Abstract—A key issue in enterprise resource planning (ERP) implementation is how to find a match between the ERP system and an organization’s business processes by appropriately customizing both the system and the organization. In this paper, we advance a framework for supporting management decision-making about customization choices and the capabilities required to accomplish them. In this framework, we identify various customization possibilities for business processes as well as ERP systems. We also identify technical and process change capabilities required for system and process customization. Combining customization options with change capabilities, we present a useful way for managers to identify feasible customization options for their particular organization. Such a framework also helps managers to recognize the gap between desired customization options and change capabilities. A case study is used to illustrate the application of the framework.

Index Terms—Case study, change capability, enterprise resource planning (ERP) systems, process customization, system customization.

I. INTRODUCTION

E NTERPRISE Resource Planning (ERP) software is one of the fastest growing segments of business computing today. According to a report by Advanced Manufacturing Research, the ERP software market is expected to grow from $21 billion in 2002 to $31 billion in 2006 and the entire enterprise applications market which includes Customer Relationship Management and Supply Chain Management software will top $70 billion [1]. The reason behind this phenomenal growth is the promise that ERP systems can provide an integrated business computing solution and improve a company’s ability to compete in the marketplace. Often cited benefits of ERP systems include the integration of data and applications, replacement of old, fragmented legacy systems, cost advantages and quicker deployment of packaged systems as compared with in-house development, and the adoption of best practices in organizational processes [2], [3].

For individual companies, however, the implementation of ERP systems presents the greatest challenge for many MIS managers today [4]–[8]. Reports of ERP implementation failures are common. Reasons for failures include spending more money on ERP than the company can afford, being incompatible with strategic partners, conflicting with its management style and structure, being overwhelmed by the required organizational changes to fit the system, and dealing with ever changing ERP technology and its infrastructure [9]–[11]. Dow Chemical, for example, spent seven years and close to a half of a billion dollars in implementing a mainframe ERP system and then realized that a client-server architecture would be more appropriate [12].

Any successful ERP implementation requires a fit between the ERP system and the organizational processes it supports [13], [14]. An important characteristic of ERP systems is that they are packaged software solutions rather than customized systems. As such, they come with built-in assumptions and procedures about organizations’ business processes. These assumptions and procedures seldom match exactly with those of the implementing organization’s existing processes [15]. In traditional information systems development, the computer system is usually designed to fit the organization and its processes. With packaged software solutions such as ERP systems, it is difficult to completely mold the system to fit existing business processes. In fact, many researchers and practitioners have suggested that it is easier and less costly to mold business processes to ERP systems rather than vice versa [12], [13]. Thus, even for those companies that have successfully implemented large-scale information systems projects in the past, ERP implementation still presents a challenge, because it is not simply a large-scale software deployment exercise. ERP implementation is often accompanied by large-scale organizational changes [14], [16]. Consequently, a key issue in ERP implementation is how to find a match between the ERP system and an organization’s business processes by appropriately customizing both the system and the organization.

In this paper, we advance a framework for supporting management decision-making about customization choices and the capabilities required to accomplish them. In this framework, we identify various customization possibilities for business processes, as well as ERP systems. We also identify technical capabilities required for technical ERP customization options and process change capabilities needed for process customization. Combining the process and technical customization options with technical and process change capabilities, we present a useful way for managers to identify feasible customization options for their particular organization. Such a framework also helps managers to develop a long-term view of ERP implementation by suggesting that ERP implementation be viewed as a series of interdependent customization and implementation projects.
TABLE I
LITERATURE ON ERP IMPLEMENTATION ISSUES

<table>
<thead>
<tr>
<th>ERP characteristics</th>
<th>Implementation issues</th>
</tr>
</thead>
</table>
| Packaged software   | -ERP software is developed for a market not for an organization [21] [22] [23]  
                      | -User participation is limited during the development of ERP software [12] [24]  
                      | -*Industry best practices* embedded in ERP is not universal [27] |
| Complexity          | -ERP implementation requires the integration of data, processes, and operations throughout the enterprise [7] |
| System Adaptation   | -ERP systems need to be changed to fit existing or reengineered business processes [12] [15] [33] |
| Organization Adaptation | -Organizations need to be changed to fit the ERP system [4] [14] |

II. RELATED LITERATURE

A large body of literature on information technology (IT) implementations has been developed during the past several decades [17]–[19]. However, our understanding of the factors and processes that lead to ERP implementation successes or failures is still limited because ERP implementation is relatively new and is different from traditional information systems development projects [11], [20]. Here, we focus on some of the unique characteristics of ERP systems and review the implementation issues related to these characteristics. A summary is provided in Table I.

