Agile practices to accelerate the delivery of software: a quantitative study with software professionals

Adriano Olímpio Tonelli  
Universidade Federal de Lavras  
tonelli@dcc.ufla.br

Paulo Henrique S. Bermejo  
Universidade Federal de Lavras  
Bentley University  
bermejo@dcc.ufla.br

Mariana Azevedo Santos  
Mitah Technologies  
mariana@bsi.ufla.br

André Luiz Zambalde  
Universidade Federal de Lavras  
zamba@dcc.ufla.br

Marcelo Silva de Oliveira  
Universidade Federal de Lavras  
marcelo.oliveira@dex.ufla.br

Luiz Marcelo Antonialli  
Universidade Federal de Lavras  
lantonialli@uol.com.br

Abstract

This study aims to investigate the perceptions of software professionals in relation with impact of agile practices to accelerate the delivery of software products. Quantitative data obtained from a sample of 109 professionals were collected and analyzed by Pearson correlation and factor analysis. The results of factor analysis showed that the perception of professionals in relation to agile practices can be grouped into seven factors. From the correlation analysis, it was observed that customer satisfaction is the main variable influencing the perceptions of professionals regarding the use of agile practices to meet the deadlines on software projects.

1. Introduction

The question of how best to organize software development to ensure rapid product delivery has been debated for decades [1]. These debates reflect the complexity of software development and the consequent difficulty that companies face in improving their ability to swiftly deliver solutions to customers. According to research by the Standish Group [2], about 63% of software development projects had deficiencies in meeting deadlines, budget, or quality benchmarks. In the face of these continuing challenges to productivity, an increasing number of software companies have been seeking alternative models that will improve the efficiency of development and increase the speed of delivery [3].

Elaborated in the 2000s following the declaration of the Agile Manifesto [4], agile methodologies are based on principles fundamentally different from traditional approaches to software development. Specifically, these principles emphasize (1) individuals and interactions over processes and tools, (2) working software over comprehensive documentation, (3) customer collaboration over contract negotiation, and (4) responding to change over following a plan [4]. While traditional software development models emphasize documentation and sequential processes based on requirements elicitation, architecture development, coding, and testing, agile methodologies emphasize iterative development and conceptual simplicity to allow developers to create products quickly and with heavy interaction with customers [1, 5, 6].

Given the nimbleness that its principles allow, agile methodologies have garnered increasing interest from companies seeking to improve product delivery and quality and to identify solutions to problems associated with constant changes in requirements, fixed-date releases, lack of customer feedback, and fixed budgets [1, 7, 8].

With the growing adoption of agile methodologies in the market, there have been an increasing research interest in agile software development. Numerous studies have focused on factors influencing the successful adoption of agile methodologies, including shared experiences among team members [6, 9], top-level management support [10], team autonomy and diversity [11], self-discipline in individual team members [10], team member communication [6, 12], the utility of documentation [6]. Other studies have analyzed social factors related to agile software development, such as organizational culture [6, 12] and team characteristics [12–14]. Still others have studied practitioners’ perceptions of specific agile practices, predominantly XP practices [7, 15–17].

However, although increasingly examined in the literature, the concept of agile software development is still insufficiently understood, especially regarding the outcomes of agile practices [6, 9, 18] and how firms can control and influence agile development [11]. Most studies employ experiential, anecdotal, or case study
design [1, 9], with a few exceptions (e.g., Misra, Kumar, and Kumar [9]; and Lee and Xia [11]), quantitative research based on large-scale surveys are scant. Furthermore, most of the available studies focus on a single agile methodology, generally on XP or Scrum [1]. The literature lacks studies that consider a broader spectrum of agile methodologies. This gap is critical because agile practices are not always adopted discretely: based on the needs of a specific organization, a variety of practices from different agile methodologies may be mixed the practices to create hybrid approaches [9, 18, 19].

In light of this lacuna in the existing literature, the present study seeks to examine two questions; (1) which agile practices contribute to improving the ability of organizations to accelerate the software product delivery? And (2) what influences the professional’s perception of the contribution of agile practices to accelerated the software product delivery? This study investigates these questions through quantitative, survey-based research. The paper provides empirical evidence on how to combine and apply different agile practices. In doing so, this work seeks to expand the knowledge about the application of different agile practices from different methodologies to enhance the agility in the delivery of software products.

