Anchoring and Adjustment in Software Estimation

Jorge Aranda
February, 2005

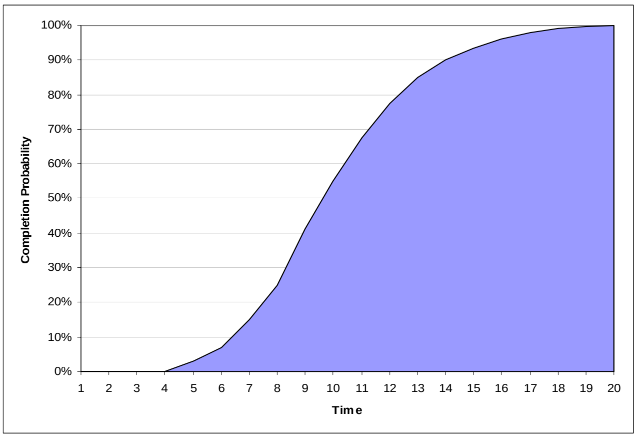
University of Toronto

Outline

- Fundamentals, Related Work
  - Software Estimation
  - Judgmental Biases, Anchoring and Adjustment
- Software Estimation Experiment
  - Plan, Execution
  - Results
  - Follow-up Study
- Conclusions

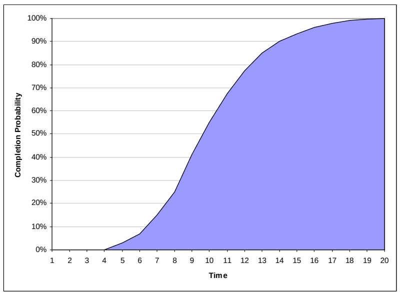
Software Estimation
What is it?

- Project completion probability distribution



Software Estimation
What is it?

- Estimate: Prediction of effort needed to complete a project
  - Prediction has a probability *p* of being above real effort
    - Researchers aim for balance (*p* = 50%)
    - Estimators fall in optimism (*p* just above 0%)
    - Managers assume certainty (*p* = 100%)



Software Estimation Techniques

- Model-based techniques
  - COCOMO, SLIM, ESTIMACS, Checkpoint
  - Default academic idea of what estimation should do
  - Assumption: Software development fits into a general model; model's equation can be found
  - Core: Size-effort correlation
  - Note: People are better at estimating effort than size
  - Results: Poor, although calibration is helpful
- Learning-oriented techniques
  - Analogies, neural networks
  - Assumption: Past performance is good indication of future performance
  - Results: Good for known territory, bad otherwise

Software Estimation Techniques

- Expert-based techniques
  - Individual estimation, Delphi
  - Assumption: Humans handle uncertainty better than models/tools
  - Bad reputation in academia
    - Frequently thought of as mere "guessing"
    - Boehm doesn't even consider freeform individual expert estimation as an estimation technique
  - Widespread use in industry
    - Surveys indicate 62%-85% use expert estimation primarily (compare to <10% primary use of models)

Software Estimation Techniques

- Isn't all estimation expert-based?
  - Models require human judgment for input
    - Estimated size of application
    - Relevance of situational parameters (team experience, familiarity with problem domain, etc.)
  - Analogy-based estimation requires picking sources for analogy
    - Humans are currently better than tools at choosing analogies
  - Model and analogy-based estimates are normally adjusted if they don't "feel" right
- If human judgment is always required, we should connect to research in psychology

Software Estimation

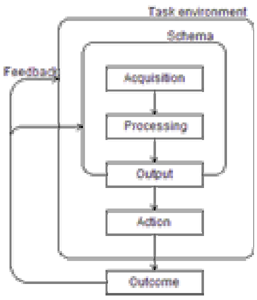
- Brown & Siegler: "Psychological research on real-world quantitative expert estimation has not culminated in any theory of estimation, not even in a coherent framework for thinking about the process".
- But there are results from human judgment research we can use

Software Estimation and Human Judgment

- Some results linking software estimation and human judgment:
  - Estimators do not distinguish between 50%, 75%, 90% and 99% confidence in their estimates
  - Managers prefer estimators that give narrow estimation ranges, even if they are wrong
  - Customer expectations play a role in the outcome of an estimation process
  - Experience is not a good indicator of accuracy
  - Estimates are a factor in actual effort of projects (self-fulfilling prophecies)

