Chapter 3.1
A Modern Socio-Technical View on ERP-Systems

Jos Benders
Tilburg University, The Netherlands, & Radboud University Nijmegen, The Netherlands

Ronald Batenburg
Utrecht University, The Netherlands

Paul Hoeken
Radboud University Nijmegen, The Netherlands

Roel Schouteten
Radboud University Nijmegen, The Netherlands

ABSTRACT

This chapter sketches an Organization Design perspective called "Modern Socio-technical Design", and subsequently discusses the implementation of Enterprise Resource Planning Systems from this perspective. The authors argue that the praxis of ERP-system implementation is often at odds with socio-technical insights, leading to various problems that ERP-end users are confronted with. These tensions may not be inevitable, but simply result from taken-for-granted organization assumptions underlying ERP-implementation praxis. The socio-technical insights are intended to help practitioners reflect on ERP-implementation praxis, and discuss to what extent an ERP-system is appropriate and if so, where socio-technically inspired choices may be made within configuration processes.

DOI: 10.4018/978-1-60566-264-0.ch029

If you automate a mess, all you get is an automated mess

—Anonymous Saying

INTRODUCTION

Do ERP and teamwork coincide? Koch and Buhl (2001) studied 24 cases where teamworking and ERP-systems were introduced simultaneously. Their answer to the question is negative, as they argue:

[s]ince the concepts of teamwork and ERP-systems appear widely diffused, one might expect that both are closely aligned when they are implemented [...] As we demonstrate, however, this is not the case [...] Although ERP is possible to configure in such a way that autonomous teamwork on the

shop floor is supported, we found that ERP and teamwork rarely interact directly. When they do, they are potentially competing change programs, and indirect competition predominates. (Koch & Buhl, 2001, p. 165)

They argued that the problem was not the configuration of ERP-systems for autonomous teamwork, but that there were (1) no modules available for this configuration process and (2) no consultants with the necessary knowledge. To illustrate this, they discuss the case of a machine building company where an attempt was made to align ERP-systems and teamwork. The attempt was unsuccessful however, as it started from different premises: the consultants implementing the ERP-system focused on enhancing production planning and control from a central perspective and "did not push for supporting teamwork" (2001, p. 173). Furthermore, "in-built features" of the ERP-package used "were realized in a way that led to a strengthening of other parts of the planning than the teams" (2001, p. 173). Finally, the technical aspects of implementing the system were so complex and time-consuming that organizational aspects received little attention. The members of the self-managing teams in the project team could not turn this tide. Whilst the teams were authorized to take certain decisions, the key tasks of (local) production planning was centralized. In a second round of ERP implementation, the shop-floor teams' experiences were not taken into account and the new tasks were confined to data entry and providing feedback on production orders. Koch and Buhl stress that the outcome was not a necessity but "a mixture of intended and not intended actions both from the ERP-coalition" and members of the self-managing teams (2001, p. 174).

Their findings do not stand alone. At a more general level, Soh and Sia (2004) studied how ERP-systems were used in three hospitals in East Asia. They wondered whether empowerment or control would prevail in how these systems were used. The result of their study was that while both outcomes are possible, in praxis control tended to get the overhand. In terms of Orlikowski (2000), the 'control' potential of ERP-systems is apparently and in the course of time more easily enacted than the 'empowerment' potential (cf. Boudreau & Robey, 2005).

Koch and Buhl's study gives rise to the question why it is apparently so difficult to combine self-managing teams and ERP-systems. Answering this question calls for a more integrative view on organization design because teams are embedded in organization structures and information systems such as ERP-systems are to support decision-making in such organizations. This view remains implicit in Koch and Buhl's study, but is necessary if their recommendation of developing "practical templates" to support configuring ERP-systems for self-managing teams is to be realized. In a broader perspective, self-managing teams are seen as a hallmark of modern organization, for instance as part of "high performing work systems".

