Realistic and Useful: Toward Better Estimates

Draft version 1.2 - April, 2007
(Draft version 1.0 - May, 2006)

Carlo Pescio, pescio@eptacom.net

Abstract:
The PERT technique of three estimates is a traditional, well-known approach to expert estimation. Estimators are asked for their optimistic, most likely and pessimistic estimation; then, a beta probability distribution is built to fit the data. Having a probability distribution for each activity allows a Montecarlo simulation to estimate the overall project effort with better accuracy. Still, experiments and practice suggests that estimators are often giving very narrow ranges, even in front of significant uncertainty, leading to unrealistic estimate.
Magne Jørgensen has developed an alternative approach that has been proven to increased realism. However, Jørgensen's technique is not directly based on a probability distribution, and therefore it is not directly possible to apply a Montecarlo simulation.
Here a novel approach, unifying Jørgensen's technique and PERT 3 estimates, is proposed and empirically validated using published data. Empirical validation shows that both realism and accuracy improve significantly using the unified approach.

1. Introduction

Estimation is a fundamental activity in software development, and considerable effort has been spent on devising formal estimation models. Still, in most cases, estimation is performed informally, using experience, intuition, possibly some historical data. This approach is known as *expert estimation* [Jørgensen2004d], meaning that one or more experts are providing an estimation based on their *experience*.

Although widely adopted, expert estimation is far from perfect. A major issue with expert estimation is lack of *realism* [Jørgensen2004a], [Jørgensen2004c]: experts are often overoptimistic (estimated times are often exceeded) and overconfident (they provide narrow ranges for their estimation, even in front of high uncertainty).

A widely known approach to expert estimation is the PERT technique of 3 estimates [Moder1995]. Under this approach, experts are asked for an optimistic estimate, a "most likely" estimate and a pessimistic estimate (hence the name of the technique). Then, assuming a beta probability distribution, where standard deviation is assumed to be one-sixth of the range of the [optimistic, pessimistic] interval, the technique provides the expected duration.

Jørgensen [Jørgensen2004a] introduced a new approach to expert estimation, aimed at improving the realism of estimates. He first asked for the most likely effort T, then for the probability of actual effort being higher than 2T or lower than ½ T. Realism has been experimentally determined to increase under this setting.
Note that the [½ T, 2T] range was initially inspired from the NASA *Manager's Handbook for Software Development*. That range is somewhat arbitrary, and could be replaced with a different one, without changing the estimator's mindset when reasoning about the estimate.

Jørgensen [Jørgensen2004b] has also devised a different technique, which removes the arbitrary range while still providing a more realistic estimate. First, the optimistic, most likely and pessimistic estimates are requested (like in PERT 3 estimates method). Then, he asked the estimator to assess the probability of actual effort exceeding the pessimistic estimation, and the probability of getting below the optimistic. Obviously, this removes the fixed range, and gives the estimator a chance to keep a narrow range (that most developers see as an indication of high competence) while conveying some information about uncertainty.

There are, however, a few issues with Jørgensen's techniques. A seemingly counterintuitive fact, documented only for the fixed interval approach, is that *realism* increases, but *mean magnitude of relative error* (MMRE[1]) does not. That suggests the technique is good at eliciting the inherent uncertainty, but can't make enough use of it. Techniques like the PERT 3 estimates, instead, are trying to make some use of the uncertainty at the activity level (deriving what is considered the expected effort for the single activity) and at the project level (using, for instance, a Montecarlo simulation based on the probability distribution of each activity).

Ideally, we would be able to combine some elements of Jørgensen's technique and of PERT 3 estimates into a unified approach, leveraging the higher realism of Jørgensen's approach and the statistical benefits of PERT. Before we attempt to do so, however, it is advisable to conduct a deeper analysis of strengths and weaknesses for the two approaches.

