

A Multiple-Case Study of Effort Estimation based on Use Case Points

Bente Anda, Hans Christian Benestad and Siw Elisabeth Hove

Simula Research Laboratory,
P.O. Box 134, NO-1325 Lysaker, Norway
Tel. +47 67828306
{bentea, benestad, siweh}@simula.no

Abstract

There is much interest in industry for the estimation of software development effort based on use cases, but little scientific evaluation of applying use cases in estimation has been reported. We investigated one particular method, the use case points method, in a multiple-case study. The Software Engineering Department at Simula issued a tender for a system, and 35 companies responded, with estimates ranging from 78 to 654 hours. We chose four companies to develop the system. They all implemented the same functionality, but their development processes varied, ranging from a light, mainly code-and-fix process, to a heavy process with much emphasis on analysis and design.

The use case points method estimated this project to 430 hours. This was equal to the actual effort spent on implementing the system by the company with the lightest development process. In our opinion, the results from this study may represent a basis for measuring size of the use cases and choosing productivity factor (hours per use case point) when estimating based on use cases. The three other companies spent 587, 829 and 943 hours respectively, showing that a heavier development process and more emphasis on non-functional requirements may increase effort by more than 100%.

Keywords Use cases, Estimation, Multiple Case Study

1 Introduction

Use cases are often used as input for estimating software development effort. Several studies show that a particular estimation method based on use cases, the use case points method, performs well early in a project [1-4,10,11]. This method requires little technical insight and little effort. The studies showed, however, a need for more scientific evaluation of the method.

We investigated the use case points method in a multiple case study in which 35 companies responded to a tender for a web-based system. Four companies were chosen to develop the system based on the same requirements specification. The Software Engineering Department at Simula Research Laboratory was the client for the system, the purpose of which was to handle the studies conducted by the department. The companies were chosen based on both business and research criteria. The chosen companies followed different development processes, ranging from a light, code-and-fix approach to a heavy process with much emphasis on front-end activities such as analysis, design and project management. Thus, we expected that the companies would span the field of likely development effort for implementing the specified functionality.

The results from this study show a strong correlation between number of transactions of the use cases and effort spent on realizing them, which supports the use of transactions as a suitable measure of the size of the functionality. The use case points method converts the number of transactions of each use case to a number of use case points, and suggests a minimum productivity factor of 7.5 hours per use case point when the non-functional requirements are trivial, and the development team is well-qualified for the task. Using this

factor, the use case points method gave an estimate of 430 hours. The estimates from the companies that bid for the system ranged from 78 to 654 hours, with a mean of 275 hours. The actual effort spent on the project ranged from 431 – 943 for the four companies. The minimum effort was spent by a company that did not have a defined development process and that had little focus on the design of the code. In our opinion, these results show that when the use cases are well-described, effort spent on implementing functionality may be predicted based on the use cases. The results further show that emphasis on development process and the design of the code may lead to an increase in effort of more than 100%.

The rest of this paper is organized as follows. Section 2 describes the use case points method and previous experiences with it. Section 3 describes the multiple-case study, i.e., the requirements and the four development projects. Section 4 describes the research method. Section 5 discusses transactions as a measure of size of the use cases. Section 6 compares the estimate produced with the use case points method with expert estimates and actual effort expended on the four systems. Section 7 discusses the scope and validity of the results from this study. Section 8 concludes and gives directions for future work.

2 The Use Case Points Method

This section describes the use case points method and previous experiences with it.

2.1 The Method

The use case points method is inspired by the function points method. It was proposed by Karner, who also validated it on three projects [8]. The assumptions behind the use case points method are 1) that the use cases can be used to measure the size of the functionality of

the system to be developed, and 2) that the amount of functionality is an important parameter when estimating software development effort.

