Expert vs Algorithmic Estimation

2018-03-23

Software Effort Estimation

Effort estimation is an activity that is a part of software planning. Traditionally effort estimation was done by the project manager, but nowadays the whole development team participates. Uncertainties make effort estimation difficult in the early stages of development, but the estimate becomes more accurate as the project advances. The effort is usually underestimated compared to the real effort measured at the end of the project. Effort estimation plays several roles in software development, for example, it is used to calculate the estimated cost and budget of the project's activities and communicate with customers and developers of how much time the project will take (Sommerville 2010, chap. 23).

The next sections discuss the two models that can be used for effort estimation: **Experience based estimation** and **algorithmic estimation**, and then makes a comparison between the models.

Experience Based Estimation

Experience-based estimation is based on the previous experience of the project's developers (experts) and it will lead to an informed judgment of the effort. The difficulties in experience-based estimation are estimating the productivity of the developers which depends on attributes such as the developer's skill level, their familiarity with the problem and discipline. The lack of commonalities compared to previous projects can also make the estimation process difficult The estimation activity should involve a group (development team) and each of the members should explain their estimates. Involving a group reveals factors to others that they may not have considered helping to iterate towards agreed group estimate (Sommerville 2010, chap. 23). One such approach that involves a group is **planning poker** (Cohn), which is played in the following way:

1) Player pick cards from a deck of cards that reads 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100.
2) If the values are not similar the members with highest and lowest values explain their choice. A timer can be used to limit the length of the discussion about the design.
3) Game is played until the values converge, usually, two or three rounds, otherwise previous step is repeated.

The benefits of planning poker are:

1) It brings together multiple expert opinions to do the estimating process.
2) It creates dialogue ensuring that the estimations are justified.
3) Averaging individual estimates and group discussion about the estimates leads to better results.

Algorithmic Estimation

Algorithmic estimation is a formulaic approach to effort estimation based on estimated on product attributes and process characteristics. According to Sommerville (2010) (chap. 23.5) an often used simple formula is

$$\texttt{effort} = A \cdot \texttt{size}^B \cdot M,$$

where

- A is a constant factor depending on the local organizational practices and the software type.

- B is the value of the exponent, usually between 1.0 and 1.5 because the effort does not usually increase linearly with the project size due to increased complexity and communication overhead.
- M is multiplier made by combining process, product and development attributes such as dependability requirements and experience of the team.
- `size` may be assessment of code size or functionality estimate.

Some of the problems with the algorithmic model are that the `size` is difficult to estimate. Also, the estimates of B and M are subjective. In order to estimate the parameters accurately, the organization requires historical data, which only large organizations may have. Parameter values published in studies can also be used, but they may not relate to their organizational values. The errors in the estimated parameters can lead to large errors in the effort estimate rendering it useless. One model for software cost estimation is **COCOMO II** for which the parameters are derived fitting regression formula using a large number of historical projects.

Comparison

According to (Lassenius 2015) experience based models such as planning poker work generally better than algorithmic models. Both models require knowledge of past projects for good quality estimations. I could imagine that in the future with feeding machine learning algorithms with quality data from the project algorithms could surpass the experience based human-made estimates, but it is left to be seen.

Traditionally the effort estimation was only done by the project manager but nowadays influenced by agile development practices all developers participate in the effort estimation. This creates better estimates and facilitates communication about the estimates.

Bibliography

Cohn, M., *Planning poker estimating in detail* [online]. Available from: https://store.mountaingoatsoftware.com/pages/planning-poker-in-detail [Accessed 23 Mar 2018].

Lassenius, C., 2015. *10 - software project management, spring 2015 on vimeo* [online]. Available from: https://vimeo.com/aaltose/review/122133092/9b1cd21ac8 [Accessed 23 Mar 2018].

Sommerville, I., 2010. *Software Engineering: 9th Edition*. 9th ed. Addison-Wesley Publishing Company.