Cognitive Limits of Software Cost Estimation

Ricardo Valerdi
Massachusetts Institute of Technology
rvalerdi@mit.edu

Abstract

This paper explores the cognitive limits of estimation in the context of software cost estimation. Two heuristics, representativeness and anchoring, motivate two experiments involving psychology students, engineering students, and engineering practitioners. The first experiment, designed to determine if there is a difference in estimating ability in everyday quantities, demonstrates that the three populations estimate with relatively equal accuracy. The results shed light on the distribution of estimates and the process of subjective judgment. The second experiment, designed to explore abilities for estimating the cost of software-intensive systems given incomplete information, shows that predictions by engineering students and practitioners are within 3-12% of each other. The value of this work is in helping better understand how software engineers make decisions based on limited information. The manifestation of the two heuristics is discussed together with the implications for the development of software cost estimation models in light of the findings from the two experiments.

1. Introduction

The process of estimating the cost of software has been of interest to researchers for decades. Some have developed sophisticated algorithms calibrated with historical data to improve the estimation process [1, 2, 3]. Others have found ways to combine different estimation methods such as bottoms up and analogy to arrive at estimates with a high degree of confidence [4, 5]. While this research has helped shift the field of software cost estimation from an art to more of a science, the process of estimation remains prone to human errors and biases. These can be especially problematic when there is little information available about the people, technologies, development environment, and process used for developing software.

Even in the face of missing information, humans make assumptions that help them develop software cost estimates. While these assumptions are not always justified, they certainly influence the outcome and accuracy of software cost estimates. The fields of human decision making and cognitive science help to further inform this issue.

Tversky and Kahneman [6] proposed that many human decisions are based on beliefs concerning the likelihood of uncertain events. Occasionally, beliefs concerning uncertain events are expressed in numerical form as odds or subjective probabilities. Their work showed that people rely on a limited number of heuristic principles which reduce the complex task of assessing probabilities and predicting values to simpler judgmental operations. Many heuristics exist in software engineering [7]; arguably the most popular one in software cost estimation is the cube root law [8] which contends that the software development time in calendar months is roughly three times the cube root of the estimated effort in person-months provided by a model like COCOMO II. This paper does not focus on technology-based heuristics, but rather on decision making heuristics that rely heavily on subjective assessments by software engineers.

The subjective assessment of probabilities resembles the subjective assessment of physical quantities such as distance or size. For example, the apparent distance of an object is determined in part by its clarity. The more sharply the object is seen, the closer it appears to be. Similarly, in software engineering, the cost of developing software often depends on the intuitive judgments by the stakeholders involved relative to their point of view.

It is proposed that two heuristics developed by Tversky and Kahneman [6] can shed light into the process of decision making in software cost estimation. The first is *representativeness* which is based on the concept that people are concerned with the degree to which *A* is representative of *B*. The symbol *A* could represent a completed software project and *B* could be a new project being estimated. The experiments described in this paper are influenced by this heuristic which is manifested in the context of predictions of every day values and software-intensive systems.

A second heuristic proposed by Tversky and Kahneman is called *anchoring* which is concerned with the ability for people to make an estimate by starting from an initial value that is adjusted to yield the final answer. The initial value, or starting point, may be suggested by the formulation of the problem, or it may be the result of a partial computation. This heuristic has been studied in the context of software processes [9] and has been found to influence both upward and downward adjustments under controlled experiments of software estimates [10]. The second experiment described in this paper is motivated by this heuristic and demonstrates the convergence of cost estimates as a function of life cycle phases.

1.1 Research Questions

Cognitive science deals with any kind of mental operation or structure that can be studied in precise terms [11]. It is well known that humans have motives, drives, and are limited in knowledge and capacity to learn, solve problems, and make decisions. The processes of "how" decisions are made are adequately captured by the aforementioned heuristics and associated theories. But there is little understanding about "how well" specific populations are able to make decisions. This leads to the following research question:

> *How accurate are software engineers at estimating future values given limited information?*

In order to test accuracy, the population of software engineers is compared to other populations to determine their relative ability to estimate. A derivative of this question deals with the preference of information for decision making, namely:

> *How much do engineers rely on historical data versus a cost model to perform cost estimates?*

The exploration of these questions informs the field of software cost estimation on two fronts. First, it provides empirical evidence to help better understand the cognitive limits of software engineers in terms of their ability to estimate. Second, it allows for a comparison between software engineers and other populations; technical and non-technical as well as student and practitioner. The results provide insight into the ability of software engineers to estimate certain phenomena.