The use case points method is described in detail in, for example, [12]. The method produces an estimate in three steps:

1. The actors in the use case model are categorized as *simple*, *average* or *complex*. A simple actor represents another system with a defined API and is given a weight of 1; an average actor is another system interacting through a protocol such as TCP/IP (weight = 2); and a complex actor may be a person interacting through a graphical user interface or a web page (weight = 3).
2. Correspondingly, the use cases are classified as *simple*, *average* or *complex* based on the number of transactions [8] or number of scenarios [14]. Simple use cases have 3 or fewer transactions or scenarios, and each such use case is assigned a weight of 5. Average use cases have 4–7 transactions or scenarios and are assigned a weight of 10. Complex use cases have 8 or more transactions or scenarios and are assigned a weight of 15¹. Analysis classes may also be taken into account, when determining the complexity of a use case [8].

The weights are summed up to give a total number of use case points for the system.

3. The productivity factor, the number of hours necessary to realize one use case point, is most often initially set to 20, but is adjusted depending on the following:

- 13 technical factors covering non-functional requirements on the system. These factors are converted into one technical complexity factor, TCF, for the projects.

¹ The original description of the use case points method does not provide an explanation of why these weights are used for actors and use cases.

The TCF is multiplied by the original productivity factor. The minimum value of TCF is 0.75 and is used if the non-functional requirements in the project are standard. The maximum value is 1.3 and is used when the non-functional requirements on the system are complex.

- 8 environmental factors covering the qualifications and motivation of the development team. These factors are converted into an environmental factor, EF, which is also multiplied by the productivity factor. The minimum value of EF is 0.5 and is used when the development team is well-qualified and motivated for the task. The maximum value is 1.5.

The minimum productivity factor suggested is thus 7.5 hours per use case point.

The productivity factor is multiplied by the number of use case points to produce an estimate.

2.2 Experiences with the Use Case Points Method

Our research group has conducted several studies in which the use case points method has been evaluated and adapted to different industrial projects [1-3,10,11]. Tables 1 and 2 offer an overview of these projects. The studies investigated the assessment of the size of the use cases, the productivity factor and which activities of a project may be estimated based on use cases. Table 1 characterizes the projects, and Table 2 shows the results of applying the use case points method and compares this with expert estimate and actual effort.

The number of transactions of the use cases was used to calculate use case points in most of these projects. In Study 5, the use cases were not detailed out with transactions, in that study the project manager assessed the complexity of each use case.

Table 1. Characteristics of projects evaluating the use case points method

Study	Ref.	Application domain	Prog. language	Team size	No. of use cases	Non-functional requirements
1	[3]	Finance	Java	6	9	End-user efficiency
2	[3]	CRM for a bank	Java	6	16	End-user efficiency, usability
3	[3]	Banking	Java	5	11	As above
4	[1]	Internet shopping	Java, C++ and Smalltalk	8	22	End-user efficiency, usability
5	[11]	Banking	Java	6	63	Short response time, usability and changeability
6	[11]	Real-time system	C++	14	63	Short response time, changeability and security
7	[10]	Telecom	C, Erlang, Perl, Java	Appr. 20 teams of 10 people	288/77*	Very high demands on response time, changeability and security
8	[10]	As above	As above	As above	254/108*	As above

*This study was on estimating incremental development. The first number is the total number of use cases for the system, and the second is the number of use cases that were modified in this release.

Table 2. Estimates and actual effort of the projects

Study	UCP estimate	Expert estimate	Actual Effort	Productivity (hours per UCP)	Activities estimated	Deviation use case estimate
1	2550	2730	3670	29.3	All	-31%
2	2730	2340	2860	19.4	All	-5%
3	2080	2100	2740	27.8	All	-24%
4	4086	2772	3360	15,3?	All	+22%
5	10831	7000	10043	16,4	Not proj. management	+8%
6	14965	12600	13933	21,9	As above	+7%
7	*	*	Appr. 100 person years	36	Not system test	-21%
8	*	*	Appr. 100 person years	36	As above	-17%

*Exact figures cannot be given for reasons of confidentiality

We have not found other studies than the above that have evaluated the use case points method when estimating industrial projects. Use case points have, however, been found useful to measure system size [4]. Four estimation tools, three of them commercially available, implement the use case points method or variants of it [9, 15-17]. A relationship between external use cases and source lines of code (SLOC) was found in projects with students [6]. An approximate number of hours to implement a use case, depending on the size of system, was suggested in [13]. The challenge of measuring the size of use cases that hide complex business logic, and suggestions for improving use case based estimation to handle this issue, can be found in [14]. Use cases are also often used informally in estimating effort, as described for example in [2].

