Stevens, Gunnar; Nett, Bernhard:

Business Ethnography as a research method to support evolutionary design

Gunnar Stevens received a diploma in Computer Science at the University of Bonn and now is Ph.D. candidate at the Institute for Information Systems (Cooperation and Media Systems), University of Siegen. After years of work in the industry, he actually leads a project on End-User Development at Fraunhofer FIT and is involved in the Eclipse Open Source project.

Bernhard Nett received his Ph.D. in sociology, political science and economics at RWTH Aachen (1997). He has worked at the universities of Aachen, Bonn, Freiburg, Berlin (TU), St. Augustin (Fraunhofer-FIT), Siegen and Troyes, France (invited professor). His research interests are Business Ethnography, Software Engineering, CSCW, CSCL and ICT4D.

Abstract:

Product finding has turned out to be a major challenge for commercial software development. In practice, strategies to reduce risks have led to an interweaving of software production and software use, and a demand for evolutionary design conceptions. In theory, however, there still is little research on product innovation, for example, in software engineering. This paper, therefore, describes Business Ethnography (BE) as one method to support reflexive, evolutionary software development and related research, at the same time.

Introduction

“Users must be treated as co-developers, in a reflection of open source development practices (even if the software in question is unlikely to be released under an open source license.) The open source dictum, "release early and release often" in fact has morphed into an even more radical position, "the perpetual beta", in which the product is developed in the open, with new features slipstreamed in on a monthly, weekly or even daily basis. It's no accident that services such as Gmail, Google Maps, Flickr, del.icio.us, and the like may be expected to bear a "Beta" logo for years at a time.

Real time monitoring of user behavior to see just which new features are used, and how they are used, thus becomes another required core competency. A web developer at a major online service remarked: "We put up two or three new features on some part of the site every day, and if users don't adopt them, we take them down. If they like them, we roll them out to the entire site.” O'Railly (2005)

In today´s software branch, production and consumption of software are often mediated in continuous distributed processes, in which innovation-in-use plays a central role. The related role of situated action for innovation, which was first studied by the evolutionary economy of

the 19th century (cf. Reinert and Reinert, 2006), has thus become a research topic for different disciplines, for instance, Cultural Studies (du Gay et al., 1997, Hepp, 2004), Innovation Research (Rogers, 2003, von Hippel, 1994, von Hippel, 2005) and Information Systems (Orlikowski, 2000, Boudreau and Robey, 2005, Jones and Karsten, 2008, De Sanctis and Poole, 1994). In Software Engineering, product finding was, for a long time, simply excluded from software development. However, even in Software Engineering, pioneer works on the role of product development has contributed to increasing interest in the topic (Floyd et al., 1989b, Fischer, 1998, Messerschmitt and Szyperski, 2004, Raymond, 2001).

Digital technologies allow for new forms of the mediation of innovation, in which software applications have become products themselves, and artefacts may serve as boundary objects (Star & Griesemer 1989; Engeström and Miettinen, 1999; Fischer, 1999). However, this productive role of the artefact should not be taken for granted. In fact, it is still too little understood as an innovation potential, and thus remains a difficult challenge for researchers and practitioners, as well. One has to add that even this challenge is not yet always understood in necessary detail. Therefore, it is little astonishing that there is little methodological support even for those aware of the problem of product finding.

This paper describes Business Ethnography (BE) as one method supporting evolutionary design conceptions and related forms of product finding, and which attempt to overcome as well the static as the voluntaristic bias of today´s mainstream design conceptions. The paper starts with a short description of recent trends in software development, from which it concludes the necessity for evolutionary learning within research & development in the form of reflexive development. After some reflections on challenges of theory building, appropriability is presented as general perspective on technology within evolutionary design conceptions.

In pragmatic terms, appropriability can be demanded from individual tools, from whole software infrastructures and even from software development itself. As a related method to support evolutionary technology development, *Business Ethnography* (BE) is presented as one contribution to make software development appropriable for users and stakeholders within technology-development projects. Some concluding remarks form the end of this paper.

The emergence of distributed development

For evolutionary economy, the need for continuous innovation is not a residual, but an essential one: “Creative destruction is the essential fact about capitalism, stabilized capitalism

is a contradiction in terms" (Schumpeter, 1975, p. 83). With increased competition, competences to innovate become a *conditio sine qua non* for organizational survival (Kelly & Storey, 2000, p. 104). Therefore, almost any company makes considerable efforts to better commercialize their industrial knowledge, to create new ideas, and to provide sustainable growth to reach the competitive position they aspire.

However, empirical research shows that there is a constantly high failure rate in developing innovative products. Booz et al. (1982) and Cooper & Kleinschmidt (1987) claim that about 45% of the resources devoted to product development and commercialization are expended on unsuccessful projects. In addition, about 35% of all products launched fail in a commercial sense (cf. Crawford, 1979). The actual work of Kuhn (2007), who conducted a survey over current studies, draw a similar picture: failure rates are consistently significant, although the failure rate in literature vary to a large extent; e.g. some authors talk about a failure rate of 33%, Sividas & Dwyer (2000) about 50%. In the case of costumer goods Andrew and Sirkin (2003) estimate a failure rate about 50%-90%, Haber (2008) even talks about a failure rate of 80%-90%.

