

A Software Cost Estimation for the Legal Practice System

Table of Contents

TABLE OF CONTENTS.....	TABLE OF CONTENTS
INTRODUCTION.....	INTRODUCTION
REQUIREMENT CLASSIFICATION.....	REQUIREMENT CLASSIFICATION
INFLUENCING FACTORS.....	INFLUENCING FACTORS
FUNCTION POINT SCORES	FUNCTION POINT SCORES
TIME ESTIMATION	TIME ESTIMATION
CONCLUSION	CONCLUSION

Introduction

The purpose of this document is to produce an estimate of how long it will take to deliver the automatic assessment system for the Legal Practice Centre. This is to allow our group to get an idea of whether we are on course to finish the project when planned and also to get an idea of how much extra time will be involved with adding on the desirable and optional functional requirements of the system.

The software cost estimation detailed in this document uses the function point method. In this, the effort required to create a piece of software is related to the complexity and size of the system itself. It is based on looking at the functional requirements of the system and by assessing the size and complexity of each one. This technique allows us to get time estimates before any code has been written, which is highly advantageous.

Each requirement is placed in a category, which relates to what the requirement is about (e.g. inputting data or an inquiry). Using factors defined for each category, it is estimated whether each requirement is simple, average or complex to accomplish. Weights are defined for each category and complexity classification and these are summed over all requirements.

Lastly, there are several factors related to the project as a whole which affect how long requirements take to accomplish in a system, such as how decentralised the data processing is and how adaptable the system must be. From the extent of influence of these other factors, an influence factor score is calculated and this is used to adjust the function point score to a more accurate value.

This document is split into a number of parts. Firstly, the classification of the functional requirements in the requirements document is shown together with a table showing the numbers of requirements that fall into each type. Then, the influencing factors are detailed.

Thirdly, tables are shown presenting the function point scores that are produced assuming that:

Only the mandatory requirements are implemented.

The mandatory and desirable requirements are implemented.

The mandatory, desirable and optional requirements are implemented.

After this, empirical data from IBM will be used to try to relate the number of function points to how long it should take us to produce the software with each of the three different scenarios concerning how much of the system we implement. In the last section the implications of the results concerning our group's time management will be explored.

Requirement Classification

The table below lists each functional requirement of the automatic assessment system along with what category the requirement is of and its classification. The category can either be:

- Input data
- Inquiries
- Output data
- Databases
- Reference data

There was uncertainty at many points as to which category to place a particular requirement in. For example, all inquiries would retrieve data from the database and data that was input would then get stored in the database. However, in the end the category chosen was the one that was thought as the most suitable and the same decision method was used for all of the requirements to try to ensure consistency regarding choosing categories for requirements.

The classification column uses information about different criteria for that category, such as the number of different data elements for the input category, in order to estimate how complex the requirement will be to create in a system. These criteria and the way in which they map onto the different classifications was done using tables provided by Dr Gerald Luetgen.

There were a few requirements, such as those relating to particular hardware configurations or concurrency issues, that did not seem appropriate to be placed into one of the five categories. This is because they did not fit into any of the categories and they were more to do with conditions the system must run under rather than things it must do. These requirements have been omitted from the following table.

Req. ID	Description	Category	Classification
1	Lecturers can add topics.	Input data	average
2	Lecturers can remove topics.	Input data	simple
3	Lecturers can edit details of a topic.	Input data	average
4	Lecturers can view details of a topic.	Inquiries	simple
5	Lecturers can add a student.	Input data	average
6	Lecturers can remove a student.	Input data	simple
7	Lecturers can edit details of a student.	Input data	average
8	Lecturers can view details of a student	Inquiries	simple
9	Lecturers can easily remove all students from the system and the end of each year.	Input data	simple
10	The system stores which topics each student is studying, which are entered by the lecturer.	Input data	simple
11	The compulsory topics are automatically entered into each student's record.	Input data	simple
12	Lecturers can add topics to a student's record.	Input data	simple