3 The Development Projects

The system to be developed was a web-based system for handling the studies conducted by the Software Engineering (SE) department at Simula Research Laboratory. Thirty-five Norwegian development companies responded to a tender. The tender included a requirements specification with 3 actors and 9 use cases. The use cases were described at user goal level with transactions [6], and the business logic was simple. An example of a use case is given in Figure 1. The non-functional requirements were inherent in the technology that was used to develop the system, and the requirements specification did not include specific requirements on the quality of the code in terms of, for example, maintainability or reusability.

Create or edit study

Preconditions:

1. Login has been executed, user is authenticated, and authorized as study administrator
2. An existing study has been selected using the use case “Select and report study”, or a command has been given to create a new study

Steps:

1. The user inserts study information. Study responsible and publications may be selected from lists. Study material files may be attached to the study by uploading files accessible from the user's machine.
2. The user selects a command to save the study to the database
3. The system validates the data, writes the information to the database, and displays a message that the study was stored properly.

Exception

- 3.1 If validation fails, the system displays a message containing information that will help the user to complete registration. He may then start at step 2.

Post conditions:

1. The study has been saved to the database. Time and user has been logged.

Figure 1. Example of a use case

The bids included a firm price, ranging from 21000 NOK to 559500 NOK, an effort estimate, ranging from 78 – 654 hours with a mean of 275 hours, and a description of the company’s development process. The bidding process is described in more detail in [7]. The distribution of the estimates in hours is shown in Figure 2.

Four companies (henceforth, A, B, C and D) were chosen to develop individual systems. The companies were chosen based on both business and research criteria. The business criteria were price, experience with similar solutions, experience with the programming environment (Java), size of the company (in order to enable similar teams) and apparent understanding of the requirements. We wanted companies that were very likely to complete the project in a satisfactory way, as failure to do that would invalidate our research.

