

 Open access • Journal Article • DOI:10.18848/1832-3669/CGP/V03I01/55654

## **(Dis)trust in Software Projects: a Thrice Told Tale: On Dynamic Relationships between Software Engineers, IT Project Managers, and Customers — [Source link](#)**

Dominika Latusek, Dariusz Jemielniak

**Published on:** 01 Jan 2007 - The International Journal of Technology, Knowledge, and Society: Annual Review (Common Ground Research Networks)

**Topics:** Software review, Software development and Distrust

Related papers:

- [Exploring Complex Organizations: A Cultural Perspective](#)
- [The Discovery of Grounded Theory: Strategies for Qualitative Research.](#)
- [The powers of association](#)
- [Organizational ethnography: Methods and inspirations](#)
- [Kultura – odkrywana czy konstruowana?](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/dis-trust-in-software-projects-a-thrice-told-tale-on-dynamic-22uke6u2yo>

The International  
**JOURNAL**  
*of*  
TECHNOLOGY,  
KNOWLEDGE  
& SOCIETY

Volume 3, Number 1

(Dis)trust in Software Projects: a Thrice Told Tale:  
On Dynamic Relationships between Software  
Engineers, IT Project Managers, and Customers

Dominika Latusek and Dariusz Jemielniak

THE INTERNATIONAL JOURNAL OF TECHNOLOGY, KNOWLEDGE AND SOCIETY  
<http://www.Technology-Journal.com>

First published in 2007 in Melbourne, Australia by Common Ground Publishing Pty Ltd  
[www.CommonGroundPublishing.com](http://www.CommonGroundPublishing.com).

© 2007 (individual papers), the author(s)  
© 2007 (selection and editorial matter) Common Ground

All rights reserved. Apart from fair use for the purposes of study, research, criticism or review as permitted under the Copyright Act (Australia), no part of this work may be reproduced without written permission from the publisher. For permissions and other inquiries, please contact <[cg-support@commongroundpublishing.com](mailto:cg-support@commongroundpublishing.com)>.

ISSN: 1832-3669  
Publisher Site: <http://www.Technology-Journal.com>

The THE INTERNATIONAL JOURNAL OF TECHNOLOGY, KNOWLEDGE AND SOCIETY is a peer refereed journal. Full papers submitted for publication are refereed by Associate Editors through anonymous referee processes.

Typeset in Common Ground Markup Language using CGCreator multichannel typesetting system  
<http://www.CommonGroundSoftware.com>.

# (Dis)trust in Software Projects: a Thrice Told Tale: On Dynamic Relationships between Software Engineers, IT Project Managers, and Customers

Dominika Latusek, Kozminski Business School (LKAEM), Poland

Dariusz Jemielniak, Kozminski Business School (LKAEM), Poland

*Abstract: Software development traditionally has been a field particularly prone to delays, exceeding budgets, and misunderstandings (May 1998; Connel, 2001; Humphrey, 2002; Goodwin, 2002; Kesteloot, 2003). Only 1/4 of IT projects is completed successfully – i.e. in time, with the assumed cost, and fulfilling the promised functionality (Smith and Keil, 2003). Although there is some improvement over the last years, software creation is still one of the most unpredictable businesses in the world. It should not be surprising then that high-tech environment often is described as stressful and demanding (Kunda, 1992; Hochschild, 1997; Perlow, 1998; Cooper, 2000; Jemielniak, 2005). It is the nest of the so-called normative control: management by indirect reign, through the means of evoking motivation and dedication of the employee, rather than physical direction and supervision (Kanter, 1977; Mintzberg, 1998). On the other hand, many things have to be taken on face value: program code is a black-box for anyone but the programmer, only the client understands the real needs for the software, only the manager realizes what the real deadlines and constraints are, etc. In this light it may be interesting to delve into the issue of actual organizational practices in IT environment. Thus, in this paper we will try to analyze how the trust in software projects is understood, perceived and exercised by the key organizational actors involved. According to our research, there are three major groups of actors in software development: managers, clients, and software engineers. These roles play an important defining function in IT reality. The stories told by our interviewees all point to trust, or rather the distrust in software development. In the proposed paper we would like to confront the three views in an ethnographical manner.*

Keywords: Ethnography, High-Tech, Software, Trust, IT, Distrust, Enactment

## Introduction

**S**OFTWARE DEVELOPMENT TRADITIONALLY has been a field particularly prone to misunderstanding, exceeding budgets, and delays (May, 1998; Connel, 2001; Humphrey, 2002; Goodwin, 2002; Kesteloot, 2003). In fact, only ¼ of IT projects is completed successfully – i.e. in time, at the assumed cost, and fulfilling the promised functionality (Smith and Keil, 2003). Although there has been some improvement in such statistics over the years (Standish Group International “Extreme Chaos” Report, 2001), software creation is still one of the most unpredictable businesses in the world. It should not be surprising then that a high-tech environment quite often is described as very stressful and demanding (Kunda, 1992; Hochschild, 1997; Perlow, 1998; Cooper, 2000; Jemielniak, 2005). In such a challenging environment and under high time pressure, trust could be a rare, but still crucial resource (Latusek, 2007). In this paper the authors intend to study organizational practices related to trust-building in project-oriented work in high-tech companies.

