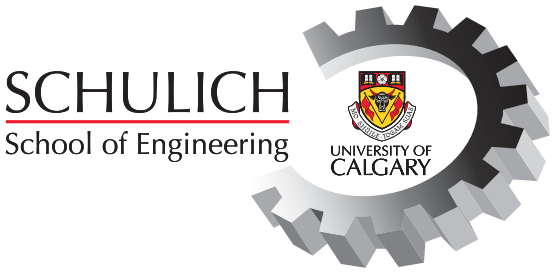

Department of Electrical and Computer Engineering

SENG 421 - Software Metrics
Assignments

Behrouz Far
Fall 2008
(Revision 1.01)

(2008-09-03)

Assignment no. 1

Deadline: September 23, 2008 (Tuesday)
Grade: %20 of the total lab marks

This assignment is an individual assignment composed of two parts. The goal is to increase motivation of the audience to follow the topic throughout the course by presenting them with the most fundamental questions that software metrics tries to answer.

1. Part 1: Information collection and survey

Search articles, textbooks, online materials, etc. to find answer to the following questions:

1. What is software measurement about?
2. Why software measurement is important?
3. How does software measurement affect software quality and productivity?
4. What does empirical investigation mean in the software engineering context?
5. Is software measurement equivalent to software metrics? What makes them different?
6. What are the common software metrics that you already know, heard about or used?
7. What attributes of the software you suggest to be measured?
8. What is software measurement process?
9. How to implement software measurement?
10. What are challenges and difficulties of applying software metrics?

Write your own answers to the above questions. Quotes from articles, textbooks and online material must be properly referenced. Submit one report per student.

Report length:
Minimum 3 standard pages report (12 pts, single spaced minimum 36 lines, letter-size paper). The report must start with a short abstract explaining what is included in the report and end with your own conclusions (or evaluation) of the topics covered.

We suggest that you start with searching the keywords and topics related to software metrics in order to familiarize yourself with the terminology. Some keywords to start with are: function point, productivity, effort estimation, resource estimation, COTS evaluation, TMM (test maturity level), complexity metrics, object-oriented metrics, reverse engineering metrics, market/customer oriented metrics, product quality metrics, reliability and testing metrics, process improvement metrics, performance metrics, metric suites and tools, metrics for agile processes, security metrics.

A good starting point article is:

- *Stan Rifkin*, "What makes measuring software so hard?" IEEE Software, May/June 2001, Vol. 18, No. 3, pp. 41-45. Available: http://computer.org/software/homepage/2001/0301rifkin.htm

The following archive may be found useful:

- Thomas Fetcke's Software Metrics Sites: http://user.cs.tu-berlin.de/~fetcke/metrics-sites.html

Other useful links and articles are given on the course web page.

2. Part 2: Role playing and active discussion

Suppose that you are going to recommend buying a copy machine to your company for system-wide use. You have devised two copy machines with the specifications given below.
Your task is to advise the decision making committee which one is better. Write about 2 pages of report using the measurement concepts and metrics terminology (i.e., Lecture notes 1st and 2nd week, Chapters 1-2 of the recommended Textbook) to compare and recommend/not recommend one and explain why.

Machine 1:

> A high-speed black and white laser printer with 640MHz processor, 128Mbyte memory, 100Mbyte hard disk, 20ppm printing speed, 100 base-T network adapter Card. The toner costs about $200 for 4,000 pages output. 200 pages optional paper tray. Maker's data says that the system should run without failure for 4,000 pages. Price $900.

Machine 2:

> A low-speed colour laser printer with 128MHz processor, 32Mbyte memory, 100Mbyte hard disk, 8ppm printing speed, network adapter Card is optional $300. The toner costs about $500 for 4,000 pages output. 100 pages optional paper tray. Maker's data says that the system should run without failure for 4,000 jobs. Price $1,200.

You must come up with a crisp suggestion that the manager will base her/his decision on it. Using models, graphs and/or tables to back up the argument are considered better than reporting in narrative style. Some useful hints are:

- Defining the scales for the attributes that are going to be compared, mapping the measured (given) values to the scales and discuss the properties of the scales.
- Defining a model for the requirement assessment from both the manager and user viewpoints.
- Issues with goal, quality and compatibility of the parameters, such as developing a GQM model.
- Defining a quality model based on the attributes and trying to map the measured (given) values to the model.

