12-2009

Cultures of Agility – Agile Software Development in Practice

Karlheinz Kautz
Copenhagen Business School, Department of Informatics, karlheinz.kautz@rmit.edu.au

Casper Fabricius Pedersen,
Copenhagen Business School, Department of Informatics, cp@doek.dk

Ole Monrad
Copenhagen Business School, Department of Informatics, olmo@udd.cbs.dk

Cultures of Agility – Agile Software Development in Practice

Karlheinz Kautz, Casper Fabricius Pedersen, Ole Monrad
Copenhagen Business School, Department of Informatics, Howitzvej 60, DK-2000 Frederiksberg, Denmark
Email: **Karl.Kautz@cbs.dk**, **cp@doek.dk**, olmo@udd.cbs.dk

Abstract

Agile Software Development (ASD) has gained much attention as an approach that provides values and principles for dealing with turbulent environments and change as they are now common for many systems development projects. Literature stresses the importance of organizational culture for agile development, but only little empirical research about their relationship has been conducted. On this background we carried out a cultural analysis in 3 organizations which perform systems and software development according to agile principles. We found that agile development thrives in different organizational cultures as long as the 4 core values are present to a significant extent. This is possible because agile methods bear to compromise even with central elements of the agile approach; this is the essence of agility.

Keywords Agile software development, organizational culture.

INTRODUCTION

Agile Software Development (ASD) has gained much attention as an approach that provides values and principles for dealing with turbulent environments and change as they are now common for many systems development projects. The concept of ASD serves as an umbrella for a number of pragmatic approaches which have emerged based on practice and out of a critique of traditional, document driven development methods (Highsmith 2002). In the agile manifesto the advocates of these approaches state four core values: (1) *individuals and interactions* over processes and tools; (2) *working software* over comprehensive documentation; (3) *customer collaboration* over contract negotiation, and (4) *responding to change* over following a plan (see www.agilemanifesto.org).

Several authors underline the importance of culture for agile development (Cockburn 2001; Highsmith 2002; Coplien & Harrision 2005; Nerur et al. 2005) and argue that an innovative and open organizational culture is necessary to develop software according to agile principles. But a recent review of the agile literature (Dybå & Dingsøyr 2008) has shown that with the exception of Robinson & Sharp's (2003; 2005) work there is little empirical research about their relationship. On this background we pose the questions: What is the relationship between organizational culture and agile development? And is there one such culture? To answer these questions we have performed a case study in 3 organizations which conduct systems and software development according to agile principles. In the following we provide the theoretical background for our study, present our research method, introduce the 3 organizations and their practice, and provide and discuss our findings with regard to the relationship between organizational culture and agile software development.

THEORETICAL BACKGROUND

We ground our work on Schein (2004), who defines culture as "*A pattern of basic assumptions – invented, discovered or developed by a given group as it learns to cope with its problems of external adaptation and internal integration—that has worked well enough to be considered valid and, therefore, to be taught to new members as the correct way to perceive, think and feel in relation to those problems.*" He conceptualises an organizational culture as having three distinct layers. The first layer holds the basic assumptions possessed by all members of that culture. These assumptions concern the members' fundamental values of reality, human relationships, human nature and the environment. They are taken for granted, subconscious, invisible and most difficult to change. The second layer holds the values and beliefs to which members subscribe. They are espoused, overt and debatable unlike basic assumptions. They are visible in the organization's adaptation to the external environment and in the integration of its internal processes. In this context the organization's mission, its attitude towards its customers, its resources and measurements, and its approaches to improvement play an important role as do its worldview, its management and communication style, its distribution of responsibilities and its employees' mutual respect for each other, as well as their organization in groups and the groups' boundaries. In the third and most visible cultural layer, labelled artefacts and creations, visual and tangible manifestations of culture are found, such as the organizations' buildings and physical appearance, their technology, art, traditions, and visible and audible behaviour patterns expressed through language and stories. We will use these characteristics of organizational culture to analyse the 3 case organizations.

Schneider (1994) has elaborated a cultural analysis framework which is used by e.g. Highsmith (2002), one of the prominent promoters of agile development, to explain his understanding of the relation between culture and agile development. In this framework Schneider (1994) differentiates whether an organization focuses on actual experience, reality and present needs, or on anticipating future possibilities and whether its judgement and assessment processes are impersonal and objective or personal and subjective and he distinguishes 4 basic organizational cultures which he labels competence culture, control culture, collaboration culture and cultivation culture. The 4 core cultures can be summarised as follows (see table 1).

	Competence	**Control**	**Collaboration**	**Cultivation**
contents	Future possibilities	Reality/Present needs	Reality/Present needs	Future possibilities
process	Objective	Objective	Subjective	Subjective
appreciates	Performance	Power and certainty	Attachment	Self-realization
cherishes	Superior individual	Compliance with plans	Team work	Creative unfolding
prioritises	Work	The system	People	Idea
asks	How?	What?	Who?	Why?
is led by	Expertise	Authority	Roles	Charisma
is organized as	Projects	Hierarchy	Permanent teams	As little as possible
recruits by	Competitive strength	Reliability	Capability to cooperate	Genius

Table 1: The Characteristics of the 4 Organizational Core Cultures

In a competence culture the organization's focus is on future possibilities and the organization's judgement process is governed by objective measures. The organization appreciates its employee's performance and cherishes the competent, superior individual. It prioritises work and its guiding question therefore is how a work task can be solved. With regard to management style the organization is led by expertise. It is organized in projects and recruits staff to improve its competitive strength.

