TOWARDS A BETTER UNDERSTANDING OF STAKEHOLDERS' ROLES IN SOFTWARE CUSTOMIZATION

Carlo Gabriel Porto Bellini[a], João Luiz Becker[b] and Denis Borenstein[c]

[a] School of Management, Federal University of Rio Grande do Sul, Porto Alegre, Brazil
cgpbellini@ea.ufrgs.br
[b] School of Management, Federal University of Rio Grande do Sul, Porto Alegre, Brazil
jlbecker@ea.ufrgs.br
[c] School of Management, Federal University of Rio Grande do Sul, Porto Alegre, Brazil
denisb@ea.ufrgs.br

ABSTRACT
This paper discusses an ongoing research aimed at developing socio-technical metrics for managing the participation of customers in software customization. The particular concern here is the *customer team* – professionals from the client organization of customized software projects who are assigned special business and IT roles for interacting with outsourced developers in such projects. The research is grounded on current gaps in several knowledge fields, as well as on specific claims of the software industry, being qualitative in nature – including a conceptual model, one case study, and one action research. The theoretical and methodological frameworks are discussed, and a further paper under development is meant to discuss preliminary empirical results.

Keywords: *software customization, customer teams, metrics, customer-supplier relationship.*

1. INTRODUCTION

It is still not common in research on software quality to delve into non-technical issues (Ravichandran and Rai, 2000). In the particular case of implementing customized software, the field is also not completely aware of the importance of managing customers with a formal and objective set of measures accounting for their responsibility in projects. *Customized software* – roughly defined, the one whose source code is developed according to each customer's demands – deserves unique attention from developers regarding business processes, culture, strategy, assets and other aspects of the client organization. However, results will be ineffective if all the hard work is to be done by outsourced developers alone.

The present work aims to help the information systems (IS) field by introducing a research framework for improving our knowledge and managerial practices regarding the participation of customers in the development of customized software. We focus on the *customer team* – professionals from the client organization of customized software projects who are assigned special business and information technology (IT) roles for interacting with outsourced developers in such projects. Both customer and outsourced developer teams share project authority and responsibility, and of particular concern here is the development of managerial metrics for process quality to be complied with by customer teams in customized software projects.

Expected benefits from this research cover a wide range of academic and industry interests, from which we highlight the following:

- greater transparency and accuracy in contracting the participation of customer teams;
- real-time assessment of customer team performance (a side effect would possibly be greater academic interest in studying the satisfaction of outsourced developers who interact with customer teams);

- better estimation of the actual performance of outsourced developers, from comparing their performance to that of the customer teams which they interact with and to the overall performance of the projects they are engaged in;
- informed distribution of people (from their historical performance) in customer teams;
- anticipated knowledge of customer teams about the performance criteria from which they will be evaluated by employers (the client organizations); and
- the building of a rationale unifying areas of great interest for the IS field, including customization, quality management, seller-buyer interaction, and teamwork.

The methodological steps towards developing customer metrics include building a conceptual model for the participation of customers in software customization, and performing one case study and one action research, in order to:

- understand the fundamentals of joint work in software customization;
- unify theoretical and practitioners' contributions for managing software teams;
- gather perceptions about practices and needs of customers and developers of customized software for managing their teams;
- identify actual practices of joint work between customer and developer teams in customized software projects; and
- build and validate a set of metrics for managing the participation of customer teams in customized software projects.

The research is qualitative in nature, although it includes a set of causal relations. Further research should be conducted to test hypotheses on those relations.

The paper is structured as follows: first, the theoretical support for the research is discussed, with special focus on software development, management, customization, outsourcing, quality, teams and metrics; second, the research's conceptual model is presented, depicting the main theoretical elements and assumptions; finally, we propose a set of methodological procedures for implementing the research.

2. THEORETICAL BACKGROUND

The research benefited from several heterogeneous knowledge fields, like those of measurement (e.g., measurement theory, data collection/analysis, and software metrics), strategy (e.g., business orientation), information systems (e.g., technological determinism, productivity paradox, duality of technology, and business-IT fit), knowledge management, organizational learning and change (e.g., experiential learning, incremental change, and institutionalization), work-systems planning (e.g., socio-technical perspective, virtual organizations, and organizational networks), outsourcing and contracts (e.g., partnerships, transaction costs economics, and innovation), software engineering (e.g., methodologies, best practices, and process/product quality), engineering (e.g., risks, customization, integrated product development, and collaborative development), and marketing (e.g., relationship marketing, and new product development). Key contributions of some of these fields for our research are commented below.

2.1. Software Development

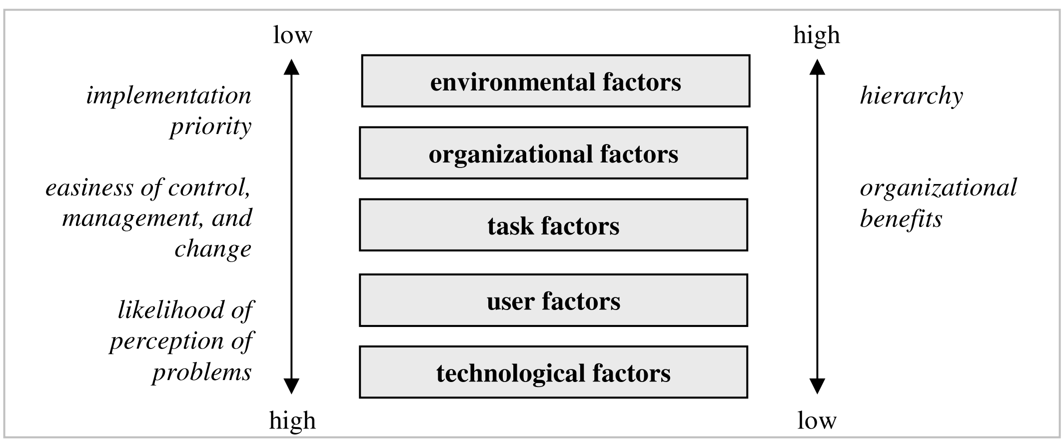
We focus on software in the form of a computer algorithm. We also see software as a product (Nidumolu and Knotts, 1998), notwithstanding its intangible nature (Smith and Keil, 2003) and some services-like attributes it presents (Palvia et al., 2001). From the software concept, we propose an *information system* (IS) to be a software application which is central to a company's business processes (Chan et al., 1997; Sabherwal and Chan, 2001). This definition does not differ significantly from the literature (for instance, found in Alter, 1996, Lee, 1999, and Bednar, 2000). When we then refer to "software", we are meaning the software component of an IS.

The literature on software development is abundant (Pressman, 2001). IS development should be systematic and independently repeatable, aiming at quality (Osmundson et al., 2003). It may be conceptualized as unfolding through the following general steps: pre-development investigation, conceptual design, design and product engineering, systems test, development, and manufacturing (Terwiesch and Loch, 1999).

