Chapter 7
Time Prediction Methods and Principles

7.1 Unpacking and Decomposition

The prominent approach for reducing a problem's complexity is to decompose it into less complex subproblems, solve each of these, and then aggregate the subsolutions into an overall solution. In time prediction contexts, this approach is typically the basis of what has been referred to as the bottom-up method, the activity-based method, or predictions based on a work breakdown structure. Generally, across a range of domains, decomposition has been found to improve judgement quality and increase prediction accuracy [1]. In the domain of time predictions, however, there are also situations in which decomposition leads to overoptimistic and less accurate judgements [2].

The time prediction literature has examined two types of decomposition strategies: *unpacking*, which consists of merely listing or thinking about the subcomponents of a task before predicting the time usage as a whole, and *decomposition*, which consists of unpacking, predicting the time usage for each unpacked component, and then aggregating the time predictions into a prediction of the total time usage.

Unpacking: Unpacking tend to give higher time predictions. For example, if you ask people to list all the persons for whom they must buy Christmas presents, they will tend to predict that they need more time to complete their Christmas shopping compared to those who did not generate such a list. Unpacking strategies may be based either on the self-generation of components, as in the example above, or on reminding the participants of possible subcomponents, for example, by using a checklist for activities to be included. Checklists, or reminders, are usually very effective and easy to implement and may improve the accuracy of time predictions, particularly in situations in which there is a tendency towards overoptimistic time predictions [3]. In one study illustrating how unpacking tends to increase time predictions, participants were instructed to format a document to match a printed, edited version of the same document. When asked to predict how long it would take to format the document without any reminders of the work's components, the participants predicted, on average, eight and a half minutes. When asked to predict how long it

T. Halkjelsvik and M. Jørgensen, *Time Predictions*, Simula SpringerBriefs on Computing 5, https://doi.org/10.1007/978-3-319-74953-2_7

would take with reminders of the work's components, such as including italics and special characters (ə, ŏ, î), the participants predicted, on average, about 13 minutes [4].

In many cases, the increase in time predictions from unpacking a task contributes to greater realism [5]. Projects that used checklists when predicting time were, for example, more accurate and less overoptimistic (with predictions, on average, 10% too low) than projects that did not (with predictions, on average, 38% too low) [3]. Although unpacking may generally contribute to more accurate and less overoptimistic time predictions, this may not always be the case. Pointing out obvious components of a task, small and simple components, or components that are part of the last steps of a task may not lead to more accurate time predictions [4].

There are also other possible negative effects of unpacking in time prediction contexts. Attempts to identify when, where, and how to complete a task that involves concrete, specific plans of the steps involved may sometimes increase the level of overoptimism. This has particularly been observed in predictions of *when* a task will be completed (completion date) [6]. A step-by-step unpacking of a task may omit important components and focusing on the steps involved may make people think that the task will be performed exactly as imagined, without delays or interruptions, which, in turn, may lead to an illusion of control and overly optimistic time predictions.

Decomposition: Decomposition-based time predictions are based on dividing work into additive or multiplicative components, followed by predictions of the time required for each component, and, finally, aggregating the individual time predictions. Prediction of a project's total time usage may, for example, consist of time predictions of the development of part A (100 hours), part B (400 hours), part C (50 hours), and part D (50 hours), in addition to administrative activities (20% of the non-administrative work), which, in total, yields a time prediction for the project of $100 + 400 + 50 + 50 + 0.2 \times (100 + 400 + 50 + 50) = 720$ hours.[1]

The predicted time usage of smaller tasks is more likely to be overestimated, or at least less underestimated, than that of larger tasks. Since decomposition means predicting the time usage of smaller tasks, we should expect a higher sum of time predictions compared to non-decomposed predictions of the total work. If decomposition means higher time predictions, it means less bias towards too low time predictions for work that tends to be underestimated and stronger bias towards too high time predictions for tasks that tend to be overestimated. This effect was demonstrated in the following two experiments [7].

In the first experiment, two groups of research participants predicted the time of six small office tasks (e.g. delivering letters, making phone calls, and proofreading). One group predicted the total time of the first three tasks and then made separate time predictions for each of the last three tasks. The other group predicted the time of the same six tasks, but with a total time prediction for the last three tasks and separate

[1]Note that we are allowed to add the time predictions of the components only when they are the expected (*mean*) time usage of each component. Addition of the most likely time usage values yields time predictions that are too low. See Sect. 3.5 for more details on this.

time predictions for each of the first three tasks. The resulting time predictions were, as expected, higher when predicting the tasks separately (decomposed) than as a whole. In the office task situation studied, the general tendency was towards too high time predictions. This led to a stronger bias towards too high time predictions for the decomposed time predictions. The decomposed predictions were, on average, 9 and 10% too high, whereas the predictions of the tasks as a whole were only 2 and 5% too high.

The time predictions of the second experiment, which also included an office task and two groups of participants, had a general tendency towards being too low. As in the first experiment, the decomposed time predictions were higher but now led to more accurate time predictions. The decomposed predictions were, on average, 0 and 8% too low, whereas the predictions of the tasks as a whole were, on average, 13 and 26% too low.

Other studies have demonstrated that it is not always easy to know when decomposed time predictions will be more accurate. A study on software development, for example, found that decomposed time predictions of software development tasks were, on average, higher but also less accurate than time predictions of the tasks as a whole [8].

Accurate decomposition-based time predictions depend on the successful identification and aggregation of *all* the relevant components or, if that is not possible, the inclusion of sufficient time in the prediction of the total time usage to account for unknown components and unexpected events. Furthermore, aggregation of decomposed predictions requires the prediction of mean values (see Sect. 3.5), and potential dependencies between activities needs to be taken into account. These challenges are not so much an issue for non-decomposed time predictions, such as analogy-based ones. More on that topic in the next section.

Take home message 1: The use of checklists or reminders of work components typically have a positive effect on time prediction accuracy.

Take home message 2: The more you unpack or decompose the problem, the higher the time prediction will be, unless you forget components. Higher time predictions mean lower time prediction accuracy in situations in which tasks tend to be overestimated and higher accuracy for tasks that tend to be underestimated.

Take home message 3: Accurate decomposed time predictions require the successful identification of all components. If that is not possible, sufficient time should be added to account for unknown components and unexpected events.

