

Estimates – How Accurate Are They



Executive Summary

Using the data from completed software projects,¹ this paper will provide you with information about how people have gone about estimating their software projects and how well they did it.

Submitters of software project data to the ISBSG provide details of the estimation techniques they use in their projects, as well as both estimated and actual project statistics.

Four different estimates are sought: project duration, effort, cost and size.

There are now over 400 projects in the ISBSG Data Repository for which estimation data is available. This paper presents an analysis of those projects. It summarises the estimation techniques used, the accuracy of the estimates, and relationships between estimates.

In most respects these projects are typical of the full set of projects in the ISBSG Data Repository, so the value of the findings presented here is the same as the value of the ISBSG Data Repository as a whole. The ISBSG believes that the Repository represents the best part of the software industry because projects in the Repository are complete (and therefore more successful than many projects), and come from organisations with sufficient process maturity to include a software metrics program.

 When using the ISBSG equations and data, be aware that they probably reflect the best 25% of the industry.

The findings presented here pertain to this particular collection of projects. It is possible that some findings are not generally applicable. Nevertheless, most observations are in accordance with intuition. It can be expected that this analysis provides a good picture of the state of estimation in good software projects.

¹ Refer to Appendix for details of the project demographics of the data used for the analysis in this paper.

Table of contents

Executive Summary	1
The Project Details	3
A General Picture: Accuracy of Estimates	4
Estimation techniques	5
Individual estimates	6
Conclusion.....	11
APPENDIX- Software Project Sample Demographics used	13
Size and other key project indicators:	14
The International Software Benchmarking Standards Group (ISBSG)	15
Please help us to collect data	15
Partners.....	15

The Project Details

In the ISBSG Repository there are 439 projects for which some form of data about estimation is available. These projects represent a broad cross-section of the software industry.

- The profile of these projects is similar to the total collection of projects in the ISBSG Data Repository. This is important, because it means you can expect that the results from this paper apply just as well to the whole repository.
- These projects differ in only one respect: PC projects are more common in these 439 projects than in the repository, at the expense of multi-platform projects. (There is little difference in mainframe and midrange platforms.) There are, however, no significant differences in size, effort, duration, team size, or Project Delivery Rate (PDR); and no notable difference in where the projects come from, the languages they use, etc.

A General Picture: Accuracy of Estimates²

The two most important estimates are project effort and duration. (Cost is dominated by effort, and size is mainly relevant as an input to estimating effort and duration.)

From the projects for which we know both the estimated and actual effort plus estimated and actual project delivery date, we can make the following observations:

- 25% met both estimates; 11% were better than expected (delivered on time or early, using less than 90% of the estimated effort); 14% were as expected (delivered on time, and actual effort within 10% of the estimate).
- 39% met one estimate but not the other. It is more common to be early or on time but to use more effort than estimated (25%), than to be late but to estimate effort accurately (14%).
- 36% met neither estimate. They were delivered late, and underestimated effort by more than 10%.

Good estimates tended to be made for smaller projects with short durations, usually developed for single users. Poor estimates, particularly of delivery date, tend to be made for new developments and for client-server projects that support a large number of users.

Large and small errors in estimating effort correspond closely to large and small errors in estimating cost. It is clear that cost and effort are strongly related. This supports the intuition that cost is determined mainly by effort, since the major resource consumed by a software project is human effort.

Interestingly there is no relationship at all between project delivery rate and the accuracy of the estimates. You might expect that projects delivered ahead of schedule, or with less effort than estimated, would have low (i.e. good) PDR, but there is no indication of this at all. Projects delivered in line with their estimates had a wide range of PDRs.

 There is no relationship between Project Delivery Rate, (productivity), and the accuracy of the estimate.

² In this analysis, an estimate is considered to be 'accurate' if it is within 10% of the actual value. This threshold was chosen arbitrarily, although choosing a fairly low value seems appropriate because it allows projects to be grouped with fine granularity. The choice of threshold clearly influences the percentages of estimates that are considered to be accurate. Readers should bear this in mind when quoting percentages from this paper.

Estimation techniques

Size estimates are usually based on a data model (often involving a CASE tool); a functional specification; or analogy with a previous project.

