Relations between Project Size, Agile Practices and Successful Software Development

Magne Jørgensen

Simula Metropolitan, Oslo, Norway & University of Oslo, Norway

ABSTRACT. The use of agile methods in the execution of large-scale software development is increasing. To find out more about the effect of this on project performance, information was collected about 196 Norwegian IT-projects. Increased project size was associated with decreased project performance for both agile and non-agile projects, but the projects using agile methods had better performance than the non-agile projects for all examined project size categories. Flexible scope, frequent deliveries to production, a high degree of requirement changes and more competent providers are candidates to explain the better performance of agile projects.

Keywords: agile methods, project performance, project characteristics

A traditional response to increased size and complexity of work is to implement more planning and management formalism [1]. Agile software development methods, on the other hand, try to remove or reduce much of the traditional project management formalism. Does this mean that agile, as indicated in [2] mainly work for smaller projects? Or do agile methods work well for larger projects as suggested in [3, 4]. The available empirical evidence is mixed and does not allow strong claims. In addition, the evidence does not give much insight into when, if at all, agile methods tend to work well for larger projects. This shortage of empirical evidence motivated the survey reported in this paper, aiming at answering the following two questions:

1) How well do larger agile software projects perform compared to smaller projects and non-agile projects?

2) Which agile practices and characteristics are connected with better performance?

The Survey

Respondents and data collection

The survey participants were Norwegian software professionals visiting three different seminars on project management in 2016 and 2017. The software professionals provided information about their *last* completed projects. 216 responses were received. After removing responses without the minimum information needed for the analysis, i.e. the budget size category, the development method and the perceived performance of the project, there were 196 unique responses remaining. The project information was given anonymously, in Norwegian, using the survey tool Qualtrics. There was a "don't know" option for all project information items to ensure that the respondents only answered when they felt they had sufficient knowledge.

The software professionals had on average 13 years of experience, with 70% having 8 or more years. 69% of the respondents were from the provider side and 31% were from the client side. 71% had technical roles in the reported project, e.g. architects or developers, and 29% had managerial roles, e.g. product owners, team leaders and project managers.

Project characteristics

The project characteristics requested from the participants is described in Table 1. The included variables are those that were found to distinguish between successful and failed software projects in an earlier survey [5]. To avoid too few observations for some categories the analyzed category "high" ("low") includes both "very high" ("very low") and "high" ("low") responses.

Table 1. Project characteristics*

Characteristic	Categories
Budget size (used as measure of project size)[1]	Small (<1 mill Euro) Medium (1-10 mill Euro) Large (>10 mill Euro)
Development method[2]	Agile Non-agile
Requirement volatility[3]	High (>30% changes) Low (<=30% changes)
Perceived flexibility of scope	High Low
Perceived detail of upfront project plan	High Low
Perceived detail of upfront requirement specification	High Low
Frequency of deliveries to production[4]	>4 per year <= 4 per year

Contract type	Time & materials Fixed price
Perceived provider competence	High Low
Perceived client competence	High Low

*: The full questionnaire is available to interested readers upon request.
1: The budget size categories small, medium and large are the same as those found to separate the effect of agile practices in [4].
2: There is no commonly accepted definition of what it means to work agile. I used the respondents' own perception of whether they worked agile or not in the first analysis and added analyses of the effect of different agile practices and characteristics in the second analysis.
3: The threshold of 30% is based on what was closest to the median level of perceived amount of requirement change of the projects.
4: The original categories were "none", "1-4" and "more than 4", where the two first were joined. Notice that even non-agile projects, e.g., incremental or timeboxing-based projects, may have deliveries to production during the project execution.

Project performance

After describing characteristics of the project, each participant assessed the performance of their last completed project, as he/she perceived it, using the scale: *very successful – successful – acceptable – problematic – very problematic* for each of the success dimensions: client benefits (value), cost control, time control, productivity and technical quality.

To define the project's overall performance, we used the following categorization:

Successful:	Successful or better on all five success dimensions,
Acceptable:	Acceptable or better on all five success dimensions
Failed:	Very problematic on at least one success dimension.

Data collection challenges

Different participants may be involved in the same projects, leading to the possibility of duplicate projects in our data set. The variance in organizations of the participants, as analyzed from the list of seminar participants and the typically large size of their organizations, indicates that the number of duplicates, if any, is very low.

Participants from the client and the provider side, as well as participants in different roles, may have different knowledge and perceptions of a project's performance. While this subjectivity may affect the accuracy of the reported success and failure rates, it is less likely to change the direction of the connection between development methods, project size and project performance.

An examination of the list of participants shows that the majority of them belong to or worked for large organizations with mainly administrative software applications. Consequently, the results may mainly be valid within this context.

Results

In total, 16% of the software projects were categorized as successful, 52% as acceptable and 7% as failed. The small and medium sized projects had the best performance with 15% and 22% categorized as successful, 55% and 50% as acceptable, and 7% and 4% as failed, respectively. The larger projects had 5% categorized as successful, 41% as acceptable and 14% as failed. The decrease in project performance with increased project size corresponds to findings in other studies, e.g., [6].