2. PERT technique of 3 estimates: strengths and weaknesses

Strengths

- It's a well-known technique, taught in many project management courses and supported by several widely adopted tools (e.g. Microsoft Project).
- Directly supports the idea that it's better to express an estimate as a range, not as a single number [Rule2000]
- Brings some degree of probabilistic reasoning into the estimation process. Although the expected effort is derived from a simple deterministic formula[2], the probability distribution of individual activities can be combined into a probability distribution for overall project effort. Both Montecarlo simulation and (simplistic) models based on the Central Limit Theorem have been applied[3].
- Prediction Intervals[4] for project effort can then be easily calculated from the project-level probability distribution.

Weaknesses:

- Estimates tend to be unrealistic: most often, the intervals given are too narrow to closely represent the inherent uncertainty. As Jørgensen reports in [Jørgensen2004c], '*the interviews with the software developers indicated an "unwillingness" to provide sufficiently wide effort PI under conditions of acknowledged high uncertainty*'.
- The "most likely" estimate is often overoptimistic, and given its strong weight in the expected duration formula, we end up with overoptimistic expectations.
- It is not trivial to derive a good PERT estimate from historical data (see [Jørgensen2004a]).
- People find the idea of giving their "most likely" estimate (the *modal* value) and getting back another "expected" estimate (the *mean* value) counter-intuitive. It would be more intuitive to provide a range (and maybe some other data but *not* the most likely estimate) and simply

---

1 MRE (magnitude or relative error) is defined by | actual - estimated | / actual. Given several samples, MMRE (mean MRE) is the mean value of MRE over the samples.

2 expected = (optimistic + 4 * most likely + pessimistic) / 6

3 The probability distribution of the whole project is extremely useful to answer questions like "what is the 80% Prediction Interval for the entire project" (contrast this with a mere knowledge of the P.I. for individual activities).

4 An X% Prediction Interval is a range [Ta..Tb] such that there is an X% probability of the actual duration falling between Ta and Tb.

get back the expected value. In many cases, I've found that estimators have difficulties recalling the difference between modal and mean value, so even a rationalization of the PERT process may not be trivial.

3. Jørgensen's model of 3 estimates + 2 probabilities: strengths and weaknesses

Strengths:

- Provides more realistic estimates than PERT. It builds on previous works [Jørgensen2004a] but removes the arbitrary constraint on the range ([½ T, 2T]).
- Directly supports the idea that it's better to express an estimate as a range, not as a single number. More than so: resonates with the idea (and experience) that people don't give their "true" minimum and maximum estimate, but instead provide "reasonable optimistic" and "reasonable pessimistic" estimates. The technique acknowledges so, and asks them to provide a confidence level on those estimates.

Weaknesses:

- There is more effort involved. Estimators have to provide 3 estimates like in PERT, plus two confidence levels.
- It is not trivial to derive from historical data, when compared with the fixed [½T, 2T] range (just like PERT).
- The "most likely" value is still being asked for; therefore, the same reasoning as above (PERT weaknesses) applies.
- Estimators might be uncomfortable providing confidence levels on the optimistic and pessimistic estimate. Training and guidelines might be necessary.
- Individual activity estimates can't be easily combined into a probabilistic estimate of the overall project effort. There is no obvious probability distribution associated with individual estimates, and it is not trivial to derive one.
- Confidence levels are provided based on intuition and on historical data only, but no formal checking of any kind is done. Therefore, there is not even a guarantee that there is any sensible probability distribution that could fit the data provided by the estimator.

4. A synthesis of methods

Merging PERT and Jørgensen's technique into a unified approach would reap some benefits:

- Increase realism of the estimates.
- Encourage probabilistic reasoning.
- Provide some theoretical assessment of the quality (or viability) of the estimation. The estimators can always restate the estimate if there is no sensible probability distribution for the figures given.
- Allow Montecarlo simulation of project effort, combining individual (activity-level) estimates. This reduces the estimation error at project level, and allows for prediction intervals to be calculated for the entire project.

Ideally, we should also try to remove some of the effort involved, and/or some of the most counter-intuitive steps in the process. In practice, if we don't ask for the "most likely" estimate we satisfy both those goals. Note that avoiding the "most likely" estimate may also help reducing the *anchoring* effect [Jørgensen2004d], [Armstrong2001] of an initial, "convenient" estimate.