In his historical survey about failed innovation (like the invention of microwave in the 1940th), Bauer (2006) pointed out that there are no internal guarantees to create a successful innovation. There are good reasons, why innovation development is not just empirically, but also theoretically an inherently uncertain and risky endeavor, where the possibility to failure is not an accidental, but an essential feature. In spite of the inherent risk to fail, it can be nevertheless a mandatory demand to innovate in at least two cases: in the case of the novelty and in the case of the saturation of a market. In both cases, product finding is a wicked problem that cannot follow conventionalized criteria (cf. Rohde et al., 2009), because conventionalized knowledge either does not exist or does not lead to any interesting novel product.

In Software production the need to innovate in a wicked situation is no exception, but the rule. Here to be innovative involves a structural dilemma: one has no experience about the future when being confronted with high expectations in regard of innovative applications. One strategy to deal with this dilemma is by improving the completion rate by smaller, more manageable projects and by reducing functionality (Beck, 2000). Another strategy that had also become popular in the last years is to innovate cooperatively in open software ecosystems (Messerschmitt and Szyperski, 2004, Raymond, 2001) to increase the efficiency of innovation development and spreading the risk to fail.

Software can be easily reproduced and adapted through its digital character. This may reduce costs of incremental changes dramatically. Software production is characterized by two concurrent, yet opposing trends: software becomes a continuously developed mass-product as well as a highly individualized artefact. These trends in software production are supported by the Internet, a ubiquitous transportation and communication infrastructure for digital goods, which enables new opportunities. Through the new production and consumption forms of software the managing of development in dynamic socio-technical “ecosystems” has become a new major challenge for current Software Engineering.

Software artefacts may evolve in (institutionally) independent, but (functionally) interdependent development traces. Related absence or under-development of connections between these traces can be interpreted as fragmentations of socio-technical “ecosystems”. Fragmentations, therefore, may be a source of undesirable effects. However, fragmentation is not just a technical problem, but closely related to the organisation of communication between the relevant social actors.

The increasing relevance of software evolution in complex, dynamic “ecosystems” is only slowly influencing a paradigm shift in analytical as well as constructional research. Product finding within the development of software development and related problems and opportunities have received only limited awareness within the literature. Problems of software development have generally been interpreted along the paradigm of a problematic construction of unproblematic products, without reflecting the rationality of this paradigm, at all. To overcome this lag, there is still quite some way to go from a mechanic to a truly socio-technical approach.

The mechanic view rests on the paradigmatic example of software as an isolated product in a static, fully known environment. In contrast, the socio-technical view rests on the paradigmatic example of software as an in-determined product in evolutionary socio-technical contexts, which are sometimes called “ecosystems”. Especially in the beginning of computer science as a discipline of its own, theoretic reflection on software development mainly focused on the incorporation of known specifications into computer programs. This paradigm emphasized formal correctness, but neglected practical aspects of the development processes, such as the product finding as a whole. Even in the age of the ‘perpetual beta’ (O'Reilly, 2005), the socio-technical view on software as an evolving artefact is in no way self-evident in Software Engineering.

Paradoxically, it mainly was the establishment of Software Engineering as a genuine research field that has broadened perspectives on development processes. When ethnographic research

became part of its methodological portfolio, the importance of production conditions and human resources for the development process has become increasingly emphasized. Furthermore, while the '*ceteris paribus*' assumption of mainstream approaches like the "waterfall model" (Royce, 1987, Boehm, 1976) still ignore development processes outside production, conceptions like the STEPS model have overcome this static perception of software development. In particular, (Floyd et al., 1989a) emphasized that during the entire life-span of a product, there is continuous development of the objects (the software artefact, the application field, etc.) as well as of the subjects (the user, the designers, etc.).

Thus the understanding of continuously evolving software has been further broadened from a bird and a worm eye perspective: from the bird-eyes view research on open-source projects (Henkel, 2007, Raymond, 2001, Scacchi, 2007) and software ecosystems (Messerschmitt & Szyperski, 2004) increased our understanding of the mechanisms of innovation development in open environment and the division of labor in distributed evolving software. From a worm-eyes view, research on design activities of end user (Mackay, 1990) and the appropriation of technology (Orlikowski, 2000, Boudreau & Robey, 2005, Pipek, 2005a) elaborated our understanding of the production of situated innovations emerging in daily life.

The related new socio-technical perspective is not only a new way to interpret software development, but it includes new opportunities to organize software development and software. For instance, the identification of in-situ design activities was a prerequisite for the identification of related technology and process properties. For instance, the importance of tailorable software environments and evolutionary development models (Wulf and Rohde, 1995, Wulf, 2001) was elaborated a means to exploit the creativity of in-situ design activities among users. Pipek (2005) used the co-evolutionary character of material forms and interpretation schemes for the design concept of Use-Discourse Environments.