Seventy-four percent of the projects were categorized as agile. These projects, see Table 2, had better average success rate than the non-agile projects for all three size categories. Figure 1 displays this interaction effect for projects with acceptable project performance. An analysis using a general linear model (GLM) with the variable development method (agile and non-agile) nested into the variable budget size (small, medium and large) gives that the difference in proportion acceptable projects is statistically significant, with agile being more successful, for small ($p<0.01$) and medium ($p=0.03$) sized projects, but not for large sized projects ($p=0.12$).

Table 2. Relationship between budget size category, development method and project performance*

Project performance	Development method	Small (n=120)	Medium (n=54)	Large (n=22)
Successful (n=31)	Agile	19%	24%	7%
	Non-agile	0%	19%	0%
Acceptable (n=102)	Agile	65%	58%	50%
	Non-agile	19%	31%	25%
Failed (n=13)	Agile	2%	3%	14%
	Non-agile	23%	6%	13%

* The percentages are the proportion of successful, acceptable and failed projects for projects same budget size category and same development method.

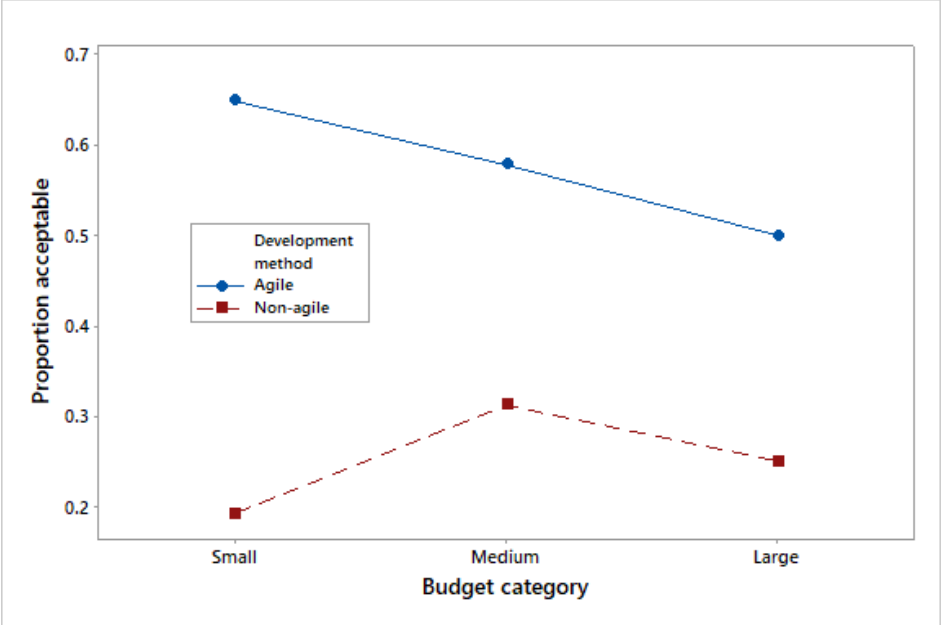

Figure 1. Interaction plot of projects with acceptable performance

The analysis of practices and context characteristics (factors) potentially connected with better performance of agile projects was completed as follows. First, the factors more frequently observed in agile than in non-agile projects were identified through a chi-square analysis. These factors may explain the better performance of agile projects even if they have a similar, positive effect on non-agile projects. Second, the connection between all factors and acceptable project performance (the performance category with most observations) was analyzed.

The factors associated with a statistically significant (here set as $p<0.05$) higher proportion of agile projects were *high requirement volatility* (50% of agile projects and 33% of non-agile projects had more than 30% requirement changes, $p=0.04$), *frequent deliveries to production* (68% of agile projects and 32% of non-agile projects had more than four deliveries to production per year, $p<0.01$) and *flexible scope* (79% of agile projects and 47% of non-agile projects had a perceived high degree of scope flexibility). There were no statistically significant differences in proportion of projects with *detail of project plan* (60% of agile and 53% of non-agile projects were perceived to have little detail in project plans, $p=0.75$), *detail of requirement specification* (55% of agile and 54% of non-agile projects were perceived to have little detail in requirement specification, $p=0.96$), and *contract type* (51% of agile and 58% of non-agile used fixed price contracts, $p=0.53$).

Table 3 displays the results for the proportion of projects with acceptable performance for the analyzed factors. Notice that the sum of observations is lower than the full dataset of 196 projects due to “don’t know” answers.