7.2 Analogies

Judgement-based time predictions are likely to involve some type of recall of time spent on similar tasks or projects in the past, in other words, predictions depend on the use of analogies. For example, when we mentally simulate the completion of a

task, visualizing step by step what to do, we somehow have to rely on experience from previous similar tasks to know whether it will take 30 seconds, two minutes, or one hour to perform one of the steps. In this sense, all types of judgement-based time predictions are based on analogies. There is, however, an important difference between the intuition-based (unconscious) use of analogies and their explicit (conscious) use. This section is about the explicit use of analogies.

Analogy-based time predictions sometimes result in improved prediction accuracy. For instance, students completing a computer assignment gave less overoptimistic completion time predictions (delivery date predictions) when they collected and used the time spent on similar previously completed tasks to predict the time on the new task [9]. One benefit of the use of close analogies instead of decomposition is that a realistic number of unknown components and unexpected events, which in a decomposition-based time prediction would have to be properly identified, is often an inherent part of the analogy's time usage. This means that unknown components and unexpected events are incorporated into the analogy-based prediction.

Analogy-based methods, however, have other challenges that must be addressed to achieve accurate time predictions. In particular, it is essential to understand the work we try to predict sufficiently well to find close analogies. If we only have a vague and incomplete understanding of the new task, we will tend to recall less relevant and even irrelevant analogies. The less we know about what we want to predict, the more the work looks like the first analogy that comes to mind, regardless of the true similarity between the current and the previous work.[2]

When the analogies identified are not very close to what we want to predict or when the uncertainty is high, it is usually better to rely on several analogies instead of only one. Unless there are very good reasons for weighting one analogy more than others, a simple average (or preferably the median or trimmed mean) of these analogies is likely to give as accurate time predictions as more sophisticated combination strategies [11]. More on combinations of time predictions in Sect. 7.6.

One or more of the analogies identified may be unusual in one way or another. A task identified as a close analogy may, for example, have been solved with unusually low or high productivity. In such cases, it may make sense to adjust the simple average of the analogies towards the time usage of the average productivity of a broader range of similar tasks [12]. We explain this approach in the example below.

Let us say you want to go hiking to the top of a 1200-metre-high mountain and the distance is about 15 kilometres. You remember that you spent about five hours on your trip to reach the top of a mountain of approximately similar height last year and that the distance was about 14 kilometres (almost the same). Using this hike as your analogy yields a prediction of about five hours for the hike. However, you also know that your typical mountain hiking speed is about five kilometres per hour, which would yield a prediction of three hours for the hike. In other words, it seems you were much slower than usual on the hike you want to use as your analogy. Which prediction should you trust? The one based on the closest analogy or the one based

[2]This is an implication of the feature matching theory, when applied to comparisons of task. See [10].

on the average of several not-so-close analogies? A simple rule of thumb for this and similar situations is to use a 50–50 weighting of the closest analogy and the average speed. The new time prediction, which is a hybrid of the analogy- and average-based time predictions, is then (5 hours + 3 hours)/2 = 4 hours.

Generally, when there is a great similarity in the time usage of similar tasks, more weight should be given to the closest analogy or analogies. When, on the other hand, there is a large variance in the time usage or productivity of similar tasks, more weight should be given to the average of a larger set of analogies.[3]

The following real-world case exemplifies what may happen if the closest analogies are not recognized as being unusual. A large governmental organization successfully completed a software development project. Everything went as planned, the project involved exceptionally skilled members, and productivity was well above typical levels for such projects. When a new, reasonably similar development project was planned, the organization used the time spent by this previously completed project as the time prediction of the new project. What happened? The new project was now more like a *normal* project, and the project suffered great cost and time overruns. The scope of the project even had to be reduced to avoid a huge failure. The use of closest analogies is a useful method, but be aware of its limitations.

Take home message 1: Explicit use of relevant past experience (analogies) may lead to accurate time predictions, especially when it is possible to identify analogies *very* similar to the task to be predicted. When no very similar analogies can be identified, decomposition-based time predictions may lead to more accurate time predictions.

Take home message 2: If uncertainty is high or the analogies identified are unusual, for example, with respect to productivity, the predictions should be adjusted towards the average time usage or productivity of a larger set of similar tasks.

7.3 Relative Predictions

It may sometimes be useful to predict how much more or less time a task will take *relative* to another task rather than predicting the task's time usage. When predicting the time needed to paint a house, we may, for example, judge that it will require twice as much time to paint wall B compared to wall A. This type of time prediction method is called relative time prediction and is, amongst others, used in software development time prediction contexts [13]. A main motivation for the method is the

[3] This 50–50 rule of thumb assumes that the correlation between the time usage of the closest analogy and the time usage of mountain trips of similar length is about 0.5. If, for example, the correlation was as high as 0.8, it might be better to predict the time usage as 0.8 times the analogy-based prediction plus 0.2 times the prediction based on a broader set of tasks. In the extreme case in which there is no correlation between the closest analogy and the new task, the predicted productivity should be based only on the average productivity of the broader set of tasks. The above rule of thumb includes a few assumptions, for instance about the variance of the variables, but may work well in many contexts.

belief that we are frequently better at predicting relative time usage than absolute time usage. Another motivation for relative time prediction is that it may be more robust with respect to *who* is doing the work. Assume that we do not yet know who will paint the walls and that people differ greatly in how fast they paint. Consequently, an absolute time prediction without knowing who the painter will be may be of little value, whereas a relative time prediction may still be useful, assuming that that the relative difference in time spent on the different walls is fairly constant across painters.

Relative predictions may be stated in *percentages*—for example, that painting wall A takes 200% (twice as much) of the time it takes to paint wall B—but may also be stated as *additive* relative predictions—for example, that wall B will take 30 work hours more than wall A. One challenge with the use of percentages was demonstrated in an experiment with software professionals. One group was asked to predict the time needed to complete project A as a percentage of the time required for project B and another group was to predict the time needed for project B as a percentage of the time required for project A. On average, the first group believed that project A was 78% of project B and the second group believed that project B was 70% of project A [14]. Clearly, it would be paradoxical if project A required less time usage than project B and, at the same time, project B required less time usage than project A. In the same study, another group of software professionals gave relative predictions in the form of differences in work hours. These predictions made more sense and suggested that people struggle more with getting the percentages than the differences right when predicting relative time. The disadvantage of using absolute differences (e.g. 30 hours more) instead of proportions (e.g. 30% more) is, however, that the time predictions become more person dependent.

