The Impact of Organization, Project and Governance Variables on Software Quality and Project Success

Noura Abbas
University of Southampton
School of Electronics and Computer Science
Southampton, UK, SO17 1BJ
n.abbas@ecs.soton.ac.uk

Andrew M Gravell
University of Southampton
School of Electronics and Computer Science
Southampton, UK, SO17 1BJ
amg@ecs.soton.ac.uk

Gary B Wills
University of Southampton
School of Electronics and Computer Science
Southampton, UK, SO17 1BJ
gbw@ecs.soton.ac.uk

Abstract— **In this paper we present a statistically tested evidence about how quality and success rate are correlated with variables reflecting the organization and aspects of its project's governance, namely retrospectives and metrics. The results presented in this paper are based on the Agile Projects Governance Survey that collected 129 responses. This paper discuss the deep analysis of this survey, and the main findings suggest that when applying agile software development, the quality of software improves as the organization measures customer satisfaction more frequently, and as the impact of retrospective increases. Project success improves as quality, frequency of measuring customer satisfaction, organization experience in agile development, retrospective impact, the team participation in retrospective and the team contribution to retrospective, increases.**

Keywords: empirical research,agile projects governance, survey, correlations, retrospective, software quality, project success

I. INTRODUCTION

Probably the most noticeable change to software development methodology in the last 15 years has been the introduction of the word "agile". As any area matures, there is a need to understand its components and relations, as well as the need of empirical evidence about how well agile methods work in real life settings.

A survey that studies agile projects governance was conducted in September 2009 by the researcher. The descriptive results of this survey are under review for publishing. The main purpose of this survey was to investigate agile projects governance by collecting data about how people are monitoring the progress of projects developed using agile methods, practices and principles according to the agile manifesto (Highsmith et al. 2001). The survey was particularly interested in projects using agile retrospectives, reflection meetings and metrics.

This survey was part of a research that focused on studying the relationship between different organization, project and governance variables on software quality. In this paper we present the analysis of these different relationships based on the survey data.

II. BACKGROUND

A. *Information Technology Governance*

According to the Compact Advanced Learner's Dictionary (governance 2009), the word governance (noun) is the action or manner of governing. IT governance is emerging as an important area by academics and practitioners (Webb et al. 2006). IT Governance has evolved from corporate governance, strategic information systems, and strategic information systems planning. A systematic definition of IT governance was suggested by Webb and Pollard (Webb et al. 2006) as "the strategic alignment of IT with the business such that maximum business value is achieved through the development and maintenance of effective IT control and accountability, performance management and risk management". The previous definition was the result of analyzing twelve definitions found in the literature, which revealed five elements that constitute IT governance all were included in the proposed definition. Given different strategies and organizational structures, governance arrangements can vary from centralized approaches to decentralized ones or a hybrid approach that balance the first two (Weill et al. 2004).

According to a report by the IT Governance Institute, IT governance, like other governance subjects, is "the responsibility of the board and executives". In addition, the report mentioned that IT governance usually occurs at different layers, including team leaders reporting to and receiving direction from their managers, managers reporting up to the executive, and the executive to the board of directors (Report 2003).

B. *Governance of Agile Software Development Projects*

In an article published in Dr. Dobb's Portal in 2007 (Ambler 2007) the author stated that " it is a lot easier to govern agile projects than traditional ones". He supported this with two reasons, both are related to stakeholders' involvement, the first is that producing working software on a regular basis will give the stakeholders a great visibility to the work done by the team. Second, as the stakeholders are controlling the budget and schedule, they will direct the team effectively.

A workshop on software development governance has been running since 2008 within the international conference on software engineering (ICSE). The workshop focuses on the implementation of governance through tools and techniques in order to provide teams and organizations with the ability to effectively steer the business of software development. (Dubinsky et al. 2009). Governance for agile teams was included as one of the workshop themes. Two papers from this workshop will be discussed here. The first one studied the software development challenges that faced a company in an agile transaction. The study concluded that two challenges were related directly to functionality, lack of feedback loops, and lack of business theme prioritization. The rest were related to the company's transition to agile methods (Lehto et al. 2009). The second paper analyzed three governance events within a project that implemented agile practices including studying the metrics that triggered each event. The study concluded that governance iterations can be unified within agile development iterations which can be useful in identifying issues and resolving them in an effective and timely manner (Talby et al. 2009).