IT projects, usually pursued by cross-organizational teams (customers and software providers), are a perfect example of temporary groups, an increasingly common form of organization. According to Meyer-

son et al. (1996) these are characterized by the following features: (1) have a clear time-horizon of functioning, (2) are formed to execute a relatively clear plan, and (3) their success depends on perfect coordination of activities. In the discussion on what ties together people cooperating within such a setting Meyerson et al. (ibid.) write: “*temporary systems exhibit behavior that presupposes trust, yet traditional sources of trust – familiarity, shared experience, reciprocal disclosure, threats and deterrents, fulfilled promises, and demonstrations of nonexploitation of vulnerability – are not obvious in such systems*” (ibid., p. 167). There is trust in such systems (so-called *swift trust*), but has some unusual properties: (1) it flows “*from the nature and magnitude of the interdependence in the setting and the implicit threat that stems from this interdependence*” (ibid, p. 175); (2) it flows from “*dispositions, categorical assumptions and implicit theories*” (ibid., p. 178), because there is almost no opportunity to early monitor the actions of the partners in the temporary groups setting; (3) it makes risk tolerable, i.e. enables cooperation in the setting where perceived risks are particularly high.

Social groups involved in IT project development quite often are antagonized (Jemielniak, 2007), also because of the complex nature of the tasks, involving both individual (silent) and group-based (interactive)



actions (Kociatkiewicz and Kostera, 2003). As the presented research also reveals, the language of all parties engaged in an IT project is not a language of cooperation built on the foundation of trust. In fact, it is about the converse: the discourse of distrust, risk and control. So, considering the general feeling of suspiciousness that pervades the relationships - between the IT provider and the client, as well as within the provider firm (between managers and software engineers) - the study concentrates on the enactment and perception of cooperation in a climate of profound distrust.

In this light, the goal of this article is twofold. First, it lies in presenting the parties' perceptions of each other throughout the projects. What do the everyday encounters of customers, providers and software engineers look like? To what extent their diametrically different views can come to terms? Second, it is intended to investigate the process of cooperation in IT projects, with particular emphasis on (dis)trust. Through a qualitative, strictly local study, the authors hope to be able to capture the dynamics of collaboration, and identify the manifestations of trust, or, alternatively, other elements that secure cooperation. The existing literature on distrust, however limited, may support the conjecture that cooperation without trust is also possible, and - even more - it may be valuable (Hardin 2004; Cook et al. 2005).

## Method

The method of qualitative open-ended unstructured interviews, chosen for this study, is particularly useful for delving into the individual's perceptions and beliefs (Whyte, 1984; Schwartzman, 1993). The presented research is ethnographical (Rosen, 1991), and the aim of the article is to present the understanding of the problem in the eyes of organizational actors, rather than apply the authors' own view of it. Instead of a pre-conceptualized model an ethnographic collage of ostensibly contradictory stories was consciously chosen (Clifford and Marcus, 1986). In place of one idealized story, three different perspectives from the same scene are presented, possibly constructing a fourth one (Wolf, 1992). Thus, the article is purposefully not trying to compose one image, reconciling the diverse perspectives, but rather aiming at confronting and exposing the diametrically different approaches, in normal circumstances silenced out and dismissed (Martin, 1993; Höpfl, 1995).

This study is following the call from Latour (1986) for performative studies of organizational reality and, therefore, assumes the underprivileged role of the researcher (Greenwood and Levin, 1998; Jemielniak, 2006). By grounding the interpretation in the findings (Glaser and Strauss, 1957) the study is placed in the

interpretative, symbolic interactionist, and social constructivist tradition (Czarniawska-Joerges, 1992).

The project takes on the perspective of practitioners: representatives of all three parties engaged in cooperation. In the particular context of trust it allows for various interpretations and a deeper reflection on the perception of the phenomenon (Möllering, 2006). It also helps in eliminating a bias, widely present in trust literature: of looking at the relationships exclusively from one perspective, notably that of the trust-giver (Beckert, 2005).

The total number of the interviewees (involved in several software development projects in Poland) included 10 people who described themselves as managers, 40 who said they were software engineers, and 15 who identified themselves as clients. All interviews lasted about 50-80 minutes, were recorded (by the consent of the informants) and transcribed. The few presented excerpts are the ones considered exemplary to many others, which for obvious space limitations had to be excluded.

