The influence of organizational structure on the adoption of extreme programming in software and product development

Cristiano Tolfo[a,b], Raul Sidnei Wazlawick[a], Fernando Antonio Forcellini[a], Marcelo Gitirana Gomes Ferreira[a,c]

[a]Federal University of Santa Catarina
[b]Federal University of Pampa
[c]Santa Catarina State University

e-mails: tolfo@deps.ufsc.br; raul@inf.ufsc.br; forcellini@ deps.ufsc.br; marcelo.gitirana@gmail.com

Abstract: If fully applied, the XP - Extreme Programming - method may provide a high degree of performance and quality for software companies, as well as for companies which develop their products with embedded software or develop systems somehow related to their product development process. However, differently from the more prescriptive methods for software development, XP is strongly based on interaction among people and related human aspects such as cooperation, communication and democratic leadership. Considering that the presence of such aspects depends on the way the organizational structure of the software companies is organized, this article analyzes this structure under the perspective of the XP method suppositions. The result of this analysis consists on the identification of favorable and unfavorable aspects to XP adoption - which serves as parameters for careful consideration on the convenience of implementing the aforementioned method.

Keywords: methods of software development, software companies, organizational structure, extreme programming, agile development, product development process.

1. Introduction

Agile methods are presented as an alternative to the formal and complex traditional software development methods. The Extreme Programming (XP) stands out among the several agile methods in use today. The XP method aims at guaranteeing the success of the software development process co-existing with vague and ever-changing customer requirements (BECK; ANDRES, 2004).

Since its creation in the Nineties, several software development companies have shown interest in adopting it. Regarding the implementation of a new method, McBreen (2002) warns that when a company decides to adopt a new software development method it is necessary to assure it is adjusted to the company and to the people working on its projects. The new method should suit the company profiles and its members' profiles as well.

According to Cunningham (2003), the organizational values of some software development companies are totally contrary to the XP values. In such cases, the adoption of the method can be regarded as unfeasible.

Taking into account the above considerations, this paper discusses an important aspect for using XP which is usually neglected by the literature: The organizational structure of the software development companies. This organizational structure relates not only to the company organization chart, with the definitions of positions and functions, but also to human aspects that influence its decision-making process: The adoption of a new software development method, for instance. It is important to consider the compatibility of the organizational structure of the company with the XP method in order to avoid the wear and tear of a hurried implantation of the method.

This paper aims at identifying aspects of the organizational structure of software companies – as well as of companies that develop products with embedded software or systems in any way related to their product development process – that are favorable or adverse to the adoption of XP. These aspects may serve as parameters for these companies consider their compatibility with the XP method. In order to reach this objective, first of all, we present the practices and values of XP, as well as the concept of organizational structure and its dimensions. Then, we analyze each dimension of the organizational structure under the perspective of XP practices and values. This analysis allowed us to prepare a questionnaire that helps identifying favorable and adverse aspects of adopting the agile method. The questionnaire has been applied to a software company with the purpose of demonstrating the influence of the organizational structure in the adoption of the method XP.

It is worth emphasizing that this study of is part of the initial stage of a research project conducted by the

Product and Process Engineering Group (GEPP) of the Federal University of Santa Catarina (UFSC). According to Rozenfeld et al. (2006), increasingly the software presents itself as an integral part of the product or as a tool to support the business processes throughout the Product Development Process (PDP). The author proposes the integration of the PDS to the PDP as in Figure 1. In this sense, Cooper (2001) points out that software development projects are not isolated activities inside the organization. This author says that under the product development, software development is usually part of the development of the hardware, also interacting with areas such as marketing and production planning. These areas should thus be managed and coordinated concurrently.

Facing the reality presented in the previous paragraph, and driven by studies on the implementation of agile principles in the development of tangible products such as those presented by Chin (2004), Highsmith (2004) and Smith (2007), the GEPP is conducting some studies on the relationship between PDP and agile methods. Among the main objectives of these studies are the application of agile principles in the context of the PDP and also the proposition of a Model of Reference for the Software Development Process –based on agile principles - that is properly integrated to the Unified Model of Reference (for the PDP) presented by Rozenfeld et al. (2006). In this context, Karlström and Runeson (2006) have demonstrated the feasibility of applying XP method for the production of embedded software for products developed with the Stage-Gate Model.

According to Karlström and Runeson (2006), the attitude of the management is one of the key issues to the success of the integration of XP with Stage-Gates models. Despite this, these authors did not provide any tool to support a previous consideration of the influence of the organizational structure in the adoption of XP method, as we do in this paper.

2. Extreme programming

The XP method is characterized by a set of values and practices that can lead to new forms of developing software. According to Abrahamsson et al. (2003), the Software Engineering discipline already recognizes and uses most of