But problems arise soon in the process of developing software (Burchill and Fine, 1997). Theoretical gaps in software management (Ravichandran and Rai, 2000) – for instance in assessing project management quality (Osmundson et al., 2003) – constrain our understanding and improvement of the whole process. Even software engineering, which is quality driven (Avison and Fitzgerald, 1999), is clearly short sighted due to prioritizing technical issues (Glass et al., 2002) – such a bias is present in the whole software development field (Ravichandran and Rai, 2000; Palvia et al., 2001).

Consequently, there is no surprise in the amount of research reporting problems in software development (e.g., Stamelos et al., 2003, Osmundson et al., 2003, Smith and Keil, 2003, Hoving, 2003, Shaw, 2003, Clegg et al., 2003, Sharma and Rai, 2003, Pich et al., 2002, Ropponen and Lyytinen, 2000, Ravichandran and Rai, 2000, Faraj and Sproull, 2000, Krishnan et al., 2000, Keil et al., 2000, Guinan et al., 1998, Chatzoglou and Macaulay, 1997, and Clark, 1997), most of them relating to productivity, schedule, budget, functionality, and customer satisfaction. Nevertheless, risk management is but a relative new trend (Sauer, 1999; Ropponen and Lyytinen, 2000), and with the growing complexity of systems (Avison and Fitzgerald, 1999; Church and Te Braake, 2001) and the fast pace of technology (Currie and Glover, 1999) rethinking the software process is in need.

In order to find the underlying causes for this, Shaw (2003) proposed that perceived and actual problems may differ. Figure 1 suggests that factors influencing the software process are in reverse order to their implementation priority in practice. This means that the most important factors (higher in the hierarchy) are seldom implemented, which is an explanation for why most projects fail, as well as for why technological issues (appearing at the bottom) are the first – and sometimes the only – concern in projects.



Source: adapted from Shaw (2003).

Figure 1. Hierarchy of factors influencing IS implementation.

2.2. Software Customization and Outsourcing

Outsourcing is a means to developing software rooted in the efficiency imperative (Anderson, 2002). In fact, even software organizations may outsource the development process, since no single company has all the required market capabilities. Outsourced software projects may involve (but not necessarily involve) building one-of-a-kind products, as well as customized products may be (but not necessarily are) developed under an outsourcing contract. Our research, however, focuses on outsourcing agreements for developing customized IS – *customization* understood as the degree to which software development can be tailored to individual project needs (Nidumolu and Knotts, 1998; Stamelos et al., 2003), or to each client's needs and budget.

Ultimately, a customized IS contains a significant software component individually developed for each situation (see Figure 2).

Customized IS products are aimed at matching IS to an organization's business processes, although it is recognized that a perfect match is virtually impossible. Such a matching is assumed to be a key to the company's successful performance – that is one reason why an organization is unlikely to perceive benefits when imitating competitors (Sabherwal and Chan, 2001). Additionally, it is well recognized that customized products benefit both customer and supplier (Stump et al., 2002), supporting a win-win business perspective (Blackstone Jr. et al., 1997).

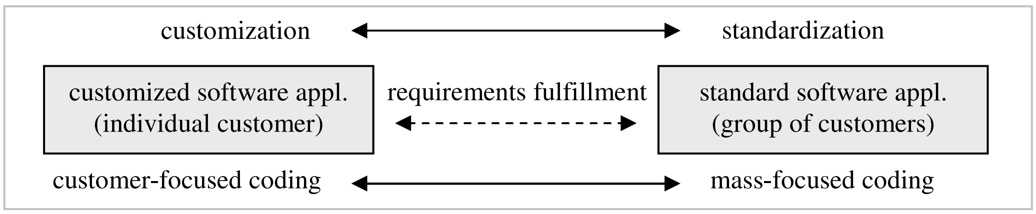


Figure 2. Requirements fulfillment continuum.

Collaborative work is the starting point for developing customized IS, and partnerships should be the major driver of its design (Anderson, 2002). Partnerships involve trust, cooperation and teamwork (Naoum, 2003), and, in the case of collaborative development of new products – a partnership-like arrangement based on active roles performed by both seller and buyer (Athaide and Stump, 1999) –, sellers and buyers are interdependent, minimizing the opportunism of the latter as well as market and technological uncertainties related to transacting innovations (Stump et al., 2002).

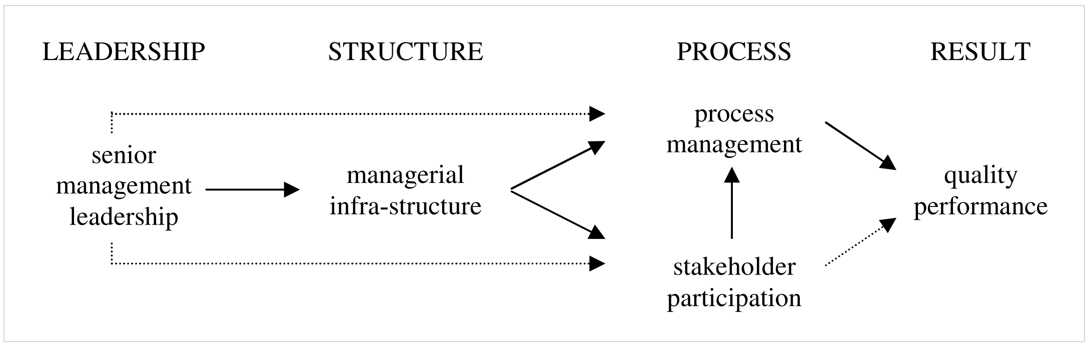
Consequently, customers and manufacturers are closer than ever (Zhu and Kraemer, 2002) and many tasks requiring business skills are even transferred to end-users or performed by mixed teams (Feraud, 2000). In the particular case of software projects, intense customer participation not only leads to a possible better fit between problems and solutions, but also makes it more likely that the user will positively assess the system produced (a causal mapping involving customer participation in projects, willingness to participate, system evaluation, and satisfaction can be built from Burns and Madey, 2001, Kim et al., 2002, Hunton and Beeler, 1997, and Galliers and Swan, 1999).

Although the over-optimism of the outsourcing literature (Lacity and Hirschheim, 1999), drawbacks are naturally present, particularly when involving customization. Among the most critical aspects, outsourcing strategic routines is not at all desirable, since an organization must protect its core business and competence; the technological knowledge gap between customer and supplier may impose important risks for the former (Lacity and Hirschheim, 1999) – therefore formal and detailed agreements between customer and supplier are essential in such a governance design (Ho et al., 2003; Lacity and Hirschheim, 1999; Avison and Fitzgerald, 1999); and other problems involve vendors incurring in high investments for customizing products, disputes around patent ownership, exclusivity clauses, buyer opportunism, and information asymmetry (Stump et al., 2002).

