Functional Size Measures and Effort Estimation in Agile Development: a Replicated Study

Valentina Lenarduzzi[1], Ilaria Lunesu[2], Martina Matta[2], Davide Taibi[3]

[1] Università degli Studi dell'Insubria,
21100 Varese, Italy
[2] Università degli Studi di Cagliari
Piazza d'Armi, 09123 Cagliari, Italy
[3] Free University of Bolzano
Piazza Domenicani 3, 39100 Bolzano, Italy
valentina.lenarduzzi@gmail.com, {martina.matta, ilaria.lunesu}@diee.unica.it
davide.taibi@unibz.it

Abstract. To help developers during the Scrum planning poker, in our previous work we ran a case study on a Moonlight Scrum process to understand if it is possible to introduce functional size metrics to improve estimation accuracy and to measure the accuracy of expert-based estimation. The results of this original study showed that expert-based estimations are more accurate than those obtained by means of models, calculated with functional size measures. To validate the results and to extend them to plain Scrum processes, we replicated the original study twice, applying an exact replication to two plain Scrum development processes. The results of this replicated study show that the accuracy of the effort estimated by the developers is very accurate and higher than that obtained through functional size measures. In particular, SiFP and IFPUG Function Points, have low predictive power and are thus not help to improve the estimation accuracy in Scrum.

1 Introduction

In projects developed with Scrum [19], effort estimations are carried out at the beginning of each sprint, based on developer experience. However, as reported by several empirical studies, developers, involved in agile processes, usually underestimate their effort in agile processes [5, 6, 7, 8 and 13].

In order to understand if functional size measures allow for more accurate effort estimates, a case study on a Moonlight Scrum process was conducted in our previous work [1]. There we investigated whether functional size measures can help to improve the effort estimation accuracy of Scrum user stories and compared the accuracy of the resulting effort model with the developers estimated effort. The study shows that, in Moonlight Scrum, the estimation of developers is more accurate than estimation based on functional measurement and therefore functional measures do not help developers in improving the accuracy of effort estimation in Moonlight Scrum.

Since the case study was applied to a slightly modified version of Scrum, we expect that different results might be obtained when applying the approach to a plain Scrum process.

Thus, we investigate these two research questions:

RQ1: Can we extend the results obtained to the original case study to plain Scrum processes?

RQ2: Does IFPUG Function Points help to increase the effort estimation, compared to SiFP?

In order to answer to our research questions, in this work we performed two exact replications of the original case study [4], directly involving the original authors of the original case study, together with those responsible for the development of the new two development processes.

No changes to the original study design were applied, except for the context in which the study was executed. In this case, the development process changed from Moonlight Scrum to a plain Scrum process.

The result of this work will provide input for future research directions over than the validation of existing results.

The information reported in this paper is organized as suggested by the guidelines for replicating controlled experiments [3]. Section 2 describes the original case study. Section 3 presents the new study contexts and design, highlighting the similarities and differences with the original design. Section 4 presents the results of the study and compares them with the original study. Section 5 discusses results and describes the threats to validity and finally, Section 6 draws conclusions.

2 The Original Case Study

In this section, we will describe the original study [1], providing information on the study design and describing the research questions, the goal of the study, the measures identified, and the protocol adopted. Then we will describe the study context, highlighting the variables that affected the design of the study, and finally we will provide a brief overview of the major findings.

2.1 Study Design

The goal of the original study was formulated by means of the Goal Question Metric approach [2] as: *analyze* the development process *for the purpose of* evaluating the effectiveness of functional measures for effort estimation *from the viewpoint of* the developers *in the context of* a Scrum development process.

One of the most important requirements was that measures must be collected within a maximum of five minutes per user story at the end of the usual Scrum planning game, so as to not influence the normal execution of the required Scrum practices.

For this purpose, we identified a set of measures to be collected for each user story at the end of every sprint meeting.

To measure user stories, we first investigated the feasibility of existing functional size measures. Since standard Function Points such as IFPUG [15] or FISMA require a lot of effort to be collected, and most of the required information was not available in our context, we opted for Simplified Function Points (SiFP) [12]. We collected

SiFP instead of IFPUG Function Points because SiFP provides an ñagileò, simplified and alternative measure that is compatible with IFPUG Function Points [15, 17].