In spite of such innovative conceptions, the mainstream of existing design methodologies still neglects the fact that software is subject to continuous development, in which space, time, culture and product families used may form causes of fragmentation. Therefore, the related competence to identify reasons for practical problems is still arbitrarily limited, and software products treated as trans- (or better: proto-) social nature.

Reflexive technology development

In particular in design research, the relation between theory and praxis becomes very complex, because researchers are expected to produce artefacts that change given, proble-matic situations: unlike in historical research, pro-actively intervening into the field, therefore,

is not per se a pitfall of design research, but an essential part of constructional research. Invention by inventing (a gradual, reflexive form of trial and error) is an essential part of design research as one may be able to design an artefact, but not its complete future impact. Therefore, to deal with the unexpected (for instance, use forms) has to be a part of any reflexive competence in scientific Research and Development.

Of course, evaluation has been an element of engineering, in a way, even its core. However, evaluation was reduced to expected features of fully understood (as: constructed) systems – not only in relation to their technical functioning, but also in their socio-technical nature as applications. Most R&D research in software engineering simply tried to avoid wicked problems and related methodological complexity, but practitioners did not have the same opportunity: for them, software engineering was like an attempt to optimize something the nature of which is fully unknown. Thus it was not the knowledge provided (i.e. the optimization schemes), but lacking knowledge (about the current, socio-technically constituted situation), which formed the problem. Design situations were not even identifiable in mainstream research, as they were simply taken for granted (better: as fully knowledgeable).

Thus also the evaluation schemes and testing in software engineering were about expected features. The value of such evaluation could be decisive and unveil problematic design decisions. However, it remained impotent in relation to other, practically often very important problematic design problems. Even worse: by assuming the excluded type of problems from software engineering, it appeared as if related problems could not be tackled in any rational form, at all. Furthermore, as to understand how the unexpected is treated in Research & Development projects and how it could become a means for reflexive proceeding, one needs related ethnographic research. The implicit, but effective reduction of reproducible quantitative testing as the only legitimate form of empirical research in software engineering, for a long time, hindered methodological progress towards more reflexivity.

In contrast, design-research has to address three independent, but related issues without a-priory exclusions of phenomena:

- observing technology in use (working with the artefact)
- developing technology in reflective action (working on the artefact)
- building grounded theories (working on the concepts)

Related socio-technical research is confronted with the situation that the full elaboration of concepts is only reached from a retrospective theoretical reflection of emerging practices and applications. An example is given by Kuutti (1996, p. 18), who mentions that forms of “direct manipulation” are used in practice as early as the sixties, while Hutchins et al. (1986) only

published their theoretical work on this praxis of “direct manipulation” as a theoretic concept twenty years later. But even in retrospective, insight is in no way self-evident and often requires the reconstruction of practices and sense-making processes.

An approach that focused on the special relation between practical intervention and theoretical reflection is Business Ethnography (BE) (cf. Nett and Stevens, 2008, Rohde et al., 2009), one fundamental of which is to acknowledge the historic contingencies of social practices that are developed, among others, in the interplay between the construction and the appropriation of artefacts. BE, therefore, sees novel practices and artefacts often co-emerging (e.g. Orlikowski and Hofman, 1997, Pipek and Wulf, 1999).

Technical applications are not seen as merely theoretical deductions from a static, given and fully understood world of natural laws, but, as socio-technical (and fallible) correlations to habitual human practices, interpretative elements of fragmented, risky human experience. Applications are results of socio-historical contexts – and vice versa. Which nexus prevails is an empirical question related to the individual case at stake (which itself often needs reflexive identification). Therefore, the focus of BE is the purpose or business of the actors developing an application; this is way it is called “*Business Ethnography*”.

In this context, BE tries to study everyday practices not as static entities, but in their potential for general self-organized socio-technical development. This does not mean that related decisions are free of conflicts, power, or ambivalence, in contrast. BE does neither premise a privileged access of science to truth nor a general right to decide for others nor does it deny that decisions for others may become a necessity in certain circumstances. BE takes research as theoretically and practically interwoven with practice and science as an institutional setting of power, but an anticipation of human emancipation, at the same time.

BE as an action-research approach confronts itself with its own decision-making when intervening into practice, which may turn out as contingent. It is not the aim of BE to avoid (value-based, but contingent) own decisions, but to use, analyze and communicate them exemplarily in the light of a public sphere to become more rationally through the discourse of the scientific community. BE, therefore, has a strong affinity to Grounded Theory in methodological as well as methodical terms. Methodologically, the affinity is given by the fact that both share the abductive stance of Pragmatism when trying to build general theories on the empirical ground of experiences within limited projects. Methodically, the affinity is given by the fact that both draw a similar conclusion from evolutionary stance of Pragmatism, arguing that research should be aware of its double-nature as theory building and practical

action taking (see Strübing, 2008, pp. 14): Applications can be constructed within particular projects, but inherit (or demolish) social experiences.
Due to the close affinity to Grounded Theory, the concept of reflective technology development can also be characterized as *Grounded Design*.

