2006

Requirements communication culture in mobile services development

Steinar Kristoffersen

University of Oslo, steinkri@ifi.uio.no

Follow this and additional works at: http://aisel.aisnet.org/ecis2006

Recommended Citation


This material is brought to you by the European Conference on Information Systems (ECIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ECIS 2006 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
REQUIREMENTS COMMUNICATION CULTURE IN MOBILE SERVICES DEVELOPMENT

Steinar Kristoffersen, Institute for Informatics, University of Oslo, Norway, steinkri@ifi.uio.no

“Culture is one thing and varnish is another.”
Ralph Waldo Emerson

Abstract

The software business is changing: partly due to a global re-distribution of engineering competencies, partly due to abundant, but marginally profitable opportunities of providing lightweight infotainment services in a global network. The software development process has not changed accordingly, and within a broader trend of globalization, companies look towards offshore outsourcing of software development as one way of cutting development costs. However, many stakeholders are discontent with the quality of deliveries from offshore projects. Many of the problems are explicated by the demographically dispersed nature of a global economy: Different cultures, language barriers and unrealistic expectations. This paper addresses these issues from another angle. Examining in-depth the development processes of an international company, it is concluded that outsourcing, rather than causing upheaval to software development, ought to be seen as subjected to the same naive division of labour as many previous attempts of ‘rationalizing’ this process. Moreover, the counter-measures usually proposed imply ‘much more of the same’ which did not work to begin with either. Therefore, globalization is putting empirical studies of software development back on the agenda and it indicates a need to re-conceptualize the entire software process. A better understanding of how programmers work is needed to back the development of methods for co-located as well as distributed development.

Keywords: Requirements engineering, communication, culture, outsourcing

1 INTRODUCTION

The economy pertaining to offshore outsourcing of software development is growing at a staggering 25 percent per year\(^\text{1}\). In the public opinion, offshore development of software is seen as a threat to jobs and innovation at home\(^\text{2}\), as it seems to be based on brilliant engineers abroad working for salaries which are unbelievably low.

Outsourcing decisions are of course related to the cost of using local programmers (Kjell 2005). It may save as much as 60 percent\(^\text{3}\). It is also a contributing factor to the cost/quality aspect of using offshore development, that, e.g., India alone produces as many technical graduates as that of the entire European continent and about twice as many as that of the United States\(^\text{4}\). Today India alone has an offshore development industry of somewhere in the region of $15 billion. Mockus and Herbsleb point additionally to such alluring reasons for outsourcing as getting closer to international customers, complying with conditions for favourable tax conditions and short lead time if time differences can lead to round-the-clock development (Herbsleb and Mockus 2003).

There are some obvious challenges to outsourcing, however:

“Communication and coordination issues in large software engineering projects have always been formidable (e.g.)\(^\text{5}\). Increasingly, engineers and managers must add the challenges of coordinating

\(^{1}\) http://www.offshore-software.org/2004_01_01_software-outsourcing-news_archive.html; Tue, Jan 27, 2004
\(^{2}\) http://www.digi.no/php/art.php?id=276238
\(^{3}\) http://www.faqts.com/knowledge_base/view.phpml/aid/25380/fid/131
\(^{4}\) http://www.webresourceindia.com/offshore_development_india.asp
work across sites, spanning national, language, and cultural barriers (see, e.g., 6). Driven by market and resource requirements, the push toward globalization has generated a wide variety of problems for software developers.

Previous research 8 suggests that cross-site communication and coordination issues cause a substantial loss of development speed (Herbsleb et al. 2001).”

Studies provided by Heeks, Krishna et al. highlight some of the cultural differences that might arise in a distributed working relationship:

“Global 9’s systems were drawn from a culture of objectivity and accountability. Forcing one set of values onto the other was hard, and Shiva 10’s values proved quite resilient. It took enormous efforts before the Shiva project leader would produce a standardized monthly progress report, and Shiva staff refused to participate in Global’s employee satisfaction survey (Heeks et al. 2001).”

There exists, intuitively, even more compelling propositions about the challenges of global outsourcing, which include: Troublesome communication across sites; participants having different backgrounds; participants not having participated in similar projects previously, participants having different training, coming from different cultures and speaking different native languages. Moreover, Herbsleb and Mockus maintain that in these projects participants are much less likely to have serendipitous contact with each other across sites, due to the lack of face-to-face hallway encounters, and lunch meetings. There are fewer chance encounters with remote colleagues in other words and implicitly, therefore, this is where the improvement of outsourcing projects might be found (Herbsleb et al. 2001).