In a control culture the organization's focus is on present needs and the organization's judgement process is governed by objective measures. The organization appreciates power and certainty and cherishes compliance with its plans. It prioritises its predefined and structured work system and therefore does not ask how and why a task has to be done, but what task has to be performed. With regard to management style the organization is led by authority and it is hierarchically organized. It recruits employees to secure the organization's reliability.

In a collaboration culture the organization's focus is on present needs and the organization's judgement process is governed by subjective measures. The organization prioritises people and appreciates the employees' attachment to the organization. Its guiding question thus is who is performing a task and for task performance it cherishes teamwork. Therefore work is organized in teams, usually in permanent teams. Managers led the organization as role models and recruit employees with regard to their capability to cooperate.

In a cultivation culture the organization's focus is on future possibilities and the organization's judgement process is governed by subjective measures. The organization appreciates its employees' self-realization and cherishes creative unfolding. Therefore it prioritises ideas and asks why something should be done. The organization is organized as little as possible to allow for the generation of as many ideas as possible and for this purpose it recruits its employees by genius. The organization is led by its managers' charisma.

We will also use Schneider's work to analyse the cultures we found in the 3 case organizations. Highsmith (2002) postulates that the competence and the collaboration culture are predestined for agile development, while a control culture calls for more rigid approaches and a cultivation culture would only tolerate a very light methodology. In their study of the XP method (Beck 2000) and organizational culture Robinson and Sharp (2005) suggest that XP can thrive in a range of organizational cultures, they found it applied in hierarchical, collaboration and random cultures – a classification which they have taken from Cockburn (2001).

Our three case organizations did not use one particular agile method, but had tailored their approaches based on several methods. Before we describe the case organizations we will now first introduce our research method which therefore also includes a brief description of the agile methods against which we compared the practice we found in the organizations.

RESEARCH METHOD

Our research follows the approach of engaged scholarship (Van de Ven 2007) which is a participative form of research for seeking advice and perspectives of key stakeholders to understand and theorize about a complex problem. Given the limited literature concerning our research topic, our investigation is based on a comparative, exploratory, qualitative case study (Creswell 2003) of 3 Danish software development companies called Alpha, Beta and Gamma which all (claim that they) apply an agile approach to systems development.

Our research approach is inspired by Walsham (1995), who stresses that in all types of research including case study research theory is important as an initial guide to data collection during the iterative process of data analysis and as a final product of the research. While it is often stated that it is not possible to generalize and certainly not to theorize from case study research, Walsham (1995) suggests that it is possible to generalize case study findings among others in the form of a contribution of rich insight. So inspired, we have used the theoretical background concerning organizational culture to guide our data collection and analysis in order to contribute to the existing body of knowledge with rich insight about the relationship between organizational culture and agile development.

The empirical data for the case study was collected in 15 semi-structured, open-ended interviews. To avoid general and retrospective statements we chose 1 project in each organization which was still running at the time of the investigation. The interview data was supplemented with publically available and internal company documents. The interviews were tape-recorded, but not transcribed. Instead an extensive narrative account of each case was written which served as a frame of reference for the subsequent analysis and interpretation of the data (Fincham 2002). The subsequently presented description of the organizational cultures, their constituting artefacts, values, and basic assumptions are based on the researchers' interpretations and have been verified by the case organizations. The combination of interpretation and collaboration between the three authors with different levels of involvement brought interpretive rigor to the project.

As part of our analysis we also summarize which phases, roles and practices were applied in the 3 projects. This approach has been successfully used by Abrahamsson et al. (2002) for their review of the current state of ASD methods. As Scrum (Schwaber & Beedle 2002) which has a focus on managing agile development projects, and eXtreme Programming (XP) (Beck 2000) which focuses more on the actual development activities according to the literature are two of the major and most adopted agile methods (Abrahamsson et al. 2002; Highsmith 2002) we have chosen them as a reference frame for our analysis. We also take Feature Driven Development (FDD) (Codd at al. 2000) into consideration as it has been used as an agile approach for very large development projects.