3. Deliverables:

Two parts of reports as specified above.

Note that the reports must be delivered by the due date in both printed and electronic form. Electronic version of assignments must be uploaded to the Blackboard and delivered together with the printed ones. Notice that electronic assignments **WILL NOT** be accepted if delivered via email.

4. Evaluation:

The report will be marked as follows:

1. Contents of the document : 70%.
   How well have you described the topic?
   How well have you described the underlying concepts, definitions?
   Did you separate the authors' ideas and those of yours clearly?

2. Presentation style and referencing : 30%
   Do you have an abstract, introduction and conclusions?
   Do you have paragraphs with headings where appropriate?
   Do you have references in the text?
   Do you have a reference list at the end with links to papers, books or URLs where appropriate?
   Do you have an acknowledgements section if you got any help from anyone?

Notes:

1. Submit one report per student.
2. Write your own answers to the above questions. Quotes from articles, textbooks and online materials must be properly referenced. Reports are evaluated comparatively with the current and previous years' reports and other online materials. In case of proven plagiarism the assignment will be marked zero and suspected cases will be reported for further investigation. Please read the section in the University Calendar on Plagiarism/Cheating/Other Academic Misconduct.
   (http://www.ucalgary.ca/pubs/calendar/2008/how/How_MB.htm)

Assignments no. 2 and 3 (combined)

Deadline: October 28, 2008 (Tuesday)
Grade: %40 of the total lab marks

This assignment is a combined team assignment. The team is composed of 3 to 4 members.

1. Purpose: Reinforcement of the theoretical subjects studied and Teamwork

In this assignment you are asked to apply the 10 steps of the GQ(I)M approach to goal driven measurement for a realistic software development project. This will help you reinforce the concepts studied during the course.

2. Scenario

(a) You are a project manager in a software development company.
Your prime concern with the developed software by your group is one of the followings:

1. Functions and documents are not delivered in time.
2. The actual spending surpasses the estimated budget.
3. The customer's requirements are fragile and the development team should be able to cope with this.
4. The customers seem not fully satisfied with the software.
5. The customers seem not fully satisfied with the after services and maintenance.
6. Checking effectiveness of increasing the portion of software developed outside your group.
7. Minimize defects in the developed software.
8. Evaluating whether to employ cleanroom development in your software product.
9. Evaluating the productivity of the members of your development team.

You as a project manager need a set of measures that will help your project make progress toward this goal.

(b) You are an executive in a software development company.
Your prime concern with the products developed by your company is one of the followings:

(1) Enhance...
- Quality
- Customer Satisfaction
- Efficiency
- Productivity
- Ability
- Performance
- Motivation

(2) Reduce...
- Defects
- Effort

Time-To-Market
Delay
Cost
Complexity
Timeliness
Size

(3) Reach...
CMM Level 2 or 3 or 4 or 5
ISO 9001 certification
ISO/IEC 15504 Level 2 or 3 or 4 or 5

You as an executive need a set of measures that will help your company make progress toward this goal.

You should proceed as follows:

1. Select your team members (3 to 4 members)
2. Select one of the goal topics from either group (a) or (b)
3. Register your team members and the topic
4. Follow the 10 GQ(I)M steps and use the templates and worksheets provided in:

   R.E. Park, W.B. Goethert, W.A. Florac, "Goal-Driven Software Measurement - A Guidebook", CMU/SEI-96-HB-002.

   to produce a full report on how to devise a measurement system to achieve such goals.

The entire Guidebook and templates for worksheets are downloadable from the SENG 421 course WWW site.

You may want to start with reading relevant sections of the *Goal-Driven Software Measurement - - A Guidebook* (189 Pages) and follow the steps mentioned there. Many of the concepts can only be comprehended and decided through the discussion among the team members.

Some sample projects are posted on the course web page.

One of the difficulties that you may face is that each of the GQ(I)M steps (specially steps 6-10) may require details that may only be available in a real situation. It may give the students too much freedom to assume the details and proceed. You may want to discuss some of the assumptions with the instructor or TAs in the review lab sessions.