SiFP are calculated as SiFP= 7 * #DF + 4.6 * #TF, where #DF is the number of data functions (also known as logic data files) and #TF is the number of elementary processes (also known as transactions). For this reason, we collected information for DF and TF separately. Moreover, we also split TF into two sub-processes: input processes iTF (data received from the server) and output processes oTF (data sent to the server). TF was finally calculated as (iTF+oTF)/2.

Then, before running this study, we asked our developers what information they take into account when estimating a user story. All developers answered that they consider four pieces of information, based on the complexity of implementing the GUI and the number of functionalities to be implemented. They usually consider each GUI component as a single functionality that requires sending or receiving information to/from the database. The complexity of the communication is related to the number of tables involved in the SQL query.

For these reasons, we also considered the following measures:

- GUI Impact: null, low, medium, high: complexity of the GUI implementation identified by the developers.
- # GUI components added: number of data fields added (e.g., HTML input fields)
- # GUI components modified: number of data fields modified.

Finally, we also collected some context information, such as the story type (new features or maintenance), so as to understand whether new development tasks should be estimated differently from maintenance tasks.

2.2 Study Context

The case study was applied in the context of the development of a web-based application [14] developed in C#/Asp.net with a simple 3-tier architecture that allows the development of independent features among developers.

The application developed is a relatively small application, composed of 12,500 effective lines of code developed using a Moonlight Scrum process [11], a special version of Scrum.

The development was carried out by four part-time developers (Masterôs students) with 2 to 3 yearsô of experience in software development and was organized as follows:

- The duration of each sprint is three weeks.
- Daily meetings are replaced by reporting on an online forum twice a week.
- Only one developer can work on a user story.
- Each developer works 8 hours per week.
- Every developer works in isolation during non-overlapping hours.
- The work is coordinated by the Scrum master via the weekly meetings.

2.3 Study Results

The project was analyzed for four months. A total of 136 user stories were examined, of which 65% were related to the development of new features, while only 35% were related to maintenance. Moreover, in this process, most of the user stories were related to the development of graphical features with high or medium complexity.

Functional measures were collected only for 55 user stories (40.4%) since the remaining user stories did not contain enough information for functional size measurement (e.g., GUI features that do not deal with data transactions).

The analysis of correlations between SiFP and effort reported in all user stories did not provide any statistically significant result showing very low goodness of fit. Even when we tried to cluster the user stories by story type and GUI impact, the results showed the same behavior.

A similar pattern was shown for the correlation between the number of GUI components added or modified and the multivariate correlations among GUI components added, GUI components modified and Data Files provided statistically significant results paired with low correlation.

The results finally showed that functional measures are not applicable to a Moonlight Scrum process.

Since the study focused on Moonlight Scrum, a slightly modified version of Scrum, we expected some variations in applying the same approach to a full-time development team working on a plain Scrum process.

3 Study Context and Design

Our studies are designed so as to accurately replicate the original study conducted in [1] by using the same research goals and study design as reported in Section 2.

In this section, we will describe the contexts of both studies. Then we will highlight similarities and differences of the context and the design of the new studies with, respectively to, the context of the original study.

The projects analyzed in the new study were developed at the Software Factory lab of the University of Cagliari (Italy).

The development process was Scrum, with the support of a Kanban board [16], without tight WIP limits, in order to visualize in each instant the work in flow.

The development processes were organized as follows:

- The duration of each sprint is two weeks.
- Daily meetings must last at most 10 minutes.
- Developers work two days per week for eight hours a day (16 hours per week).
- A user story can be developed only by one developer (no pair programming).
- Every developer works in the same room during the same hours in order to improve collaboration and communication.
- The development is coordinated by a coach with perfect knowledge of the project and the technologies, who is also involved in the development.

- All developers are actively involved in Sprint retrospectives, planning, and retrospective discussions, making important contributions in order to obtain a good final result.

As in the original case study, the project was developed using a 3-tier architecture, which allowed the development of independent features among developers.

3.1 Case Study 1: Matchall2

In the first case study, we monitored the development of a module for Matchall2, an industrial web-based application aimed at providing labeling facilities (namely a bookmarklet) that allow classifying and categorizing pictures and videos with custom tags. The project was developed from March 2013 to May 2013 for a total of nine weeks (4 sprints).