A relative prediction process sometimes used by project planners is the *story points*-based time prediction method,[4] outlined below.

Time prediction based on story points

- Divide the work into manageable tasks, for example, tasks believed to take a maximum of one person-week.
- Pick one task, preferably one that is medium large and well understood by all experts, to be your *reference task* (or baseline task).
- Give the reference task a number of story points, for example, agree on 10 story points. The number of story points of the reference task is arbitrary and could be any number.
- Predict the time usage of all the other tasks relative to the reference task. If one task is believed to take half the amount of time it takes to complete the reference task (10 story points), it should be given five story points. A task that is believed to take 50% more time than the reference task should be given 15 story points and so on.

[4]The term *story points* is derived from the concept of user stories, which are short descriptions of what users want to achieve with the software to be constructed.

- Knowledge (or prediction) of the time usage of the reference task or knowledge about the typical productivity (story points per hour[5]) allows for the conversion of story points into actual time units, such as work hours, or how much a team is capable of completing the next week. Knowledge about the productivity may be derived from previous tasks, or based on feedback from actual productivity (e.g. story points per work day) of the first deliveries of the work.

There are few empirical studies on the benefits and drawbacks of *relative* time prediction methods compared to the more common *absolute* time prediction methods. Seemingly, there are contexts in which time prediction accuracy improves [15] and contexts in which it worsens when relative time predictions are given [16]. One general advice about relative time predictions is to be careful with and perhaps even avoid comparisons of tasks that differ substantially in size. The reason for this is that tasks are often perceived as more similar than they actually are (assimilation effect). A task that is, in reality, 10 times larger than the reference task may consequently be more underestimated when using relative instead of absolute time prediction methods.

Take home message 1: Relative time prediction methods may simplify the prediction process and reduce the importance of knowing who is going to complete the work.

Take home message 2: Generally, relative time predictions do not seem to be more accurate than absolute time predictions. When tasks differ substantially in size, relative prediction methods may lead to prediction biases, such as underestimating larger tasks and overestimating smaller tasks.

7.4 Time Prediction Models

Historical data may be used to build formal (mathematical) models predicting the future. While researchers seem to enjoy developing complex prediction models, publishing hundreds of them in academic journals, there is surprisingly little evidence in favour of complex prediction models compared to simpler ones. In 97 comparisons between simple and complex prediction models (from domains other than time prediction), the complex models had no advantage when judged by the level of prediction accuracy [17]. Instead, the complex models increased the prediction error by 27%, on average. A simple model is defined here as one that the user understands how to apply, how previous outcomes have been represented in the model, how the relations are integrated in the model, and, finally, the relation between the model and the predictions.

Evaluations of the accuracy of prediction models in the domain of time prediction seem to arrive at the same result. We find no indication that complex models yield more accurate time predictions than simple models with few variables [18]. On the

[5]This productivity measure is typically called the *velocity* by those using this method.

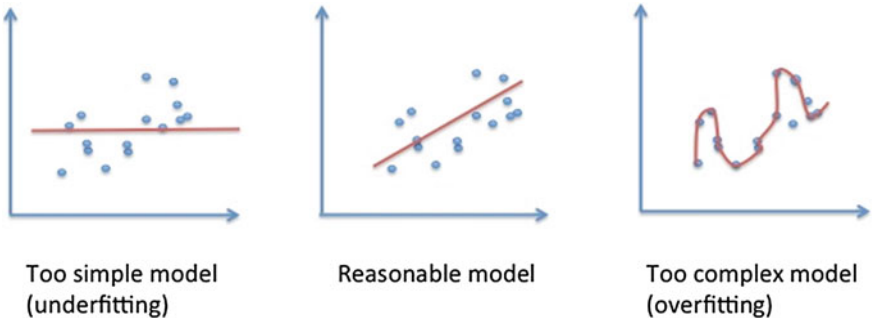

Fig. 7.1 A model (red line) can fit the data points poorly (left), reasonably well (middle), and too well, leading to overfitting (right)

contrary, complex models tend to be less accurate in many realistic time prediction situations [19]. One likely reason for the lower accuracy of complex models is the lack of stable relations between important input variables, and the output (the predictions). For example, the effects of task size (e.g. number of requirements, features to be developed, steps to be taken) and task complexity (e.g. the number of workers to coordinate) on time usage seem to vary greatly across different work contexts. If such effects differs greatly across contexts and these differences are not properly modelled, the more complex time prediction models will tend to overfit the historical data. That is, the models become very good at explaining past time usage at the expense of their ability to predict the future. Illustrations of a too simple model, a reasonable model, and an overfitted model of the relation between task size (x-axis) and time usage (y-axis) are displayed in Fig. 7.1. If you ever encounter a time prediction model that uses terms you do not understand, such as *neural networks* or *fuzzy logic*, or that asks you to provide all kinds of detailed information about the task, it is usually a good idea to be sceptical; model overfitting is highly likely.

When developing a formal model to predict time usage, it is usually safest to use *local* data, such as data from your own context, the organization where you work, or at least data from very similar contexts [20]. Time prediction models intended to be applicable across a range of contexts are typically less accurate [19].

The following is an example of how the ingredients of a simple local time prediction model can be derived:

1. Identify a meaningful measure of the *size* of the work, for example, pages to be written, tests to be graded, or features to be developed for your particular context.
2. Identify a meaningful categorization of the *complexity* of the work. This should correspond to identifying contexts with significantly different levels of productivity, for example, pages with simple versus complex content, multiple choice tests versus essays to be graded, and software features involving presentation only versus interactive features.

3. Obtain the actual time usages, preferably from local historical data, for the above-mentioned components.

Based on the above ingredients, we may, for example, develop the following very simple test-grading time prediction model.

- Our size measure is the number of tests.
- A meaningful categorization of the complexity of the test grading is multiple choice (M) and essay (E).
- Historical, local (in this case, from the person conducting the test) data suggest that we need, on average, five minutes to grade a multiple-choice test and 15 minutes for an essay. In addition, we typically need 10 minutes for general preparation.
- The final model is then as follows: time usage (in minutes) = $10 + 5 \times M + 15 \times E$.
- A prediction of how long it will take to grade five multiple choice tests and seven essays is then $10 + 5 \times 5 + 15 \times 7 = 140$ minutes.