2. Methods

Two experiments were conducted to assess the ability of participants to estimate common quantities as well as the duration of development for a software-intensive system given an elapsed period of time. The first experiment was inspired by previous work on optimal predictions in everyday cognition [12] but was extended to the area of software engineering by applying the idea of cognitive estimation limits to the area of software-intensive systems. The original set of questions was kept the same so that data from previous studies could be compared to newly obtained data. Results were obtained for this experiment through the use of a survey instrument provided in Appendix A. The second experiment involved only engineering students and practitioners since it was intended to assess the ability of participants to estimate the duration, in person months, of the development of a software-intensive system.

2.1 Participants

Participants represent three different populations, each of them making predictions about different phenomena. The first population, made up of 142 undergraduate students, participated in the experiment as part of a psychology class and is referred to as *psychology students* throughout the paper. The second population, made up of 36 graduate-level engineering students, participated in the experiment as part of a lecture in a project management class and is referred to as *engineering students* throughout the paper. The third population, made up of 49 software and system cost estimation professionals, participated in the experiment as part of a day-long workshop on cost estimation and is referred to as *practitioners* throughout the paper. The engineering students had anywhere between 0-2 years of work experience in cost estimation whereas the practitioners have an average of 12 years and were familiar with advanced cost estimation principles.

2.2 Description of Experiment #1

The first experiment was conducted by giving individual pieces of information to each of the participants in the study, and asking them to draw a general conclusion. For example, many of the participants were told the amount of money that a film had supposedly earned since its release, and asked to estimate what its total "gross" would be, even though they were not told for how long it had been playing. In other words, participants were asked to predict t_{total}

given t_{past}. No additional information was given about the film such as the genre, country of origin, actors, or production studio.

In addition to the returns on films, the participants were asked about things as diverse as the number of lines in a poem (given how far into the poem a single line is), an individual's life span (given his current age), the duration of a Pharaoh's reign (given he had reigned for a certain time), the run-time of a film (given an already elapsed time), the total length of the term that would be served by an American congressman (given how long he has already been in the House of Representatives), the time it takes to bake a cake (given how long it has already been in the oven), and the amount of time spent on hold in a telephone queuing system (given an already elapsed time). All of these items have known values and well-established probability distributions. The intent of the experiment was to determine whether there was any difference in the composite answers of each population. The eight questions are provided in Appendix A, Part I.

2.3 Description of Experiment #2

The second experiment was conducted in a similar fashion except it only involved the engineering students and practitioners because of the technical content. The focus was to capture the estimation tendencies of the populations given a limited amount of information. The first part of the experiment contained questions about the expected duration of a software-intensive project given an elapsed period of time. Participants were given four system life cycle phases to use as their mental model: conceptualize, develop, operational test & evaluation, and transition to operation. Similar to experiment 1, no additional information was given about the project such as application domain, development organization, or historical performance. Participants were asked to predict the total effort needed for a project, t_{total}, given a certain amount of effort had already been expended on one or more life cycle phases, t_{past}. In the first question, t_{past} = 300 person months for the Conceptualize phase. In the second question, t_{past} = 300 person months in the Conceptualize and Develop phases. In the third question, t_{past} = 300 person months for the Conceptualize, Develop, and Operational Test & Evaluation phases. The three questions are provided in Appendix A, Part II.

The second part of the experiment asked participants to predict the total systems engineering effort for a software-intensive system, t_{total}, given the predicted effort from a cost model, $t_{predicted}$, and a historical data point, $t_{historical}$, from a similar system of equivalent scope and complexity. A relatively new cost model, COSYSMO, was selected for this experiment to avoid any unbalanced expertise from practitioners. Moreover, both the engineering students and the practitioners received an initial tutorial on the use of COSYSMO and its definitions to ensure that there was a minimum level of knowledge across the populations. In the first question, $t_{predicted}$ = 100 person months and $t_{historical}$, = 110 person months. In the second question, $t_{predicted}$ = 1,000 person months and $t_{historical}$, = 1,100 person months as shown in Appendix A, Part III.