In brief, Scrum distinguishes a pre-game phase with a planning and architecture-level design sub-phase, a development phase consisting of iterative cycles called sprints and a post-game, integration and release phase. There are six identifiable roles: scrum master, product owner, scrum team, customer, user, and management. Product backlog, effort estimation, sprints, sprint planning meeting, sprint backlog, daily scrum meetings, and sprint review meetings are the practices which are defined by Scrum. XP consists of 6 phases: exploration, planning, iterations to release, productionizing, maintenance and death. It distinguishes the following roles and responsibilities: programmer, customer, tester, tracker, coach, consultant, and manager. The XP practices are stand up meeting, planning game, small and short releases, metaphor, simple design, testing, refactoring, pair programming, collective ownership, continuous integration, 40-hour week, on-site customer, coding standards, open workspace, 'just rules'. FDD differentiates develop an overall model, build a feature list, plan by feature, design by feature and build by feature as its phases and includes these roles: project manager, chief architect, development manager, chief programmer, class owner, domain expert, domain manager, release manager, language guru, build engineer, tool smith, system administrator, tester, deployer, technical writer. It involves the practices domain object modelling, developing by feature, individual class/code ownership, feature teams, inspection, regular builds, configuration management, and progress reporting.

In the following we present the 3 case organizations and projects and describe which of these phases, roles and practices they applied.

THE CASE ORGANIZATIONS

Alpha was founded in 1989 and has today 21 employees. The organization has long-term relationships with a few customers, with whom they work on the basis of time and material contracts. The project which builds the basis for our data collection is concerned with the development of a customer relationship management system for one of their main customers, a real estate business. Alpha is a member of the Agile Alliance, a world-wide association of agile development organizations. Although each project is free to develop its own development process, in practice the development is strongly influenced by Scrum and XP. Three developers and one supporter work on the project, one of the developers acts as project leader and Scrum master. The organization follows a management style where the individual employee has a mandate to mostly independently make his or her own decisions.

Beta is a consultant and product development company founded in 1998 with 25 employees, 8 – 10 of them work on Beta's main product: a platform for digital media products which is developed in different subsystems and is used in version 2.1 by a number of customers, who offer digital media products. The development tasks are performed in 14 day sprints and are distributed among the developers every second week. Beta's development process is inspired by Scrum, XP and the organizational patterns for agile development as introduced by Coplien and Harrison (2005). The company's chief technical officer acts as product owner.

Gamma is an IT division with independent economy and organization within a large and old public sector communication provider; it has 300 employees and a matrix structure. It exclusively performs in-house development. For each new project a new project team is established; projects have a size of approximately 200 man hours and last between 6 - 9 weeks. The development approach in Gamma is generally inspired by a 'lean' way of thinking (Highsmith 2002), but has much resemblance with the classical waterfall model. The project under investigation is different in this regard as it explicitly subscribes to an agile approach. It develops a sales portal for the company's primary services which Gamma produces for an in-house customer to which it has a traditional customer-supplier relationship. At the time of the investigation 4 developers were linked to the project which performs iterations of 3 weeks length.

Three Instances of Agile Development

Our research is not concerned with the relationship between organizational culture, agile development and the success of agile development projects; however the interviewees in the three organizations put forward that they all deliver products of high quality with high customer and user satisfaction on time and within budget. None of the 3 companies follows one particular agile method to the point as none of them utilizes all process elements of one method, but none of any other. On the contrary all companies' approaches have many elements which can be found in several methods and some which are only found in particular methods indicating that agile methods both overlap, but also complement each other.

Phases

All 3 projects have gone through a Scrum planning phase and kind of an architecture phase, but none of them has performed an architecture phase exactly the way the method proposes it. All projects had an iterative development phase. We can not make any statements about XP's introductory exploration phase, but all projects have performed a planning phase, where estimations were made and where the requirements were carved up in tasks and stories. All projects have also gone through several iterations. The project at Gamma also has a productionizing phase with acceptance and performance tests which is very similar to Scrum's post game phase. Alpha and Gamma also have a maintenance phase as both systems have been operational for a number of years, while this phase at Beta with its newer product which is in use only at selected clients is rather characterized by further development than by maintenance. Although none of the 3 projects explicitly uses a feature-driven phase model which is a basis for FDD all of them organize their work with the aim to produce value creating features.

Roles

All 3 projects have XP's programmer role which in Scrum is called Scrum team, but in none of them XP's customer role can be found to any significant extent. XP requests that the customer produces (the) user stories and performs the functional tests. This is not the case in any of the 3 projects. Alpha and Gamma involve the customer at least in weekly meetings, but at Beta the developers have an even more limited interaction with the customer. The interaction happens through the Scrum role product owner. At Alpha and Beta the developers test themselves, while Gamma has a separate test division and thus a dedicated tester role. XP's tracker and trainer roles are very similar to Scrum's scrum master role, a role which all organizations have although it is not explicitly defined at Gamma. Beta and Gamma also have the consultant role as both have external consultants on their projects. We can find XP's overall manager role in all 3 companies, but Alpha allows the individual developers to make far-reaching decisions instead of taking these at management level. All projects have a product owner although the term is only used at Beta, on the other hand Beta does not have Scrum's customer role as they carry out product development. With regard to FDD Gamma has a classic project leader role, while at Alpha and Beta the project leader role is taken by either some developers or the product owner. None of the organizations has a chief architect or class owners, but understanding the development leader role as similar to the Scrum master, we find it in all 3 organizations, in particular explicitly in Alpha and Beta. Beta also has a domain expert in the form of the product owner. Gamma also has the more classical roles such as release responsible, system administrator, tester and deployer, while Beta actually has a language guru.

