# Chapter 19
# Human Resource Related Problems in Agile and Traditional Software Project Process Models

**Stefan Koch**
*Bogazici University, Turkey*

**Gerhard Turk**
*Vienna University of Economics and Business, Austria*

## ABSTRACT

*This paper explores the impacts associated with different software process models on the staff. The main research question addressed is whether any process model leads to a higher satisfaction of staff, and less human resource related problems, including staff turnover or increased stress levels. This issue is empirically investigated using a set of projects from 15 different software developing companies. Agile-oriented models are not necessarily limited to small projects, but both groups showed nearly identical distributions for team size and duration. Interestingly, rigid-type models tend to exhibit higher effort estimations, and lower correctness in these estimations. Also customer satisfaction is slightly lower. With regard to human resource issues, the differences are not major overall, but there are some noticeable exceptions. In general, satisfaction and acceptance are higher at lower stress and overtime levels for agile-type project participants, but, interestingly and contrary to theory, people wish for more responsibility. Agile-type projects also enjoy some advantages in information sharing and communication, and in some quality aspects. Rigid-type projects show considerable higher abilities to cope with absence of personnel.*

DOI: 10.4018/978-1-4666-1601-1.ch019

## INTRODUCTION

Software development is, above all, a task performed by human beings. Nevertheless, human factors and human resource management have not been a major issue in the context of software engineering and software development, with some notable exceptions (Brooks, 1999; DeMarco & Lister, 1999; DeMarco, 2002; Cherry & Robillard, 2008; Good & Romero, 2008; Tenenberg, 2008). In this paper, we will explore the impacts associated with different software process models on the staff. With this term, we subsume all people directly involved in the respective projects, and therefore include analysts, programmers, management and others.

The main research question addressed is whether any process model leads to a higher satisfaction of staff, and less human resource related problems, including staff turnover or increased stress levels. Naturally, impacts of staff satisfaction on the quality of the final product, and on efficiency and effectiveness of work should not be underestimated. For example, eXtreme Programing (XP) explicitly has a "no overtime" rule designed to maintain spirit and motivation (Beck, 1999). We addressed this question with a literature review on human resource related topics in software development, and an empirical study. The empirical study included interviews with a number of project participants, mostly project managers from different software developing companies.

The outline of the paper is as follows: We will start with a literature review which covers the types of software process models included in our study, as well as possible human resource related problems and prior research on the intersection of both issues. Then we will detail the empirical study undertaken, starting with a description of the methodology, followed by the results themselves. The paper closes with discussion and directions for future research.

## LITERATURE REVIEW

### Process Models in Software Development

The concept of a process model in software development is still not uniformly defined, and some of the approaches to be found in the literature are not detailed enough to be called a full-blown process model. The basic idea we will use is that a process model contains a list of time-successive steps in a problem-solving process, thus constituting a guideline for developing software, structuring the overall project, helping in achieving goals including quality levels, and aiding in planning, controlling and monitoring (Liu et al., 2008). Also a process model helps in giving a common understanding to all involved parties, and defining clear interfaces between different tasks. Process models also gain special importance in multi-project management (Demirkan & Nichols, 2008).

In the literature, a plethora of approaches can be found including the V-model, SA/SD, XP, Scrum and many more. Although they are difficult to group, for this paper we will adopt a widespread differentiation between those models which are of a more consecutive, waterfall-based nature (Royce, 1970) like V-model or SA/SD, for which Syed-Abdullah et al. (2006) also use the term designed-based approaches, and those which show more prototyping, spiral-oriented features (Boehm, 1988). This last group is currently primarily embodied by agile approaches like Scrum or XP, which are based on the Agile Manifesto. This holds as central tenant that both the business and technology environment continue to change at an increasing pace. Therefore the main question to be addressed is how to better handle these changes, not stop them early in the life cycle by anticipating them all, while still aiming at achieving high quality and timeliness. A focal point therefore is the flexibility that is to be ensured (Ollson, 2008). This leads to a set of principles, which value individuals and interactions over processes and tools, work-

ing software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan, while explicitly acknowledging the value of the items on the right. While all the instruments of more process- and planning-oriented methods are recognized as useful, they are the ones seen as needing to give in problematic situations.