“In organizations with rapidly changing environments and unstable projects, informal communication is particularly important 11. For example, as requirements change, it is hard for the formal mechanisms of communication, such as specification documents, to react quickly enough (Herbsleb et al. 2001).”

Similarly, the core of the countermeasures proposed by Heeks et al., indicate that issues (even those relating to cultural differences) in outsourcing are generally related to some form of co-ordination: Synching, using “straddlers”, building bridging relationships, etc (Heeks et al. 2001).

Prikладници, Nicolas et al. summarize that the following measures are important in coming to terms with globally distributed software development, particularly with respect to enabling “multinationals and virtual corporations to operate successfully across geographic and cultural boundaries” (Prikладници, Nicolas and Evaristo 2003):

A universal and well-defined software development process is needed in outsourcing projects
Requirements engineering ought to be seen as a seminal activity, which governs the process
Planning needs to be thorough and proper, following it up with careful management of the execution as well as risk factors
Building a global team with an exchange program for executives can make the challenges linked to social and cultural differences more manageable

Ironically, such recommendations do no seem to make that much of a difference in ordinary non-outsourced and local projects, either (Basili and Perricone 1984). The problem actually seems to be that some challenges of software development persistently plague software engineering in general: It

---

10 An American telecommunications company (Heeks et al. 2001).
11 A leading software exporter in India (Heeks et al. 2001).
is complex, unpredictable, and variable in terms of requirements and circumstances of use (The British Computer Society). Motivation seems ample, therefore, to examine aspects of offshore outsourcing in terms of dimensions that are common to software engineering research in general.

The most important finding of the research reported in this paper is that when developing new services for the telecommunications infotainment market (quite horizontally), the “first requirement” is to be able to accommodate continually changing requirements and scope, as and when the product adapts to malleable value chains, new technologies and aggregating layers of agents and actors in the service architecture. This is the one requirement that does not change. Outsourcing developers complain about the lack of specifications, and thus implicate that they, indeed, expect to be able to get them. This we see as an expression of distributed projects naively subjected to a “scientific” organization of work (Taylor 1911). This is an analogy to many historical attempts to conceive of applications development as a “software factory” in a true perspective of reductionism. The outsourcing organization responds to that by accounting for their achievements (and underachievements, for that matter) in terms that are relevant to this idealized organization of work.

Outsourcing works, though, in this particular organization as in most others. There is really no indication that performance is any worse than in non-outsourced collaborative projects (Bruce, Leverick and Littler 1995). It should not be seen as working well despite a local order of software engineering wherein which project managers and developers contribute to formulate the specifications when they themselves need them. Quite the opposite, it works because project managers and developers are involved in writing specifications, meeting the customer when:

“It is difficult for the customer to make himself understood, so the spec is often not enough. It is essential to meet him face-to-face, from time to time (quote from project manager).”

Of course this goes against a fundamental principle of outsourced development, and usually the recommendation is to make sure requirements do not “creep” and that management of the project is strong “within”. This paper might be taken as an indication that such recommendations are part of the problem, rather than the solution.

The next section deals briefly with risk management as one pivotal aspect of understanding software engineering. After that, the methodological perspective underlying the analysis of this paper is presented. We are aware that using an ethnomethodologically influenced approach to a hybrid approach of web-based questionnaires, albeit not at this stage primarily intended to be read quantitatively, and face-to-face interviews, might encounter opposition. Our hope is that it can be appreciated as a contribution in its own right, almost orthogonally to the rest of the paper. Section 4 presents the empirical results, before the discussion concludes the paper.