Re-constructing distributed construction

Discussing the connection between the material and the meaningful reality easily leads to the question what comes first, the meaning or the material. Deterministic and voluntaristic technology conceptions give two different answers with the claim of generality regarding the causal structure: the deterministic position argues that the material objects came first and enabled related intentions. The voluntaristic technology argues the other way around that the intention came first and led to the forming of the material objects.
Another differentiation between a static and a dynamic perspective helps to understand the rationality of the arguments of determinism and voluntarism better. In a static perspective, the artefact can be abstracted from its connections, and become a mere realization of intention. More precisely, in the static phase (which present the paradigmatic case for the static view), the material and meaningful objects collapse to a unity: the material side expresses the meaning and vice versa. In this case, it is meaningless to ask the question what comes first, which appears like the question about the chicken and the egg. Furthermore, there are no critical incidents: the artefact is produced to function in the way it does, and the users use it in exactly this way. Everything is lucid in this perspective. The price, however, is that product evolution, innovation, is possible only as a planned, fully successful process.
In the dynamic case an artefact is a somehow “untrue” realization of the idea which motivated its production: the product shows unexpected impacts. In this case, the voluntaristic and the deterministic position can be interpreted as two sides of emerging innovation. While possible innovations can be constructed by scrutinizing the static perspective, wicked situations, crises, in which the material and meaningful object do not express each other in cases of existing technology, can be used as a means to elaborate existing experience about their limitations.
The notion of Software Engineering as applied science has been interpreted in a deductive way, in which theories are seen as bases of applications. This would not be that problem, if the theories, at the same, would have been seen as to be based in practice. Instead they were simple considered as externally defined and fully given. Due to two reasons, the engineering disciplines, therefore, are confronted with the critique that they are applying a reductionist, merely instrumental view on the world: The exclusion of contingencies (of which emergence

is a special case) prohibits, on the one hand, the development of systematic practical expertise, and any critical scrutiny of decisions in design which are related with contingencies, on the other.

This leads in some part of Information System (IS) and Design Research (Harrison et al., 2007; Wulf, 2007) to a *praxis-turn* of the discipline. This turn emphasis the situatedness of action (Suchmann 1987) and leads to the adoption of ethnographical methods (Randall et al., 2007) and nowadays also an aesthetical focus (Sengers and Gaver, 2006). For instance, the structuration approach in IS explores phenomena of innovation-in-use by explaining phenomena of non-intended use with the help of the appropriation concept. However, the way it uses the appropriation concept makes it sometimes fall back onto the established perspective which sees the intended use forms are closer to some "true" application than the emerging ones.

As another example, Suchman (1989) interprets situations as irreducible, constitutive settings of human action, but has no concept of development, such as, for instance, Peirce could provide. Last, but not least, Star and Griesemer´s (1989) notion of the artefact as a *boundary object* mediating between different realities treats them as a "black box" and thus does not ask about material preconditions which could make the artefact to become a better boundary object.

The *praxis turn* in Design Research is supportive to get aware of the contingent and situated character of praxis. However, there is still a lack in discussion of the constitutive structure of evolution and the resulted methodological consequences design research should draw from it. Here design research could learn from the Peircean logic the dialectic of organic synthesis (presented by perceptual inferences) and controlled analysis (presented by abductive reasoning) as essential parts of (knowledge) development (cf. Baltzer 1994; Müller 1999; Hoffman 2005). From such a pragmatist stance it is evident that one important form of theory building on innovation is based on a reconstruction-logical analysis, where one asks retrospectively from the existence of phenomena about the necessary conditions which made them possible. Such a retrospective analysis allows for the revealing of the constitutive structure of phenomena. In regard of technology development, the following forms of mediation phenomena become relevant:

Generalizations of situated innovations

One issue of phenomenon that needs an explanation is how an emergent object may become of general interest. Corresponding to that issue is the transition of an artefact into a common object, mediating the interests of different parties.

The communicability of situated innovations

A second issue that needs an explanation is how to communicate about an emergent object by using existing concepts without redeeming the innovative element. Corresponding to this issue is the transition of the artefact into an indexical object, that itself mediates experience.

The experienceability of situated innovation

A third issue that needs an explanation is how emergent objects appear in established reality constructions. Corresponding to this issue is the transition of the artefact into a present at-hand tool, mediating between own and foreign reality conceptions.