Customers are then learning to negotiate better contracts (Lacity and Willcocks, 1998), while suppliers search for more impartial – interdependent – relations with customers (Stump et al., 2002). Our research contributes for such demands, particularly on building more detailed contracts by improvig the knowledge about actual performance criteria to be observed by customer teams, as well as on how to measure them, judge the measurements and apply results in future projects involving the same customer or its customer team.

2.3. Software Quality

There is clearly a need for improving software management capabilities (Hosalkar and Bowonder, 2000) and accordingly there are opportunities for putting together quality, measurement, and business (Stein, 2001). A first theoretical model unifying such areas seems to be that of Ravichandran and Rai (2000), but it does not differentiate stakeholder types in software development, nor does it take into account the project nature (customized or not, outsourced or not, for instance). As indicated in the model (see Figure 3), senior management does not influence directly the actual participation of each stakeholder group, but instead it sets the overall conditions for the process to flow smoothly and successfully. This is in accordance with one assumption of the socio-technical design of work systems (Trist and Murray, 1993) stating that teamwork should operate in a relatively autonomous fashion.



Note: continuous lines indicate direct positive effect; dotted lines indicate indirect positive effect.
Source: Ravichandran and Rai (2000).

Figure 3. Software quality management.

Improvements in the field of software quality are, however, only slowly changing the realm of software development and application. As already mentioned, this may be related to the technological bias still present in most projects, but a more intriguing – and potentially harmful – subject is the trend of "best practices" in the software industry. There is no doubt that CMM- and ISO-like models provide valid directions for improving the software process and the overall organization for quality; after all, learning from high-performance projects is fundamental (Stensrud and Myrtveit, 2003). But the overconfidence on such models notwithstanding the lack of publicized empirical validation of their effect on company performance (Krishnan et al., 2000; Ravichandran and Rai, 2000) and user satisfaction (Gotzamani and Tsiotras, 2002) may be misleading and, worse, become institutionalized (Galliers and Swan, 1999) due to, for instance, external pressure for certification and advertising (Gotzamani and Tsiotras, 2002). Careful examination of such models and the specific application context is nevertheless necessary in order to prevent that benefits are only perceived by the models' vendors.

If not due to inappropriate instantiation of best practices, problems may also arise from the very models. One case refers to their focus on processes, instead of product orientation (Avison and Fitzgerald, 1999; Baiman et al., 2000). Another situation is that of the *knowledge workers* – like the IS professionals –, who are simply not suited to follow standardized work practices (Scarbrough, 1999). Managing software teams is actually a case in point (Faraj and Sproull, 2000).

Additionally to problems like those briefly mentioned above, the literature on product development (software included) is biased towards setting criteria to be observed only by external contractors (in our case, outsourced software developers). For instance, Baiman et al. (2000) talk about profits for buyers and penalties for suppliers; the literature on *satisfaction* is consistent in its customer focus (e.g., Burns and Madey, 2001; Kim et al., 2002); Avison and Fitzgerald (1999) ask customers to build skills for choosing vendors, but they do not make similar claims on vendors selecting customers with, say, good reputation in collaborative software development; Burchill and Fine (1997) focus on the need of product developers to understand the customer's use context, but they bypass the supporting role played by the latter for a successful project outcome; Bialowas and Tabaszewska (2001) propose a ranking system with which customers can assess suppliers, saying nothing, though, about a possible similar framework to proceed in the reverse way; and Gooden (2001) reminds us that customers require suppliers to be certified in quality management standards, although no similar system for certifying customers exists – notwithstanding, Jiang et al. (2002) alert for the fact that little has been done to prevent user risks before developing systems.

However, applying quality standards only to the contractor signals to a lack of attention to one critical antecedent for the success of customized software: the participation of the customer in its development. This situation assumes dangerous consequences when we realize that IS strategic alignment is a core executive challenge (Chan et al., 1997). Customers are key in such alignment, once IT benefits are only made available from adapting the technology to the organizational context during its implementation (Lassila and Brancheau, 1999). Why, therefore, do we put the customer apart when establishing performance criteria for projects involving external developers *and* customers? Our work argues that there is no reason for this (although we understand the prevailing customer orientation of the industry); nevertheless, we also propose that customer performance metrics should be different from contractor performance metrics, due to the very nature of each team and their role in projects.

2.4. Software Teams

In this section we comment on why teams are the appropriate design for bringing together software professionals in customization efforts, specially why we focus on customer *teams* as the unit of analysis.

If a human *group* can be framed as a number of people in contact to each other (Hofstede, 1994), then a *team* is a group whose complementary skills and purposes of its members make it able to perform tasks for which the members are equally responsible (Church and Te Braake, 2001). In a team approach, conflicts between line and staff are more easily managed (Scarbrough, 1999), an entrepreneurial vision is embraced (Richards and Gupta, 1985) and knowledge creation is nurtured (Leidner and Jarvenpaa, 1995). Peled (2000) further presents extensive attributes commonly found in teams and largely appropriate for supporting our research – the discussion of each one is nevertheless out of the scope here.

We share the vision that software development is best accomplished when performed in teams, and we suggest that software projects should be populated by the following actors: a *sponsor* (customer/user executive with high decision power who sets major project attributes, like deadlines), a *manager* (customer/user assigned to operational and systemic processes, responsible for the fulfillment of the planned issues), a *customer/user team* (business operational people), and a *technical team* (programmers, system analysts, and software engineers). Our previous definition for a customer team is based on matching the capabilities of customer/user and technical teams.

2.5. Software Metrics

Software engineering is about metrics (Fenton and Neil, 1999; Gopal et al., 2002; Kirk and Jenkins, in press), and metrics constitute the dominant approach to measurement (Abran et al., 2003). According to Sommerville (2001) and Pavur et al. (1999), metrics refer to process control and to prediction of product features. Moreover, metrics strengthen research (Straub et al., 2002), represent measurement and feedback mechanisms for making sure that something is being correctly made (Corbin, 1991), and impact engineering and management processes (Gopal et al., 2002).

For the specific case of the software process, metrics are key (Eisner, 1997; Gopal et al., 2002), supporting IS project managers during estimation, technical work, project control, and process improvement (Pressman, 2001). The centrality of metrics in widely deployed models for managing the software process (e.g., CMM) is another testimony of their importance; they are indeed the only current factor differentiating companies with high and low process maturity (Rainer and Hall, 2003).