2. Methodology

2.1. Sampling and data collection

This quantitative study conducted a survey of 109 professionals engaged in different aspects of the software development process. Two types of sampling were used to select the participants: expert sampling and geometrical, or snowball, sampling [23]. The use of these sampling methods is justified because the study required the participation of (i) specialists or individuals with an expertise in software development and (ii) hard-to-reach individuals with characteristics compatible with the research objectives. Such participants are usually best identified by individuals in personal or professional networks and unknown by the researcher.

A structured questionnaire was made available on the web tool Survey Monkey, and data were collected between January and May 2011. The questionnaire contained 59 questions: 2 questions related to demographics and 57 questions related to an equal number of agile practices. For the latter, we used a 6-point Likert scale ranging from “very low” to “very high” to assess of participants’ perceptions of the impact of the practices on software product delivery. Of the 57 practices, the participants were asked to evaluate 3 or more practices that they were using at the time or had used in the past. This precaution was taken to prevent flawed and inconsistent responses.

The agile practices included in the questionnaire were derived from the practices identified by Santos et al. [20] as most relevant in the marketplace. Table 1 lists the practices included in this study.

2.2. Data analysis

Factor analysis using Varimax orthogonal rotation [21] was conducted to identify the structure underlying sets of practices that impact the accelerated delivery of software products. Only practices with loads above 0.55 were included in the factors. The Kaiser–Meyer–Olkin (KMO) latent root and Bartlett’s test of sphericity [21, 22] were used to test validity. Included practices were also restricted to those with more than 35 responses to eliminate practices insignificant to this study.

To determine the factors that influence professionals’ perceptions of the impact of agile practices on the speed of software product delivery, we used Pearson correlation analysis. According to Malhotra and Birks [22], Pearson correlation is widely used to summarize the strength of association between two metric variables. Data relating to the perceived impact of agile practices on product delivery speed were correlated with four variables: stage of adoption of agile practices, time of adoption, customer satisfaction for products delivered, and team satisfaction with the agile practices. The correlations were analyzed considering significance levels of 0.05 and 0.01.

2.3. Description of variables

Considering the perception that practitioners have about agile practices, this research was based on the following groups of variables:

Perception of agile practices – This set of variables assessed, on a 6-point scale, the participants’ perception of the agile practice’s positive contribution to accelerated software delivery. 51 practices were identified (Table 1) from a review of the main Agile Methodologies available in the market.

Satisfaction – this group includes variables related to customer satisfaction and team satisfaction. Customer satisfaction is the professional’s perception of customer satisfaction with the speed of product delivery. This was measured on a 6-point scale (very low, low, regular, fair, high, very high). Team satisfaction measures the level of team members’ satisfaction with the speed in delivery. This variable
was also measured on a 6-point scale (very low, low, regular, fair, high, very high).

Adoption of agile practices – This group includes variables related to the time since adoption and the stage of implementation of agile practices in the organization. Time since adoption considered the time (in months) that the professional’s organization has employed agile practices. Stage of implementation at the professional’s present company was measured using the Capability Maturity Model (CMMI) [23] based on a 4-point scale: (1) initial, (2) partially implemented, (3) implemented without measurements, and (4) implemented with measurement and continuous improvement.