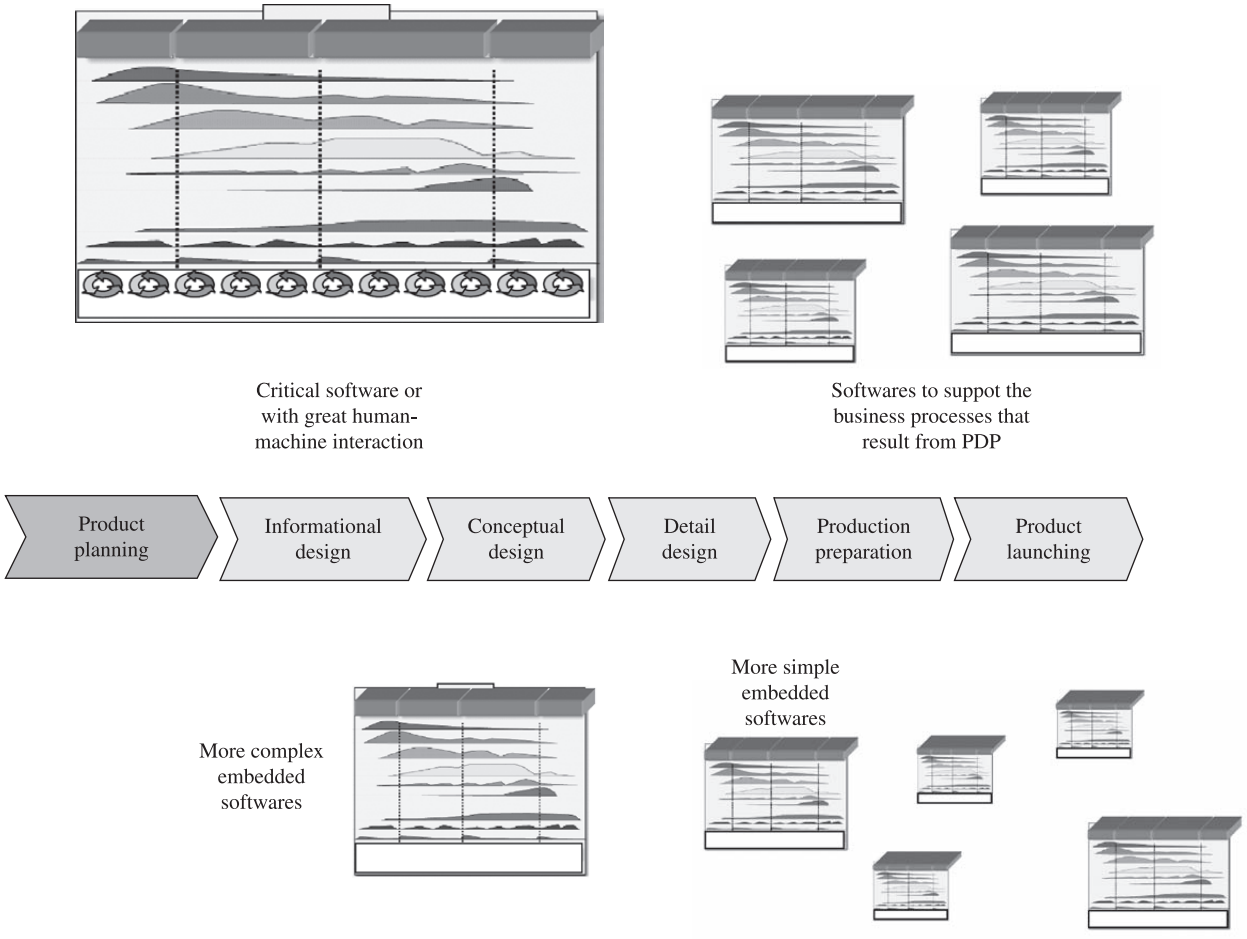

Figure 1. Integration of PDP with the various software development processes – from Rozenfeld et al. (2006).

the XP methods, such as testing and refactoring. However, the XP method has a particularity: the need for synergy among its practices, which complement each other.

The values that guide the XP method are: communication, simplicity, feedback and courage (BECK; ANDRES, 2004). Such values should be presented in the following practices prescribed by the method:

- On-site Customer – the customer should have an effective participation in the project, defining priorities, testing the functionalities and giving feedback to the development software team.
- Planning Game – at XP, the development of the software is an iterative and incremental process. At the end of each iteration a new functional software is released to the customer. During the planning game, the development team and the customer define together what should be implemented in the next iteration and estimate the time needed for that.
- Metaphor – developers and the customer discuss and define the product through shared metaphors.
- Simple Design – the system should be designed as simple as possible, and should represent the real needs of the customers, without duplicities or ambiguities.
- Small Releases – the system (incomplete yet) should be quickly put in production and after that developers should make frequent small releases, so customers can use them and give feedback.
- Pair Programming – two developers produce most of the software together, taking turns during the process.
- Testing – at XP, the tests are intensified and automated.
- Refactoring – in order to develop software in an incremental way, frequent revisions and improvements should be accomplished in the code, making understanding and maintenance easier.
- Continuous Integration – the code produced is integrated several times a day. Integration tests are made to assure the integrity of the system.
- Sustainable pace – the 40 hours per week work journey should be respected, avoiding stress. Stressed developers are less productive.
- Coding Standards – standard programming patterns are established so developers have a universal language.
- Collective Code Ownership – developers may change any part of the system, as long as its functionality is preserved.

For Shore and Warden (2008), the applicability of XP has much more to do with the company and the involved people than with the type of the project itself. In this sense, Tolfo and Wazlawick (2008) and Tolfo et al. (2009) emphasize that the XP paradigms differ from those found in the traditional methods of software development because they focus heavily on the human relationships. In other words, the conception and the evolution of XP depend on the acceptance, commitment and interaction from people involved; in this case, they are the stakeholders*.

In relation to stakeholders, Beck and Andres (2004), as well as many other authors, does not rigidly establish the roles played by them. The people involved in an XP project are usually classified into two groups: the members of software development team (developers**), and the costumers (or their representatives). These are the people studied in this paper.

3. Organizational structure

Taking into account that this paper considers the adoption of XP Method significantly influenced by organizational structure, we find fundamental to understand this structure and its dimensions.

The company's organizational structure results from a process in which the authority is distributed, activities are specified, and a communication system is established. This structure allows people to undertake their activities and to exercise their authority to achieve organizational objectives. (VASCONCELLOS; HEMSLEY, 1997).

Chiavenato (2002) identifies a number of aspects that are part of the organizational structure, such as how the authority is established; how the roles are distributed; how decisions are made and so forth. So we can state that the organizational structure relates to the way in which the work environment is structured inside the company, and that this structure has a significant effect on the behavior of its members, affecting thus its overall performance.