3. Results

The predictions about everyday events by the three populations were on the whole extremely accurate. The results of the responses from the psychology students are provided in Figure 1.

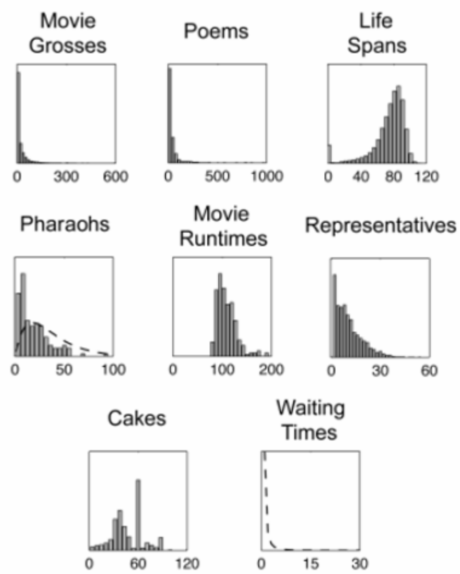


Figure 1. Relative Probabilities of t values for Psychology Students, n = 142 [12]

The distributions for movie grosses and poems are approximately power-law which accurately indicates that the majority of movies gross very little money but there are a few which become blockbuster hits. For example, out of over 7,300 films worldwide from the period 1900-2006 only three films grossed over $1B. Similarly, the majority of poems are very short but there are a few which are very long.

Of particular interest is the similarity in the distribution of the answers across the three populations and the proximity in the mean values for t_{total}. The psychology students and engineering students were equally accurate in estimating t_{total} for the eight questions in the first experiment compared to the practitioners as shown in Table 1.

Table 1. Mean Values of Results for Experiment 1

	Psychology Students (n = 142)	Engineering Students (n = 36)	Practitioners (n = 49)
Movie Grosses (in Millions)	40	41	42
Poems (lines)	22	20	21
Life Spans (years)	76	73	78
Pharaohs (years)	30	23	23
Movie Runtimes (Minutes)	120	105	108
Representatives (years)	18	21	22
Cakes (minutes)	53	48	50
Waiting times (minutes)	10	7	9

When it came to estimating t_{total} for the scenarios presented in the second experiment, there was a negligible difference between engineering students and practitioners as shown in Table 2. Note that the standard deviation is shown in brackets below the mean value. The number of samples differs slightly from experiment 1 because of missing data from one participant.

Table 2. Mean and Standard Deviation of Results for Experiment 2

	Engineering Students (n = 36)	Practitioners (n = 48)
Through one phase (PM)	1516 *[1011]*	1386 *[758]*
Through two phases (PM)	666 *[266]*	594 *[241]*
Through three phases (PM)	401 *[129]*	390 *[145]*
Project X (PM)	112 *[7]*	110 *[9]*
Project Y (PM)	1140 *[128]*	1122 *[111]*

The difference in estimates for t_{total} between engineering students and practitioners for the first three questions was 9%, 12%, and 3%, respectively. Interestingly, engineering students estimated consistently higher than the practitioners in all three scenarios. However, the mean values of their estimates were very close considering the small amount of information provided to both populations in the survey. The coefficients of variation for the three scenarios were 0.66, 0.39, and 0.32 for the engineering students and 0.55, 0.41, and 0.37 for the practitioners. This indicates that both populations followed a similar pattern of increased intra-group agreement indicated by a reduction of the standard deviation of the distribution of their answers relative to the mean of the distribution.

The results from the second part of experiment #2, also displayed in Table 2, show that the difference in estimates for t_{total} between engineering students and practitioners was 2% for both scenarios. Engineering students again estimated consistently higher than the practitioners but this was relatively negligible considering the amount of information that was provided to them. Coefficients of variation were 0.06 and 0.11 for the engineering students and 0.08 and 0.09 for the practitioners which demonstrate an equivalent set of responses from both populations.