Practices

All companies to a large extent use all of Scrum's practices – estimation, product backlog, sprints, sprint planning meeting, sprint backlog, daily scrum meeting, sprint review meeting. At Beta Scrum is fully implemented, while Alpha deviates from the proposed estimation practice as it is the Scrum master alone, who produces the estimates and at Gamma changes of a sprint's contents which should be frozen for a sprint happen frequently. Most of FDD's practices - object modelling, feature development, inspections, frequent builds, configurations management, progress reporting - are also in use. At Beta object modelling of the domain is performed to a high degree, while Alpha and Gamma with their older systems follow this practice to a lesser extent. Inspections are always performed at Beta, sometimes at Gamma and very little at Alpha. Only about half of XP's practices are employed in the 3 companies and this to a varying extent. Open office space, 40 hours week, fair rules and procedures are all to a full extent used in all 3 companies. They also to some extent perform planning games, refactoring, and shared ownership in all organizations, in addition Alpha and Beta also to a large extent work with small releases.

In summary, we can conclude that the 3 companies perform a form of agile development. They follow one method, namely Scrum, to a large extent, but supplement their approach as Scrum does not give much advice concerning actual development processes, but stays on the project management level. The projects therefore also use phases, roles and practices from XP and FDD.

THE ORGANIZATIONAL CULTURES

We now present the cultural analysis of the 3 case organizations as our next step to investigate the mutual impact of organizational culture and local, agile methods.

Alpha's Culture

Artefacts

Alpha is located in an office building where it shares office space with a number of smaller companies. It presents itself on a colourful homepage which focuses on their employees by showing many pictures of the staff members. Alpha has developed its own project management tool, where the employees 'log' their work every 15 minutes. There exists however little written documentation, only what creates visible value for the customer is documented. Alpha's traditions focus on communication and knowledge: the Alpha day, a seminar day once or twice a year on strategic issues; every 3 months a planning day, where the company's long-term plans are discussed, takes place supplemented by a monthly meeting with all staff; knowledge sharing happens during the weekly Friday morning breakfast. Alpha fosters its relationship with the other companies in the building through a joint Christmas party and regularly organizes excursions for the staff and their families. There are two stories which are important for the company's self-image: (1) its founder's original way of working swiftly, quickly and iteratively with the customers still builds the background for the company's flexible and pragmatic work routines today and (2) it was the development manager, who when he was hired, 'discovered' that Alpha actually works according to the agile principles.

Values

The agile principles are in line with Alpha's worldview. Alpha has a flat management structure. The management group consists of 4 people, everyone else is a developer. The individual employees have the liberty and the responsibility to make their own decisions based on the company's ethical principles about what is right and what is wrong. The overall management style can be characterised as 'all power to the developers' which attracts a certain kind of employees. In principle all developers are equal and have the same decision mandate. In practice some of the developers have however larger experience and on this basis exert some authority.

Alpha sees itself as a counter model to the sales-driven, 'tie-wearing' way of running a company. Alpha's developers appreciate the freedom and trust management and customers give them. Alpha's mission is to have an open and honest relation to their customers. The customer should not be the victim of a smart seller. Alpha does not want to have customers who disclaim responsibility for their systems. As a consequence the customers appreciate Alpha's openness and honesty concerning problems and delays and feel that Alpha is dedicated to solving their problems. As it is important to inform and involve customers as soon as problems are discovered, the developers themselves contact the customers to ask for information or to inform them about problems. In this context Alpha tries to avoid fixed price contracts, where everything has to be specified, although the customers often do not have a clear idea of their needs and requests. To elucidate these needs together with the customer Alpha uses graphical mock-ups or blackboard drawings. The relationship to the customer is further supported by the software tool which registers development time quite precisely as the customers have access to the registration system and can see how much time has been used for which task. Beyond this, gut feeling and intensive communication with the customers are used to assess the status of a project.

Alpha believes in the strength of its staff. Each staff member has 100 hours per year which s/he can freely use for competence improvement. Knowledge and experience are shared by moving staff between different projects. Business-, process- and project knowledge is shared through daily stand-up meetings.

Basic assumptions
This leads to Alpha's 7 basic assumptions: (1) Everyone takes decisions based on Alpha's culture. (2) Responsibility is the liberty to self-determine how things are done. (3) The customers' success is Alpha's success. (4) Alpha has demands for its customers. (5) Conversation is more important than written documents. (6) Trust, honesty and openness between Alpha and the customers are decisive. (7) Alpha requests that its employees are customer-friendly and independent.

Beta's Culture

Artefacts
Beta is located in a modern, open office space with meeting and group rooms behind glass walls. The company's web page appears to be business-oriented and quite formal. Beta has a strong male culture; there are only men employed. They have developed their own vocabulary: a spike is an unexpected or special task which has to be taken care of immediately; a ball is the estimation unit which is used to take focus away from time and to make it easier to estimate tasks in relation to each other. Beta's international focus means that all code, comments, documentation and other work documents are in English.