2 RISK MANAGEMENT

The notion of risk management has historically been at the core of software engineering methodologies, and indeed, many of the implicitly anticipated problems of globally outsourced software projects, such as lacking management commitment, poor requirements definition and change management, communication problems and not having a clear and explicit process model, are commonly held as important risks in the field (Boehm 1991; Lyytinen, Mathiassen and Ropponen 1998). We are interested therefore, to see how actors involved in global outsourcing formulate notions of risk. Selecting one seminal article to compare our findings (Keil et al. 1998), we explore the differences. It is not our intention to pre-empt eventual differences in the perspectives of researchers concerned with traditional IS or mobile services and global outsourcing, respectively, at least not in an absolute sense. That is why we have not surveyed the complete literature on risk management. Neither is it the focus of this paper to extract a representative set of risk assessments from people involved in distributed projects. Therefore, it has not been necessary to investigate the opinions in managers across several companies. The ambition of this paper is more modest; to look at members’ own accounts of risk factors in globally distributed software development, in order to characterize the situation as seen

and managed by those members. Theoretically and methodologically, many of the existing explanations of the unique nature of outsourced software development pertain to ideal and abstract notions of culture, communication, distribution and complexity. Risk management, in this perspective, becomes a set of practices that deal with culture, communication, distribution and complexity, e.g. In contrast, this paper aims to show how the practices involved in risk management (among other things) is part of a “common software engineering culture” constituted by work practices that makes a naively rational view on the software development process work, after all and on the contrary of the stipulated process, and this especially being the case for globally outsourced projects.

3 RESEARCH METHOD AND ANALYTICAL FRAMEWORK

This paper is based on a study carried out in 2005. It concerns the usage of methods in a globally distributed development environment of a company involved in developing mobile content provisioning services. This is not a very large corporation, yet they employ approximately a hundred people, of which half are software developers in an Eastern European country. The locally based operation in Oslo, Norway is mainly occupied with sales and marketing, targeting customers all over the world and clearly just as successful in the US as in Europe, and even more so in East Asia. In East Asia, moreover, they have an outsourcing relationship (from the point-of-view of their East-European subsidiary) going with a smaller development organization.

The data collection for this paper consisted of face-to-face interviews with five managers at various levels from CEO to consultant at the local site of the company, plus questionnaires. The interviews were relatively open, yet structured by an interview guide which aimed to bring about coverage of questions regarding the use of methods, documentation practices and innovation in the company. In order to reach the developers and the managerial level abroad, a web-based questionnaire was developed. Twelve people responded to the questionnaire, more than half of which (7) are locally based, working with various managerial responsibilities. For instance, one is responsible for editing a content provisioning web-site and another for liaising with large media companies, production companies, and Hollywood studios. Most of the local respondents describe themselves as product managers. Their responsibilities comprise the “integration of mobile interactivity into their customers’ overall communications strategies”, which could entail testing handsets, compiling suites of content, marketing and writing up requirements, or even consulting. Five respondents are based at the outsourcing location “offshore”, working with requirements engineering and project management.

It is important to emphasize that the aim of the questionnaire was not to perform quantitative analysis or test a hypothesis with statistical significance. This might be a fruitful approach in future research, for which this first round will work as a pilot. However, at this stage, it was considered simply to be a reasonably cost-effective approach to carrying out interviews ‘at a distance’.

The analysis of this paper, quite in contradistinction to the much of the previous work on outsourcing, which due, probably, to the global nature of the outsourcing phenomena has taken a much more ‘macro’ perspective, is based on the participants’ own reflection, accounts and ‘shared-and-taken-for-granted’ knowledge of the situation. In this respect, it is heavily influenced by ethnomethodology.

In most research papers in IS and related fields, one will find that ethnomethodology is most commonly associated with ethnographic data collection. Data and experiences upon which ethnomethodological analysis are based usually come from ‘shadowing’ the actors, and it is often believed that ‘naturally occurring’ data do carry particular import to such analysis. Participant observation is, with its emphasis on ‘real-world, real-time’ of course at the very heart of ethnography, which, in the next instance, lends itself, at least on the surface, nicely to an ethnomethodological analysis since it focuses on the local circumstances and socially observable accounts of work ‘as-and when’ it is carried out (Crabtree et al. 2000). This is not predominantly the case for this paper. The question (and perhaps objection), then, becomes naturally, is it possible to do an ethnomethodologically informed analysis based on a mix of face-to-face and web-form based interview-data? We realize that this is an approach that might strike a lot of people as odd, but must then refer to the rich variety of experimental approaches applied in ethnomethodology for support (Garfinkel 1967).
It is important to bear in mind, then, that, participating in an interview or responding to a questionnaire in itself can be seen, of course, as an everyday, locally situated and ‘accountable’ activity, from which we can learn, ethnomethodologically speaking, just as much or more about what people do in those particular settings, as one can learn about their work. This is not an attempt to promote a naïve punch-line along the lines that people do not say exactly what they mean when they are interviewed or respond to web-based questionnaire (although that is probably the case as well, from time to time). It means that an interview or replies to a questionnaire rather than being treated as a positive imprint of the external world could be seen as ‘data in itself’, as indeed is the case for this paper, and subjected to a reflexive analysis.