## The Manager

When asking managers from IT companies about their perspective on the cooperation within their projects, one common thread linking all the interviewees, regardless of the size of their company, or the character of their client, was striking. Initially, the managers emphasized their appreciation of the client's knowledge of business:

I usually meet with really well-educated, well-prepared people. I think that we're the ones who should improve the knowledge about business, because we have our roots in technology, so we have to learn the business, learn all the industries, in order to understand their needs.

However, the managers sooner or later made it clear, that the customer does not "really need to know" about the technology and the tools used. In fact, in the managers' eyes the clients are childlike, or even childish, in the way they see the solutions, and thus do not need to be acquainted with the details:

We write documents using the language of business, to make it clear for the customer. Exactly because we want him to be able to accept it, to tell us "OK, that's what I wanted. This is how we want it to work". We don't give him the technical details, because they have no meaning. In reality, it completely doesn't matter to him if there is a text field, or a numeric field, if it has 30 or 40 digits... It's really a minor issue.

Many of our interlocutors assumed that going into specifications and techniques is a sure way to get into trouble. In fact, the managers we spoke to were quite unanimous that the customers do not even care about how the things are done:

When we talk about the expected functions of the program, I think that the customers are really well prepared. (...) As regards technology... well now nobody really cares... when they approve the products, they don't do it from the technological point of view.

Keeping in mind such an approach of the customer, it becomes clear why the technological theme is rather being avoided by the managers and why it has negative connotations. Normally in the background, the issue of technology comes into play only when something is not working. It is the moment when the otherwise ideally invisible infrastructure suddenly may become the decisive factor behind the frustration of the customer:

Absolutely. They can assess the system looking whether it makes their work easier or more complicated. (...) Technological details... well, if they access the system through the web, and it works slowly, they start to think about technology – then, apparently, the technological issues aren't well arranged.

This approach to the technological issues poses a challenge for the providers. In reality, there may be a great discrepancy in the assessment of a certain project by the IT professionals and the customer. In the words of the managers, clients lack knowledge and cannot be trusted to have competence to evaluate the product:

I doubt the customer can objectively tell the quality. (...) Of course, he can tell if something is working or not. (...) Well... it must be a part of quality, if something is user-friendly or not, if it's easy to operate. But here the subjectivity comes into play, because everybody has certain preferences. Looking deeper, the customer isn't able to assess if the system that works has been designed and produced according to the professional standards or not (...) we can produce systems that work, but are of poor quality. (...) You see, something can even work well, but when a professional takes a look at it then he can tell that something is, for instance, not optimal, etc. So, it may be poorly produced, and not give any bad impression on the outside. But is it the quality for the customer? I think that when we build something, then we do it according to the rules, to the art of programming, not

just to make it work. But it is rather our quality, I don't think it may be translated for the quality for the customer.

In addition, what really bothers the providers is that the customer does not comprehend the internal logic driving the IT business. Therefore, the client may be suspicious and distrustful towards them, not even trying to realize the provider's position:

I wish we could change their attitude... that we enter the deal to earn money, but it doesn't necessarily mean that we want to con them, to extort money, to give them some crap for huge money. (...) Sometimes I have the feeling that they think that we're there just to swindle... of course, we want to earn money, that is why we do business.

The final criterion of dealing with the customer in the project is, then, to fulfill their demands, in spite of how unreasonable they may seem:

There is even a joke: if the customer has really vexed you, if you want to punish him, then do exactly what he wants you to do. That is what a professional firm should do – only you have to remember to write down exactly what the customer wanted to show him later on.

Accommodating all the customer's requests sometimes means even pretending doing so, only to retain the client and get them convinced that their needs are of top importance...

Sometimes it's not an easy task, sometimes there are obstacles that we couldn't predict. (...) Things happen, the world is not perfect. And, irrespective of our actions, there may be delays, but then the customer says "you must be joking". Then we stand on our heads do to something about it, or... well, very often we pretend doing it, but we have no choice.

The viewpoint of managers on the IT projects seems to be determined by the perspective of the company – they are aware of the power of the customer (who is, at the end, the source of revenue and future recommendations), and they are essentially willing to accommodate all of the client's wishes, or at least make the impression that they do their best. While the customer appears as a point of reference, it was pretty surprising to us that the managers did not talk at all about software engineers. Do they not perceive them as important? Or perhaps they take it for granted that they are players in the same team? The engineers' perspective, presented in the following parts of this paper, should cast an additional light on this relationship.

## The Customer

The image depicted by the managers was diametrically different from the one that we learned from the clients themselves. The clients (or, to be more precise, the people from the customer's side responsible for contacts with the providers) tend to value highly their knowledge about the intricacies of information technologies. They want to go into the technical (not functional!) details of the solution and even modify it insisting on their own ideas. Some of the customers talk about it plainly: they know better than the providers not only what they want, but also how it should be done. The very reason of hiring the external organization to fix the IT issues is not the lack of knowledge, but the lack of time or resources. Sometimes the clients even claim that their competence is higher, as it is much more in-depth and focused on a specific business.