Beta promotes the story that they were the first IT company in Denmark to be agile. They are co-founders of some agile user groups and are very active in the Danish agile user community where some employees contribute with professional articles and blogs. As process consultants Beta's employees focus on software development and less on organizational development and Beta is characterised by a curiosity for technical novelties, everyone including the management follows up on and experiments with the newest technologies. The company is attentive to new developments and important trends and tries to implement them very early; e.g. they introduced open source software to the bank and finance industry in Denmark already 1998.

Beta has a tradition called knowledge transfer Friday: The consultants return to the company and to the office landscape to share their knowledge. This often leads into Friday afternoon leisure activities. Beta has regular family activities, an annual Christmas lunch in a restaurant with partners, and team building activities such as skiing excursions.

Values
Beta's mission is to provide quick and speedy high-quality expertise. They take on time-critical projects and assignments and aim at changing their customers' development processes by actively intervening in concrete projects. Beta has achieved a professional reputation of high regard which is imperative for the company and which is taken care of by its employees, who are regularly invited as speakers on international conferences. Beta wants to be seen by their customers as humble, but professional. The Chief Technology Officer acts as product owner for the digital media product and is in constant dialogue with customers, sales personnel and support staff. The digital media product is a standardised product, but simultaneously each customer gets their own unique solution.

Beta has long been inspired by agile principles; these have become so much part of the company that they are not very visible on a day to day basis. Pair programming e.g. is only used to a very limited extend, primarily for education; however there are always two developers on each user story, who talk together. Productivity is important for Beta. The company has a home-grown system for time registration which primarily is used for billing, but which also provides an indication about the project's productivity. There is some resistance against further registration and measurements as well as against the use of project management tools such as Gantt-diagrams and metrics. The daily morning stand-up meetings are the primary means of all communication with regard to the actual iteration of the product. Education on the product happens through introductory training and pair programming. Quality assessment and improvement is dealt with through a continuous integration server which builds the system, performs all tests and gives feedback on detected problems and errors. Each Friday general knowledge sharing as a basis for further product and process improvement takes place.

Beta has a very flat structure. All developers are directly under the directors in the hierarchy. Beta's founder together with the company's chief technical officers tries to lead the organization by example, they show the way by always being the first to take on a task or an initiative. Beta pursues full openness: everyone can see what everyone else earns and all profit is shared. The employees are proud of their craftsmanship. Everyone is on a high professional level and they strive for being among the best consultants in their field. To give something back to the development community at large, the employees contribute to the open source movement. There are not many formalised roles, but the individual developer has to take the responsibility for his development stories. The daily meetings allow for a swift follow-up of the developers' areas of responsibility and their assigned tasks. Everyone in the development team has been at a Scrum Master course and they are all equal with regard to their knowledge about agile processes. In the digital media project there is nevertheless a certain difference between experienced developers and less experienced developers, who thus much more do what they are asked for. At

Beta knowledge is shared and not kept back for one's own advantage. The employees care for each other, are observant and are willing to help each other beyond the boundaries of any formal groups. There is no central architect or leading programmer, but a joint codes base which all can access.

Basic assumptions

This leads to Beta's 4 basic assumptions: (1) At Beta, everyone is open and considerate to each other. (2) At Beta, everyone loves to develop software and wants to be the best in doing it. (3) At Beta, people are more important than processes. (4) Family is as important as work.

Gamma's Culture

Artefacts

Gamma is situated in an old-fashioned concrete building; there are posters on the wall with the organization's strategic goals and with slogans from a workshop. It was founded as a result of closing down various decentralised IT functions in the overall organization. The developers use techno-language and many abbreviations. The company uses a traditional system which is integrated in the organization's ERP system to register their work hours. The project has an own project management system, where work hours are registered as the developers' estimate of actual project work. As a public sector organization, all development tasks over a certain limit have to go through a public, European-wide tender process. In 2002 Beta consultants worked for Gamma, especially the sales project, to introduce one dedicated agile method, namely XP.

Values

The mother organization is a company everyone in Denmark can relate to. It fulfils an important social task. The work of the sales portal project is very meaningful; many people use the product on a daily basis. The culture in the portal project is very different from the rest of Gamma. The mother organization is a hierarchical organization with many management levels. The sales portal project has long existed in a free space with a many of the best developers assigned to the project. The organization's overall mission is to support building an effective organization with better customer service and with surplus for its shareholders. In this respect Gamma is an ambitious and attractive organization where the sales portal project appears to be a front runner of this development. With the help of the registration system the project leader and the customer responsible monitor the working hours which have to be held within the limits of the agreed contract. Gamma works without frozen iterations, changes are always possible and allowed. Gamma aims at quick response time. The development team in contrast to the rest of the organization is able to provide offers based on only a few hours of analysis and thus swiftly reacts on market opportunities and demands. The developers have themselves direct contact with the customer when clarifications are needed. There is a big difference among the customers with regard to their perception of the advantages of agile development – some are convinced, while others are still thinking in terms of more traditional approaches.