For delivery date, effort, and cost:

- 38% of the projects only use work breakdown estimation
- 16% of the projects only use FP-based estimation
- 27% of the projects use both
- 18% of the projects use neither, instead using life cycle models or tools
- “management directive” pre-determines the delivery date in 17% of the projects.

If a size estimate is available, it is almost always used to help estimate delivery date, effort, and/or cost. If no size estimate is available, work breakdown estimation or management directive generally determines the estimate delivery date, effort, and cost.

There is no association between other project attributes (organisation type, development type, etc) and the estimation techniques used.

In most cases, there is little relationship between which estimating techniques are used and how accurate the estimates are. In general, functional size-based techniques slightly out-perform work breakdown techniques.

 Projects using functional size estimation techniques are producing the most accurate estimates.

Individual estimates

In this section, we analyse each of the four types of estimate and consider the following questions: how often are projects underestimated and overestimated; what over-runs or under-runs are typical; and what types of projects are likely to be overestimated or underestimated, etc.

The samples analysed here are smaller than the full set of 439 projects. Some projects contained information on the estimation techniques used, but did not give the estimates themselves. Of the remaining projects that provide at least one estimate, few give all four estimates.

Effort estimates

Data is available for 200 projects.

Effort is usually underestimated:

- 19% (38 projects) overestimated effort by at least 10%
- 23.5% (47 projects) estimated effort to within 10% of the actual value
- 57.5% (115 projects) underestimated effort by at least 10%.

Of the four key project attributes, effort is the worst estimated. For the other three, at least half of the projects are estimated accurately. For effort, less than one quarter of projects are estimated accurately.

The overestimates occurred in small projects that are only about half the size of the Repository average. Actual effort averaged 70% of estimated effort.

Looking at the 115 underestimated projects, 12 had errors of 200% or more (that is, the actual effort was more than triple the estimate). Another 14 had errors of 100% or more, and a further 29 had errors of 50% or more. In other words, over one quarter of all projects underestimated effort by at least 50%. On average, the actual effort was about double the estimate (mean underestimate was 100%; median underestimate was 50%). The largest error saw effort underestimated by a factor of over 20 times.

 For the projects with effort underestimated, on average the actual effort was about double the estimate.

Across the whole 200 projects, the mean error is an underestimate of 60%. The median is an underestimate of 16%.

For the most part, large errors in estimated effort are accompanied by similarly large errors in estimated cost.

There is no relationship between which techniques were used to estimate effort and the accuracy of the estimate. There are also no patterns in terms of which types of projects are estimated better or worse than other types.

Delivery Date & Project Duration

Estimated and actual delivery date and project duration are known for 222 projects.

Delivery date is estimated relatively well; more than half of the projects were delivered early or on time:

- 4.5% (10 projects) were delivered early;
- 51.5% (114 projects) were delivered on time
- 44% (98 projects) were delivered late.

The projects delivered early were relatively long-running ones. The average for the whole Repository is about 6 months; these 10 projects averaged an estimate of 9 months, but actually took about 7 months. Thus, if a project was delivered early it was by about 22% on average. Early delivery was achieved in most cases at the expense of extra effort.

In the 114 projects delivered on time, there is a wide spread of durations, sizes, and other project characteristics. No patterns can be seen.

Of the 98 projects delivered late, 13 were one month late, 21 were two months late, 22 were three months late, 25 were 4 to 6 months late, and 17 were more than six months late. Four were a year or more late, with the worst over-run being 17 months. Late projects averaged 100% over-run in duration. Once a project is more than one month late, the average over-run is 113%.

 Late projects average an over-run of 100% in duration.

Accuracy of the estimates varies according to how long the project was planned to take in the first place.

- Projects that were planned to take a year or more were generally delivered on time. If they were late it was by up to 3 months, but for most (18 out of 27) the actual duration was within 10% of the estimate. The average over-run was less than one month.
- Projects that were planned to take 6 to 12 months did less well. The actual duration was within 10% of the estimate in half of them. 15% of the projects underestimated duration by 6 months or more. Overall, the average over-run was nearly 2 months, or 20%.
- Projects that were planned to take 4 to 6 months did somewhat better, although the worst underestimates came from this group. (Three of these projects were delivered more than a year late.) Again, half of the projects estimated duration to within 10% of the actual value. 10% underestimated duration by 6 months or more. The average over-run was 1.6 months, or 32%. (The percentage is bigger, even though the number of months is smaller, because the average duration is shorter.)
- Projects that were planned to take up to 3 months have the biggest errors in percentage terms (because the planned duration is shortest. For example, a project that was planned to take one month and was delivered 3 months late has a 300% error), but the actual values are not so bad. The average over-run was 1.3 months.