Quite often, the right choice of model is a huge challenge. Lots of research has focused on comparing different models, and trying to determine in which contexts different approaches have merit or not (Boehm & Turner, 2003; Reifer, 2002; Stephens & Rosenberg, 2003). For example, agile approaches have faced a lot of criticism, one of them being that planning, processes and documentation are essential, and agile development is nothing more than an excuse for hackers to do as they like, coding away without planning or design (Rakitin, 2001). Another point of criticism is the reliance of agile methods on individual competency and craftsmanship, with participants working together in self-organizing teams even including members of other organizations, e.g. customers (Cockburn & Highsmith, 2001; Lindvall et al., 2002). It is argued that the number of developers possessing excellent technical knowledge paired with the necessary social skills is naturally limited. Also the possible size of agile development teams is discussed. While Cockburn and Highsmith (2001) cite successful projects with up to 250 people, other authors see a size limit in the area of 15 to 20 people. For any analysis, it is important to note that in many cases, a model is not used according to a certain book or source, but is tailored to fit a certain project (Garcia-Crespo et al., in press). Also the distinction between the types of process models as described above is not as distinct in practice many times. For our empirical study, we have therefore not used a binary variable, but a 6-grade Likert scale to rate between full agility and full rigidity. This is somewhat akin to what Boehm (2002) and Boehm and Turner (2003) have proposed and used in form of a continuum.

## Human Resource Problems in Software Development

In all team-based settings, a series of problems can arise during work. The first determining factor in this is the selection of team members which can work together without major frictions (Reel, 1999), and to optimize cost, duration and risk (Tsai et al., 2003). Especially in the initial phases of a project, distribution of work packages is often the most pressing task, during which necessary communication phases are neglected. Main sources for team problems are in most cases deadlines, differing opinions on a factual level, personal difficulties among staff, opaque process steps and a lack of well-defined interfaces. Naturally, the end of a project puts the members of a team in most cases under the highest time pressure, and therefore results in mental phenomena such as mild irritability. For international projects also the mentality of different nationalities can play a major role in stressful situations, or even cause them. It is not always possible to keep every employee in the team, so some team members might need to get dismissed, which for projects is also often difficult due to accumulated experience and increased overall pressure. Sengupta et al. (1999) found that staffing and assimilation delays cause considerable problems for managers. Regarding time problems, two different aspects need to be considered: On the one hand, scheduled deadlines must be respected and, on the other hand, managers often have little time for the employees' needs. Managers are responsible for ensuring that milestones are met. If they also have very little time for the staff, and thus are not in good and permanent contact with the employees, they might become aware of any problems too late. In this research, we will focus on any such problems related to the staff. This includes the number of people reporting illnesses, especially before major milestones, the number of absences without valid excuse, any contract terminations by employees, employee satisfaction and complaints, cases of fake work

or refusal to work, interpersonal problems like boycotts of certain co-workers, using work assets for personal tasks or faking time sheets, errors, delays and quality problems, as well as employee motivation and stress level in general.

## Process Models and Human Resource Related Effects

To date, there is not much research linking human resource related problems and the process models used in software development projects. The process model followed structures both the flow of work and also the different interactions, and thus can be hypothesized to have an impact on employees, their satisfaction and stress levels. For example, Cherry and Robillard (2008) empirically study ad-hoc collaborations which are spontaneous and mostly based on face-to-face communication. In agile approaches, these collaborations are much more pronounced than in traditional approaches, which mostly rely on documentation (Cherry & Robillard, 2008). It was found that these collaborations take significant time, but are rarely taken into consideration during task assignment. Nevertheless they are important for knowledge sharing. The increased presence of those interrupting collaborations in one type of process model could therefore increase satisfaction with knowledge transfer and social effects, but might increase stress levels due to neglect in task assignment and frequent interruptions. Jiang et al. (2003) also identify the organizational environment, like centralization and especially formalization, which can directly be linked to process models, as an important factor for team effectiveness. Tenenberg (2008) argues for institutional analysis of software development teams in the lens of collective action problems to uncover policies that enhance cooperative and restrict self-serving behaviour, the latter of which can lead to human resource related problems. In the conclusions, he directly links to agile processes.

