I494 Lecture Notes

Class 6 – October 4, 2010

In this class we developed an understanding of different estimating techniques. Ultimately we want to use our ability to estimate to drive the prioritization of our requirements. Our goal should be to avoid guesswork, and to have a repeatable process we use to define our work.

Just to set the stage for our process, we need to define duration and effort. Duration is elapsed time to complete a project, activity, or task. Effort is the labor hours to complete a project, activity, or task. Remember, duration is NOT the same as effort - we don’t work 24 hours a day.

If you involve multiple people in a project there can be a great amount of variance in estimates, so the trick is to develop a model that can withstand a good deal of uncertainty. Estimates are predictions, and we ultimately want to have a low variance on our estimates relative to their true accuracy. Estimates can be revisited as you complete work. The tracking of actual effort is important so that the project manager can make a determination of the completion date for your project.

There can be multiple reasons why estimates vary, and of course then the corresponding completion times will vary in similar fashion. Here are some common reasons:

- Varying skill levels
- Unexpected events
- Efficiency of work time
- Mistakes and misunderstandings
- Random variations

You might be interested to know that in controlled experiments there can be as much as a 30X difference in efficiency for completing software development tasks. The same experiments showed that the correctness also varied by an equally large amount. Luckily the two are not correlated. Think about what it might mean if they were…

If we consider skill level, we normally estimate a more-skilled person will complete a task in less time than a less-skilled person. Be cautious of speed, as you need to understand how quality problems might hurt you. If someone is fast and sloppy you are in trouble. Usually, more skilled people make fewer errors, but because they end up being assigned to more complicated tasks their errors might be more costly than a less-skilled person.

We can consider a number of different ways to estimate, each of which has certain good properties, and less good properties. We can use:

- Similarity to other tasks
- Historical data
Comparing to tasks that you have completed before is a good way to determine how long YOU will take to complete a task. Usually efficiency grows slowly, so you probably won’t get burned by a terribly wrong estimate – as long as you are the person completing the tasks. Alternatively, you can use historical data about how long it takes others to complete tasks. This gives a range of possible estimates, but for teams makes sense in the aggregate. Consulting an expert can provide a view towards how long they would take to complete, and if they will do the work you’ll want a contract to control risk.

However you get a number, we want to use the power of a group to refine the number and get buy-in on how long a project will take. We discussed the Delphi technique, which uses the following steps:

For each task:
1. Each person makes an estimate
2. Display as a histogram
3. Have each person explain if at the extremes • both low and high extremes
4. Repeat steps 1 through 3
5. Each person makes a final estimate
6. Take the average as the best estimate

Consider the following graph of estimates used in the class as an example:

We can see that there are two extreme cases. A lot of people thought it would take one hour and some other people thought it would take six hours. There’s a big difference in these two values, especially if you multiply out over all the tasks in a project. In the Delphi technique people that estimate at the extremes are asked to verbally describe why they arrived at their estimates. The value of this sharing is that people learn either that the high estimate has more deeply thought out the problem, or has low skill and is just overestimating to manage risk. Conversely the person at the low end might have an easy solution to the problem, or is naïve.
We showed an iteration of the estimates and attained the following chart:

![Chart](image)

And after explaining again the extremes we received a final set of estimates:

![Chart](image)

Though this is a staged example, we did one in the class that gave a real-life view into how this technique works.

We then extended the technique using the three-point method, which asks people to consider the best case (Optimistic), worst case (Pessimistic), or the expected case (Most likely) for task completion. It is critical that each of these cases makes an assumption that the task will ultimately be completed correctly, so the worst case is not the absolute worst case, which is when we fail on task completion. The three-point method combines the estimates algebraically as follows:

\[
\text{Estimate} = \frac{(O + 4M + P)}{6}
\]

We wrapped all of this estimating into a method for creating a prioritized list of requirements. To do this we need to consider the Cost to satisfy the requirement (cost to build or buy) to the Value derived from the requirement. I described in class the Analytical Hierarchical Process, which uses the following steps that you can construct in a spreadsheet:

Copyright 2010 – Dennis P. Groth
Step 1:

- Create an $N \times N$ matrix of requirements

![Diagram showing a matrix with requirements and comparison scale]

- Compare Requirement 1 to Requirement 2
  
  1 – equal
  3 – slightly higher
  5 – strongly higher
  7 – very strongly higher
  9 – absolutely higher
  Negative if lower

Step 2:

- Fill in the upper triangle

![Diagram showing filled matrix with comparison values]

1 – equal
3 – slightly higher
5 – strongly higher
7 – very strongly higher
9 – absolutely higher
Negative if lower

Copyright 2010 – Dennis P. Groth
Step 3:

- **Fill out the remainder**

<table>
<thead>
<tr>
<th></th>
<th>Req-1</th>
<th>Req-2</th>
<th>Req-3</th>
<th>Req-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req-1</td>
<td>1</td>
<td>-5</td>
<td>9</td>
<td>-7</td>
</tr>
<tr>
<td>Req-2</td>
<td>5</td>
<td>1</td>
<td>9</td>
<td>-3</td>
</tr>
<tr>
<td>Req-3</td>
<td>-9</td>
<td>-9</td>
<td>1</td>
<td>-9</td>
</tr>
<tr>
<td>Req-4</td>
<td>7</td>
<td>3</td>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>

1 – equal  
3 – slightly higher  
5 – strongly higher  
7 – very strongly higher  
9 – absolutely higher  
Negative if lower

Step 4:

- **Calculate decimal value**
  
  If value < 0, decimal value = $1/\text{value}^{*}1$

<table>
<thead>
<tr>
<th></th>
<th>Req-1</th>
<th>Req-2</th>
<th>Req-3</th>
<th>Req-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req-1</td>
<td>1</td>
<td>0.2</td>
<td>9</td>
<td>0.14</td>
</tr>
<tr>
<td>Req-2</td>
<td>5</td>
<td>1</td>
<td>9</td>
<td>0.33</td>
</tr>
<tr>
<td>Req-3</td>
<td>0.11</td>
<td>0.11</td>
<td>1</td>
<td>0.11</td>
</tr>
<tr>
<td>Req-4</td>
<td>7</td>
<td>3</td>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>
Step 5:

- **Normalize**
  
  \[ \text{value} = \frac{\text{value}}{\text{sum of column}} \]

<table>
<thead>
<tr>
<th></th>
<th>Req-1</th>
<th>Req-2</th>
<th>Req-3</th>
<th>Req-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req-1</td>
<td>0.076</td>
<td>0.046</td>
<td>0.321</td>
<td>0.090</td>
</tr>
<tr>
<td>Req-2</td>
<td>0.381</td>
<td>0.232</td>
<td>0.321</td>
<td>0.210</td>
</tr>
<tr>
<td>Req-3</td>
<td>0.008</td>
<td>0.026</td>
<td>0.036</td>
<td>0.070</td>
</tr>
<tr>
<td>Req-4</td>
<td>0.534</td>
<td>0.696</td>
<td>0.321</td>
<td>0.630</td>
</tr>
</tbody>
</table>

Step 6:

- **Calculate score**
  
  **Average of the rows**

<table>
<thead>
<tr>
<th></th>
<th>Req-1</th>
<th>Req-2</th>
<th>Req-3</th>
<th>Req-4</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req-1</td>
<td>0.076</td>
<td>0.046</td>
<td>0.321</td>
<td>0.090</td>
<td>0.13</td>
</tr>
<tr>
<td>Req-2</td>
<td>0.381</td>
<td>0.232</td>
<td>0.321</td>
<td>0.210</td>
<td>0.29</td>
</tr>
<tr>
<td>Req-3</td>
<td>0.008</td>
<td>0.026</td>
<td>0.036</td>
<td>0.070</td>
<td>0.03</td>
</tr>
<tr>
<td>Req-4</td>
<td>0.534</td>
<td>0.696</td>
<td>0.321</td>
<td>0.630</td>
<td>0.55</td>
</tr>
</tbody>
</table>