## Table 1 Missing data analysis

<table>
<thead>
<tr>
<th>Practice</th>
<th>N</th>
<th>Missing</th>
<th>Practice</th>
<th>N</th>
<th>Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Count</td>
<td></td>
<td></td>
<td>Count</td>
</tr>
<tr>
<td>Small teams</td>
<td>90</td>
<td>19</td>
<td>Code ownership</td>
<td>32</td>
<td>77</td>
</tr>
<tr>
<td>Multifunctional teams</td>
<td>59</td>
<td>50</td>
<td>Build 10 minutes</td>
<td>17</td>
<td>92</td>
</tr>
<tr>
<td>Solo programmer</td>
<td>43</td>
<td>66</td>
<td>Class ownership</td>
<td>13</td>
<td>96</td>
</tr>
<tr>
<td>Product owner</td>
<td>53</td>
<td>56</td>
<td>Screening bugs</td>
<td>41</td>
<td>68</td>
</tr>
<tr>
<td>Scrum master</td>
<td>46</td>
<td>63</td>
<td>Configuration management</td>
<td>24</td>
<td>85</td>
</tr>
<tr>
<td>Leader developer</td>
<td>43</td>
<td>66</td>
<td>Test last develop</td>
<td>23</td>
<td>86</td>
</tr>
<tr>
<td>Features teams</td>
<td>28</td>
<td>81</td>
<td>Test first develop</td>
<td>21</td>
<td>88</td>
</tr>
<tr>
<td>Parallel teams</td>
<td>33</td>
<td>76</td>
<td>Functional test</td>
<td>46</td>
<td>63</td>
</tr>
<tr>
<td>Small teams in large staffs</td>
<td>30</td>
<td>79</td>
<td>Test cases</td>
<td>32</td>
<td>77</td>
</tr>
<tr>
<td>Daily meeting</td>
<td>53</td>
<td>56</td>
<td>Unit test</td>
<td>41</td>
<td>68</td>
</tr>
<tr>
<td>Stand-up meeting</td>
<td>63</td>
<td>46</td>
<td>Test automation</td>
<td>33</td>
<td>76</td>
</tr>
<tr>
<td>Iteration planning meeting</td>
<td>61</td>
<td>48</td>
<td>Methodology refining</td>
<td>10</td>
<td>99</td>
</tr>
<tr>
<td>Retrospective</td>
<td>50</td>
<td>59</td>
<td>Continuous integration</td>
<td>36</td>
<td>73</td>
</tr>
<tr>
<td>Product backlog</td>
<td>50</td>
<td>59</td>
<td>Fixed iterations</td>
<td>49</td>
<td>60</td>
</tr>
<tr>
<td>Customer onsite</td>
<td>59</td>
<td>50</td>
<td>Burn down chart</td>
<td>34</td>
<td>75</td>
</tr>
<tr>
<td>Use cases</td>
<td>47</td>
<td>62</td>
<td>Kanban</td>
<td>52</td>
<td>57</td>
</tr>
<tr>
<td>Scenarios</td>
<td>31</td>
<td>78</td>
<td>Planning games</td>
<td>32</td>
<td>77</td>
</tr>
<tr>
<td>Vision document</td>
<td>40</td>
<td>69</td>
<td>Velocity</td>
<td>26</td>
<td>83</td>
</tr>
<tr>
<td>Evocative document</td>
<td>22</td>
<td>87</td>
<td>Features estimative</td>
<td>33</td>
<td>76</td>
</tr>
<tr>
<td>User story</td>
<td>46</td>
<td>63</td>
<td>Mission estimative</td>
<td>12</td>
<td>97</td>
</tr>
<tr>
<td>Architectural spike</td>
<td>23</td>
<td>86</td>
<td>Progress report</td>
<td>22</td>
<td>87</td>
</tr>
<tr>
<td>Domain model</td>
<td>25</td>
<td>84</td>
<td>Checklist</td>
<td>25</td>
<td>84</td>
</tr>
<tr>
<td>Business objective</td>
<td>29</td>
<td>80</td>
<td>Parking lot</td>
<td>9</td>
<td>100</td>
</tr>
<tr>
<td>UML diagrams</td>
<td>42</td>
<td>67</td>
<td>Potentially shippable</td>
<td>43</td>
<td>66</td>
</tr>
<tr>
<td>Refactoring</td>
<td>44</td>
<td>65</td>
<td>Agile coach</td>
<td>21</td>
<td>88</td>
</tr>
<tr>
<td>Pair programming</td>
<td>37</td>
<td>72</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1 Missing data analysis
3. Results

3.1. Sample profile

Analysis of the survey data revealed that participants were predominantly project managers (27.5%), technical leaders (11.0%), and project leaders (10.1%). There was no incidence of customer participation, although this class of stakeholder was included in the questionnaire. Figure 1 illustrates the participants' occupations.