Judgmental Biases

- Judgmental bias: Deviation from reality that prevents the objective consideration of a situation
- Hogarth's conceptual model of judgment



Judgmental Biases

- Acquisition biases
  - Availability
    - Does the letter *R* appear more frequently in the first or in the third position of English words?
  - Selective perception
    - We perceive information we expected to perceive, and disregard conflicting evidence
  - Concrete information
    - Direct advice is given more thought than abstract information

Judgmental Biases

- Information processing biases
  - Inconsistency
    - Difficulty to apply the same criterion to a repetitive set of cases
  - Representativeness
    - When classifying a piece of information, we assign it to the class on which it typically belongs, not in which it statistically belongs
  - Worthless data
    - No specific data at all is better than worthless data

Judgmental biases

- Information processing biases (cont.)
  - Law of small numbers
    - Which sequence of coin tosses is more likely; six heads in a row or H-T-T-T-H-T?
  - Regression
    - “Student performance improves after a reprimand, and worsens after a reward”
  - Groupthink
    - Groups may take decisions no group member would have taken individually
  - Anchoring and adjustment
    - (We’ll come back to it in a moment!)

Judgmental Biases

- Output biases
  - Scale effects
    - Probabilities are assigned differently when required as percentages than as x:y odds
  - Illusion of control
    - Planning and forecasting induce feelings of control over the uncertain future
- Feedback biases
  - Overconfidence
    - Practice (and lack of proper feedback) causes an increase in confidence, without an increase in actual performance
  - Hindsight bias
    - In retrospect people are rarely surprised of the outcome of a previously uncertain situation

Anchoring and Adjustment

- Tversky & Kahneman's roulette experiment
  - Low anchor (10) leads to low estimate (25%)
  - High anchor (65) leads to high estimate (45%)
- If judgment is difficult we appear to grasp an *anchor* (a tentative, even if unlikely, answer) and *adjust* it up or down according to our intuition
- Adjustment is frequently insufficient to compensate anchor

Anchoring and Adjustment

- Evidence exists for anchoring and adjustment in wide variety of activities
  - General knowledge issues
  - Probability estimates
  - Legal judgment (ask for large compensations!)
  - Real estate pricing decisions
  - Negotiation
- Anchor does not need to be related to solution
  - However, semantic anchoring effects are more potent than purely numeric anchoring

Anchoring and Adjustment

- No thorough explanation for phenomenon, but:
  - It occurs if people pay sufficient attention to anchor
  - Knowledgeable people are less susceptible
  - Anchoring appears to operate unintentionally (it is difficult to avoid even when people are forewarned)

Anchoring and Adjustment in Software Estimation

- Software estimation is a prime candidate for anchoring effects:
  - Judgment under lots of uncertainty
  - Quantitative estimates
  - Anchors are happily tossed among managers and developers
    - “Do you think you’ll finish by mid February?”
  - Lack of solid framework for software development makes it easy to justify biased estimates

Anchoring and Adjustment in Software Estimation

- Relevant recent research
  - Customer expectations may play a role in estimates
  - Anchoring and adjustment biases assignment of work hours to Work Breakdown Structure analyses

Software Estimation Experiment Research Questions

- Does the phenomenon of anchoring and adjustment influence software estimation processes?
- Is the influence of anchoring and adjustment stronger for estimators that rely solely on expert estimation?
- Does the confidence (or lack thereof) estimators have in their answers compensate for possible anchoring and adjustment biases?
- Is the anchor effect stronger around anchors that naturally attract estimates due to business cycles –such as "12 months"?