The team was composed of eight students participating in the course: two graduate students, four undergraduates, and two PhD students. One of the PhD students had a good level of knowledge of the project and all the relevant technologies. Therefore, he played the role of team coordinator/coach. A local entrepreneur played the role of the product owner.

3.2 Case Study 2: Serts

The Serts project aimed at implementing a semi-automatic tool, called SERTS (Software Engineering Research Tool Suite) with the goal of simplifying the analysis of data collected in software repositories such as Bugzilla, CVS, SVN, Git, and Jira. The project allowed navigating through versions and releases, storing the data in an internal database, so as to speed up subsequent analysis of the software, such as the calculation of metrics and the extraction of software graphs. The project was developed from September 2013 to November 2013 during a period of eight weeks (3 sprints).

The team was composed of six students, one undergraduate and five PhD students.

Like in the Matchall2 project, one of the PhD students had a good level of knowledge of the project and played the role of team coordinator/coach.

3.3 Commonalities and Differences to the Original Case Study

The original study design and procedure were strictly followed. The only difference of the new studies is related to the development processes applied and the development teams and the new measure investigated (IFPUG Function Points).

In the replicated studies, the process was plain Scrum instead of Moonlight Scrum. For this reason, developers in the original study worked in non-collocated spaces and during non-overlapping hours, whereas in these replications, the developers worked in the same space during the same timeframe.

A detailed comparison of the studies is available in Table 1.

Table 1. Context comparison among the three studies

	Original Case Study	Matchall2 Case Study	Serts Case Study
Development Process	Moonlight Scrum	Scrum	Scrum
Reporting	Online Forum	Kanban Board	Kanban Board
Developersôlocation	Distributed	Collocated	Collocated
Overlapping hours	No	Yes	Yes
Working hours/week	8	16	16
#developers	4	8	6
#weeks	18	9	8
#sprints	6	4	3
Project Type	Client-Server (web app.)	Client-Server (web app.)	Client-Server (desktop app.)

4 Results

In this section, we report the results of our two case studies. In both cases, we first analyzed the results for the functional measures (SiFP and IFPUG Function points). In order to understand if a different definition of SiFP can be adapted in our study to increase the accuracy of the effort estimation, we analyzed further correlations among the factors considered for the calculation of SiFP (#DF # iTF, and #oTF).

Then, we analyzed results of the correlations among the factors considered by our developers when they need to estimate a user story (GUI Components Added, Modified and Data Files).

Finally, we compared the accuracy of effort estimation predicted by our developers to the actual effort estimation.

Table 4, Table 5 and Table 6 in Appendix A report detailed results of the analysis.

4.1 Matchall2 Case Study Results

The Matchall2 project was composed of 81 user stories collected in 4 sprints in a total of 408 working hours. 75 user stories were related to the development of new features, five to refactoring, and one to bug fixing. All user stories had low GUI impact.

Since the number of user stories related to refactoring and bug fixing is not statistically relevant, and all user stories had the same GUI impact, we only analyzed the results for the new development user stories, without cluster results for story type or GUI impact.

After eliminating three outliers ï identified according to Cookôs distance [18], we reduced the number of user stories considered to 78. Table 2 shows descriptive statistics for the attributes analyzed in the Matchall2 case study.

The analysis of correlations between SiFP and effort does not provide any statistically significant result (see Fig. 1). A Pearson correlation coefficient of 0.121 (p-value =0.140 and r^2=0.015) was calculated for the 78 data pairs as presented in the scatter graph in Fig. 1. As a consequence of the low correlation, the accuracy is not acceptable (MMRE=66%, MdMRE=66%). The multivariate correlation analysis among the factors considered for the calculation of SiFP shows a similar trend as for the SiFP analysis, indicating that this information is also not significant for improving effort estimation in Scrum (see Table 5).

As expected, also the analysis of correlations among IFPUG Function Points and effort has a similar trend as these obtained with SiFP with no statistically significant results, as shown in Fig. 2 (Pearson=0.145, p-value =0.099 and r2=0.021). For this reason, the accuracy is also not acceptable (MMRE=116% and MdMRE=53%).