13	Lecturers can remove a topic from a student's record.	Input data	simple
14	Lecturers can add exercises to a topic.	Input data	simple
15	Lecturers can edit exercises.	Input data	simple
16	Lecturers can remove exercises.	Input data	simple
17	Lecturers can view the questions in an exercise.	Output data	average
18	Lecturers can add questions to an exercise.	Input data	complex
19	Lecturers can edit questions.	Input data	complex
20	Lecturers can remove questions.	Input data	simple
21	Lecturers can enter a comment to be displayed for each answer to a multiple-choice question.	Input data	simple
22	A lecturer should be able to control when students can attempt an exercise. A lecturer should be able to 'publish' an exercise when he or she has finished creating all of the questions for it.	Input data	complex
23	New types of questions can be added to the system without any of the original system needing to be modified.	Input data	average
24	Students enter their unique username in order to use the program.	Input data	simple
25	Students must be able to select an exercise and then attempt it.	Input data	average
26	When a student gets a question right, a comment is displayed (if one was entered).	Output data	simple
27	When a student gets a question right, some kind of "reward" is displayed.	Output data	average
28	When a student gets a question wrong, the system informs them that they got it wrong.	Output data	simple
29	When a student gets a question wrong, the system shows a comment (if one was entered).	Output data	simple
30	The student proceeds to the next question after being told if they got the current question right or wrong.	Output data	simple
31	At the end of the exercise, the system displays the student's score and a list of the questions that the student got wrong.	Output data	simple
32	When the student has attempted all questions, but not got all of them right, the student can retake only the questions that they got wrong.	Output data	complex
33	When a student has got all of the questions in an exercise right, they can retake the entire exercise.	Output data	complex
35	The first time a student answers an exercise, their score in it is stored.	Input data	average
36	Lecturers can look at a list of which exercises a particular student has attempted and the marks they got for those exercises.	Inquiries	simple

37	Lecturers can look at a list of which students have attempted a particular exercise and each students' score for that exercise.	Inquiries	average
38	Lecturers can look at a list of which students have not completed a particular exercise.	Inquiries	average
39	Lecturers can print out a list of which exercises a particular student has attempted and the marks they got for those exercises.	Output data	simple
40	Lecturers can print out a list of which students have attempted a particular exercise and each students' score for that exercise.	Output data	simple
41	Lecturers can print out a list of which students have not completed a particular exercise.	Output data	simple
44	The system should only allow students on the Legal Practice Centre course and lecturers teaching on the Legal Practice Centre course to access the system.	Input data	simple
45	Lecturers have to enter a password in order to use the parts of the program relating to altering student details, topic details and exercises.	Input data	simple
46	Students also have to enter a password in order to use the program.	Input data	simple

The table below summarises the results of the above table in terms of how many requirements fall into which categories and classifications. It also makes the distinction between whether or not these requirements were mandatory (M), desirable (D) or optional (O). For details of which particular requirements fall into these categories, please refer to the requirements document.

In the table below the columns 'M', 'D' and 'O' show the number of requirements of that category and classification that are mandatory, desirable and optional respectively. The column 'M + D' is the sum of the 'M' and 'D' columns and the column 'M + D + O' is the sum of the 'M', 'D' and 'O' columns.

Category	Classification	M	D	O	M + D	M+D+O
Input data	simple	7	8	1	15	16
	average	5	1	1	6	7
	complex	1	2	0	3	3
Inquiries	simple	2	1	0	3	3
	average	0	2	0	2	2
	complex	0	0	0	0	0
Output data	simple	5	3	0	8	8
	average	0	1	1	1	2
	complex	0	2	0	2	2
Databases	simple	0	0	0	0	0

	average	0	0	0	0	0
	complex	0	0	0	0	0
Reference data	simple	0	0	0	0	0
	average	0	0	0	0	0
	complex	0	0	0	0	0

Influencing factors

Below are the list of influencing factors that could either make the requirements easier to implement or harder to implement depending on how much influence these factors have on the system.

Factor	Score
1) Interplay with other application systems (0-5)	0
2) Decentralised data (processing) (0-5)	2
3) Transaction rate (0-5)	2
4) Processing	
(a) Calculations/arithmetic (0-10)	3
(b) Control (0-5)	2
(c) Exceptions (0-10)	4
(d) Logic (0-5)	2
5) Re-usability (0-5)	1
6) Data conversions (0-5)	1
7) Adaptability (0-5)	1

Function Point Scores

On the next three pages are tables showing the function point scores for the following three scenarios:

Only the mandatory requirements are implemented.

The mandatory and desirable requirements are implemented.

The mandatory desirable and optional requirements are implemented.

As it can be seen, the number of function points more than doubles with the desirable requirements added to the system and it increases to a slightly higher number when the few optional requirements are included as well. The next step is to translate these function point scores into a more meaningful form, and this is done in the next section.

Function point score for mandatory requirements:

Category	Number	Classification	Weight	SUM(FPs)
Input data	7	simple	3	21
	5	average	4	20
	1	complex	6	6
Inquiries	2	simple	3	6
	0	average	4	0
	0	complex	6	0
Output data	5	simple	4	20
	0	average	5	0
	0	complex	7	0
Databases	0	simple	7	0
	0	average	10	0
	0	complex	15	0
Reference data	0	simple	5	0
	0	average	7	0
	0	complex	10	0
Sum				73
Influencing factors	1) Interplay with other application systems			0
	2) Decentralised data (processing)			2
	3) Transaction rate			2
	4) Processing			
	(a) Calculations/arithmetic			3
	(b) Control			2
	(c) Exceptions			4
	(d) Logic			2
	5) Re-usability			1
	6) Data conversions			1
7) Adaptability			1	
Sum (1-7)				18
Influence factor				0.88