We don't hire contractors because we don't know how to do something. When we get into the project we usually have in mind the image of the system, the specific techniques of getting it done, and we know very well how it should be done.

[Q:] So why do you need the IT providers?

Because, considering the resources we have, we're not able to carry out the projects by ourselves; it is easy – we don't have the resources, we don't have the tools, the equipment. But very frequently we know much better than the provider how they should do things.

That knowledge, from the perspective of the customer, is, however, underestimated by the provider. The representatives of the client may feel disrespected and offended when the provider tries to sell them something that is impossible to deliver, counting on their ignorance. Although they claim that "a little bit of cheating" has always been an inherent element of selling, the customers believe they are able to verify the offers pretty quickly.

Well, we know very well when the salesperson is going to take us in... simply sell us the moon. It is the old rule of salespeople all over the world in every single firm: "we can promise you everything".

This atmosphere of mutual mistrust is augmented by a "two-fold" lack of understanding. As we already know, the customers complain that the providers do not understand the role of IT in their business; but on the other hand it is also difficult for them to do the converse, to understand the forces driving the provider's business. The solution providers do not make it clear to the customer how they benefit from

their business and the lack of such understanding is a fertile ground for a suspicious approach.

If a firm X supplies something to my company, and I pay 100 for it, then they should earn at least 20, because if they don't earn, there is a clear problem in the collaboration, they're trying to cheat me. First of all, I want to know that they really benefit from it, and how much they earn. (...) I don't want to cooperate with companies that sell me something "without the margin". If somebody comes to me telling that he's going to sell me something "below his own expenses", then I can see the problem outright. It means that he wants to take me for a ride. Nobody does things like that. (...) Every time, I first would like to understand what the interests of all parties are. So that we could meet.

As a consequence, in virtually every interview there is a recurring appeal of the customers for more communication on the part of the provider. They seem to be aware that the quality in IT is a result of close collaboration; but at the same time they complain that the providers do not make an effort to understand their needs, to the point that they even feel being patronized. In the end they receive a solution not fitted to their needs, and instead of constant long-term collaboration, there is a solitary work of the provider and, at the end, a big fight over the final product that is not satisfactory to anyone.

You need to make sure. Through talking, through joint workshops, when the provider can show us the solution, sometimes we ask them to demonstrate how it works (...) and in this way we want him to prove that he can do everything he promises, and we also want to make sure that it's exactly the thing we have in mind. (...) The worst situation is when we order something, they deliver this, and then we start to fight, and everyone is right. Because they've invested some money; and we're obliged to pay them although we've not got what we ordered.

The conflict, most salient in the final stage of collaboration, is usually seen as inevitable by the customer, as they simply want to get what they have wanted from the very beginning. The provider is the one to be blamed for the problems, as they did not try to understand the actual needs of the customer, they disregarded their need for communication, they "knew better" what the customer needed. Many customers claim that it is really the time of accepting the solution, when the provider is truly listening to them, for it is also the moment where they can finally use the only two tools they have at hand: firstly, the payment (which may be delayed or denied) and

secondly, the positive or negative recommendation for other potential customers.

When asked for the general reflection about the cooperation with the IT solution providers, the customers were surprisingly unanimous. They were not mentioning specific problems, and they were not talking about their ideal provider. They were, however, all complaining about the way the providers perceive them: as whimsical kids, as incompetent laymen, and sometimes even as enemies in the fight for financial profit. From the perspective of the customers, the providers tend to underestimate the stakes that both parties have in the project, and very often do not understand that bringing the project to the end, as quickly and efficiently as possible, is also to the best interest of the client.

### The Engineer

The last one in the discussed trio is the software engineer. In the unanimous view of the clients and the managers, s/he is the one who possesses some particular competence. A software engineer does not reciprocate the pleantry though: the image of a client, as well as of a manager, is, in our interviewees words, characterized mostly by their ignorance.

It looks like that: a client calls our implementation department and they usually can handle this, and in most cases the things are (...) totally banal, resulting mostly from the fact that we are still in the Middle Ages in some respects, and many people really don't have much experience with computers or technology and really stupid questions arise. So there is [a department] to protect us from this trifle.

The respondent had a quite low view of the clients. He was certain that most problems the customers encounter are in fact a result of their ignorance, civilization backwardness. He was also happy that most of the problems are addressed by a special department. Interestingly, the label of ignorance (*user-loser* as one of the interviewees put it) was applied also to software engineers from the customer's company:

A guy called recently, and he supposedly is a software engineer of sorts, and he said there is no electricity, and what he should do now. How to install a program if there is no electricity. Whether he is supposed to stay there, or go. So what could I tell him? Wait until the electricity comes back, we will not generate this electricity with legs or something. And this guy supposedly was a software engineer, from some bank. There is no electricity and what should he do...

Other software engineers can be subject to jokes and humiliating legends told to outsiders, as long as their knowledge does not meet the standard, or, as long as they represent the customer.