4. Analysis

The two experiments performed shed light on the estimation accuracy of the three populations. The psychology students served as a control group for comparing engineering student's and practitioner's ability to estimate every day values. As the results from the first experiment show, all three populations predicted values of every day events with relatively equal accuracy with the exception of the Pharaoh question. Both the magnitude of errors and the variance in judgments across participants were substantially greater for this question than for the other questions. A Pharaoh is a title used to refer to any ruler, usually male, of the Egyptian kingdom in the pre-Christian, pre-Islamic period. Compared to other questions in the survey, which were of more contemporary tone, participants would typically not be aware of the typical rule of Egyptian rulers thousands of years ago. Therefore, they must depend on their judgment of present day events to produce an estimate.

Despite the lack of direct experience, the predictions of each population were not completely off the mark: Their judgments were consistent with having implicit knowledge of the correct form of the underlying distribution but making incorrect assumptions about how this form should be parameterized (i.e., its mean value). The predictions for the reigns of Pharaohs suggest a general strategy people might employ to make predictions about unfamiliar kinds of events, which is surely an important prediction problem faced in everyday life. Given an unfamiliar prediction task, people might be able to identify the appropriate form of the distribution by making an analogy to more familiar phenomena in the same broad class, even if they do not have sufficient direct experience to set the parameters of that distribution accurately. This phenomenon is what is precisely described by the *representativeness* heuristic. By estimating by analogy, participants were able to approximately guess the mean length of a Pharaoh's reign. However, the analogy method is inaccurate when knowledge and experience are obstacles to the process as is often the case with software cost estimation. Large databases of historical projects may be available for use in estimation by analogy method but, when the context of the projects is not known, the utility of the projects may be overestimated and may actually lead to inaccurate conclusions about the applicability of the current project.

The results from the second experiment make it very clear that the cone of uncertainty is decreasing. The distribution of predictions of t_{total} by engineering students and practitioners decreased as the system life cycle progressed. In other words, as more of the project was complete, the smaller the standard deviation of responses for t_{total}. These results confirm previous hypotheses about a software engineering phenomenon referred to as the cone of uncertainty [13, 14]. Responses from the three stages have been plotted and rotated ninety degrees to the right to demonstrate the visual convergence of results. The responses from engineering students, shown in Figure 2, have a higher variance compared to the responses from practitioners, shown in Figure 3.

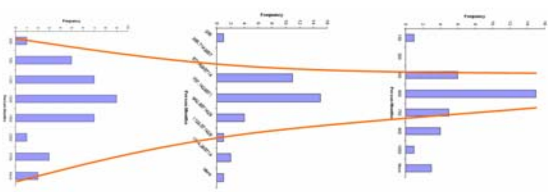


Figure 2. Engineering Student Estimates for Three Scenarios, n = 36

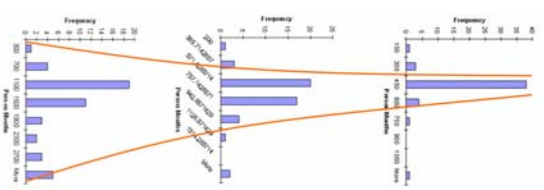


Figure 3. Practitioner Estimates for Three Scenarios, n = 48

Results from the second experiment also show that engineering students slightly overestimate compared to practitioners. The overestimation is even more apparent when the distributions of responses are visually compared. Even though the distributions are approximately Gaussian and the mean values are within 3-12% of each other, the variance of responses from the engineering students is slightly higher.

Results from the final part of the second experiment, where two scenarios are provided and participants are asked to estimate t_{total} given $t_{predicted}$ from a model and $t_{predicted}$ from historical data. This is an example of the case vs. base estimation where people often place undue weight on specific a specific example (case) and insufficient weight on more global sources (base-rate) even when the latter are highly predictive [15]. Both populations tended to ignore COSYSMO in the presence of a historic case, showing that the *representativeness* heuristic wins big again.