In the most similar research to the one presented here linking human resource related problems to process model adopted, Syed-Abdullah et al. (2006) empirically analyzed the effects of agile methods, namely XP, on the well-being of a set of 17 development teams composed of undergraduate students. The main effects on well-being in general are job design, performance monitoring, human resource practices and team leader's support. They employed participative observation, as well as focus group interviews and questionnaires. Initial results show that XP has a positive effect on the level of enthusiasm (as opposed to depression) in the most dynamic project, but no statistical effects on anxiety – contentment. In another empirical paper, Pikkarainen et al. (2008) show that agile practices improve both informal and formal communication. While the use of SCRUM and some XP practices was found to facilitate team and organizational communication of the dependencies between product features and working tasks, the use of agile practices requires that the team and organization use also additional plan-driven practices to ensure the efficiency of external communication between all the actors of software development. Cohn and Ford (2003) detail their experiences in introducing an agile methodology (SCRUM) to organizations. They explain that in such cases, some developers actually resist, because they might enjoy creating non-code artifacts, or that they dislike the more frequent interactions with management. They also report cases where developers became overwhelmed by the freedom of not having a detailed plan governing their actions. In one case, a group of overly enthusiastic people adopting agile practices too quickly caused problems and frictions within the team. In some cases, the dissatisfaction even turned into complaints at the HR department with regard to how the project was managed. All of those effects either directly constitute human resource related problems, or might be the cause of stress or dissatisfaction.

*Table 1. Demographic data on participating organizations*

| | N | Min. | Max. | Mean | Median |
|---|---|---|---|---|---|
| **Age (years)** | 13 | 5 | 50 | 19.2 | 15 |
| **Number of employees** | 9 | 20 | 92000 | 21564.4 | 110 |
| **Number of countries active in** | 13 | 1 | 80 | 13.9 | 2 |
| **Turnover in 2008 (Mio. EUR)** | 4 | 3 | 11764 | 5298.9 | 4714 |

Additional empirical data on agile methodologies were reported by Sfetsos et al. (2006), based on interviews with 30 managers and developers from 15 Greek firms applying XP. Among other factors, they found communication and synergy between skilled personnel to be a significant success factor.

## METHOD AND DATA SET

### Methodology

For this empirical research, we used a set of 15 randomly selected companies in Austria. In this set, both smaller companies, but also large and multinational organizations are represented (see Table 1 for data on the organizations).

Within each company, an interview with one person, quite often a project manager, was performed. In all cases, the most recent project was used as reference. For this the project had to be completed or at least very near to its end. All data was anonymized before any analysis, no personal data about the interview partners were collected. The majority of the interviews was conducted live, one of them was done by telephone due to scheduling problems. The interview was a semi-structured one, using a guideline containing a set of closed questions and also several open questions to catch any comments not covered by the pre-conceived list. This mix was chosen so as to be able to have some quantitative analysis for common problems and attributes, while being able to capture more rich data in an interview setting, or to probe deeper in some cases. Therefore, we chose not to use a questionnaire and mail or e-mail it to possible respondents. The interview guidelines and scales were pre-tested once with an IT project manager, which resulted in small changes to the wording of some questions.

Most of the questions had to be rated on a 7-point Likert scale, using +3 to -3, with the endpoints question-specific. This form was mostly used for the questions pertaining to human resource related problems or motivational aspects, but also, e.g., for resulting customer satisfaction ratings. Of considerable importance was also the question for rating the process model itself, here a 6-point scale between very agile (1) and very rigid (6) was employed. As has been discussed above, process models are almost never used according to book, and are difficult to put in one simple category. This is somewhat similar to the continuum that Boehm (2002) and Boehm and Turner (2003) have used. For our data set, the majority of respondents classified their own process model as rather an agile one. That ratio is exactly 2:1. That means there were ten project managers that modeled their actions as rather agile and five classified the process model to be of a rather rigid nature. Only two project managers indicated that they have used a comprehensive process model as described in the literature, and in only two cases the extreme positions in the spectrum were selected. This underlines the importance of not sticking to binary categories, and published process models, but allowing for company and project specific variations. In an additional part of the interview, quantitative data

on the project, like size or duration, and also on results were collected.