Taking into account the information considered by the developer to estimate the user stories, the univariate correlation between the sum of GUI Components Added and Modified and the effort, the correlation is very low and accuracy is not acceptable (Pearson= -0.017, p-value=0.440, r^2=0, MMRE=109% and MdMRE=76%). Moreover, also considering the results of the multivariate correlation among effort and GUI Components Added, GUI Components Modified, and DF, (Fig. 3) the results are still not statistically significant and accuracy is still not acceptable (MMRE=135% and MdMRE=93%).

Finally, as in our original study, we compared the accuracy of effort estimation predicted by our developers to the actual effort estimation (Fig. 4). The results show that expert-based effort estimation is much better than estimation predicted by means of functional size measurement, reporting an MMRE of 39% and an MdMRE of 25%.

Table 2. Descriptive statistics for the Matchall2 case study

Variable	Avg	Min	Max	Std. Dev.
Actual Effort (hours)	2.33	1	8	1.50
SiFP	6.11	4.6	7.20	0.64
IFPUG	13.64	6	60	17.23
GUI Components Added	0.55	0	3	0.69
GUI Components Modified	0.51	0	5	0.99
input Transactions (iTF)	1.32	0	8	2.84
output Transactions (oTF)	0.54	0	3	1.15
Data Files (DF)	0.87	0	3	1.26

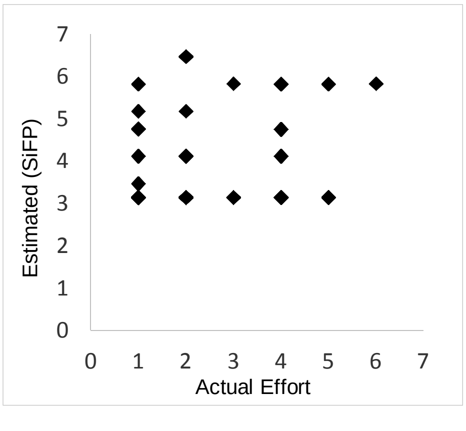

Fig. 1: Actual Effort vs Estimated Effort with SiFP

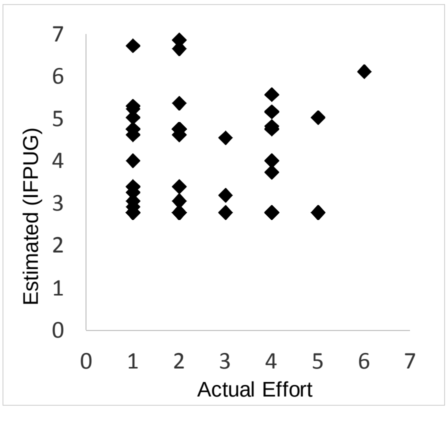

Fig. 2: Actual Effort vs Estimated Effort with IFPUG Function Points

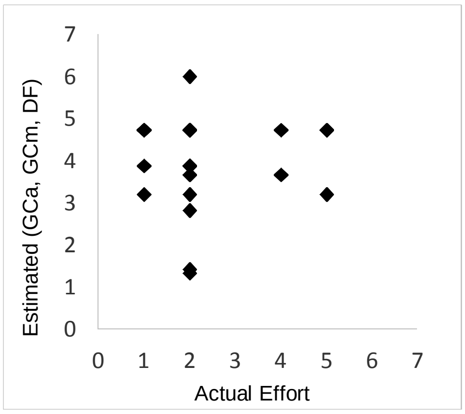

Fig. 3: GUI Components added, modified and DF

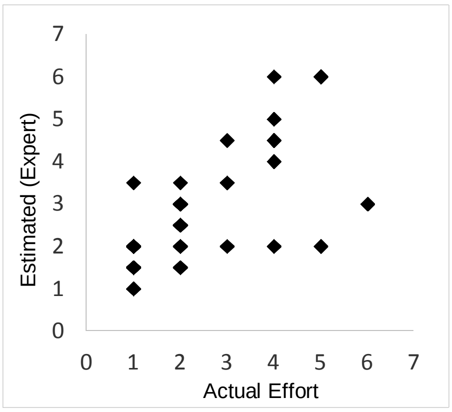

Fig. 4: Actual Effort vs Developersô estimated effort

4.2 Serts Case Study Results

The Serts project was composed of 25 user stories, collected in three sprints, for a total of 832 working hours. All stories are related to the development of new features. No stories were related to refactoring or bug fixing.

As for the Matchall2 project, GUI impact was always low for all user stories. Therefore, we do not cluster results for story type or GUI impact.