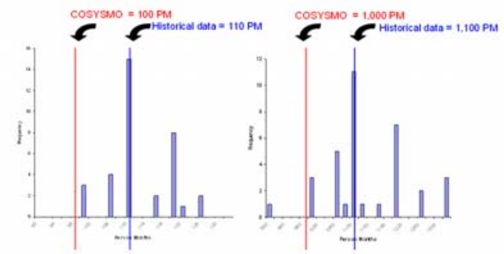


Figure 4. Engineering Student Estimates for Three Scenarios, n = 36

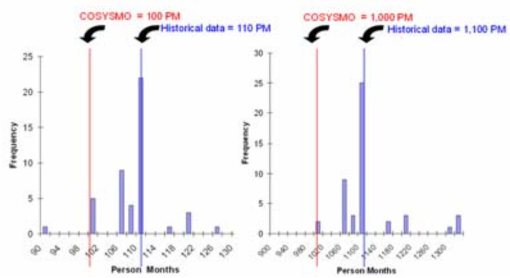


Figure 5. Practitioner Estimates for Three Scenarios, n = 48

Both the engineering students (Figure 4) and the practitioners (Figure 5) demonstrated a bias towards historical data and even overestimated the effort estimate despite the information provided. The responses from the engineering students were more distributed than the practitioners as observed in other sections of the experiment.

4.1 Threats to Validity

As discussed in other empirical software engineering studies [16], it is necessary to identify possible threats to validity that could bring into question the experiment and its results.

The execution of the experiment itself could affect the internal validity of this study. Namely, the survey administration for the psychology students was performed by one set of researchers while the survey administration for the engineering students and practitioners was performed by another. While this was not done deliberately it could affect consistencies in survey administration and potentially affect the quality of the results due to the difference in experimental setting.

Another experimental threat is that the survey participants, when given the set of questions to answer, were trying hard to find the right answer because they may have perceived this as a test of intelligence. This is a well known effect in educational measurement and is often referred to as the Pygmalion effect.

One aspect that would make this experiment feel quite different than a real world situation is that motivation for participating is very different than in a real project. Therefore, the biases may not be as visible in the experiment, especially for practitioners. This could also explain the chronic overestimation by both populations in experiment #2.

Despite the healthy sample size, the survey was not distributed to a representative sample of software engineers. Quite the contrary, the practitioners that participated are known to be involved in several process improvement initiatives. They are also employed by organizations which have traditionally motivated their employees to follow a high degree of process maturity. This could be considered a biased sample because of the tendency to be familiar with mature practices and, as a result, could severely affect the external validity of the results.

The sample of students was also not random. The students that participated in the experiments were undergraduate psychology students and graduate engineering students. Both are considered to be highly motivated and educated compared to the normal population and therefore could have know the correct answer to the questions being asked. It is less likely that they did not know the answer since they could have provided an "educated guess" which was likely to be relatively accurate.

Even with these known issues of internal and external validity, it is believed that the results of the experiment are informative to the questions at hand since the populations are likely to become decision makers in software organizations in the future.

5. Discussion

Empirical data has been provided to explore the estimation accuracy of software engineers compared to two student populations. On the whole, judgments of everyday quantities such as movie times and life expectancy were quite accurate and exhibited known distribution profiles. Other everyday quantities, such as the reign of Pharaohs, were not as precise but nevertheless provided insight into the heuristics used by people to arrive at quantities of unfamiliar topics.

Much work is left to do in understanding the underlying reasons why people can turn observed coincidences into heuristics. Somehow, the human mind is capable of acquiring useful knowledge about the world and employing rational statistical mechanisms to make predictions about future occurrences. The exploration of these concepts in software engineering can lead to future theories and hypotheses that will further inform how people use their cognitive abilities to make judgments.

5.1 Implications

Two main implications result from these experiments. First, it was shown that students are equally good estimators compared to practitioners, although they tend to overestimate perhaps because of their inexperience working on real programs. But the consistency in their responses supports the argument in favor of the suitability of students as subjects for software engineering experiments [17, 18, 19, 20]. While students are not ideal for all types of experiments, they have proven to be adequate participants for experiments in cost estimation.