You should start with a business goal (Step 1), which is the title of the project you have selected. In Step 2 (identify what we want to know) try to ask 5-6 questions for the *entities of interest* (i.e., inputs and resources; internal artefacts; activities and flows; products and by-products) and then group the questions that address a common entity to convert them to a few (about 3-4 or more) subgoals (Step 3). In order to convert the subgoals to measurement goals you need to define *entities* and *attributes* for each question listed for each subgoal. This may be a long and repetitive

task. In Step 5 (formalizing measurement goals) you must define the Purpose-Perspective-Environment and Constraints trio for each *object of interest*. There may be 10-20 or more of the objects of interest, depending on the project. In Step 6 (Identifying Quantifiable Questions and Indicators) you may derive questions for all measurement goals but proceeding to the indicators for only one of the measurement goals will be sufficient. In Step 7 you should prepare a cross-reference checklist for the data elements to be collected and the indicators identified in step 6. In Step 8 you formalize the measures by defining scale, range and precision for all the entities and prepare the definition checklist for them. Step 9 is about *analysis*, *diagnosis* and *actions*. You may need to consider the actions only for new projects. Finally, in Step 10 you must fill in the measurement plan template that will serve as your final report for this project together with the output documents for each step of the GQIM process.

3. Deliverables:

Your deliverable (report) should consist of:

1. Title page (project title, delivery data, team members' name, student ID and email)
2. One page executive summary.
3. Report on the measurement system to achieve the goal of the project (no minimum/ maximum page limit. Enclose templates for each of the 10 steps. Start with the goal and questioning how to achieve the goal, goal decomposition and proceed to the later steps. Follow guidelines given during the 3rd week of the course. If there are too many subgoals, you should proceed to the end with at least one of the subgoals. If you may not be familiar with the measured data elements yet you may refer to the recommended textbook for the course Ch 7-11.)
4. One page project self assessment (fill-in form enclosed).
5. One page including
    a. Conclusions related to the project (summarizing the results)
    b. Constructive comments related to this assignment (your own viewpoint related to the assignment, how it can be improved, etc.)

4. Evaluation:

Evaluation of the reports is based on the effort the team has put into the project and is measured by "originality", "correctness" and "completeness" of the project.

Notes:

1. Submit one report per team.
2. Write your own answers to the above questions. Quotes from articles, textbooks and online materials must be properly referenced. Reports are evaluated comparatively with the current and previous years' reports and other online materials. In case of proven plagiarism the assignment will be marked zero and suspected cases will be reported for further investigation. Please read the section in the University Calendar on Plagiarism/Cheating/Other Academic Misconduct.
   (http://www.ucalgary.ca/pubs/calendar/2008/how/How_MB.htm)

Team members (name, ID, email):

1.
2.
3.
4.

Project Title:

Please give your own evaluation of your project by marking one and only one of:

A (we did a very excellent job)
B (we did a satisfactory job)
C (we could manage somehow to finish the job)
D (we had no idea and probably we did not do well)
E (we did not finish this step)

for each of the steps 1-10 together with an overall assessment at the end.

	Step	Self Assessment				
1	Identifying business goals	A	B	C	D	E
2	Identifying what you want to know or learn in order to achieve the goals	A	B	C	D	E
3	Identifying subgoals	A	B	C	D	E
4	Identifying entities and attributes related to subgoals	A	B	C	D	E
5	Formalizing measurement goals, in terms of purpose, perspective and environment	A	B	C	D	E
6	Identifying questions and indicators that will be used to achieve measurement goals	A	B	C	D	E
7	Identifying data elements that will be collected to construct the indicators	A	B	C	D	E
8	Defining measures to be used, and make these definitions operational	A	B	C	D	E
9	Identifying actions that will be taken to implement the measures	A	B	C	D	E
10	Preparing a plan for implementing the measures	A	B	C	D	E
	Overall assessment	A	B	C	D	E

Note. If you give yourself an E on each item above give us the reason in a few lines.