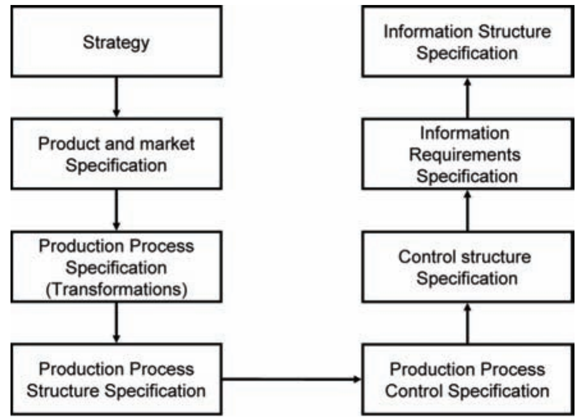
In the remainder of this chapter we first present an organizational design methodology that provides an integrated view on structuring organizations so that suitable organizational environments are created for self-managing teams and subsequently, after this structure has been designed, the informational requirements are analyzed so that information systems may be configured and implemented. This so-called "Modern Socio-technology" incorporates some organizational design principles which, as Koch and Buhl's (2001) work shows, tend to sit uncomfortably with ERP-systems in practice. These tensions are discussed after presenting Modern Socio-technology. This analysis is necessary as a first step for developing the templates for which Koch and Buhl signaled the need.

MODERN SOCIO-TECHNICAL SYSTEMS DESIGN

What was to become socio-technical systems design (STSD) started with studies in the late 1940s in a number of British coal mines. In 1951, Trist and Bamforth published a founding article on STSD while the London-based Tavistock Institute played a key role in further developing socio-technical design into practical applications. During the 1950s and 1960s these notions were picked up in many countries, with Norwegian and Swedish researchers playing key roles. In the Netherlands a strand of socio-technical scholars and practitioners developed a widely accepted research based organizational design methodology (De Sitter, Den Hertog, & Dankbaar, 1997; De Sitter, 1998). This Dutch variant, called Modern Socio-technology (MST), builds on the classic STSD. In the 1970s Ulbo de Sitter played a key role in developing this socio-technical systems theory (with some roots in German sociology). During the 1980s this design theory was enriched with a proper design methodology based on action research. MST mainly differs from STSD by its integral approach. Whereas classic STSD provides a set of static and partial design principles, MST offers detailed structural principles in terms of design content, while at the same time specifying a theory of change by means of worker participation and training (Van Eijnatten, 1993). To emphasize the integral character of this approach, Van Eijnatten and Van der Zwaan (1998) labeled it Integral Organizational Renewal (IOR).

Since MST provides an integrated body of knowledge comprising analysis methods as well as (re)design rules (Van Eijnatten & Van der Zwaan, 1998), it is this Dutch variant of STSD we use for our analysis of the effects of ERP on organizational design. Team based organizations are a central concept in this approach aimed at meeting organizational requirements, as well as improving the quality of working life. To achieve this the design order principle as depicted in Figure 1 was developed (Groep Sociotechniek, 1986; De Sitter, 1998).

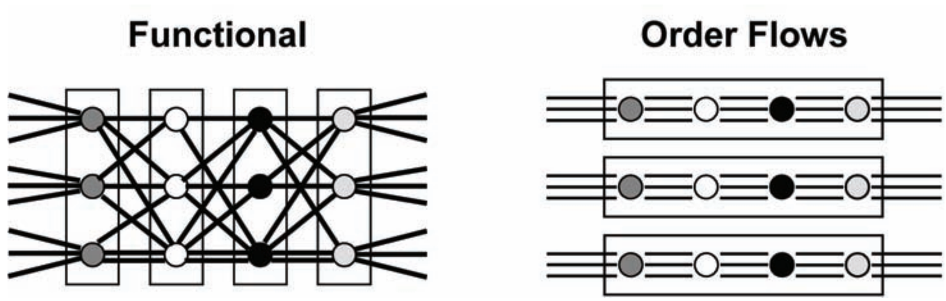
Figure 1. The Socio-technical design order principle (based on De Sitter, 1998)