Software Estimation Experiment
Experiment Design

- Experiment consisted of a software estimation exercise
  - Problem: Estimate how long will it take to deliver a software application based on:
    - Initial requirements specification
    - Client and development team situational information
    - Approximately 10 pages of material
  - Participants work on problem individually
    - Can take as long as they desire
    - Can use estimation technique(s) of their choice
  - Required answers:
    - Estimate in months
    - Justification
    - Confidence range (in percentage)

Software Estimation Experiment
Experiment Design

- In documentation, future user of system is quoted as saying one of *(emphasis added here)*:
  - "I'd like to give an estimate for this project myself, but I admit I have no experience estimating. We'll wait for your calculations for an estimate."
  - "I admit I have no experience with software projects, but I guess this will take about **2 months** to finish. I may be wrong of course, we'll wait for your calculations for a better estimate."
  - "I admit I have no experience with software projects, but I guess this will take about **12 months** to finish. I may be wrong of course, we'll wait for your calculations for a better estimate."
  - I admit I have no experience with software projects, but I guess this will take about **20 months** to finish. I may be wrong of course, we'll wait for your calculations for a better estimate."
- All other data were equal among conditions

Software Estimation Experiment
Experiment Design

- Note that:
  - Difference among extreme anchors is an order of magnitude
    - Difference is large, but plausible considering range of estimates at early project stages
  - Anchor is semantically linked to problem
  - User does not push his guess as a starting point for negotiation
    - He labels his own estimate as a guess
  - Participants *read* the quote, did not *hear* it coming from a customer
    - Less likelihood of attempting to please user (social bias)

Software Estimation Experiment
Execution

- 29 participants
  - 62% graduate students, 38% software professionals
  - 62% with previous experience
  - 34% with experience in medium to large projects (self-assessed)
- Intended even distribution among conditions
  - 9 responses for “2 months” condition
  - 6 responses for “12 months” condition
  - 8 responses for “20 months” condition
  - 6 responses for control condition

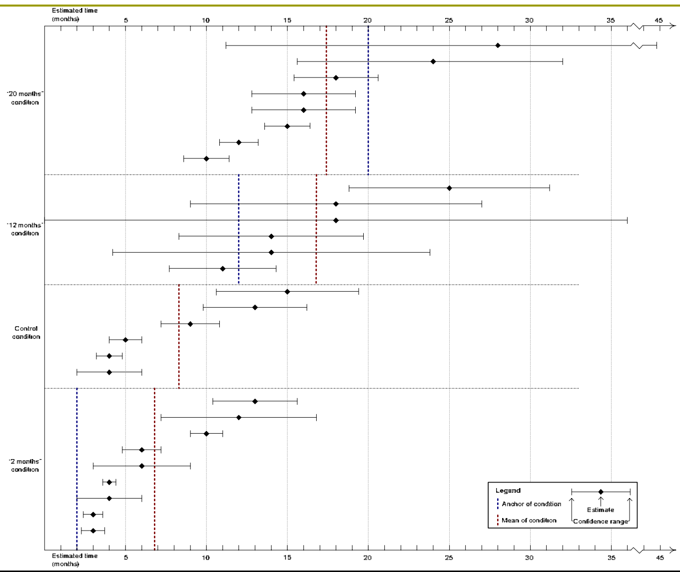
Software Estimation Experiment General Results

- Very wide range of estimates
  - Shortest estimate: 3 months
  - Longest estimate: 28 months
  - Average estimate: 12.1 months
- Confidence limits increase range to:
  - Minimum: 2 months
  - Maximum: 44.8 months
- Average +/- confidence percentage: 31%
  - Minimum: 10%
  - Maximum: 100%

Software Estimation Experiment General Results

- Primary estimation techniques used:
  - Expert-based estimation (72%)
    - WBS analysis: 45%
    - Intractable process: 27%
  - Model-based estimation (28%)
    - Lines of code: 18%
    - Function points: 10%