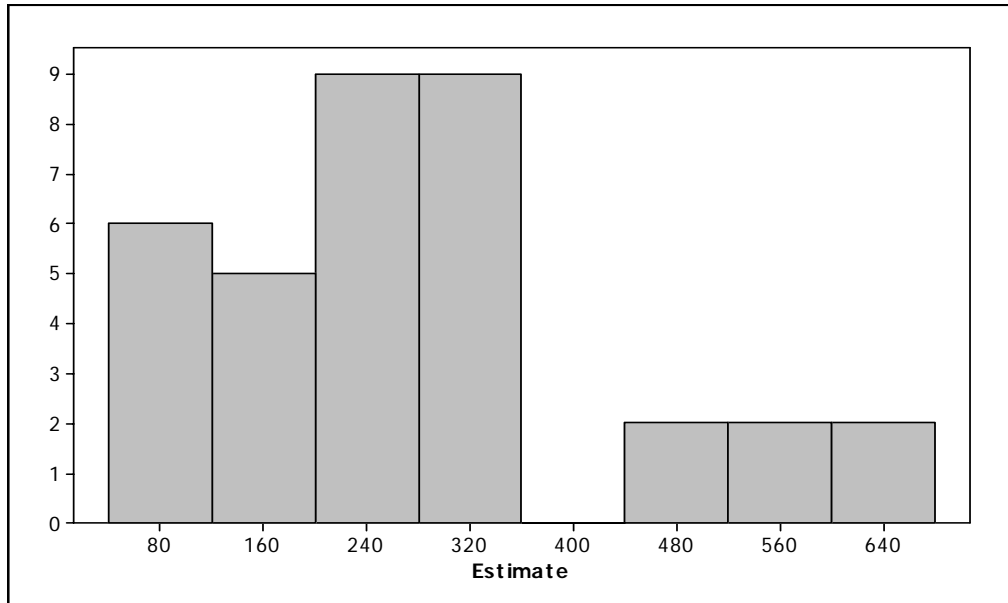


Figure 2. Expert estimates for the system

The four companies developed the systems in parallel and used from nine to 13 weeks. Before acceptance, the systems were thoroughly tested by the client (us). The functionality developed by the different companies was very similar. It was emphasized that functionality additional to that specified in the requirements specification was not desired².

4 Research Method

The research method was a multiple-case study with four development projects developing the same system in parallel.

4.1 Organization of Research and Development Projects

In this project, the SE department at Simula Research Laboratory was both researcher and client on the same project. This required a separation of concerns, and so the people involved

² One of the final systems can be found at www.simula.no/DES

in the project were organized in two separate teams. One team had the role of the client and consisted of a project manager and user representative. They were both employed by SE at the start of the project and had long experience from IT companies, but no research experience. The other team was responsible for the research and consisted of a researcher and a research assistant. In addition, an experienced, external consultant worked on an hourly basis to ensure that SE behaved realistically in the role of client, and that the development projects were affected as little as possible by the research.

The requirements specification had been developed by three people not directly involved in the development project.

4.2 Choice of Companies

The companies' price and development process, as described in the bids, were used in the selection of companies. We chose four companies that differed on these issues, to ensure that their proposals would span the field of likely development processes for such a project. It was made clear to the companies that although the prices were fixed, we would not use it against them if they expended less than the estimated effort. That is, they should not feel obliged to expend all the estimated effort. Table 3 shows the price, estimate in hours and development processes of the four companies.

Table 3. Characteristics of the development projects

Company	Price	Estimate	Development process
A	160000 NOK	220 hours	A rather light, mainly code-and-fix process (focus on visual design).
B	360000 NOK	341 hours	A heavy process with focus on analysis and design
C	70000 NOK	100 hours	Very light code-and-fix process
D	450000 NOK	650 hours	A heavy process with focus on analysis, design and project management.

The following aspects of the development projects were kept as equal as possible:

- The *development teams* were similar in terms of size and qualifications to avoid large differences in effort due to these factors. All the teams consisted of one project manager and two developers. Some of the companies also used additional resources when needed. The developers all had at least three years' experience with Java development.
- All companies used Java as the *development language*, but they used different development tools. They were asked not to use pre-existing software components besides standard open-source components for the java platform (like Struts and log4j), and company-specific libraries.
- The *development environment*, in the sense of communication and interaction with the client. Most of the communication between SE's client team and the development teams was done with the issue tracking system BugZero [18]. The client team made an effort to behave consistently towards the different companies.

All the companies and the people involved knew that they were participating in a research project, and agreed to it. They were paid extra for participating in research activities such as interviews and preparing and sending time sheets. Effort expended on these activities was recorded separately.

4.3 Data Collection

The following data was collected about the development processes and the effort spent on the project:

- Bids: The companies gave brief descriptions of their development process with their bids.

- Contract meeting: Contract meetings were held with the four companies; these also provided information about their development processes.
- Curricula vitae: SE received CVs for all the team members.
- Time sheets: The team members kept daily records of effort on use cases and activity during the development project. They were asked to record effort on use cases where appropriate, and instructed that effort on several use cases should be distributed over the appropriate use cases. The time sheets were sent to the researchers daily, and the research assistant checked that the recorded time was realistic. In case of anomalies, the team members were asked how effort had been expended.
- Interviews: The team members were interviewed about their qualifications, their development process, and their priorities with respect to non-functional requirements. The interviews were also used to validate the time sheets.