According to Robbins (1998), there are six key elements (see Table 1) that must be taken into account when designing the organizational structure of a company. These elements are henceforth called 'organization structure dimensions' – or simple 'dimensions'. Such dimensions are discussed in the next section of this paper.

4. Organizational structure and XP method

Analyzing each of the six dimensions of the organizational structure proposed by Robbins (1998) from the perspective of XP practices and values, this section demonstrates how the organizational structure of organizations can influence the adoption of the method. This analysis has also allowed for the identification of favorable and unfavorable aspects of their adoption, as well as the creation of a checklist for every dimension of the organizational structure.

* According to Xexéo (2006), they stand are people interested on the software system, that is, all those who affect or are affected by its results.

** This paper also applies the name 'developers' as a reference to analysts, architects, programmers, testers, all of them responsible for assuring quality as well as those involved in the conception and development of the software.

Table 1. Key elements of organizational structure (ROBBINS, 1998).

The key questions:	The answer is provided by:
To what degree are articles subdivided into separate jobs?	Work specialization
On what basis will jobs be grouped together?	Departmentalization
To whom do individuals and groups report?	Chain of command
How many individuals can a manager efficiently and effectively direct?	Span of control
Where does decision-making authority lie?	Centralization and decentralization
To what degree will there be rules and regulations to direct employees and managers?	Formalization

4.1. Work specialization and XP

Even in dynamic environments as in product and software development companies, there are some issues to be solved and responsibilities to be taken which tend to become routine. However, the adoption of XP requires from developers more responsibilities than usual.

According to Teles (2004), XP developers should analyze, design and codify the system, since in this method there are no clear divisions between analysts, designers and programmers. Each developer performs these different roles at different moments in the project. Anderson and Schragenheim (2003) argues that the XP method needs generalists and experienced developers.

However, we must carefully analyze the implications arising from developers who assume uncommon functions and activities such as testing, collective code and maintain frequent contacts with customers. These developers may not be willing to take these additional responsibilities, or may not have the necessary knowledge or skills to perform other functions.

Companies with rigidly defined roles for developers – database experts, system analysis, graphic interface, quality of software and so forth - tend to be more refractory to the adoption of XP than those who have functional flexibility, i.e. where developers often take different roles and perform different tasks.

Despite of being generalists, members of an XP team should not dispense good technical skills. The adoption of XP practices requires an experienced team. A team primarily composed by novice developers will face major difficulties with XP practices.

In addition to technical skills, XP developers should also have certain personal skills, which are not always found in numerous developers, for instance, self-confidence to deal with the customers, ability to impose opinion or accept someone else's if necessary; capacity to coordinate and communicate intensely with other developers and stakeholders.

Figure 2 presents a checklist related to the Work Specialization dimension.

Due to the remarks and considerations made above, it is possible to identify the organizational aspects shown in Table 2 (favorable and unfavorable to the adoption of the XP) related to the Work Specialization dimension.

4.2. Departmentalization and XP

In Brazil, according to the Science and Technology Ministry (BRASIL, 2004), most software development companies are classified as micro or small. That is why these companies may not have well defined departments to each software development activity (design department, programming department, quality department, etc.); however developers are commonly spread in cubicles or rooms. The companies that develop their products with embedded software or that develop systems related to their product development process, on the other hand, have a software development department interconnected with the other departments of the organization.

Departmentalization may cause competition, work fragmentation and communication difficulty among software developers. XP practices imply integrated team work, which means that if developers working on a same software project are separated in isles, rooms, floors or even different cities, there will be difficulties in adopting XP. Difficulty may increase in the same proportion to physical distance among developers.

The physical organization of the work environment is a relevant influence in the adoption of the XP method. Its practices depend on proper physical space. Just the same, the efficiency of XP values such as communication and feedback depends on people proximity. To McBreen (2002), after experiencing XP it is possible to run it with larger teams and/or distributed, but initially it is recommended for developers to work in a single environment.

According to Williams (2000), it is fundamental to carry out the Pair Programming practice in proper physical environment, providing enough room available, sitting side by side, both having access to keyboard, mouse and pleasant screen viewing. Room must be large so to fit all developers, and no matter what material should isolate the pairs, since chat with other pairs can contribute to determined tasks.

1. Do developers usually carry out more than one function; for example, accumulate roles as Programmer, Designer, and Tester? 2. Considering the developers knowledge and skills, are they able to assume more than one role in a project? 3. Are developers reluctant to assume more personal responsibility? 4. What is the experience level of the software development team?

Figure 2. Checklist - Work specialization dimension.

Table 2. Favorable and unfavorable to aspects for Work specialization dimension.

Dimension: Work Specialization	
Favorable aspects	**Unfavorable aspects**
- Generalist - Experience - Responsibilities - Personal skills	- Specialist - Novice - Low Qualification - Lack of personal skills

A single environment provides low documentation, because information flows easily when done informally. Also, it enables developers to follow up their colleagues' work, which is necessary to change the collective code, to refectory, to define task cards from history cards and to run other activities carried out mainly by all developers.

One must take into consideration whether companies willing to adopt the XP method have a proper physical environment to apply it, making its practices and values effective. If physical environment is improper, investigation will need to be carried out in order to check whether that environment is changeable or not, and especially if people are willing to fit into this new environment.

Creating a single environment for developers used to work in separate cubicles or rooms may not be an easy task. In some cases, removing cubicles or divisors can lead to the feeling of loss of individuality, or loss of individuality over room possession, so that one may understand that privacy limits are not being respected,

It is important for developers to be warmed up to work in a collaborate system. For such, they must identify if there are signs of personal conflict or high levels of competition among developers, which may stop the establishment of Pair Programming, collective code use, functional flexibility and casual and honest communication.