The Design Order Principle: Reduction of Complexity

The production structure of an organization should reduce the variety and the number of possible interferences as much as possible. The principle is to effectively break down complex demand/ transformation systems into a number of far less complex sub systems that are as independent as possible. In practice this implies parallelization and segmentation of order flows. Based on this, control structures can be governed by autonomous groups according to the principle of minimal critical specification. This actually follows Ashby's law of requisite variety (Ashby, 1969) holding that a system's control capacity should be at least equally as the variety it needs to control. By parallelization the number of interferences and dependencies between transactions can be drastically decreased. Parallelization can be contrasted with production specification into functional departments, in which every department is responsible for only one kind of transformation. Figure 2 illustrates the complexity reducing effect of parallelization in a functional organization (job shop like) compared

Figure 2. Functional Organization compared to Parallel Order Flows (based on De Sitter, 1998)



to a set of parallel order flows.

Socio-technical systems design implies a top-down development of the organizations' production structure, and a bottom-up development of the control structure needed. Starting at the left-hand side of Figure 1, the (top) management level, business strategy initially drives product and market specification. Based on this the required production processes and structures are designed. Within this top-down chain of actions, production process specifications consists of a large number of 'transformations' ranging from integrated production lines to traditional job shops. Depending on the scale and requirements of these transformations the production structure specification is defined. Each transformation has to be controlled towards a number of aspect related targets, such as quality, quantity, efficiency, costs, environmental impact and timely delivery. All these control activities can also be aggregated accordingly, ranging from task specialization in a bureaucratic hierarchical structure to self-directed work teams for specific product market combinations. At this first part of the MST design, basic decisions on production and job structure design ('what needs to be done?') are taken.

Moving to the right-hand side of Figure 1, the next steps particularly consist of the required information specification. These are derived from the control activities (information requirements specification) as specified top-down, and – bottom-up – developed into an aligned information structure (information system or set of information systems). Here, information structure specification implies the effective and efficient support of production and control activities. In this chain of the design different levels for control are to be distinguished. At the macro level strategic control of external relational are addressed. At the meso level inter-group coordination is concerned, while at the micro-level control teams and employees are responsible for controlling individual transformations. At this second part of the MST design, basic decisions on control and job design ('how should it be done?') are taken.

The Minimal Critical Specification Principle: Segmentation

Once parallel flows are created, task assignments allocated to units or groups should aim for an 'optimal' level of independency. This may be achieved by splitting the flow into a number of so-called "segments". This implies that tasks are grouped in such a way that the number and content of interfaces with other organizational units are minimized. Each interface creates the risk of interference and disturbance and hence a need for co-ordination. As shown in Figure 2, the reduction of the number of interfaces is achieved because incompatible grouping of transformations, such as welding and coating of metal parts, or nursing and operating in hospitals, are divided by flows into segments. In defining the number of transformations or people involved one should note that co-ordination and direct communication between segments or people will accumulate accordingly, leading to higher levels of required co-ordination. In this respect, the socio-technical design of teams by segmentation is also guided by the principle of minimal specification.

The Task Completeness Principle: Quality of Working Life

In an MST-design, segments are generally operated by self-managing teams. The preceding steps made sure that the tasks assigned to the teams may actually be carried out as independently as possible. The local control makes quick interventions possible in case of unexpected events.

The notion of maximum local control also applies at the level of individual employees. The organizational advantage of such local control lies in the potential for quick local interventions, but also in positive effects on employee behavior. This subscribes to Karasek's plea to balance job demands (i.e. control needs) and decision latitude (i.e. control capacity; cf. Karasek, 1979). A job is considered a 'good' job if it (1) consists of complete tasks and sufficient control capacity to deal with control needs conclusively, and (2) offers sufficient challenges to job holders. Creating such jobs removes a source of stress, namely that employees see undesirable events happen but are not allowed to intervene. By creating "good jobs", workers are to be motivated with positive effects on, behavior and absenteeism, and thus productivity.