One major difference lies with the fact that ERP systems are packaged software. Carmel and Sawyer [21] compared the development processes of packaged software with traditional information systems at the industry, development, cultural milieu, and team levels. Their analysis suggests that vendors of packaged software have to satisfy many customers with varying needs and requirements in order to capture the necessary market share and profit to justify their investment. At the same time, there is tremendous time-to-market pressure on the vendors to come out with their new products to stay ahead of their competitors. In addition, packaged software developers are separated from the users organizationally, as well as physically and they usually do not participate in the implementation of the software package. Intermediaries such as sales and customer support staff and third party consultants often provide the linkage between software developers and users [22]–[24]. Consequently, unlike traditional software development projects where a system is tailor-made to suit the existing business requirements and processes, packaged software implementation involves the users changing procedures and business processes in order to use the package, changing some of the programs in the package to fit their unique requirements, and relying on package vendors for assistance and updates to the software package [25].

The disconnect between an organization’s information and process requirements and the solutions provided by ERP is especially pronounced due to the complex nature of the ERP systems [26]. In their study of ERP implementations in Singapore, Soh et al. [27] illustrates that the so-called “industry best practices” embedded in ERP systems is hardly universal. They suggest that the misfits between business requirements and ERP capabilities can be company-specific, industry-specific, or country-specific and may be classified into three categories: data, process, and output. Davison [28] highlighted the cultural implications of misfits in ERP implementations.

Another difference between ERP and traditional information system development project is that ERP projects, which tend to be enterprise-wide, are typically larger in scope than traditional software development projects, which often focus on one or more departments or business processes. The risks associated with ERP projects are relatively higher than those traditional projects [29]. Enterprise-wide projects affect many more users and these users may have different and possibly conflicting needs and requirements. In addition, ERP projects require the integration of data, processes, and operations throughout the enterprise, often on a global scale, which makes ERP projects far more complex than traditional systems development projects [7]. To achieve integration, dissimilar components and independent business processes need to be fitted together by conforming to a uniform standard. For example, integration may be accomplished by changing each business process to fit with the system using the ERP system as a standard.

The fit between business processes and ERP systems and among business processes is believed to be critical to the success of ERP implementation [3], [13], [30], [31]. Hong and Kim [31] assessed the impact of data, process, and user fit between ERP system and organizational requirements on implementation success. They found a positive correlation between the initial organizational fit and implementation success. However, for most organizations, such a fit can only be achieved through mutual adaptation of the ERP systems and organization processes [32].

The adaptation of the ERP systems involves the customization of the ERP system to fit existing or reengineered business processes. Davenport [12] identified ERP system customization as module selection, table configuration, and code modification. Similarly, Glass [33] classified ERP system customization into configuration, extension, and modification. Brehm et al. [15] provided a more detailed categorization of ERP system customization, which include nine types of adaptation: configuration, bolt-ons, screen masks, extended reporting, workflow programming, user exits, ERP programming, interface development, and package code modification. Each type may require changes to be made at different layers of the ERP system. Consequently, such changes may affect the initial ERP implementation, as well as future maintenance, upgrade, and
Conversions. It is worth noting that these frameworks are only concerned with the change of systems to fit business processes.

An important stream of ERP implementation research focuses on the adaptation of the organization to ERP systems. This research explores the process of ERP implementation and its associated organizational changes from initiation to development to completion over time [4], [14]. ERP implementation is viewed as a long and complex process with successes and failures at different stages of the implementation. In a study of ERP implementation in 15 different companies, Ross [34] discovered that most companies go through the following five stages in their implementation process: design, implementation, stabilization, continuous improvement, and transformation. Markus and Tanis [3] describes a typical ERP implementation in terms of four phases: the chartering phase, the project phase, the shakedown phase, and the onward and upward phase. Koh et al. [35] applied this process theory to identify problems and issues of ERP implementation using a case study. Stages models of IT implementation have also been used in analyzing ERP implementations [13], [36]. Although these models depict the ERP implementation process differently, they share many similar activities and decisions that need to be made by managers. One of the key decisions in the early stage of the implementation process is whether to accept the assumptions about business processes built into the system [34]. This decision affects the amount of customization needed to the software, as well as to the organization.

The systems development life cycle methodology needs to be modified for the unique characteristics of ERP implementation [27], [37]. Implementation in the traditional SDLC refers to the later stage of system development in which a completed system is installed, deployed, and placed into operation or production. The stage also includes the conversion to the new system, the training of users, and final documentation of the system [38]. ERP implementation involves the understanding of existing business processes and ERP technologies, the customization of business processes, and ERP modules and tables to fit each other, and the management of large-scale business process change and system integration projects. Hence, analysis phase activities (e.g., understanding of critical organization processes) and design phase activities (e.g., knowledge of the ERP software) of SDLC have to be merged for ERP implementation [27]. Brehm and Markus [37] stressed the importance of interaction between vendors and adopters during the ERP implementation.