The stand-up meetings and the weekly Monday meeting are the primary communication channels as there exists only limited documentation of the comprehensive and complex business logic of the sales portal. The project has a steep learning curve for new employees and thus supports new staff through a master apprentice approach. New employees have to be very good at object-oriented and especially Java programming. The project goes strong thus staff has to be quick to understand new challenges to be able to react promptly. The team as a whole has a responsibility to get the job done. There is a distinct distribution of responsibility between the development team and the maintenance team – the developers have access to the operational software. Every Friday morning there is a time slot for knowledge sharing. After project closure all projects are evaluated by an external quality assurance function as a basis for improvement. There exists large respect for the individual employee at Gamma, openness and helpfulness prevail. Staff can work at home and has flexible office hours. The customer strongly trusts that the development team can implement their ideas independently.

Basic assumptions

This leads to Gamma's seven basic assumptions: (1) At Gamma, we respect and trust each other and are trusted by the customers. (2) We are ahead of the game with our agile principles. (3) In the sales portal project we show special competence and do special things. (4) We are quicker and create more value than other Gamma projects. (5) We are no slow public organization any more. (6) The mother organization is a better place than the public believes. (7) The mother organization provides an important service to society.

FINDINGS - IDENTIFYING CORE CULTURES

In the following cross case analysis we identify the core cultures of the 3 case organizations and the investigated projects. Table 2 summarizes the results of the comparison.

With regard to the organizations' focus, Alpha develops software for a customer, who has some requirements and who wants some problems solved, and although the organization's professional developers of course also think of the future, they understand as their primary task to provide value to the customer by satisfying the customer organization's need and not to ponder about more extensive possibilities. The same is valid for Gamma although their mother company is continuously looking for new products. In contrast, Beta is engaged in product development with no particular customer; there the developers try to exploit as many ideas and possibilities to reach as many customers as possible. With regard to the organization's overall development process management both at Alpha and Beta is considerate that decisions concerning a particular project are taken by the development team itself and that human, subjective, and emotional judgement – gut feeling – is as a rule the best to follow. Also at Gamma's sales portal project, there are strong traits of a subjective, though group-based decision process within an otherwise hierarchical organization.

Alpha appreciates its employees' self-realization. Developers should dare to talk to the customer, be able to make their own decisions and should pursue further personal development. The project at Gamma tries to recruit the best programmers in the field, thus performance is valued most, while at Beta attachment is most appreciated – the employees should feel good and at home in Beta's developer 'family'. Alpha cherishes teamwork which is demonstrated by the developers' close cooperation and the mutual testing of their software. In contrast both Beta and Gamma cherish the superior individual with its strong technical competencies most. Both Beta and Gamma prioritise work highest, understood in such a way that it is deeply rooted in the culture of the two projects to strive for a solution of uttermost quality. Alpha in extension of its emphasis on self-realization, prioritises the employees most.

The organization	**Alpha**	**Beta**	**Gamma**
's contents	Reality/Present Needs	Future Possibilities	Reality/Present Needs
's process	Subjective	Subjective	Subjective
appreciates	Self-realization	Attachment	Performance
cherishes	Team work	Superior individual	Superior individual
prioritises	People	Work	Work
asks	Why?	Who?	How?
is led by	Charisma	Expertise	Authority
is organized as	Permanent teams	Permanent teams	Hierarchy
recruits by	Capability to cooperate	Competitive Strength	Competitive Strength

Table 2: The Core Cultures of the 3 Case Organizations

The 3 case organizations differ with regard to their central question: At Alpha the emancipated, individual developer independently gets acquainted with a task or challenge and asks 'why' a certain approach is better than another. At Beta the question is 'who', they stress that the right developers have to work together on the right assignments, while at Gamma the project is part of a complex technical network and thus focuses on 'how' a problem can be solved.

Alpha is led by charisma, as with its value-based management form, the management team represents a catalyst and inspiration for the staff. At Beta the experienced developers and the chief technology officer, act as Scrum masters and supervisors for the younger staff. This can be considered as management through expertise. Gamma has a comprehensive hierarchy with many management positions and is managed through authority. At Alpha and Beta the projects are organized as permanent teams, while Gamma with its matrix structure is organized by work teams which are composed hierarchically with the exception of the sales portal project which has a somewhat flatter organization.

Finally, at Alpha it is more important that staff fits into the culture and likes to have a relationship with the customer than that they have technical capabilities, thus people are recruited based on their capability to cooperate. At Beta and Gamma more importance is attached to the technical competence of the employees – they select the best developers for the task and thus emphasize their competitive strength most. In all 3 organizations we find an organizational culture and development practices which complied to a significant extent with the four core values of agile development (1) *individuals and interactions* over processes and tools; (2) *working software* over comprehensive documentation; (3) *customer collaboration* over contract negotiation, and (4) *responding to change* over following a plan. Now applying Schneider's (1994) framework, we find the following core cultures.