Reading this paper it is useful to consider the analytical platform that ethnomethodology as such represents, inasmuch as it gives significant status rather than treating as only residual aspects of social phenomena, the subjective orientation and multiple levels of logics; and the situated rationale of the participants themselves, to the very extent that it constitutes a systematic set of methods for ‘achieving’ an organization of their work, as software developers and project managers, and producing meaningful accounts of such work. This means, therefore, that participants are not treated as “cultural dopes” or pieces on a chess-board which are simply subjected to institutionalized forces from without, which, behind their backs, as it were, expose actors to a systematic “grammar” that can only be viewed from the “prioritized” vantage points of science. Instead, the methodological point of view of this paper is that of the actors themselves.

Of course, the ethnomethodological objection to functional explanations thus subscribed to is particularly amicable toward looking for alternatives to the dominant assertions about cultural factors and communication problem influencing the performance and organization of offshore outsourcing of software. Some might even argue that it is biased or anti-theoretical in this respect. This is not the case. The ethnomethodological stance levels out the bias that was there from before, by which research on offshore outsourcing has given residual status to the participants’ own accounts of culture and communication, treating it either as a positive image of “the way things are” (in cases where it overlaps with the theoretical perspective of the researcher) or as an external force from which the actors actions are essentially determined.

4 RESULTS

Looking at the web-based interview form, which was administered to 12 managers in the distributed development organization, we find that there is some systematic bias in the reports. For instance, the attitudes taken from the “western” perspective of the Oslo operation seems to confirm the findings from previous research (Coward 2003). The product managers claim that:

Development and testing takes too long, sometimes because the people responsible do not manage to be sufficiently proactive

Deliveries are uncertain (due to lack of control, e.g., the developers not working by a prioritized order of projects)

Cultural difference are a big challenge, for instance it is claimed that the development organization generally have a greater need for more detailed specifications

Communication is poor, for instance by time zone differences and language barriers making communication less effective

From the developing organization offshore, on the other hand, the respondents pointed to the general attitude that “outsourcing” product managers had towards them, perhaps seeing tight schedules and insufficient funding as a concretization of such attitudes. More notably, they objected that “the customer” was changing the scope of objects of the project underway, adding functionality after the design and development “checkpoints.” And this is an important formulation by the project managers; they see communication problems as a manifestation of product managers in Oslo lacking the capacity to state their desires in the “lingua franca” of systems development, namely the requirements specification. Therefore, it is necessary, they claim, to meet with the product managers face-to-face.

5
The situation “inshore”, however, is quite different (almost the opposite), as shown by the following quote from one of the managers about how a game application typically is conceived:

“You go to the customer and ask him, ‘so you want a web page,’ right, then the question is how do you want to market it, in printed press, okay, and how you want to reach the customer is via SMS, and then we have a standard interface for that, that is that info in, sending it, game number, our partner logs onto our server and provides the info and then we push the content, and we can bill. Or if our customer have got a solution then we write the report, there are lots of opportunities, lots of standards...The point is if you ‘know your game’ when you get to an opportunity, well, because the customer does not know what he want, but if you have a standard package there, it is no hocus-pocus...We’ve developed solutions where people pay for games, even though that is not allowed. The operator does not care, you are not supposed to be able to pay for your shirts, but everyone is doing it.”

This excerpt indicates that there is, indeed, no specification ‘out there’ to grab and fix on paper, since the customer does not know what he wants, i.e., the application development is not needs driven. This is not really surprising, but we ought to notice that the ‘operation’ of selling is made with reference to “finished solutions” and “standards,” which, it turns out, are not standardized at all. Firstly there are many competing standards to choose from; secondly, one tends to ‘stretch’ beyond that to make the project outperform its own premises.

That has obvious consequences, but the findings go deeper: These projects are supposed not to be large-scale development projects. Given multifarious standards and solutions ‘that we have already,’

“Turnarounds are much quicker and does not take any time. “

However, mobile services are offered to the market in a layered architecture, reflecting a value chain with multiple actors each taking a ‘cut’ of each others revenue. In each step of the aggregate, a new stakeholder is introduced with their individual brand, approach, desires and experiences, which, in the next instance will influence the ‘requirements’.