Software Estimation Experiment General Results



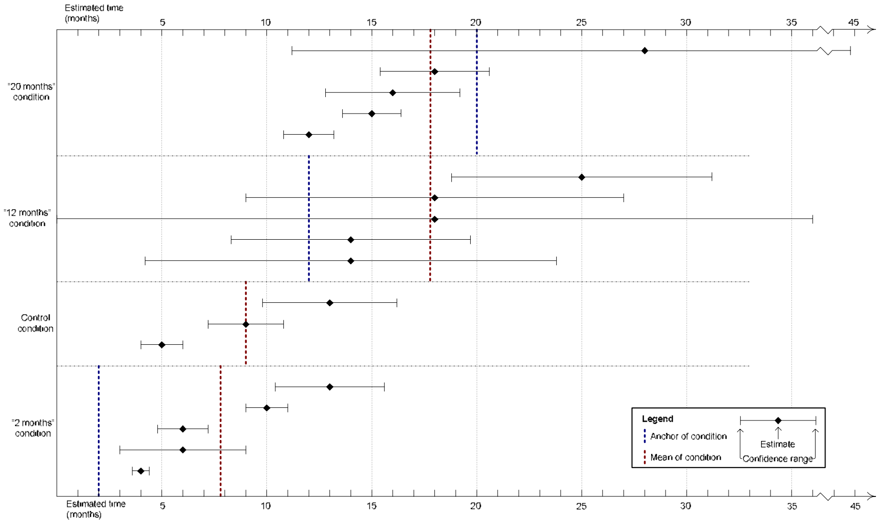
Software Estimation Experiment General Results

	"2 months"	Control	"12 months"	"20 months"
Mean	**6.8**	**8.3**	**16.7**	**17.4**
Median	6	7	16	16
Std. Dev.	3.7	4.4	4.5	5.6

Software Estimation Experiment General Results

- Estimates from the “2 months” condition are significantly different from those in the “20 months” condition ($p < 0.001$)
- Estimates from the control condition are significantly different from those in the “20 months” condition ($p < 0.01$)
- Estimates from the “2 months” condition were not found to be significantly different from those in the control condition ($p > 0.1$)
- Estimates from the “12 months” condition are significantly different from those in the “2 months” condition ($p < 0.01$) and from those in the control condition ($p < 0.05$), but not from those in the “20 months” condition ($p > 0.1$)

Software Estimation Experiment Experienced Participants Results



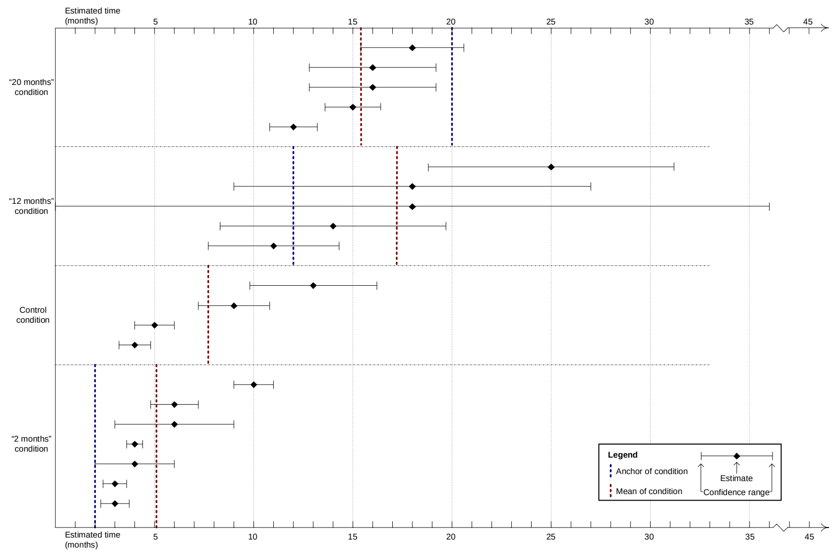
Software Estimation Experiment Experienced Participants Results

	"2 months"	Control	"12 months"	"20 months"
Mean	**7.8**	**9.0**	**17.8**	**17.8**
Median	6	9	18	16
Std. Dev.	3.2	3.3	4.02	5.5

Software Estimation Experiment Experienced Participants Results

- Estimates from the "2 months" condition are significantly different from those in the "20 months" condition ($p<0.02$)
- Estimates from the control condition are significantly different from those in the "20 months" condition ($p<0.05$)
- Estimates from the "2 months" condition were not found to be significantly different from those in the control condition ($p>0.1$)
- Estimates from the "12 months" condition are significantly different from those in the "2 months" condition ($p<0.01$) and in the control condition ($p<0.05$), but not from those in the "20 months" condition