Assignment no. 4

Deadline: November 18, 2008 (Tuesday)
Grade: %20 of the total lab marks

This assignment is a group assignment. The team is composed of 3 to 4 members. The team members and groups must be as of the Assignment 2-3 in order to assess the improvement of the teamwork.

1. Purpose: Measuring Function Point

In this assignment you are asked to measure the function point for a typical software system. This will help you reinforce the concepts studied during the course.

2. Background:

The overall objective is to determine adjusted function point count for a software system. There are several steps necessary to accomplish this. The actual sequence or order of steps is not necessary. Many counters will complete step 5 through out the entire count – gathering information as they go;

1. Determine type of function point count
2. Determine the application boundary
3. Identify and rate transactional function types to determine their contribution to the unadjusted function point count.
4. Identify and rate data function types to determine their contribution to the unadjusted function point count.
5. Determine the value adjustment factor (VAF)
6. Calculate the adjusted function point count.

The unadjusted function point (UFP) count is determined in steps 3 & 4. It is not important if step 3 or step 4 is completed first. In GUI and OO type applications it is easy to begin with step 3.

The final function point count (adjusted function point count) is a combination of both unadjusted function point count (UFP) and the general system characteristics (GSC's).

3. Scenario:

Suppose that your group is a software development team assigned to build one of the projects listed on suggested project list (Appendix Page). You are supposed to measure the Function Point for this project.

You may proceed as follows:

1. Start with defining the requirements for your project. Note that you should proceed to the extent that the detailed requirements are sufficient to measure the function points.

2. Read relevant sections of the "Function Points Analysis Training Course" (110 Pages) (by David Longstreet, David@SoftwareMetrics.com, www.SoftwareMetrics.com). This document is downloadable from course web page. The document is a step-by-step guide to measure FP. Follow the steps mentioned there. Many of the concepts can only be comprehended and decided through the discussion among the team members.

4. Deliverables:

Your deliverable (report) should consist of:

1. Title page (project title, delivery date, team members' name, student ID and email).
2. One page executive summary.
3. Project specification (1-2 pages) clearly mentioning what is included (system functions, modules) and support systems (DB server, Web server, etc., if needed)
4. FP measurement results for the different modules and overall project (no page limit).
5. One page of conclusions and comments including
    a. Conclusions related to the project (summarizing the results)
    b. Your comments related to this assignment (Your own viewpoint related to the assignment. What do you think about this assignment? Was it useful? How can it be improved? etc.)

5. Evaluation:

Evaluation of the reports is based on the effort the team has put into the project and is measured by "originality", "correctness" and "completeness" of the project.

Notes:

1. Submit one report per team.
2. Write your own answers to the above questions. Quotes from articles, textbooks and online materials must be properly referenced. Reports are evaluated comparatively with the current and previous years' reports and other online materials. In case of proven plagiarism the assignment will be marked zero and suspected cases will be reported for further investigation. Please read the section in the University Calendar on Plagiarism/Cheating/Other Academic Misconduct.
   (http://www.ucalgary.ca/pubs/calendar/2008/how/How_MB.htm)

Assignment no. 5

Deadline: December 2, 2008 (Tuesday)
Grade: %20 of the total lab marks

This assignment is a group assignment. The team is composed of 3 to 4 members. The team members and groups must be as of the Assignment 2-3 and 4 in order to assess the improvement of the teamwork.

1. Purpose: Measuring Effort Using Software Metrics Tools

In this assignment you are asked to measure software cost and effort for a realistic project using COCOMO II tool. This will help you reinforce the concepts studied during the course.

2. Tools:

The tool based on Constructive Cost Model (COCOMO II).

3. Background:

COCOMO II includes three-stage series of models:

1. The earliest phases will generally involve prototyping, using the *Application Composition model* capabilities.
2. The next phases will generally involve exploration of architectural alternatives or incremental development strategies. To support these activities, COCOMO II provides an early estimation model called the *Early Design model*.
3. Once the project is ready to develop, it should have a life-cycle architecture, which provides more accurate information on cost driver inputs, and enables more accurate cost estimates. To support this stage, COCOMO II provides the *Post-Architecture model*.