It can be easily observed that the research focusing on governance for agile projects is still in its early stages. Therefore, the agile projects governance survey will help understanding the topic by collecting data about what we think is related to agile projects governance.

III. The Method

The agile projects governance survey was a web-based survey. It was designed and run using the SurveyMonkey online survey tool. The questions were generated based on the undertaking and information gathered about IT governance and agile governance and on the questions raised during a quantitative empirical study conducted by the researcher. The survey has been reviewed and approved by the University of Southampton Ethical committee. The collected data was kept confidential and were used for research purpose only. The data and the results are anonymous; therefore, it was not possible to identify people, organizations, or projects from the data or from the results.

The data was collected during the month of September 2009. The survey was sent to all the major agile mailing lists available on Yahoo and Google groups. Also it was sent to Facebook agile groups. The message included an introduction to the survey, its purpose and goals with an indication that the survey is anonymous and a web link to the survey was included.

A. The Design

The survey consisted of three sections: gathering information about the respondent, their current or most recent project, and agile governance.

The "who you are" section asked about the respondent's position, experience, his/her organization sector, size, and how long he/she has been involved in agile development. Each question had a list of options that sometimes included an "other" option. The purpose of this section is to understand the respondents' background and their experience so existing relationships between different variables could be investigated.

The "on your current or most recent project" section asked specific questions about a project including the team size, the iteration and project length, the quality of the code produced, whether the project measured customer satisfaction and whether the project was successful. This section was included to get accurate data on the impact of agile code quality and project success. Therefore, it was necessary to collect information about a specific project rather than asking the question generally. This data will allow studying the impact of agile methods on project success, and software quality. Furthermore, it will help exploring the relations between project success and each of project length, team size, and iteration length.

A list of options followed each question. Ranges were used for the team size, iteration and project lengths, while scales were used for the other questions. Regarding the project success, it is always difficult to define success as it is different from a project to another, sector to another, and it is subjective. Therefore, the survey left it to the respondent's judgment, and gave the respondent a scale range from "definitely" to "clearly failed" including a "too early to say" option for projects that have just started.

The last section was about agile governance which is the main purpose of the survey. As no previous survey focused on this aspect, it is important to first explore the state of the art of agile governance, particularly, the use of retrospectives and metrics in agile projects. Most importantly, this data can be useful in identifying relationships and connections which may help providing advice so teams can improve the use of these techniques. The section started with a brief description of an agile retrospective as a meeting held at regular intervals where the team reflect on what went well and what did not and how to become more effective in future iterations/sprints. The questions in this section focused on retrospectives and reflection meetings, their frequency, length, comments recording, team participation and their impact on the team practices. Also, the section asked whether the respondent's organization collects any metrics. This question allowed multiple answers, as an organization may have more than one way to collect metrics, automatic, manual, part of the process or may have tried it and found it useless. The final question presented a list of metrics and asked whether the respondent's company is using them. Each metric had a five point scale ranging from always to never.

Both phrases, agile retrospective and reflection meetings were used so the questions are as clear as possible, which will lead to more accurate and suitable responses. The phrases are assumed to be synonymous. The survey questions are included in Table I in Appendix.

The survey has received 129 responses. All respondents completed the first section, 117 completed the first and the second section, and 106 completed all three sections.

B. The Analysis

The survey tool provides the data files in Excel, CVS format, as well as a coded (numerical) data set. The coded file was revised and changed as needed. This included recoding some questions so all the question follow the same coding pattern. This means the size is coded so the small code presents the smaller size, and for the scales, the smaller code presents bad quality, less frequently, the never option and so on. All the frequencies were calculated after coding and checked against the results to make sure the coding was done correctly.