As pointed out earlier, adding more complexity to time prediction models does not often pay off. You should consequently have good reasons to add many more variables or implement more complex relations to your time usage prediction model.

What if the work to be predicted is so special that you have no relevant historical data? Or what if you did not bother to collect any data before now? A good solution, when feasible, is to *start the work* to see how productive you are and then try to extrapolate from the data collected in the first stage of your work. This technique is advocated by software professionals in the NoEstimates movement [21]. This movement argues that most time predictions are a waste of time and even harmful. It argues that, instead of making time predictions, one should measure progress and extrapolate this progress to determine when tasks will be finished.[6] In spite of the unrealistic suggestion that people should stop requesting and giving time predictions before a project starts, it is indeed a good idea to extrapolate time usage from the first stage to predict the time usage of the remaining work. Essentially, the NoEstimates movement advocates the use of a simple time prediction model based on highly local data. Few, if any, time prediction models are better than that.

Sometimes we know in advance that time prediction models will perform poorly. This includes situations with so-called *broken leg information*.[7] Broken leg information is very specific information about a task that makes the average of previous performance less diagnostic of future performance. Knowledge that a person's leg is broken should, for example, lead to a radical change in predictions of how quickly that person will walk to work from home. Similarly, if we know that a worker is extremely skilled and experienced, our model based on the average productivity of the typical worker will not work well.

It is sometimes difficult to know whether one is facing a broken leg situation, and it is important to avoid thinking that most situations are special. Human judgement

[6] This is actually a type of time prediction.

[7] The introduction of broken leg information (broken leg cues) and a discussion on the need to sometimes deviate from history-based predictions is described in Paul Meehl's famous paper from 1957 [22].

about *when* a model is valid is essential in many time prediction contexts. 'Know when to hold' em and know when to fold 'em', to quote words of wisdom from Kenny Roger's song "The Gambler".

Take home message 1: Unless there is strong evidence in favour of complex formal time prediction models, use a simple model.

Take home message 2: Collect data from your own context (local data) to build the models. If no such data are available, use data from very similar contexts.

Take home message 3: Judge whether any information about the task makes the use of the model invalid. On the other hand, do not think that most cases are special and thus qualify as broken leg situations.

7.5 Consider Alternative Futures

One method that has been suggested for increasing the realism of time predictions is to think about what can go wrong and try to recall difficulties in past tasks. This approach has an intuitive appeal, because a typical explanation for cost and time overruns is people's failure to take problems and difficulties into account. Still, at least two studies attempting to exploit the strategy of recalling a problematic past or imagining a problematic future have been unable to document improvements in time predictions [23, 24].

A few studies have even suggested that the opposite may happen. People urged to identify more reasons for failures or more risks factors may produce even more optimistic time predictions than those identifying fewer such reasons,[8] see also [26]. This result is not as strange as it sounds and similar results have been found for a range of other types of judgements. If it is easy to come up with negative information, such as when asked for only one reason for failure or the three most important risk factors, then one might think that this is a particularly problematic project, since it is so easy to come up with the reason or risk factors. When, on the other hand, it is hard to come up with reasons for failure, such as when asked to come up with 12 or as many as possible reasons for failure, one may use the difficulty in finding the *last* reasons as an indication of the overall risk. If you feel that it is very hard to come up with the last reasons for failure, you may use this perception as an indication that failure is unlikely and thus make more optimistic time predictions. Thus, identifications of a large number of risk factors may affect time predictions in unexpected ways.

Let us try a small experiment. Predict the number of minutes you need to read the next 10 pages of this book. Do not read on before you have made a prediction.

Now, make a *new* time prediction about the same reading task, but assume that nothing unexpected will happen, that there will be no interruptions, and that there will be no need to reread parts that are difficult to understand.

[8] See [25]. Note that several later papers by the first author were retracted due to fraud.

Table 7.1 Example of a sequence of time predictions for one individual

Day	Time prediction	Actual time usage
1	15 minutes	19 minutes
2	17 minutes	Failed/gave up
3	16 minutes	27 minutes
4	16 minutes	18 minutes
5	17 minutes	26 minutes
6	18 minutes	15 minutes
7	13 minutes	14 minutes

Did you arrive at a different time prediction for the second, best-case situation? If you are like 25% of the participants in a similar experiment [27], you did not update the time prediction at all and, if you are like most of the other participants, it is likely that the difference between your first and second predictions is very small and not likely to reflect the real difference between a realistic and a best-case situation. What we think is a realistic outcome tends to be too close to the best-case outcome. This finding is not restricted to time predictions and is present in many types of predictions, for example, predicting how often one will exercise in the coming weeks [28].

In an unpublished study, we asked students to predict the time required to solve Sudoku (a type of number placement puzzle) tasks. All the Sudoku tasks were of similar complexity, the participants had experience in solving them, and we did not expect much learning or improvement from one Sudoku task to another. The students first predicted the time usage for a Sudoku task, then completed the Sudoku task, and finally received feedback about the actual time usage. This was done for several Sudoku tasks over seven days. An interesting time prediction pattern emerged that can be phrased as follows: *The future me is as good or better than the historically best me*—not the average, not the worst, but the best me. An example of a typical student's time predictions and actual time usage is given in Table 7.1.

The initial time prediction of the student was not really bad, just a bit low. The second time prediction assumed a performance better than the first day, perhaps based on the assumption of learning. In reality, the student failed completely at the task (the maximum available time was 60 minutes). In spite of this, the student believed that the next performance would be even better than that predicted the day before. The following days, until day six, the student systematically used more time than predicted but still believed that his performance would be a better than what he did at his best. On day six, the student spent less time on the task than predicted and immediately updated the time prediction to something even better than his new best performance. This student was not alone in thinking that his best performance was the most likely outcome for the next task. Many of the other students had this tendency in their time predictions, as well. This tendency towards insufficient differentiation of the ideal, best-case situation, where everything goes as planned, and the most likely case, where a normal amount of disturbances and problems occur, is likely to contribute to the amount of overoptimistic time predictions [23, 27].

One successful approach for dealing with this tendency seems to be to force a more explicit contrasting of the ideal and most likely outcomes. In short, one first asks for a time prediction given an ideal (best-case) situation and describes what an ideal situation means (lack of disturbance, no unexpected problems and optimal productivity), then one asks for the most likely (or realistic) use of time. This setup seems to make people understand that their time prediction of the most likely outcome should differ from the best-case scenario and they will therefore make higher and often more realistic time usage predictions.