It is interesting to observe, in terms of alleged communication problems, that that is always “somebody else’s problem”. In this particular organization, e.g., the product managers “locally” complained about communication problems vs. the East European site, which in its term complained about communication problems vs. a development site even further East, in China. In terms of the “reverse direction” of communication, however, the developers at the East European site do not perceive themselves as being “at the sharp end” of communication problems, although they do indicate that the product managers might not know how to produce a requirements specification, and when they do, it is likely to be changed again and again after programming has actually started.

Thus, there seems to be an indication here that “communication problems” ought to be conceived of much specifically and precisely, and in this case it circles around the capacity to communicate what one wants the developers to develop. But this is exactly what one did not know, and still agreed to build, approaching the matter in a “standard-yet-to-be-developed” fashion from the Oslo side initially.

The specification of requirements, and the work thus involved, rightly seems to deserve ‘cornerstone’ status with regards to further analysis. It is actually possible already to bring this to the fore, analytically speaking, and say that lacking the competencies, and/or treating the writing of exact requirements as an ‘option’ along the tradition that is expected by the software developers and the technical managers, as such is glossed by referring to it as a communication issue or a cultural issues, as indeed one of the managers locally did when he referred to the “offshore” developers as “not understanding our requirements specification culture”. The sales organization and product manager ‘manage the customer’s expectations’ and ‘plan the project’ with ‘adequate’ understanding of what the project entails. It is an observable and accountable praxis to treat such adequate descriptions as constitutive of the social order, i.e., reasonable platform to initiate projects, in this case. The project’s offshore managers, however, inversely question the adequacy of the specification that they receive, exactly in a way that indicates a ‘disruption’ of the working relationship.

The outsourcing organization deals with this situation by asking the developers to write the requirements’ specifications:
“Our outsourcing organization wants a requirements’ specification, but since we do not have enough people we’ve asked them to write a proposal themselves, but then they are annoyed when we tell them they did not get it right. I would have liked to have more senior people here, like a proper development department, which could pull this off.”

Contrasted then with expectancies of globally distributed projects as resting on a weak set of requirements, or at least an extraneous requirements specification, which are actually written by the party farthest away from the customer in absolute as well as ‘cultural’ terms, we surprisingly find that globally outsourced projects are considered “better” by local product managers with regard to finishing on time than in-house ones, and only less trustworthy than locally outsourced projects in this respect. Four from seven of the product managers (not necessarily the same for each dimension) agree that globally outsourced projects finish on time, are well planned and documented, and have requirements which are within expectations. Local managers tend to think those projects which have been “locally” outsourced, e.g., carried out by consultants in the same city, do better than even in-house projects, in terms of finishing on time.

This is most likely not a reliable ‘certification’ of project properties, since they tend to inconsistently be allocated to different sites depending on project type, as the following two excerpts show:

“The requirement is always the same: produce an innovative and effective product! If there is an in-house capability, then that is my first choice. Local is next and global is third. The final determinant is: ‘what will produce the best result?’”

And:

“Often the level of complexity from a technical point of view is the criteria. Simple things you tend to solve in-house.”

Looking further at similar questions, we find that projects basically are ‘the same but different’ regardless of being organized as in-house or outsourced. In-house projects are, on one hand, considered by their “inshore” product managers to be much more likely of meeting the requirements than alternative project organizations, which is exactly what the “offshore” product managers believe is the case with the projects that are “local” to themselves, i.e., in this figure categorized as “globally outsourced”.

The trend seems to be that both parties believe that they do better than the others, but that their own global development partner does better than the “inshore outsourcing” alternative, with one exception being that product managers inshore believe this particular alternative fares better in terms of delivering projects on time.

We asked the managers at both sites to rank items from a list of software development risks, which have previously been developed by Keil, Cule et al (1998). Treating the notion of risk rationally, they assert that one needs to identify and rank risk according to its relative importance, in order for managers to concentrate on those areas which seems most “cost-effective”. Classification of risk is, according to Keil, Cule et al. (1998) an important tool since it implies which risk alleviation strategies are most applicable and effective. Keil, Cule et al. point out the importance of revisiting the list of prominent software project risk factors developed by Barry Boehm (1991), since “the organizational and technological landscape has changed considerably since [this] work appeared (Keil, Cule et al. 1998, p. 77)”. We consider that to be even more so the case, now, with the emergence of global, ubiquitous computing in the guise of mobile multimedia telephony and games.