3.2. Validity and reliability

As a first step in data validation, variables with fewer participant evaluations were removed prior to factor analysis in order to obtain more representative samples. Only practices with 35 or more assessments were considered in factor analysis. Table 1 represents the frequencies and the missing data of participant assessments for each of the practices considered in the study. The practices considered for factor analysis, with frequencies greater than 35, are shown in bold.

Following the analysis of missing data and responses frequency of each practice, the method Kaiser–Meyer–Olkin (KMO) and Bartlett’s sphericity test were executed. As illustrated in Table 2, the KMO index was 0.784. According Hair et al. [21], this can be considered median and validates the factorial analysis for the set of research data. In relation to sphericity test, significance less than 0.001 indicate that the correlation matrix is not identity, and then, is suitable for the application of factorial analysis.

| Kaiser-Meyer-Olkin measure of sampling adequacy | .784 |
| Bartlett’s test of sphericity                  | -5.351 |
| df                                              | 325  |
| Sig.                                            | 0.000 |

Table 2 KMO method and Bartlett’s sphericity test result for the research
Using factor analysis, eight factors were extracted from the data, explaining 66.097% of the variance. According Hair et al. [21] identify this as an acceptable parameter for adaptation to the technique applied. Table 3 shows the practices classified into each of the eight factors extracted.

Factor 1 – Bug correction and tasks on sprint. The first factor includes practices related to partial deliveries and named releases. The practices of Kanban, screening errors, fixed iterations, and potentially shippable are significantly correlated and can be grouped in order to explain the professionals' perception of these practices' contribution to timely project delivery. This factor could be explained by the following: planning sprints with bug correction could contribute in a better planned release, ensuring the potentially shippable product's delivery thereby avoid unnecessary refactoring in the future.

Factor 2 – Development guide by test/Test-driven development with instant feedback. The second factor clusters test-driven development practices and instant feedback. We could explain this factor as: the team writes test cases and code and performs unit tests; when the code passes the test, they continuously integrated all codification, helping to reduce development efforts on sprint overall [24].

Factor 3 – Building agile teams. Practices grouped by this factor are multifunctional teams, product owner, leader developer, and customer on-site. The third factor represents the customer involvement with the team on development. In line with previous studies [6, 9, 18] the involvement of the customer could be of help to the team to focus on customer needs, diagnosing problematic situations, while the product owner writes customer-centric items, thereby making the team’s work more effective and productive.

Factor 4 – Requirements organization and evaluation. Practices grouped by this factor are user story, iteration planning meeting, and retrospective.

### Table 3 Extracted factors

<table>
<thead>
<tr>
<th>Practices</th>
<th>Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Fixed iterations</td>
<td>.644</td>
</tr>
<tr>
<td>Kanban</td>
<td>.682</td>
</tr>
<tr>
<td>Potentially shippable</td>
<td>.688</td>
</tr>
<tr>
<td>Screening bugs</td>
<td>.705</td>
</tr>
<tr>
<td>Development guide by test</td>
<td>.804</td>
</tr>
<tr>
<td>Unit tests</td>
<td>.730</td>
</tr>
<tr>
<td>Continuous integration</td>
<td>.649</td>
</tr>
<tr>
<td>Multifunctional teams</td>
<td>.817</td>
</tr>
<tr>
<td>Product owner</td>
<td>.562</td>
</tr>
<tr>
<td>Leader developer</td>
<td>.677</td>
</tr>
<tr>
<td>Customer on-site</td>
<td>.623</td>
</tr>
<tr>
<td>Iteration planning meeting</td>
<td>.699</td>
</tr>
<tr>
<td>Retrospective</td>
<td>.692</td>
</tr>
<tr>
<td>User story</td>
<td>.682</td>
</tr>
<tr>
<td>Use cases</td>
<td>.812</td>
</tr>
<tr>
<td>Vision document</td>
<td>.646</td>
</tr>
<tr>
<td>Scrum master</td>
<td>.732</td>
</tr>
<tr>
<td>UML diagrams</td>
<td>.587</td>
</tr>
<tr>
<td>Solo programmer</td>
<td>.814</td>
</tr>
<tr>
<td>Product backlog</td>
<td>.786</td>
</tr>
<tr>
<td>Functional tests</td>
<td>.582</td>
</tr>
</tbody>
</table>

