What we estimate and why it is important

Effort vs duration

Activities often take longer than planned even though the effort has not increased

Effort = amount of work if one person were doing it

Duration = actual amount of work with however many people available doing it

The effects of over- and under-estimating

Under-estimate

- Project can fail due to budget being exceeded
- The allocation of not enough money can result in substandard work as staff work extra hard to produce what they can

Overestimate

- An excessively high estimate may lead to the work being lost to a competitor
- Parkinsons Law (‘work expands to fill the time available’) means that an exces¬-sively generous estimate may lead to lower productivity

Estimates and targets

Hard to be exact

An estimate of effort or time is really a most likely effort/time with a range of possibilities on each side of it

Choose a target:

- Aggressive – could get done quickly but high chance of failure
- Generous – likely to expand the length of time needed, but have a safer chance of the target being met’
- A reasonable target can become a self-fulfilling prophecy

Checklist

If you are using someone elses estimates, if you can then ask them:

- What methods were used to produce the estimates?
- How is the relative size of the job measured (in other words, what are the size/effort drivers)?
- How much effort was assumed would be required for each unit of the size driver (in other words, what productivity rates are you assuming)?
- Can a past project of about the same size be identified which had about the same effort?
- If a job with a comparable size cannot be identified, can past jobs which had similar productivity rates be found?

Estimation best practices

- Use the most reliable data available
- Spend as much time as possible to produce the estimates
- Use appropriate methods
- State the basis of the estimate
- Establish best practices through lessons learned

Using expert judgement

Use completion of other tasks to get inform-ation for estimates

You need to know:

- What activities are going to be carried out
- How much work for each

The **best person** to tell us about the time it takes to complete a task is someone familiar with the tasks to be carried out and the environment in which they are done

Advantages

Using expert judgement (cont)

- People doing the work are involved with the estimating process
- Involves the people with the best experience of similar work and work environment in the past

Risks:

- Task may be a new one of which there is no prior experience
- Human error
- Estimate is essentially a guess and its hard to know how accurate
- May need to talk to several ‘experts’

The Delphi approach

- A group of experts are asked to produce, individually and without consulting others, an estimate supported by some kind of rationale
- Replies collected by a moderator
- Circulated anonymously
- Everyone reads everyone else and has opportunity to revise opinion
- Opinions should converge on a consensus

Estimating by analogy

The function point approach (and, indeed, the more generic approach of using size drivers and productivity rates) is based on the assumption that we have the details of the size driver values and actual effort of past projects. Often, however, such records do not exist.

Analogy or comparative approach could be used.

Steps:

1. Identify the key characteristics of the new project.
2. Search for a previous project which has similar characteristics.

By **Natalie Moore** (NatalieMoore)

cheatography.com/nataliemoore/
www.jchmedia.com/

Published 7th November, 2015.
Last updated 12th May, 2016.

Estimating by analogy (cont)

3. Use the actual effort recorded for the previous project as the base estimate for the new one.
4. Identify the key differences between the old and the new projects (it is unlikely that the old project is an exact match for the new one).
5. Adjust the base estimate to take account of the identified differences

If there is no single past project then use parts of old projects

Agile estimation

Need a way to estimate that:

- Allows budget creation
- Plans for the future
- Reminds us that estimates are guesses
- Acknowledges inherent complexities and uncertainty with software dev
- Keep things simple – estimate includes everything
- Be fast

Relative sizing

Estimating absolutely is harder than relatively

Relative sizing means how big is this compared to that

1. How fast can the team go
2. Size stories relatively
3. Set expectations around dates

Agile estimates are unit less

Point based system

1 point = small, no sweat easy

3 points = medium, bigger but we can handle it

5 points = large, this will take some effort

Sit down with customer, ask a lot of questions, guess how big this is

Do this for each part of the project

Agile estimation (cont)

Do it better as a team – get team involv-ement

Agile because there are problems with waterfall estimates:
- Clear on being guesses
- Usually overly optimistic
- Beginning has too many unknowns
- The only question we should be attempting to answer at the beginning is “is this project even possible with the time and resources we have got?”

Approaches to estimating

Bottom up (analytical or activity-based estimating)

1. Break the task into component sub-tasks
2. and then break the component sub-tasks into sub-sub-tasks
3. And so on, until we get to elements that we think would not take one or two people more than a week to complete
4. To get an overall estimate of the effort needed add up all the effort for the component tasks

Recommended if you have no historical records of relevant past projects to guide you

Disadvantage: time-consuming as you have, in effect, to draw up a detailed plan of how the project is to be carried out first

Top Down

1. Look for some overall characteristics of the job to be done
2. From these, produce a global effort estimate
3. Nearly always based on our knowledge of past cases

Parametric approach

One way of base number calculation in top down

Size drivers: Number of variables to be completed

Other drivers often include:

- Availability of tools
- Communication overheads, including time waiting for approval
- Stability of the work environment. (Risk of changes to resources and requirements)
- Size of the project team. Larger jobs with lots of people involved are often less efficient

Function points

Not all IT projects involve writing software

Many use off the shelf

If do need programming tradition to use lines of code to estimate the size, but problems with this:

- The code is a very technical product – it would need a software expert to estimate the number of lines of code
- You will not know the exact number of lines of code until quite near the end of the project; most other size drivers are known at the beginning, or at least at an early stage, of the project
- Lots of languages and some need less code

Better to use function points which estimates the amount of work based on the outputs of the project / features of the program

We can use a function point count to find out the relative productivity of develop-ment projects that have already been completed

By **Natalie Moore** (NatalieMoore)

cheatography.com/nataliemoore/
www.jchmedia.com/

Published 7th November, 2015.
Last updated 12th May, 2016.