Indeed, the fundamental element of the client's image is technical illiteracy. Although the software engineer was often "protected" from the contact with the client, still most of the stories about occasional meetings with customers were univocally negative. Not even in a single one story had the interviewee spontaneously described a competent client or good rapport, while the stories of benightedness abounded, in a tone similar to the following:

[Q:] What frustrates you most in your work?

Well, that the clients expect miracles, for example that there is some feature in some other application, so why don't we have it here, and how can we achieve the same result, and why can't we, and so on. There was a woman who had a really serious server running, and she kept shutting it down by unplugging the cord. And then she was surprised there are errors in the databases and bugs. And why doesn't the same screen appear back again when she plugs the cord back. (...) I don't know if it was a joke or not, that a client had a computer cup-holder broken, and then it turned out it was a CD-ROM, he was putting his coffee there. And was demanding, in a tone like "it is broken so go and fix it". Sometimes they want miracles.

Something that causes frustration is also the inability to communicate the limits of technology. For somebody not particularly versed in technical issues understanding what is possible and what is not is extremely difficult and "expecting miracles" is quite understandable. Naturally, software engineers are first to take the blame when anything goes wrong:

[Q:] What happens when the client is dissatisfied?

Well, we take the brunt, we interact with the client and the client, say, believes that we are responsible for bugs, for the inexplicable loss of data, and so on. Later we can practically find out that somebody deleted something, but nobody cares then. Initially they make a fuss, and listen only later. And we take the first brunt.

Such a feeling of injustice was a recurrent motif of many interviews. Although all software engineers admitted that technical bugs occur practically always, the first reason for the client's dissatisfaction was in all cases not technical. In fact, most of the interviewees said that the main reason for the customer's lack of satisfaction is ignorance, unreasonable seeking fault on the part of programmers. What is more,



another serious problem in relations with the client commonly reported by software engineers was vague specification of a product:

[Q:] What is the biggest problem in IT projects? Quite often it goes this way... Well, maybe not that often, but it happens that a client wants... A client says that we don't offer what they want, but when we ask what do they want then, it starts all over. They'd like a system that prepares coffee, generates bank accounts, and handles traffic-lights. It is impossible to write a software like that. Well, maybe possible, but totally without any sense.

The lack of vision in expressing the desired outcome was recurrent very often in the interviews. According to the jocular principle included in a book popular among programmers, a programmer should follow the 'Law of Least Astonishment': the program should always respond to the user in the way that astonishes him least" (James, 1986) – such jokes also set the perception of a typical client in the eyes of engineer. To make matters worse, sometimes the client realizes what they need only after receiving the ordered software:

....One of the main problem is that client, when the project has already started, changes the requirements. I would say it actually is the biggest problem in IT systems.

[Q:] Does it really happen?

It happens on everyday basis. But I think it is also a problem of the requirement analysis. Sometimes we leave some things unsaid, and we do not fully realize what the client wants. Such thing always show up some time later, alas, it sometimes happens that they occur in the very end. The client gets the product and it turns out it is not what he wanted.

The tangle of misconceptions, unrealistic promises made by the sales force, high time pressure, all add up to the surprisingly stable power triangle. The issue of trust, or rather distrust, seems to be an extremely useful explanatory lens for interpreting the gathered data.

### Lean not unto Thine Own Understanding

Management literature has acknowledged the utter importance of trust in business (Gambetta, 1988; Shepard and Sherman, 1998; Vangen and Huxman, 2003; Wicks et al., 1999). Among many virtues attributed to trust, its role in facilitating cooperation (Smith et al., 1995) and lowering transaction costs

in collaboration between companies (Jones, 1995) are often emphasized. The notion of trust is used in management and organization science much more often than the idea of distrust (Möllering, 2006). The reason for that may be the tacit assumption that when we distrust others we either do not engage in, or withdraw from cooperation (Sztompka, 1999). Here, the functionality of distrust consists in warning against a prospective partner. But, as our data may imply, the attitude of distrust does not necessarily have to restrain us from entering cooperation, but it may merely suggest that we have to look for another kind of a mechanism that will allow us to take the risk of engaging into a project with a partner (a "substitute of trust", Sztompka, 1999).

The research presented shows that distrust does not have to be a calamity, and its dynamics may in fact support stabilization of the organizational system (Lewicki et al., 1998). The engineers, managers and customers all rely on the distrust they feel and it certainly helps to keep the workplace power balance intact. It is what we call the "occupational glue" of all three vocations that also plays an important role in their professionalization (Carr-Saunders, 1928; Wilensky, 1964; Abbott, 1988; Brante, 1988). Actually, it helps to sense-make the organizational routine in all three cases (Weick, 1969). The research also seems to support the claim that the ambivalent attitudes of trust and distrust may coexist. As Lewicki et al. (1998) point out, human relationships in fact have the property of "thickness" and it is perfectly possible to experience simultaneous trust and distrust towards the same partner<sup>1</sup>; hence, trust and distrust should be rather understood as separate, however interrelated, phenomena (ibid.). The thesis about actual coexistence of trust and distrust draws upon Luhmann's (1979) intuition that trust and distrust actually play the same role: they reduce uncertainty to the point that enables action (in the more recent literature Beckert (2005) related to that function of trust as "tranquilizer").