MODERN SOCIO-TECHNICAL VS. ERP-SYSTEMS DESIGN

As stated, deploying ERP-systems may have negative consequences for employees and organizations. Below, we aim to understand these negative effects by projecting the socio-technical principles on (explicit and implicit) design of organizations through ERP. We systematically confront the key MST principles discussed above with ways in which ERP-systems are commonly implemented.

ERP-Systems at Odds with Design Order Principle

A first and key difference between a socio-technical design and ERP implementation is the starting point. ERP, both as a business concept and an Enterprise Information System (EIS) automation concept, was originally developed to fully integrate different information systems that (particularly large) organizations deal with. Instead of creating middleware applications to connect separated Information Systems (for instance production planning and billing systems), ERP radically replaces them all. With ERP, the total information architecture needs to be redesigned in order to automate all processes in a similar way. The input and creation of information by users is designed to take place according to a one single point of entry principle, to avoid data redundancy. Business rules are formally translated in work and information flows throughout all of the ERP-modules, thereby similarly modelling departmental and functional roles. And finally the information representation is designed by standard templates and forms. In ERP-systems the traditional presentation, application and database layers are strongly integrated, with one single database and meta-model as its main core. Therefore, socio-technical organization design departs from design criteria, derived from a strategic position, to design the production structure and the control structure. The information structure is derived from the production and control structure. In the case of ERP, the information system provides the starting point and must be configured to fit organization structure and processes. These often follow the ERP-system rather than the other way around. In practice the complexity of ERP software enforces that organizations tend to 'stick to the standard' offered by the ERP vendor (Benders, Batenburg, & Van der Blonk, 2006).

This design order problem becomes particularly clear when multi-site organizations are considered. The central concern is about the fit between the systems to be integrated on the one hand, and the particular practice of organizational subunits on the other. The more subunits deviate from other subunits and the more these subunits are dependent on each other, the more likely that ERP implementations will need to depart from standardization of the information structure (i.e. the IS infrastructure or architecture). Customization of the ERP-software by bolt-ons, add-ons and spreadsheet workarounds are discouraged, while from a MST perspective these might be allowed to assist local decision-making and enhance control capacity. In other words, ERP violates the parallelization principle in cases these parallel flows demand differing information support functionality. This is problematic as ERP implementations usually entail the implementation of only one business model in the software to save on implementation and software maintenance costs (Swanson, 2003).

ERP-Systems at Odds with Minimum Critical Specification Principle

In ERP-systems design there is a central database while integration by control is organized in functional software modules making use of one common IS/IT environment. The modular design of ERP-systems, however, also implies functional decomposition as there are separate modules for control domains such as finance, quality management, logistics and HRM. In addition, different functionalities, such as data input, query dialogues and management reports, are separated with ERP-systems. As modules are configured by functional specialists, the design of ERP-systems leads to a tendency to create tasks that are functionally decomposed as well. Obviously, this segregation of control aspects contradicts the fulfilling of the socio-technical requirement of integrating primary and supporting functions. In terms of Figure 2, the existing functional organization (similar operations grouped together) is maintained. The complex product flows in between different organizational units is followed by the software (as is the case with workflow management software). Figure 2 illustrates the risk of this approach, seen through a socio-technical lens. At the left, the functional structure is shown. A product that has to undergo various functional operations is taken from one functional department to the next, leading to complex routings through the organization. The socio-technical solution is, wherever possible, to place the operations in the same sequence as needed to make this particular product in multi-functional departments, as shown in the right-hand part of Figure 2.

Current ERP practices usually keep the complex functional structure intact, and follow the product through the different departments with an information system. From a socio-technical view this situation could be called 'technology-enabled complexity maintenance': instead of simplifying the situation to be controlled, the complex situation is maintained and the control possibilities are improved. In effect, this process orientation is the electronic equivalent of the 'chasseur', the French name for a person who used to be sent into a factory to track and speed up orders. The risk of using an ERP-system is that the symptoms of a complex structure are fought, but that the underlying problem of unnecessary complexity is not solved (De Sitter et al., 1997).