Figure 2 presents the checklist related to the Departmentalization dimension.

Due to the remarks and considerations made above, it is possible to identify the organizational aspects shown in Table 3 (favorable and unfavorable to the adoption of the XP) related to the Departmentalization dimension:

4.3. Chain of command and XP

Software companies may have several types of job hierarchy or some level of job hierarchy. In the company hierarchy may include owners, administrative managers, software managers, product quality managers and etc. The companies that develop their products with embedded software or that develop systems related to their PDP usually have a larger number of hierarchical levels than the software companies. Thus, their software development department may be more susceptible to external influences resulting from other departments and people with greater decision-making power in the Chain of Command.

Companies intending to adopt XP needs to be agile to fulfill demands such as always keeping in touch with the customer, establishing open relationship and communication with the software development team and taking quick decisions about the Project. Thus the number of hierarchical levels can influence in adopting the aforementioned method.

Besides, if adopting the XP depends upon the decision of several company people such as the company owners, managers and sub manager, barriers may occur because of the need for accepting values which may be on the other way around of the market or of common norms of the company such as:

a) Open scope contract – for customer giving up risk, causing insecurity during negotiation;
b) Less documentation - may seem less safety;
c) Pair programming – may include the myth about cost in order to maintain two employees developing the workload of one;
d) More time for tests – can be faced as waste of time.

Meaning, some beliefs and myths about XP – which are usually common – can ruin the method acceptation, and the more people testimony about it, the more chances are to reject it. It is worthwhile to consider that even the choice for a new technology seems a technical decision; it is after all a business decision, which may even depend upon the company's customer.

Higher job positions may be the one in charge of approving and addressing resources to software projects, which is identified as "gold owner" in XP. In the literature available about XP, the role of the "gold owner" is generally neglected for the company assumes they had their approval and financial support for the project.

However, a fundamental issue for implementing XP is acceptance by those in charge of addressing resources to the Project. Except when the project manager has autonomy to decide about the method, the "gold owner" should agree to invest resources on XP projects.

Hierarchical levels can also be established within the software development teams by means of jobs – even informally. Hierarchical levels may be configured on jobs such as team leader, senior programmer and junior programmer.

Tomayko and Hazzan (2004) distinguishes 2 types of structures in software development teams: hierarchy structure teams and democratic structure teams.

1. How is organized the work environment of software developers who are involved in the same Project? 2. Considering aspects such as collaboration, communication, team work and good relationship among developers, is it possible to create a single work environment?

Figure 2. Checklist - Departmentalization dimension.

Table 3. Favorable and unfavorable to aspects for Departmentalization dimension.

Dimension: Departmentalization	
Favorable aspects	**Unfavorable aspects**
- Shared work environment	- Departments and rooms

In the democratic team, members are faced as colleagues and programmers opinions are considered worth the same. Democratic teams tend to level up people and even when necessary to establish hierarchical levels, this occurs more informally so that less experienced developers recognize the ones more experienced in the area, and to whom they will report when necessary (TOMAYKO; HAZZAN, 2004).

Hierarchy structure teams have different levels of hierarchy. In this case, there might be a good degree to delegate powers or highly concentrated top levels. If power is more concentrated, low ranked programmers should transfer higher levels to reach the one who holds decision-making powers.

Even if the software development team has not several hierarchical levels, it is important to identify how the roles defined within the team and what are the implications on people's relationship, that is, what importance do people give to role status. In XP, it is ideal that jobs and levels pass almost unnoticed across the team.

To Sommerville (2000), people from informally structured groups communicate more effectively than formally structured groups. In hierarchical groups, communication tends to flow top-down and people with same level cannot communicate among them.

Difference of status among members of a team may mean that communication is generally carried out in only one direction for superior level members tend to master communication and low-status members are often reluctant in striking up a conversation or making critical observations

According to Robbins (1998), the Chain of Command is directly related to authority degree. The stronger the chain of command, the less information and decision-making members have. The weaker the chain of command, the more information and self-management members have.

Besides, the stronger the Chain of Command, the higher the number of levels communication will go through, increasing the possibility of distorted information and waste of time until communication is fully reached (VASCONCELLOS; HEMSLEY, 1997).

Asproni (2004) recalls that a control command management style is inadequate for an environment in which individuals and communication is more valuable than processes and tools. The hierarchy system flattening allows information spread quick and easy, enabling personal contact change for writing communication. Therefore, communication becomes more based on trust than roles and rules.

XP do not prescribe that jobs should be dismissed, but requires from developers autonomy for making decisions about the software development process and interaction with the customer. However this XP request can cause resistance when managers do not trust their team potential, or when the company structure is hierarchical.

Figure 3 presents the checklist related to the Chain of Command dimension.

Due to the remarks and considerations above, it is possible to identify the organizational aspects shown in Table 4 (favorable and unfavorable to the adoption of the XP) related to the Chain of Command dimension:

4.4. Span of control and XP

Leblanc (2004) reports on some XP issues which deserve observation. Among them, the author calls attention to the size of the software development team. According to Leblanc, the XP was idealized for small-scale teams containing 2 to 12 developers. This is based on the premise that small teams are more flexible and adapt better to project changes.

Leblanc (2004) also takes into consideration that teams should start with two people so that the Pair Programming occurs. Both people take turns as made necessary, sharing ideas to find the best solution for a problem.

Also, agile processes such as XP require small-scale teams because in such conditions, team work happens smoothly as well as to the feeling that each employee is important. That helps motivating developers to be part of the evolving process, improving their productivity, quality and performance (ASTELS; MILLER; NOVAK, 2002).