From the 12 people in our first “managerially-oriented” questionnaire, no-one put “lack of top management commitment to the project” on the top. Noticeably, compared to Keil et al.’s list, 5 out of 12 people put this last on their list! Similarly, no-one put “failure to gain user/stakeholder commitment on the top of their list, although two people placed this second. Instead, 5 out of 12 put this as number 10 out of 11. The last of the customer mandate-oriented risks (Keil et al. 1998), “lack of adequate end-user involvement” was more uniformly distributed from the top to the bottom of the list of priorities. Commitment was seen as equally unimportant with both camps; inshore product developers as well as offshore project managers.
Looking at a slight modified version of the data, however, brings out some new dimensions. Firstly, by aggregating the scores from 1-3 into “Important”; 4-6 into “Less important”; 7-9 into “Almost unimportant” and finally 10-11 and n/a into “Unimportant”, we find that not only does “Changing scopes and requirements get a majority of the votes as them most important risk factor, but the other risk-factors relating to requirements management, e.g., 9 out of 12 similarly considered “Requirements not frozen/lack of change management” to be important (in the top half) as well as misunderstanding the requirements which are put in the top half by 10 people.

![Diagram](image)

**Figure 1: The view on risk**

Some of these variables are quite balanced, e.g., “Lack of management commitment to the project” is seen as moderately important and unimportant, at the same time. Looking more closely at the data, we find that user mandate-oriented risks (Keil et al. 1998) are not ranked as high priority risks by the “offshore” project managers, only by product managers placed locally. Looking at resources and staffing, however, we do not have sufficient evidence to say that there is a difference as to how people in the different types of locations perceive the situation, yielding an interpretation in the direction of this simply being project-dependent. In terms of having the right competencies, however, the following quote is one interesting indication:

“The biggest problem is having the capacity to do the specification work, but I have tried to get ‘offshore’ to do more of it. The problem is that the project manager ‘offshore’ have got too many things to manage, and that is a problem right now, no one has the time to follow up continuously, but if we had had one person assigned to that...We have not written a design, yet, the specification, you know are made like ‘we need this and this and this’ and then we sort it out along the way. It might be a high risk venture, but we’ll see”

Summarizing the findings, the scope and requirements of a project in our case of mobile services development is not considered by “inshore” product managers as something that they control. “Offshore” project managers ask for better specifications, according to an idealized project process which is part of the “outsourcing institution”, as it were. In return they are asked to write the requirements’ specification themselves. This ought, according to a reductionist and rational view of software engineering to be working poorly, but in our case we found the opposite to be true. It actually works quite well. In the next section we will compare these findings with pervious work on software project risks, and see how it can be explained.

## 5 DISCUSSION

One of the challenges in studying outsourcing projects quantitatively is that there is great variation between projects (Sjöberg et al. 2005), perhaps too great to be able to generalize findings and recommendations. Instead, therefore, this paper is based on an in-depth study of one organization, the
findings for which is compared to allegedly general findings from previous work (e.g., (Keil et al. 1998).

We have studied an organization which is deeply engaged in developing various components of content provision services for the international mobile market. Similarly to Smith and Mitra (Smith, Mitra and Narasimhan 1996), we found that the prevalent frameworks for analyzing distributed development have perhaps been slightly too narrowly focusing on:

- Outsourcing as the cause of, rather than the result of, a redistribution of competencies
- Communication and cultural issues as determinants ‘from without’, exerting constraints on projects, and
- Outside issues, at ‘face value’, wholly or partly stipulating which managerial practices ought to be applied to “bridge” such gaps