In order to study any existed relationship between the different variables, correlation was used to analyze the data. Correlation is a measure of the relationship between these variables, however, in order to know what type of correlation is more appropriate, we need to explore the. Screening the data showed that that our data is not normally distributed. Therefore Spearman's correlation coefficient (rs) will be used, this correlation is nonparametric and it can be used when the data is not normally distributed. The correlation coefficient has to lie between -1 and +1, where a coefficient of +1 indicates a perfect positive relationship and a coefficient of -1 indicates a perfect negative relationship. A correlation coefficient value of ±.1 represents a small effect, ±.3 is a medium effect and ±.5 is a large effect. We have to be careful about correlation coefficients interpretation because they give no indication of the direction of causality (Field 2005).

The correlations were calculated using SPSS. SPSS was used as a tool for applying the analysis. First, because the software is provided by the University with introductory training, many books are available for self training, and most importantly it is a well respected tool among statisticians.

IV. THE RESULTS

This section will explore any existing relationships between the different variables studied in the survey. As there are many variables and many interesting correlations, the results will be discussed in four categories, each covering a number of related variables:

- Organisation variables (size, experience in agile development)
- Project variables (length, iteration length, team size, success, code quality, frequency of customer satisfaction measure)
- Retrospective variables (frequency, length, records, impact, team participation, team contribution)
- Metrics variables (general collection, usage of each metric)

The following subsections will discuss each category, the relationships within the category variables, and their relationships with other categories' variables.

A. *Organization Variables*

In this section we investigate organization's size and/or agile development experience have any impact on project success, code quality or the measure of customer satisfaction. Spearman correlation coefficient was applied on the coded data, and SPSS output is presented in table 1 in Appendix. Regarding the organization size the following significant correlations occur:

- There is a positive relationship between organisation size and the followings other variables:
  - Team size, rs =.38, (p <0.01) and also
  - Iteration length, rs =.30, (p <0.01)
- There is a negative relationship between organisation size and project success, *rs* =.26, (*p* <0.01)

Here we can see that large organizations tend to have bigger teams and longer iterations. Although this was for one project only, it could help understand the adoption of agile methods in large organizations. In addition, it seems that agile development is less successful with large organization. This negative relation also was founded when the agile adoption survey 2008 was analyzed as the correlation found (- .18 with p <0.01) between organization size and project success. The same results were found by (Livermore 2007) who reported significant negative correlation between implementation success and the size of the organization attempting to implement an agile methodology.

Regarding the organization's experience with agile development the correlation table shows that:

- There is a positive relationship between organisation agile experience and the following variables:
  - The frequency of customer satisfaction measure, *rs* =.29, (*p* <0.01)
  - Code quality, *rs* =.20, (*p* <0.05)
  - Project Success, *rs* =.30, (*p* <0.01)
  - Retrospective impact, *rs* =.21, (*p* <0.05)
  - Retrospective contribution, *rs* =.33, (*p* <0.01)
  - Retrospective records, *rs* =.28, (*p* <0.01)

As expected, organizations with more agile experience are doing much better regarding project success and code quality. Also, more experienced organizations measure customer satisfaction more frequently. Applying agile retrospectives seems to be more mature in more experienced organizations.

B. *Project Variables:*

The purpose of the second section of the survey is to study one specific project as reported by each respondent. Applying Spearman correlation on project' variables produced table 2 in Appendix were we can see that:

- There is a positive relationship between iteration length and project length, *rs* =.38, (*p* <0.01).
- There is a positive relationship between the frequency of measuring customer satisfaction and the followings other variables:
  - Code quality, rs =.47, (p <0.01)
  - Project success, rs =.40, (p <0.01)

- There is a positive relationship between code quality and project success, *rs* =.35, (*p* <0.01)

To summarize, the longer projects tend to have longer iterations. Also, the three factors, project success, code quality and measuring customer satisfaction are related positively. The point here is that measuring customer satisfaction once and getting good results is not good enough, instead the team is advised to measure customer satisfaction frequently. Interestingly, there is no significant relationship between team size and project success, yet it is a negative relationship (smaller teams tend to be more successful). Previous survey conducted by Livermore didn't find a significant relationship between these two factors (Livermore 2007). This comes in conflict with authors who argued that team size is an important factor and that agile methods do best with small teams (Cockburn et al. 2001; Boehm et al. 2003; Beck et al. 2004; Cockburn 2005).