Alpha – a cooperation culture

Alpha's core culture is cooperation, but shows also a number of traits of cultivation. A cooperation culture requires partnership with the customer which is build up over time (Schneider 1994). Alpha has done exactly this with their main customer and tries to do this with its other customers. 'The customer's success is our success' is one of Alpha's basic assumptions and this is well aligned with the organization's strategy to build up long term relations with the customers. Culture is an integrated part of the management philosophy.

Alpha is based on value-based or more precise culture-regulated management. The idea of the single individuals' autonomy governed by the organization's culture is at the same time a link to a cultivation culture (Schneider 1994). According to Anderson (1997) an organization needs to have some fundamental principles as a basis to formulate its values. Here, honesty is an important principle at Alpha. If something gets delayed or an error has been made, the first thing a developer does is to get hold of the customer and to communicate the problem. The employees appreciate that they can have an honest and close relationship to the customer which is closely related to both the cooperation and the cultivation culture.

The cooperation culture aims at a synergy between the different parties in an organization as well as between the organization and its customers and other external stakeholders (Schneider 1994). Alpha recruits according to this culture. Therefore Alpha's developers are no perfect, technical 'superheroes', but people who know how to co-operate, to take responsibility, and to communicate with customers.

Beta – a competence culture

Beta presents a distinct competence culture with some traits of cooperation and cultivation. Consultancy companies are typically competence cultures (Schneider 1994) and at the same time this culture is well suited for the development of niche products for selected customers – such as Beta's digital media product (Schneider 1994). Beta has no trait of a control culture as this would be against 'gut feeling' (Schneider 1994) which Beta readily and often utilizes. 'Family is as important as work' is one of Beta's basic assumptions and here the culture shows characteristics of a cooperation culture which in contrast to a competence culture has the family as its archetype (Schneider 1994). Also Beta's policy concerning salary and sharing of profit between all employees shows more traits of a cooperation culture than of a competence culture.

It is a characteristic of a competence culture that management has the same or even better technical competences than staff (Schneider 1994): Beta's managing director has been a proficient developer; today he is involved in agile methods on a level where he gives presentations and helps other organizations to implement an agile approach. Competence as core culture is also mirrored in the fact that Beta experiments with new technology and advanced prototypes of their product.

Beta has been founded with a view to be a front runner of the trade. This idea is still central and attracts competent staff. The company is a consultant company organized around technical specialists which is reflected in their product. Beta does however not only recruit experienced, technically superior staff, but also employs and trains newly graduated people without experience, but potential.

Gamma – a subculture in a culture

Gamma has no unambiguous core culture. As our analysis both takes Gamma as an organization and the sales portal project into account, the sales project presents an antipode to the rest of the organization. This is related to that 3 of the project's basic assumptions 'In the sales portal project we show special competence and do special things', 'We are quicker and create more value than other Gamma projects' and 'We are ahead of the game with our agile principles' all emphasize the project's attitude of being different and more effective than the rest of the organization.

Gamma's management form however is hierarchical and authority-based and it would be wrong to argue that the project explicitly has gone against the hierarchy's rules and procedures, but repeated good results and goodwill from the closest management have made it possible to bend the otherwise strict rules a little. The project's development method has become a compromise between a 'pure' agile form and a more top-down form and in the same way the development team's culture is a mixture, but also a subculture in the overall Gamma culture.

According to the literature it is quite natural and probably also reasonable that Gamma is overall placed within a solid control structure. Monopolies are natural control structures (Schneider 1994) and Gamma mother company has until recently been a monopoly. Now, like the mother company Gamma is trying to learn to compete on the market and this requires changes in the organizational culture. The sales portal project sees itself as a spearhead of this change and this strengthens the team's identity as a subculture.

The project deals with a complex business problem and has for a long time been critical for the whole organization, therefore many of the very best developers have been assigned to the project. This indicates traits of a competence culture.

CONCLUSION

Highsmith (2002) argues that the development method has to be compatible with the organizational culture of the development organization and vice versa. We found 3 different cultures within all of which agile development takes place. This indicates that there is not only one such culture. Figure 1 presents the cultural profile of the 3 investigated organizations and projects. It shows that all 3 organizations and projects beyond having one core culture also include elements of other cultures.