The *Application Composition model* is used in estimating early stage issues, where source code is not available. It uses counts entities such as user interfaces, software/system interaction to measure Object Points (OP) and then use the following relation to derive the effort (E):

$$E = OP / PROD$$

where

OP is the object point
PROD is the productivity rate defined below

Developers' experience and capability	Very Low	Low	Nominal	High	Very High
PROD	4	7	13	25	50

The *Early Design model* is used to evaluate alternative software/system architectures and concepts of operation. An unadjusted function point count (UFC) is used for sizing. This value is converted to KLOC. The Early Design model equation is:

$$E = 2.45 \times KLOC \times EAF$$

The KLOC may be computed using the COCOMO II tool or by converting the object point to KLOC.

The effort adjustment factor (*EAF*) is calculated as using 7 cost drivers.

	Cost Driver	Description	Counterpart Combined Post-Architecture Cost Driver
1	**RCPX**	Product reliability and complexity	**RELY, DATA, CPLX, DOCU**
2	**RUSE**	Required reuse	**RUSE**
3	**PDIF**	Platform difficulty	**TIME, STOR, PVOL**
4	**PERS**	Personnel capability	**ACAP, PCAP, PCON**
5	**PREX**	Personnel experience	**AEXP, PEXP, LTEX**
6	**FCIL**	Facilities	**TOOL, SITE**
7	**SCED**	Schedule	**SCED**

The *Post-Architecture model* is used during the actual development and maintenance of a product. The Post-Architecture model includes a set of 17 cost drivers and a set of 5 scale factors determining the projects scaling component. The Post-Architecture model equation is:

$$E = 2.45 \times \left(KLOC\right)^{b} \times EAF \qquad b = 0.91 + 0.01 \sum_{j=1}^{5} SF_i$$

The *SF* and *EAF* can be calculated using the default values given by the COCOMO II tool.

Additional documents related to COCOMO II tool and user manual can be downloaded from the following URLs:

1. Original manuals and documets for COCOMO Project (document repository)
   http://sunset.usc.edu/research/COCOMOII/index.html
2. COCOMO II Model Manual (local copy)
   http://www.enel.ucalgary.ca/People/far/Lectures/SENG421/PDF/COCOMO/modelman.pdf
3. COCOMO II User Manual (local copy)
   http://www.enel.ucalgary.ca/People/far/Lectures/SENG421/PDF/COCOMO/userman.pdf

4. Scenario:

Suppose that your group is a software development team assigned to build one of the projects listed on suggested project list (Appendix Page). You are supposed to measure the total effort required to build the project.

You may proceed as follows:

1. Start with defining the requirements for your project. Note that you should proceed to the extent that the detailed requirements are sufficient to measure the object points (or function points).
2. Read the COCOMO II user manual (available as a help file on the Windows system and also as a separate PDF file) and run the tool. Start with defining a new project and define modules, that you defined in Step 1, and proceed through the 3 phases to refine your estimation of effort. Note that sometimes you may need to revise earlier stage decisions and repeat estimation to avoid conflicts.
3. Try to change a few parameters (like project duration, risk, etc.) and how they may affect your original estimations.
4. Use the reporting tool of COCOMO II to generate report pages.

5. Deliverables:

Your deliverable (report) should consist of:

1. Title page (project title, delivery date, team members' name, student ID and email).
2. One page executive summary.
3. Project specification (1-2 pages) clearly mentioning what is included (system functions, modules) and support systems (DB server, Web server, etc., if needed). This can be the same as Assignment 4.
4. Estimation results for the 3 phases of the model (no page limit).
5. One page of conclusions and comments including
    c. Conclusions related to the project (summarizing the results)
    d. Your comments related to this assignment (Your own viewpoint related to the assignment. What do you think about this assignment? Was it useful? How can it be improved? etc.)

6. Evaluation:

Evaluation of the reports is based on the effort the team has put into the project and is measured by "originality", "correctness" and "completeness" of the project.