In a study with software development professionals, this technique increased the time predictions by 30%, which, in that particular context, led to more realistic predictions [27]. Higher time predictions do not always mean more accurate predictions, but if you have a friend who is typically late or a team that typically underestimates time usage, try asking for the ideal-case prediction first and then require a most likely time usage prediction.

Take home message 1: Imagining alternative negative scenarios, such as possible reasons for failure and risk factors, does not seem to improve the accuracy of time predictions.[9]

Take home message 2: People have a tendency to give predictions of most likely time usages that are the same or too close to those given when instructed to assume best-case or ideal conditions. In short, what we think is realistic time usage tends to be too close to the best-case time usage.

Take home message 3: Imagining and predicting time usage assuming ideal conditions before predicting most likely time usage tends to lead to higher and, in many contexts, more accurate time predictions.

7.6 Combinations of Time Predictions

We once asked our students to predict the height of the Royal Palace in Oslo, Norway. Before looking at the students' predictions and not knowing the actual height of the Royal Palace, we made the following bet with them: Our prediction, which will be the median value of your predictions, will be as good as or closer to the actual value than at least half of your predictions. Why is this bet very safe?

Twelve students participated in the bet and gave the following predictions (in increasing order): 10, 15, 15, 20, 21, 23, 30, 40, 60, 75, 120, and 370 metres. The median of these numbers is 26.5 metres. This turns out to be very close to the actual height of the Royal Palace, which is 25 metres. Our prediction was consequently not only more accurate than most of the students' predictions, but more accurate than *all* of them. Admittedly, the median of a group of predictions is seldom better than all

[9] This point should, of course, not be used to argue against systematic risk analyses and assessments of how risk factors affect the uncertainty of projects, which are very useful for project planning and management.

the predictions, as in this case, but our bet that the median would beat at least half of the predictions was very safe. In fact, the median prediction will *always* be as good as or better than half of the predictions used to calculate the median.

This mathematical fact may be demonstrated as follows:[10] The worst case for the median (our prediction) is that the actual value is more than or less than all the predictions made by the students. In that case, the median will be better than exactly 50% of the predictions. If, for example, the actual height of the Royal Palace is only nine metres, the median will be better than exactly 50% (30, 40, 60, 75, 120, and 370 metres) of the 12 individual predictions. In all the other cases, where the actual height lies somewhere within the range of predictions, the median will be more accurate than at least 50% of the individual predictions. If, for example, the actual height were 125 metres, the median of 26.5 metres would be better than seven of the predictions (10, 15, 15, 20, 21, 23, and 370 metres).

Research studies in construction planning, economic forecasts, patient survival rates, and many other areas all demonstrate that prediction accuracy tends to increase with the use of combined predictions.[11] This research also suggests that combinations using the median or the *trimmed mean* (mean when, e.g., the 25% highest and 25% lowest predictions are removed) are typically more accurate than using the mean of the predictions [30]. In the example where the students predicted the height of the Royal Castle, we see that the mean of the students' predictions would be less accurate due to the influence of a couple of very high predictions, whereas a trimmed mean, for instance, removing the two lowest and two highest predictions, would be pretty accurate.

The benefits of combined time predictions depend on the individual predictions being fairly *independent* of each other. When predictions are influenced by the same misleading factor, the same shortcoming of the time prediction method, or the same narrow experience of those making the predictions, the advantage of combining the predictions will be greatly reduced. Furthermore, when we have good reasons for believing that some experts are systematically and substantially more accurate than others, we should rely on their predictions or at least weight their predictions more than those of the others; that is, we should not ignore that the 'wisdom of smaller, smarter crowds' may be better than the 'wisdom of crowds'.[12] When, however, it is difficult to know who is systematically and substantially more accurate, as in most

[10] Although the explanation is simple, the accuracy of the median prediction has surprised and fascinated many, starting perhaps with Sir Francis Galton. See Galton's 1907 discovery that the median (not the mean, as often claimed) prediction of the weight of an ox, including many non-experts' judgments, were better than the predictions of the experts (see [29]).

[11] See [30]. The review in this paper found an improvement in prediction accuracy for all 30 contexts studied. The average improvement in prediction accuracy was 12.5%.

[12] The idea of *vox populi* (the voice of the people) is that even the predictions of people with low expertise will lead to good predictions when their predictions are combined. This idea, originating with Galton, has been repeated in the best-selling book *The Wisdom of Crowds* by James Surowiecki [31]. Most of the research suggests, however, that smaller groups with more qualified people result in better predictions. Clearly, it would be strange if combining many unqualified predictions would guarantee a qualified prediction. See [32]. See also [33].

time prediction contexts, it is better to stick with the median or the trimmed mean of all the predictions.[13]

So far, we have only talked about *mechanical combinations* of time predictions, that is, calculated combinations of independent, individual time predictions. What about the role of *discussions* and *groups* in combining predictions? While most people seem to believe that groups usually improve the quality of judgements, for example, through meetings, psychologists have long held the view that groups tend to decrease the quality of judgements. Business managers, for example, like to solve problems in meetings and to generate new ideas through brainstorming sessions. Psychologists, on the other hand, report that brainstorming in groups is not a good idea [35]. Who is right?

There are good reasons to be sceptical about predictions based on group discussions. It is well documented that group discussions sometimes lead to the censorship of opinions and so-called *groupthink*, where the desire to get along with the group's members leads to a preference for arguments that have already been expressed in the group [36]. There are, however, also cases in which group discussions are reported to produce better judgements, such as when trying to detect lies [37]. How about group discussions in the realm of time predictions? Will time predictions by groups reflect the madness or the wisdom of crowds?[14]

A number of potential effects seem to be involved in determining the effect of group discussions on time predictions. A series of studies on undergraduate students confirms the pessimistic view that information sharing is biased [39]. Time predictions produced by groups of students were more overoptimistic than the typical predictions given by individuals, perhaps because the group discussions focused on factors related to successful completion and less on potential negative factors.

In contrast, studies on groups of software professionals predicting the time usage of a development project produced results in favour of group-based predictions. The software professionals predicted the time usage individually (before engaging in group discussions), then contributed to a consensus-based time prediction in small groups of experts, and finally made individual predictions (after the group discussion). Both the group-based time prediction and the individual time predictions made after the group discussions were higher and more realistic than the individual time predictions made before the group work [34]. Based on qualitative analyses of the discussions, it seemed as if the groups were able to identify a larger set of activities that needed to be taken into account. The mechanical combination of judgements improved the time prediction accuracy compared to the individual time predictions, but the group-based time predictions improved the accuracy even further in this particular context.