Table 3 reports descriptive statistics for the attributes analyzed in the Serts case study. In this case, no outliers have been identified by means of the Cookôs distance.

Table 3. Descriptive statistics for the Serts case study

Variable	Avg	Min	Max	Std. Dev.
Actual Effort	33.28	1	120.00	33.74
SiFP	4.93	4.6	5.60	0.51
IFPUG Function Points	37.38	4	67	16.03
GUI ComponentsAdded	1.92	1	6.00	2.13
GUI Components Modified	0.04	1	1.00	0.19
iTF	1.88	1	6.00	2.17
oTF	0.77	1	6.00	1.36
DF	2.65	1	4.00	1.62

As for the Matchall2 case study and for the original case study, the analysis of correlations between SiFP and effort did not provide any statistically significant results. A Pearson correlation coefficient of 0.411 (p-value =0.019 and R^2=0.169) was calculated for the 25 data pairs as presented in the scatter graph in Fig. 5. As a consequence of low correlation, even though the p-value is within an acceptable range, accuracy is still not acceptable (MMRE=120.00%, MdMRE=76.00%). As for the Matchall2 case study, we analyzed further correlations among the factors considered for the calculation of SiFP. The results show a similar trend as for the previous data (MMRE=93.00% and MdMRE=54.00%), indicating that this information is also not significant for improving effort estimation in Scrum (see Table 5).

Also in this case, as expected, the analysis of correlations between IFPUG Function Points and effort reports a similar trend to the one obtained with SiFP (Pearson =0.444, p-value=0.013 and r^2=0.197, MMRE=145.00% and MdMRE=119.00%) confirming that functional size measures are not suitable for supporting developers in predicting the effort of the user stories in Scrum.

Taking into account the information considered by the developer to estimate the user stories, the results confirm those obtained in the other case studies both when considering the univariate correlation between GUI Components Added and Modified and when considering the multivariate correlation among GUI Components Added, Modified and DF, reporting very low correlation and a not acceptable goodness of fit (see Table 6 for detailed results).

Finally, as in our original study, we compared the accuracy of effort estimation predicted by our developers to the actual effort estimation (Fig. 8). The results show again that expert-based effort estimation, even if it is not very accurate, is still better than estimation predicted by means of functional size measurement, reporting an MMRE of 52% and an MdMRE of 58%.