According to Pfleeger (1995), a metrics plan describing who/how (tools, techniques, people), what (is to be measured), where/when (in the measurement process) and why the metrics (after all, they must be useful for the user – Leung, 2001) is of utmost importance for their effectiveness. Similarly, the abstraction level of measures should be taken into account for composing metrics, since often it is not possible to measure elements of software quality directly; developing a unidimensional measure is actually a vigorous theoretical and statistical modelling task (Segars, 1997). Finally, Leung (2001) calls for a metrics plan that considers the level of aggregation of the software organization's work system (e.g., company, business unit, project group, component). So, contextual mediating contingencies should be understood before any measurement initiative, since projects usually involve very dynamic variables with complex statements and fuzzy relations with each other (Barros et al., 2004).

The literature presents distinct definitions for metrics, and even papers published in top-ranked journals seem not to (explicitly) share a common understanding of the subject – sometimes treating metrics as unidimensional items, sometimes as compound measures. Here we assume the definition given by Pressman (2001):

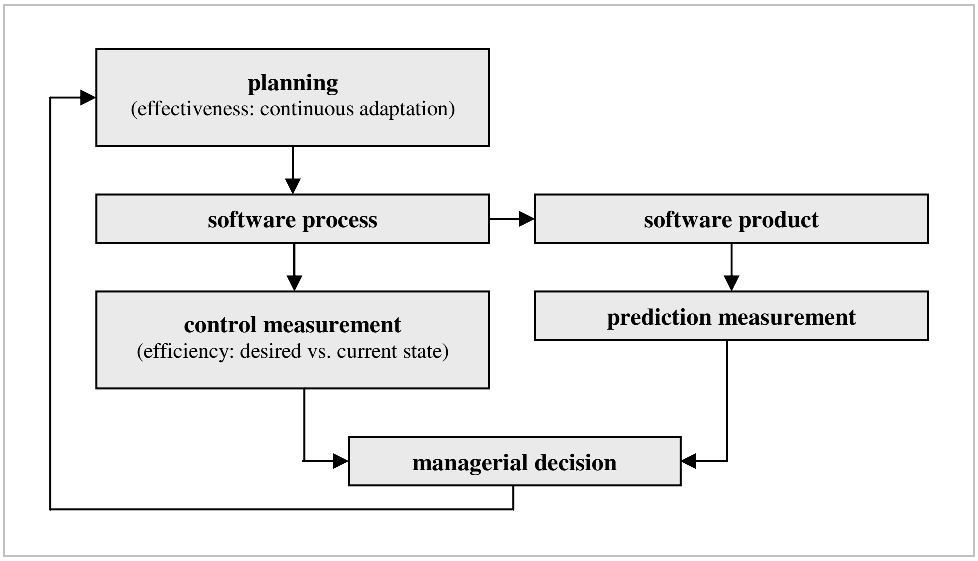
- *measures* are quantitative information of the degree, amount, dimensions, capacity or size of some product's or process' attribute;
- *metrics* are quantitative measures of the degree to which a system, component or process presents some attribute; and
- *indicators* are metrics or combinations of metrics which provide perceptions about a software process, project or product.

2.6. Management of Customer Teams

Historically, companies devote little attention to establishing performance criteria for project and product development processes (Kaplan and Norton, 1996). However, only by continuously identifying and correcting performance deviations an organization is able to consistently compete (Blackstone Jr. et al., 1997). In fact, only what is correctly measured might be correctly managed (Feigenbaum, 2001); on the other hand, without constant measurement and monitoring there can be no process management, and without process management there can be no process improvement (Avison and Fitzgerald, 1999).

According to Jackson and Keys (1984), management and improvement require attention to two process classes: planning – setting the *effective* route towards the objective – and control – setting the *efficient* route. Planning may also be defined as a continuous activity aimed at stimulating creativity and assuring adaptation in highly dynamic and uncertain environments in a participative and interactive process (Richards and Gupta, 1985). Just planning, however, does not necessarily lead to project success, but its absence augments the probability of project failure (Dvir et al., 2003). Control, in turn, may be viewed as knowledge about the distance between the current and the desired state of a system, as well as about who is responsible for adjusting the deviation (Blackstone Jr. et al., 1997). In order to achieve that, a sensor (measuring mechanism) and a governor (decision mechanism) are needed (Douglas, 1983). Figure 4 integrates planning and control into software development.

Projects with increased volatility in requirements – like the software ones – are harder to manage (Keil et al., 2000). In that sense, software teams are usually completely re-designed in each new project, making it difficult for their members to develop a shared work history (Faraj and Sproull, 2000). Moreover, programmers – unlike physicians and engineers – do not have professional standards to follow (Pavur et al., 1999), and this is assumed to play important negative effects in software teams. In fact, software teams are a challenge for management (Boardman, 1995), and one consequence is the common late managerial intervention in problematic projects (Keil et al., 2000).



Source: adapted from Sommerville (2001), Blackstone Jr. et al. (1997), Richards and Gupta (1985), and Jackson and Keys (1984).

Figure 4. Planning and control in software development.

Due to circumstances like the ones shortly discussed above, our research envisions improving our knowledge and managerial practices about the participation of customer teams in software customization projects. We do not aim, however, to achieve extreme power over the routines of teams and individuals through performance metrics, due to efficiency, effectiveness and ethical reasons. For instance:

- some professional knowledge is best for an organization if kept tacit (Bloodgood and Salisbury, 2001);
- for people to take healthy organizational initiatives, they need to feel secure in the work environment (Bednar, 2000), and therefore metrics cannot serve as a punishment tool – just the opposite, metrics play a key role in acknowledging good performance, since appropriate rewards are also needed (Blackstone Jr. et al., 1997; Kirsch et al., 2002; Osmundson et al., 2003; Pich et al., 2002); and
- production control systems often institutionalize the manufacturing of defective products, since people realize that they have a "fault quota" and that it is not their function to identify the occurrence of such faults (Morgan, 1986).

3. RESEARCH MODEL

Figure 5 synthetically depicts the distinguishing elements and assumptions of our conceptual model:

- the two actors – customer and outsourced developer – involved in the projects of interest here, as well as their interacting teams;
- the key (socio-technical) dimensions underlying the set of metrics looked for;
- the presumable positive impact the metrics will exert on the customer (better self-assessment, alignment of the resulting customized software product with business processes, and efficient deployment of resources) and on the external developer (better assessment of the customer's participation in projects); and
- the ultimate goal of developing the metrics – to increase the satisfaction of both customer and outsourced developer regarding the customization of specific software products.

According to the large tradition of organizational theory, the structure of an organization is likely to adapt to contingency factors. Although influencing the work performance and shaping people behavior, it would be nothing more than a means to an end, or arbitrary decisions for facilitating the organizational repertoire of actions (Nelson and Winter, 1982). Technology, in turn, is usually seen as a contingency factor (Donaldson, 1996) influencing the organizational structure. But although the changing nature of technology, it is evident from the literature (e.g., Zack, 1999) that there are some common elements in most technology prescriptions for enabling the knowledge work like that of IS professionals.

