

# ICS 121 / Informatics 111: Software Tools and Methods Fall Quarter 2005

## Assignment 1

Due Monday, October 17, 2005, at 11:59pm

### Part 1. Effort Estimation (30 points)

Effort estimation is the practice of making accurate estimates of how long it will take to complete a piece of work. People in software are notoriously bad at this; notice the number of routine projects that run over their schedules and budgets. The objective of this question on this assignment is to build up your skill at making estimates of how long it takes you to complete a task. This question will be repeated on each assignment to ensure that you go into industry with a substantial data set about yourself. The basic plan for this task is to make an estimate of how long it will take to complete the assignment (in advance), record how long you spent on various activities, analyze this log, and calculate your error. This sequence of steps simulates the billing process in a consulting environment.

**1.1 A priori estimate. (2 points)** When you start working on this assignment, write down an estimate of how long it will take you to answer all of the questions. (You will not be penalized for an inaccurate estimate; the purpose of this exercise is for you to become better at effort estimation.) This estimate should be a single number (in hours).

**1.2 Logging. (16 points)** As you are working on the assignment, record what you are doing and how long you spent. For this question, quantity of data, i.e. number of entries, is important. As a rule of thumb, you should add a log entry every time you switch tasks. For example, if you do something for two hours straight, that can be one log entry. However, if you do two or three things in half an hour, you must have a log entry for each of them. You do not need to include time for logging itself, but should include the time spent answering the other sections of Part 1. (Try to avoid infinite recursion!)

Here are some example log entries. Your logging is not limited to the sample entries. Add ones appropriate for your own tasks.

Items	Date	Part No.	Description	Time
1.	10/07/05	All	Read assignment and create estimate	22 min
2.	10/07/05	3	Read Mythical Man-Month	47 min
3.	10/09/05	3	Read Mythical Man-Month	27 min
4.	10/09/05	2	Web searching	30 min
5.	10/10/05	2	Search ANTPAC	20 min
6.	10/12/05	2	Go to library	92 min
7.	10/13/05	2	Read "Software Runaways"	2 h 12min
8.	10/13/05	2	Web searching, collecting information	3h 2min
9.	10/14/05	2	Work on web page	33min
10.	10/14/05	3	Critique Mythical Man-Month	2 h 53 min
11.	10/15/05	2	Work on web page	1 h 10 min
12.	10/15/05	1	Analyze log entries and calculations	33 min
13.	10/17/05	All	Create PDF and Submit assignment	43 min

**1.3 Analysis. (6 points)** For each of the entries in the time log, allocate them to a category. Tally up the time spent in each of the categories and on the entire assignment. Calculate summary statistics and plot a bar graph by category.

The categories that we will use are based in part on the ones used in the Personal Software Process, as designed by Watts S. Humphrey[1]. We will adding some categories, because we're working on tools and methods assignments, which have relatively little programming.

- Project Management
- Implementation
- Testing
- Reading
- Writing
- Research
- Process Improvement

You may add your own categories if you find that many of your activities don't fit well into those provided.

Here's how the example log entries might be broken down.

Items	Description	Time	Category
1.	Read assignment and create estimate	22 min	Project Management
2.	Read Mythical Man-Month	47 min	Reading
3.	Read Mythical Man-Month	27 min	Reading
4.	Web searching	30 min	Research
5.	Search ANTPAC	20 min	Research
6.	Go to library	92 min	Research
7.	Read "Software Runaways"	2 h 12min	Reading
8.	Web searching, collecting information	3h 2min	Research
9.	Work on web page	33 min	Writing
10.	Critique Mythical Man-Month	2 h 53 min	Writing
11.	Work on web page	1 h 10 min	Writing
12.	Analyze log entries and calculations	33 min	Process Improvement
13.	Create PDF and Submit assignment	43 min	Project Management

Here's what a table of summary statistics might look like.

Category	Time	Percentage
Project Management	65 min	7.2%
Reading	3h 26min	22.8%
Research	5h 24min	35.8%
Writing	4h 36min	30.5%
Process Improvement	33 min	3.6%
Total	15h 4min	

[1] Watts S. Humphrey, A Discipline for Software Engineering, Addison-Wesley, 1995.

**1.4 Error Calculation. (2 points)** These next two steps simulate figuring out whether you made a profit on this project. Calculate the percentage error of your initial calculation. (This is the number of hours, or minutes, that you were off, divided by the initial estimate.)

**1.5 Discussion. (4 points)** What did you learn from this exercise? Were there surprises in how long various activities took? Are you good at estimation? Will you use the knowledge you've gained in planning future assignments?

**Deliverables**

- An initial estimate of hours (a single number for this assignment)
- A log of your time spent on the assignment
- A table with the time broken down by category
- A bar graph showing time spent by category
- Comparison of estimate vs. actual
- Discussion of result

## Part 2. Case History of a Spectacular Software Project (40 points)

For this part of the assignment, you will create a web page (or collection of web pages) describing the case history of a spectacular software project. These projects are either spectacular successes, spectacular failures, or have made spectacles of themselves in other ways, e.g. infamy, technical accomplishment.

Case histories are assigned by the LAST DIGIT of your student number. (You may choose a project that is not on this list with the prior approval of the instructor.) A comment has been provided for each project to serve as a starting point for your research or as a suggestion for a particular aspect of the project to investigate.