ERP-Systems at Odds with Task Completeness Principle

ERP-implementations directly affect job decision latitude in various ways. During the configuration process, (future) users are authorized to take particular decisions. In granting authorizations, ERP implementers directly influence job decision latitude. However, as with other organizational changes, it appears that only in exceptional cases ERP-implementers take these effects on job content explicitly into account. Instead, predefined user groups and role structures tend to be used. The control perspective often comes back in the form of the "segregation of duties", a key principle in administrative organization which is to prevent creating opportunities for fraud. Control cycles are not closed, as modern socio-technical design prescribes.

A similar aspect concerns authorizations for data access and data entry. In standard authorization schemes, these are often concentrated with a limited number of users, generally those at higher hierarchical levels. This may cause problems at the shop floor, as work can not proceed in the absence of the authorized. A frequently used option to 'work around' these authorization problems is granting employees more access rights than controllers see as proper (Pollock & Cornford, 2004; Le Loarne, 2005). A user may formally or informally arrange access to additional user IDs and passwords to be able to perform all necessary tasks. In a socio-technical design user requirements would be the starting point for getting access. Obviously, data entry jobs within ERP-systems consisting of monotonous and short cyclical tasks are, in socio-technical terms, seen as "passive jobs" (Karasek, 1979). In a socio-technical design, data entry tasks would be integrated with other tasks into complete jobs. More broadly, ERP-users often need to put data in for other functions in the organization. A comment as "SAP creates work" (Le Loarne, 2005, p. 526) signals that this may not always be efficient and is certainly not always perceived to be efficient. The empirical examples described above support this notion.

Soh and Sia (2004, p. 25-26) see the ability to track products as a form of empowerment: what they call ERP's 'process orientation' allows employees to track the progress of individual products. Compared to a situation where this is not the case and hence employees are confronted with orders, insight into these orders' process statuses may be seen as progress for employees. However, as long as they are not authorized to take action, this may have the effect of increasing stress levels, because of lack of control capacity: seeing problems happen without being able to solve them, or insight alone is not sufficient. As a result, the control capacity needs to be adjusted as well.

Implications

Implementing ERP-systems goes along with organizational changes. Their breadth and depth, however, seem generally underestimated. As Koch and Buhl (2001) showed, organizational consequences are not always, and probably generally not, taken into account when implementing ERP-systems. Consequently, unintended and negative results are likely to occur. As Koch and Buhl's machine building case showed, organizational changes as a result of ERP and teamwork ask for contradictory directions for change. Whereas ERP-implementers strive for standardization and centralization, teamwork implies empowerment and decentralization, enhancing team autonomy. Especially in terms of job decision latitude, ERP may easily be at odds with team working. If the consequences of an ERP-implementation for job design are not explicitly considered, teams' potential to deal with environmental complexity and flexibility is not used or even negated. ERP's focus on standardization, authorization schemes and central control limits the job decision latitude (control capacity) at individual and team level (Karasek, 1979). As a result, the balance between control need and control capacity at individual and team level (an objective of MST and team working for reaching organizational goals) is disturbed and increasing stress levels and organizational inefficiencies may result.

However, these contradictory directions do not necessarily have to lead to negative results. As Buhl and Richter's use of participatory design tools shows, the implementation of an IT-system "can be productive and constructive if they are explained to other employees and if they, for their part, get room to and time to develop alternative models and their own perspectives" (Buhl & Richter, 2004, p. 270). These participatory design tools fit into MST's design methodology and action research approach. Starting from organizational requirements and building autonomous teams as the building blocks of the organization, the technical systems must fit this organizational design. Participation of team members in the configuration of ERP enhances mutual understanding between different groups in the organization and, as a result, the system's productivity and the worker's enactment of the technology (cf. Orlikowski, 2000). Furthermore, it does justice to the teams' autonomy and decision latitude.