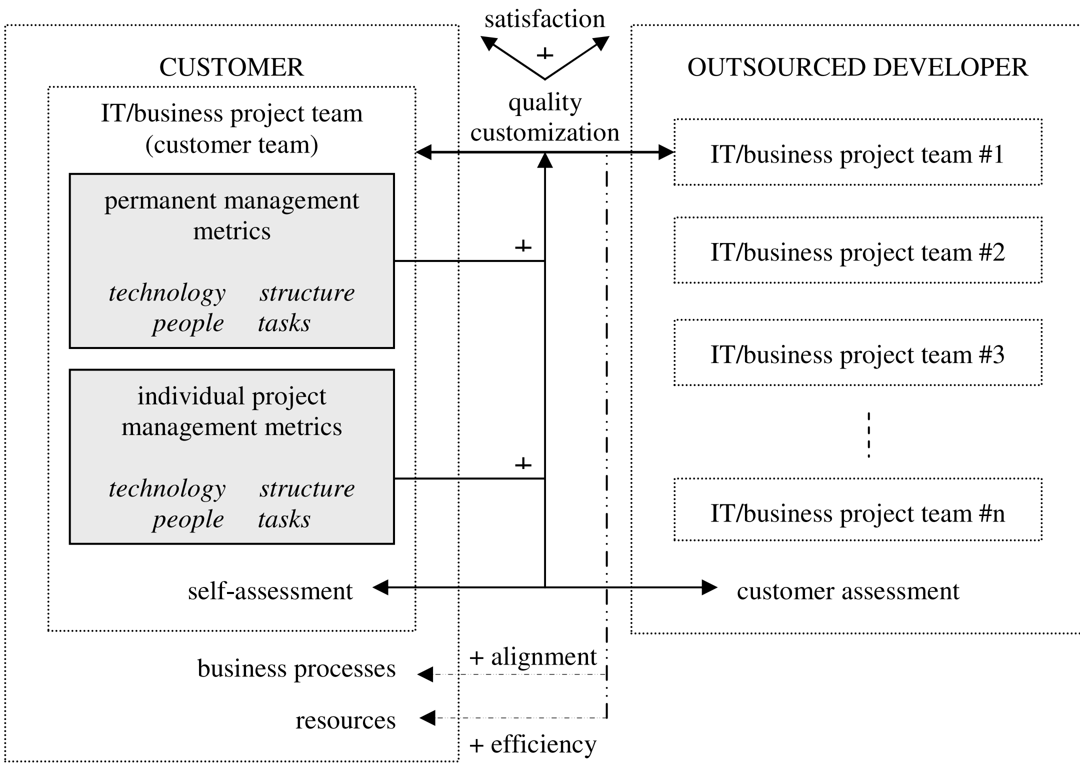


Figure 5. Main elements and assumptions of the conceptual model.

So we take relatively for granted both structure and technology according to some theoretical frameworks from the academic and the industry ("best practices") literature. We then set (but not in this paper, due to its scope) an ideal and adaptable configuration of structure and technology for supporting the work of customer teams. This configuration is not trivial, however, since attention must be paid to the possibility of mutual contingency influences (the causality between structure and technology is not completely known, as pointed out by current researches – e.g., Dolci, 2004). This anticipated definition of structure and technology is indicated in Figure 6 as forming a basis from which to develop software.

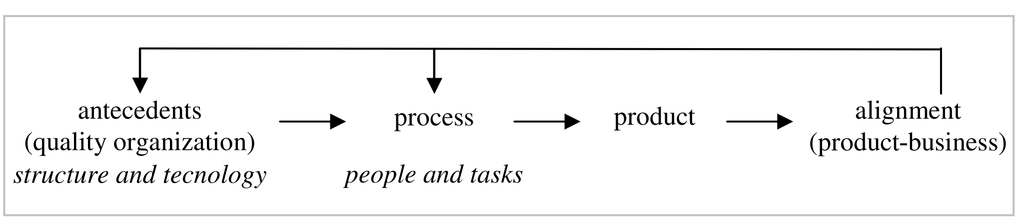


Figure 6. Socio-technical framework for software customization.

Such an assumption – setting relatively stable structural and technological conditions for software customization – is supported by the already commented model of Ravichandran and Rai (2000) in Figure 3, according to which an organizational infra-structure should be set prior to running process management and involving different stakeholders.

4. METHODOLOGY

The empirical part of the research will be developed within the Brazilian software industry – in which one of the authors works since de mid-1990s developing, managing and studying database, Internet and company applications. In order to get a richer picture about industry practices, needs and trends, that author also cooperates in several research endeavors in close proximity to companies from the Pólo de Informática de São Leopoldo – a leading Brazilian IT cluster.

The methodology is broadly represented by a conceptual model developed from a thorough literature review, one case study of a major software customization project (the implementation of an ERP system in a Brazilian university), and one action research within another high-profile software project (the development of an information system for the Brazilian government). Together the methods would support (1) an understanding of the fundamentals of software customization, (2) the gathering of perceptions about practices and needs of customers and developers of customized software for managing their teams, (3) the identification of actual practices of joint work between customer and developer teams in customized software projects, and (4) the building and validation of a set of metrics for managing the participation of customer teams in customized software projects.

The conceptual model, comprising a rationale integrating several areas relevant for software customization, is already accomplished and represents a major achievement of the research. The other two steps will be workable in the next months, and five external researchers and practitioners (in software engineering, marketing, and organizations) will formally validate the various data collection instruments, data analysis procedures, and results.

4.1. Case Study

The case study is aimed at providing insights into current practices and needs of customized-software clients and developers for managing their teams. Towards this end and for contrasting team reports with actual practices – in the sense of espoused/in-use theories (Argyris, 1992) –, open-ended interviews with experienced teams from both sides and the observation of a customized software process will be carried out, and a first nomological net reflecting desired practices for customer teams will then be validated. Two questions are to be answered in the case study:

- how do customer and developer teams perceive their interaction in a software project?
- what are the criteria currently employed by customers and developers of customized software to manage the participation of their teams?

We highlight three fundamental reasons for performing a case study: (1) it helps deepen our knowledge about a given reality (Palvia et al., 2003; Hussey and Hussey, 1997), (2) it does not require all major variables to be known in advance (Boudreau et al., 2001; Benbasat et al., 1987), and (3) it frames complex elements for future investigation (Stake, 2000). These aspects relate to the central argument of our research that the customer is usually seen as an “unsuspected” part in software development when it comes to quality assurance and assessment, and so there is a tradition that must be questioned and investigated from a completely new point of view.