5 Measuring the Size and Complexity of Use Cases

The number of transactions of the use cases is a much used measure of the size of the use cases, and the use case points method uses the number of transactions as a basis for assigning a number of use case points to each use case. Nevertheless, there has been little evaluation of how well the number of transactions correlates with effort expended on implementing the use cases.

Table 4 gives an overview of the use cases in this system and the number of transactions in each. The effort recorded on each use case is the total effort required to realize it, including analysis and design, implementation, testing and error correction.

Table 4. Use case transactions and effort in hours

Use case	Trans.	Effort A	Effort B	Effort C	Effort D
1. Delete study	1	19	37	9	18
2. Create aggregated study report	1	22	52	5	27
3. Manage system properties	1	23	29	5	14
4. Grant administrative privileges	1	31	48	3	32
5. Manage system data	2	21	10	22	22
6. Login	2	37	36	22	35
7. Create csv-report	3	57	143	16	116
8. Select and report study	3	67	136	40	82
9. Create or edit study	4	157	148	99	147
Total on use case		434	639	221	493
Effort not assigned to individual use cases		153	304	210	336s

There was a strong correlation between the effort expended on use cases and the number of transactions of the use cases (Pearson correlation = 0,729, P-Value = 0,000). This supports claims that transactions are a good measure of the size of use cases, at least in situations in which the use cases are detailed and describe the business logic.

Company C recorded less effort than the other companies on use cases. We believe that this was because they worked in a less structured manner and consequently were less conscious about exactly which use case they were working on. There were variations in how much of the total effort the companies spent on each of the use cases. In our opinion, these variations are mainly due to the following:

1. Use cases that were implemented early may require more effort than use cases that were implemented later in the development project, and companies varied as to which use cases they implemented first.
2. The amount of user interaction varies among the use cases, and there were also some differences among the companies with respect to their emphasis on visual design.

6 Estimating the System with Use Case Points

Using the number of transactions of each use case as a basis for measuring the number of use case points, the functionality of this system corresponds to 57 use case points. Furthermore, we use the minimum productivity factor of 7.5 hours per use case point, since the non-functional requirements are trivial and the team is well qualified³. Figure 3 shows how the system is estimated.

1 simple actor with weight 1 = 1,
2 complex actors with weight 3 = 6
8 simple use cases with weight 5 = 40
1 average use case with weight 10 = 10 = 57 use case points
57 * 7.5 (productivity factor) = 430 hours

Figure 3. Estimating effort for realizing the system

The use case points method gives an estimate of 430 hours, meaning that for a qualified development team it should be possible to produce an acceptable system realizing these nine use cases in 430 hours. Table 5 shows the expert estimate, productivity (in hours per use case point) and actual effort for the four companies that implemented the system. Actual effort includes effort on all activities in the project.

Table 5. Expert estimate and actual effort for the four companies.

Company	Expert estimate	Productivity	Actual effort
A	250 hours	10.3 hours pr. UCP	587 hours
B	341 hours	16.5 hours pr. UCP	943 hours
C	100 hours	7.5 hours pr. UCP	431 hours
D	650 hours	14.5 hours pr. UCP	829 hours

³ 55 hours per use case for simple systems have been proposed in [13], implying a productivity factor for each use case point similar to the one we have applied here.

Table 5 shows that 430 hours is equal to the minimum effort for implementing the system, and that it is closer to actual effort than is the expert estimate for all companies except company D. The use case points estimate is also closer to actual effort than were most of the expert estimates from the companies that bid for the system (Figure 2).

Table 5 also shows a large variation in actual effort among the companies, even though they developed the same functionality, used the same programming language and had similar qualifications. The companies followed different development processes, with varying emphasis on analysis, design and project management. The differences in development process entailed differences with respect to emphasis on the design of the system. Before delivery for acceptance test the team members were asked in interviews how they considered the quality of their own code in terms of maintainability and reusability. Their responses are shown in Table 6.