From the constitution-theoretical point of view, experienceability, communicability and generalizations are necessary conditions of innovation in their way form an ephemeral emerging object to a permanent new social structure (incorporated in artifacts and routines) and correlates to corresponding organically given constitutions of man. In concrete cases of technological innovation, all three categories play a role in different degree and should by systematically included in development process. However, the given research approaches are not able to tackle all issues or draw a too individualistic picture of the needed competency of men, and hypostatizing a structural model based on “great men” in history (either in the form of the romantic “genius” or in generalizations about Schumpeter´s “dynamic entrepreneur”). Therefore, existing concepts have to be further elaborated related to the question how these form of innovation can be realized in a social manner of a particular project or in society as a whole taken both into account embodied experienceability as well as discursive rationality.

This requires a related reconstruction of social experience in relation to the anticipated application. The analysis of possible applications and existing experience on their limitations only informs about technological opportunities, not about technological feasibility or social acceptability. But, therefore, it may contribute to related techno-political sense-making on the social and on a micro (project) level, as well.

In the following we want to give an outline, how this issue is taken in the BE into account. In addition we want to give a brief outline, what this means in terms of evolutionary technology development.

BE: research & development on technology projects

Business Ethnography (BE) was originally developed as the empirical part of the action-research oriented design conception of Integrated Organization and Technology Development (OTD) (Wulf and Rohde 1995, Wulf et al. 1999). OTD is a process model to support a

technology expert in his efforts to identify and tailor technology dedicated to help a client's self-organization instead of replacing it technologically. Related projects were based on a set of workshops, in which researchers and organization members took part to analyze and define requirements or to discuss design alternatives (cf. Rohde, 2006). BE was to inform the technical experts about the status quo in the given setting. As a research & development method of its own, BE remained conceptualized as a visible intervention into the field established by the cooperation of the project partners.

BE tries to understand development as an evolution of praxis in front of an open future, which is accompanied with non-standardizeable, situated learning process. The methodological consequence drawn by BE is that rigor in such cases cannot be reached by 'hypothesis-testing' methodologies. In this point BE differs from other action-research approaches like Canonical Action Research (CAR) (Davison et al., 2004) that based on hypothesis testing methodology. The argument is that new qualities of novel applications emerging in research projects cannot be adequately ascertained, if they are ex ante subsumed under pre-defined categories. Instead, the categories have to be abducted from the emergent phenomenon itself. Instead of deductive, rigor is to be reached by abductive reasoning.

The qualitative research undertaken originally was based more on interviews than on own field observations. This did not only help the ethnographers to understand the given situation and possible boundary objects (Star & Griesemer 1989), but additionally helped them to establish social capital (Ackerman et al. 2004) between the actors in the project and supporting experts (Nett et al. 2006).

The goal of BE is to understand everyday work practices in a particular context. One of the most important elements of BE is the central role of interviews with project partners on their cooperation practices, which form the basis of analyses. The interviews not only give insights into the distributed, sometimes even contradictory character of the organizational model(s) guiding the actors, but also into deviations from "normality", either perceived by the interviewees or deduced by the interviewer from analyses of the perspectives and experiences of different actors.

BE differentiates between formal organizations, on one hand, and practices enabling them, on the other. It thus focuses on differentiations between routines, disturbances and normative aspects in everyday-work practices. BE aims at the actors´ perception of the situation in the field, but helps to produce a new picture, at the same time: an integral part of the BE is to confront the project partner with the analyses of the interviews with them, and ask them to comment.

The reason for that is two-folded. First this is a common method in action research to validate the analyses, which is adapted in BE. Secondly, this strategy is used to allow for self-organized learning processes: the feedback confronts the interviewees with a perception of their situation that has undergone a methodological interpretation by the ethnographers. Therefore, it is perceived by the interviewees as an expropriation of the experience that they expressed. This expropriation allows for BE analysis as a reconstruction of the potentiality and multi-perspectivity in the field, which is a form of alienation (*Verfremdung*) of the project business for the actors. The feedback of this alienated picture of the project business (including, for instance, anonymous views upon the anticipated product, its impact for the various actors and their related fears and hopes) to the actors allows the project to discuss from a distant (alienated) position, and thus to become able for discursive 're-' and 'ap-propriations'.

The basis for possible re-appropriations of shared anticipations is their alienation, which is not organized as some amorphous "irritation", which only shatters normal perception. Instead, BE analyses and the feeding-back of the potentiality and multi-perspectivity in the field only make the vagueness visible, which has been covered behind a shared anticipation before. Thus the alienation of the shared anticipation is the unveiling of own speculations, which substituted knowledge and was related to the vagueness of former own anticipations. Thus the alienation is no goal in itself, but a prerequisite of more insightfully shared product anticipations, and important thing in product finding.

BE also offers data for analyses of learning processes and organizes common discussions of the interview partners about the validity of their anticipations, their impact for the understanding of the given situation and for the common project, as well. As a compound of action research and ethnography, BE has been applied in several projects, in which the ethnographer cooperated with the project partners to achieve common project aims. Organizing expropriation / re-appropriation loops of related knowledge with the project partners helped them to reflect on their local expertise and develop new strategies.