4.2. Action Research

The action research envisions further validation of the set of metrics to be complied with by customer teams. In this method, the researcher is also an actor, meaning that one of the authors will be engaged in ordinary activities of software development – project management, system analysis, and coding. The point here is that the researcher experiences a unique perspective when fully committed to the situation under surveillance – in the particular case, the daily routine of customer and developer teams during the development of customized software. Other procedures to be accomplished in the action research include open-ended interviews with project stakeholders and the use of software tools for gauging project performance. The question to be answered is:

- is the model for managing customer teams appropriate for the needs of clients and outsourced developers?

The action research is anchored in the general beliefs that (1) it is suited for the socio-technical view of software development (Benbasat et al., 1987) and for the (2) power relations between clients and consultants (Eden and Huxham, 1996; Hussey and Hussey, 1997), (3) it contributes for both academic and pragmatic knowledge (Grant and Ngwenyama, 2003) and for the understanding of a social context (Palvia et al., 2003), and (4) superior studies are accomplished by the very actors of a problematic situation (Checkland, 1985).

It is worth noting that a possible confounding variable in the case study and the action research might be low levels of quality of the software development teams which the researchers interact with. Thus, it is necessary to carefully identify the source (customer or outsourced developer team) of project errors and accomplishments, in order to accurately credit performance levels to teams. The socio-technical perspective of the software process, providing systemic and rich variables for analysis, accounts for such a need.

CONCLUDING REMARKS

Managing the participation (authority and responsibility) of customer teams in the development of customized software seems to be neglected in the literature and inaccurately worked out by practitioners, albeit essential for the software industry. Moreover, non-technical elements of the software process still have been regarded as secondary in importance or priority. Our work therefore tries to compensate such a bias by developing a set of socio-technical quality metrics for managing customer teams.

The proposed qualitative research seems to innovate in substantive aspects of the software process, mainly in that customer teams were deemed suited for pursuing performance metrics. Moreover, the very building of a rationale for software customization integrating several knowledge fields is an endeavor without known parallels in the literature.

REFERENCES

Abran, A., Sellami, A. and Suryn, W. (2003), Metrology, Measurement and Metrics in Software Engineering, *9th International Software Metrics Symposium*, Sydney.

Alter, S. (1996), *Information Systems: A Management Perspective*, Benjamin & Cummings, Menlo Park.

Anderson, E. T. (2002), Sharing the Wealth: When Should Firms Treat Customers as Partners?, *Management Science*, 48 (8), 955-971.

Argyris, C. (1992), *On Organizational Learning*, Blackwell Business, Oxford.

Athaide, G. A. and Stump, R. L. (1999), A Taxonomy of Relationship Approaches during Product Development in Technology-based, Industrial Markets, *Journal of Product Innovation Management*, 16 (5), 469-482.

Avison, D. E. and Fitzgerald, G. (1999), Information Systems Development, in Currie, W. L. and Galliers, B. (eds.), *Rethinking Management Information Systems: An Interdisciplinary Perspective*, Oxford University Press, New York, 250-278.

Baiman, S., Fischer, P. E. and Rajan, M. V. (2000), Information, Contracting, and Quality Costs, *Management Science*, 46 (6), 776-789.

Barros, M. O., Werner, C. M. L. and Travassos, G. H. (2004), Supporting Risks in Software Project Management, *The Journal of Systems and Software*, 70, 21-35.

Bednar, P. M. (2000), A Contextual Integration of Individual and Organizational Learning Perspectives as Part of IS Analysis, *Informing Science*, 3 (3), 145-156.

Benbasat, I., Goldstein, D. K. and Mead, M. (1987), The Case Research Strategy in Studies of Information Systems, *MIS Quarterly*, 11 (3), 369-386.

Bialowas, P. and Tabaszewska, E. (2001), How to Evaluate the Internal Customer-supplier Relationship, *Quality Progress*, 34 (7), 63-67.

Blackstone Jr., J. H., Gardiner, L. R. and Gardiner, S. C. (1997), A Framework for the Systemic Control of Organizations, *International Journal of Production Research*, 35 (3), 597-609.

Bloodgood, J. M. and Salisbury, W. D. (2001), Understanding the Influence of Organizational Change Strategies on Information Technology and Knowledge Management Strategies, *Decision Support Systems*, 31, 55-69.

Boardman, J. T. (1995), Wholes and Parts – A Systems Approach, *IEEE Transactions on Systems, Man, and Cybernetics*, 25 (7), 1150-1161.

Boudreau, M.-C., Gefen, D. and Straub, D. W. (2001), Validation in Information Systems Research: A State-of-the-art Assessment, *MIS Quarterly*, 25 (1), 1-16.

Burchill, G. and Fine, C. H. (1997), Time Versus Market Orientation in Product Concept Development: Empirically-based Theory Generation, *Management Science*, 43 (4), 465-478.

Burns, J. and Madey, G. R. (2001), A Framework for Effective User Interface Design for Web-based Electronic Commerce Applications, *Informing Science*, 4 (2), 67-75.

Chan, Y. E., Huff, S. L., Barclay, D. W. and Copeland, D. G. (1997), Business Strategic Orientation, Information Systems Strategic Orientation, and Strategic Alignment, *Information Systems Research*, 8 (2), 125-150.

Chatzoglou, P. and Macaulay, L. A. (1997), The Importance of Human Factors in Planning the Requirements Capture Stage of a Project, *International Journal of Project Management*, 15 (1), 39-53.

Checkland, P. (1985), Achieving "Desirable and Feasible" Change: An Application of Soft Systems Methodology, *Journal of the Operational Research Society*, 36 (9), 821-831.

Church, K. and Te Braake, G. (2001), The Future of Software Development, in Khosrowpour, M. (ed.), *Managing Information Technology in a Global Economy*, Idea Group Publishing, Hershey, 321-325.

Clark, B. K. (1997), *The Effects of Software Process Maturity on Software Development Effort*, PhD. dissertation (Computer Science), University of Southern California.

Clegg, C., Axtell, C., Damodaran, L., Farbey, B. et al. (2003), *The Performance of Information Technology and the Role of Human and Organisational Factors: Report to the Economic and Social Research Council, UK*, http://www.system-concepts.com/stds/clegg.html (accessed Apr 15, 2003).

Corbin, D. S. (1991), Establishing the Software Development Environment, *Journal of Systems Management*, 42 (9), 28-31.

Currie, W. L. and Glover, I. A. (1999), Hybrid Managers: An Example of Tunnel Vision and Regression in Management Research, in Currie, W. L. and Galliers, B. (eds.), *Rethinking Management Information Systems: An Interdisciplinary Perspective*, Oxford University Press, New York, 417-443.

Dolci, D. B. (2004), A Dualidade entre Organizações e Tecnologia da Informação: Um Ensaio Teórico com Vistas a Explorar a Influência das Mudanças Organizacionais nos Sistemas de Informação. PhD proposal (Information Systems), Federal University of Rio Grande do Sul, Porto Alegre.