C. Retrospective Variables

The survey asked different questions about agile retrospectives. The first question asked when the team performed them. As this question allowed multiple answers, a more suitable way to analyze its relationship with other variables is cross tabulation. The Surveymonkey tool provides this feature. Applying cross tabulation on this question gave interesting results.

Out of the 69 respondents who performed retrospectives after each iteration:

- 45% had a team of 1-6 people and fewer than 4% respondents had a team over 50
- 42% often measured customer satisfaction
- 56% had high code quality
- 56% had definitely successful projects
- 42% said retrospectives changed the team practices
- 68% manually collected metrics

Out of the 21 respondents who performed retrospectives when the team had problems:

- 33% had a team of 1-6 people and fewer than 4% respondents had a team over 50
- 33% always measured customer satisfaction
- 57% had high code quality
- 57% had definitely successful projects
- 38% said retrospectives changed the team practices
- 48% manually collected metrics

These results suggest that retrospectives are more suitable for small teams. Also the results recommend performing retrospectives after each iteration for the best outcomes regarding code quality and project success. The results suggest that using retrospectives when the team has a problem can be useful. Furthermore, it seems like the teams who perform retrospectives are also applying metrics. In other words, when a team does it right, it does for all aspects.

Spearman correlation was used to study the relationships between the other retrospective variables. Table 3 in Appendix presents the correlations between these different aspects and with the other variables in the survey.

- There is a positive relationship between retrospectives impact and the followings other variables:
  - o Customer satisfaction, rs =.31, (p <0.01).
  - o Code quality, rs =.24, (p <0.05).
  - o Project success, rs =.33, (p <0.01).
  - o Retrospective contribution, rs =.24, (p <0.05).
  - o Retrospective participation, rs =.44, (p <0.01).
  - o Retrospective recording, rs =.27, (p <0.01).
- There is a positive relationship between retrospective contribution and the followings variables
  - o Project success, rs =.27, (p <0.01)
  - o Retrospective impact, rs =.24, (p <0.05)
  - o Retrospective participation, rs =.54, (p <0.01)
- There is a negative relationship between retrospective contribution and iteration length, *rs* =.37, (*p* <0.01)
- There is a positive relationship between retrospective participation and the following variables
  - o Project success, rs =.26, (p <0.05)
  - o Retrospective impact, rs =.44, (p <0.01)
  - o Retrospective contribution, rs =.54, (p <0.01)
  - o Retrospective records, rs =.36, (p <0.01)

These results suggest that agile retrospectives are effective when applied properly. As reported earlier, the retrospectives had more impact when the whole team participated, everybody had their say and the retrospective comments were recorded. The findings suggested negative relations between team size and both retrospective impact and contribution.

D. Metrics Variables

The survey asked two questions about metrics. The first one asked generally about the organization policy regarding collecting metrics. This question allowed multiple answers, so similar to the retrospective frequency question, cross tabulation was applied, and here are the results highlights:

Out of the 40 respondents, whose companies automatically collected metrics

- 32% always measured customer satisfaction
- 67% had high code quality

- 57% had definitely successful projects
- Out of the 61 respondents, whose companies manually collected metrics:
- 43% always measured customer satisfaction
- 56% had high code quality
- 57% had definitely successful projects

The results show a relationship between collecting metrics and different variables; however, it is not as strong as expected.