---

[13] See [34]. Note that, when there are few, perhaps only two or three time predictions to combine, the mean will be the only meaningful way to combine them.

[14] This refers to two famous books on judgments in groups. The first one was published in 1841 and suggested that groups are very poor decisions makers. See [38]. The other book was published in 2004 and suggested the opposite, that groups make surprisingly good judgments and decisions. See [31].

Planning poker is an example of a structured method for time prediction in groups, inspired by the Delphi method [13, 40]. The following steps are typical steps in a time prediction game of planning poker:

1. The participants write down their individual (independent) time predictions on a card or choose premade cards with the appropriate numbers.
2. The participants simultaneously show their cards to each other (thus, the participants are not affected by each other and their initial predictions remain independent).
3. The participants, in particular those with the highest and lowest time predictions, justify and explain their time predictions.
4. The participants discuss their differences and new insights.
5. Steps 1–4 are repeated until a consensus or a fixed number of iterations (e.g. three) has been reached. If no consensus is reached, a mechanical combination using the median, trimmed mean, or mean of the predictions is used.

A study of the use of the planning poker method in a real-world software development context found that it decreased the median time prediction error (measured as the deviation from the actual use of time) from 50 to 33% [41]. An additional benefit was that the structured group-based time prediction led to a better and shared understanding of the work and its risks.

The use of combined predictions is also relevant to uncertainty assessments, as in deriving realistic prediction intervals. In a study of different strategies for combinations of uncertainty assessments in a software development context, group discussions generated prediction intervals that were more accurate than mechanical combinations of the intervals [42].

Take home message 1: The median of several time predictions is always more accurate than at least half—and usually more—of the individual time predictions.

Take home message 2: Unless you have good reasons to believe that the source of one of the time predictions is systematically and substantially more accurate than the other sources, use the median or the trimmed mean of several time predictions.

Take home message 3: Combinations of independently produced time predictions usually give better predictions than combinations affected by each other or that have common influences.

Take home message 4: Structured group discussions are likely to improve the time prediction accuracy, especially when discussions lead to the identification of more activities and potential challenges.

7.7 Let Other People Make the Prediction?

Results from research in social psychology suggest that other people, typically referred to as *observers*, tend to provide less biased time predictions than those

who will actually complete the task, typically referred to as *actors*. The underlying idea is that actors are personally involved, motivated to finish quickly, and consequently biased. In contrast, observers have no stake in the performance and will tend to give more objective and less overoptimistic time predictions.

The results from a study on students and their assignments support this argument. The study found that those who were required to hand in the assignment (the observers) tended to be overoptimistic, but those who predicted the completion date for another student (the actors) were not [9]. The *actors* believed they would complete the work an average of 1.3 days earlier than they did. The *observers*, who predicted the completion time of other students, were too pessimistic and predicted a delivery time an average of 1.7 days later than the actual delivery time. Thus, observers may be less optimistic, but not necessarily more accurate.

When it comes to the prediction of time usage—and not completion times, as in the study above—the improvement by use of observers instead of actors is even less clear. One study found no difference in accuracy or bias between predictions of how long the actors themselves would spend building a computer stand and predictions of how long the average person would take [23]. Another study, predicting time usage in voicemail operations, found that observers with a high level of experience were even more optimistic than novices performing the task, whereas intermediate observers, perhaps with a better recollection of the difficulties of learning the operations, were more realistic [24].

There are often good reasons for letting people predict the time usage of their own work. In particular, this is the case when people know a great deal about *how* they plan to solve a task and *how much* time they have spent on similar tasks in the past. Several studies on software development have shown, perhaps for this reason, that predicting one's own time usage is associated with higher accuracy and with no increase in the level of optimism compared to predictions of other people's work [43, 44].

Judgements by observers may be more accurate than those by actors when assessing the *uncertainty* of time predictions, such as predictions of confidence intervals of time usage. This seems to be the case particularly when historical information on past prediction accuracy is available. One study found software developers to be strongly overconfident about the accuracy of their time predictions for their own tasks even with historical prediction accuracy information easily available, whereas observers, in this case other software developers, gave much more realistic uncertainty assessments based on the same information [45]. It seems as if actors tend to emphasize their specific knowledge about how to solve the task and neglect information about their previous time prediction accuracy. Observers, on the other hand, have little to rely on besides past time prediction accuracy. Information about previous time prediction accuracy is, as argued earlier, typically a better indicator of time prediction uncertainty than knowledge about how the specific task will be carried out.

Take home message 1: Actors, that is, those completing the tasks, tend to give more optimistic time predictions than observers in some contexts. Nevertheless, there are several contexts in where time predictions become more accurate when people predict

the time usage of their own work rather than when the work is predicted by less involved observers. Generally, it may be better to let people predict the time usage of own work, especially when they have more knowledge about how to complete it and their previous productivity on similar tasks.

Take home message 2: Observers' time prediction uncertainty assessments seem to be more realistic than those of actors, especially when historical data about previous time prediction accuracy are available.

7.8 Removing Irrelevant and Misleading Information

At a seminar with software professionals, we once explained how arbitrary numbers (anchors) and other irrelevant information often influence our judgement. Following this introduction, the participants were asked to estimate the programming productivity of their last project. Before estimating their productivity, they were asked a question that included either an unreasonably high anchor ('Do you believe your programming productivity was less than 200 lines of code per hour on your last project?') or an unreasonably low anchor ('Do you believe your programming productivity was more than one line of code per hour on your last project?'). They were also warned that this question was meant to mislead them and that they should try to be as realistic as possible. Do you think the information they received about the anchoring effect along with the explicit warning removed the anchoring influence?