Table 6. Subjective assessment of code quality

Company	Opinion on code quality
A	The quality is acceptable
B	The quality is acceptable
C	It is too costly to plan for changes, so we have not focused on maintainability or reusability
D	We have focused on code quality and expect this to be good.

In our opinion, Tables 5 and 6 show that the use case points estimate of 430 hours, obtained with the minimum productivity factor of 7.5 hours per use case point, was a good estimate for an acceptable solution that realized the use cases and thus satisfied the functional requirements to the system. However, the results also show that an increased focus on the design of the code led to an increase in the productivity factor of almost 50%, and that a heavier development process led to an increase of another 50%.

7 Scope and Validity of the Results

This section discusses the scope and the validity of the results from this study.

7.1 The Scope of the Results

In this study the use cases were described with much detail and the business logic was simple.

The development projects were small, the non-functional requirements were trivial and the development teams were all sufficiently qualified for the task. The members of the teams that developed the systems were asked in interviews whether they found this development project realistic, that is, to what extent it resembled other projects with which they had experience. They all considered the project realistic, but some thought it to be most similar to developing a subsystem of a larger project. We therefore expect the results from this study to be applicable in similar industrial development projects, and also that they could serve as a basis for applying use case points in larger projects.

The use case points method may be particularly useful in situations where little technical expertise and experience from similar projects is available. An example of such a situation is that of a software client requesting a new system and facing the challenges of 1) adjusting the desired functionality to the available budget, and 2) choosing a software contractor requesting a realistic price.

7.2 The Validity of the Results

The 35 expert estimates with which the use case points estimate was compared in this study were made in the context of a real bidding process in which the companies bid for the development of a real system.

The use case points estimate was also compared with actual effort from four companies. We believe that the effort data with which the use case points estimate is compared is of high quality. Effort was recorded daily by the team members in detailed time sheets, and was validated each week by a research assistant. Most of the teams recorded effort in more detail on this project than they usually did. They were, therefore, asked in interviews about their experiences with recording effort at this level of detail. Most of the team members had not found this problematic, but some felt that it was difficult to be accurate at such a high level of detail.

There are no fixed rules or guidelines for describing use cases. Consequently, the use cases for this system can be described in different ways, which could result in slightly different numbers of transactions and hence of use case points and estimate. The exact match between the use case points estimate and the minimum effort spent on developing the system was therefore, in our opinion, a lucky coincidence. Nevertheless, a standard use case format was applied based on [6], and the description and structuring of the use cases for the purpose of estimating was performed independently of the estimation by a person who had little familiarity with the use case points method. Thus, we consider that the results from this study provide evidence for the use case points method as a promising method for producing early estimates.

In the comparison of the effort on the four systems, we disregarded differences in usability, but there were no large differences between the systems.

8 Conclusions and Future Work

We have investigated the application of use cases in estimating software development effort in a multiple-case study, in which 35 companies bid for developing a system. Four of those companies were chosen to actually develop a system based on the same requirements specification. The teams from the four companies had very similar qualifications, and the functionality of the four resulting systems was almost equivalent. The teams followed different development processes and placed different emphasis on the design of the code. We estimated the necessary effort for developing the system using a particular method for estimating based on use cases; the use case points method.

The results from this study support previous claims that the use case points method can support early estimation of software development effort, and the results may provide a basis for applying use cases in estimation. The results from this multiple-case study also show that a heavier development process, as well as more emphasis on the design of the code, may increase effort by more than 100%. Many more studies are, however, needed to determine how use cases should be described to provide a basis for estimating effort and on the impact of requirements on the design of the code on effort.

The following activities are planned for future work:

- The code of the four systems will be analyzed to identify more in detail how they differ.
- The development processes followed by the four companies, and the effects of these processes, will be investigated more in detail. Interviews, time sheets and code will be used to describe the development processes followed by the four companies. Effort data

from both development teams and Simula's client team, as well as the code, will be used to investigate the effects of the different processes.