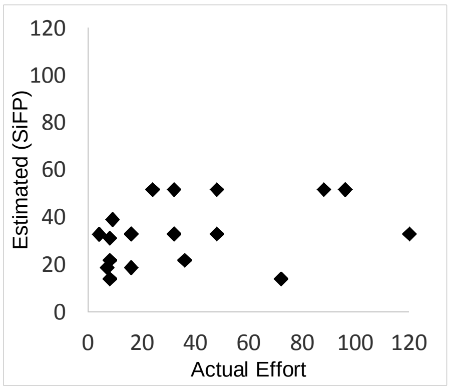

Fig. 5: Actual Effort vs Estimated Effort with SiFP

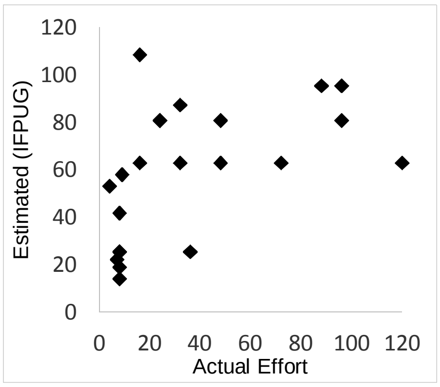

Fig. 6: Actual Effort vs Estimated Effort with IFPUG Function Points

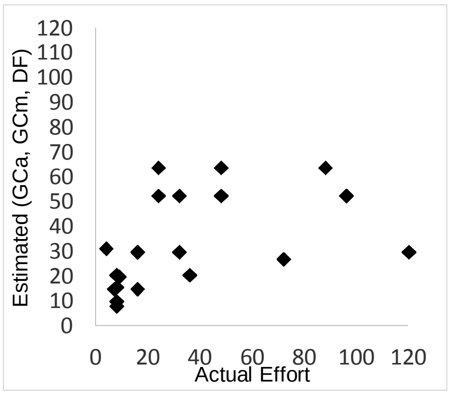

Fig. 7: GUI Components Added, Modified and DF

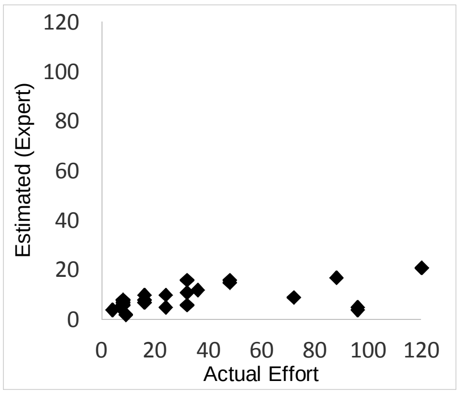

Fig. 8: Actual Effort vs Developersô Estimated Effort

5 Discussion

Based on the results obtained in the data analysis, we can answer our research questions.

As for the RQ1, on the extension of the results obtained to the original case study to plain Scrum, the analysis confirms the results obtained in our previous study. The main outcome of this replication is the confirmation of the low predictive power of SiFP in Scrum and that there are no correlations among the information considered by our developers and the actual effort. Moreover, also the analysis of the information needed to calculate SiFP do not help to improve the accuracy.

The analysis of the correlations among IFPUG Function Points (RQ2) and effort, show a similar trend of the one obtained with SiFP confirming that functional size measures are not suitable for predicting effort in either Moonlight Scrum or plain Scrum.

As side result of this work, we also confirm the accuracy of the conversion among SiFP and IFPUG reported in [17], reporting a MMRE= 27% and MdMRE=24%.

Results also confirm that, as in previous studies [5, 6, 7, 8, 13], developers usually underestimate the effort.

The original case study had been developed with a special version of Scrum, which is why we expected that the low prediction accuracy of functional measures was due to the nature of the project and not to the process.

Concerning the internal validity of the study, the developers were Masterѻs and PhD students with experience in software development ranging from two to five years. Moreover, the identified functional size measures are designed to estimate complete projects or components while in this case studies we applied it to Scrum user stories and not to the whole project.

As for external validity, this study focused on two Scrum processes, with part-time developers who work only two days per week. We expect some variations in applying the same approach to a full-time development team working on a plain Scrum process.

Regarding the reliability of this study, the results are not dependent on the subjects or on the application developed. We expect similar results for the replication of this study with other Scrum processs.

6 Conclusions

In this work, we replicated a case study with the goal of understanding if it is possible to introduce functional size metrics to the Scrum planning game.

With this study, we contribute to the body of knowledge by providing an empirical study on the investigation of functional size measures for effort estimation in agile processes, and in particular in Scrum.

To achieve this purpose, we first gave an overview of the previous study and then we described the difference with the case study we ran.

The results of our study confirm that functional size measures, and in particular SiFP and IFPUG Function points, do not help to improve estimation accuracy in Scrum. Moreover, even trying to re-compute the formula for the calculation of SiFP does not help to improve the accuracy of effort estimation.

Accuracy does not increase when considering other measures usually considered by developers when they evaluate the effort required to develop a user story.

As side result of this work, we also confirm the accuracy of the conversion among SiFP and IFPUG reported in [17], with an MMRE= 27% and MdMRE=24%.

Future works will include studies to better understand the information considered by the developers when they estimate user stories and the replication of this study in another industrial context.

Acknowledgments. This research is supported by Regione Autonoma della Sardegna (RAS), Regional Law No. 7-2007, project CRP-17938 LEAN 2.0.