Such a synthesis is indeed possible. We can ask the estimators for:

- optimistic time
- probability of duration being smaller than the optimistic estimate
- pessimistic time
- probability of duration being larger than the pessimistic estimate

Then, by assuming some kind of probability distribution (Beta, Triangular, etc) we can (heuristically) find a good probability distribution fitting the data. At that point, we will have a "most likely" estimate for the individual activities (the mean value), and also the ability to combine the probability distribution of several activities using (e.g.) a Montecarlo simulation. From the simulation, we can obtain the desired Prediction Intervals at the project level (e.g. the P80 or P90 Prediction Interval for project effort).
If the process of finding a probability distribution for an activity fails, because no distribution within the chosen family can fit the input data, we can also inform the estimator and provide some advice on how to improve the estimate.

Note that while [Jørgensen2004a] asked for the probability of actual effort being inside the interval, we ask for the probability of actual effort being outside the interval. From a formal point of view, there is no difference as one pair of data implies the other. In practice, over a few informal experiments, we found the chosen option to be more easily understood, but some formal experiment would be needed.

The ideas above have been validated in industrial settings for about one year, using a mathematical model built upon commercial, general-purpose computational tools. Given the encouraging results, they have been finally implemented in BetterEstimate™, an integrated, freely available tool (www.eptacom.net/betterestimate). The mathematical details can be found in [Pescio2007]; a short introduction is given in sidebar 1. In what follows, we'll call the unified approach described so far "BetterEstimate".

5. A first empirical validation

Although the BetterEstimate approach has been used in several real-world projects with good results, the focus has always been on successfully delivering products and not on collecting high-quality experimental data. Still, until some true experiment is carried out, we can use published experimental data to (empirically) answer two fundamental questions:

1) Can we really do without the "most likely" estimate?

2) Do we really improve combined estimates through simulation?

In what follows, we used Jørgensen's data (the "Alternative Approach" data in [Jørgensen2004a], see Table 1). It is important to note that Jørgensen's data are based on a [½ T, 2T] range, and that no tool was available for the estimators to "see" the probability distribution. Therefore, this is just a first-order empirical validation, as it is not fully conducted using the BetterEstimate approach.

Question 1 can be answered by using ½ T as the optimistic estimate, 1-MinConf as the probability of duration being smaller than the optimistic estimate, 2T as the pessimistic estimate, 1-MaxConf as the probability of duration being larger than the pessimistic estimate range. From those data, we obtain the expected (activity-level) duration for each task in Jørgensen's Alternative framing. We can then compare the Mean Magnitude of Average Error (MMRE) and also the Mean Balanced Relative Error (MBRE[5]) obtained using the original data / method and the BetterEstimate

5 BRE is defined as | actual - estimated | / min(actual, estimated). Given several samples, MBRE (mean BRE) is the mean value of BRE over the samples.

approach. BRE is less known that MRE, but is considered a more balanced (hence the name) measure of estimation errors, while MRE is favoring overestimates [Moløkken2005], [Foss2003].

Question 2 can be answered by considering a fictitious large project, involving all the activities identified as projects in Table 1. We can then observe whether or not the results of the simulation provide a better estimate of the total project effort, compared to just adding together the individual estimates.

Table 1

Project ID	Estimated	MinConf	MaxConf	Actual Effort
1	6	0.95	0.95	14.5
2	123	0.99	0.95	200
3	10.5	0.999	0.8	14.5
4	15.5	0.999	0.8	18
5	50	0.95	0.9	80
6	15	0.95	0.9	14
7	22.5	0.8	0.8	17
8	37.5	0.8	0.8	23.5
9	28	0.7	0.95	31
10	15	0.7	0.95	10
11	10	0.999	1	27
12	200	0.8	1	206
13	50	0.8	0.9	71.5
14	120	0.8	0.9	186.5
15	300	0.5	0.5	998
16	370	0.8	0.8	494
17	160	0.8	0.8	437
18	50	0.8	0.95	56
19	250	0.95	0.95	600
20	48	0.8	1	22
21	2527	0.999	0.99	2500
22	212	0.999	1	230
23	172	0.9	0.9	160