One perspective towards our findings is that they are no different from the ‘ordinary rhetoric’ which have always motivated software engineering research; seeing projects as complex, deliveries as hard to estimate and predict in terms of time, cost and quality and needing to secure the commitment of major shareholders and lead executives. However, although much is the same as asserted by e.g., Keil et al. (Keil et al. 1998), there are also many differences. We believe that this is due to the characteristic properties of projects here being outsourced, which is, typically, reasonable small and innovative, but most of all, “un-fixed” in terms of their requirements although they were conceived and explicitly possible to conceive as they were based on standard technologies and proven, albeit meagre, business models. This alone indicates that one ought to be sceptical of claims that such projects’ performance can be improved by “supermanaging” it with an (even) unified process, (even) stronger focus on requirements, (even) better planning and (even) more teambuilding (cf. (Prikладніків, Nicolas and Evaristo 2003). This is not only due to the ‘inevitably creeping’ requirements, but because it represents a conceptualization of the software process as one in which the project is subjected to risk from without! This is a common perspective in software engineering (Boehm 1991). Similarly, Keil et al. treat risk as external to the project and something alien and foreign to the product itself (Keil et al. 1998). This does not seem at all to be the case for the software processes described in this paper. The projects referred to by product managers in the organization are exactly about taking risks. This further raises the question about risk and risk management, not as a rational exercise of calculating the severity and probabilities of events, but as ‘work in itself’, ethnomethodologically seen as accounting for the application (or lack thereof) of systematic methods. After all, since software engineering methodologies are explicitly oriented towards dealing with extraneous risk, ‘risk from without’ in a predictive and rational fashion, ‘acting as a software engineering professional’ entails referring to such terms.

Keil et al suggest that they have found eleven stable, global and universal risk factors. The introduction of new technology was not considered particularly significant. This is different from our findings, where a majority of managers say that they consider the introduction of new technology to be an important risk. This is understandable; inasmuch as the projects are sold with reference to their ‘finished’ status and that the conceptual complexity of services for mobile phones, at the current stage, arguably is still limited. Keil et al., moreover, suggest that the importance of top management support is obvious, and in their panels’ many managers espoused this factor as a risk that overshadowed all others. This is very different from our results, after carefully analyzing the data we found that this was, indeed, one of the least important risk factors that people considered. Some people thought that it was important, but no-one classified it as “somewhat important”.

“Another prime area of concern to the panellists was the failure to gain user commitment which was viewed as critical because it helps ensure that users are actively involved in the requirements determination process, and it creates a sense of ownership, thereby minimizing the risk that the system will be rejected. To some, strong user commitment was seen as something that could even compensate for a lack of executive commitment (Keil et al. 1998, p. 79)”.

This points firmly in the direction of Keil et al. describing entirely different types of projects than what we do, and this of course explains also that both “panels” of managers can attribute the greatest
importance to risk management (including the role of new technology), but mean by it quite different things.

Interestingly, Keil et al.’s respondents put great emphasis in the detailed and stable set of requirements, as do our respondents. They claim that “requirements drive the entire project”. This is also something that implicitly underlies the request for better requirements from our project managers, however, we find notably that “our” project are successful in terms of time, quality and even “meeting the requirements” even in the alleged absence of such requirements.

Does this indicate that our case is poorly representing the outsourcing situation of most companies, then, since the projects described here are at least perceived as highly innovative? Intuitively and sometimes even claimed by members, the degree of innovation in a project is an important determinant in the outsourcing decision. There exists, however, empirical evidence that exactly the opposite is the case:

“We expected that firms that invest heavily in R&D pursue innovation, seek long-term profits, and aggressively undertake risky initiatives, and therefore are likely to develop IT systems internally rather than purchase them through the market. The results indicate exactly the opposite; that is, the greater the intensity in R&D, the higher the propensity to outsource IT (Oh 2003).”

We maintain, therefore, that our findings are representative and relevant. It indicates on the other hand, perhaps, that the instrument by which risks are measured and categorized provide insufficient coverage of innovative projects. One of the most emphasized findings from Keil et al.’s study is that the risk that managers believed to be most important was also the ones most often not under their direct control (Keil et al. 1998). Even more clearly, they organize their findings in a conceptual framework comprising two dimensions, perceived relative importance of risk and perceived level of control, and most of the risk factors which scored highly on both dimension could be seen as pertaining to the requirements definition of the project. But in the cases that we describe here, exactly the opposite is the case, requirements are completely beyond the development organization’s control. For instance, for one customer in China, the value network comprises up to five layers of agents which each of them might get a cut from each others’ revenue. Each layer will influence the ‘requirements’ and contribute to drift in the project, and this is not a factor that should be seen as external to the project, is it ‘part-and-parcel’ of doing software engineering in this case.

We believe we have shown that in these projects, requirements should not be seen as changing; the product of the organization that we have studied is exactly about being able to change the product underway and this is a stable requirement. This is consistent with managers’ claim that:

“The requirement is always the same: produce an innovative and effective product!”