Addressing the question of cooperation in a distrustful environment, our conjecture would be that successful cooperation in the cases we have analyzed is not as much about trust, but rather, borrowing the term from Giddens (1994), about "*shared understanding*". All three parties in the project perceive each other in a rather suspicious and neglecting way, but it does not restrain them from cooperating. „*Everybody lies, but it doesn't matter, because nobody listens*”, as a famous pun by Nick Diamos has it. Cooperation happens, as the parties share the attitude: nobody feels really disappointed. Besides, managers, software engineers and customers, despite some disrespect and patronizing attitude towards each

<sup>1</sup> In the article the authors provide a perfect example: I may trust that my co-author is a perfect theoretician and therefore I will invite her to collaborate on the article, however, as I completely distrust her teaching skills I will be wary of co-teaching a class with her.

other, know that they all are essential to bring a project to the end. They are aware of their differing approaches to the project, but they either disregard it or do their best to persuade the partner that they indeed share a common perspective. They all “configure” each other (Grint and Woolgar, 1997) in the sense that all groups intend more to impose their own perception of organizational reality, rather than understand the other one. At the same time, all of them have a power and they are ready to make use of it any time: clients may ultimately delay payments, software engineers may make use of their IT competence, the managers may promote or demote software engineers. For sure, they all have high stakes in the projects, and perhaps it is also that everybody has the goods on somebody; and that is the link that forces the partners to cooperate, regardless of the distrustful and disregarding attitude towards each other. Also, many things have to be taken at face value: program code is a black-box for anyone but the programmer, only the client understands the real needs for the software, only the manager realizes what the real deadlines and constraints are, etc.

Reading the literature on trust, we may sometimes get an impression that it is a sort of “magic formula”, glueing the organizational actors all together (Kostera and Kozminski, 2001; Hatch, Kostera and Kozminski, 2004), an ever deficient resource that could cure almost every problem of contemporary organizations (Möllering, 2006). Reading of our empirical data, in contrast, may rather give the feeling that people engaged in software development projects distrust each other, yet still work together. It seems that the background expectancy among occupations within the industry is often one of expected ill will, at least this is the image that emerges from the interviews. They talk the talk of cooperation, but walk the talk of competition and self-interest. Nobody expects from their partner anything, but realization of their self-interest. Hence, the two theoretical proposition briefly sketched in the article: about the functionality of distrust and about its coexistence with trust, seem to find grounding in the words of our interviewees. We believe both of them deserve further examination in empirical research.

## References

- Abbott, Andrew (1988) *The System of Professions. An Essay on The Division of Expert Labor*, Chicago-London: The University of Chicago Press.
- Alvesson, Mats (1993) Organizations As Rhetoric: Knowledge-Intensive Firms And The Struggle With Ambiguity, in: *Journal of Management Studies*, Vol. 30, Issue 6, pp. 997-1015.
- Baier, Anne C. (1986) Trust and antitrust, *Ethics*, vol. 96, pp. 231-260.
- Barber, Benjamin (1983) *The Logic and Limits of Trust*, New Brunswick - New Jersey: Rutgers University Press.
- Beckert, J. (2005). *Trust and the performative construction of markets* (MPIfG Discussion Paper 05/8). Cologne: Max Planck Institute for Studies of Societies.
- Berger, Peter L. and Luckmann, Thomas (1966/83) *Spoleczne tworzenie rzeczywistości (The Social Construction of Reality)*, Warszawa: PIW.
- Brante, Thomas (1988) Sociological Approaches to the Professions, in: *Acta Sociologica*, 31(2), pp. 119-142.
- Carr-Saunders, Alexander M. (1928/66) Professionalization in: *Professionalization*, ed. Vollmer, Howard M. and Mills, Donald L, Englewood Cliffs-New Jersey: Prentice Hall.
- Clifford, James and Marcus, George E. (eds.) (1986) *Writing Culture: The Poetics and Politics of Ethnography*, San Francisco: University of California Press.
- Coleman, James C. (1990) *Foundations of Social Theory*, Cambridge - Mass.: Harvard University Press.
- Cook, Karen, Hardin, Russell and Levi Margaret (ed.) (2005) *Cooperation Without Trust*. New York: Russell Sage Foundation.
- Cooper, Marianne (2000) Being the “Go-To Guy”: Fatherhood, Masculinity, and the Organization of Work in Silicon Valley, in: *Qualitative Sociology* vol. 23. no. 4.
- Czarniawska-Joerges, Barbara (1992) *Exploring complex organizations: A cultural perspective*, SAGE: Newbury Park – London – New Delhi.
- Deutsch, Morton (1958) Trust and suspicion, *Journal of Conflict Resolution*, vol. 2, pp. 265-79.
- Fukuyama, Francis (1995) *Trust: the Social Virtues and the Creation of Prosperity*, New York: Free Press.
- Gambetta, Diego (ed.) (1988) *Trust: Making and Breaking Cooperative Relations*, Oxford: Basic Blackwell.
- Giddens, Anthony (1994) Risk, trust, reflexivity. in: Beck, U., Giddens, A. & Lash, S. (eds.) *Reflexive Modernization*. Cambridge: Polity Press, 184-197.
- Gill, Michael J. (2003) Biased against “them” more than “him”: stereotype use in group-directed and individual-directed judgments, *Social Cognition* 21(5), pp. 321-348.
- Glaser, Barney and Strauss, Anselm (1957) *Discovery of Grounded Theory: Strategies for Qualitative Research*, Chicago: Aldine.
- Greenwood, Davydd and Levin, Morten (1998) *An Introduction to Action Research*, Beverly Hills-London-Delhi: SAGE Publications.
- Grint, Keith and Woolgar, Steve (1997) *The Machine at Work*, Cambridge: Polity Press.
- Hardin, Russell (ed.) (2004) *Distrust*, New York: Russell Sage Foundation.