Donaldson, L. (1996), The Normal Science of Structural Contingency Theory, in Clegg, S. R., Hardy, C. and Nord, W. R. (eds.), *Handbook of Organization Studies*, Sage, London.

Douglas, A. S. (1983), Systems Analysis and the Computer, *Journal of the Operational Research Society*, 34 (8), 779-785.

Dvir, D., Raz, T. and Shenhar, A. J. (2003), An Empirical Analysis of the Relationship Between Project Planning and Project Success, *International Journal of Project Management*, 21, 89-95.

Eden, C. and Huxham, C. (1996), Action Research for the Study of Organizations, in Clegg, S. R. Hardy, C. and Nord, W. R. (eds.), *Handbook of Organization Studies*, Sage, London, 526-542.

Eisner, H. (1997), *Essentials of Project and Systems Engineering Management*, John Wiley & Sons, New York.

Faraj, S. and Sproull, L. (2000), Coordinating Expertise in Software Development Teams, *Management Science*, 46 (12), 1554-1568.

Feigenbaum, A. V. (2001), How to Manage for Quality in Today's Economy, *Quality Progress*, 34 (5), 26-27.

Fenton, N. E. and Neil, M. (1999), Software Metrics: Successes, Failures and New Directions, *The Journal of Systems and Software*, 47, 149-157.

Feraud, G. (2000), What Makes IT Professionals Tick?, in Marchand, D. A. and Davenport, T. H. (eds.), *Mastering Information Management: Complete MBA Companion in Information Management*, Pearson Education, 83-86.

Galliers, B. and Swan, J. (1999), Information Systems and Strategic Change: A Critical Review of Business Process Re-engineering, in Currie, W. L. and Galliers, B. (eds.), *Rethinking Management Information Systems: An Interdisciplinary Perspective*, Oxford University Press, New York, 361-387.

Glass, R. L., Vessey, I. and Ramesh, V. (2002), Research in Software Engineering: An Analysis of the Literature, *Information and Software Technology*, 44, 491-506.

Gooden, R. (2001), How a Good Quality Management System Can Limit Lawsuits, *Quality Progress*, 34 (6), 55-59.

Gopal, A., Krishnan, M. S., Mukhopadhyay, T. and Goldenson, D. R. (2002), Measurement Programs in Software Development: Determinants of Success, *IEEE Transactions on Software Engineering*, 28, 863-875.

Gotzamani, K. D. and Tsiotras, G. D. (2002), The True Motives Behind ISO 9000 Certification: Their Effect on the Overall Certification Benefits and Long Term Contribution Towards TQM, *International Journal of Quality & Reliability Management*, 19 (2), 151-169.

Grant, D. and Ngwenyama, O. (2003), A Report on the Use of Action Research to Evaluate a Manufacturing Information Systems Development Methodology in a Company, *Information Systems Journal*, 13 (1), 21-35.

Guinan, P. J., Cooprider, J. G. and Faraj, S. (1998), Enabling Software Development Team Performance during Requirements Definition: A Behavioral Versus Technical Approach, *Information Systems Research*, 9 (2), 101-125.

Ho, V. T., Ang, S. and Straub, D. (2003), When Subordinates Become IT Contractors: Persistent Managerial Expectations in IT Outsourcing, *Information Systems Research*, 14 (1), 66-86.

Hofstede, G. H. (1994), *Cultures and Organizations*, HarperCollinsPublishers, London.

Hosalkar, A. and Bowonder, B. (2000), Software Development Management: Critical Success Factors, *International Journal of Technology Management*, 19 (7/8), 760-772.

Hoving, R. (2003), Executive Response: Project Management Process Maturity as a "Secret Weapon", *MIS Quarterly Executive*, 2 (1), 29-30.

Hunton, J. E. and Beeler, J. D. (1997), Effects of User Participation in Systems Development: A Longitudinal Field Experiment, *MIS Quarterly*, 21 (4), 359-388.

Hussey, J. and Hussey, R. (1997), *Business Research*, Palgrave Macmillan, New York.

Jackson, M. C. and Keys, P. (1984), Towards a System of Systems Methodologies, *Journal of the Operational Research Society*, 35 (6), 473-486.

Jiang, J. J., Klein, G., Chen, H.-G. and Lin, L. (2002), Reducing User-related Risks during and prior to System Development, *International Journal of Project Management*, 20 (7), 507-515.

Kaplan, R. S. and Norton, D. P. (1996), *The Balanced Scorecard*, Harvard Business School Press, Boston.

Keil, M., Mann, J. and Rai, A. (2000), Why Software Projects Escalate: An Empirical Analysis and Test of Four Theoretical Models, *MIS Quarterly*, 24 (4), 631-664.

Kim, J., Lee, J., Han, K. and Lee, M. (2002), Businesses as Buildings: Metrics for the Architectural Quality of Internet Businesses, *Information Systems Research*, 13 (3), 239-254.

Kirk, S. R. and Jenkins, S. (in press), Information Theory-based Software Metrics and Obfuscation, *The Journal of Systems and Software*.

Kirsch, L. J., Sambamurthy, V., Ko, D.-G. and Purvis, R. L. (2002), Controlling Information Systems Development Projects: The View from the Client, *Management Science*, 48 (4), 484-498.

Krishnan, M. S., Kriebel, C. H., Kekre, S. and Mukhopadhyay, T. (2000), An Empirical Analysis of Productivity and Quality in Software Products, *Management Science*, 46 (6), 745-759.

Lacity, M. C. and Hirschheim, R. (1999), Information Technology Outsourcing: What Problems Are We Trying to Solve?, in Currie, W. L. and Galliers, B. (eds.), *Rethinking Management Information Systems: An Interdisciplinary Perspective*, Oxford University Press, New York, 326-360.

Lacity, M. C. and Willcocks, L. P. (1998), An Empirical Investigation of Information Technology Sourcing Practices: Lessons from Experience, *MIS Quarterly*, 22 (3), 363-408.

Lassila, K. S. and Brancheau, J. C. (1999), Adoption and Utilization of Commercial Software Packages: Exploring Utilization Equilibria, Transitions, Triggers, and Tracks, *Journal of Management Information Systems*, 16 (2), 63-90.

Lee, A. S. (1999), Researching MIS, in Currie, W. L. and Galliers, B. (eds.), *Rethinking Management Information Systems: An Interdisciplinary Perspective*, Oxford University Press, New York, 7-27.

Leidner, D. E. and Jarvenpaa, S. L. (1995), The Use of Information Technology to Enhance Management School Education: A Theoretical View, *MIS Quarterly*, 19 (3), 265-291.

Leung, H. K. N. (2001), Quality Metrics for Intranet Application, *Information & Management*, 38, 137-152.

Morgan, G. (1986), *Images of Organization*, Sage, Thousand Oaks.