5.1 Results for Question 1

One project in Table 1 (Project ID 15) cannot be estimated using the BetterEstimate approach. Indeed, that project would **not** fit any sensible probability distribution (triangular, beta, log-logistic) as they all have a constantly growing distribution value, so the cumulative distribution at time 998 would be **strictly** higher than the CDF at time 300 under all models. This is a first confirmation that the lack of a tool-supported method to verify confidence levels may induce estimators in providing unreliable estimations.

We therefore adjusted the project list, excluding Project 15. Note that at this point, both the MMRE and r-ALT should be recalculated on the restricted set. MMRE gets slightly smaller (34%, was 36% including Project 15); we'll consider r-ALT below, when discussing realism.
It is also interesting to consider Project 21. Its duration is an order of magnitude higher than any other project in the list. The precision in the estimate is also very high (estimated 2527, actual 2500). Consequently, the MRE for this project is extremely small, and this is (artificially) deflating the MMRE. We will therefore evaluate and compare the results of the BetterEstimate approach with and without Project 21. Removing Project 21, MMRE grows back to 36%.
Jørgensen did not use MBRE for those data, but it can be easily calculated. For the original project list, MBRE is 64%. Excluding Project 15, MBRE drops to 57%. Further excluding Project 21, MBRE grows to 59%.

The results obtained from the BetterEstimate approach are represented in Table 2 (with the exclusion of Project 15).

Table 2

ID	Estimated Effort
1	8.312
2	181.059
3	16.863
4	24.893
5	70.925
6	21.277
7	28.395
8	47.325
9	25.616
10	13.723
11	14.837
12	174.967
13	54.258
14	130.219
16	466.936
17	201.918
18	50.050
19	346.323
20	41.992
21	3762.207
22	314.539
23	207.114

Table 3 summarizes the results. Note that we have a slightly worse MMRE (37% Vs. 36%), and a significantly better MBRE (46% Vs. 59%). It should also be noted that the standard deviation for the BetterEstimate approach is constantly smaller (both for MMRE and MBRE) compared with the standard deviation using the original data. This suggests that the BetterEstimate approach gives overall better distributed results.
It is important to remind that data wasn't collected using the BetterEstimate method / tool. The results we obtained, therefore, are not as good as shown in real-world applications. However, we can see that the BetterEstimate approach performs quite well, even without asking for the most likely estimate.

Table 3

	MMRE original data	MMRE BetterEstimate	MBRE original data	MBRE BetterEstimate
excluding P15	34	38	57	46
excluding P15 and P21	36	37	59	46

Realism of the results can be evaluated using the concept of bias, r-ALT, r-TRAD from [Jørgensen2004a]. Results are summarized in Table 4: bias is significantly smaller for the BetterEstimate approach, and the r-TRAD is significantly higher (in absolute value), indicating higher realism. Note that both bias and r-ALT changes dramatically when you remove a few points (in the original list, with all projects included, bias was 1% and r-ALT was -0.26). This suggests that a sensitivity analysis should be performed on the original data; the BetterEstimate approach seems to be less sensitive, but a sensitivity analysis could be useful anyway.

Table 4

	Bias original data	Bias BetterEstimate	r-ALT original data	r-TRAD BetterEstimate
excluding P15	-12	-3	-0.09	0.13
excluding P15 and P21	-13	-8	0	0.17

5.2 Results for Question 2

We considered two fictitious projects, one made of all the projects in the original list, excluding project 15, and another with the further exclusion of project 21. Table 5 and 6 summarizes the results; in Table 5, results from the BetterEstimate approach are obtained through Montecarlo simulation. Note that since we're dealing with a single value here, we don't calculate MMRE and MBRE but MRE and BRE.