Appropriable software development: artefact and context foci

Analyzing factual technologic innovation processes demands for dynamic analyses that allow for static ones only in between dynamic innovation processes, as within the static conception, there generally is no innovation. Analyses of technological innovation are pretty complex issues, which demand for a highly differentiated conceptual framework to understand the cases at stake. In the context of technology development, the complexity involved does not

allow for simple solutions. In the case of product evolution, for instance, missing product features may result in innovation blockades, but a lack of understanding of the product, as well.

The concept of appropriability reconstructing human-machine-relations, therefore, demands for detailed analyses, which cover developer and user perspectives, as well. The concept is not an "empty" one, as it allows for the identification of different strategies for evolutionary development. Appropriability can be demanded for different sides of technology development. In regard of the individual artefact, it implies that the application may be tailored by the end user (or some of her representatives, for instance, a software gardener (Gantt & Nardi 1992). However, an application does not make a lot of sense when been tailored to end user demands, as long as it does not remain inter-operable with other tools, at least, on the level of the given platform. Therefore, the demand on appropriability reproduces itself on the level of the software infrastructure.

Again the concept of appropriability turns out to have different dimensions. To start with the most demanding challenge, the development of *software infrastructures* has to analyze estisting ones according to a differentiated conceptual framework described before. Analyses of contradictions in respect of related expectations can unveil implicit opportunities of given infrastructures. This challenge is the basis of Grounded Design. When grounded theory understands the given structure of problems and solutions (in medicine, the origin of theories: diseases in diagnoses and therapies) as a point of departure, but not necessarily as the result of analyses, grounded design use established applications and related contradictions of expectations as a means for development.

On the individual application level, related analyses are more easily to be conducted. However, here one is confronted with the question how to make sense of shifting and contradictory expectations. In this regard, again there are two opportunities. One opportunity is to use a technological solution to deal with contradicting expectations. For instance, Guittard et al. (2006) try to deal with changes of perceptions in projects by displaying the evolving "socio-semantic web". Automatically generated representations of central concepts leading project activities are dedicated to show, among others, what happens within research & development projects, and to allow for related reflexive processes.

Another opportunity to deal with shifting and contradictory expectations is also drawing upon the confrontation of the actors with the multi-perspectivity at stake without automatizing analysis, but by basing it upon "classical" qualitative analysis. The double function of the researcher&developer as a member of the individual project and the scientific community

becomes a source of analyses: as a project member, the *Business Ethnograph* is oriented to produce a project-related perspective of the commonly anticipated application; as a scientist, he later describes the contradictions and problems he had to face when attempting that.

A superficial view upon this approach of confronting project members with their own anticipations and expectations might perceive it as little constructive. In fact, although the related procedural means to support reflexive development turned out to be highly successful in practice, it took some efforts to understand the nature its benefits, the cycle of ex- and re-appropriation of project aims and product anticipations by means of alienation. *Business Ethnography* (BE) turned out to be one of the most elementary forms to support evolutionary product finding and may be combined with other attempts presented.

Conclusion

BE is a contribution to reflective technology development and thus can be characterized as an evolutionary-design conception. At the same time, it is a design research approach, as well. By fostering a (voluntary) expropriation of technology-related experiences, ideas and feelings (such as related expectations and fears), it allows for an analysis of the multi-perspectivity of the given field and may uncover potentialities incorporated in the work routines. By feeding back this analysis, BE develops an alienation ("*Verfremdung*") of shared anticipations. This subdues common re-appropriation of anticipation to common scrutiny, in the technology-developing projects at stake, mostly the product anticipation as the "incorporation" of the project goals. While this process allows for a more detailed picture of the singularities at stake in the given project, BE also aims at possible generalizations of the given innovation, which it discusses in the scientific community. The basis of the generalizations is the reconstruction of failed assumptions, which caused unexpected results in socio-technical practice.

Literature:

Ackerman, M.; Huysman, M.; Carroll, J.M.; Wellman; B.; De Michelis, G.; Wulf, V. (2004): Communities and technologies: an approach to foster social capital? In: Proceedings of CSCW 2004, pp.406-408

Andrew, J. P. & Sirkin, H. (2003) Innovating for Cash, Harvard Business Review, 81, 76-83.

Baltzer, U. (1994): Erkenntnis als Relationengeflecht. Kategorien bei Charles S. Peirce [Knowlegde as netting of relations: categories by Charles S. Peirce], F. Schöningh.

Bauer, R. (2006): Gescheiterte Innovationen: Fehlschläge und technologischer Wandel, [Failed Innovations. Backlashes and technological change], Campus.

Beck, K. (2000): *Extreme Programming Explained: Embrace Change,* Addison-Wesley.

Boehm, B.W. (1976): Software Engineering, IEEE Transactions on Computers, 25, pp. 1216 - 1241.

Booz, Allen & Hamilton (1982): New Product Management for the 1980's. Booz, Allen & Hamilton, Inc.

Boudreau, M.C. & Robey, D. (2005) Enacting Integrated Information Technology: A Human Agency Perspective, Organization Science, 16, pp. 3-18.