Another important implication of this work is the fact that both populations were influenced more by historical information than by the answer provided by the cost model. Furthermore, participants in the second experiment overestimated the effort needed to develop a system despite the historical data provided. Cost modeling research should continue to work towards the development of sophisticated models but should note that software engineers will not depend on the answer provided by the models alone. They will incorporate historical data, their own heuristics based on past observations, and personal biases regarding the situation at hand. These heuristics and biases need to be considered not only from a technological standpoint [21] but also from a cognitive standpoint in order to fully understand and control them.

6. References

[1] J. Bailey, V. Basili, "A Meta-Model for Software Development Resource Expenditures", *Proceedings of the Fifth International Conference on Software Engineering,* March 1981, pp. 107-116.

[2] Boehm, B.W., C. Abts, A.W. Brown, S. Chulani, B. Clark, E. Horowitz, R. Madachy, D.J. Reifer, and B. Steece, *Software Cost Estimation With COCOMO II,* Prentice Hall, 2000.

[3] Putnam, L.H., W. Myers, *Five Core Metrics: The Intelligence Behind Successful Software Management*, Dorset House, 2003.

[4] M. Jorgensen, "Top-Down and Bottom-Up Expert Estimation of Software Development Effort", *Information and Software Technology*, Vol. 46, No. 1, 2004, pp. 3-16.

[5] M. Jorgensen, U. Indahl, and D.I.K. Sjoberg, "Software Effort Estimation by Analogy and Regression Toward the Mean", *Journal of Systems and Software*, Vol. 68, No. 3, 2003, pp. 253-262.

[6] A. Tversky, D. Kahneman, "Judgment Under Uncertainty: Heuristics and Biases", *Science*, Vol. 185, 1974, pp. 1124-1131.

[7] Endres, A., D.H. Rombach, *A Handbook of Software and Systems Engineering: Empirical Observations, Laws, and Theories*, Pearson Addison Wesley, 2003.

[8] D.A. Cook, T.R. Leishman, "Lessons Learned from Software Engineering Consulting", *Journal of Defense Software Engineering*, February 2004.

[9] M. Jorgensen, D.I.K. Sjoberg, "Software Process Improvement and Human Judgement Heuristics", *Scandinavian Journal of Information Systems*, Vol. 13, 2001, pp. 63-80.

[10] J. Aranda, S. Easterbrook, "Anchoring and Adjustment in Software Estimation", *ESEC-FSE*, September 2005.

[11] Lakoff, G., M. Johnson, *Philosophy in the Flesh*, Basic Books, New York, 1999.

[12] T.L. Griffiths, J.B. Tenenbaum, "Optimal Predictions in Everyday Cognition", *Psychological Science*, Vol. 17, No. 9, 2006, pp. 767-773.

[13] McConnell, S., *Software Estimation: Demystifying the Black Art*, Microsoft Press, 2006.

[14] T. Little, "Schedule Estimation and Uncertainty Surrounding the Cone of Uncertainty", *IEEE Software*, Vol. 23, No. 3, 2006, pp. 48-54.

[15] A.S. Goodie, E. Fantino, "Base Rates Versus Sample Accuracy: Competition for Control in Human Matching to Sample", *Journal of the Experimental Analysis of Behavior*, Vol. 71, 1999, pp. 155-169.

[16] A. Jedlitschka, M. Ciolkowski, "Reporting Experiments in Software Engineering", *Fraunhofer Institute for Experimental Software Engineering*, *Technical Report* ISERN-06-01, 2006.

[17] P. Berander, "Using Students as Subjects in Requirements Prioritization", *International Symposium on Software Engineering*, 2004, pp. 167-176.

[18] J. Carver, L. Jaccheri, S. Morasca, and F. Shull, "Issues in Using Students in Empirical Studies in Software Engineering Education", *International Software Metrics Symposium*, 2003, pp. 239-249.

[19] J. Carver, F. Shull, and V. Basili, "Observational Studies to Accelerate Process Experience in Classroom Studies: An Evaluation", *International Symposium on Empirical Software Engineering*, 2003, pp. 72-79.