Last Digit	Project	Comment
0	FBI Virtual Case File	A large IT project that is currently in the news, and not for good reasons. Case history should include information from the report on the Trilogy project by National Research Council of the National Academies <a href="http://books.nap.edu/books/0309092248/html/">http://books.nap.edu/books/0309092248/html/</a>
1	NASA Mars Climate Orbiter	Was it only a numerical conversion error that cause this space probe to crash in to Mars? Or was this just a consequence of the “faster, better, cheaper” approach to space exploration?
2	Google	Track its history from university research to a company with a market capitalization of over \$50 billion and an indispensable tool.
3	FAA Air Traffic Control System	More than two decades later, we still don’t have a replacement system. Are we likely to soon?
4	Denver International Airport baggage-handling system	The dream was a high-speed baggage handling system that could route individual pieces of luggage. The reality fell short, but was this a hardware or software failure?
5	OS/360	“The Mythical Man-Month” describes the various lessons learned by Fred Brooks on this project.
6	Y2K Crisis	There were dire warnings of something terrible happening when the year rolled over. Were people panicking for no reason? Or did companies do a really good job of fixing things? Does that make it a success or failure?
7	SABRE Travel Network	Four decades later and it’s still in wide use. Is this due to a brilliant design, clever maintenance, or an inability to move forward? <a href="http://en.wikipedia.org/wiki/Sabre_%28computer_system%29">http://en.wikipedia.org/wiki/Sabre_%28computer_system%29</a>
8	Space Shuttle Flight Control Software	Article by Fishman covers up to 1996, need to follow-up to present
9	Chrysler Comprehensive Compensation system	XP was first used on this project. <a href="http://www-128.ibm.com/developerworks/java/library/j-beck/">http://www-128.ibm.com/developerworks/java/library/j-beck/</a>

## Requirements for Content

Your web page should be like an encyclopedia entry and include both technical and management details about the project. Your case history must include the following information.

- **Description of the system.** How was the system originally intended to function and how does it actually function?
- **History.** What went wrong (or right) along the way? Examples: schedule over-run, budget over-run, low defect rate, recalls and patches. What is current (or final) status of the projects?
- **Size of system.** Can be quantified in a number of ways – budget, schedule, team size, lines of code, function points. Not all figures will be available for all projects.
- **Causes of success or failure.**
  - **Software technology-** noteworthy uses of software tools, methods, notations, process models, etc.
  - **Non-software factors-** were there managerial, political, or social factors that contributed to the project's success or failure?
- **Lessons learned.** Was the project a success, failure, or other kind of spectacle? Why? What can we learn from the case history?

Since each project is slightly different, some of the items may be less relevant.

The web page should be approximately 1000 words. This is not a size limit, but a figure for you to gauge the amount of work required. The length of a web page can vary depending whether there are figures, sub-pages, and the number of links.

## Requirements for Mechanics

You must include links to a variety of sources, meaning at least three distinct web pages. In this case, distinct means written by different people or from different sites. The same or similar article by one person on two sites are not distinct. Articles by different authors, but from the same site are OK.

You must include a reference to at least one archival source. An archival source is a book, article in a peer-reviewed journal, conference, or magazine. Some good places to look are IEEE conferences and journals and ACM conferences and journals.

You must hand in the web page as a zip file. The file should unzip into a directory with a name with the following format: {your UCInetID}-{project name}. (Drop the curly braces. For example: jmsmith-SpaceShuttleFlightControlSoftware) There should be an index.html file (written in HTML) in the directory, so your page can be linked and indexed easily. Your directory can also contain image files, .css files, or whatever else is necessary.

Aesthetics of the page will be important. (But content is more important.) Do use styles in a consistent and meaningful manner. Don't use too many colors.

### **Part 3. The Mythical Man-Month (30 points)**

Write a critique of Chapter 16, “No Silver Bullet” of The Mythical Man-Month by Frederick P. Brooks. Your critique should be 500-750 words long and consist of a summary of the main points in the chapter and your analysis of Brooks' claims. Here are some questions for you to address: What is essential complexity? What is accidental complexity? What are the four reasons that Brooks gives for why essential complexity is irreducible? Do you think he is right or wrong? Why or why not? Give a concrete, specific example (e.g. the name of a commercial tool) of one software technology that helps with essential complexity and one software technology that helps with accidental complexity. You should provide a pithy and cogent argument for your choice of examples.

Your critique will be graded on grammar, spelling, style, and organization, as well as content and ideas. If you quote from the text of the chapter, you should provide appropriate references.

### **Handing In Your Assignment**

Your assignment must be submitted electronically to [checkmate.ics.uci.edu](http://checkmate.ics.uci.edu). (Informatics 111 students, please send in your assignment through ICS 121.) You will submit three files.

1. A file called Part1.doc or Part1.pdf containing all the written work from Part 1.
2. A zip file with the web page from Part 2 (don't forget the naming convention!).
3. A file called Part3.doc or Part3.pdf containing all the written work from Part 3.

Please note that for parts 1 and 3, only the Word .doc and PDF .pdf formats are acceptable. Do not zip these three files into one big .zip file. Do not use .rar instead of .zip.