If a size estimate is available, it is almost always used as an input for estimating duration; either directly, or as an input to an estimating tool. If no size estimate is available, duration is normally determined by management directive or by work breakdown techniques.

The estimating techniques have some impact on the accuracy of the estimated duration. Estimates based on a software life cycle model do poorly, underestimating duration by 68% on average. Work breakdown techniques underestimate duration by 39% on average, and are exactly right 45% of the time. FP-based estimates are either very good or very bad. Just over half are exactly correct, but 20% are wrong by 100% or more; the mean error is a 45% underestimate. If functional size measurement is used in conjunction with an estimating tool, the estimates of duration improve.

 If functional size measurement is used in conjunction with an estimating tool, the estimates of duration improve.

The average error drops to a 40% underestimate, and over 60% of estimates are now exactly right (so the average is the same as work breakdown techniques, but more of the individual estimates are accurate). In the projects whose delivery date was determined by management directive, most were delivered on time or 1 month late; only 10% were delivered more than 2 months late. Projects whose delivery date is determined by legal requirements or client directive do best, with an average error of only about 10%.

In summary, duration tends to be estimated fairly well, particular for projects planned to take 12 months or more. If duration is overestimated, it is probably by about 20%. If duration is underestimated, the average error is 100%.

Cost estimates

Cost tends to be estimated more accurately than effort. Although underestimates are again more common and larger than overestimates, really bad errors are rarer and most errors are smaller than is seen with estimates of effort.

Estimated and actual costs can be analysed for 86 projects:

- 16% (14 projects) overestimated cost by at least 10%
- 49% (42 projects) estimated cost accurately, to within 10%
- 35% (30 projects) underestimated cost by at least 10%.

The proportional size of these groups is about 1 (overestimate) to 3 (accurate) to 2 (underestimate).

When cost is overestimated, on average the actual cost is 76% of the estimate. Most of these are fairly small projects, of fewer than 200 FP. No other patterns are evident.

When cost is underestimated, for five projects the errors are huge (300% to 700%), and another five exceed 50%. When the five huge errors are excluded, the average error is an underestimate of 32%; the overall median error is an underestimate of 25%.

 For projects with cost underestimated, the median³ error is an underestimate of 25%.

A very clear trend can be seen when estimation techniques are considered. For projects using just work breakdown techniques, 40% are accurate to within 10%, a further 10% are accurate to within 20%, and half of them mis-estimate cost by 20% or more; underestimates are three times as common as overestimates, and tend to be slightly worse. For projects using functional size-based techniques, 70% are accurate to within 10%, a further 20% are accurate to within 20%, and only 10% mis-estimate cost by 20% or more. FP-based techniques are more accurate, more of the time.

 When functional size-based techniques are used for a cost estimate, the estimate is within 20% of the actual cost 90% of the time.

No other patterns can be seen between project characteristics and the accuracy of cost estimates. Interestingly, this time it is not the case that overestimated projects are all small projects, or that the largest or longest projects have the largest underestimates.

When both cost and effort are known, the association between them is strong. If the projects are ranked in order of effort error (biggest underestimate down to biggest overestimate), and then ranked in order of cost error, the correlation between the ranks is very high, at 0.82. Big errors in estimating effort go with big errors in estimating cost. This supports the intuition that effort is the major determinant of cost.

In summary, for half of the projects the cost estimate is accurate. Of the rest, underestimates are about twice as likely to occur as overestimates, and the error is likely to be slightly worse. Cost estimates produced using functional size-based techniques are more accurate than estimates produced from a task breakdown, and are within 20% of the actual cost 90% of the time.

Size estimates

Under 'Estimation techniques' we saw that 50% of projects used functional size techniques as part of estimating duration, effort, or cost. They may have used an estimate of size, or a size value counted carefully from the project specification; the latter is perhaps most likely.