Notes:

1. Submit one report per team.
2. Write your own answers to the above questions. Quotes from articles, textbooks and online materials must be properly referenced. Reports are evaluated comparatively with the current and previous years' reports and other online materials. In case of proven plagiarism the assignment will be marked zero and suspected cases will be reported for further investigation. Please read the section in the University Calendar on Plagiarism/Cheating/Other Academic Misconduct.
   (http://www.ucalgary.ca/pubs/calendar/2008/how/How_MB.htm)

Appendix: Suggested Project List (SENG 421)

Below is a list of suggested projects for SENG 421. This list may be used as a reference or as the last choice. **You are encouraged to define your own project (students may receive additional 2% bonus points for their originality of the project)**.

1) Transit tracking system
Description: Calgary's Transit System can help a lot of student's from freezing out in the winter. Currently the website has a scheduling system that probably works out of a database backend and shows the timings of a bus at different tops. Your job is to write an application that will give "real time data" as to where the bus is at any given time. It would be nice if the program could be intelligent enough to know how long it takes the bus to reach from one stop to another given past experiences.

2) Gas price tracker
Description: With the high gas prices a lot of websites are posting gas prices across Calgary. These prices are basically entered by individual users. When a user goes to such a website the site lists the highest and lowest 10 prices of the day. However this does not say much. Users want to know how much they are saving. If the user has to drive 30 km to get the lowest price it may or may not be worth it. It would be very helpful if the user could get a more intelligent answer as to how much will he/she will save going to that gas station from the user's origin.

3) Document management system
Description: Executives in companies share a lot of documents. Documents that were written and viewed years ago sometimes need to come alive when someone needs to view it. Generally the practice is to keep these documents on a share drive on the network. However when documents are saved in the same place and the share drive grows, the task of browsing though all the documents one by one is a very hard one. It would save executives a lot of agony if they could somehow search for documents. Keep in mind the possibility of overwriting of files by different users.

4) Course management system for students
Description: Some students are not sure whether or not they can handle a certain course load given their schedule and other commitments. Generally when they sign up for courses all it says is when the course is and how long it lasts and whether or not it conflicts with other courses that he/she is signed up for. It would help if the application is intelligent enough to make suggestion as to whether a certain course load will be difficult, easy, impossible etc. A lot of factors can be considered. It is up to the creative programmer to decide which factors should be used and what kind of information should be presented to the user.

5) Hotel reservation system
Description: Write a specification for a hotel booking system which keeps track of current guests, available rooms, future bookings and payment of accounts. Include a part for room

service orders which keeps track of the orders, bills them to the appropriate room, and will show the total sales for a given period.

6) Hospital
Description: Write a specification to manage a hospital system. There will be a waiting list of patients needing different treatment e.g. surgical, medical. The bed state should be determined and if beds are available, the next appropriate patients on the list notified. Nurses should be allocated to wards depending on ward sizes, what type of nursing is needed, operating schedules etc.

7) Library
Description: Write a specification to organize the lending of library books. Only so many may be borrowed, and if a user has the maximum one must be returned before another is borrowed. Books are subject to recall. The librarian will have special borrowing privileges. There may be several copies of a particular book on the shelf, or they may all be out on loan.

8) Online registration system
Description: Write a specification for an online registration system in which students could add courses, drop courses, view their courses, view all courses, look up their grades for courses, etc. Teachers could view the students in their courses, and assign grades online. Administrators could create, read, update and delete course info, including assigning teachers to courses, assigning courses to classrooms, etc. The system could have some intelligence, such as denying registration to students with unpaid fees, doing prerequisite checks, etc.

9) Realtor system
Description: Write a specification for a system to manage a real estate catalogue. It should keep track of properties with their amenities, selling price or rental etc, and be able to be queried by potential customers with different priorities. Take into account constraints on the property such as offers made, terms and conditions, etc.

10) Electronic shop
Description: Write a specification for an electronic shop that allows several vendors set it up and sell their goods to several customers on the Web.

11) Software Design documents management system
Description: Write a specification for a software design documentation repository system for saving several forms of documents from various phases of a software development project and indexing them for future use, using data-centered structure. The documents may be either unstructured (pure text) or structured (Word, PDF, Excel, etc.). The system should answer to queries from users via intranet.

12) Other
Define your own project. Consult with course instructor and/or TAs regarding the depth and scope of the project and obtain their approval.