For analysing the data, we try to establish in most cases a relationship between the process model adopted and the dependent variable, mostly ratings for human resource related issues. Descriptively, we report mean and median ratings for both groups. To test these for differences, we use the exact Wilcoxon rank sum test, which can deal with small sample sizes, and also with ties that are relatively frequent in this data set. In this approach, we use the classification into two groups (as can also be seen in Table 2). In a separate approach, we employ a series of univariate OLS regression analyses to establish in each case a simple linear model between the dependent variable considered, e.g. stress level, and the independent variable process model, and in that case use the full scale of six different levels which was employed to capture this data (see also Table 2). For these cases, we also check the non-parametric Spearman correlation coefficient.

## Data Set

Table 2 gives a first overview of the data set and some project characteristics collected[1]. We will now continue to give some results on differences between the groups according to process models in general, while in the next section will be focusing on the human resource related problems.

The two groups did not show any major differences with regard to team size or duration (Wilcoxon rank sum test, $p > 0.1$), although the estimated effort was significantly higher for the rigid-type projects (Wilcoxon rank sum test, $p < 0.1$). While this is in accordance with the literature, which sees this type of process models in

*Table 2. Data set of projects with characteristics*

| Group | Project | Process model | Team size | Duration (months) | Estimated effort (person-months) | Deviation in effort from estimate (percent) | Deviation in duration from plan (percent) | Customer satisfaction (7 point scale from +3 very high to -3 very low) |
|---|---|---|---|---|---|---|---|---|
| Agile | 1 | 2 | 15 | 14 | 24.0 | 100 | 8 | 2 |
| | 2 | 1 | 12 | 12 | 23.5 | 0 | 0 | 3 |
| | 3 | 2 | 8 | 8 | 6.0 | 12 | 0 | 3 |
| | 4 | 3 | 4 | 3 | 5.0 | 10 | 5 | 2 |
| | 5 | 3 | 6 | 11 | 27.5 | 20 | 0 | 2 |
| | 6 | 3 | 6 | 10 | 18.0 | -10 | -10 | 1 |
| | 7 | 1 | 5 | 9 | 12.5 | 0 | 0 | 3 |
| | 8 | 2 | 8 | 5 | 4.8 | 0 | 20 | 2 |
| | 9 | 3 | 7 | 12 | 10.0 | 24 | 34 | 2 |
| | 10 | 3 | 10 | 7 | 22.5 | 10 | 14 | 2 |
| | **Mean** | **2.4** | **8.2** | **9.1** | **15.4** | **16.6** | **7.1** | **2.2** |
| Rigid | 11 | 5 | 10 | 6 | 40.0 | 0 | 0 | 3 |
| | 12 | 6 | 9 | 8 | 15.0 | 15 | 0 | 0 |
| | 13 | 4 | 5 | 5 | 25.0 | 12 | 20 | 2 |
| | 14 | 5 | 6 | 15 | 30.0 | 30 | 0 | 1 |
| | 15 | 5 | 8 | 8 | 15.0 | 100 | 30 | 2 |
| | **Mean** | **5.0** | **7.6** | **8.4** | **25.0** | **31.4** | **12.5** | **1.6** |

use for larger projects, and also needing higher effort due to their nature, the results for team size are more surprising. Theory would prescribe agile process models to be more suitable for smaller groups, which is still true in this dataset with a mean of about 8 persons, while more rigid models are employed in larger projects. This size is about equal to what Reifer (2002) found. This difference is not to be seen here. Regarding correctly estimating the effort, the result that agile projects seem to fare better is also surprising, as effort estimation in general will be more easy to perform in a rigid environment of fixed requirements, although the difference is statistically not significant (Wilcoxon rank sum test, $p > 0.1$). It is interesting, that those projects of rigid type which were finished on time seem to have been doing so at the expense of customer satisfaction: Exactly the two projects with this type of process model which finished on time scored lowest in this regard. The average customer satisfaction resulting from the agile models was at 2.2 (median 2) and the rigid models at 1.6 (median 2), which overall did not constitute a significant difference when comparing these two groups (Wilcoxon rank sum test, $p > 0.1$). Using the full scale of process models, the model adopted actually becomes a significant factor in a linear model using customer satisfaction as dependent variable ($p < 0.05$), with the model achieving an adjusted R^2 of 0.31. The factor stays in the model if other control variables like team size or total effort are added. Agile process models place a high value on customer satisfaction as well as early and continous inclusion of user representatives, which seems to affect the outcome as underlined by this empirical result. It was striking that with regard to the hierarchical structure of the project team all persons indicated a flat hierarchy, although according to theory, rigid models in general favor a hierarchical structure. Outside the team, very hierarchical forms were present in the organizations.