The more the group enlarges, the more difficult it is to ensure that all members communicate efficiently (SOMMERVILLE, 2000). In the XP method, communication is overvalued because developers must know about their colleagues' activities to work in a very intense interactive environment.

According to Hampton (1990), size and structure of the group may influence its capacity to operate satisfactorily and efficiently, in which several studies confirm size effect over the capacity of groups to solve problems.

From those studies, Hampton (1990) does some generalizations: groups of approximately 5 to 11 members tend to make more precise decisions than groups out this rate; smaller groups are more capable of having agreements

1. How many hierarchical levels (related to the software development) are there in the company? 2. Are there hierarchical levels (formal or informal) in the software development team? 3. How is the communication among the members of the development team? 4. How is the communication among the development team and management?

Figure 3. Checklist - Chain of command dimension.

Table 4. Favorable and unfavorable to aspects for Chain of command dimension.

Dimension: Chain of command	
Favorable aspects	**Unfavorable aspects**
- Few hierarchical levels	- Several hierarchical levels
- Informal relations	- Formal relations
- Democratic teams	- Hierarchical teams
- Democratic organization	- Hierarchical organization
- Rapid and open communication	- Slow and reserved communication

than large groups; large groups may suffer subgroup formation with their own goals, incompatible with the large groups.

Figure 4 presents the checklist related to the Span of Control dimension.

Due to the remarks and considerations made above, it is possible to identify the organizational aspects shown in Table 5 (favorable and unfavorable to the adoption of the XP) related to the Span of Control dimension:

4.5. Centralization and decentralization and XP

Fowler (2003) highlights that for a people-oriented process such as XP, empowerment is fundamental for developers to take technical decisions and provide estimates. The author also states that such change should directly influence in the autonomy and authority leveling among managers and developers.

XP practices such as On-Site Customer, Planning Game, and Small Releases require action power and reaction power by developers. In order to hold such power, the company must grant more authority and information to people closer to the 'action' – that is, to whom is closer to the product (software) or the customer.

In the XP, speed and efficiency establish the need that several problems should be solved by developers themselves. If that do not happen, the development team may get stuck to bureaucracy, preventing the use of the method.

Moreover, XP practices such as Planning Game will only make sense in an environment where developers feel they are heard and respected by their manager.

If investors, owners or managers are very demanding and controlling, they may face empowerment as risk increase for the company. Not empowering and, therefore, restraining developers' decision-making, may turn companies prone to a prescriptive software development process.

On the other hand, in order to hierarchical superiors become willing to empower, they must trust the software developers, specially their knowledge, meaning managers should feel that developers are able to make decisions. Besides, developers should be willing to take more risks and responsibility, since they receive delegation, work and risk margin increase.

In XP, decisions about the business must be taken by customers or by managers, and technical decisions must be a development team responsibility. Beck and Andres (2004) enumerates the following topics to be decided together by the team:

a) Estimates – how much time will be taken in order to implement a functionality;
b) Technical alternatives consequences – estimating technical consequences, the team provide data to the manager make strategic decisions about the business;
c) Process – how work and team will be organized, set of practices to be used, processes to revise practices;
d) Schedule - within delivery cycle, decide what stories start first.

Thus, in order to use the XP method, the work environment should be collaborating and cannot be restrained by rules predetermined by superiors. However, there must be a good set of choice available to developers. Besides, managers should play the role of motivators and facilitators in the software development process.

Managers should establish leadership to enable dissemination of organizational values required by the XP method. Chiavenato (2002), based on White and Lipitt studies, describes three types of leadership:

a) Autocratic Leadership: only a leader decides and establishes the guidelines, not participating in the group, determining which tasks each will perform;
b) Liberal Leadership: there is total freedom to group or individual decision-making. Leaders only participate when presenting alternatives and providing information to their groups. Tasks are up to the group;
c) Democratic Leadership: guidelines are discussed and decided by the group stimulated and observed by the leader. The group itself defines providences and techniques to achieve goals with technical counseling from the leader. Therefore, when necessary, tasks get new perspectives and debates. Task division is up to the group.

Even counting with ideal developers for the XP, adopting it may become impossible if the company managers have an autocratic leadership with a mechanistic view, in which decisions are made top-down.

1. How many developers usually work in the same software Project?

Figure 4. Checklist - Span of control dimension.

Table 5. Favorable and unfavorable to aspects for Span of control dimension.

Dimension: Span of control	
Favorable aspects	**Unfavorable aspects**
- Small teams	- Large teams

In the XP method, developers must find by themselves how to improve and make their job quick. If an organization is fundamentally hierarchical and top-down, a software process more passive of control may be quite adequate – traditional method. However, agile methods such as XP are more adequate and adjust easily in a collaborate culture (HIGHSMITH, 2002).

It will probably be difficult to migrate from one mode of job in which detailed orientations on the tasks are provided and move to another type of work in which decisions should be taken by developers together with the customer, so that it is necessary to take risks and empower.

Empowerment can also have implications in the software development team itself. The Collective Code XP practice provide a clear example about this situation: A less experienced programmer needs to be at ease within the team and be courageous to change a more experienced code or question some decision on the software development. Hierarchical teams with uneven power provide an unfavorable environment to the XP adoption.

Moreover, in the XP method, authority may change depending who is more competent to perform a given task, and communication content tend to have more information and advices than orders or instructions.

Figure 5 presents the checklist related to the Centralization and Decentralization dimension.

Due to remarks and considerations above, it is possible to identify the organizational aspects shown in Table 6 (favorable and unfavorable to the adoption of the XP) related to the Centralization and Decentralization dimension:

4.6. Formalization and XP

Formalizing job positions affects the degree of autonomy of whoever is occupying the job position during the decision-making process. Highly formal companies provide a minimum degree of autonomy to this person do what he or she is supposed to do (ROBBINS, 1998).