This section considers a different question: if size in function points was estimated early in a project, how does the estimate compare to the final 'properly counted' size?⁴

³ The median is the middle value of the sample group. It is more useful than the average in this example due to the impact of the three projects with huge errors.

⁴ For information about different size estimation levels, refer to www.isbsg.org *Sizing* and Practical Project Estimation 2nd edition.

Errors can arise from several sources:

- inaccuracies that are inevitable when an estimate is produced from incomplete information
- inaccuracies that might be caused by using a poor technique to do the estimate
- 'scope creep', when the final system contains functionality that was not part of the initial specification.

Three main techniques are used for estimating size. Between them they cover 85% of the projects:

- 40% estimate size from a data model
- 28% estimate size from the functional specification
- 17% estimate size by analogy with previous projects.

Generally, size is estimated well. Perhaps this is to be expected in a database of projects primarily submitted by organisations that establish size by some form of functional sizing method.

Size estimates are provided for 130 projects:

- 16 overestimated size by at least 10%
- 65 were estimated accurately to within 10% (32 were exact)
- 49 underestimated size by at least 10%.

The proportional size of these groups is about 1 (overestimate) to 4 (accurate) to 3 (underestimate).

Estimates produced from a data model fall into the three groups in approximately the same 1 to 4 to 3 proportions, but for estimates produced from the functional specification, the proportions are more like 1 to 6 to 3. In other words, estimates produced from the functional specification are more likely to be accurate. The accuracy of estimation by analogy cannot be assessed, since projects reporting that technique did not provide the actual estimates.

When size is overestimated, the average error is about 20%. These projects tend to be smaller than average.

Twenty of 49 underestimates exceed 50%. The largest error is nearly 400%, meaning the actual size was almost five times the estimate. On average, the actual size was 44% greater than the estimate. These projects tend to be larger than average.

In summary, for half of the projects the size estimate is accurate. Of the rest, underestimates are about three times as likely to occur as overestimates, and the error is likely to be two or more times as bad. Estimates produced from the functional specification are more likely to be accurate than estimates produced from a data model.

Conclusion

A collection of over 400 projects has been analysed to see how, and how well, people estimated their software projects.

Although the findings summarised below pertain to this particular collection of projects, the ISBSG believes that they provide a good picture of the state of estimation in good software projects.

Estimation techniques:

- When a size estimate is available, it is usually used to help estimate delivery date, effort, and/or cost. If a size estimate is not available, work breakdown techniques are used. Functional size-based techniques generally produce slightly more accurate estimates. Delivery dates set by Management directive are often achieved, while those set by legal requirements and customer directive are achieved even more often.

Accuracy of different types of estimate:

- Size tends to be estimated quite accurately. Estimates based on a data model are less accurate than estimates based on a functional specification.
- Delivery date is often estimated well. Over half of the projects were delivered early or on time and 60% were delivered no later than one month late. Once a project is more than a month late, the average error is more than 100% over-run. Projects planned to run for a year or more are delivered on time; the worst errors occur in projects planned to run for about 6 to 8 months.
- Effort is estimated worst. Nearly 60% of projects underestimate effort by at least 10%. Some enormous errors occur, with actual effort up to 20 times the estimate. On average, effort is underestimated by about 100%. There are no patterns to explain which types of projects are estimated better or worse than other types.
- Errors in cost estimates are closely related to errors in effort estimates, supporting the intuition that effort largely determines cost. But cost estimates are not generally as inaccurate as effort estimates. This may be because extra effort recorded against a project is unpaid.
- There is no relationship between project delivery rate and the accuracy of estimates.
- For everything except effort, half or more of the projects are estimated accurately; for effort, the number estimated accurately drops to less than a quarter of the projects.

Overestimates & Underestimates:

- A late project averages an over-run of 100%.
- If effort is underestimated, the average over-run is 100%. The average across all projects is to underestimate effort by 60%.
- If cost is underestimated, the average over-run is 25 to 30%.
- Overestimates are rare, usually small, and occur in small projects. When anything is overestimated, it is probably by about 20%.

Factors influencing accuracy:

- In general, smaller projects are more likely to be estimated accurately or overestimated. Larger and more particularly, longer projects are more likely to be underestimated.
- Enhancement projects are estimated more accurately than new developments, particularly as to delivery date.
- Estimates appear to be less accurate for projects involving relatively new technologies (client-server architecture, Visual Basic, PC) and those with a large and varied user base.