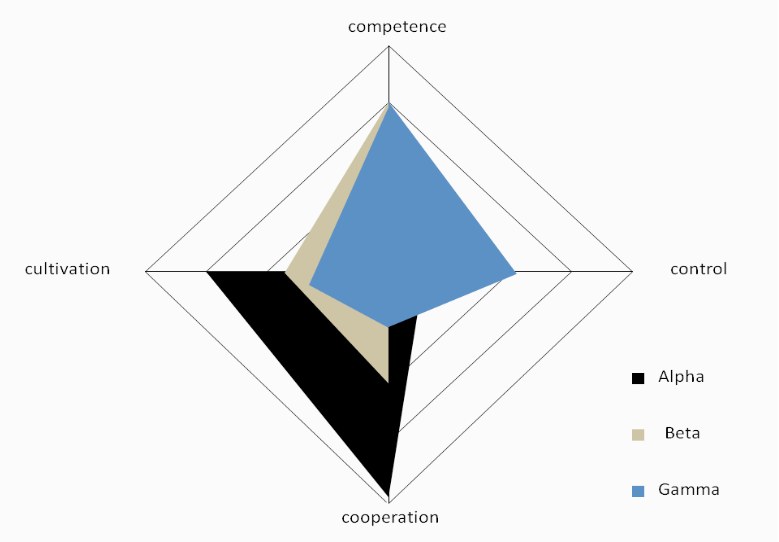

Figure 1: The Case Organizations' Cultures

Through our culture analysis we found that agile development thrives in different organizational cultures such as competence and collaboration cultures, but also in cultures which have a strong element of control, but where the 4 core values are to a significant extent present. This is possible because agile methods bear to compromise even with central elements of the agile approach; this is the essence of agility.

Our cases illustrate that in some organizations agility is deeply rooted in management and here the agile methods might meet fewer obstacles and are applied with fewer compromises; in other companies a single development team might follow an agile approach. Organizational culture has an impact on the way agile development is enacted. In practice it is the method which is adjusted to the organization and not the organization to the method.

With our work we confirm Fitzgerald et al. (2006), who provide empirical evidence and make an argument that in practice combining several agile methods, instead of sticking to one method only, is a viable approach to systems development. We also confirm Highsmith (2002) that agile development prospers in competence and collaboration cultures, but we also back up Robinson and Sharp (2005) and show that it works in more hierarchical cultures as well.

In our study we did not find an organization with a dominant cultivation culture, in which an agile approach was applied. Highsmith (2002) would argue that with its emphasis on creative unfolding a cultivation culture is less suitable for agile development, while Robinson and Sharp (2005) found that agile development takes also place in organizations with little or no central control and command. More empirical research is needed to understand the relation between organizational culture and agile development and to settle in particular the issue of the relation between the cultivation culture and agile development.

REFERENCES

Abrahamsson, P., Outi, S., Ronkainen, J., Warsta, J. 2002. *Agile Software Development Methods – Review and Analysis*. Oulu, Finland: VTT Publishing.

Anderson, C. 1997. "Value-based Management," *Academy of Management Executive* (11:4), pp 25-27.

Beck, K. 2000. *Extreme Programming Explained: Embrace Change*. Boston, MA : Addison-Wesley.

Cockburn, A. 2001. *Agile Software Development*. Boston, MA: Addison-Wesley.

Coad, P., LeFebvre, E., DeLuca, J. 2000. *Java Modelling in Color with UML: Enterprise Components and Process*. Upper saddle River, NJ: Prentice Hall.

Coplien, J.O. and Harrision, N. B. 2005. *Organizational Patterns of Agile Software Development*. Upper saddle River, NJ: Prentice Hall.

Creswell, J. W. 2003. *Research design - Qualitative, Quantitative and Mixed Methods Approaches*. Thousand Oak, CA: Sage Publications.

Dybå, T. and Dingsøyr, T. 2008. "Empirical Studies of Agile Software Development: A Systematic Review," *Information and Software Technology* (50: 9-10), August 2008, pp 833-859.

Fincham, R. 2002. "Narratives of Success and Failure in Systems Development," *British Journal of Management* (13), pp 1-14.

Fitzgerald, B., Hartnett, G., Conboy, K. 2006. "Customising agile methods to software practices at Intel Shannon, *European Journal of Information Systems*, (15:2), April 2006, pp 200-213.

Highsmith, J. 2002. *Agile Software Development Ecosystems*. Boston, MA : Addison-Wesley.

Nerur, S., Mahapatra, R., Mangalaraj, G. 2005. "Challenges of Migrating to Agile Methodologies," *Communications of the ACM* (58:5), pp73-78.

Robinson, H. and Sharp, H. 2003. "XP Culture: Why the twelve practices both are and are not the most significant thing", *Agile Development Conference*, June, 25-28, 2003, Salt Lake City, UT, USA: IEEE Computer Society, pp 12-21

Robinson, H. and Sharp, H. 2005. "Organizational culture and XP: three case studies," *Agile Development Conference*, July, 24-29, 2005, Denver, CO, USA: IEEE Computer Society, pp 49-58.

Schein, E. 2004. *Organizational Culture and Leadership*. 3rd Edition. San Francisco, CA: Wiley & Sons.

Schneider, W. 1994. The Reengineering Alternative: A Plan for Making Your Current Culture Work. New York, NY: Irwin Professional Publishing.

Schwaber, K., and Beedle, M. 2002. *Agile Software Development with Scrum*. Upper Saddle River, NJ: Prentice Hall.

Van de Ven, H.A, 2007. *Engaged Scholarship A Guide for organizational and social Research*. New York, NY: Oxford University Press.

Walsham, G. 1995. "Interpretive Case Studies in IS Research: Nature and Method," *European Journal of Information Systems* (4), pp 74-81.

COPYRIGHT