- Hatch, Mary J., Kostera, Monika and Kozminski, Andrzej K. (2004) *The Three Faces of Leadership: Manager, Artist, Priest*, Oxford – Cambridge: Blackwell Publishing.
- Hochschild, Arlie R. (1997) *The time bind: When work becomes home and home becomes work*, New York: Metropolitan.
- Höpfel, Heather (1995) Organizational rhetoric and the threat of ambivalence, *Studies in Cultures, Organizations and Societies*, 1/2, 175-187.
- James, Geoffrey (1986) *The Tao of Programming*, Infobooks, Santa Monica.
- Jemielniak, Dariusz (2005) Time for IT: timing in software projects, *Managing Modern Organizations with Information Technology*, ed. M. Khosrow-Pour, Hershey: Idea Group.
- Jemielniak, Dariusz (2006) The Management Science as a Practical Field: In Support of Action Research, *International Journal of Knowledge, Culture and Change Management* vol 6(3), pp. 163-170.
- Jemielniak, Dariusz (2007) Managers as lazy, stupid careerists? Contestation and stereotypes among software engineers, *Journal of Organizational Change Management* vol 20(4).
- Jones, Thomas (1995) Instrumental stakeholder theory: a synthesis of ethics and economics, *Academy of Management Review*, vol. 20 no.2, pp. 404-437.
- Kanter, Rosabeth Moss (1977) *Men and Women of the Corporation*, New York: Basic Books.
- Kociatkiewicz, Jerzy and Kostera, Monika (2003) Shadows of Silence, *Ephemera*, vol. 4, pp. 305-313.
- Kostera, Monika and Kozminski, Andrzej K. (2001) Four Theatres: Moral Discourses in Polish Management, *Management Learning*, vol. 32(3), pp. 321-343.
- Kraut, Robert E. and Streeter, Lynn A. (1995) Coordination in software development, *Association for Computing Machinery Communications of the ACM*, New York.
- Kunda, Gideon (1992) *Engineering Culture: Control and Commitment in a High-Tech Corporation*, Philadelphia: Temple University Press.
- Latour, Bruno (1986) The powers of association, in: J. Law (ed.) *Power, Action and Belief - A New Sociology of Knowledge?*, London, Boston, Henley: Routledge&Kegan Paul.
- Latuszek, Dominika (2007) When trust does not matter: The study of communication practices between high-tech companies and their clients in the environment of distrust, forthcoming in: Brennan, L. and Johnson, V. *Computer-Mediated Relationships and Trust: Managerial and Organizational Effects*. Hershey, PA: Idea Group Publishing.
- Lewicki, Roy J., McAllister, Daniel J. & Bies, Robert J. (1998) Trust and distrust: new relationships and realities, *Academy of Management Review*, 23 (3), pp. 438-458.
- Lewis, J. David and Weigert, Andrew (1985) Trust as a social reality, *Social Forces*, vol. 63, pp. 967-985.
- Luhmann, Niklas (1979) *Trust and Power*, New York: John Wiley.
- Martin, Joanne (1993) *Cultures in Organizations. Three Perspectives*, New York-Oxford: Oxford University Press.
- Meyerson, Deborah, Weick, Karl E. & Kramer, R. M. (1996) Swift trust and temporary groups, Kramer, R. E. & Tyler, T. (eds.) *Trust in Organizations – Frontiers of Theory and Research*. Thousand Oaks, CA: Sage, pp. 166-196.
- Mintzberg, Henry (1998) Covert leadership: on managing professionals. Knowledge workers respond to inspiration, not supervision in: *Harvard Business Review*, Nov-Dec, 140-147.
- Möllering, G. (2006) *Trust: reason, routine, reflexivity*. Oxford: Elsevier.
- Perlow, Leslie A. (1997) *Finding Time. How Corporations, Individuals, and Families Can Benefit from New Work Practices*, Ithaca-London: ILR Press.
- Rosen, Michael (1985/91) Breakfast at Spiro's: Dramaturgy and Dominance, in: Frost, P. J, Moore, L. F et. al. *Reframing Organizational Culture*, Newbury Park – London – New Delhi: SAGE.
- Rosen, Michael (1991) Coming to Terms with the Field: Understanding and Doing Organizational Ethnography, *Journal of Management Studies*, 28 (1), pp. 1-24.
- Schwartzman, Helen (1993) *Ethnography in organizations*, Sage: Newbury Park – London – New Delhi.
- Shepard, Blair H. and Sherman, Dana M. (1998) The grammars of trust: a model and general implications, *Academy of Management Review* vol. 23, no. 3, pp. 422-437.
- Smith, Ken, Carroll, Stephen, and Ashford, Susan (1995) Intra- and interorganizational cooperation: toward a research agenda, *Academy of Management Journal*, vol. 38, pp. 7-23.
- Sztompka, Piotr (1996) Trust and emerging democracy: lessons from Poland, *International Sociology*, vol. 11, no. 1, pp. 37-62.
- Sztompka, Piotr (1999) *Trust: a Sociological Theory*. Cambridge: Cambridge University Press.
- Vangen, Siv and Huxham, Chris (2003) Building trust in inter-organizational collaboration, *The Journal of Applied Behavioral Science*, vol. 39, no. 1, pp. 5-31.
- Weick, Karl E. (1969/79) *The Social Psychology of Organizing*, Reading, Massachusetts: Addison-Wesley.
- Whyte, William F. (1984) *Learning from the Field A Guide from Experience*, Newbury Park – London – New Delhi: SAGE.
- Wicks, Andrew C., Berman, Shawn L. and Jones, Thomas M. (1999) The structure of optimal trust: moral and strategic implications, *Academy of Management Review*, vol. 24, no. 1, pp. 99-116.
- Wilensky, Harold L. (1964) The professionalization of everyone?, *The American Journal of Sociology* vol. 70 no. 2 pp. 137-158.
- Wolf, Margery (1992) *A Thrice Told Tale: Feminism, postmodernism, and ethnographic responsibility*, Stanford: Stanford U.P.