From a resource-based perspective, the success of ERP implementation is affected by the kind of IT-based resources possessed by an organization and how they are assembled, coordinated, and deployed [39]–[42]. IT-based resources can be classified into tangible IT resources (e.g., IT infrastructure), intangible IT resources (e.g., knowledge bases), and human IT resources (e.g., technical and managerial IT skills) [42], [43]. The ability to assemble, coordinate, and deploy IT-based resources in combination with other resources is referred to as an organization’s IT capability. Feeny and Willcocks [41] included the following as a set of core IT capabilities: IS/IT leadership, business system thinking, relationship building, architecture planning, making technology work, informed buying, contract facilitation, contract monitoring, and vendor development. Organizations have access to different IT resources and, thus, possess different IT capabilities. Such differences may explain the divergences among organizations in the use of IT and in the benefits they have gained from the usage [39]. They may also one of the major reasons why organizations choose different ERP customization options during ERP implementation.

III. CUSTOMIZATION IN ERP IMPLEMENTATION

The primary goal of customization in ERP implementation is to achieve a fit between the ERP system and the process that the system supports. Thus, both the system and the process can be changed or customized to achieve the goal. When the system is customized to fit the process, we refer to this kind of customization as technical customization. Similarly, when a process is customized to fit the system, we refer it as process customization.

A. Technical Customization

When installing an ERP system, companies have many choices about how to change and customize the software package [37]. Most ERP software packages are developed in a modular approach, where each module contains specific functionality and configurable options [44]. In addition, an open or proprietary programming environment is often provided by ERP vendors to their customers for modifying the system. Thus, companies have three types of technical customization options: module selection, table configuration, and code modification [12]. In module selection, companies choose to implement one or more modules using the default configuration set by ERP vendors. In this case, technical customization is achieved through the company’s decision as to which modules to implement. This type of technical customization makes minimum alterations to the system and, by itself, is rarely sufficient in an ERP implementation. Some small companies, however, may take this low cost and low risk approach. For example, Thermacore decided to implement SAP’s accounting module using the default charts of accounts without any changes [45].

Since most ERP systems are table-driven, another type of technical customization is to select configuration options in the tables so that the system fits organizational needs. A key requirement for table configuration is to understand the meaning and consequences of each configurable option in each table. Since there are numerous tables in a typical ERP system, this can be a very complex and time-consuming task, especially when interdependencies among options across various tables and modules need to be considered. For example, Dell Computer spent more than a year on table configuration alone [12]. The benefits of this type of technical customization include the ability to tailor the system without coding, the full support from the vendor, and the ease of future upgrades.

The third type of technical customization is code customization, where the source code of the ERP system is changed, the functionality is augmented, or a new interface is developed to allow the ERP system to interact with other systems [46], [47]. Some ERP systems come with a proprietary programming environment to support code customization; others can
interface with higher level programming languages such as C++. Code customization provides companies the greatest flexibility in adapting the system to organizational needs and allows companies to integrate the ERP system with any existing production systems. On the other hand, it also presents the highest risks and costs in technical customization. It is quite expensive to obtain competent technical staff or consultants to perform code customization and there are risks of failures and budget overruns. Also, too much code customization leads to incompatibility with newer versions of the system and, thus, difficulties in future upgrades. Furthermore, certain integration benefits built into the original ERP design may be lost as a result of code customization. Clearly, as we move from module customization to code customization, the costs and risks increase; the benefits, however, may or may not increase.

ERP vendors have a rather different view of technical customization than the view of adopting organizations [46]. While most vendors provide the above programming environment for supporting adopting organizations, they consider technical customization as an evolving process where they continuously add modules, extend configuration tables, and improve tools for code modification to meet the needs of adopting organizations. Their goal is to reduce the degree, costs, and risks of customization so that adopting organizations can restrict their customization efforts to module selection and table configuration.

B. Process Customization

Fit can also be achieved by changing the process rather than the system. Process customization is the degree to which the business process is changed to fit the system. A business process is defined as a set of logically related tasks that use the resources of an organization to achieve a defined business outcome [48]. This definition indicates that a business process consists of tasks, resources, the outcome, relationships among tasks, and relationships among resources and tasks. Based on the changes made to these elements, we classify process customization into three categories: no change, incremental change, and radical change.

In the case of no change, process customization involves only changes in tasks and resources, but no changes in relationships among tasks and configurations of resources. An example of such process customization is task automation in which computer technologies are substituted for manual labor. Then, the resources used to accomplish the task have been switched from manual labor to computers but the other elements of the business process remain the same.

The second category of process customization is incremental change in which improvements are made not only in tasks and resources, but also in relationships among tasks and relationships among tasks and resources. The nature of the process and its outcome measures, however, has not changed. The focus of the change is solving problems found in the process. Most total quality management (TQM) initiatives are examples of incremental change [49].