Table 3. Proportion projects with acceptable performance*

Factor	Category	Agile (n=146)	Non-agile (n=50)
Requirement volatility	High (n=80)	58%	13%
	Low (n=97)	61%	29%
Delivery frequency	>4 per year (n=99)	70%	25%
	<=4 per year (n=60)	49%	21%
Scope flexibility	High (n=71)	85%	33%
	Low (n=26)	50%	40%
Detail of project plan	High (n=59)	67%	18%
	Low (n=81)	53%	21%
Detail of req. spec.	High (n=63)	55%	13%
	Low (n=76)	61%	26%
Contract type	Fixed price (n=66)	60%	17%
	Time & materials (n=60)	60%	23%

* The percentages are the proportion of acceptable projects for projects with same factor category and same development method. There are too few observations (low statistical power) for some of the combinations of categories to conduct meaningful tests of statistical significance for the interactions in Table 3. The differences should consequently be interpreted as indications of relationships, not as strong evidence.

The results in Table 3 suggest that experiencing high requirement volatility did not greatly affect the proportion of acceptable agile projects, while the proportion of acceptable non-agile projects decreased from 29% to 13%. Frequent delivery to production seems to have had a much stronger positive connection with better performance for agile than for non-agile projects. This practice was also much more common among agile projects and may therefore contribute to a better performance of agile projects both by being more frequently used and by having a stronger positive connection. Higher scope flexibility was connected with much higher proportion of acceptable performance for agile projects, and a lower proportion for non-agile projects. The factors including detail of project plan, detail of requirement specification and contract type did not contribute much to explaining an improved performance of agile projects.

Table 4 suggests that as the project size increased from small to medium/large a high degree of requirement changes further increased the superior performance of the agile projects. A higher delivery frequency was associated with larger increase in acceptable agile than in acceptable non-agile projects. Similarly, higher flexibility of scope was associated with increased performance of small agile and decreased performance of small non-agile projects.

Table 4. Proportion projects with acceptable performance, per size category*

Factor	Category	Agile		Non-agile	
		Small (n=94)	Medium/large (n=52)	Small (n=26)	Medium/large (n=24)
Requirement volatility	High (n=80)	62%	54%	13%	14%
	Low (n=97)	65%	47%	20%	38%
Delivery frequency	High (n=99)	73%	65%	-	38%
	Low (n=60)	54%	41%	13%	27%
Scope flexibility	High (n=71)	86%	84%	14%	-
	Low (n=26)	55%	40%	57%	-

* The percentages are the proportion of acceptable projects for projects with same factor category, budget size category and development method. There are too few observations (low statistical power) in some of the categories to conduct meaningful tests of statistical significance for the interactions in Table 4. The differences should consequently be interpreted as indications of relationships, not as strong evidence. The fields with "-" have fewer than five observations, due to missing data about a project or few occurrences, and the proportions were not calculated.

If agile projects attract more competent providers or clients, this may contribute to the difference between agile and non-agile projects. An analysis of the project data demonstrated that the agile software projects were indeed perceived to have more competent clients and providers (Chi-square test of independence gives p=0.02 and p=0.01, respectively). A binary logistic regression model with the elements client competence (high vs. low), provider competence (high vs low), development method (agile vs non-agile), requirement volatility (high vs. low), delivery frequency (high vs. low) and scope flexibility (high vs low), using the performance measure acceptable (1=acceptable, 0=not acceptable) as the dependent variable give much higher likelihoods (odds ratios of 5.7 and 2.4) of observing an acceptable project when having a high compared to a low or medium competent provider (p=0.046) and client (p=0.27, not statistically significant). More studies are needed to analyse how client and provider competence interact with agile practices and contexts to explain differences in project performance.

Conclusions

The survey of 196 Norwegian software projects provides empirical support for the use of agile methods on larger as well as smaller software projects, especially when including flexible scope and frequent delivery to production, and in contexts with high requirement changes. A contributing factor may be that agile projects tend to have more competent providers and clients.

References

[1] Child, J., *Predicting and understanding organization structure.* Administrative Science Quarterly, 1973. **18**(2): p. 168-185.

[2] Dybå, T. and T. Dingsøyr, *Empirical studies of agile software development: A systematic review.* Information and software technology, 2008. **50**(9): p. 833-859.

[3] Dingsøyr, T., N.B. Moe, T.E. Fægri, and E.A. Seim, *Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation.* Empirical Software Engineering, 2018. **23**(1): p. 1-31.

[4] Jørgensen, M. *Do Agile Methods Work for Large Software Projects?* In *International Conference on Agile Software Development*. 2018. Porto, Portugal: Springer.

[5] Jørgensen, M., *A survey on the characteristics of projects with success in delivering client benefits.* Information and Software Technology, 2016. **78**: p. 83-94.

[6] Sauer, C., A. Gemino, and B.H. Reich, *The impact of size and volatility on IT project performance.* Communications of the ACM, 2007. **50**(11): p. 79-84.

Biography: Magne Jørgensen (magnej@simula.no) is professor in software engineering at the University of Oslo and chief research scientist at Simula Metropolitan. He has extensive industry experience as consultant and manager and currently serves at the Norwegian digitalisation board where he advises governmental software projects. His research interests include project management, evidence-based software engineering and expert judgment. His recent book on effort estimation can be downloaded for free from: tinyurl.com/timepredictions.