High formalization ensures that the members act in the same and predictable way, being related to the level managers believe on their subordinates' autonomy capacity. Thus, the higher the formalization level (documents and norms), the less managers will trust on the members of the team (TRACY, 1992).

If formalization is high in the company, there will be need for documentation. That is, communication among programmers, manager and customers happen mostly through documentation because of direct communication in a way that information is sent slowly, being possibly lead to misinterpretations

According to Ambler and Jeffries (2002), several organizations have overvalued documentation on their software projects especially because of the fear of losing the development team members who have information and knowledge non-documented.

The author considers such attitude an impotent belief when put into practice because according to him, developers rarely follow documentation left by someone else, and tend to investigate the code left, or start new implementation. Ambler and Jeffries (2002) suggests that excessive attention about documentation should give place to more important concerning such as developing a high-quality code.

XP represents an alternative to the issue of documentation excess so as to assure everyone's knowledge about the project, which can be observed by practices such as Collective Code, Refactoring, Programming Standards and Pair Programming. Such practices allow everyone to know what their workmates are developing. They also allow the system to be simple and revised enough for easy comprehension so that the absence of a team member will not cause misunderstandings about the project.

Therefore, software companies which request high level of documentation can find difficulties in migrating to the XP method, which require documentation as only essential to understanding the system, being most part of it converted to oral communication. In this case, unfavorable aspects in the adoption of XP come from the difficulty members have in letting go the safety they find in documentation.

To Ambler and Jeffries (2002) one of the main issues that raise difficulties to adopting an agile approach to software development companies is the preference for very prescriptive software processes. According to Ambler and Jeffries (2002), this aspect is more common in big companies which have their software development process well defined. It is possible to assume that such companies – given their history and importance – have a considerable level of departments, rules and formalism, though such feature may as well be in some smaller software companies.

Ambler and Jeffries (2002) calls attention to the possible structural challenges in adopting an agile methodology to companies seeking certification or any quality program such as ISO or CMM for these companies often follow more prescriptive and well documented processes. Although some studies have shown that it is possible to compatibilize these technologies (PAULK, 2001; JEFFRIES, 2000), certainly the XP method barriers should increase in companies with this type of profile.

According to the MCT (BRASIL, 2004) data, in the national context only 7% of the software companies were ISO or CMM, certified until 1999. This number may rely on the fact that these programs are designed for large-scale companies, being difficult to adjust them to the reality of small companies. Small-scale companies usually have a reduced control sphere, few hierarchical levels and generally do not show great formality. In principle, such features represent a proper environment for XP. However, low formalization may often characterize organizational flaws such as informality in the software process and inefficient management system.

With regard to the companies that develop their products with embedded software or that develop systems related to their PDP, Rozenfeld et al. (2006) points out that these companies have the software development as an additional component of their product development, and would possibly have great difficulty in adopting a model such as CMMI, because the software is not the focus of their businesses.

Figure 6 presents the checklist related to the Formalization dimension.

Due to the remarks and considerations above, it is possible to identify the organizational aspects shown in Table 7 (favorable and unfavorable to the adoption of the XP) related to the dimension Formalization:

1. Does the team actively participate on solutions to problems related to the project? 2. What is the leadership style of the company?

Figure 5. Checklist - Centralization and Decentralization dimension.

Table 6. Favorable and unfavorable to aspects for Centralization and Decentralization dimension.

Dimension: Centralization and Decentralization	
Favorable aspects	**Unfavorable aspects**
- Empowerment - Democratic leadership	- Hierarchical models of control - Autocratic leadership

1. Are management orientations passed to developers with precision and in detail or general, leaving room for choice? 2. What is the software development level required in the company? 3. Does the company have a quality program or had any type of certification?

Figure 6. Checklist - Formalization dimension.

In the end of the analysis of each of the organizational structure dimension, under the perspective of XP practices and values, one can perceive that the questions elaborated in each dimension form a checklist. That is, this set of questions becomes a tool which can be applied by software companies to point out their specific structure aspects.

5. Case study: the influence of organizational structure in the adoption of the XP method for the Portal Company

This item presents the application of the Checklist in a software company. Through this case study, we intend to empirically prove the influence of the organizational structure on the adoption of the XP method.

In this paper, we have named the company object of the case study as Portal Company, in which its main business is to develop internet solutions such as corporate portals and content management. Besides questionnaires, visits to the company, work follow-up, we have also interviewed developers and stakeholders.

The organizational structure of the Portal Company has been illustrated into two ways. First, an overview so as to observe people allocated in the chart of the entire organization. Parallel to that, we have illustrated in a more specific way, privileging the structure in which developers are allocated on the software projects – for instance, in the case of a functional structure, projects or matrix.

In the overview, we have noticed that the company's organizational structure is constituted by strategic, tactical and operational levels. In the strategic level, there are the stakeholders, who are facing a relationship with the market and business. Among stakeholder are the board, council, administrative management and project management.

In terms of tactic and operational, there are people more directly involved with the software development. We have perceived a model similar to project-oriented organization structure for each manager coordinates their teams according to a project portfolio. The latter features a more specific view of the organizational structure.

From these observations (general and specific) on the Portal Company, it is possible to view each of the organizational structure dimension – which will be presented next. We highlight that the reports below are based on questionnaires which form the checklist presented on item 4. Meaning, we have carried out investigation to answer the checklist questionnaires.

Table 7. Favorable and unfavorable to aspects for Formalization dimension.