Table 5

	Estimate (original data)	Estimate (BetterEstimate)	Actual total effort
Excluding P15	4492	6202.1	5412.5
Excluding P15 and P21	1965	2440.1	2912.5

Table 6

	MRE original data	MRE BetterEstimate	BRE original data	BRE BetterEstimate
Excluding P15	17	15	20	15
Excluding P15 and P21	32	16	48	19

Here we can see how the BetterEstimate approach performs significantly better than the original approach. In the best case, MRE drops from 32% to 16% and BRE from 48% to 19%.

From the Montecarlo simulation, we can also obtain Prediction Intervals for the fictitious project (see Table 7). We can see that actual total effort falls within PI(80) for both projects, and within P(50) for one, which makes a lot of sense: we expect P(50) to get about 50% hits.

Table 7

	PI(50)	PI(80)
excluding P15	[5339, 6914]	[4690,7358]
excluding P15 and P21	[2133, 2680]	[1914,2927]

We can therefore conclude that, for this empirical evaluation, the ability of the BetterEstimate approach to conduct a Montecarlo simulation significantly reduces errors at project level.

6. Further validation with experimental data

Jørgensen has also accumulated some experimental data for the 3+2 approach. This data set is extremely useful, as it contains both the expected effort (which is not asked for in the BetterEstimate approach) and all the data required by BetterEstimate. Those data have not been published[6], but Magne has provided me the entire data set (for all projects that were actually completed), as per Table 8.

6 Some of the projects were the same discussed in [Jørgensen2004a]. However, the estimates used here are from different people, although involved in the same projects. The data was estimated in the same sequence as represented in table 8 (Most likely, Min, Max, MinConf, MaxConf).

Table 8

Project ID	Most Likely	Min	Max	MinConf	MaxConf	Actual Effort
1	50	40	60	0.9	0.7	98.5
2	213	150	250	0.6	0.8	195
3	20	8	30	0.8	0.8	41
4	197	120	250	0.6	0.8	250
5	83	50	100	0.7	0.8	75
6	2605	1975	4125	0.8	0.99	2530
7	170	90	220	0.8	0.9	143
8	45	20	90	0.95	0.95	60
9	70	40	90	0.95	0.95	64
10	35	15	70	0.95	0.95	39
11	35	15	70	0.95	0.95	36
12	35	15	70	0.95	0.95	70
13	16	4	20	0.7	0.9	14

Once again, a project (number 13) has to be excluded from the data set, as no fitting probability distribution can be found (for a deeper look at the reasons, and about the feedback this is providing on the input data, see [Pescio2007], which further analyzes this specific project).
Once we do that, we can easily compare the Most Likely estimate provided by expert estimators and the Most Likely estimate provided by BetterEstimate (and based only on Min, Max, MinConf, and MaxConf)[7].

6.1 Effects on MMRE and MBRE

Both MMRE and MBRE improves by using the BetterEstimate approach (see Table 9):

Table 9

	MMRE estimator	MMRE BetterEstimate	MBRE estimator	MBRE BetterEstimate
excluding P13	21.7	20.7	35.6	32.2

The relative improvement is respectively by 4.6% and 9.6%.
Also, the standard deviation is significantly reduced for both MMRE and MBRE, suggesting that the BetterEstimate approach provides more consistent results (see Table 10).

Table 9

	MMRE STDDEV estimator	MMRE STDDEV BetterEstimate	MBRE STDDEV estimator	MBRE STDDEV BetterEstimate
excluding P13	18.4	17.9	40	36.4

Here the relative improvement is respectively by 2.7% and 9.0%.

We could also create a fictitious project as in section 5.2, using data in Table 8. In this case, however, the input data is rather peculiar, because the errors in the estimates balance almost perfectly, and the error made by simply adding together the Most Likely estimates is 1.21% (which is much better than we can usually expect in real projects).
Clearly, that does not leave much room for improvement, and indeed, a Monte Carlo simulation based on the BetterEstimate data will perform worse (3.43%).

[7] Note that, strictly speaking, we are **not** applying the BetterEstimate method here: the estimators have been asked for a Most Likely estimate, which may have influenced their choice of min/max. Still, this data set can be useful to further validate the concept.