Software Estimation Experiment
Expert-based Techniques Results



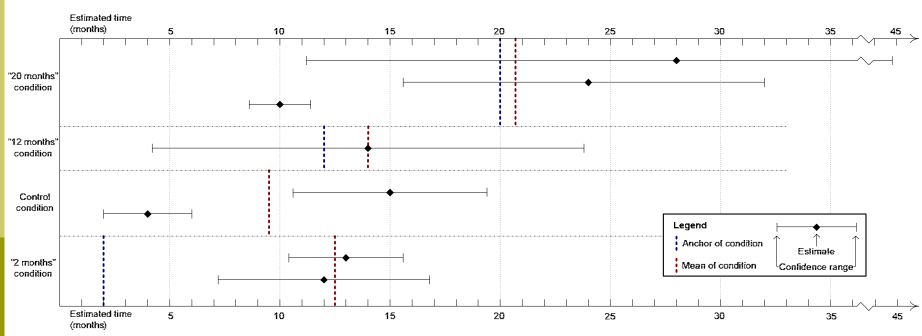
Software Estimation Experiment
Expert-based Techniques Results

	“2 months”	Control	“12 months”	“20 months”
Mean	**5.1**	**7.8**	**17.2**	**15.4**
Median	4	7	18	16
Std. Dev.	2.3	3.6	4.7	2.0

Software Estimation Experiment
Expert-based Techniques Results

- Estimates from the “2 months” condition are significantly different from those in the “20 months” condition ($p<0.001$)
- Estimates from the control condition are significantly different from those in the “20 months” condition ($p<0.02$)
- Estimates from the “2 months” condition were not found to be significantly different from those in the control condition ($p>0.1$)
- Estimates from the “12 months” condition are significantly different from those in the “2 months” condition ($p<0.001$) and from those in the control condition ($p<0.05$), but not from those in the “20 months” condition

Software Estimation Experiment
Model-based Techniques Results



Software Estimation Experiment Model-based Techniques Results

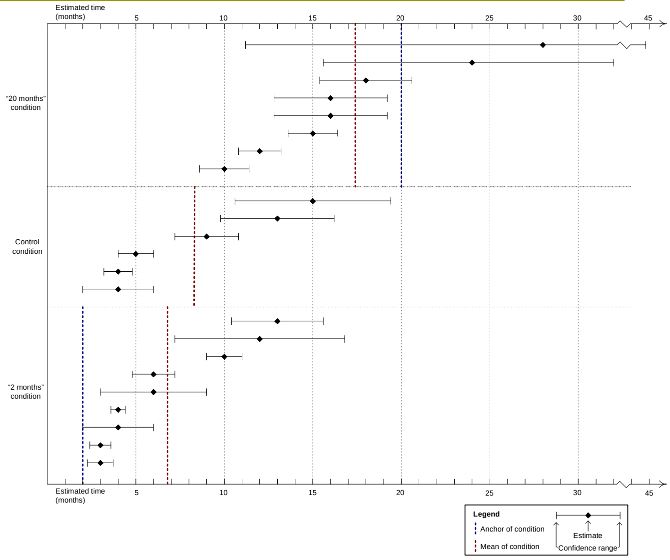
	"2 months"	Control	"12 months"	"20 months"
Mean	**12.5**	**9.5**	**14**	**20.7**
Median	12.5	9.5	14	24
Std. Dev.	0.5	5.5	n/a	7.7

Software Estimation Experiment Model-based Techniques Results

- No comparison between conditions was found to be statistically significant ($p > 0.05$ in all cases)

Software Estimation Experiment
General Results -2 – 20 months diff.

Consider the maximum (pessimistic) values on the "2 months" condition and the minimum (optimistic) values on the "20 months" condition...



Software Estimation Experiment
Maximum-Minimum Results

	"2 months" maximums	Control	"20 months" minimums
Mean	**8.7**	**8.3**	**12.8**
Median	7	7	13
Std. Dev.	4.8	4.4	2.2