The information did not even come close to removing the effect of the anchors. Those with the low anchor thought that their productivity on the last project had been, on average, 12.4 lines of code per hour, while those in the high-anchor group thought that their productivity had been about three times higher, 35.2 lines of code per hour. Actually, the warnings did reduce the anchoring effect somewhat. A third group of participants, without the initial teaching and warnings, were even more affected by the same anchor manipulations.[15]

This anchoring result is similar to numerous other disappointing results on how difficult—perhaps impossible—it is to remove biases stemming from misleading and irrelevant information [47]. Not only is it difficult to reduce the effect of misleading information through instructions to ignore it, even more elaborate means to reduce bias do not seem to be of much help either. In a study on software professionals, we found that methods such as highlighting the relevant text and removing/hiding the irrelevant information by crossing it out with black ink did not remove all of the biasing effect on the time predictions.[16] The only safe way to avoid being affected by misleading or irrelevant information is, consequently, to completely avoid it, for example, by letting another person filter out misleading and irrelevant information before the time predictions are made. Thus, a useful time prediction technique or

[15] The results are described in [46].

[16] See [48]. The crossing out the irrelevant information method did, however, work much better than the highlight the relevant information method.

principle is simply to remove irrelevant information and neutralize misleading information before those in charge of the time prediction receive it.

Take home message: No known method can be used to remove the influence of irrelevant and misleading time prediction information, such as prediction anchors. The only method that works is to avoid exposure to such information. This may, for example, be accomplished by having another person remove irrelevant information and neutralize misleading information.

7.9 From Fibonacci to T-Shirt Sizes: Time Predictions Using Alternative Scales

In many contexts, we only need a rough prediction of how much time an activity will take. To avoid spending time on deriving time predictions with higher precision than needed, it may be useful to make predictions at less precise scales or use approximate task size categories. The following are examples of scales and categories sometimes used for this purpose:

- The Fibonacci-inspired scale (where the two previous values are added to determine the next value), with increasing differences between the numbers: 1, 2, 3, 5, 8, 13, 21, ….
- The base 2 scale with even more increasing differences between the numbers: 1, 2, 4, 8, 16, 32, 64, ….
- The 10 × scale: 10, 20, 30, 40, 50, ….
- The T-shirt categories: X-small, small, medium, large, and X-large, where each category is represented by a time value, for example, medium tasks take, on average, 20 h and large tasks, on average, 60 h.

When predicting time using an alternative scale, it is important to be aware of the *central tendency of judgement* [49] and of unintended loss of information. The central tendency of judgement tells us that we tend to be affected by what is perceived to be the middle of the chosen scale. The less we know about something and the greater the uncertainty, the more we are affected by the perceived middle value of the scale or interval we use. In many ways, this is a rational reaction to uncertainty (see the discussion on the magnitude bias in Sect. 5.5), but it may also lead to biased time predictions.

The scale effect was demonstrated in an experiment where we asked one group of students to use the linear (1, 2, 3, 4, …) scale with numbers up to 40, which has the middle number 20. Another group of students used the Fibonacci-inspired, nonlinear scale 1, 2, 3, 5, 8, 13, 20, 30, 40, which has the middle number 8 [50]. The students were asked to predict the number of work hours they would use for a software development task using numbers from the scale they had been assigned. The students who used the linear scale gave a median time prediction of 20 hours, while those with the nonlinear scale gave a median prediction of only eight hours,

which, in this context, was clearly too low. Mainly the least experienced participants were affected by the use of the nonlinear scale. The results consequently suggest that we should be careful when using nonlinear scales, especially in a context with high uncertainty and low experience. If, as in our study, the middle value of the alternative scale is lower than that of the linear scale, the choice of that scale will lead to lower time predictions, especially when the uncertainty is high.

When using time usage categories, such as in t-shirt estimation (e.g. small = 10 hours, medium = 40 hours, and so on), the effect of the middle category (the medium size) may be similarly strong. People tend to select the middle category when they do not really know what to predict or they think the uncertainty is high; that is, people sometimes use the medium or middle category as a replacement for 'I don't know'. Clearly, there is much power in deciding what is perceived as the middle value or category of a scale used for responses.

A scale's low *precision* may also lead to an unintended loss of information. Say, for example, that one knows, based on previous experience, a task takes 40 hours. Being forced to select between 32 and 64 hours in a base 2 scale or between a medium (20-hour) and a large (60-hour) task in t-shirt estimation may be unfortunate in this case. Consequently, if one has chosen to rely on such scales, it should be possible to deviate from the scale numbers when there are good reasons for it.

Nonlinear scales have been claimed to be more natural or intuitive than linear scales.[17] Even if this were true, the limited empirical evidence on this topic in time prediction contexts does not suggest that nonlinear scales lead to more accurate time predictions [50]. We have conducted several (unpublished) experiments on this topic, all with the same disappointing result that the use of nonlinear scales provides no clear accuracy benefits.

Take home message 1: The use of low-precision scales, such as Fibonacci-like scales, may speed up time prediction work.

Take home message 2: Do not expect the use of alternative scales, such as nonlinear scales, to improve time prediction accuracy.

References

1. Armstrong JS, Green KC, Graefe A (2015) Golden rule of forecasting: be conservative. J Bus Res 68(8):1717–1731
2. Jørgensen M (2004) Top-down and bottom-up expert estimation of software development effort. Inf Softw Technol 46(1):3–16

[17] Studies of indigenous cultures and small children suggest that our intuitive number system (approximate number system) is nonlinear, with increasing intervals at increasing magnitudes. This number system has several advantages. It is able to compress high numbers on a short numerical scale and it reflects the fact that a difference in one unit is more important for small numbers (2 vs. 3) than for large numbers (1000 vs. 1001). See, for example, [51].