Cooper, R.G. & Kleinschmidt, E.J. (1987): New products: What separates winners from losers? In: Journal of Product Innovation Management, 4, pp. 169-184.

Crawford, C.M. (1979): New product failure rates – facts and fallacies, Research Management, 22, pp. 9-13.

Davison, R. M.; Martinsons, M. G. & Kock, N. (2004): Principles of canonical action research, Information Systems Journal, 14, pp. 65-86.

DeSanctis, G. & Poole, M. S. (1994): Capturing the complexity in advanced technology use: Adaptive Structuration Theory, Organization Science, 5, pp. 121-147.

Engeström, Y. & Miettinen, R. (1999): Introduction. In: Yrjo, E., Miettinen, R. and Punamaki, R. (eds.): Perspectives on Activity Theory. Learning in Doing Social, Cognitive and Computational Perspectives, Cambridge University Press, pp. 1-18.

Fischer, G. (1998): Seeding, Evolutionary Growth and Reseeding: Constructing, Capturing and Evolving Knowledge in Domain-Oriented Design Environments, Automated Software Engineering, 5, pp. 447 - 464.

Fischer, G. (1999): Symmetry of Ignorance, Social Creativity and Meta-Design. In: Proceedings of the 3rd ACM Conference on Creativity and Cognition, Loughborough, pp. 116-123.

Floyd, C., Reisin, F.M. & Schmidt, G. (1989a): STEPS to Software Development with Users. In ESEC '89, Vol. 387, (Eds, Ghezzi, C. and McDermid, J. A.), Springer, pp. 48–64.

Floyd, C.; Reisin, F.M. & Schmidt, G. (1989b): STEPS to Software Development with Users Source. In: Proceedings of ESEC-FSE '89. Springer, pp. 48 - 64.

Gantt, M. & Nardi, B. (1992): Gardeners and gurus: patterns of cooperation among CAD users, Proceedings of the SIGCHI conference on Human factors in computing systems, Monterrey, CAL, pp. 107-117

Guittard, C.; Zaher, H. & Cahier, J.P. (2006): Socio-semantic web: towards socially constructed “topic maps” by communities, in: European Academy of Management Conference (EURAM 2006), Oslo

Haber, T.E. (2008): Resistenz gegenüber Innovationen, Gabler.

Harrison, S.; Tatar, D. & Sengers, P. (2007): The Three Paradigms of HCI. In: Proceedings of alt.chi, CHI2007 ACM Press.

Henkel, J. (2007): Offene Innovationsprozesse, [Open Innovation Processes], Gabler.

Hepp, A. (2004): Cultural Studies und Medienanalyse [Cultural Studies and media analysis], Verlag für Sozialwissenschaften.

Hoffmann, M.H.G. (2005): Erkenntnisentwicklung. Ein semiotisch-pragmatischer Ansatz [Knowledge development. A semotic pragmatist approach]. Klostermann.

Hutchins, E.L., Hollan, J.D. & Norman, D. (1986): Direct manipulation of interfaces. In: User-Centred System Design Lawrence Erlbaum Associates, Hillsdale, NJ, USA, pp. 87-124.

Lieberman, H.; Paternó, F.; Klann, M.; Wulf, V. (Hg.) (2006): End-User Development: an

Emerging Paradigm, in: Lieberman, H.; Paterno, F.; Wulf, V. (Hg.): End User Development, Springer

Mackay, W.E. (1990): Users and customizable Software: A Co-Adaptive Phenomenon. Boston, MIT

Müller, R. (1999): Die dynamische Logik des Erkennens von Charles S. Peirce [The dynamic logic of cognition of Charles S. Peirce], Königshausen u. Neumann.

Nett, B.; Dyrks, T.; Mueller, C. & Durissini, M. (2006): Neither Essence nor Accident: Situated knowledge and its importance for the Community Broker, in: Ljungberg, J. (ed.): Proceedings of the 14th European Conference on Information Systems (ECIS 2006), Göteborg, Sweden 2006

Randall, D.; Harper, R. & Rouncefield, M. (2007): Fieldwork for Design: Theory and Practice, Springer Verlag Gmbh.

Rohde, M. (2006): Integrated Organization and Technology Development (OTD) and the Impact of Socio-Cultural Concepts - A CSCW Perspective, Dissertation, Department of Communication, Business and Information Technologies, Roskilde, Denmark, 2006

Sengers, P. & Gaver, B. (2006): Staying open to interpretation. In: Proceedings of DIS ’06, ACM Press, pp. 99-108.

Jones, M. R. & Karsten, H. (2008): Giddens's Structuration Theory and Information Systems Review, MIS Quarterly, 32, pp. 127-157.

Kelly, D. & Storey, C. (2000): New service development: initiation strategies, Library Consortium Management: An International Journal, 2, pp. 104-122.

Kuhn, J. (2007): Markteinführung neuer Produkte [Market launch of new products], Gabler.