Radical change is the third category of process customization. It involves the fundamental rethinking and radical redesign of the elements in a business process, including the measures of performance. Literature on Business Process Reengineering provides numerous examples of radical change. Organizations that are converting from a functional view of business to a process view often make radical changes to their business processes as well [49].

These three categories of process change represent increasing degrees of process customization on a continuum from no change to radical change. How much change is needed depends on how well the new technology fits with the existing process. Management has the choice of changing the process to fit the system and vice versa.

C. ERP Customization Choices

With technical customization and process customization as dimensions, we derive a table for describing various ERP customization choices, as illustrated in Table II.

Each cell in the table represents a possible choice for achieving fit between the process and the ERP system. Each row shows the process change options given a chosen system configuration option. Each column identifies the system configuration options to fit a particular process change decision. For example, the first row shows the three available options if managers decide to adopt the process embedded in the system and make any necessary changes in the business process to fit the system. In this case, the system process is often regarded as the best practice or is deemed as the ideal process for the organization.

The no customization cell is a valid choice when there is already a good fit between the existing business process and the system process. Thus, no process customization is necessary and a good fit can be achieved by selecting appropriate system modules. Although rare, it is still possible when, for example, an organization’s process was used as a model for developing the system. This cell involves the lowest implementation risk among all cells provided that the assessment of fit is accurate. Managers should ensure that the perceived fit between the existing process and the embedded system process is not just their wishful thinking. The cell process adaptation involves making moderate process changes to fit system modules. It assumes that the existing business process is similar to the system process and incremental changes are sufficient to achieve fit. The risk associated with this cell is primarily related to the organization’s ability in improving the existing business process. If the existing business process is drastically different from the system process, then the cell process conversion involving radical changes to the current process is necessary to fit the system process. This option is significantly riskier than process adaptation.

The second row assumes that technical customization is limited to configuration of the system through tables. Most ERP systems come with configurable tables that model typical variations of business processes. Using the technical changes involved in table customization, organizations can tailor the system to a near fit to the process. Any further changes needed to achieve complete fit can be accomplished by changing processes. The cell fit system to process is an ideal situation, where fit has been accomplished through table customization.
The risk factor for this option is slightly greater than no customization. Mutual adaptation means that similar amounts of customization are applied to both the system and the process. Consequently, the cell is riskier than fit system to process and process adaptation. When there is still a significant gap between the system and process after table customization, fit process to system may be required to redesign the process to achieve fit. This option is even riskier than process conversion because it not only requires radical process change but also changes in the system.

In the third row, system customization is not limited to only table configuration; changes to the source code are also considered to be viable. While vendors discourage code customization, they do provide programming environments to facilitate such changes because there are variations of business processes that cannot be modeled through table customization. For various reasons, managers may decide to make no changes to the process and achieve the fit entirely through changes to the system. This option is referred to as system conversion. Organizations need substantially more expertise to work with the system than with table configuration and, hence, system conversion is riskier than the fit system to process cell. The system conversion and process adaptation cell involves moderate process improvements, as well as code customization. Depending on the nature of process changes, this option may not be as risky as the system conversion cell because it may increase flexibility in the process and require fewer coding changes. Finally, system and process reengineering refers to a scenario in which managers may choose to redesign the business process and make necessary changes to the system to support the redesigned process. Such changes in both the process and the system represent the most risky cell in the table.

The customization options depicted in Table II represent possible customization choices available to managers. No one option in Table II is necessarily better than the others per se. Consistent with the resource-based perspective, we argue, however, that some options are better for an organization than others and this is determined by the organization’s capability to make necessary system and process changes. Such capability requirements consist of technical change capability and process change capability.

**A. Technical Change Capability**

Technical change capability refers to an organization’s overall ability to customize ERP systems. The first ability within technical change capability is concerned with the understanding of default ERP system processes, configurations, and built-in options. It is crucial to recognize the underlying management principles and assumptions made by system vendors [20], [25]. This understanding forms the basis for planning and making appropriate changes to the system. Organizations may differ in this ability in terms of the scope and depth of their understanding of the adopted ERP system. For example, some organizations have a good understanding of the entire system and others may focus on a module or a part of the system. That is, the scope of understanding can differ. Similarly, the depth of understanding can vary as well. For example, one organization knows the processes that are supported by the system, while another organization also understands the assumptions and interconnections within the system well enough to plan any desired changes.
The second technical change capability is the ability to develop and modify large-scale software in a networked database environment. This ability is directly related to the extent to which organizations are able to make desired system changes. It is crucial that organizations have the ability to use a variety of software application development tools such as custom tools provided by ERP vendors, database management tools, and programming languages. The more tools an organization is able to employ, the broader the scope of their technical change capability. With each development tool, the degree of proficiency in using that tool signifies the depth of an organization’s technical change capability.