3. Furulund KM, Moløkken-Østvold K (2007) Increasing software effort estimation accuracy using experience data, estimation models and checklists. In: IEEE seventh international conference on quality software, 2007. pp 342–347
4. Hadjichristidis C, Summers B, Thomas K (2014) Unpacking estimates of task duration: the role of typicality and temporality. J Exp Soc Psychol 51:45–50
5. Kruger J, Evans M (2004) If you don't want to be late, enumerate: unpacking reduces the planning fallacy. J Exp Soc Psychol 40(5):586–598
6. Buehler R, Griffin D (2003) Planning, personality, and prediction: the role of future focus in optimistic time predictions. Organ Behav Hum Decis Process 92(1):80–90
7. Forsyth DK, Burt CD (2008) Allocating time to future tasks: the effect of task segmentation on planning fallacy bias. Memory & Cognition 36(4):791–798
8. Connolly T, Dean D (1997) Decomposed versus holistic estimates of effort required for software writing tasks. Manage Sci 43(7):1029–1045
9. Buehler R, Griffin D, Ross M (1994) Exploring the 'planning fallacy': why people underestimate their task completion times. J Pers Soc Psychol 67(3):366–381
10. Tversky A (1977) Features of similarity. Psychol Rev 84(4):327–352
11. Clemen RT (1989) Combining forecasts: a review and annotated bibliography. Int J Forecast 5(4):559–583
12. Jørgensen M, Indahl U, Sjøberg D (2003) Software effort estimation by analogy and 'regression toward the mean'. J Syst Softw 68(3):253–262
13. Cohn M (2005) Agile estimating and planning. Pearson Education, NJ, USA
14. Jørgensen M (2013) Relative estimation of software development effort: it matters with what and how you compare. IEEE Softw 30(2):74–79
15. Fredriksen I (2009) Empirical research on relative and absolute effort estimation in software development projects. Master's thesis, University of Oslo
16. Haugen NC (2006) An empirical study of using planning poker for user story estimation. In: Proceedings of the conference on AGILE 2006 IEEE, Washington, DC, pp 23–34
17. Green KC, Armstrong JS (2015) Simple versus complex forecasting: the evidence. J Bus Res 68(8):1678–1685
18. Jørgensen M (1995) Experience with the accuracy of software maintenance task effort prediction models. IEEE Trans Software Eng 21(8):674–681
19. Jørgensen M (2007) Forecasting of software development work effort: evidence on expert judgement and formal models. Int J Forecast 23(3):449–462
20. Kitchenham BA, Mendes E, Travassos GH (2007) Cross versus within-company cost estimation studies: a systematic review. IEEE Trans Softw Eng 33(5)
21. Duarte V (2015) No estimates: how to measure project progress without estimating. Oikosofy
22. Meehl P (1957) When shall we use our heads instead of the formula? J Couns Psychol 4(4):268
23. Byram SJ (1997) Cognitive and motivational factors influencing time prediction. J Exp Psychol Applied 3(3):216–239
24. Hinds PJ (1999) The curse of expertise: the effects of expertise and debiasing methods on prediction of novice performance. J Exp Psychol Applied 5(2):205–221
25. Sanna LJ, Parks CD, Chang EC, Carter SE (2005) The hourglass is half full or half empty: temporal framing and the group planning fallacy. Group Dyn Theory Res Pract 9(3):173–188
26. Jørgensen M (2010) Identification of more risks can lead to increased overoptimism of and over-confidence in software development effort estimates. Inf Softw Technol 52(5):506–516
27. Jørgensen M (2011) Contrasting ideal and realistic conditions as a means to improve judgment-based software development effort estimation. Inf Softw Technol 53(12):1382–1390
28. Tanner RJ, Carlson KA (2008) Unrealistically optimistic consumers: a selective hypothesis testing account for optimism in predictions of future behavior. J Consum Res 35(5):810–822
29. Galton F (1907) Vox populi (The wisdom of crowds). Nature 75(7):450–451
30. Armstrong JS (2001) Combining forecasts. In: Principles of forecasting, vol 30. International Series in Operations Research & Management Science. Springer, Boston, MA, pp 417–439
31. Surowiecki J (2004) The wisdom of crowds. Doubleday, New York

32. Goldstein DG, McAfee RP, Siddharth S (2014) The wisdom of smaller, smarter crowds. In: Proceedings of the fifteenth ACM conference on economics and computation. ACM, pp 471–488
33. Budescu DV, Chen E (2014) Identifying expertise to extract the wisdom of crowds. Manage Sci 61(2):267–280
34. Moløkken-Østvold K, Jørgensen M (2004) Group processes in software effort estimation. Empirical Softw Eng 9(4):315–334
35. Diehl M, Stroebe W (1987) Productivity loss in brainstorming groups: toward the solution of a riddle. J Pers Soc Psychol 53(3):497
36. Baron RS (2005) So right it's wrong: groupthink and the ubiquitous nature of polarized group decision-making. In: Zanna MP (ed) Advances in experimental social psychology, vol 37. Academic Press. San Diego, CA, pp 219–253
37. Klein N, Epley N (2015) Group discussion improves lie detection. Proc Natl Acad Sci 112(24):7460–7465
38. Mackay C (1841) Memoirs of extraordinary popular delusions and the madness of crowds. Routledge, London
39. Buehler R, Messervey D, Griffin D (2005) Collaborative planning and prediction: does group discussion affect optimistic biases in time estimation? Organ Behav Hum Decis Process 97(1):47–63
40. Linstone HA, Turoff M (eds) (1975) The Delphi method: techniques and applications, vol 29. Addison-Wesley, Reading, MA
41. Moløkken-Østvold K, Haugen NC, Benestad HC (2008) Using planning poker for combining expert estimates in software projects. J Syst Softw 81(12):2106–2117
42. Jørgensen M, Moløkken K (2002) Combination of software development effort prediction intervals: why, when and how? In: Proceedings of the 14th international conference on software engineering and knowledge engineering. ACM, pp 425–428
43. Jørgensen M (2004) Regression models of software development effort estimation accuracy and bias. Empirical Softw Eng 9:297–314
44. Lederer AL, Prasad J (1995) Causes of inaccurate software development cost estimates. J Syst Softw 31:125–134
45. Jørgensen M, Gruschke TM (2009) The impact of lessons-learned sessions on effort estimation and uncertainty assessments. IEEE Trans Software Eng 35(3):368–383
46. Mair C, Shepperd M, Jørgensen M (2014) Debiasing through raising awareness reduces the anchoring bias. ualresearchonline.arts.ac.uk/7334/1/BPS_Poster_2014_Mair_Shepperd_A0.pdf. Accessed May 2017
47. Løhre E, Jørgensen M (2016) Numerical anchors and their strong effects on software development effort estimates. J Syst Softw 116:49–56
48. Jørgensen M, Grimstad S (2008) Avoiding irrelevant and misleading information when estimating development effort. IEEE Softw 25(3):78–83
49. Hollingworth HL (1910) The central tendency of judgment. J Philos Psychol Sci Methods 7(17):461–469
50. Tamrakar R, Jørgensen M (2012) Does the use of Fibonacci numbers in planning poker affect effort estimates? In: 16th international conference on evaluation & assessment in software engineering (EASE 2012). IET, pp 228–232
51. Dehaene S, Izard V, Spelke E, Pica P (2008) Log or linear? Distinct intuitions of the number scale in Western and Amazonian indigene cultures. Science 320(5880):1217–1220