### 3.3. Factor analysis

Factor 1 – Bug correction and tasks on sprint. The first factor includes practices related to partial deliveries and named releases. The practices of Kanban, screening errors, fixed iterations, and potentially shippable are significantly correlated and can be grouped in order to explain the professionals' perception of these practices' contribution to timely project delivery. This factor could be explained by the following: planning sprints with bug correction could contribute in a better planned release, ensuring the potentially shippable product’s delivery thereby avoid unnecessary refactoring in the future.
Using user stories to detail the customer’s requirements, in addition with a product backlog in the planning meeting, could improve the understanding of the task during the sprint. So, the combination of user story and product backlog can contribute to reduce team’s inefficiency, and, consequently, to the risk of missed deadlines. Also, this could avoid the some negative experiences on retrospective meeting, for example, bad estimates based on bad information.

Factor 5 – Documentation. Practices addressed in this factor were use cases and vision document. The fifth factor is related to a project’s documentary elaboration. The vision document, in negotiation phase, elucidates to the customer how the developed system and final product works. The vision document could be helpful in the domain analysis, once this document can feasibly implement the use cases practice in its structure.

Factor 6 – Design using diagrams. This factor encompasses practices related to the leadership of the scrum master and to UML diagrams. The factor indicates that developers participating in this study may still use UML in their agile projects. This could be helpful to scrum masters and team members for communication in a standardized manner, without compromising the project’s scheduled delivery through a misunderstanding, especially if the teams are technologically non-homogenized.

Factor 7 – Solo developer. The seventh factor indicates that research participants perceived that solo programming can contribute to productivity and timely delivery.

Factor 8 – Functional tests on backlog items. Practices contemplated in this factor are functional tests and product backlog (priorities list). The eighth factor includes test cases created to measure the functional quality of the backlog items. Because the tests are estimated and made by tasks, deadlines are not compromised, they can still help to identify specific failures of user stories.

The extracted factors describe combinations of agile practices that, in the perception of practitioners, are correlated with an impact on the timely delivery of software projects. Projects are based on releases to the customer, with potentially shippable product delivery (Factor 1). In each release, teams write tests and code them, providing continuous integration to gradually form a complete software product to be delivered (Factor 2). These tests can be conducted from a priorities list, known as a product backlog, whose items are tested using functional tests (Factor 8).

In the project, teams are multifunctional and rely upon the active and continuous participation of the customer (Factor 3). In this way, requirements are developed based on customer’s narratives, known as user story, on planning meetings.

The project documentation is lean and easy to understand, emphasizing the characteristics of focus on development highlighted by agile methodologies (Factor 5). Specifications used in projects are based on UML resources, which helps communication between the scrum master and team members (Factor 6). Finally, the development is driven from the solo developer approach, in which a team member is responsible for implementing user story and continuously integrating with the team’s work (Factor 7). Subsequently, the implementation of these user stories is reviewed and all facts (negative and positive) experienced in developing are identified at a retrospective meeting, to be improved in the project (Factor 4).

3.4. Correlation analysis

From the correlation analysis of the variables having perceived value to agile practices with the four variables, team satisfaction, customer satisfaction, time since adoption, and stage of implementation, we found that in most cases, the variable with perceived relation to agile practices is correlated with at least with one of the variables related to satisfaction, adoption time and stage of adoption. Exceptions were: product owner, features teams, parallel teams, small teams in large staff, daily meetings, product backlog, use cases, scenarios, evocative document, diagrams UML, refactoring, 10 minutes builds, and configuration management.