The third technical change capability is an organization’s ability to manage large-scale systems development projects. This includes setting unambiguous and realistic goals for the project, providing necessary resources to match the goals, maintaining clear communication channels among project participants, monitoring project progress, and detecting and correcting any problems within the project [38]. ERP implementation may be viewed as a series of technical change and process change. That places severe constraints on those with significant performance improvements.

Third, organizations must be capable of managing and coordinating large-scale business process changes. As discussed earlier, ERP implementation can be viewed as a series of customization projects that involves significant business process changes. As such, managing large-scale business process changes consists of managing individual projects, as well as integrating and coordinating multiple projects. To successfully manage individual projects, organizations need to have the necessary skills in project management and organizational change management. Integrating and coordinating multiple projects requires the ability to decompose large projects into smaller ones, manage the interfaces among these projects, and integrate their results to achieve overall objectives. The advantage of a series of projects is the opportunity for organizations to learn. That learning includes the exchange of knowledge within and across multiple project teams. Furthermore, knowledge acquired from earlier projects can be applied to future projects.

We measure an organization’s process change capabilities in terms of their scope and depth. For instance, the scope of the organization’s capability in managing process changes is broad when an organization is capable of managing very large-scale projects. Furthermore, we consider an organization to have high in-depth capability if that organization has a large set of alternative tools and techniques for managing process changes. We consider an organization to have low business process change capability if it has broad scope and great depth in all three abilities. Conversely, we consider an organization to have low business process change capability if it has narrow scope and limited depth in all three abilities.

### B. Process Change Capability

The other change capability that an organization needs is process change capability, which refers to its overall ability to customize business processes. Organizations first need to have the ability to understand their existing business processes and their business environment. That understanding should include not only the current state of existing business processes but also the history and evolution of these processes. Since ERP systems cross functions, enterprise-wide understanding of processes is more valuable than just the understanding of individual functional areas. Therefore, an organization would have higher process change capability if it has enterprise-wide understanding of its business processes and their history and evolution, as well as its products, customers, and other stakeholders.

Second, organizations must have the ability to design new or changed business processes, as well as implement these designs. Creative thinking and process design tools and methodologies are crucial for designing new or changed business processes. Creative thinking allows organizations to think “outside of the box” and generate many alternative designs. The ability to use process design tools and methodologies permits organizations to simulate and evaluate process design alternatives and to identify those with significant performance improvements.

### C. Overall Capability

Combining the technical change capability and process change capability, we can assess an organization’s overall capability in implementing and customizing ERP systems. Table III shows the four possible combinations of such an assessment. A novice organization has low capabilities in both technical change and process change. That places severe constraints on what an organization can do to customize its own processes, as well as the software package.

---

**TABLE III**

**CAPABILITY ASSESSMENT**

<table>
<thead>
<tr>
<th>Technical Change Capability</th>
<th>Low</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Change Capability</td>
<td>Low</td>
<td>Novice Organizational Expert</td>
</tr>
<tr>
<td>High</td>
<td>Organizer Expert</td>
<td></td>
</tr>
</tbody>
</table>

---
A technician organization has high technical change capability, but low process change capability. It has high ability to fit the system to the organization’s needs, but it may not be able to take full advantage of the possible process improvements. A technician organization should guard against the temptation to over customize the system. An organizer is an organization that has high process change capability, but low technical change capability. It is in a better position to take advantage of process improvement associated with ERP systems. Due to the inability to change the system, an organizer may not be able to realize a good fit between the process and the ERP system.

An expert organization has high capabilities in both technical change and process change. This kind of organization is in the best position to focus on business strategy without worrying about limits in its capability. It can pursue various alternatives of ERP implementation. An expert organization may over-customize both the process and the system so it should guard against the temptation to use all available expertise.

In addition to assessing its overall capability, an organization should also evaluate its capability with respect to each process and each module within the ERP system. An organization’s overall capability may be different from that with respect to individual processes and modules. For example, an organization may regard itself as an organizer overall, but consider itself a novice in manufacturing because it has little experience in changing its manufacturing processes.

Both technical change and process change capabilities will change over time as an organization goes through the ERP implementation process. A novice organization, for example, may improve its technical change ability by employing consulting firms and sending employees to technical training. Similarly, a technician organization can acquire process change ability by employing consultants to help it learn from earlier implementation projects. So organizations should periodically evaluate their own capability, identify their needs for capability improvements, and direct necessary resources to meet those needs.

V. A FRAMEWORK FOR EVALUATING ERP CUSTOMIZATION CHOICES

The framework presented in Table IV combines our discussion of process and technical customization options with process and technical change capabilities. The table shows the customization choices that are feasible for various organizational capabilities. For instance, a novice has the customization options in the upper left of the framework. An expert, on the other hand, can choose from any of the customization possibilities. Thus, an expert has far more options available to it than a novice. Customization options that require more process changes are better suited for organizers than for technicians. On the other hand, customization options that require ERP system changes are appropriate for technicians.