7. BetterEstimate technique: strengths and weaknesses

Naturally, the BetterEstimate approach should be subjected to the same kind of scrutiny of the original methods. Ideally, we should have kept most of the strengths, while compensating for some weaknesses.

Strengths

- Directly supports the idea that it's better to express an estimate as a range, not as a single number. As in Jørgensen's approach, it is acknowledged that people don't give their "true" minimum and maximum estimate, but instead provide "reasonable optimistic" and "reasonable pessimistic" estimates.
- Brings probabilistic reasoning directly into the estimation process. Estimators can immediately see the probability distribution (or density) at the activity level. There is therefore an immediate formal checking that a sensible probability distribution exists that can fit the data.
- The probability distribution of individual activities can be combined into a probability distribution for overall project effort. Prediction Intervals for project effort can then be easily calculated.
- Provides more realistic estimates than PERT.
- There are no arbitrary constraints on the range (like [½ T, 2T]). This also reinforces the idea that the [min, max] range should **not** be symmetrically distributed around an expected value [Little2006].
- The "most likely" value is **not** being asked for: the process gives you one, as most people would expect from an estimation technique.

Weaknesses:

- There is more effort involved than in PERT, although less than in Jørgensen's 3+2 technique. Estimators have to provide 2 estimates and 2 confidence levels (or, as one of my students said, two (estimate, confidence) pairs).
- It is not trivial to derive from historical data, when compared with the fixed [½T, 2T] range.
- Estimators might be uncomfortable providing confidence levels. Training and guidelines might be necessary.

8. Conclusions and further work

Merging some traditional ideas from PERT and some intuitions from Jørgensen's work can lead to a simple, yet effective approach for expert estimation, which can be easily supported by automatic tools.

Further work in this direction includes:

- Experimental validation of the BetterEstimate approach. Although the approach has been successfully used in real-world projects and empirically validated on published data, it would be advisable to assess its behavior under a well-designed, strictly controlled experiment.

- Definition of guidelines, and possibly automated support, for setting confidence levels on the optimistic and pessimistic estimates. This may include further borrowing from consolidated estimation techniques (Delphi, paired comparison, etc), adapted to fit within the BetterEstimate approach. It could also be interesting to investigate how to use real-world historical data, often incomplete or partially unreliable, by accounting for uncertainty through the confidence levels. Group dynamics should also be investigated: for instance, we could evaluate the results of having someone estimate the range, while someone else estimates the confidence levels. Also, the concept or risk should be somehow introduced and related to uncertainty.

The freely available tool can also be improved on several dimensions. There should be a way to keep track of the project history, and the reasons for change in the estimates. This could help when estimating future projects. There should be support for dependencies, so to estimate duration and not simply effort. There should be a possibility for plugging-in model-based estimation for activities: for instance, at some point we may want to estimate debugging time from coding time through a custom model. Again, this may foster some interesting research.

Acknowledgments
I'm grateful to Magne Jørgensen for his feedback on earlier versions of this paper, and for sharing the experimental data used in section 7.

Bibliography

[Armstrong2001] J.S. Armstrong, The forecasting dictionary. Principles of forecasting: A handbook for researchers and practitioners. Ed. J. S. Armstrong. Boston, Kluwer Academic Publishers**,** p. 761-824.

[Jørgensen2004a] Magne Jørgensen, Realism in Assessment of Effort Estimation Uncertainty: It Matters How You Ask, IEEE Transactions on Software Engineering, April 2004.
Online reference (draft)
www.simula.no/departments/ engineering/.artifacts/framingsubmit.pdf

[Jørgensen2004b] Magne Jørgensen, *personal communication*.

