelo and Andersen (eds.) *A Tribute to Niels Bjorn-Andersen*, London: Butterworth-Heinemann, forthcoming.

22 Brynjolfsson and Hitt, Beyond Computation: Information technology, organizational transformation and business performance, *Journal of Economic Perspectives* 14(4) (Fall 2000) 23–48.

23 Markus and Keil, If We Build It They Will Come: Designing Information Systems That Users Want To Use, *Sloan Management Review* (Summer) 1994, pp 11–25.

24 This might be called the product theory of resistance. There are at least two other major theories of resistance to change: a theory that focuses on people and one that focuses on the change process. Although savvy change managers know and can apply all three, the other two theories are beyond the scope of this article.

25 See Note 23.

26 Applegate, Frito-Lay, Inc.: A Strategic Transition (Consolidated), Boston, MA: Harvard Business School.

27 By spreading out the payments over time, Frito-Lay avoided the shakedown trauma experienced by Cigna.

28 Corbett, Blackburn and Van Wassenhove, Case Study Parnterships To Improve Supply Chains, *Sloan Management Review* (Summer) 1999, pp 71–82.

29 See Note 16.

30 Kanter, Stein and Jick, *The Challenge of Organizational Change: How companies experience it and leaders guide it*, New York: The Free Press, 1992.

31 Schaffer, *The Breakthrough Strategy: Using short-term successes to build the high performance organization*, New York: Ballinger Publishing Co., 1988.

32 Orlikowski and Hofman, An Improvisational Model for Change Management: The case of groupware technologies, *Sloan Management Review* (Winter) 1997, pp 11–21.

33 Fichman and Moses, An Incremental Process for Software Implementation, *Sloan Management Review* (Winter) 1998, pp 39–52.

34 *Ibid.*

35 See Note 16.

36 See Note 33.

37 See Note 32.

38 See Note 33.

About the Author

M. Lynne Markus is Trustee Professor, Department of Management, Bentley College. Professor Markus's research interests include the management of IT-enabled change, electronic business, enterprise systems integration, and knowledge management. She is the author of three books and numerous articles in journals such as *MIS Quarterly, Information systems Research, Organization Science, Communications of the ACM, Sloan Management Review and Management Science*. She holds a B.S. in Industrial Engineering from the University of Pittsburgh and a Ph.D. in Organizational Behavior from Case Western Reserve University.