Two points of special attention are attached to this participative approach. First, it requires that ERP-systems are truly open for configuration in terms of the underlying technology (as with customization), business rules (as with parameterization), and the financial barriers of these system adaptations. Scott and Wagner (2003) state that ERP can be customized to adapt the principles of socio-technical design. By longitudinal and participatory design analysis of an Ivy League University in the US they conclude that – in contrast to the opinion that ERP is uncontrollable (or even a "technological monster") – temporal turns and negotiations during the ERP project led to "a hybrid working rhythm that is inscribed into its socio-technical infrastructure" and hence a socio-technical information system was created. It should be noted however that the adaptability of ERP-systems in practice often deviates from the "anything goes" adage that SAP and other ERP vendors use in promoting their systems as total business or industry solutions.

The second point of special attention is that staff participation may result in single-sided attention for quality of working life. Following MST's design methodology, first and foremost the organizational structure must be built in order to meet organizational (corporate) requirements. Self-managing teams are the main concept in this design, but these teams' autonomy cannot be a goal in itself. In MST balancing organizational requirements and workers' needs (quality of working life) is essential and a logical consequence of the design process. As a result, participation of teams in, for instance, decisions about authorization schemes in ERP, rather than the technical specifications, seems highly important. For instance, Buhl and Richter (2004) show that shopfloor worker participation in accessibility rights in the system resulted in modifications that supported the teams' competencies and some autonomy to plan their own time and production capacities. These are important aspects of control capacity and therefore positively influence the balance between control need and control capacity (cf. Karasek, 1979).

Example: A Board Manufacturer

A solid board manufacturer delivers special products to a number of market segments and areas. The organization faced the problem that some clients demanded rather short delivery times but were willing to pay premium prices, whereas others accepted longer delivery times and ordered longer in advance. To serve both market segments a partial parallelization of order flows was suggested, namely of sales and order acceptance functions. Nothing changed in the manufacturing process, as the heavy machinery in the mill was too expensive to re-group.

The process of order acceptation was delegated from the central planning department to the regional sales offices. To minimize mutual interdependencies between sales offices or between sales offices and production, the production capacity was administratively distributed over contingents per area. The areas only needed to co-ordinate their activities in the occasional event of over- or underbooking.

However, the standard ERP-features did not allow for this tailor-made organizational solution. Thus, to facilitate this parallelization the ERP system was extended with a bold-on, a 'sales budget and order acceptance subsystem'. This interfaced with the sales forecast, the budgeting and production scheduling modules of the ERP software used by the organization.

DISCUSSION AND CONCLUSIONS

Our analysis is a first step in identifying some of the main potential causes of tension in implementing ERP-systems while creating suitable organizational environments for self-managing teams. As Koch and Buhl (2001) described the misalignment of ERP-systems implementation and teamworking, MST provides a useful lens to describe possibilities for aligning ERP and teamworking. Both include change programs aimed at dealing with organizational problems concerning lack (or loss) of effectiveness and flexibility. The existing literature shows that awareness of the organizational consequences of ERP implementation is an important condition for aligning ERP and teamworking. Centralization and standardization that go along with many ERP implementations are at odds with the three MST design principles we discussed. These are aimed at organizational structures that best respond to environmental complexity and in which the design and development of autonomous teams is the main concept. Being aware of these possible consequences opens the route to designing effective and efficient organization structures around autonomous teams that are supported by ERP-systems that are configured to meet the organization's requirements. Moreover, following MST design principles the work in teams or individual jobs should result in meaningful and complete jobs, due to the balance between job demands and job decision latitude. As Buhl and Richter (2004) show, communication, participation and cooperation of different participants, such as shopfloor workers, line managers, production planners and IT specialists, are important means to create the necessary awareness.