Table 5 illustrates the correlation matrix.
<table>
<thead>
<tr>
<th>Practice</th>
<th>Satisfaction</th>
<th></th>
<th>Adoption time</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Customer satisfaction</td>
<td></td>
<td>Adoption time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Team satisfaction</td>
<td></td>
<td>Adoption stage</td>
<td></td>
</tr>
<tr>
<td>Small teams</td>
<td>.247*</td>
<td>.239*</td>
<td>.358**</td>
<td>.144</td>
</tr>
<tr>
<td>Multifunctional teams</td>
<td>.354**</td>
<td>.395**</td>
<td>.247</td>
<td>.257*</td>
</tr>
<tr>
<td>Solo programmer</td>
<td>.143</td>
<td>.120</td>
<td>.327*</td>
<td>.114</td>
</tr>
<tr>
<td>Product owner</td>
<td>.269</td>
<td>.105</td>
<td>.115</td>
<td>.226</td>
</tr>
<tr>
<td>Scrum master</td>
<td>.302*</td>
<td>.128</td>
<td>−.047</td>
<td>−.085</td>
</tr>
<tr>
<td>Leader developer</td>
<td>.273</td>
<td>.228</td>
<td>.157</td>
<td>.305*</td>
</tr>
<tr>
<td>Features teams</td>
<td>−.083</td>
<td>.014</td>
<td>−.101</td>
<td>−.158</td>
</tr>
<tr>
<td>Parallel teams</td>
<td>−.027</td>
<td>.252</td>
<td>−.050</td>
<td>.031</td>
</tr>
<tr>
<td>Small teams</td>
<td>.330</td>
<td>.130</td>
<td>.267</td>
<td>.290</td>
</tr>
<tr>
<td>Daily meeting</td>
<td>.242</td>
<td>.195</td>
<td>−.139</td>
<td>.040</td>
</tr>
<tr>
<td>Stand–up meeting</td>
<td>.281*</td>
<td>.250*</td>
<td>.137</td>
<td>.253*</td>
</tr>
<tr>
<td>Iteration planning meeting</td>
<td>.289*</td>
<td>.027</td>
<td>.186</td>
<td>.160</td>
</tr>
<tr>
<td>Retrospective</td>
<td>.417**</td>
<td>.099</td>
<td>.043</td>
<td>.165</td>
</tr>
<tr>
<td>Product backlog</td>
<td>.162</td>
<td>.141</td>
<td>.081</td>
<td>−.020</td>
</tr>
<tr>
<td>Customer onsite</td>
<td>.382**</td>
<td>.260*</td>
<td>−.073</td>
<td>.102</td>
</tr>
<tr>
<td>Use cases</td>
<td>.183</td>
<td>.078</td>
<td>−.031</td>
<td>.159</td>
</tr>
<tr>
<td>Scenarios</td>
<td>.270</td>
<td>.08</td>
<td>−.052</td>
<td>.156</td>
</tr>
<tr>
<td>Vision document</td>
<td>.384*</td>
<td>.099</td>
<td>−.066</td>
<td>.353*</td>
</tr>
<tr>
<td>Evocative document</td>
<td>.290</td>
<td>.085</td>
<td>−.050</td>
<td>.239</td>
</tr>
<tr>
<td>User story</td>
<td>.493**</td>
<td>.163</td>
<td>−.096</td>
<td>.195</td>
</tr>
<tr>
<td>Architectural spike</td>
<td>.405</td>
<td>.160</td>
<td>.470*</td>
<td>.458*</td>
</tr>
<tr>
<td>Domain model</td>
<td>.553**</td>
<td>.397*</td>
<td>.358</td>
<td>.205</td>
</tr>
<tr>
<td>Business objective</td>
<td>.541**</td>
<td>.452*</td>
<td>.112</td>
<td>.241</td>
</tr>
<tr>
<td>UML diagrams</td>
<td>.278</td>
<td>.222</td>
<td>−.046</td>
<td>−.129</td>
</tr>
<tr>
<td>Refactoring</td>
<td>.209</td>
<td>−.072</td>
<td>.131</td>
<td>.213</td>
</tr>
<tr>
<td>Pair programming</td>
<td>.444**</td>
<td>.210</td>
<td>.084</td>
<td>.199</td>
</tr>
<tr>
<td>Code ownership</td>
<td>.583**</td>
<td>.409*</td>
<td>.141</td>
<td>.360*</td>
</tr>
<tr>
<td>Build 10 minutes</td>
<td>.467</td>
<td>.204</td>
<td>.243</td>
<td>.425</td>
</tr>
<tr>
<td>Class ownership</td>
<td>.182</td>
<td>.113</td>
<td>.124</td>
<td>.048</td>
</tr>
<tr>
<td>Screening bugs</td>
<td>.313*</td>
<td>.250</td>
<td>.152</td>
<td>.200</td>
</tr>
<tr>
<td>Configuration management</td>
<td>.241</td>
<td>.228</td>
<td>.243</td>
<td>.060</td>
</tr>
</tbody>
</table>

* Significant at .05 (2-tailed)
** Significant at .01 (2-tailed)

Table 4. Correlation matrix between perceptions of agile practices and stage of adoption, adoption time, customer satisfaction and team satisfaction.