[Jørgensen2004c] Magne Jørgensen, K. H. Teigen, Kjetil Moløkken-Østvold, Better sure than safe? Overconfidence in judgment based software development effort prediction intervals, Journal of Systems and Software, Vol. 70 Nr. 1-2, 2004.
Online reference:
http://www.simula.no/departments/engineering/.artifacts/pireviseddesember_copy.pdf

[Jørgensen2004d] Magne Jørgensen, A Review of Studies on Expert Estimation of Software Development Effort, Journal of Systems and Software, Vol. 70 Nr. 1-2, 2004.
Online reference:
http://www.simula.no/departments/engineering/.artifacts/expertsubmitnovember2002_copy.pdf

[Little2006] Todd Little, Schedule Estimation and Uncertainty Surrounding the Cone of Uncertainty, IEEE Software, May/June 2006.

[Moder1995] J.J. Moder, C.R. Phillips, and E.W. Davis, Project Management with CPM, PERT and Precedence Diagramming. Blitz Publishing, 1995.

[Moløkken2005] Kjetil Moløkken-Østvold, Magne Jørgensen, A Comparison of Software Project Overruns—Flexible versus Sequential Development Models, IEEE Transactions on Software Engineering, September 2005.

[Foss2003] Tron Foss, Erik Stensrud, Barbara Kitchenham, Ingunn Myrtveit, A Simulation Study of the Model Evaluation Criterion MMRE, IEEE Transactions on Software Engineering, November 2003.

[Pescio2007] Carlo Pescio, BetterEstimate: the (simple) Math Behind, draft version 1.0, 2007.
Online reference:
http://www.eptacom.net/betterestimate/TheMathBehind.pdf

[Rule2000] Grant Rule, Bees and the Art of Estimating, IEEE Software, November/December 2000, p. 23.

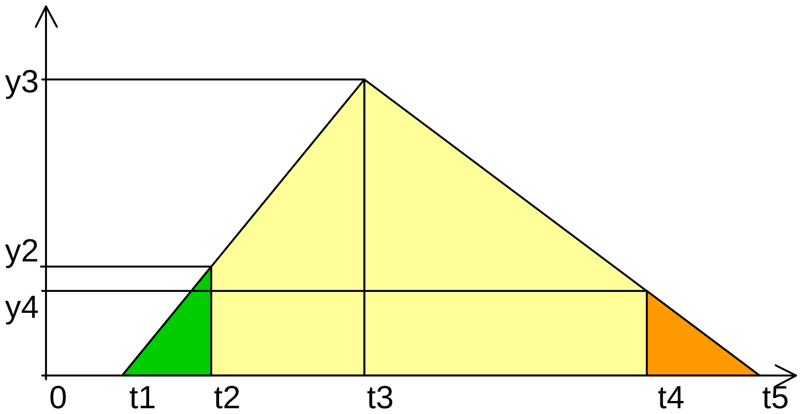
Biography

Carlo Pescio is a consultant and mentor for several companies across Europe. He approached programming in 1978 and graduated magna cum laude in Computer Science in 1991. As an architect, he has designed software for medical devices, industrial process control, banking, finance, CAD, and several other fields. His recent interests focus on software design, software economics and diagrammatic reasoning. He is as member of the IEEE Computer Society, the IEEE Technical Council on Software Engineering and the ACM. Contact him at pescio@eptacom.net.

Sidebar 1

How does it work?

Consider the following figure, where a triangular probability density is represented:



When we ask for the optimistic and pessimistic duration, we are asking about t2 and t4. Note that the probability density at t2 and t4 is not zero, as they are not the absolute minimum and maximum time.
When we ask for the probability of actual duration being smaller than t2, we are actually asking for the area of the green triangle, divided by the area of the overall triangle (t1-y3-t5).
When we ask for the probability of actual duration being bigger than t4, we are actually asking for the area of the orange triangle, divided by the area of the overall triangle.

All the other points (t1, t3, t5) are unknown, and they represent respectively the true minimum duration, the most likely duration, the maximum duration. Also, y2, y3 and y4 are unknown.

Although it seems like a trivial problem, a straightforward solution based of triangle similarity cannot be applied. Given the 4 input data, there is usually more than one solution (although there can also be no solutions, meaning no triangular density can fit the given data). Among the possible solutions, the adopted algorithm favors the widest range, to heuristically compensate for the over-optimism of most estimators.