A limitation is that our chapter focuses on what one may call 'traditional' ERP implementations, i.e. the deployment of product software solutions within organizations to integrate, automate and support business processes. Several developments can be recognized that go beyond the standard type of ERP implementation. A major movement is that more and more organizations use ERP software to extend and virtualize their organization, their supply chains and corporate networks. Most ERP software offers e-business functionalities to support this virtualization by tools and modules for e-sales, CRM, e-procurement, e-sourcing and so on. These cross-organizational functions put new pressures to adjust ERP-systems to these new organizational settings. Boersma and Kingma (2005) presented an in-depth case study of mutual ERP adaptation and virtualization through enforcing supply chain conditions. This type of external adaptation will obviously multiply fast if organizations increasingly extend their information exchange and system integration with other chain and network partners. As a consequences new developments within the software profession emerge like Service Oriented Architecture (SOA). SOA particularly fits the idea of transformation of a static functional organization to a dynamic network of services. Flexibility is the key competence to achieve, by rapidly creating new services from existing ones, and by adjusting the services network onto the fast changing environment. Almost all ERP-vendors have announced a reshaping of their products towards this service orientation. Applying SOA can enable software solutions to dynamically support specific organizational needs, while combining and re-using product software and existing IT.

These developments might retune the disadvantages of classic ERP we here brought up in this chapter. They do not change the core characteristics of ERP as an integrative and control-oriented system however. Research is rightly to explore about the impact of fast-moving technological developments, such as e-business and SOA on the (inter-)organizational fit of ERP. This might actually be an interesting momentum for ERP-vendors and consultants to rethink their architectures and implementation methods taking the socio-technical principles here referred to into serious consideration.

The Modern Sociotechnical design perspective seems to sit comfortably with the adage "First organize, then automate". In sharp contrast, the current ERP-implementation praxis can be characterized as "first implement, then re-organize". This has a couple of implications for practitioners. These include various IT-specialists, consultants from "implementation partners', and at the demand site, managers and (end) users. These different "stakeholders are likely to have partially diverging interests. The MST-perspective probably serves managers and end users best: this perspective may strengthen them to point to critical aspects during implementation. Given the political realities in many organizations, the ERP-implementers will probably be in a strong position and argue against "first organize, then automate" as this may lead to questioning the wisdom of ERP-systems. Less radically, however, the socio-technical perspective may influence choices within the scope of ERP-systems. These include:

- business process analyses should precede ERP-implementation;
- be critical about maximal coupling and consider de-coupling organizational units;
- use the notion of "local control" in authorizing end users;
- allow, when needed, local support tools.

Within ERP-design, templates for implementing self-managing teams may be developed. These can assist implementers to resolve the dilemma between centralistic, top-down control and specialization tendencies inherent in ERP-system design on the one hand, and on the other the socio-technical ideas of reducing system complexity, maximum local control and minimal critical specification.

REFERENCES

Ashby, W. R. (1969). Self-regulation and Requisite Variety. In F. Emery (Ed.), *Systems Thinking* (pp. 105-124). Harmondsworth: Penguin Books.

Benders, J., Batenburg, R., & Van der Blonk, H. (2006). Sticking to standards; Technical and other isomorphic pressures in deploying ERP-systems. *Information & Management*, *43*(2), 194–203. doi:10.1016/j.im.2005.06.002

Boersma, K., & Kingma, S. (2005). From means to ends: The transformation of ERP in a manufacturing company. *The Journal of Strategic Information Systems*, *14*(2), 197–219. doi:10.1016/j.jsis.2005.04.003

Boudreau, M.-C., & Robey, D. (2005). Enacting integrated information technology: A human agency perspective. *Organization Science*, *16*(1), 3–18. doi:10.1287/orsc.1040.0103

Buhl, H., & Richter, A. (2004). Downplaying model power in IT project work. *Economic and Industrial Democracy*, *25*(2), 247–276.

De Sitter, L. U. (1998). *Synergetisch produceren, human resources mobilisation in de productie: een inleiding*. Assen: Van Gorcum.

De Sitter, L. U., Den Hertog, J. F., & Dankbaar, B. (1997). From complex organizations with simple jobs to simple organizations with complex jobs. *Human Relations*, *50*(5), 497–534. doi:10.1177/001872679705000503

Govers, M. (2003). *Met ERP-systemen op weg naar moderne bureaucratieën?* Nijmegen: University of Nijmegen.