**About the Authors**

*Ms Dominika Latusek*

Kozminski Business School (LKAEM), Poland

*Dr Dariusz Jemielniak*

I am an organization theorist interested in action research, using qualitative methods (ethnography, unstructured interviews, shadowing) to analyze knowledge work and innovation management policies. My current research focuses on management practices in high-tech environment.



## THE INTERNATIONAL JOURNAL OF TECHNOLOGY, KNOWLEDGE AND SOCIETY

### EDITORS

**Bill Cope**, University of Illinois, Urbana-Champaign, USA.

**Mary Kalantzis**, University of Illinois, Urbana-Champaign, USA.

**Amareswar Galla**, Australian National University, Australia.

### EDITORIAL ADVISORY BOARD

**Darin Barney**, McGill University, Montreal, Quebec, Canada.

**Marcus Breen**, Northeastern University, Boston, USA.

**G.K. Chadha**, Jawahrlal Nehru University, India.

**Simon Cooper**, Monash University, Australia.

**Bill Dutton**, University of Oxford, United Kingdom.

**David Hakken**, University of Indiana, Bloomington, Indiana, USA.

**Michele Knobel**, Montclair State University, New Jersey, USA.

**Jeannette Shaffer**, Edtech Leaders, VA, USA.

**Ravi S. Sharma**, Nanyang Technological University, Singapore.

**Robin Stanton**, Australian National University, Canberra, Australia.

**Telle Whitney**, Anita Borg Institute for Women and Technology.

**Monica Zuccarini**, Università di Napoli, Italy.

Please visit the Journal website at <http://www.Technology-Journal.com> for further information:

- ABOUT the Journal including Scope and Concerns, Editors, Advisory Board, Associate Editors and Journal Profile
- FOR AUTHORS including Publishing Policy, Submission Guidelines, Peer Review Process and Publishing Agreement

### SUBSCRIPTIONS

The Journal offers individual and institutional subscriptions. For further information please visit <http://ijt.cgpublisher.com/subscriptions.html>. Inquiries can be directed to [subscriptions@commongroundpublishing.com](mailto:subscriptions@commongroundpublishing.com)

### INQUIRIES

Email: [cg-support@commongroundpublishing.com](mailto:cg-support@commongroundpublishing.com)