[20] M. Höst, B. Regnell, and C. Wohlin, "Using Students as Subjects - A Comparative Study of Students and Professionals in Lead-Time Impact Assessment", *Empirical Software Engineering*, Vol. 5, No. 3, 2000, pp. 201-214.

[21] D. Peeters, G. Dewey, "Reducing Bias in Software Cost Estimates", *Journal of Defense Software Engineering*, April 2000.

Appendix A. Survey Instrument

Survey on Intuitive Judgments

Name ________________________ Years of work experience __________

Years of experience in cost estimation (of any kind) __________

What do you consider yourself to be (check all that apply)?

☐	☐	☐	☐	☐
Program Manager	software engineer	hardware engineer	systems engineer	Other_________

Each of the questions below asks you to predict either a duration or a quantity based on a single piece of information. Read each question and write your prediction on the line below it. We are interested in your intuitions, so please don't make complicated calculations. Just tell us what you think.

Part I: 8 questions

1. *Movie Grosses.* Imagine you hear about a movie that has taken in $10M at the box office, but don't know how long it has been running. What would you predict for the total amount of box office intake for that movie? _______

2. *Poems.* If your friend read you her favorite line of poetry and told you it was line 5 of a poem, what would you predict for the total length of the poem? _______

3. *Life Spans.* Insurance agencies employ actuaries to make predictions about people's life spans – the age at which they will die – based upon demographic information. If you were assessing an insurance case for an 18 year old man, what would you predict for his life span? _______

4. *Pharaohs.* If you opened a book about the history of ancient Egypt to a page listing the reigns of the pharaohs, and noticed that at 4000 BC a particular pharaoh had been ruling for 11 years, what would you predict for the total duration of his reign? _______

5. *Movie Runtimes.* If you made a surprise visit to a friend and found that they had been watching a movie for 30 minutes, what would you predict for the total length of the movie? _______

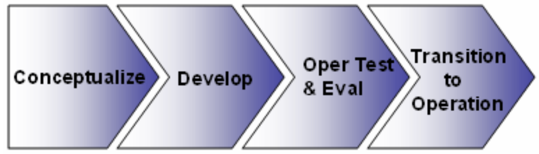
6. *Representatives.* If you heard a member of the House of Representatives had served for 15 years, what would you predict their total term in the House to be? _______

7. *Cakes.* Imagine you are in somebody's kitchen and notice that a cake is in the oven. The timer shows that it has been baking for 35 minutes. What would you predict for the total amount of time the cake needs to bake? _______

8. *Waiting times.* If you were calling a telephone box office to book tickets and had been on hold for 3 minutes, what would you predict for the total time you would be on hold? _______

Part II: 3 questions

These questions require you to be familiar with the four life cycle phases covered in COSYSMO. They are: (1) Conceptualize, (2) Develop, (3) Operational Test & Evaluation, and (4) Transition to Operation



1. *Through one phase of Systems Engineering.* Imagine that a project has taken 300 Person Months of systems engineering effort through the end of the **Conceptualize** phase. What is the total systems engineering effort you predict will be needed to deliver the system (i.e., through the completion of Transition to Operation)? _______

2. *Through two phases of Systems Engineering.* Imagine that a project has taken 300 Person Months of systems engineering effort through the end of the **Conceptualize & Develop** phases. What is the total systems engineering effort you predict will be needed to deliver the system? _______

3. *Through three phases of Systems Engineering.* Imagine that a project has taken 300 Person Months of systems engineering effort through the end of the **Conceptualize, Develop, and OT&E** phases. What is the total systems engineering effort you predict will be needed to deliver the system? _______

Part III: 2 questions

These questions assume that the COSYSMO was used to obtain systems engineering effort estimates.

1. The effort estimate for Project X provided by COSYSMO is **100** Person Months. Historical data from your organization shows that a similar system of equivalent scope & complexity took **110** Person Months to complete. What would you predict for the total systems engineering effort for Project X? _______

2. The effort estimate for Project Y provided by COSYSMO is **1,000** Person Months. Historical data from your organization shows that a similar system of equivalent scope & complexity took **1,100** Person Months to complete. What would you predict for the total systems engineering effort for Project Y? _______

END