Groep Sociotechniek. (1986). *Het flexibele bedrijf; Integrale aanpak van flexibiliteit, beheersbaarheid, kwaliteit van de arbeid, produktie-automatisering*. Deventer: Kluwer.

Karasek, R. A. (1979). Job demands, job decision latitude and mental strain; Implications for job redesign. *Administrative Science Quarterly*, *24*(2), 285–308. doi:10.2307/2392498

Koch, C., & Buhl, H. (2001). ERP-supported teamworking in Danish manufacturing? *New Technology, Work and Employment*, *16*(3), 164–177. doi:10.1111/1468-005X.00086

Le Loarne, S. (2005). Working with ERP systems: Is big brother back? *Computers in Industry*, *56*(6), 523–528. doi:10.1016/j.compind.2005.02.010

Orlikowski, W. J. (2000). Using technology and constituting structures: A practice lens for studying technology in organizations. *Organization Science*, *11*(4), 404–428. doi:10.1287/orsc.11.4.404.14600

Pollock, N., & Cornford, J. (2004). ERP systems and the university as a "unique" organization. *Information Technology & People*, *17*(1), 31–52. doi:10.1108/09593840410522161

Scott, S. V., & Wagner, E. L. (2003). Networks, negotiations, and new times: the implementation of enterprise resource planning into an academic administration. *Information and Organization*, *13*(4), 285–313. doi:10.1016/S1471-7727(03)00012-5

Soh, C., & Sia, S. K. (2004). An institutional perspective on sources of ERP package-organisation misalignments. *The Journal of Strategic Information Systems*, *13*(4), 375–397. doi:10.1016/j.jsis.2004.11.001

Swanson, E. B. (2003). Talking the IS innovation walk. In E. H. Wynn, E. A. Whitley, M. D. Myers & J. I. De Gross (Eds.), *Global and organizational siscourse about information technology* (pp. 15-31). Boston: Kluwer.

Trist, E., & Bamforth, K. (1951). Some social and psychological consequences of the longwall method of coal-getting. *Human Relations*, *4*(1), 3–38. doi:10.1177/001872675100400101

Van Amelsvoort, P. (2000). *The design of work and organisation, the modern socio-technical systems approach.* Vlijmen: ST Groep.

Van Eijnatten, F. M. (1993). *The paradigm that changed the work place*. Stockholm/Assen: Arbetslivscentrum/van Gorcum.

Van Eijnatten, F. M., & Van der Zwaan, A. H. (1998). The Dutch IOR approach to organizational design: An alternative to business process re-engineering? *Human Relations*, *51*(3), 289–318.

KEY TERMS

Control Structure: The structure (logical set up and distribution among jobs) of control tasks that together try to safeguard the intended outcomes of an organization.

ERP-Systems: Enterprise resource planning (ERP) systems attempt to integrate several data sources and processes of an organization into a unified system. A key ingredient of most ERP systems is the use of a unified database to store data for the various system modules.

Information Structure: Information structure refers to all information that is relevant to the organization's strategic and operational processes and decision making, that can be stored, used and managed by Information Systems and Information Technology (IS/IT).

Modern Socio-Technology: Modern Sociotechnology (MST) is a Dutch variant of the classical socio-technical systems design (STSD) that focuses on organizational design.

Organizational Design: The process of setting up (designing) the structure of transformations, its coordination, control and the information flows needed to manage the transformation according to the organizational strategy.

Production Structure: The structure (physical lay-out and interdependence relations) of the transformations that together result in the constitution of the goods and services that an organization intents to deliver to their customers.

Teamworking: Teamworking involves a group of workers, generally between 4 and 20 persons, responsible for a rounded-off part of the production process, and entitled to take certain decisions autonomously.

This work was previously published in Handbook of Research on Socio-Technical Design and Social Networking Systems, edited by B. Whitworth; A. de Moor, pp. 416-428, copyright 2009 by Information Science Reference (an imprint of IGI Global).