The software industry is taking on new challenges in terms of the type of projects, products and services that they target, for which the requirements necessarily are more weakly specified and evolutionary by nature. This means that developers have to work more closely with customers and product managers need to know a lot more about the technical challenges and technical opportunities pertaining to a project. But this “goes against the grain” of many other changes in society. The motivation and interest in, and respect for engineering education and culture in the developed countries are rapidly decreasing. This has to have implications for how the projects are organized, and the choice of approach. Fewer people are available locally to do the technical work that is required. Moreover, a lot of the projects are only marginally profitable and therefore organization cannot afford to keep their own full-fledged development organization. With naive understanding of the software development process as impetus, the organization of work implied by outsourcing is crudely taylorist, and it is from this that many of the ‘outsourcing’ challenges stem. An assumption of developers being able to take on a set of specifications and then program them in a “man-machine-like” fashion program a system according to specification, violates one of ethnmethodology’s central tenets, namely the “unsatisfied substitutability of objective for indexical expressions”. Similar experiences have been documented previously when a similarly reductionist perspective on software engineering has been applied (Gibbs 1994). The question remains, however, are there any differences between globally outsourced projects’ challenges and those of software engineering in general? Kliem (2004)
recognize that although many risk factors are shared between outsourced and local projects, there are some challenges that allegedly face offshore projects uniquely, for instance:

- Lack of interaction between team members
- Unclear responsibilities
- Unmanaged conflict
- Scope creep, no prioritization of change requests
- Inadequate feedback on cost and schedule (mal-) performance
- Conflicting standards of coding, behaviour, decision making- and communication procedures
- Animosity, historical or economical, between nationalities, cultures, institutions, etc.

Kliem (2004) nicely explicates the relationship between these risks and the geographically, culturally and institutionally dispersed nature of outsource software development, in such a way that it becomes possible to see geography, culture and institution as accounts of the order of globally outsourced software development, and in this respect, the expression of and manifestation of certain properties of such projects, rather than the other way around. And that is exactly what the most important contribution of this paper is. There is perhaps a cultural difference, but not been the East and the West (significantly), rather it is within “understanding of requirements communication culture” (a quote from one of the managers in our case). Bearing in mind that although there are (always) complaints raised from developers against ‘sales’, about the poor quality of specifications and the lack of change management, it is interesting that they are still, on both sides of the ‘cultural divide’, comfortable with the performance of the projects versus the requirements.

The rationalization of the software process has deep historical roots, and there are at least two clear lineages of thinking that can be read from the conceptualization of distributed and locally organized project models alike; namely that software engineering is about translation from one representation to another and that this is a “lossless” function (Kyle 2002), and, that it can, indeed, in a reductionist fashion be subdivided into clearly packaged units of work onto which other units of work can build via well-defined interfaces (Raccoon 1997). In opposition to such decisive, early software engineering philosophies, is not groundbreaking in itself to maintain that software engineering might not be rationalizable in a traditional sense (Ilavarasan and Arun Kumar 2003):

“Software work stands as less saturated work. Even if tasks are relatively integrated, complete closure of spaces for play in the structure of skills is difficult to achieve. That is, the worker has enough space to use his creativity and imagination in the work (ibid p. 4).”

One might look at propositions of synching, using “straddlers” and building bridging relationships as more subtle expressions of the same fundamental view on software engineering as a “downstream” process of distinct stages (Heeks et al. 2001). We have found it useful to turn the perspective of outsourcing strategies around. As a complement to trying to find out which strategy can be applied to the outsourcing scenario in order to make it more likely to succeed (Akmanligil and Palvia 2004), or how to select the exact location for development work given a project characteristic (Graf and Mudambi 2005), is would useful instead to find out which project characteristic or outsourcing strategy that comes out of a certain set of globally distributed working relationships.

This paper points toward revising the risk identification framework proposed by Keil et al. (1998) so that it can be used for a broader set of projects, including horizontal applications and innovative services for the mobile consumer. Even more importantly, this paper indicates that it is time (again) to start thinking about software engineering not as a process of translating requirements into a running system, but as a social and creative arena in which customers and developers meet to discuss and implement ideas, which might originate equally from developers as well as customers; regardless of programmers being from one part of the world or the other.
REFERENCES