Regarding the age of team members in the projects of the two groups, again there is a small, but statistically not significant difference. In general, both types of projects have young and old employees working on them, which leads to results around the mean of 0 with the endpoint of the scales 3 very old, which was anchored at 50 years due to the industry conditions, and -3 very young at about 20-25 years of age. For rigid-type models, the mean value was 0.2 (median 0) and for agile-type models -0.8 (median -0.5), showing a small but insignificant difference (Wilcoxon rank sum test, $p > 0.1$). There could be several reasons for that, for example training received during education might give younger people more awareness and insight about agile methodologies, which leads to both them choosing such projects, and also being chosen for them. As an additional factor, agility and flexibility are more often associated with younger people, so this again might lead to both a selection and self-selection bias in project placement.

## ANALYSIS

The main point of investigation for differences between the two groups of projects are diverse human resource related issues.

### Acceptance and Responsibilities

As a major influence on this, and also in a way an outcome of exposure to the model in practice, is the acceptability of the model to the involved persons. In general, both types of model, agile and rigid, have a quite high acceptance rate, although agile models score somewhat higher: For rigid models, the acceptance rate was consistently 2 (on the -3 to +3 scale), for agile models straight 3 70% of the time, 20% scored 2, and one outlier of -1 existed (mean 2.4, median 3). Therefore, overall, agile-type process models seem to be more acceptable for team members (Wilcoxon rank sum test, $p < 0.05$), which would be in line with theory, as these models put more emphasis on team members, their

responsibilities, and several other aspects which should help to increase the morale. Responsibilities and wishes for changes in this area (i.e. mostly the wish for increased responsibility) were also inquired as separate topics. In theory, agile models propose concepts like shared responsibilities, collective code ownership and other related ideas, while rigid models advocate a more well-defined, small set of responsibilities for each team member. Interestingly, the results, although showing only a minor and not significant difference between both groups (Wilcoxon rank sum test, $p > 0.1$), display the opposite trend: In the mean, for the agile projects, -0.2 resulted (with -3 denoting no wishes for increased responsibilities, median 0.5), with -1 (median -2) for rigid models. Overall, as the numbers in absolute terms indicate, this is not a major problem in all projects, but the direction still seems interesting. One explanation could be that collective code ownership and similar concepts in agile development are either not fully implemented, or are too abstract for some people, who would prefer more clearly defined responsibilities. There is one additional explanation for this, which is derived from the personality of people chosen for agile projects. Possibly, developers with a higher propensity towards taking on responsibilities are selected, or self-select, to participate in this type of projects, but are still not satisfied with the level found there. In rigid-type projects, they would be even more displeased.

## Sharing and Caring

Disclosure and transfer of information between different project participants seems to be working more efficiently in agile-type models. Although the values with -0.9 (median -1.5) and -1.6 (median -2) are not very far apart (Wilcoxon rank sum test, $p > 0.1$), the trend might indicate that agile models have a small advantages in achieving communication and cooperation within the project team. Also Pikkarainen et al. (2008) have shown that agile practices improve both informal and formal communication. Knowledge sharing within a project organisation is one of the most influential factor on success, and was found to be correlated to organisational culture (Ajmal, in press). While we did not directly take organisational culture into consideration, the development model and underlying values followed can be said to form an important part of this, and thus the results underline this. In a connected topic, it was also explored whether people cared about their colleagues beyond a work context, but no difference between the two groups were to be found regarding this (Wilcoxon rank sum test, $p > 0.1$), although theory again sometimes cites this as an advantage resulting from agile process models.