Because both technical capability and process capability of an organization may evolve over time, previously unrealistic options may become available as organizations learn from their experiences. Process change capability may improve when certain business processes are improved. Technical capability may improve when organizations gain experience in project management, acquire more technical talent, and so forth. Because
capabilities are dynamic, organizations should anticipate and facilitate capability growth. For example, in its initial selection of ERP implementation projects and associated customization options, an organization should choose projects and options that will allow it to grow in the future.

VI. FRAMEWORK APPLICATION: A CASE STUDY

The purpose of the proposed framework is to assist the evaluation of ERP customization options. As such, it provides a methodology for choosing customization options based on an organization’s technical and process capabilities. It suggests that customization decisions should involve: 1) determining the degree of system and process customization desired; 2) determining the organization’s capabilities to do technical and process customization; and 3) selecting a feasible cell that matches customization options with capabilities.

Given a desired ERP customization option, the framework can help assess the needs for additional technical change and/or organization change capabilities. Such needs can then be met by obtaining external consultants, offering training to existing employees, and deploying change agents.

Since in many companies ERP system implementation is a series of implementation projects, the framework can be used to plan these projects by anticipating and facilitating the growth of technical and process change capabilities and choosing projects and options that allow for capability growth over time. Taking this dynamic view, we can envision difficult customization and implementation projects becoming feasible over time as capabilities build during prior implementation projects.

In the remainder of this section, we illustrate the use of the framework through an analysis of an ERP system implementation at a private technological university. While this case study is a post-hoc analysis since the framework did not exist when the implementation projects started, the analysis shows the implicit technical and process customization choices made, how these choices changed over time, and how capabilities were built through customization choices.

This private university (PU) has a rather unique undergraduate academic curriculum and schedule. Instead of 15-week semesters, PU employs a 7-week term system, where there are four terms in each academic year. All undergraduate students are required to do junior and senior projects. This project-oriented curriculum, started in the 1970s, distinguishes it from other technological universities and is the core of its academic program. To support university operations, PU adopted and began to implement the Banner system as early as 1988.

Banner is an ERP system designed for educational institutions [50]. As such, Banner has modules for students, alumni and development, financial aid, and room scheduling, as well as modules common to most business ERP systems, such as finance, human resources, and payroll. Over 1300 universities and colleges worldwide have adopted Banner systems, including University of Arizona and Yale University [50]. Implementing ERP systems, such as Banner, in universities is no less complicated than that in large corporations since it could involve tens of thousands of users from various constituencies including current and prospective students, faculty, staff, administrators, and alumni. Many universities have fragmented computer systems attending different business functions that have never been integrated. Each university has unique business processes that distinguish itself from other competitors. Furthermore, ever changing government regulations play a major role in how state universities operate. In sum, the Banner system and ERP implementations in higher education possess the same general characteristics of ERP implementations using SAP in large business organizations.

PU selected Banner because it provided a common database and integrated applications using this common database. PU was an early adopter and served as a beta site for the Banner system. Over time, PU provided input to the vendor developing Banner, System and Computer Technology (SCT), and has implemented Banner modules, as they became ready for beta testing. The student module went online in 1988, the alumni and development module in 1995, the finance module in 1998, the HR/payroll module in 1999, the financial aid module in early 2000, and implementation of the scheduling module is in process.

The general implementation process for each module was as follows. First, the department or departments were selected. At PU, ERP implementation has taken primarily a functional orientation because, for many of the modules, a single primary department is responsible for the functions covered by the module. Second, members of the selected department and the implementation team decided how much process reengineering and redesign to do. This decision was made from a continuous improvement viewpoint. In making this decision, the functionality available in the Banner module was considered. Finally, the submodules within the module were selected, configuration options were selected, and code was added or changed to fit the module to the redesigned process. Overall, PU first made process customization choices with some consideration of the module functionality, and then customized the system modules to whatever extent was needed to achieve fit. Because of PU’s technical expertise and its role as the beta test site, such a sequence was feasible.

Over time, the customization choices made have moved from primarily technical customization to primarily process customization, but these choices depend on the criticality of the functionality to the mission of PU. We illustrate the nature of these choices by discussing the implementation of three different modules, student, finance, and scheduling. Table V shows the framework as applied to PU’s customization choices for these three modules.

A. Student Module

The student module implementation began in 1988. At that time, the implementation team viewed ERP implementation in the context of traditional systems design and development. The team determined what the primary user, the registrar’s office, wanted and what the process was. They then worked with SCT to acquire technical expertise that allowed them to make major changes to the student module in order to fit the existing process. That is, they chose the “system conversion” cell, which is feasible only if the implementation team possesses technical change capabilities (labeled Student Module 1 in Table V).