APPENDIX- Software Project Sample Demographics used

Details of the group of projects used for the analysis contained in this paper are as follows:

The projects come from 19 different countries, with the greatest representation from Australia, Brazil, Canada, Denmark, India, Japan, the Netherlands, the United Kingdom, and the USA.

Over 80% of the projects were completed in 1998 or later, and 65% in 2000 or later.

Industry sector: the main industry sectors represented are communications, manufacturing, insurance, service industry, banking, government, electronics & computers. Each has at least 30 projects, and together these provide 89% of the projects.

Organisation type: the main types of organisation represented are banking, insurance, manufacturing, communications, and government. Each has at least 30 projects, and together these provide 78% of the projects.

Business area: most projects are to do with banking, engineering, finance, manufacturing, sales and marketing, accounting, insurance or telecommunications. Each has at least 15 projects, and together these provide 83% of the projects.

Application type: the projects are dominated by transaction/production systems (41%), management information systems (12%), and office information systems (6%).

User base: 61% of projects are for use by a single business group and/or single location; 30% are for 6 or more business groups and 6 or more locations. Most projects (79%) support 6 or more concurrent users.

Development type: 45% are new developments, 51% are enhancements, and 5% are redevelopments.

Portability requirements: 70% are developed for a specific environment, and 30% are generic (developed to work across environments).

Client-server: 52% of projects use a client-server architecture, 48% do not.

Platform: 46% run on mainframes, 11% on midrange computers, and 42% on PCs.

Language: 52% use 3GLs, 38% use 4GLs, 9% use application generators, 1% use 2GLs. Access, COBOL/Cobol2, C/C++/C#, Java/J2EE/Javascript, Visual Basic, PL/I, Oracle, Cool:Gen, and SQL each has at least 15 projects, and together they account for 77% of projects.

CASE tools: are used in 35% of projects.

Software development methodologies: a methodology is used in 73% of projects. In-house methodologies dominate (47% in-house, 26% purchased and/or adapted, 27% no methodology).

Team size: small teams (1 to 4 developers) account for 28% of projects, 28% have teams of 5 to 8, and 44% have teams of 9 or more developers.

Size and other key project indicators:

Project size: size ranges from 11 to 20000 Function Points. The median is 275 FP.

Effort: for just the development team, the effort ranges from 17 to 104000 hours; the median is about 2300 hours. For the entire project effort (including support staff and user effort), the range is from 80 to 140,000 hours; the median is about 3300 hours.

Duration: projects range from 1 to 42 months. The median is 6 months.

Project Delivery Rate: PDR varies from less than 1 to over 100 hours per function point. The median is about 10 hours per function point.

Function points per elapsed month: this ranges from 1 to 500; the median is 48 function points per month.

Use the findings of this analysis to guide your approach to estimating and the allowances that you make to your estimate for the factors specific to your project. Use both macro- and micro-estimating techniques to obtain the most reliable estimate.

The International Software Benchmarking Standards Group (ISBSG)

The ISBSG is a not-for-profit organization founded in 1997 by a group of national software metrics associations. Their aim was to promote the use of IT industry data to improve software processes and products.

ISBSG is an independent international organization that collects and provides industry data of software development projects and maintenance & support activities in order to help all organizations (commercial and government, suppliers and customers) in the software industry to understand and to improve their performance. ISBSG sets the standards of software data collection, software data analysis and software project benchmarking processes and is considered to be the international thought leader in these practices.

The ISBSG mission is to help YOU and your organization improve the estimation, planning, control and management of your IT software projects and/or maintenance and support contracts.

To achieve this:

ISBSG maintains and grows 2 repositories of IT software development/maintenance & support data. This data originates from trusted, international IT organizations and can be obtained for a modest fee from the website www.isbsg.org/project-data/

Please help us to collect data

ISBSG is always looking for new data. In return for your data submission, you receive a free benchmark report that shows the performance in your project or contract against relevant industry peers.

Please submit your data through one of the forms listed on <http://isbsg.org/submit-data/>

Partners

This page will help you to find an ISBSG partner in your country:

<http://isbsg.org/meet-isbsg-partners/>