Spearman correlation was applied on the list of different metrics. The significant correlations suggest that:

- There is a positive relationship between burn charts and the following variables
  - Story points, rs =.64, (p <0.01)
  - Team velocity, rs =.57, (p <0.01)
  - Defect count after testing, rs =.21, (p <0.05)
- There is a positive relationship between story points and team velocity, *rs* =.61, *(p* <0.01)
- There is a positive relationship between function points and the following variables
  - Business value, rs =.21, (p <0.05)
  - Running testing features, rs =.25, (p <0.01)
  - Number of test cases, rs =.31, (p <0.01)
  - Time to obstacle removal, rs =.46, (p <0.01)
  - Obstacles removed per iteration, rs =.55, (p <0.01)
- There is a positive relationship between team velocity and the following variables
  - Business value, rs =.27, (p <0.01)
  - Running testing features, rs =.20, (p <0.05)
- There is a positive relationship between business value delivered and the following variables
  - Running testing features, rs =.37, (p <0.01)
  - Defect count after testing, rs =.33, (p <0.01)
  - Number of test cases, rs =.22, (p <0.05)
  - Time to obstacle removal, rs =.40, (p <0.01)
  - Obstacles removed per iteration, rs =.45, (p <0.01)
- There is a positive relationship between Running testing features and the following variables
  - Defect count after testing, rs =.40, (p <0.01)
  - Number of test cases, rs =.47, (p <0.01)
  - Time to obstacle removal, rs =.45, (p <0.01)
  - Obstacles removed per iteration, rs =.40, (p <0.01)
- There is a positive relationship between defects count after testing and the following variables
  - Number of test cases, rs =.57, (p <0.01)
  - Time to obstacle removal, rs =.32, (p <0.01)
  - Obstacles removed per iteration, rs =.32, (p <0.01)
- There is a positive relationship between number of test cases and the following variables
  - Time to obstacle removal, rs =.29, (p <0.01)
  - Obstacles removed per iteration, rs =.36, (p <0.01)
  - Defect count after testing, rs =.32, (p <0.01)
- There is a positive significant relationship between number of test cases and obstacles removed per iteration, *rs* =.82, (*p* <0.01)

The results suggest that when a team is collecting metrics, it does not collect only one. For example the team who measured the “business value delivered” metric also measured every other metric provided apart from the first two. In addition, although the function points metric is considered as a traditional one, teams who collected it also collected agile metrics.

Finally, correlation was used to test the relationship between project success and the different metrics, the results showed that there is a positive significant relationship between project success and the following metrics:

- Team velocity, , *rs* =.36, (*p* <0.01)
- Business value delivered, , *rs* =.40, (*p* <0.01)
- Running testing Features, , *rs*=.23, (*p* <0.05)
- Defect count after testing, *rs*=.31, (*p* <0.01)
- Number of test cases, *rs*=.22, (*p* <0.05)

There was no other significant relationship with the other metrics.

E. Project Success Factors

The survey asked about project success, therefore it might be useful to cross tab this question and see which factors made a project succeed in the respondents opinion. Out of the 53 projects that were definitely successful, the factors with 60% or over will be considered here:

- 72% had teams of 10 people or less (small team size)
- 77% always or often measured customer satisfaction (frequent customer satisfaction measure) (supported by correlation results)
- 79% had high or very high code quality (high code quality) (supported by correlation results)

- 78% performed retrospective after each iteration (performing retrospectives after each iteration)
- 66% had the whole team participate in the retrospective (having the whole team participating in the retrospective) (supported by correlation results)
- 88% reported that everybody can have their say in the retrospective (team contribution retrospective) (supported by correlation results)
- 64% manually collected metrics (collecting metrics)
- 60% never measured OR/I (choosing the needed metrics for the project)

This list can be used by agile team as a guide to achieve success as it is based on a survey that collected 129 responses from people who had different positions, experiences, and who came from different teams and organizations.

Previous research has discussed success factors in agile software development. A survey was conducted to collect collected data about 109 projects and used regression to analyze the data. The study listed six critical success factors for agile software development; each had a number of attributes. The suggested success factors are delivery strategy, agile software engineering techniques, team capacity, project management process, team environment, and customer involvement (Chow et al. 2008).