Technical customization of the student module is necessary because educational programs are the core advantage and competency of an educational institution. In particular, PU has
TABLE V
FRAMEWORK APPLIED TO PU’S ERP IMPLEMENTATION CHOICES

<table>
<thead>
<tr>
<th>Technical Customization Options</th>
<th>Process Customization Options</th>
<th>Process Customization Options</th>
<th>Process Customization Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module customization</td>
<td>No change</td>
<td>Incremental change</td>
<td>Radical change</td>
</tr>
<tr>
<td></td>
<td>No customization</td>
<td>Process adaptation</td>
<td>Process conversion</td>
</tr>
<tr>
<td>Table customization</td>
<td>Fit System to process</td>
<td>Mutual adaptation</td>
<td>Fit Process to System</td>
</tr>
<tr>
<td>Code customization</td>
<td>System conversion</td>
<td>System and Process adaptation</td>
<td>Expert</td>
</tr>
</tbody>
</table>

A concern is that the process changes may be radical rather than incremental. The team learned the costs and benefits of relying on PU’s technician capabilities. The benefits are a well-functioning customized module that supports PU’s project-oriented educational mission. The cost is over customization through overuse of its technician capabilities. It was realized that not all technical changes to the ERP system provided strategic benefit to PU. Current implementations of student sub-modules take a more balanced approach between technical customization and process customization (labeled Student Module 2 in Table V).

B. Finance Module

The implementation of the finance module took a very different approach. The process by which PU does its accounting and manages its finances was similar to most other colleges and universities and was not considered as a competitive advantage for PU. Learning from the costs of technical customization of the student module, the implementation team decided to minimize technical customization, and focused on training users for the financial process embedded in Banner’s financial module. That is, the “fit process to system” cell was selected, which requires expert capabilities, i.e., both technician and organizer abilities (labeled Finance Module in Table V).

PU’s technical change capabilities were well developed, but its organizer abilities were lacking. To develop the organizational change capability, PU initiated a reengineering project to improve the operation of the process. The reengineering was done in cooperation with consultants from SCT so that the resulting process would be a close fit to the process embedded in the Banner system. To fit the process to the system, PU made many changes in its financial processes, including changing its entire chart of accounts to fit Banner, changing retirement deduction calculations to the method used in Banner, and training all users that normal account balances were now negative. The changes to the process required negotiations with the user community and extensive training, both of which have served to build PU’s organizer capabilities.

C. Scheduling Module

Decisions about implementing the scheduling module are underway now. The implementation team knows that the changes required are at least those of the “mutual adaptation” cell, that is, at least incremental process change and table customization. A concern is that the process changes may be radical rather
than incremental and the system changes may require significant coding. There is a major difference in project difficulty and risk between a “mutual adaptation” project and a “system and process reengineering” project (labeled Scheduling Module in Table V). On the system side, there is concern that Banner’s scheduling module does not have adequate functionality and, thus, implementation may require coding. On the process side, the PU team is deciding how much reengineering is needed to improve the scheduling process. For example, one issue is which rooms should be centrally scheduled. Departments like the flexibility of having local scheduling control over various conference rooms and other space near their department. Central scheduling of such space, however, may provide better utilization of limited space. Making decisions about whether and how to change scheduling of space on campus is a critical part of the ERP implementation process. Since scheduling decisions affect most departments and individuals in these departments, changing the scheduling process radically could be risky.

These decisions about reengineering the scheduling process determine the process customization choice, that is, whether implementing the scheduling module requires incremental or radical process change, which in turn affects the organizational capabilities required and the size, scope, and risk of the implementation project. Through the implementation of the student module and the finance module, PU has built expert level ERP implementation capabilities. Thus, deciding to do radical process reengineering and significant code customization is feasible, which gives PU the flexibility to choose any customization option. PU or any other organization must, however, avoid projects that are more complex than needed.

The above discussion illustrates the dynamic use of the framework for evaluating ERP customization choices. At PU, the implementation of each ERP module, and sometimes each submodule, is an implementation project in itself. Each project involves decisions about how much to change the associated business process and, simultaneously, how much module customization to allow. These choices must be made within the context of organizational capabilities. Since organizational capabilities develop and change over time, the sequencing of implementation projects is important, so that each project is feasible from the perspective of organizational capabilities at the time it is initiated.

VI. CONCLUSION

The framework developed in this paper identifies nine customization options based on the degree of changes undertaken for both the ERP system and the business process. It is designed to help organizations understand which customization options are available and which of these are feasible given an organization’s capabilities. The applicability of the framework was illustrated through a case study of an organization that employed a phased implementation of several ERP modules. This phased implementation also illustrated the growing capability of an organization as it undertook ERP implementations involving technical and organizational changes and how additional customization options became feasible with increased organizational capabilities.