## Human Resource Related Problems

The main block in our empirical research dealt with a long list of human resource related problems which can occur in software development projects, or mostly, in all projects. We will now take a look at whether some or all of these problems tend to turn up in one of the groups of projects according to adopted type of process model, and how well and easily they can be dealt it in that case. The first problem concerns team members' illnesses. In only one case, a major disorder turned up during the course of the project, all other illnesses were rather short-term. Neither in numbers nor in the counter-strategies a major difference between both groups of projects showed in the data (Wilcoxon rank sum test, $p > 0.1$). Strategies employed included redistribution of work, postponement of milestones or a reduced number of iterations (only for agile-oriented projects). While there was no major difference in these points, the impacts on the projects actually differed: For agile models, the impact on duration of the project was more severe (-1.1, median -2) than for rigid models (-2.4, median -3), where almost no delay resulted. Despite the difference, it is not statistically significant (Wilcoxon rank sum test, $p > 0.1$). While in theory this could be blamed on larger number of

manpower in rigid-type projects, this cannot be accepted as a reason in this dataset, as the mean size of the projects is quite similar between both groups. It seems as if rigid-type projects might be better able to deal with (temporary) loss of team members than agile models, although due to concepts like pair programming or collective code ownership, this should actually be the other way. The issue of absenteeism is negligible in general, because in the projects there were only sporadic absenteeisms and no striking differences between the two models. The same is true for all instances of refusal of work, like leaving work undone uncommented, fake work, threats of illness or similar: Those occurred in a small number of projects only, and there is no difference between the groups of projects considered (Wilcoxon rank sum test, $p > 0.1$). Also inter-personal problems within the team, like boycott of cooperation with certain colleagues, occurred only seldomly and without difference in both types of projects (Wilcoxon rank sum test, $p > 0.1$). No major problems with management or colleagues turned up in any project. Also other effects like deliberate damaging of resources, the wasteful use of company resources, the use of equipment for personal use and cheating in various events such as payroll or hours statements, were seldomly reported, and incidents do not show any connection to the process model (Wilcoxon rank sum test, $p > 0.1$). It can be noted at this point that using company equipment for personal tasks is not seen as a major problem by management. It is widely accepted that this happens to some degree, and is not judged to be problematic. Of course, it needs to be noted that some of these effects are quite severe, and there might be some inhibition to report those within a questionnaire or interview.

## Quality and Errors

We also checked for differences in several quality-related aspects due to human resource problems. For instance, the quantity of errors as made by the team members is reported as not very low for both models. This is not seen as something very negative, with some interviewees mentioning that errors are a natural consequence of getting work done. Also, some errors are quite easily fixed in software development. Between the two types of process models, a small difference showed up, with advantages for agile methods: While the value for rigid models is slightly above 0 with 0.6 (median 1), thus leaning towards more errors, the value for the agile group is slightly negative with -0.5 (median 0), indicating a smaller probability of errors. In the OLS regression model, the adopted process model turns out to be a significant factor ($p < 0.05$, even if controlled for team size or total effort) as does the correlation coefficent (non-parametric Spearman, $p < 0.1$), underlining this relationship. Jiang et al. (2003) also have identified the organizational environment as a major factor in effectiveness. Of course this could also be caused by team member selection, as agile approaches are often deemed to need better people, which are therefore selected and thus have a smaller rate of errors. For the question of how often the results of the work do not meet quality requirements, there is no significant difference between the models, as is for forgetting important dates (Wilcoxon rank sum test, in both cases $p > 0.1$). A difference is found when the relationship between quality and quantity is investigated. The average value in relation to this issue has shown that rigid models have a worse outcome than agile models (mean of -0.2 with median 0 versus -1.2 and -1, respectively). A similar situation can be seen with regard to delivery on time, again showing advantages for the group of agile approaches, although the differences in both cases are not significant (Wilcoxon rank sum test, $p > 0.1$).

## Stress and Satisfaction

A major factor for satisfaction within a project or work environment in general is the workload. Both too high a workload, which results in overtime