V. Validity Issues

The great strength of survey research is that for a relatively little cost we can collect data about a number of variables from a large number of persons. Also, surveys are useful in collecting data on aspects of behavior that are difficult to observe directly. However, surveys also have a number of limitations. The most serious weakness concerns the validity and reliability of responses obtained as the collected data is self-reported, and poor memory, or misunderstanding of the questions can all contribute to inaccuracies in the data (Nardi 2002). Some teams may wrongly claim to be working in an agile way when they merely using a subset of agile practices.

Specifically to the agile projects governance survey, it was difficult to calculate the response rate as the survey was distributed to online email groups whose numbers are constantly changing and it is impossible to guarantee that all group members are active members in the group and thus had read the distributed email. Also, it is difficult to know much about the sample population, except they presumably have an interest in agile methods in general and the governance studied aspects in particular. Regarding the survey design, some questions were not very useful to the study such as the position and organization sector. Especially since the position question got high number of responses choosing the "other" option which indicated that there are more positions that were not included in the options list. The analysis linked the questions about a specific project to general organization questions that maybe or maybe not applied to that specific project. However, it was difficult to ask all questions about one project, as it was needed to understand the organization strategy regarding agile governance. In addition, in order to get more accurate results, the survey should have focused on either current or recent project so the data can give a better idea about the project length and its relation to other variables. Finally, it was difficult to apply different statistical tests, as the data collected are nominal and ordinal. Furthermore, the ordinal questions had a "not applicable" option which was necessary to make the survey user-friendly. However, this made it difficult to code the data and treat it as interval/ratio. So when coding the data the N/A responses were treated as missing so that the correlations are meaningful.

Finally the survey focused on one aspect of governance; however, we understand that there are other meanings as administrating and managing in general which are not covered in the survey, nor in the paper.

VI. Conclusions

This paper presented the agile project governance survey and the analysis of its results. The survey collected 129 responses of which 103 completed the survey's three sections. The respondents had more experience in IT than in agile, came from organizations that varied in terms of size, sector and experience in agile, and are relatively new to agile development.

Large organizations tend to have bigger teams and longer iterations. Although this was for one project only, it could help understand the adoption of agile methods in large organizations.

As expected, organizations with more agile experience are doing much better regarding project success and code quality. Also, more experienced organizations measure customer satisfaction more frequently. Applying agile retrospectives seems to be more mature in more experienced organizations.

The longer projects tend to have longer iterations. Also, the three factors, project success, code quality and measuring customer satisfaction are related positively. The survey results suggest that frequent measure of customer satisfaction is worthwhile. Interestingly, there is no significant relationship between team size and project success, yet it is a negative relationship.

These survey results suggest that retrospectives are more suitable for small teams. Also the results recommend performing retrospectives after each iteration for the best outcomes regarding code quality and project success. The results also suggest that using retrospectives when the team has a problem can be useful. Furthermore, it seems like the teams who perform retrospectives are also applying metrics. In other words, when a team does it right, it does for all aspects. Moreover, agile retrospectives are effective when applied properly. The retrospectives had more impact when the whole team participated, everybody had their say, and the retrospective comments were recorded properly. The findings suggested negative relations between team size and both retrospective impact and contribution.

The results suggest that when a team is collecting metrics, it does not collect only one. For example the team who measured the “business value delivered” metric they also measured every other metric provided apart from the first two.

Finally, the reported successful projects had small team sizes (10 or less), frequently measured customer satisfaction, had high code quality, performed retrospective after each iteration, had the whole team participating in the retrospective, had everybody participating in the retrospective, and collected metrics either manually or automatically.

ACKNOWLEDGMENT

We would like to thank everyone who took the time to fill the agile projects governance survey. Also, we would like to acknowledge Scott Ambler for providing comments on the agile project governance survey design.

REFERENCES

Ambler, S. (2007). "Results from Scott Ambler’s 2007 Agile Adoption Rate Survey " Retrieved 28/07/2008, from www.ambysoft.com.

Beck, K. and C. Andres (2004). Extreme Programming Explained: Embrace Change (2nd Edition), Addison-Wesley Professional.