A. Contributions to Practice

This framework allows ERP implementers to examine many implementation possibilities rather than simply following the “conventional wisdom” of fitting processes to the system. The framework shows nine customization options that depend on how much to change the business process and how much to change the ERP system. To illustrate the contribution of this framework to the practice of engineering management, we describe how a typical organization implementing an ERP system would use the framework.

The first step in using the framework is to assess how well the selected ERP system matches the business process. Typically, this must be assessed for multiple ERP modules and multiple business processes. Then, the organization can use the framework to think about the possibilities of changing the system and changing the business process to provide better fit between the system and the process. As a result, the organization will have a set of options, some involving more process change and some involving more system change, for achieving better fit.

The next step is to assess the organization’s ability to make these changes. The paper describes organizational capabilities as technical change capabilities and process change capabilities. Depending on the results of assessing these capabilities, the list of options developed in the previous step can be evaluated for feasibility for that organization. The final step is to consider a longer-term plan depending on which options are the most feasible for the organization over time.

We have suggested that ERP system implementation should be viewed as a series of implementation projects. This view has several implications for engineering management. First, organizations have overall process change and technical change capabilities that may differ from their change capabilities with respect to an individual project. Second, these capabilities can change over time as organizations learn from previous projects. Third, it is important to plan the path of implementation projects so that more difficult projects become feasible through learning. That is, an organization should not only focus on the customization options for the current projects but should also consider options for capability growth.

The result of such an approach is to view ERP implementation as a portfolio of projects. Different projects may require different levels of effort, resources, and expertise. Thus, they should be managed in their own ways. The framework provides a way of thinking about the implementation choices to be made and helps engineering managers understand and evaluate these choices.

B. Contributions to Research

The “conventional wisdom” in the literature is that organizations should change their business processes to fit the ERP system, which embeds “best practices.” Any review of case studies of ERP system implementation will find many exceptions to the conventional wisdom. Managers need better advice from the literature.

To date, there are frameworks in the literature describing the possibilities for technical changes, that is, options for adapting the ERP system to the business process, e.g., [12],
Our framework goes beyond these in several ways. First, we consider both technical and process changes, which are the options managers actually have available to them. Second, we consider these customization options in light of an organization’s capabilities to make these changes. Third, we promote a dynamic view of ERP implementation that assesses the building of organizational capabilities through a series of ERP implementations and how this increases the feasibility of various customization options over time. Finally, the framework can be used to generate testable hypotheses of the relationships between customization choices and organization capabilities.

C. Limitations

The development of the framework was motivated by practicing engineering managers trying to make decisions about ERP customization choices. The framework itself was developed from the literature, and extends existing frameworks. We have demonstrated the applicability of the framework through a case study. This case study, however, is not a full validation of the framework; it only illustrates its applicability in one organization. ERP customization is only one of the important issues in ERP implementation. Other issues such as planning, management support, and change management can be equally important. Therefore, this framework should be used together with others tools to address all important issues in ERP implementation by the project team.

Furthermore, the framework does not determine decisions for managers; rather it provides the possibilities for customization and indicates the level of technical and organizational change capabilities needed to implement each possible customization option. The framework is designed to help managers think about the feasible options, and should be used in that way.

D. Future Research

Engineering managers can use the framework, as is, in the manner suggested above. The framework, however, requires further validation by testing it both in experiments and in organizations. A future research study could compare the decisions of managers using the framework with those not using the framework in a controlled laboratory study. Once further understanding of the framework in use is developed, the framework should be tested in a similar way in multiple organizations to validate the framework more fully than was done in this study.

REFERENCES


Wenhong Luo received the Ph.D. degree in management information systems from University of Kentucky, Lexington.

He is an Associate Professor in the Department of Decision and Information Technologies, College of Commerce and Finance at Villanova University, Villanova, PA. His publications have appeared in leading journals such as Communications of the ACM, European Journal of Information Systems, Information and Management, The Information Society, Computer Supported Cooperative Work, and Group Decisions and Negotiation. His research interests include electronic commerce and information system project management.

Diane M. Strong received the Ph.D. degree in information systems from Carnegie Mellon University, Pittsburgh, PA.

She is an Associate Professor in the Department of Management at Worcester Polytechnic Institute, Worcester, MA, and Director of the MIS program. Her publications have appeared in leading journals such as Communications of the ACM, Communications of the AIS, Journal of Management Information Systems, ACM Transactions on Information Systems, Journal of Database Management, Journal of Systems and Software and Information and Management. Her research centers on data and information quality and on the organizational impacts of MIS applications systems, especially enterprise systems.

Dr. Strong is serving on the Association for Information Systems (AIS) Council and was Program Co-Chair for Americas Conference on Information Systems (AMCIS), Boston, MA, and for the International Conference on Information Quality.