and stress situations, and too low a workload have been found to have detrimental effects. Complaints about both were relatively seldom in the projects in our sample, within both groups, the mean was -1.4 (median -1 respectively -2). With regard to stress and occurrences of overtime (as opposed to complaints), there is actually a difference between projects employing different process models: Stress is actually quite common, and more so in rigid-type projects. They have a mean of 1.6 (median 2), while agile projects only reach 0.8 (median 1). For overtime, the difference is even more pronounced, with mean and median 2.0 (rigid-type projects) against 0.5 with median 1 (agile-type projects). The second difference is statistically significant (Spearman correlation, $p < 0.1$, OLS regression, $p < 0.05$, even if controlled for team size). This underlines that some of the main ideas in agile models, including reasonable workload and "no overtime" rules are actually implemented, and counters the notion of Cherry and Robillard (2008) that more frequent informal collaborations might increase stress levels. Possibly resulting from this, both in motivation and performance, agile-type projects score slightly better than the other group of projects, but in both cases the differences are not statistically significant (Wilcoxon rank sum test, $p > 0.1$). An unusually sharp decrease in performance was, on the other hand, only found in agile-type projects on two counts. For one of those, a reason was given, which was frustration of an employee when he/she could not enforce his will. A more or less final stage for dissatisfied employees is the possibility to terminate the contract. This should therefore be taken as a serious sign of discomfort with the current work situation. In our dataset, such a threat nearly never was made by an employee, with the exception of one project from each of the two groups. These two projects were also the only ones where actual layoffs took place during the lifetime of the project. Therefore, no significant difference between the two types of process models can be seen in this regard (Wilcoxon rank sum test, $p > 0.1$).

Lastly, we used the answers to all human resource problem related question for arriving at an overall staff satisfaction measure. Due to lack of any weighting scheme to differentiate between questions (or any reliable way to arrive at such), all answers receive equal weight in the summation, using the mean result for each question. On this level, the results are actually quite good for both models, hinting at high overall satisfaction levels. For the complete list of 29 questions included, rigid-type projects had a sum of means of -43, agile-type of -44.3 (out of a possible maximum of -87). Again, a difference between both groups of projects can be seen, with an advantage for agile process models, but it is quite small.

## CONCLUSION AND FUTURE RESEARCH

### Results and Discussion

In this paper we have dealt with human resource related problems and issues in software development projects, and their relation to the adapted process model. We have used a classification to distinguish between two broad groups of models: More modern, agile approaches, and more traditional, rigid-oriented types. The empirical study has shown that indeed companies do not adopt models directly as proposed in the literature, but adapt and change for each project. Therefore, we are facing a continuum of process models according to this dimension of agility and rigidity, which makes it more difficult to arrive at statistically significant results. Using interviews with members of 15 different software developing companies in Austria, we have arrived at a data set of recent projects. Regarding project characteristics, it was interesting to see that agile-oriented models are not necessarily limited to small projects, but that both groups showed nearly identical distributions

for team size and duration, contrary to current mainstream assumptions (Boehm, 2002). Interestingly, rigid-type models tend to exhibit higher effort estimations, and lower correctness in these estimations. Also customer satisfaction is slightly lower, which would coincide with the literature which highlights the importance placed by agile models on customer satisfaction and inclusion, and the age of project participants is slightly higher. This last finding can be explained by different educational backgrounds, or psychological traits together with self-selection.

With regard to human resource issues, the differences are not major overall, but there are some noticeable exceptions: In general, satisfaction and acceptance are higher, confirming the findings of Syed-Abdullah et al. (2006), at lower stress and overtime levels for agile-type project participants. Agile models mostly have a no overtime rule, so this might seem intuitive, but Cherry and Robillard (2008) have hypothesized that the more frequent informal interactions could increase stress levels. Their notion was not supported by our findings.

On the other hand, interestingly and contrary to theory, people wish for more responsibility. There are some explanations for this last finding, one being that collective code ownership and similar concepts are either not fully implemented or too abstract for some people, and another being derived from the personality of people with a higher propensity towards taking on responsibilities being selected or self-selected for this type of projects, who are still not satisfied with the level found there. This might be likened to what Cohn and Ford (2003) experienced as over-enthusiasm. Agile-type projects also seem to enjoy some advantages in information sharing and communication, and in some quality aspects. Both of these ideas have already been proposed in literature (Jiang et al., 2003; Cherry & Robillard, 2008; Pikkarainen et al., 2008). Reifer (2002), for example, found similar quality levels, but not superior ones. On the other hand, rigid-type projects show considerable higher abilities to cope with absence of personnel.