- The use case points method will be applied to more projects to provide improved guidelines for how to measure size and assess productivity in different kinds of project.

Acknowledgements

We gratefully acknowledge the following, without whom a research project of this scale would not have been possible: the companies for their participation and willingness to contribute to our research; Dag Sjøberg for initiating this work and obtaining the necessary funding, as well as for comments on this paper; Magne Jørgensen for laying the foundations for this work through his own research on the bidding process; Stein Grimstad for his job as Simula's user representative and tester for the system; Vigdis By Kampenes for her help in interviewing the development teams and testing the final systems; Gunnar Carelius for his help in choosing development companies and for technical support during the project, and Glen Farley for his help in choosing development companies and for his advice during the project.

References

- [1] Anda, B. Comparing Use Case based Estimates with Expert Estimates. The 2002 Conference on Empirical Assessment in Software Engineering (EASE), Keele, United Kingdom, April 8-10, 2002.
- [2] Anda, B., Angelvik, E. and Ribu, K. Improving Estimation Practices by Applying Use Case Models. The 4th International Conference on Product Focused Software Process Improvement (PROFES), Finland, December 9-11, pp. 383-397, LNCS 2559, Springer-Verlag, 2002.
- [3] Anda, B., Dreiem, H., Sjøberg, D.I.K., and Jørgensen, M. Estimating Software Development Effort Based on Use Cases – Experiences from Industry. The 4th

- International Conference on the Unified Modeling Language, Concepts, and Tools (UML), Canada, October 1-5, pp. 487-502, LNCS 2185, Springer-Verlag, 2001.
- [4] Arnold, P. and Pedross, P. Software Size Measurement and Productivity Rating in a Large-Scale Software Development Department. The 20th International Conference on Software Engineering (ICSE), Kyoto, Japan, April 19-25, 1998, pp. 490-493.
 - [5] Chen, Y., and Boehm, B. An Empirical Study of eServices Product UML Sizing Metrics. The 2004 International Symposium on Empirical Software Engineering (ISESE), Redondo Beach, California, August 19-20, 2004, pp. 199-206.
 - [6] Cockburn, A. *Writing Effective Use Cases*. Addison-Wesley, 2000.
 - [7] Jørgensen, M. and Carelius, G.J. An Empirical Study of Software Project Bidding. *IEEE Transactions of Software Engineering*, 30(12), pp.953-969, 2004.
 - [8] Karner, G. Resource Estimation for Objectory Projects. Objective Systems SF AB. 17. september 1993.
 - [9] Kusumoto, S., Matukawa, F., Inoue, K., Hanabusa, S. and Maegawa, Y. Estimating Effort by Use Case Points: Method, Tool and Case Study. The 10th International Symposium on Software Metrics, Illinois, September 14-16, 2004, pp. 292-299.
 - [10] Mohagheghi, P., Anda, B. and Conradi, C. Effort Estimation of Use Cases for Incremental Large-Scale Software Development, Accepted for publication at the 27th International Conference on Software Engineering (ICSE), 2005.
 - [11] Ribu, K. Estimating Object-Oriented Software Projects with Use Cases. MSc thesis, November 2001.
 - [12] Schneider, G. and Winters Jason P. *Applying Use Cases – A practical guide. 2nd ed.* Addison-Wesley, 2001.
 - [13] Smith, J. The Estimation of Effort Based on Use Case. Rational Software white paper. TP-171 10/99.
 - [14] Vinsen, K., Jamieson, D. and Callender, G. Use Case Estimation – The Devil is in the Detail. The 12th International Requirements Engineering Conference (RE), Kyoto, Japan, September 6-10, 2004, pp. 10-15.
 - [15] www.du vessa.com
 - [16] www.tassc-solutions.com/
 - [17] www.sparxsystems.com/
 - [18] <http://www.websina.com/bugzero/>