Dimension: Formalization	
Favorable aspects	**Unfavorable aspects**
- Less emphasis on regulations and standards - Less documentation and face-to-face communication	- Strong emphasis on regulations and standards - High level of documentation and written communication rather than oral communication

In the Portal company projects, especially in tactics and operational, we have noticed that each participant has their duties well defined according to their specialization (Work Specialization dimension). Among team members, there are business analysts who represent the customer, senior analysts who play the role of project leader, full and junior analysts who receive tasks related to their level of knowledge. Also, database administrators, homologators and testers participate on it in their own knowledge area.

Although there are cubicles and room division in this organization there are no established department, given the small size of the company and nature of the organizational structure resemblance to a project-oriented arrangement (Departmentalization dimension).

In this company there is low formalization and low span of control because software development companies are composed by twelve people maximum, in an informal environment and low documentation production (Formalization Dimension and Span of Control dimension

In the several levels on the Portal company levels, we have observed a hierarchical chain of command (Chain of Command dimension). Fact that leads to this evidence is the strong influence investors have on managers, who in turn transfer it to the software development team. These investors are extremely oriented for business profiting and at the end they estipulate to managers unreal software modules deadlines. At the end, managers transfer these deadlines to their teams.

Before the Portal Company's project-oriented structure, we have found that managers take all the decisions, stipulating the activities of each developer according to their area and knowledge level (Centralization and Decentralization dimension).

As the company has certain favorable aspects on the adoption of XP, such as small project-oriented teams, low formalization and weak departmentalization, we could assume that the method will be successfully accepted and implemented. However, additional information has changed this picture. Initiative on adoption of the XP method in the company, though considered successful by developers, had no continuity.

Because of developers' requests, the company hired training services and XP consultancy. After each training step, it was designed an experimental project for developing a business intelligence tool using the XP premises.

In this project it was allocated a self-manageable team, which started to dedicate themselves exclusively to developing such tool. Results are satisfactory. The project achieved deadlines and stipulated goals and after the efforts undertaken, it adopted most XP practices. However, at the end of the project, the team was broke up and its members were back on their job together with the groups coordinated by project managers, just as it used to be before.

From the developers' reports and activities experienced in the company, we have perceived that some aspects of the organizational structure unfavorably influenced the continuity in the adoption of the XP method. Among these aspects are the actual leadership style, power relations and stakeholders' different perspectives, all distributed by the Chain of Command of the organization structure.

Concerning the centralization and decentralization on decision-making, the idea of establishing self-manageable teams was not well accepted by the project managers, despite of the gains obtained with the experimental XP project in the company. Empowerment and autonomy for teams were established as a threat to superiors' authority and as a way of estimation decrease and increased risks in the projects. Such perceptions are linked to power centralization and autocratic leadership in the company.

Also, we have noticed resistance to the adoption of the XP method on the other hierarchical levels of the chain of command. Initiatives in the adoption of the method were stopped for investors and directors had interpreted it as a change which would possibly end on instability, drop on production rates during adaptation period and increased risks on business. That does not suit the investors' profile, which is extremely profit-oriented.

At last, although the software development team accepted the XP method, we have noticed that the stakeholders' different perspectives composing the company organizational structure have rendered the implantation of the method unfeasible. The managers and investors attitudes may be understood as a way to keep themselves within a comfortable zone, avoiding new situations which would bring change for their work routine and therefore implicating in risks for the company business.

6. Conclusion

The decision of implementing the XP method assumes a systemic reflection concerning the various factors which may have implications on this initiative. It is necessary to take into account not only the technical aspects (which are majorly approached on our literature), but also cultural*** and organizational aspects may be decisive for successfully adopting the agile method. In this context, the paper privileged the organizational aspects and aimed at identifying aspects in the structure of software companies which influence in the implementation of XP.

By means of analysis of the organizational structure dimensions under the perspective of practices and values of the XP, we have elaborated a set of questions which help identifying favorable or unfavorable aspects in the adoption of XP. This set of questions form the checklist, which may be used by software companies to point at their specific structure aspects.

*** See: Tolfo and Wazlawick (2008).

On the other hand, software development companies may also use the favorable and unfavorable aspects identified in this paper as a parameter for previous remarks about the convenience of implementing the XP method. That is, based on aspects pointed out in this article, they may easily find facilitators and obstacles to XP imposed by particularities of their organizational structure.

In order to empirically show the influence of the organizational structure in the adoption of the XP method, we have applied the checklist in a software company (Portal Company). Questionnaires, interviews and observations carried out at the software company are tools which have enabled validating the checklist and proving the influence of the organizational structure in the adoption of the XP method.

Besides proving the existence of organization aspects identified in this paper, the study case have revealed that although the referred company had some favorable aspects concerning XP, other organizational aspects had a major value in relation to the method implementation.

Meaning, small teams allocated in informal environment with low documentation and absence of departments have shown favorable aspects to the adoption of the XP method, though not enough before a centralizer management style and influences unfavorable to XP found on the chain of command.

Organizational aspects unfavorable to the adoption of XP prevented continuation in using this software development method. This fact proves the decisive influence the organizational structure can do when implementing the method.

The research carried out also aims at calling attention to the fact that although the XP has been run by small teams working in projects with vague requirements, the company adopting it will not always have success for organizational factors may even prevent its usage.

Another interesting fact found in the study case is that some of the stakeholders had an equivocated perspective about the agile methods purposes, especially XP. There was some expectation from the XP for providing solutions to immediate and daily issues. Such reductionist and mechanicist view did not allow some stakeholders to understand that the XP consultancy should be a starting point to the process of cultural change in order to adopt agile principles and values. To analyze this fact suggests to check the influence of mentor and coach in agile approaches[****].

This has been corroborated throughout the study carried out. Some interviewees have miscomprehended that the XP method is a set of complementary values and practices which conduces an agile philosophy. That is, some developers prized the XP practices aiming at solving operational and punctual issues of specific projects, neglecting the possibility of significant changes, such as the creation of an agile culture.