Kuutti, K. (1996): Activity Theory as a potential framework for human-computer interaction research. In: Nardi, B. (ed.): Context and Consciousness: Activity Theory and Human Computer Interaction, MIT Press, pp. 17-44.

Liebermann, H., Paternò, F. & Wulf, V. (eds.) (2006): End User Development, Springer.

Mackay, W. E. (1990): Users and customizable Software: A Co-Adaptive Phenomenon, PhD Thesis, MIT, Boston (MA).

Messerschmitt, D. G. & Szyperski, C. (2004): Software Ecosystem — Understanding An Indispensable Technology and Industry, MIT Press.

Nett, B. & Stevens, G. (2008): Business Ethnography - Aktionsforschung als Beitrag zu einer reflexiven Technikgestaltung [Business Ethnography - Action research as a contribution to a reflective technique development]. In: Science theory and design-oriented Information Science Institut für Wirtschaftsinformatik, Westf. Wilhelms-Universität Münster, 48-68.

O'Reilly, T. (2005): What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. O'Reilly.

Orlikowski, W.J. (2000): Using Technology and Constituting Structures: a Practice Lens for Studying Technology in Organizations. Organization Science, 11, pp. 404-428.

Orlikowski, W.J. & Hofman, J.D. (1997): An Improvisational Model of Change Management: The Case of Groupware Technologies, Sloan Management Review, 38, pp. 11–21.

Pipek, V. (2005): From tailoring to appropriation support: Negotiating groupware usage, PhD Thesis, University of Oulu, Oulu.

Pipek, V. & Wulf, V. (1999): A Groupware's Life. In: Proceedings of ECSCW '99 Kluwer,

pp. 199 - 218.

Raymond, E.S. (2001): The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, O' Reilly.

Reinert, H. & Reinert, E.S. (2006) Creative Destruction in Economics: Nietzsche, Sombart, Schumpeter. In Friedrich Nietzsche (1844–1900): Economy and Society Vol. 3, (Eds, Backhaus, J. G. and Drechsler, W.), Springer, US, pp. 55-85.

Rogers, E.M. (2003): Diffusion of Innovations, Free Press.

Rohde, M.; Stevens, G.; Brödner, P. & Wulf, V. (2009): Towards a Paradigmatic Shift in IS: Designing for Social Practice In Proc. of DEGRIST'09, Philadelipha, USA.

Royce, W. W. (1987): Managing the development of large software systems: concepts and techniques. In: Proceeding of the 9th International Conference on Software Engineering (ICSE'87) IEEE Computer Society Press, pp. 328-338.

Scacchi, W. (2007): Free/open source software development. In: Proceedings of the ESEC-FSE '07 ACM, pp. 459-468.

Schumpeter, J.A. (1975): Capitalism, Socialism and Democracy, New York: Harper.

Sivadas, E. & Dwyer, F.R. (2000): An Examination of Organizational Factors Influencing New Product Success in Internal and Alliance Based Processes, The Journal of Marketing, 64, pp. 31-49.

Star, S.L. & Griesemer, J.R. (1989): Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39, Social Studies of Science, 19, pp. 387-420.

Strübing, J. (2008): Grounded Theory. Zur sozialtheoretischen und epistemologischen Fundierung des Verfahrens der empirisch begründeten Theoriebildung, Vs Verlag.

Suchman, L. (1987): Plans and situated actions: The problem of human-machine communication, Cambrigde University Press.

von Hippel, E. (1994): Sticky information and the locus of problem solving, Management Science, 40, pp. 429-439.

von Hippel, E. (2005): Democratizing Innovation, MIT Press.

Wulf, V. (2001): Zur anpassbaren Gestaltung von Groupware: Anforderungen, Konzepte, Implementierungen und Evaluation [Tailorability and the design of groupware. Requirements, concepts and evaluation]. GMD.

Wulf, V. & Rohde, M. (1995): Towards an Integrated Organization and Technology Development. In: Proceeding of DIS ACM-Press, New York, pp. 55-64.

Wulf, V. (2007): Theorien sozialer Praktiken zur Fundierung der Wirtschaftsinformatik: Eine forschungsprogrammatische Perspektive [Theories of social practices as a basis for Information Systems], international reports on socio-informatics, 4, pp. 3-20.

Wulf, V.; Krings, M.; Stiemerling, O.; Iacucci, G.; Maidhof, M.; Peters, R.; Fuchs-Frohnhofen, P.; Nett, B. & Hinrichs, J. (1999): „Improving Inter-Organizational Processes with Integrated Organization and Technology Development", in: J.UCS, Vol. V, Issue 6, Graz, pp.339-365

Wulf, V. & Rohde, M. (1995): Towards an Integrated Organization and Technology Development, in: Ohlsen, G.; Shannan, S. (eds.) (1995): Proc. Of the Symposion on Designing Interactive Systems (DIS 1995) Processes, Practices, Methods and Techniques, ACM-Press, 55-65