Naoum, S. (2003), An Overview into the Concept of Partnering, *International Journal of Project Management*, 21, 71-76.

Nelson, R. and Winter S. G. (1982), *An Evolutionary Theory of Economic Change*, Belknap Press, Cambridge.

Nidumolu, S. R. and Knotts, G. W. (1998), The Effects of Customizability and Reusability on Perceived Process and Competitive Performance of Software Firms, *MIS Quarterly*, 22 (2), 105-137.

Osmundson, J. S., Michael, J. B., Machniak, M. J. and Grossman, M. A. (2003), Quality Management Metrics for Software Development, *Information & Management*, 40 (8), 799-812.

Palvia, P., Mao, E., Salam, A. F. and Soliman, K. S. (2003), Management Information Systems Research: What's There in a Methodology?, *Communications of the AIS*, 11, 289-309.

Palvia, S. C., Sharma, R. S. and Conrath, D. W. (2001), A Socio-technical Framework for Quality Assessment of Computer Information Systems, *Industrial Management & Data Systems*, 101 (5), 237-251.

Pavur, R., Jayakumar, M. and Clayton, H. (1999), Software Testing Metrics: Do They Have Merit?, *Industrial Management & Data Systems*, 99 (1), 5-10.

Peled, A. (2000), Creating Winning Information Technology Project Teams in the Public Sector. *Team Performance Management*, 6 (1/2), 6-14.

Pfleeger, S. L. (1995), Maturity, Models, and Goals: How to Build a Metrics Plan, *The Journal of Systems and Software*, 31, 143-155.

Pich, M. T., Loch, C. H. and De Meyer, A. (2002), On Uncertainty, Ambiguity, and Complexity in Project Management, *Management Science*, 48 (8), 1008-1023.

Pressman, R. S. (2001), *Software Engineering: A Practitioner's Approach*, McGraw-Hill, New York.

Rainer, A. and Hall, T. (2003), Quantitative and Qualitative Analysis of Factors Affecting Software Processes, *The Journal of Systems and Software,* 66, 7-21.

Ravichandran, T. and Rai, A. (2000), Quality Management in Systems Development: An Organizational System Perspective, *MIS Quarterly*, 24 (3), 381-415.

Richards, L. D. and Gupta, S. K. (1985), The Systems Approach in an Information Society: A Reconsideration, *Journal of the Operational Research Society*, 36 (9), 833-843.

Ropponen, J. and Lyytinen, K. (2000), Components of Software Development Risk: How to Address Them? A Project Manager Survey, *IEEE Transactions on Software Engineering*, 26 (2), 98-112.

Sabherwal, R. and Chan Y. E. (2001), Alignment between Business and IS Strategies: A Study of Prospectors, Analyzers, and Defenders, *Information Systems Research*, 12 (1), 11-33.
Sauer, C. (1999), Deciding the Future for IS Failures: Not the Choice You Might Think, in Currie, W. L. and Galliers, B. (eds.), *Rethinking Management Information Systems: An Interdisciplinary Perspective*, Oxford University Press, New York, 279-309.
Scarbrough, H. (1999), The Management of Knowledge Workers, in Currie, W. L. and Galliers, B. (eds.), *Rethinking Management Information Systems: An Interdisciplinary Perspective*, Oxford University Press, New York, 474-496.
Segars, A. H. (1997), Assessing the Unidimensionality of Measurement: A Paradigm and Illustration within the Context of Information Systems Research, *Omega*, 25, 107-121.
Sharma, S. and Rai, A. (2003), An Assessment of the Relationship between ISD Leadership Characteristics and IS Innovation Adoption in Organizations, *Information & Management*, 40 (5), 391-401.
Shaw, N. G. (2003), Identifying Relationships among Factors in IS Implementation, *Communications of the AIS*, 11, 155-165.
Smith, H. J. and Keil, M. (2003), The Reluctance to Report Bad News on Troubled Software Projects: A Theoretical Model, *Information Systems Journal*, 13 (1), 69-95.
Sommerville, I. (2001), *Software Engineering*, Addison-Wesley, Harlow.
Stake, R. E. (2000), Case Studies, in Denzin, N. K. and Lincoln, Y. S. (eds.), *Handbook of Qualitative Research*, Sage, Thousand Oaks, 435-454.
Stamelos, I., Angelis, L., Morisio, M., Sakellaris, E. et al. (2003), Estimating the Development Cost of Custom Software, *Information & Management*, 40 (8), 729-741.
Stein, P. (2001), Measurements for Business, *Quality Progress*, 34 (2), 29-32.
Stensrud, E. and Myrtveit, I. (2003), Identifying High Performance ERP Projects, *IEEE Transactions on Software Engineering*, 29 (5), 398-416.
Straub, D. W., Hoffman, D. L., Weber, B. W. and Steinfield, C. (2002), Toward New Metrics for Net-enhanced Organizations, *Information Systems Research*, 13, 227-238.
Stump, R. L., Athaide, G. A. and Joshi, A. W. (2002), Managing Seller-buyer New Product Development Relationships for Customized Products: A Contingency Model Based on Transaction Cost Analysis and Empirical Test, *Journal of Product Innovation Management*, 19 (6), 439-454.
Terwiesch, C. and Loch, C. H. (1999), Measuring the Effectiveness of Overlapping Development Activities, *Management Science*, 45 (4), 455-465.
Trist, E. and Murray, H. (eds.) (1993), *The Social Engagement of Social Science: A Tavistock Anthology*, University of Pennsylvania Press, Philadelphia.
Zack, M. H. (1999), Managing Codified Knowledge, *Sloan Management Review*, 40 (4), 45-58.
Zhu, K. and Kraemer, K. L. (2002), E-commerce Metrics for Net-enhanced Organizations: Assessing the Value of E-commerce to Firm Performance in the Manufacturing Sector, *Information Systems Research*, 13 (3), 275-295.

Received: March 25th 2004

Accepted in final format: July 30th 2004 after one revision

About the authors:

Carlo Gabriel Porto Bellini is a PhD candidate in Management at Federal University of Rio Grande do Sul, Brazil. He holds a M.Sc. degree in Management and a B.Sc. degree in Computer Science from the same university. His research interests concern software development, Internet-mediated communities, research methodology, and ethics.

João Luiz Becker earned a PhD degree from UCLA Anderson School of Management in 1986, following a M.Sc. degree from the Institute of Pure and Applied Mathematics, Rio de Janeiro, Brazil, in 1978. His main research interests are in the field of operations research, applied quantitative methods and decision analysis.

Denis Borenstein received his PhD in Management Science from University of Strathclyde, UK, and a B.Sc. in Marine Engineering from Federal University of Rio de Janeiro, Brazil. His research interests are in the areas of modeling and simulation of manufacturing/logistical systems, mass customization, flexible manufacturing systems, and the development of decision support systems.