Software Estimation Experiment Maximum-Minimum Results

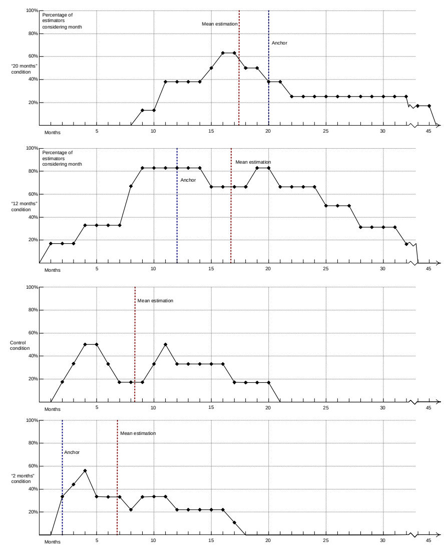
- **Maximum** values of estimates from the “2 months” condition are significantly different from **minimum** values of estimates in the “20 months” condition ($p<0.05$)
- Estimates from the control condition are significantly different from **minimum** values of estimates in the “20 months” condition ($p<0.1$)
- Maximum estimates from the “2 months” condition were not found to be significantly different from those in the control condition ($p>0.1$)

Software Estimation Experiment Estimate Ranges Results Concentrated

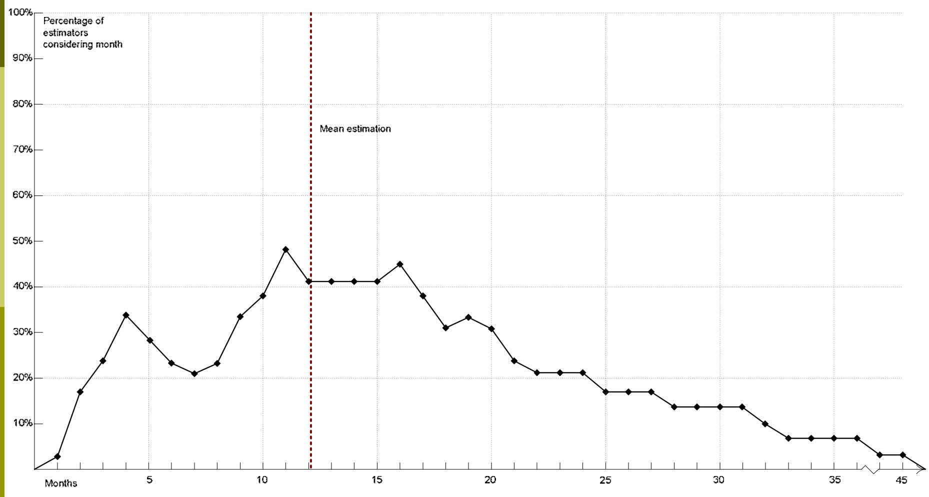
The figure to the right shows the percentage of agreement that participants in each condition had with each other.

From bottom-up, the groups are “2 months”, control, “12 months” and “20 months” conditions.

The “12 months” condition had higher ranges than usual, achieving the highest intra-group agreement, with 83%



Software Estimation Experiment
Estimate Ranges Results Concentrated



Software Estimation Experiment
Estimate Ranges Results Concentrated

- All estimators worked on the same problem
  - Maximum agreement was 48%
  - Therefore, for any outcome of project, at least 52% of estimates will be wrong

Conclusions

- Anchoring and adjustment does take place in software estimation processes
  - Strength of bias too high to be ignored
  - Results from low anchors are statistically different from high anchors
  - Results from estimates without anchors are statistically different from high anchors
- No statistical difference found between low anchors and control condition
  - Estimators optimistic/attempting to please by default?
  - Incorrect choice for low anchor?
  - More participants necessary to discover effect?

Conclusions

- No statistical difference found between "12 months" and "20 months" anchors
  - Both anchors high enough for project?
  - "12 months" group was extracted differently (same company, possibly same business values) than the other three
    - "12 months" had an average range of error of 53%, against 23-33% on other groups
- No effect of "12 months" *natural attractor* was apparent.

Conclusions

- Anchoring and adjustment effects unchanged with experienced estimators
- Stronger effect for estimators using expert-based techniques
- Model-based estimations scarce (28%), bias effect inconclusive
  - Use of model-based techniques in line with surveys
  - 55% of inexperienced estimators chose a model-based technique
  - 11% of experienced estimators chose a model-based technique

Conclusions

- What to do?
  - Shield estimators from anchors
    - Not always possible
  - Give estimates with wide min-max ranges
    - However, management will think you are inexperienced
  - Choose a development lifecycle in which estimates are less relevant and risk is managed
    - Spiral model better than waterfall