Boehm, B. and R. Turner (2003). Balancing Agility and Discipline: A Guide for the Perplexed, Addison-Wesley Longman Publishing Co., Inc.

Chow, T. and D.-B. Cao (2008). A Survey Study of Critical Success Factors in Agile Software Projects, Elsevier Science Inc. 81: 961-971.

Cockburn, A. (2005). Crystal Clear A Human -Powered Methodology for Small Teams, Addison-Wesley.

Cockburn, A. and J. Highsmith (2001). "Agile Software Development: The People Factor " Computer 34(11): 131-133.

Dubinsky, Y. and P. Kruchten (2009). 2nd Workshop on Software Development Governance (SDG). Proceedings of the 2009 31st International Conference on Software Engineering: Companion Volume, IEEE Computer Society.

Field, A. (2005). Discovering Statistics Using SPSS, Sage.

governance. (2009). "Compact Oxford English Dictionary Online." Retrieved 09/12/2009.

Highsmith, J., k. Beck, A. Cockburn and R. Jeffries. (2001). "Agile Manifesto." Retrieved 28/07/2008, from www.agilemanifesto.org.

Lehto, I. and K. Rautiainen (2009). Software Development Governance Challenges of a Middle-sized Company in Agile Transition. Proceedings of the 2009 ICSE Workshop on Software Development Governance, IEEE Computer Society.

Livermore, J. A. (2007). Factors that Impact Implementing an Agile Software Development Methodology. SoutheastCon, 2007. Proceedings. IEEE.

Nardi, P. M. (2002). Doing Survey Research: A Guide to Quantitative Research Methods Allyn & Bacon.

Report (2003). Board Breifing on IT Governance, IT Governance Institute.

Talby, D. and Y. Dubinsky (2009). Governance of an Agile Software Project. Proceedings of the 2009 ICSE Workshop on Software Development Governance, IEEE Computer Society.

Webb, P., C. Pollard and G. Ridley (2006). Attempting to Define IT Governance: Wisdom or Folly? Proceedings of the 39th Annual Hawaii International Conference on System Sciences - Volume 08, IEEE Computer Society.

Weill, P. and J. W. Ross (2004). IT Governance on One page. CRSR Working Paper N.349.

APPENDIX

TABLE I. THE SURVEY QUESTIONS

WHAT DESCRIBES BEST YOUR CURRENT POSITION IN THE ORGANISATION?

Business Stakeholder   Developer   Scrum Master   Project Manager   Tester   Quality Assurance   Architect   Other

HOW MANY YEARS OF EXPERIENCE IN IT DO YOU HAVE?

None   Less than 2 years   3-5 years   6-10 years   11-20 years   21+ years

HOW MUCH YEARS OF EXPERIENCE IN AGILE DEVELOPMENT DO YOU HAVE?

None   Less than 2 years   3-5 years   6-10 years   11+ years

WHAT IS THE TOTAL NUMBER OF PEOPLE IN YOUR ORGANISATION?

1-10   11-100   101-1000   1001-10000   10001-100000   Over 100000

HOW LONG YOUR COMPANY HAS BEEN DOING AGILE DEVELOPMENT?

We have no agile experience   Less than 1 year   1-2 years   3-5 years   6-10 years   11+ years

HOW OFTEN DO YOU MEASURE CUSTOMER SATISFACTION?

Always   Often   Sometimes   Rarely   Never

HOW DO YOU RATE THE CODE QUALITY?

Very high   High   Average   Low   Very low

WAS THE PROJECT SUCCESSFUL?

Definitely   Somewhat   Partially   Clearly failed   Too early to say

WHEN DO YOU PERFORM YOUR RETROSPECTIVES/REFLECTIONS MEETING?

After each iteration   Every other iteration   When we have problems   After each release sometimes   Rarely   Never

DO YOU HAVE A RECORD OF THE COMMENTS THAT WE MADE DURING THE RETROSPECTIVE/REFLECTION MEETING?

Always   Often   Sometimes   Rarely   Never   N/A

DOES THE WHOLE TEAM PARTICIPATE IN THE RETROSPECTIVE/REFLECTION MEETING?