Overall, therefore, both types of models have been found to be adopted in practice, but not to their full extent, but in slightly adapted, changed, and mostly moderated form. Companies seem to shy away from extremes in both cases, which seems to be a sensible choice given the results presented here. Both models have advantages and disadvantages, but neither are very pronounced. As expected, agile-oriented approaches achieve good results for some dimensions like stress or overtime, but show problems in other areas like responsibility or coping with employee absence. A good fit for a company and team would therefore enable to reap benefits from both ideas, and to withhold any major problems on the other hand. The managerial applicability of this research lies exactly in this trade-off which is a conscious decision of team leadership. Any decision on which process model to adopt needs to take these human resource related effects into consideration, in addition to other factors like cost or schedule.

## Limitations and Further Research

Naturally, any empirical study always has some limitations, which also give impetus for further research. Of course, a larger data set, including a diverse population of projects would be interesting to explore. Still, we have tried to ensure the external validity by gathering data from a widely diverse set of small and large, old and young companies. Pre-testing of the questions and scales was employed to counter any problems with construct validity. As many questions pertain to problems easily measured in other ways, it is not assumed to be a major concern. For the overall satisfaction measure as a construct, validity would need to be ensured by a large-scale investigation of team members. Sadly, this was not possible. One main threat is internal validity, as it is possible that some of the results are influenced by a selection bias in staffing projects: Theory holds that agile approaches need more capable and highly motivated team members, so exactly those might possibly be

selected. This would lead to some of the results seen here, like better quality or less satisfaction with responsibilities, which in this case would not be due to process model, but personnel attributes and selection. The same effect could result out of self-selection, i.e. a certain type of person selecting projects or companies with a more agile-oriented approach. It would therefore certainly be worthwhile to include employee qualifications and abilities, or, even better, to study the same people in differently organized environments or projects. Naturally, this is a long term task, which nevertheless might yield very interesting and relevant results. It has also been found to be difficult to assess some human resource related problems like theft or cheating in time reporting, because these severe violations might carry penalties even much later on. Achieving a high level of trust would be paramount in getting truthful results for these areas. With regard to quality and customer satisfaction, it would also be interesting to include interviews with those different stakeholders and gather more, and more objective, information on this. With this paper, we hope to have given a first glimpse at the topic of human resource related topics in connection with process models employed, and we hope research on this is going to continue. After all, software development constitutes a major human effort, and addressing the participants' needs, concerns and problems would greatly enhance our knowledge about and ability to manage these efforts successfully.

## REFERENCES

Ajmal, M. (in press). Role of organizational culture for knowledge sharing in project environments. *International Journal of Project Organisation and Management*.

Beck, K. (1999). *Extreme programming explained: Embrace change*. Reading, MA: Addison-Wesley.

Boehm, B. (1988). A spiral model of software development and enhancement. *IEEE Computer*, *21*(5), 61–72.

Boehm, B. (2002). Get ready for agile methods, with care. *IEEE Computer*, *35*(1), 64–69.

Boehm, B., & Turner, R. (2003). *Balancing agility and discipline: A guide for the perplexed*. Reading, MA: Addison-Wesley.

Brooks, F. P. Jr. (1999). *The mythical man – month: Essays on software engineering*. Reading, MA: Addison-Wesley.

Cherry, S., & Robillard, P. N. (2008). The social side of software engineering—a real ad hoc collaboration network. *International Journal of Human-Computer Studies*, *66*(7), 495–505. doi:10.1016/j.ijhcs.2008.01.002

Cockburn, A., & Highsmith, J. (2001). Agile software development: The people factor. *IEEE Computer*, *34*(11), 131–133.

Cohn, M., & Ford, D. (2003). Introducing an agile process to an organization. *IEEE Software*, *36*(6), 74–78.

DeMarco, T. (2002). *Slack: Getting past burnout, busywork, and the myth of total efficiency*. New York, NY: Broadway Books.

DeMarco, T., & Lister, T. (1999). *Peopleware: Productive projects and teams* (2nd ed.). New York, NY: Dorset House.

Demirkan, H., & Nichols, J. (2008). IT services project management: Lessons learned from a case study in implementation. *International Journal of Project Organisation and Management*, *1*(2), 204–220. doi:10.1504/IJPOM.2008.022192

García-Crespo, A., Colomo-Palacios, R., Gomez-Berbís, J. M., & Ruano-Mayoral, M. (in press). A project management methodology for commercial software reengineering. *International Journal of Project Organisation and Management*.