This limited vision of some developers of the company Portal on the agile principles was a lesson learned by GEPP with regard to their possible inclusion in the PDP. It was observed that the companies should focus not only on operational improvements and resolution of punctual problems of their PDP. They should focus primarily on the adequacy of the agile principles and values to this process.

**** See: Tolfo, Vicentini and Forcellini (2010).

7. Acknowledgement

The authors would like to thank CNPq (Brazilian government agency for scientific and technological development) for the financial support that made this work possible.

8. References

ABRAHAMSSON, P. et al. **Agile software development methods**: review and analysis. Espoo, Finland: Technical Research Centre of Finland, VTT Publications, 2002. Available from: <www.inf.vtt.fi/ pdf/publication/2002/p478. Pdf>. Access in: 04 jan. 2008.

AMBLER, S. W.; JEFFRIES, R. **Agile modeling**: effective practices for extreme programming and the unified process. New York: John Wiley & Sons, 2002.

ANDERSON, D. J; SCHRAGENHEIM, E. **Agile management for software engineering**: applying the theory of constraints for business results. Prentice Hall PTR, 2003.

ASPRONI, G. **Motivation, teamwork, and agile development**. The Agile Times. 2004. v. 4, n. 1. Available from: <www. agilealliance.org/membership/vol4.pdf>. Access in: 20 Dec. 2004.

ASTELS, D.; MILLER, G.; NOVAK, M. **A practical guide to extreme programming**. Prentice Hall, 2002.

BECK, K.; ANDRES, C. **Extreme programming explained**: embrace change. 2nd ed. Addison-Wesley, 2004. (XP Series).

BRASIL. Ministério da Ciência e Tecnologia - MCT. Available from: <http:\\www.mct.gov.br>. Access in: 10 dez. 2004.

CHIAVENATO, I. **Gerenciando pessoas**. São Paulo: Prentice Hall, 2002.

CHIN, G. L. **Agile project management**: how to succeed in the face of changing project requirements. Amazon, 2004.

COOPER, R. G. **Winning at new products**. 3th ed. Cambridge: Perseus Publishing, 2001.

CUNNINGHAM, W. **General chair address**. 2003. Available from: <http://www.agileuniverse.com/schedule/ GeneralChairAddress>. Access in: 5 nov., 2007.

FOWLER, M. **New methodology**. 2003. Available from: <http://www.Martin fowler.com/articles/newMethodology.html>. Access in: 22 ju. 2008.

HAMPTON, D. R. **Administração**: comportamento organizacional. São Paulo: McGraw-Hill, 1990.

HIGHSMITH, J. **Agile software development ecosystems**. Addison Wesley, 2002.

HIGHSMITH, J. **Agile project management**: creating inovative products. Boston: Addison-Wesley, 2004.

JEFFRIES, R. **Extreme programming and the capability maturity model**. 2000. Disponível em: <http://www.xprogramming.com/xpmag/xp_and_cmm.htm>. Acesso em: 23 jun. 2005.

KARLSTRÖM, D.; RUNESON, P. Integrating agile software development into stage-gate managed product development. **Empirical Software Engineering**, v. 11, n. 2, p. 203-225, 2006.

LEBLANC, D. **When eXtreme programming makes sense**. 2004. Available from: <http://www.devsource.ziffdavis.com/article2/0,1759,1609523,00.asp>. Access in: 5. Jan. 2005.

McBREEN, P. **Questioning extreme programming**. Indianapolis: Addison Wesley, 2002.

PAULK, M. **Extreme programming from a CMM Perspective**. 2001. Disponível em: <hristov.com/andrey/fht-stuttgart/xp-cmm-paper.pdf>. Acesso em: 21 jun. 2007.

ROBBINS, S. P. **Organizational behavior.** 8th ed. Prentice-Hall: Englewood Cliffs, 1998.

ROZENFELD, H. et al. **Gestão de desenvolvimento de produtos**: uma referência para a melhoria do processo. São Paulo: Saraiva, 2006.

SHORE, J.; WARDEN, S. **The art of agile development**. O'Reilly Media, 2008.

SMITH, P. G. **Flexible product development**: building agility for changing markets. San Francisco: John Wiley & Sons, 2007.

SOMMERVILLE, I. **Software engineering**. 6th ed. Addison Wesley, 2000.

TELES, V. M. **Extreme programming**: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade. São Paulo: Novatec, 2004.

TOLFO, C. et al. Agile methods and organizational culture: reflections about cultural levels. **Software Process Improvement and Practice**, 2009. DOI: 10.1002/spip.436

TOLFO, C.; VICENTINI, L. C.; FORCELLINI, F. A. O papel do mentor e do coach em abordagens ágeis: descrição de um caso. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 2010, São Carlos. **Anais**...

TOLFO, C.; WAZLAWICK, R. S. The Influence of organizational culture on the adoption of extreme programming. **Journal of Systems and Software**, 2008.

TOMAYKO, J. E.; HAZZAN, O. **Human aspects of software development**. Charles River Media, 2004.

TRACY, D. **Ten steps to empowerment**: a common sense guide to managing people. 4th ed. Quill, 1992.

VASCONCELLOS, E.; HEMSLEY, J. **Estrutura das organizações**: estruturas tradicionais, estrutura para inovação, estrutura matricial. São Paulo: Pioneira, 1997.

WILLIAMS, L. **The collaborate software process**. Dissertation (PhD)–University of Utah, 2000.

XEXÉO, G. **Modelagem de sistemas de informação**: da análise de requisitos ao modelo de interface. Available from: <http://ge.cos.ufrj.br/tikiwiki/tiki-download_file.php?fileId=1>. Access in: 15 sep. 2007.