References

1. V. Lenarduzzi, and D. Taibi: Can Functional Size Measure Improve Effort Estimation in SCRUM?. In ICSEA - International Conference on Software Engineering and Advances, Nice (France), 2014.
2. V. R. Basili, G. Caldiera, H.D. Rombach: ñThe goal question metric approach.ò Encyclopedia of software engineering, pp. 528ï 532, 1994.
3. J. Carver "Towards Reporting Guidelines for Experimental Replications: A Proposal." Proceedings of the 1st International Workshop on Replication in Empirical Software Engineering Research (RESER) [Held during ICSE 2010]. May 4, 2010. Cape Town, South Africa.
4. F. Shull, J. Carver, S. Vegas, N. Juristo: ñThe role of replications in empirical software engineeringò. Empirical software engineering 13(2) pp.211-218, 2008.
5. D. Jamieson, K. Vinsen, and G. Callender, ñAgile Procurement to Support Agile Software Developmentò, Proceedings of the 35th IEEE International Conference on Industrial Informatics, pp. 419-424, 2005.
6. T. Sulaiman, B. Barton, and T. Blackburn, ñAgileEVM - Earned Value Management in SCRUM Projectsò, Proceedings of AGILE Conference, pp. 10-16, 2006.
7. N. C. Haugen, ñAn empirical study of using planning poker for user story estimationò, Proceedings of AGILE Conference, pp. 9-34, 2006.
8. L. Cao. ñEstimating Agile Software Project Effort: An Empirical Studyò Americas Conference on Information Systems (AMCIS), paper 401, 2008
9. V. R. Basili, G. Caldiera, and H. D. Rombach, ñThe goal question metric approach.ò Encyclopedia of software engineering, pp. 528ï 532, 1994.
10. L. Buglione and A. Abran. ñImproving Estimations in Agile Projects: Issues and avenuesò Proceedings of the 4th Software Measurement European Forum (SMEF) Rome (Italy), 2007
11. D. Taibi, P. Diebold, and C. Lampasona. ñMoonlighting SCRUM: An Agile Method for Distributed Teams with Part-Time Developers Working during Non-Overlapping Hoursò Proceedings of the Eighth International Conference on Software Engineering (ICSEA), pp. 318-323, 2013
12. R. Meli, ñSimple Function Point: a new Functional Size Measurement Method fully compliant with IFPUG 4.xò, Software Measurement European Forum, 2011
13. V. Mahnic. óA Case Study on Agile Estimating and Planning using SCRUMò Americas Conference on Information Systems (AMCIS), pp 123-128, 2008
14. P. Diebold, L. Dieudonné, and D. Taibi, ñProcess Configuration Framework Toolò, Euromicro Conference on Software Engineering and Advanced Applications, 2014.
15. International Function Point Users Group. ñFunction Point Counting Practices Manualò, 2004
16. Willeke, Marian HH. "Agile in Academics: Applying Agile to Instructional Design." Agile Conference (AGILE), 2011. IEEE, 2011.

17. L. Lavazza, R. Meli, ñAn Evaluation of Simple Function Point as a Replacement of IFPUG Function Pointò, IWSM - Mensura 2014, Rotterdam, October 2014.
18. R. D. Cook, S. Weisberg, Residuals and Influence in Regression, Chapman and Hall, London, 1982.
19. K. Schwaber and J. Sutherland. The Scrum guide, 2001. Available online www.scrumguides.org.
20. H. Huijgens and R.V. Solingen. ñA replicated study on correlating agile team velocity measured in function and story points.ò In Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics (WETSoM 2014)

Appendix: Detailed Results

In this section we report detailed results of the correlation analysis carried out in both studies.

Table 4. Univariate Correlation Analysis Results

	Matchall2			Serts		
	SiFP	IFPUG	GUI (a+m)	SiFP	IFPUG	GUI (a+m)
Pearson	0.121	0.145	-0.017	0.411	0.444	0.422
p-value	0.140	0.099	0.440	0.019	0.013	0.016
r^2	0.015	0.021	0.000	0.169	0.197	0.178
MMRE	0.660	1.160	1.090	1.200	1.450	0.970
MdMRE	0.660	0.530	0.760	0.760	1.190	0.760

Table 5. Multivariate correlation between Actual Effort and iTF, oTF and DF

	Matchall2			Serts		
	iTF	oTF	DF	iTF	oTF	DF
Pearson	0.422	-0.063	-0.042	0.250	0.266	0.382
p-value	0.000	0.288	0.355	0.109	0.094	0.027
R^2		0.241			0.306	
MMRE		1.210			0.930	
MdMRE		0.580			0.540	

Table 6. Multivariate correlation between GUI Components Added, Modified and DF

	Matchall2			Serts		
	GUIa	GUIm	DF	GUIa	GUIm	DF
Pearson	0.114	-0.097	-0.042	0.438	-0.153	0.382
p-value	0.155	-0.423	0.033	0.013	0.228	0.027
R^2		0.018			0.265	
MMRE		1.350			0.950	
MdMRE		0.930			0.640	