The data indicate that customer satisfaction is the variable that most influenced the perception of professionals about agile practices. Thus, to be accepted and have positive perceptions by professionals, it is essential that the use of agile practices has to be able to generate results that positively impact customer satisfaction.
Secondly, it is observed that team satisfaction influences most practices associated with the team’s organization and interactions between professionals. Small teams, multifunctional teams, stand-up meetings, customer on-site, and collective ownership of code are correlated with team satisfaction.

Although it has low correlation with the number of agile practices, the time since adoption had the highest correlation with the perception of professionals in small teams, solo programmer, and spike architecture. This can be evidence that, to generate positive results, such practices need time to be embraced by an organization. This finding is observed in particular with spike architecture, because the perception of this practice presents significant and positive correlations exclusively with time since adoption and implementation stage.

4. Conclusion

This study sought to investigate the perceptions of software professionals in relation with impact of agile practices to accelerate the delivery of software products.

Specifically, the study examined two fundamental questions about the application of agile practices: (1) which agile practices can contribute to improving the ability of organizations to accelerate the delivery of software products? (2) What influences the professional’s perception of the contribution of agile practices to accelerated software product delivery?

Considering the first question, the results of factor analysis indicated that an approach in which sets of agile practices are selected rather than a methodology embraced completely can be a path to the appropriate adoption of an agile approach to software development.

The eight factors produced from the analysis reveal the relationships of a set of manifest variables (perceptions of professionals regarding the use of practices included in the different agile methodologies) with a smaller number of latent variables (factors) that explain the professionals’ perceptions regarding the use of agile practices to agility on delivery of software products. From these factors, future studies will be able to develop frameworks that explain the relationship between the use of agile practices and the agility of software products releases.

Regarding the second question, the results of correlation analysis indicated that customer satisfaction is the main determinant of the professionals’ perception of the impact of agile practices on ability to meet project deadlines. Thus, these results suggest that customer satisfaction can be a determining factor for the continued acceptance and use of agile practices by professionals in software organizations. In addition, these results suggest that customer satisfaction is an important variable in the development of frameworks for explaining the relationship between the use of agile practices and the swiftness of software product releases.

From a managerial perspective, the main contribution of this study is to suggest that the eight factors extracted by factor analysis and relevant agile practices can together develop agility in the delivery of software products.

Furthermore, from correlation analysis, this study provides insight on the factors—customer satisfaction, team satisfaction, time since adoption, and implementation stage—that influence the professionals’ perceptions of the results of agile practices.

From a researcher’s perspective, the factors provide initial concepts from which to develop models that can be used to apply techniques, such as Structural Equation Modeling, to develop and test theoretical frameworks that correlate the use of agile practices and agility in software development.

5. Limitations

This paper presents two limitations that should be considered when interpreting and applying the results. First, the results are based upon a study with a limited sample, which prevents the generalization of the results obtained. Thus, it is important that these results will be interpreted as recommendations. Future studies which should perform further investigations to confirm or refute the software professionals’ perceptions revealed in this study.

In addition, this study was based on quantitative data and thus does not deepen our understanding of how and why agile practices contribute to agile software development. Given this limitation, future studies can investigate the use of such practices and their relationship to the ability to deliver projects into the deadline.

Acknowledgements

The authors thank FAPEMIG (The Minas Gerais State Research Foundation), CAPES (Coordination for the Improvement of Higher Level or Education-Personnel Process BEX 1588/11-7), and CNPq (National Counsel of Technological and Scientific Development) for their financial support.

6. References

[2]. Group, S., New Standish Group report shows more project failing and less successful projects. 2009.

[3]. Koscianski, A. and M.S. Soares, Qualidade de software: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2nd ed. 2007.