Adjusted function point count	64.24
--------------------------------------	--------------

Function point score for mandatory and desirable requirements:

Category	Number	Classification	Weight	SUM(FPs)
Input data	15	simple	3	45
	6	average	4	24
	3	complex	6	18
Inquiries	3	simple	3	9
	2	average	4	8
	0	complex	6	0
Output data	8	simple	4	32
	1	average	5	5
	2	complex	7	14
Databases	0	simple	7	0
	0	average	10	0
	0	complex	15	0
Reference data	0	simple	5	0
	0	average	7	0
	0	complex	10	0
Sum				155
Influencing factors	1) Interplay with other application systems			0
	2) Decentralised data (processing)			2
	3) Transaction rate			2
	4) Processing			
	(a) Calculations/arithmetic			3
	(b) Control			2
	(c) Exceptions			4
	(d) Logic			2
	5) Re-usability			1
	6) Data conversions			1
7) Adaptability			1	
Sum (1-7)				18
Influence factor				0.88

Adjusted function point count	136.4
--------------------------------------	--------------

Function point score for mandatory, desirable and optional requirements:

Category	Number	Classification	Weight	SUM(FPs)
Input data	16	simple	3	48
	7	average	4	28
	3	complex	6	18
Inquiries	3	simple	3	9
	2	average	4	8
	0	complex	6	0
Output data	8	simple	4	32
	2	average	5	10
	2	complex	7	14
Databases	0	simple	7	0
	0	average	10	0
	0	complex	15	0
Reference data	0	simple	5	0
	0	average	7	0
	0	complex	10	0
Sum				167
Influencing factors	1) Interplay with other application systems			0
	2) Decentralised data (processing)			2
	3) Transaction rate			2
	4) Processing			
	(a) Calculations/arithmetic			3
	(b) Control			2
	(c) Exceptions			4
	(d) Logic			2
	5) Re-usability			1
	6) Data conversions			1
7) Adaptability			1	
Sum (1-7)				18
Influence factor				0.88

Adjusted function point count	146.96
--------------------------------------	---------------

Time Estimation

From the function points scores, empirical data from IBM was used to translate the function points into Person-Months (PMs). Some calculations were necessary in order to do this because the IBM figures were only stated explicitly for certain function point values. A Person-Month is a time equal to one person working on the system for a month. The translation of the three function point scores for our system is shown in the table below:

Scenario	Function points	Person-Months
Only the mandatory requirements are implemented.	64.24	3.24
The mandatory and desirable requirements are implemented.	136.4	8.44
The mandatory, desirable and optional requirements are implemented.	146.96	9.26

What these figures say is that it would take one person working on the system approximately 3.24 months to complete it. There are four people in our group so therefore it should take 0.81, 2.11 and 2.32 months respectively for each of the three scenarios of how much of the system is implemented. However, unlike employees of a software house, the members of our group have other modules to do and can therefore not work on the Legal Practice system from 9.00 a.m. to 5.00 p.m. every weekday.

Having looked back at the timesheets of the weeks that have so far passed, it seems that an average figure for how long this group spends on the Legal Practice system each week is around 45 hours. If a person worked from 9.00 a.m. to 5.00 p.m. every weekday with an hour for lunch then they would spend 35 hours working on a system per week so four people would spend approximately 140 hours. Therefore this group would spend only about 32% of the time on the system that a real software house (with four employees working on the system) would per week.

Therefore I believe that a more accurate set of figures for how long the entire system should take are as follows:

Only the mandatory requirements are implemented – 2.53 months.

The mandatory and desirable requirements are implemented – 6.59 months.

The mandatory desirable and optional requirements are implemented – 7.25 months.

Conclusion

Overall, it appears that if the time estimates are correct then our group will only have time to complete the mandatory requirements of the automatic assessment system. From Week 2 to Week 12 is 11 weeks and 2.53 months (how long it should take to implement just the mandatory functional requirements) is approximately 10 or 11 weeks. However, this does not include the three weeks for Easter, but group members would want some weeks to revise for other modules.

However, it is the author's belief that in actual fact the function point score is too high. The technique as a whole is very subjective and probably only becomes more accurate after a software house has used the technique on many projects and can calibrate the method using information on past projects in order to produce more accurate results.

Also, the way in which requirements are written down could alter how many requirements there are. Our group wrote the requirements document in a way so that each requirement was a single usage of the system and that even obvious things were covered. For example, Requirements 44, 45 and 46 could have conceivably been written as one requirement such as 'Lecturers and students should have to enter passwords to use the system'. Therefore, the way requirements are expressed can alter the number of requirements and therefore the final function point score.

As far as implementing the system goes, it is not believed that the findings of this report need have much influence on the current process our group is using to create the Legal Practice system. The incremental development model that we are using, which enables us to build a working system before adding on additional functionality, should enable us to produce a working system however many desirable and optional requirements we have time to realise.