Always   Often   Sometimes   Rarely   Never   N/A

CAN EVERYONE HAVE THEIR SAY IN THE MEETING?

Always   Often   Sometimes   Rarely   Never   N/A

DOES THE RETROSPECTIVE CHANGE YOUR PRACTICES?

Always   Often   Sometimes   Rarely   Never   N/A

IN YOUR COMPANY, DO YOU COLLECT ANY MEASURES OR METRICS? (MULTIPLE ANSWERS ARE ALLOWED)

We automatically generate metrics using tools
We manually generate metrics
We tried collecting metrics but we found them useless
We have to, it is part of our process
Do not know

IN YOUR COMPANY, DO YOU MEASURE/USE ANY OF THE FOLLOWING?

Burn charts   Story points   Functions points   Team velocity   Business value delivered   RTF: Running testing features   Defect count after testing
Number of Test cases   TTOR: Time to obstacle removal   OR/I: obstacles removed per iteration

TABLE II. ORGANIZATION VARIABLES CORRELATIONS

	Organization Size	Organization Agile Experience
Organization Size	1.000	.038
Organization Agile Experience	.038	1.000
Team Size	.387**	.058
Iteration Length	.308**	.007
Project Length	.122	-.111
Customer Satisfaction Measure	-.096	.298**
Code Quality	-.099	.204*
Project Success	-.267**	.402**
Retrospective Impact	-.111	.211*
Retrospective Contribution	-.091	.339**
Retrospective Participation	-.084	.289**
Retrospective Records	.134	.055

** Correlation is significant at the 0.01 level (2-tailed)

* Correlation is significant at the 0.05 level (2-tailed)

TABLE III. PROJECT VARIABLES CORRELATIONS

	Team Size	Iteration Length	Project Length	Customer Satisfaction Measure	Code Quality	Project Success
Team Size	1.000	.158	.128	.069	.121	-.060
Iteration Length		1.000	.380**	.007	-.123	-.133
Project Length			1.000	-.120	-.163	-.137
Customer Satisfaction Measures				1.000	.470**	.402**
Code Quality					1.000	.351**
Project Success						1.000

** Correlation is significant at the 0.01 level (2-tailed)

TABLE IV. Retrospective variables correlations

	Retrospective Impact	Retrospective Contribution	Retrospective Participation	Retrospective Records
Team Size	-.002	-.140	.001	.002
Iteration Length	-.119	-.375**	-.155	.057
Project Length	-.063	-.162	-.079	.022
Customer Satisfaction	.316**	.153	.139	.065
Code Quality	.240*	.185	.125	.102
Project Success	.334**	.271**	.262*	.039
Retrospective Impact	1.000	.248*	.440**	.275**
Retrospective Contribution	.248*	1.000	.543**	.172
Retrospective Participation	.440**	.543**	1.000	.376**
Retrospective Records	.275**	.172	.376**	1.000

**** Correlation is significant at the 0.01 level (2-tailed)**

*** Correlation is significant at the 0.05 level (2-tailed)**

TABLE V. Metrics Correlation

	Burn Charts	Story Points	Function Points	Team Velocity	BVD	RTF	Defect Count After Testing	Number of Test Cases	TTOR	OR/I
Burn Charts	1.000	.640**	.091	.579**	.109	.159	.217*	.094	.094	.062
Story Points		1.000	.112	.615**	.052	.052	.052	.019	-.016	-.010
Function Points			1.000	-.026	.219*	.258**	.188	.318**	.464**	.551**
Team Velocity				1.000	.270**	.207*	.132	.064	.045	.063
BVD					1.000	.370**	.333**	.220*	.406**	.450**
RTF						1.000	.400**	.474**	.450**	.403**
Defect Count AT							1.000	.578**	.327**	.322**
Test Cases								1.000	.298**	.363**
TTOR									1.000	.820**
OR/I										1.000

**** Correlation is significant at the 0.01 level (2-tailed)**

*** Correlation is significant at the 0.05 level (2-tailed)**